

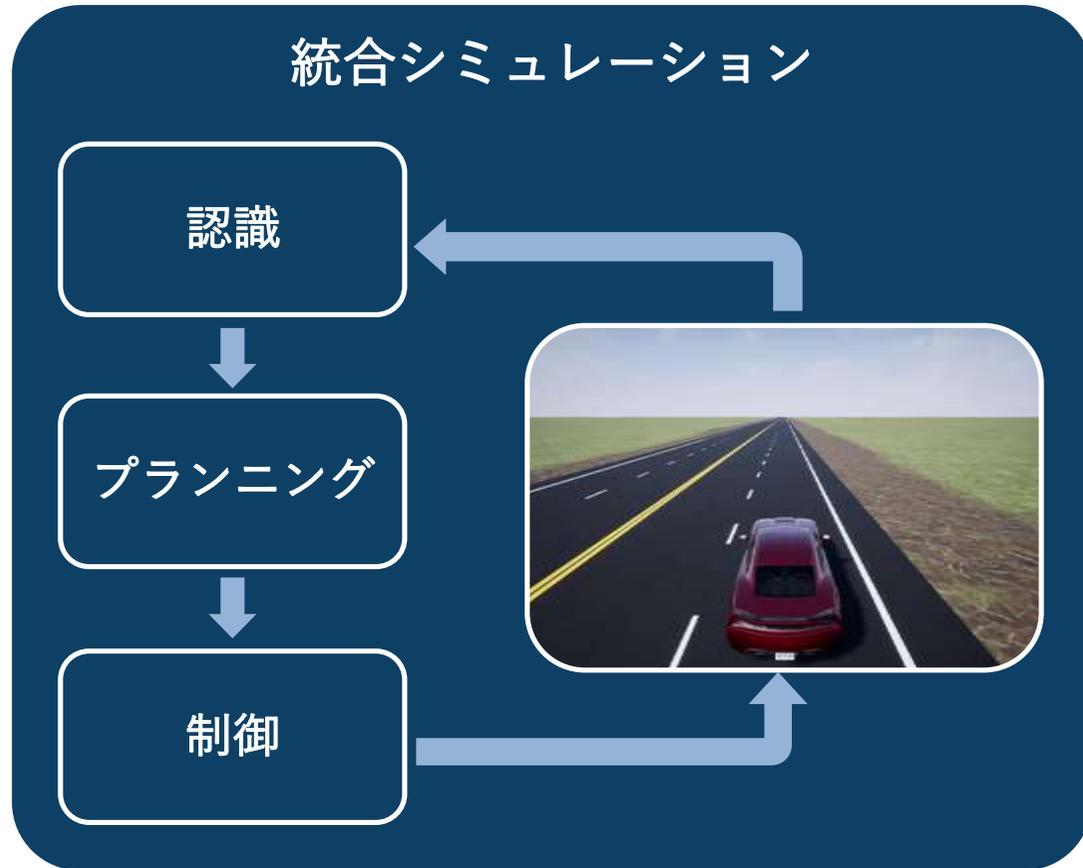
MATLAB EXPO 2019

ADAS・自動運転アルゴリズム検証のための
シナリオ生成とシミュレーション

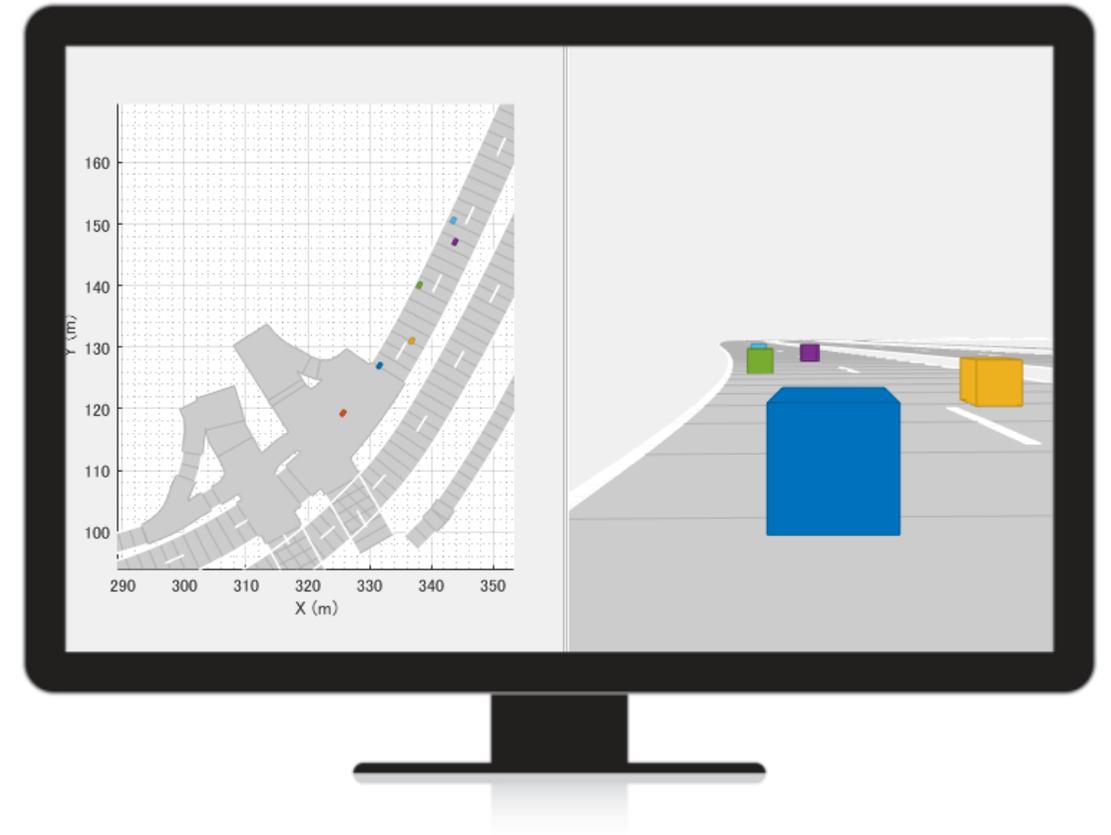
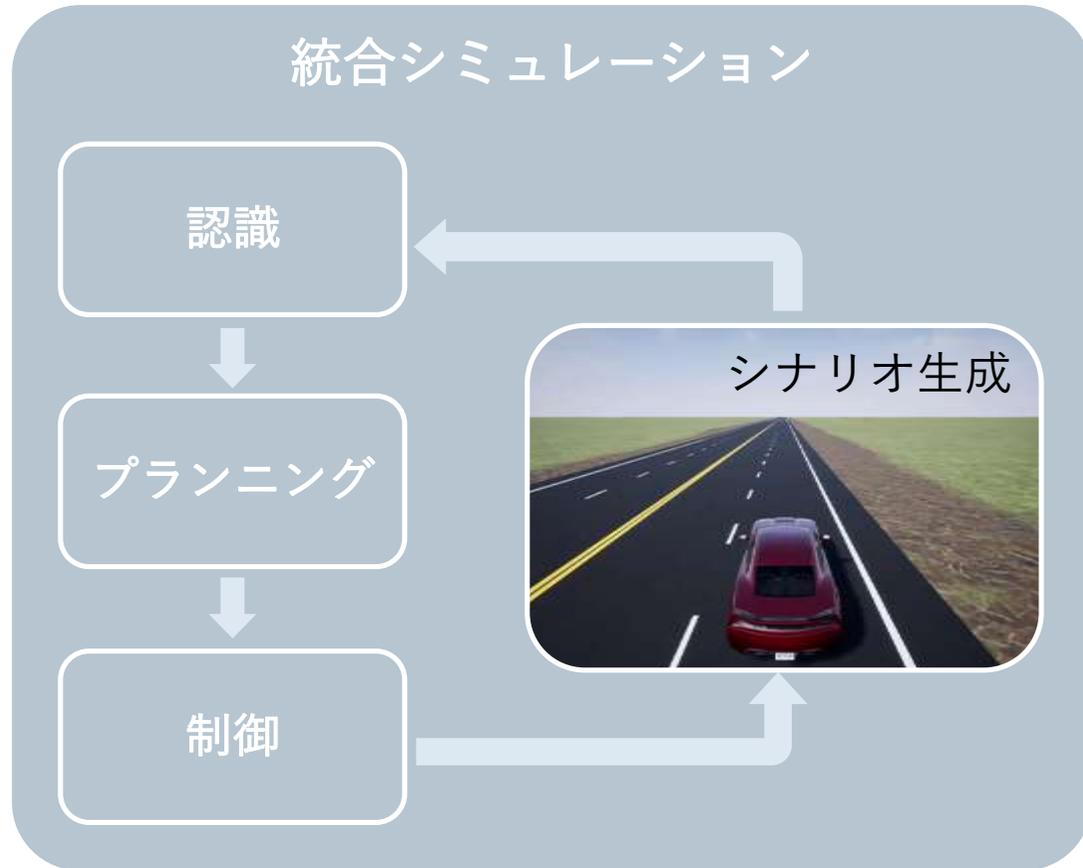
アプリケーションエンジニアリング部
大塚 慶太郎



ADAS・自動運転システムの開発 with MATLAB and Simulink



ADAS・自動運転：運転シナリオの定義 with MATLAB and Simulink



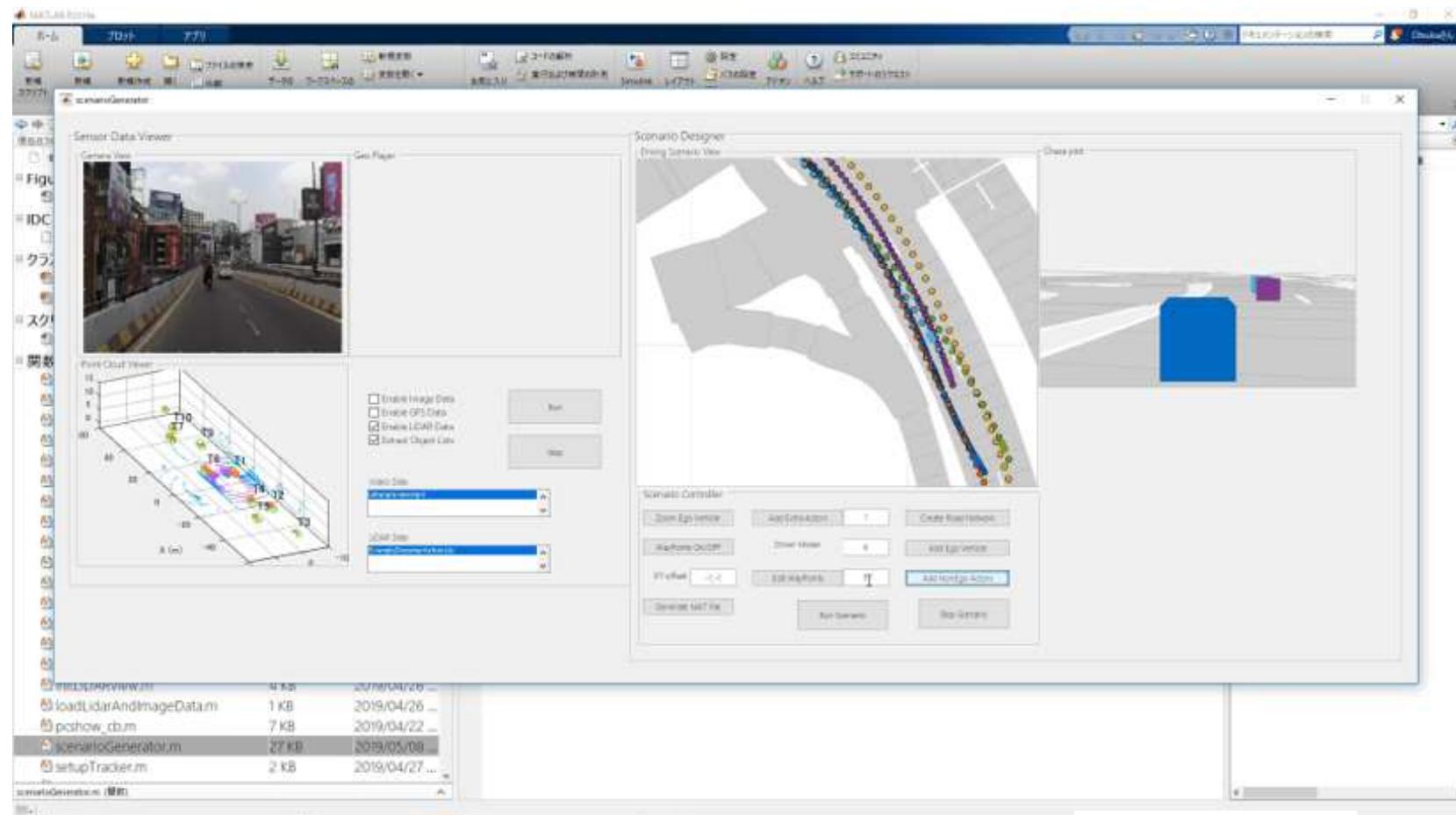
テスト車両からのデータを利用したシナリオ自動生成



カメラ
(動画像)

LiDAR

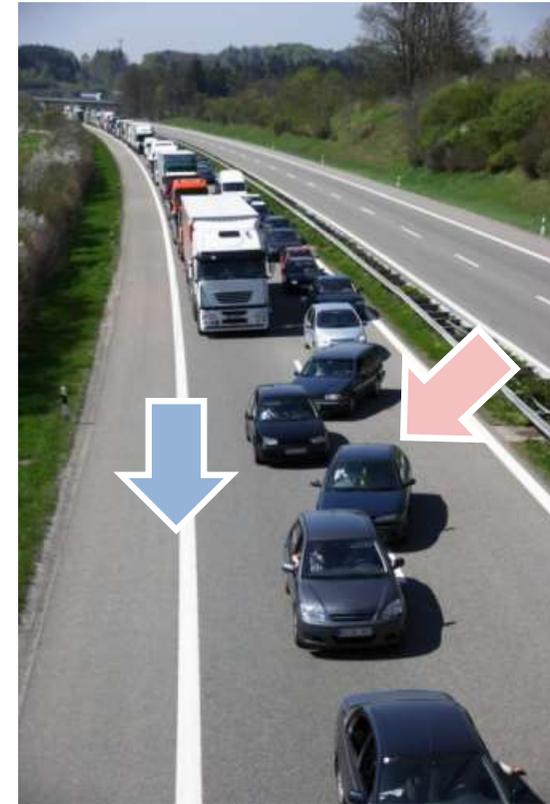
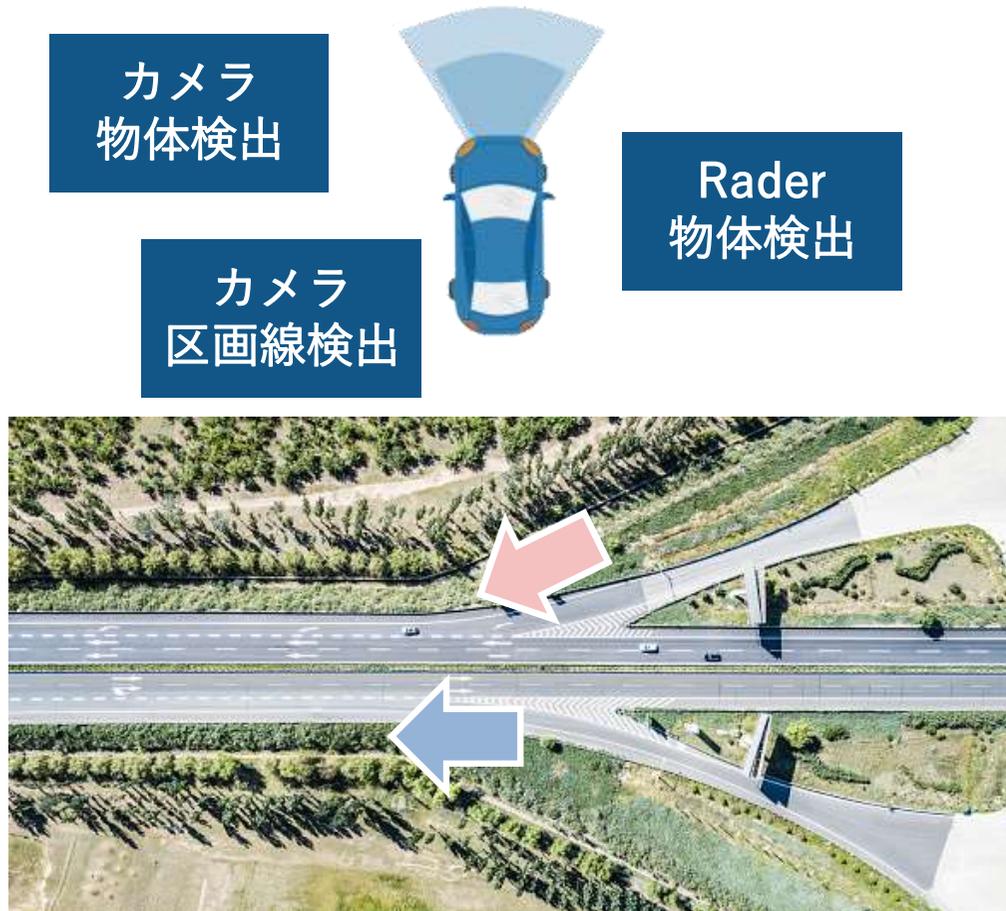
GPS



Agenda

- 運転シナリオ生成&シミュレーション概要
- Road Network作成
 - 地図データへのアクセス
 - 道路情報のインポート
- 交通参加者の定義
 - センサーデータ(LiDAR)の取り扱い
 - 点群からの物体認識ワークフロー
- シナリオ自動生成とその活用
- まとめ

運転シナリオシミュレーションの重要性



- ・ コーナーケースでのフュージョンアルゴリズム検証
- ・ 異なる特性のセンサの組み合わせ、配置位置の検証

- ・ より多くのテストケースバリエーションでの検証
- ・ 実車データ取得の難しい、危険なシナリオ

運転シナリオ生成&シミュレーションにおける困りごと

シナリオ作成自体が
タイヘン

もう少しシミュレーション
速度を改善したい

環境を増設したいが、
コストが気になる・・・

慣れているMATLAB/Simulink
環境で完結したい
(制御アルゴリズムと統合)

目的に応じて選択できる2つの環境

ゲーミングエンジン
連携



シナリオ
作成

Driving Scenario
Designer



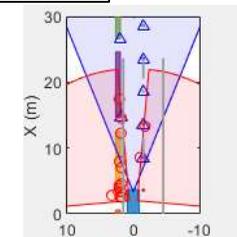
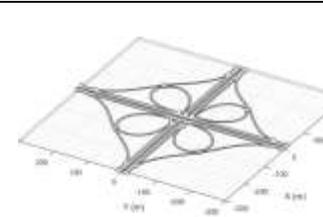
認識系アルゴリズム必要



より“リアル”な環境、
物理センサーモデル

- 認識系アルゴリズムを含む、より現実に近いシミュレーション
- 実行時間は低速、環境構築のため専門知識が必要

認識系アルゴリズム不要



抽象度の高い、簡易的表現
確率的センサーモデル

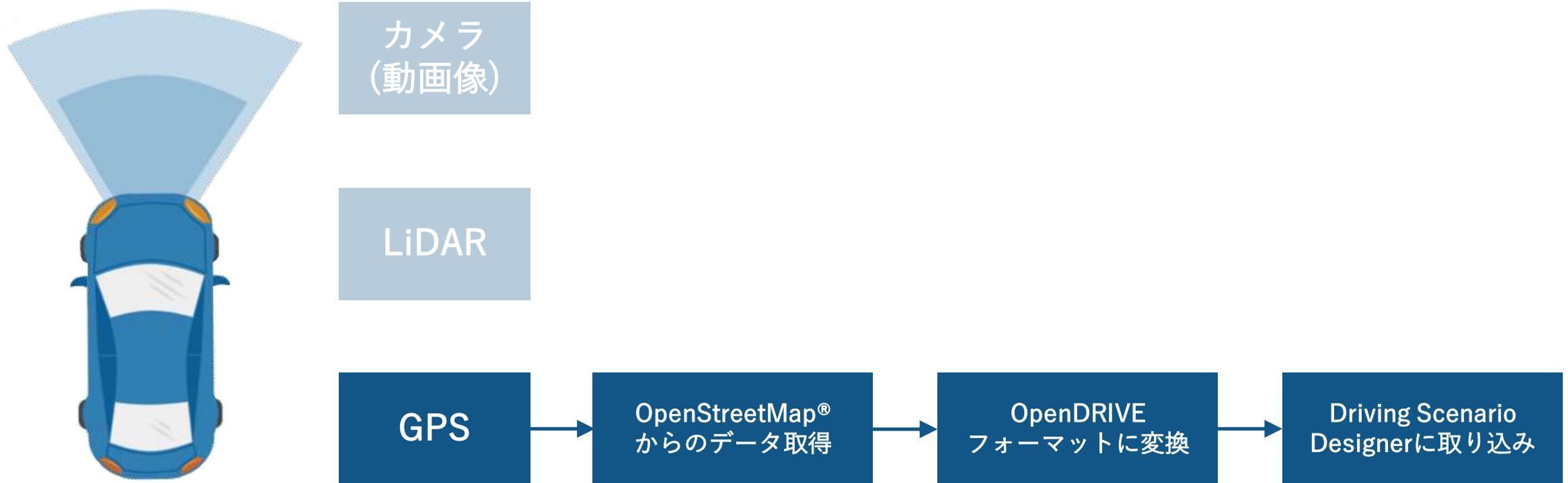
- 統計的なセンサー特性に基づいた簡易シミュレーション
- 実行時間は高速、マウス操作でシナリオ生成が可能

センサー特性や取り付け位置の検討、
センサーフュージョンや制御系アルゴリズムの検証など
幅広くお使い頂けます

Agenda

- 運転シナリオ生成&シミュレーション概要
- Road Network作成
 - 地図データへのアクセス
 - 道路情報のインポート
- 交通参加者の定義
 - センサーデータ(LiDAR)の取り扱い
 - 点群からの物体認識ワークフロー
- シナリオ自動生成とその活用
- まとめ

地図データへのアクセスと道路環境自動生成

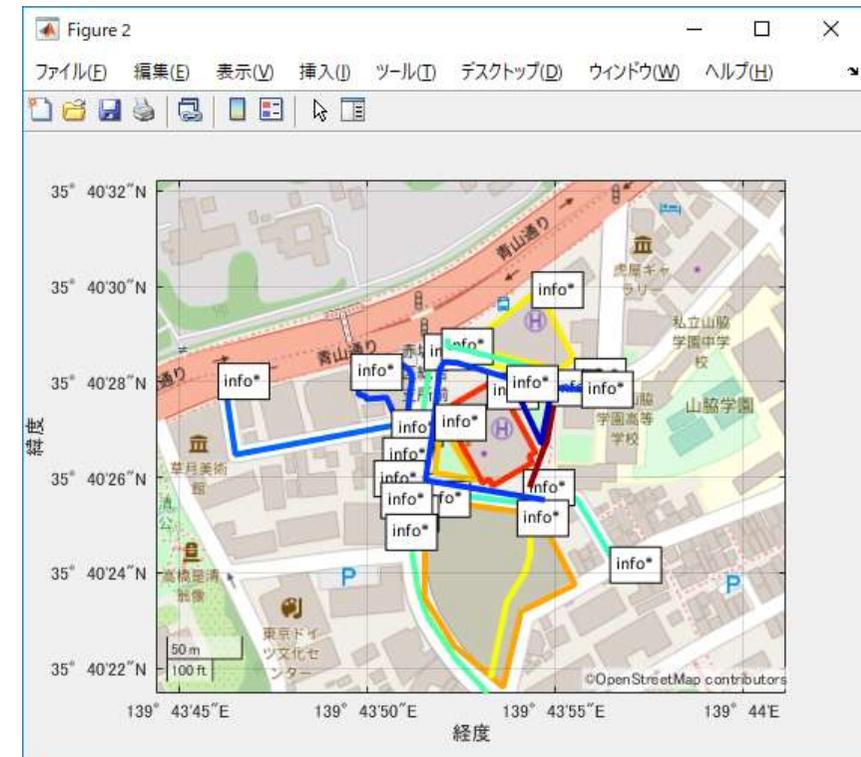




MATLABからの地図データへのアクセス

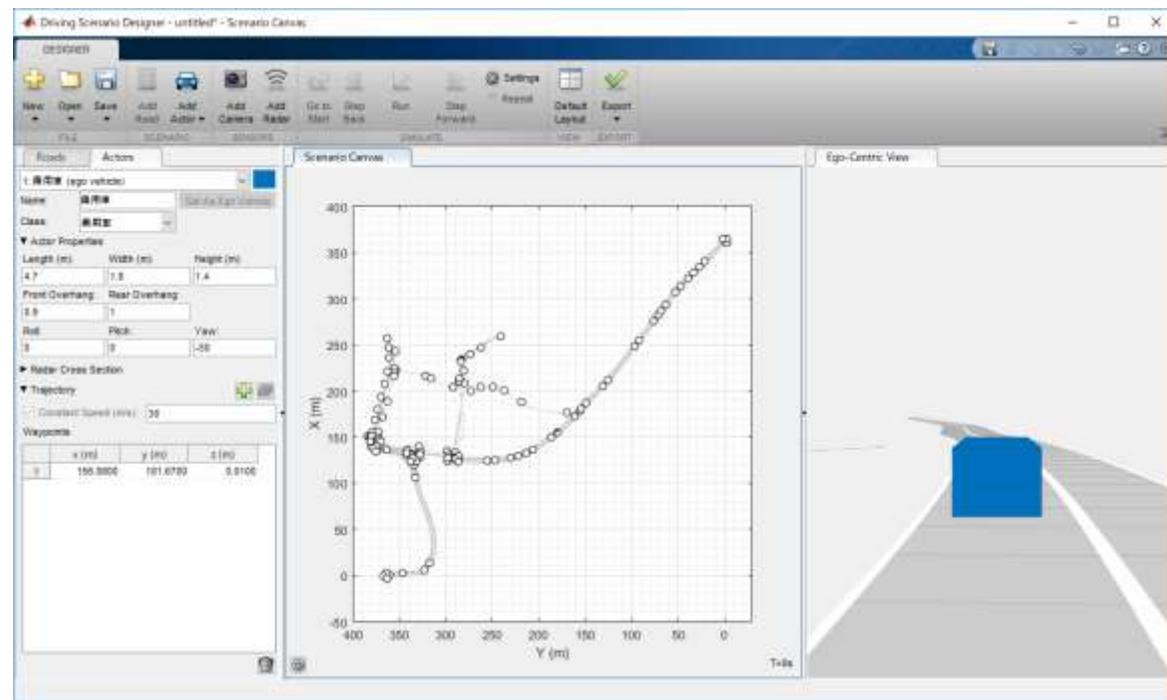
- webreadを利用した道路情報のダウンロード
 - OpenStreetMapの情報を部分的に選択して取得できる、Overpass APIの利用
 - API利用例
 - 緯度 : 35.674125
 - 経度 : 139.731326
 - Overpass API : <http://overpass-api.de/api/interpreter>

```
baseURL = 'http://overpass-api.de/api/interpreter';
quel = 'way(35.6736,139.7308,35.6746,139.7318);
out body;';
data = webread(baseURL, 'data', quel);
```



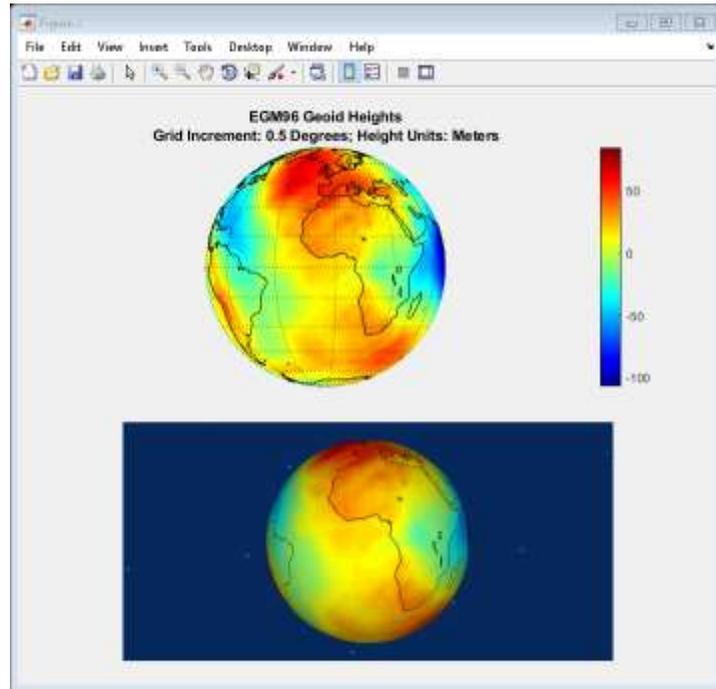
OpenStreetMapデータの取り込み – シナリオとしての活用

- Overpass API経由で取得した地図データ(.osm)は、openDRIVE形式に変換後Driving Scenarioに取り込むことが可能です
 - 手順
 - #1, .osm形式でRoad Networkを保存
 - #2, .osm形式のファイルをSUMO等のフォーマットを経由して.xodr形式に変換

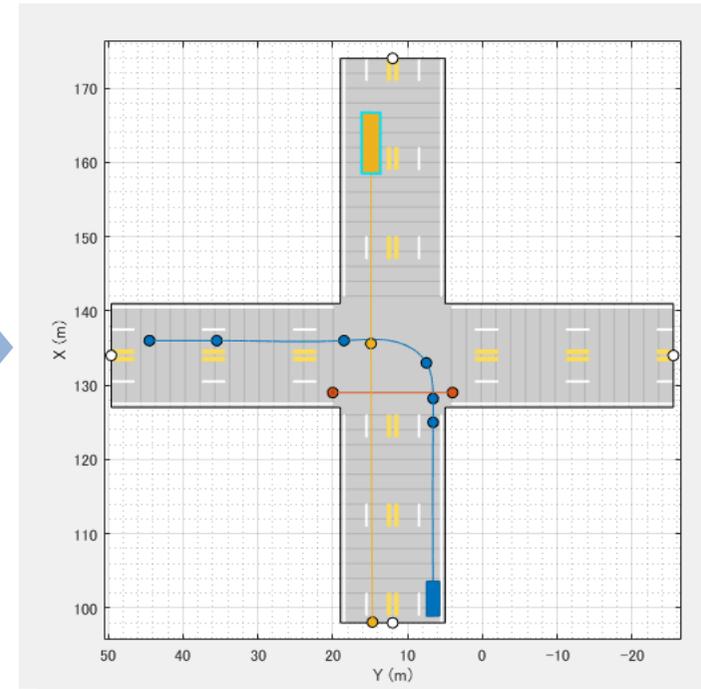


GPS基準座標(WGS64)系 > ENU座標への変換

Mapping Toolbox™



緯度、経度、高度(楕円体高)



x座標(m)、y座標(m)、z座標(m)

—	geodetic2ecef	geodetic2enu	geodetic2ned	geodetic2aer
ecef2geodetic	—	ecef2enu	ecef2ned	ecef2aer
enu2geodetic	enu2ecef	—	—	enu2aer
ned2geodetic	ned2ecef	—	—	ned2aer
aer2geodetic	are2ecef	aer2enu	aer2ned	—

高精細地図データ (HD Map)へのアクセス

HERE HD Live Map Reader

R2019a

>>hereHDLMReader

HD Localization Model

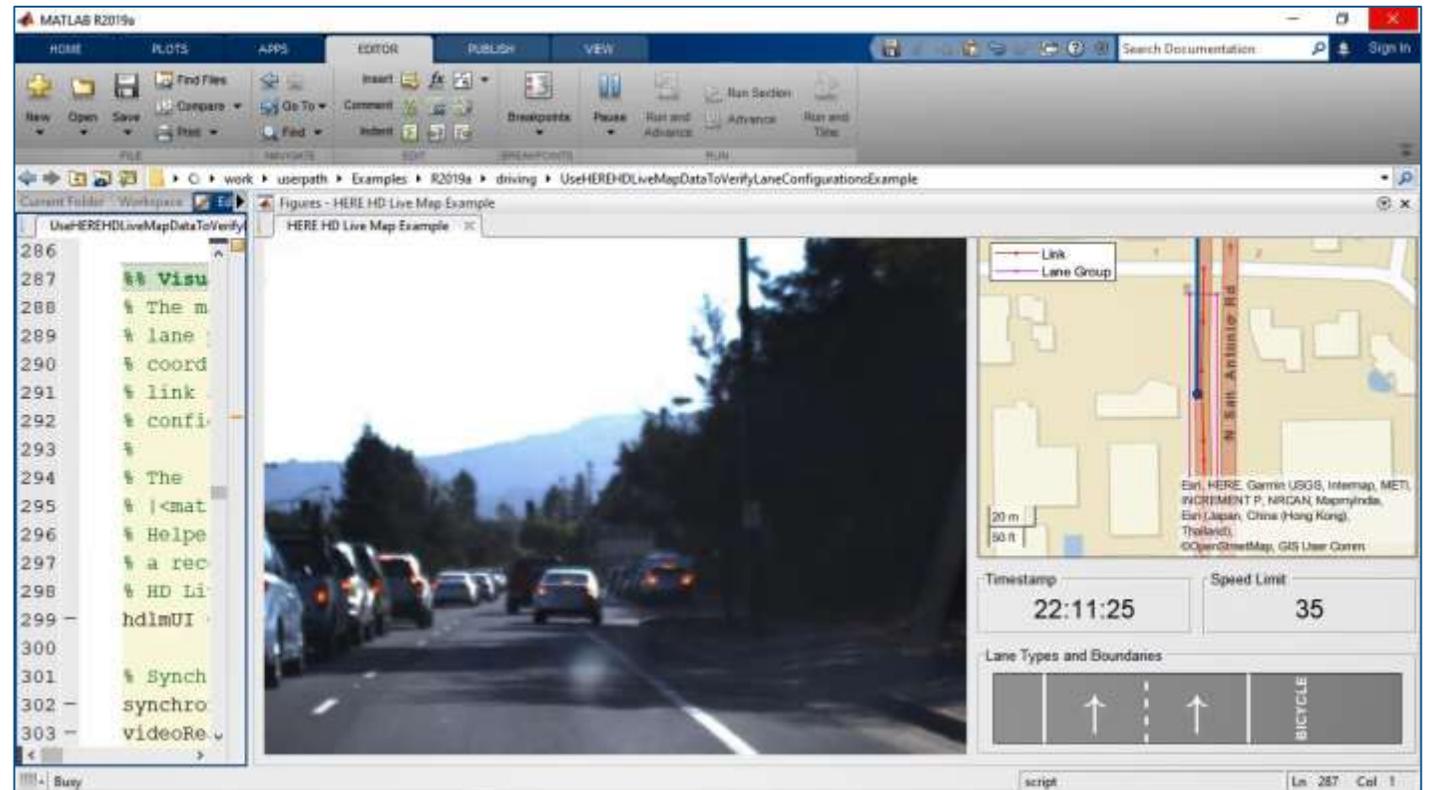
Sign Faces	Roadside Obstacles	Roadside Barriers
Pole-like Objects	...	

HD Lane Model

Lane Topology	Lane Geometry	Lane Attributes	Lane Road References
---------------	---------------	-----------------	----------------------

Road Model

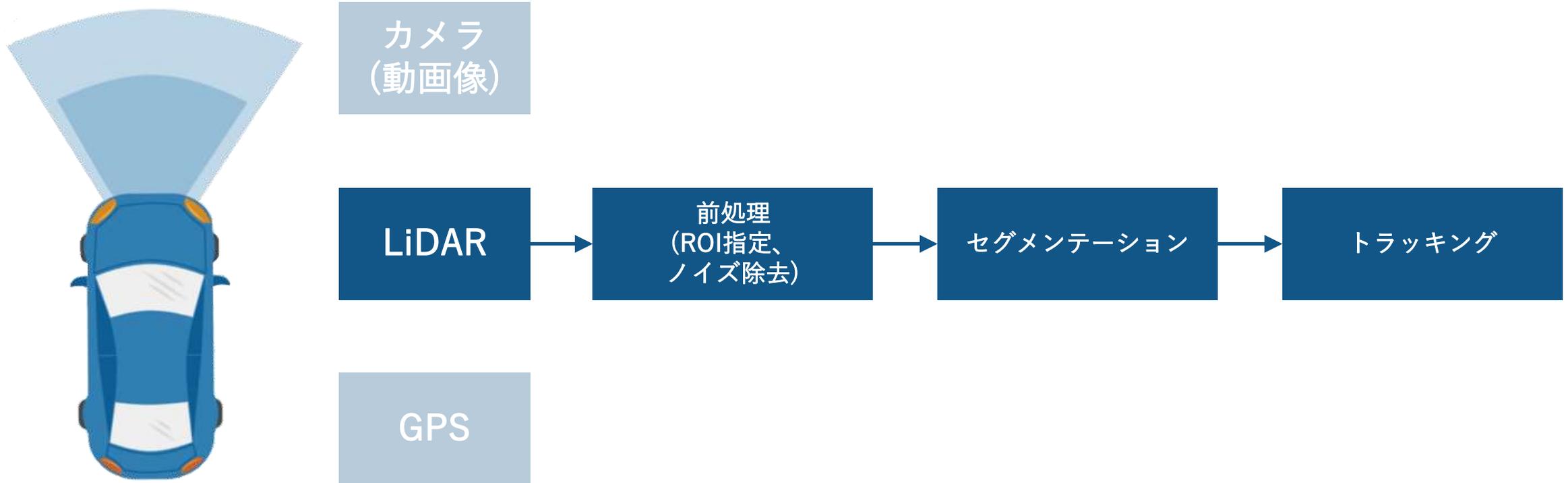
Topology Geometry	Routing Attributes	Speed Attributes
ADAS Attributes	External Ref Attributes	Routing Lane Attributes



Agenda

- 運転シナリオ生成&シミュレーション概要
- Road Network作成
 - 地図データへのアクセス
 - 道路情報のインポート
- 交通参加者の定義
 - センサーデータ(LiDAR)の取り扱い
 - 点群からの物体認識ワークフロー
- シナリオ自動生成とその活用
- まとめ

LiDARデータへのアクセスと物体認識



LiDARデータの取り扱い

Computer Vision Toolbox™

ファイル入出力

- pointCloud
- velodyneFileReader
- pcread
- pcwrite

前処理

- pcdenoise
- pcdownsampling
- pctransform



位置合わせ

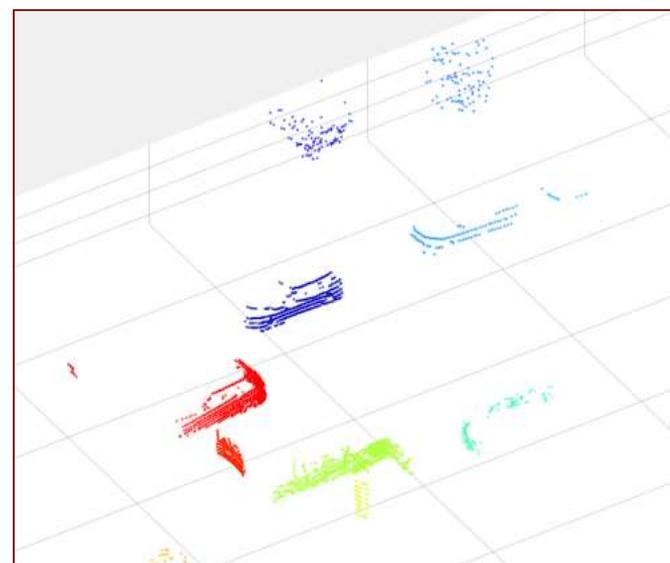
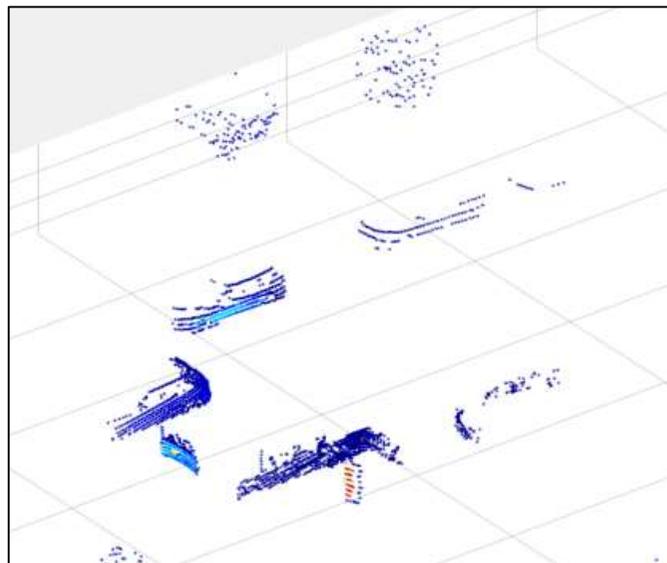
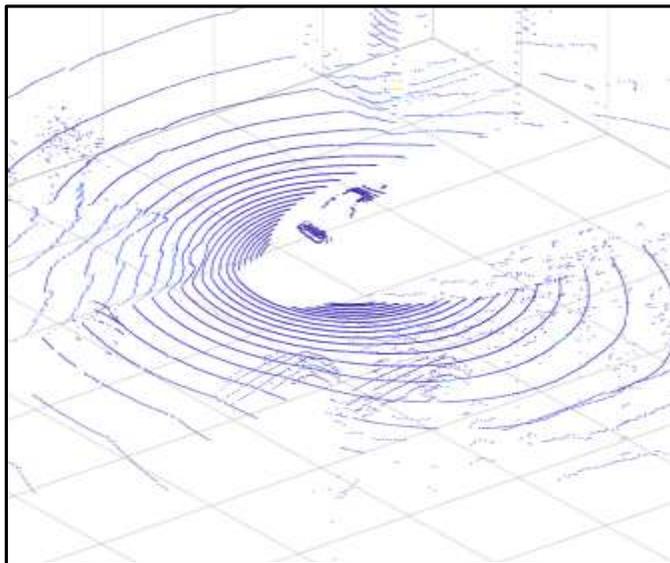
- pcregistericp
- pcregisterndt
- pcregistercpd

セグメンテーション

- pcsegdist
- segmentLidarData
- pcfitsplane
- segmentGroundFromLidarData

LiDARが初めてでも、使いやすい各種関数による可視化・解析・アルゴリズム開発

LiDAR データ前処理とセグメンテーション



データへのアクセス

```
>>velodyneFileReader
```

.pcapファイルへのアクセス

- VLP16,VLP32C,VLP32C
- HDL32C,HDL64E
- PuckLITE
- PuckHiRes

地表面の除去

```
>>pcfitplane
```

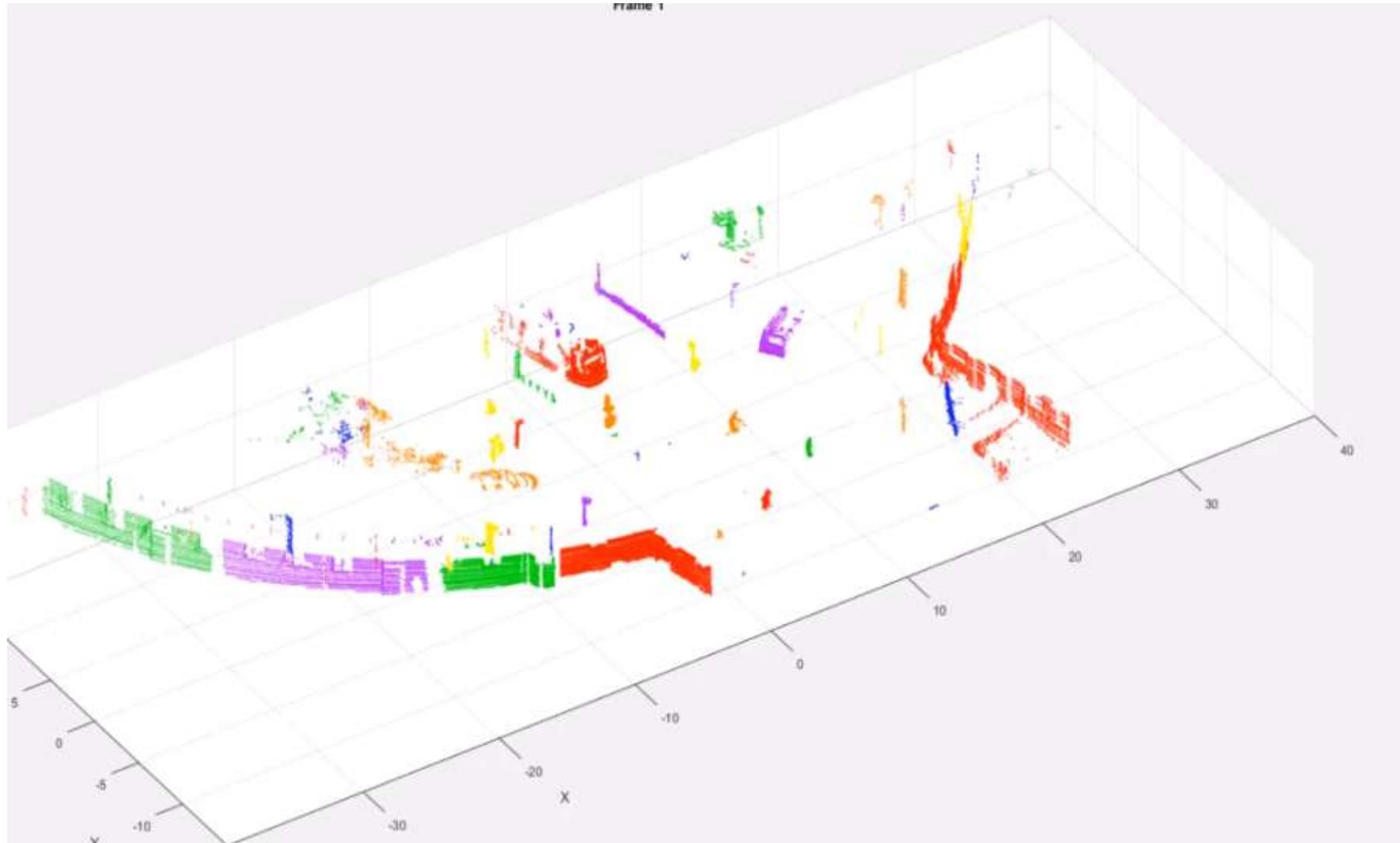
RANSACを利用した
モデルフィッティング

セグメンテーション

```
>>pcsegdist
```

ユークリッド距離を
利用したクラスタ作成

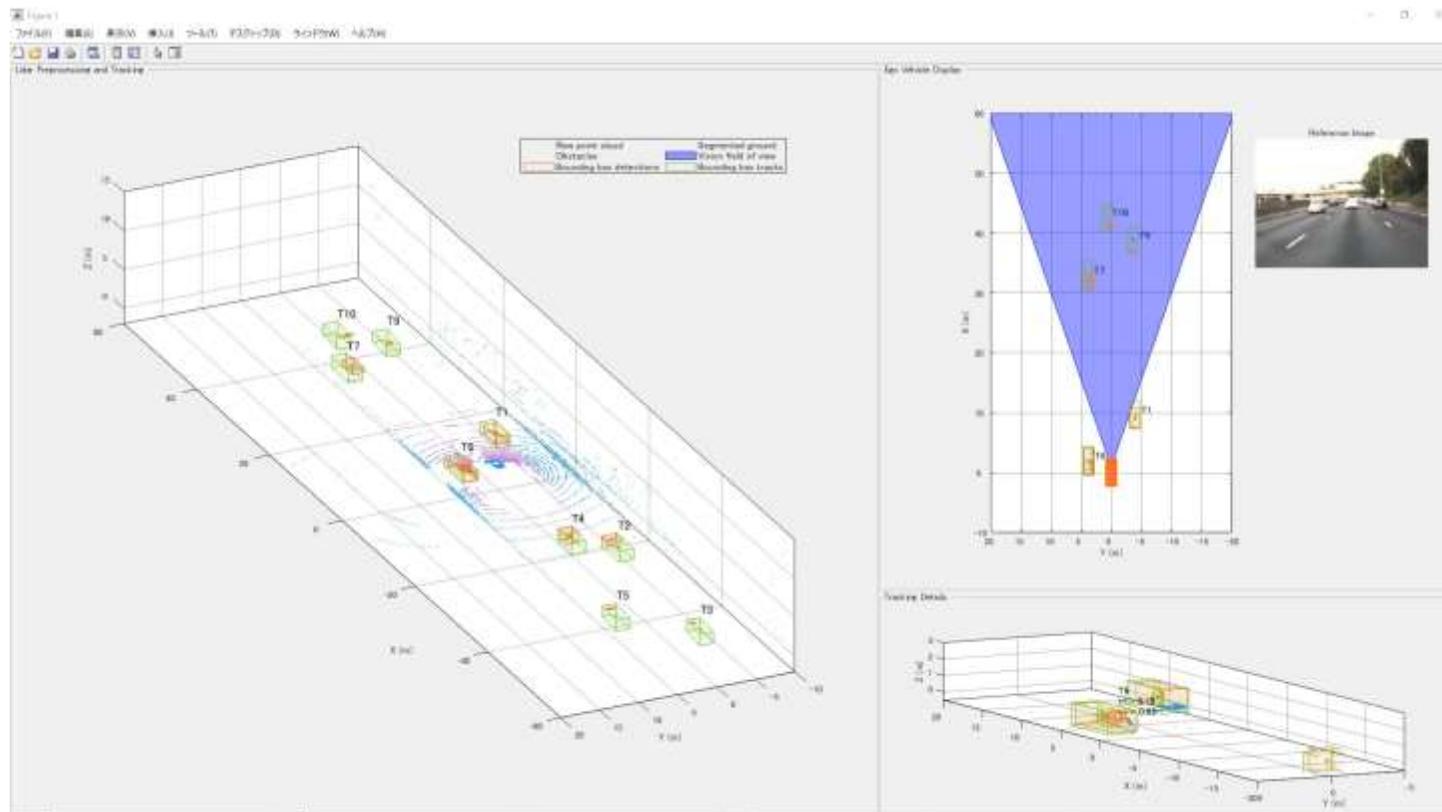
LiDAR データ 真値ラベリングと分類



LiDAR : JPDAによる相関決定とIMMフィルタによるトラッキング

Sensor Fusion and Tracking Toolbox™
Computer Vision Toolbox™

R2019a



3D Bounding Box検出モデル

クラスタのxyz各次元における最小及び最大座標点を求め、Boxをフィッティング

IMM(Interacting Multiple Model)

>>trackingIMM
複数の状態推定モデルの利用

JPDA(Joint Probablistic Data Association)

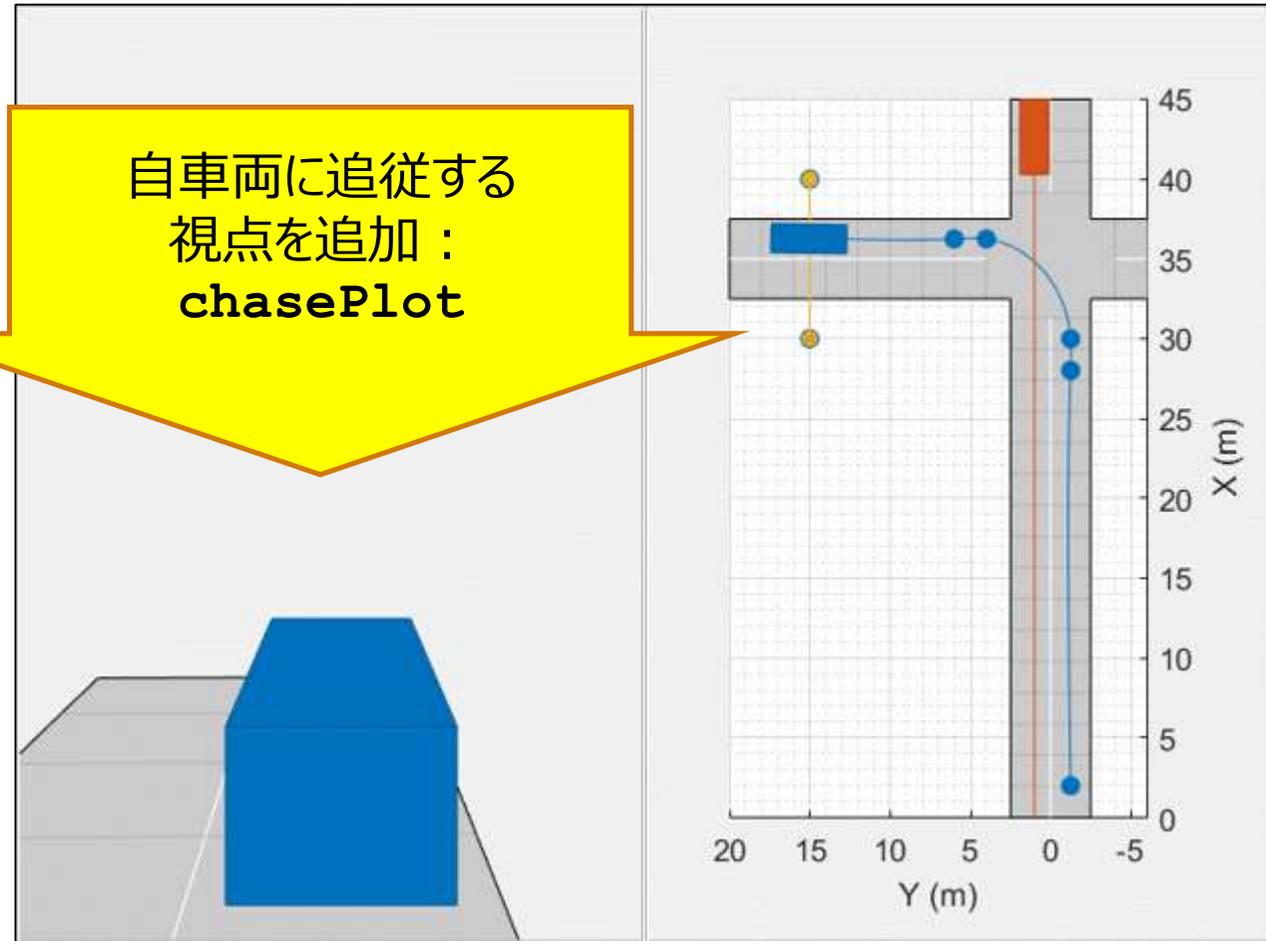
>>trackerJPDA
観測値の対応について、複数の仮説を設定

Agenda

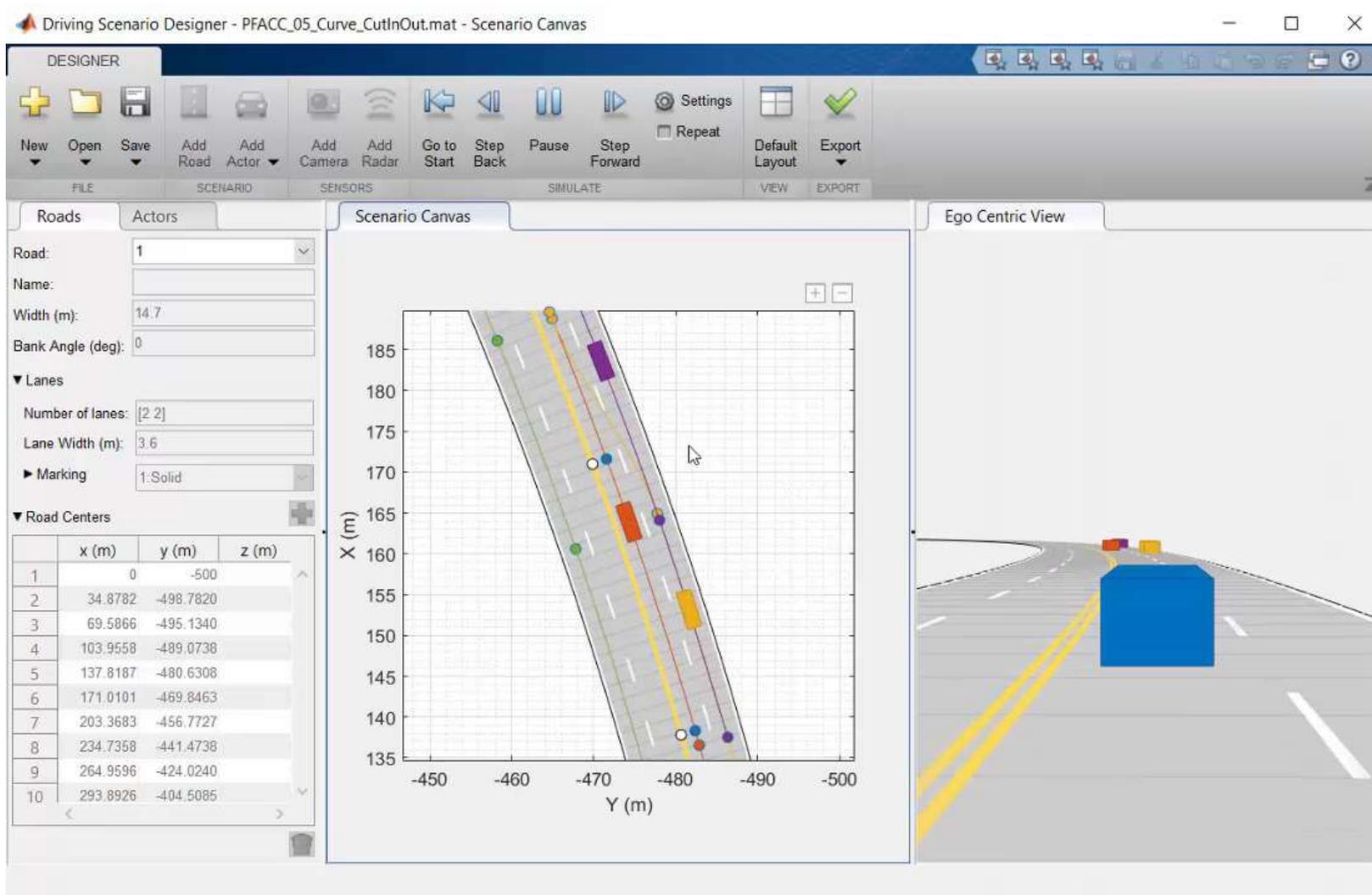
- 運転シナリオ生成&シミュレーション概要
- Road Network作成
 - 地図データへのアクセス
 - 道路情報のインポート
- 交通参加者の定義
 - センサーデータ(LiDAR)の取り扱い
 - 点群からの物体認識ワークフロー
- シナリオ自動生成とその活用
- まとめ

Driving Scenario – コマンドラインによる定義

```
%% Add chase view (left)
p2 = uipanel('Position',[0 0 0.5 1]);
a2 = axes('Parent',p2);
chasePlot(egoCar,...
    'Parent',a2,...
    'Centerline','on',...
    'ViewHeight',3.5,...      % (m)
    'ViewLocation',[-8 0]); % [x y] (m)
```



Driving Scenario Designer

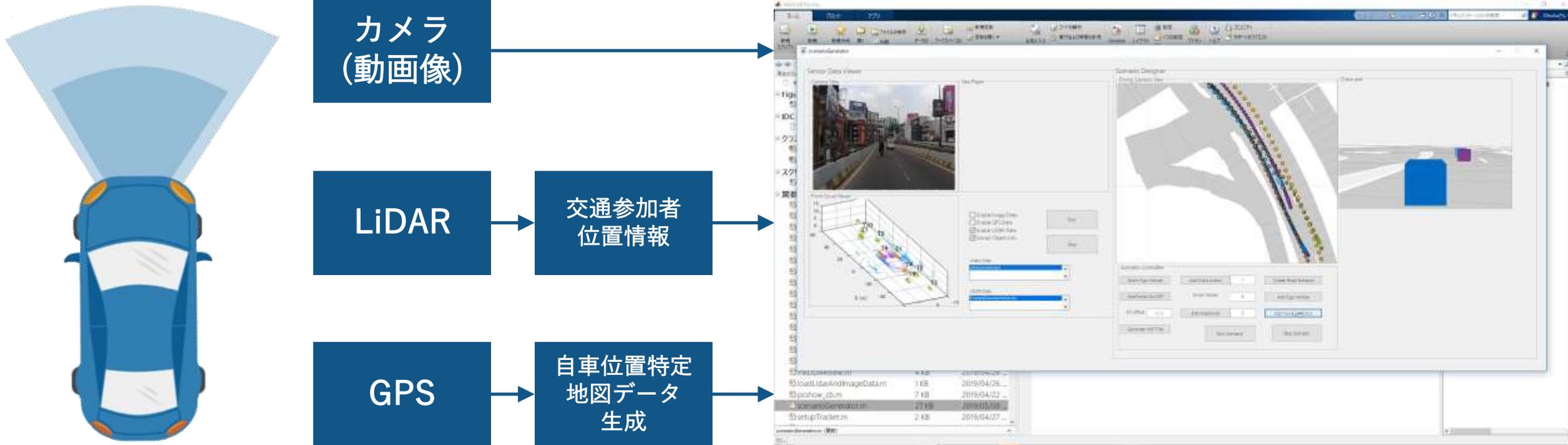


- シーン定義
 - 道路
 - レーンマーキング
 - アクターの軌跡
 - アクターサイズ
 - アクターのレーダー反射断面積 (RCS)
- MATLABコードへのエクスポート

R2018a

Automated Driving Toolbox™

テスト車両からのデータを利用したシナリオ自動生成



派生シナリオの自動生成

交通量の増減、軌跡の修正、簡易ドライバーモデルなど

Automated Driving Toolbox™
Image Processing Toolbox™

・ 交通量の増減



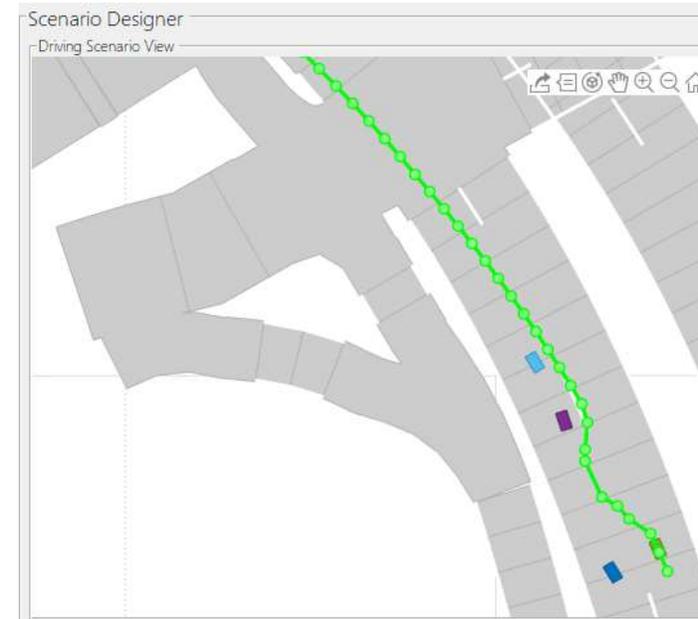
roadBoundaries

道路境界座標の取得

actorPoses

各アクターの位置、速度等の情報取得

・ 軌跡の修正



drawpolyline

対話型ROI設定(ポリライン)

drawline

対話型ROI設定(直線)

drawpoint

対話型ROI設定(点)

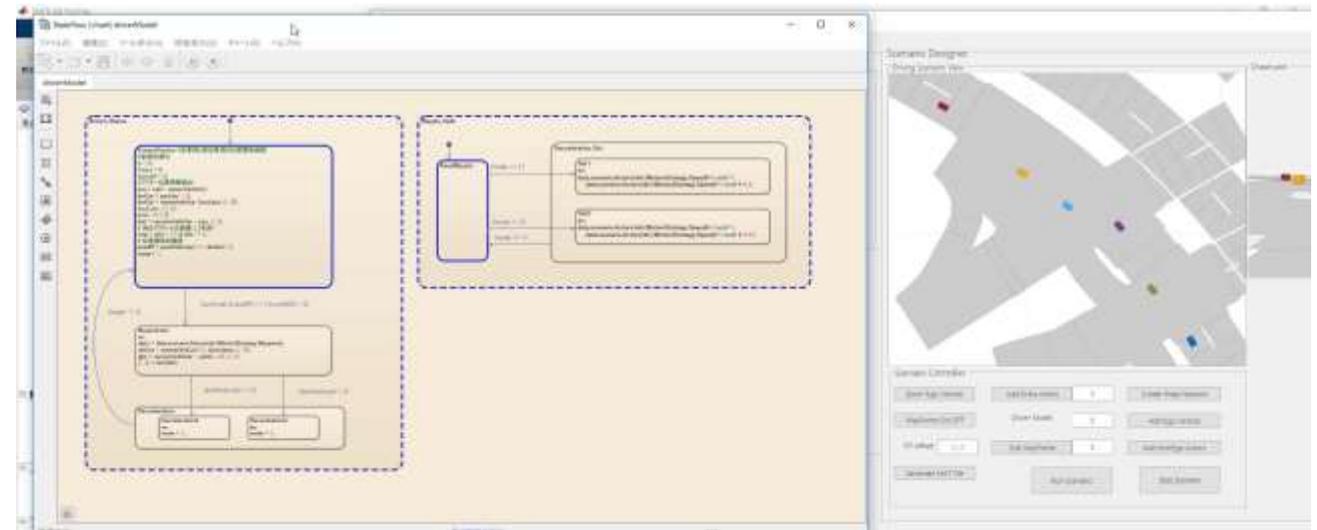
派生シナリオの自動生成

交通量の増減、軌跡の修正、簡易ドライバーモデルなど

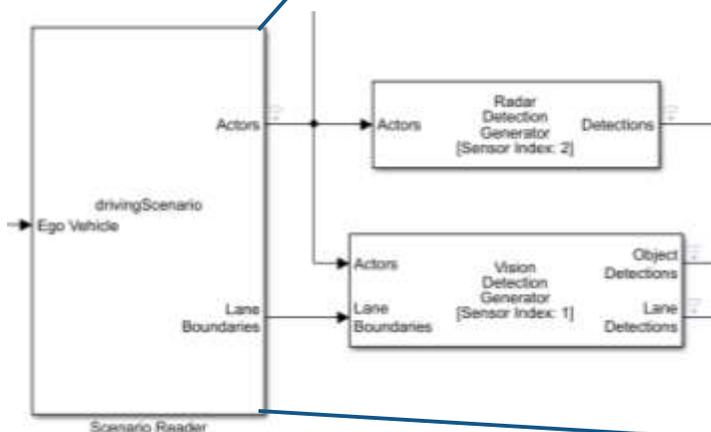
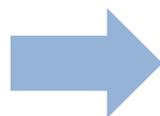
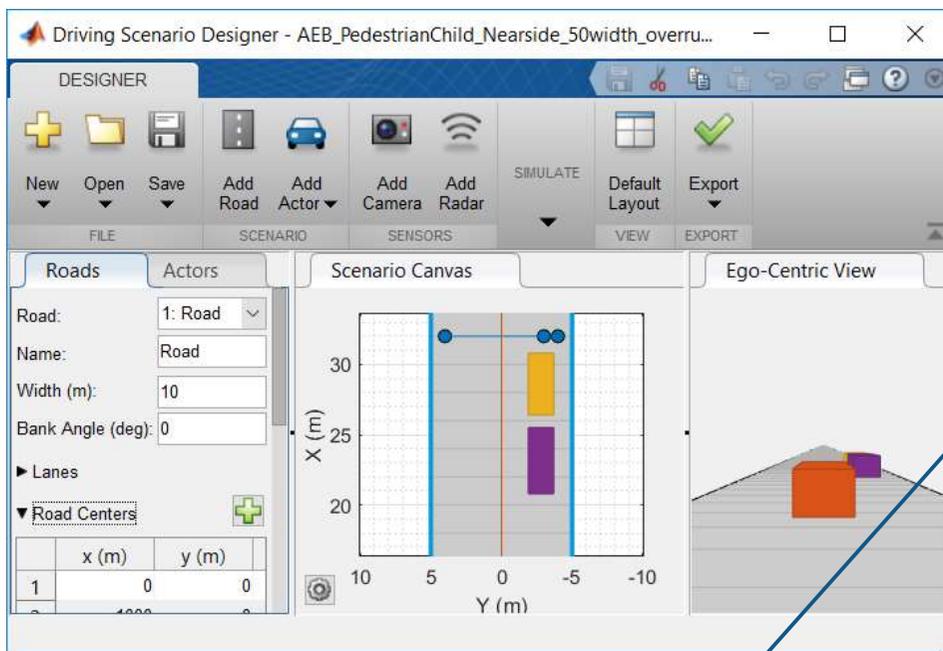
- ・ 簡易ドライバーモデル



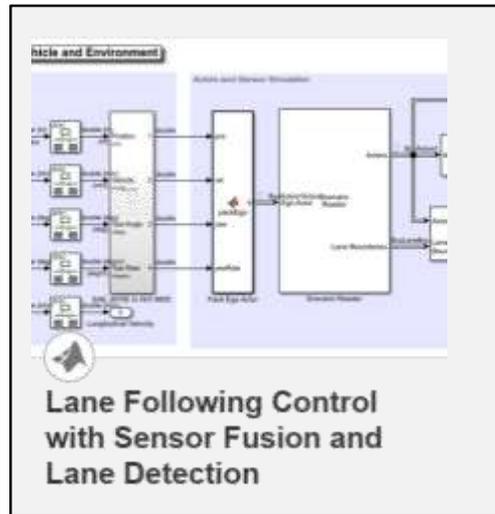
Stateflow Charts in MATLAB



作成したシナリオのSimulinkへの統合



制御・車両ダイナミクスモデルとの統合

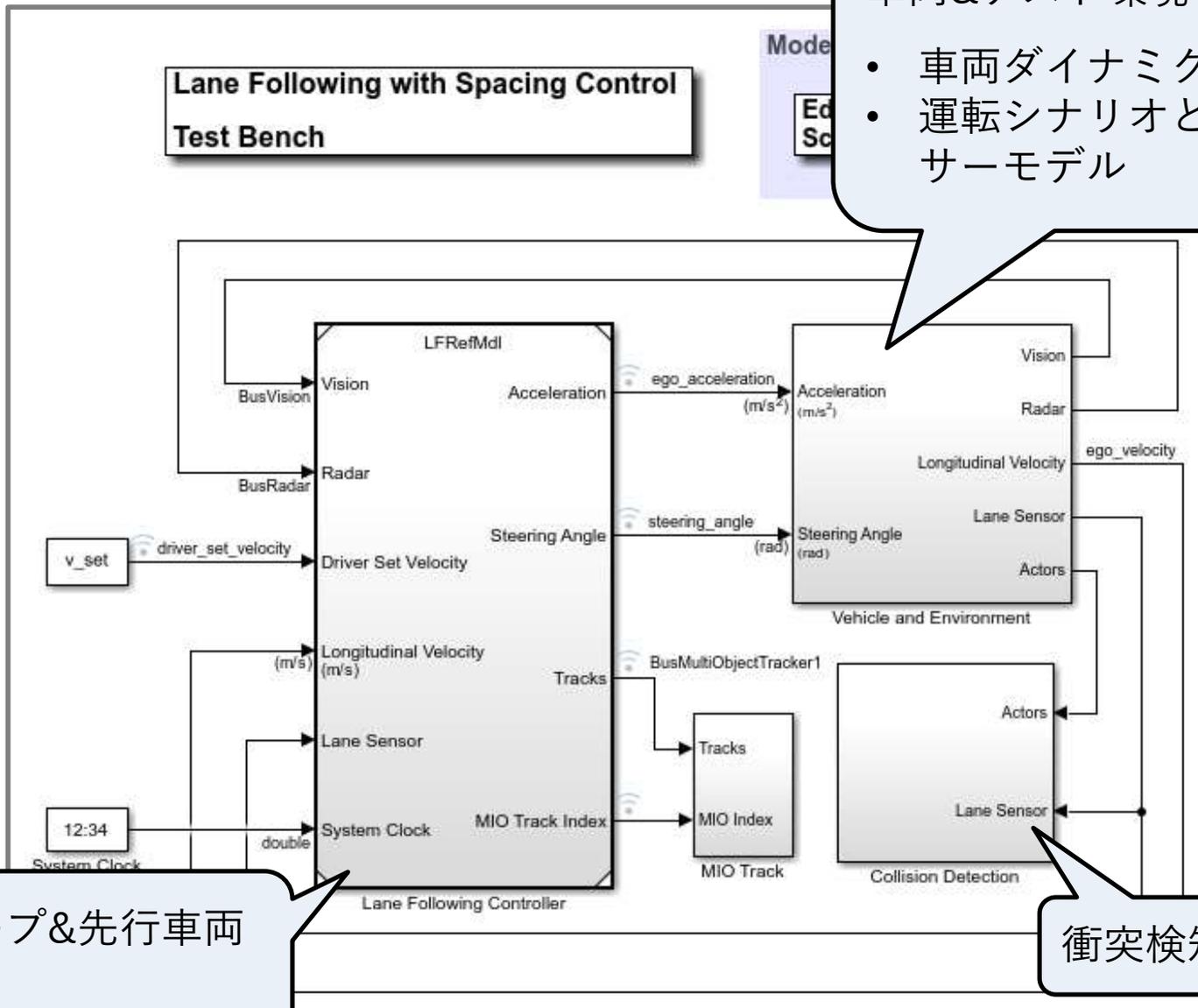


Lane following with spacing control

Automated Driving Toolbox™

Model Predictive Control Toolbox™

Embedded Coder®



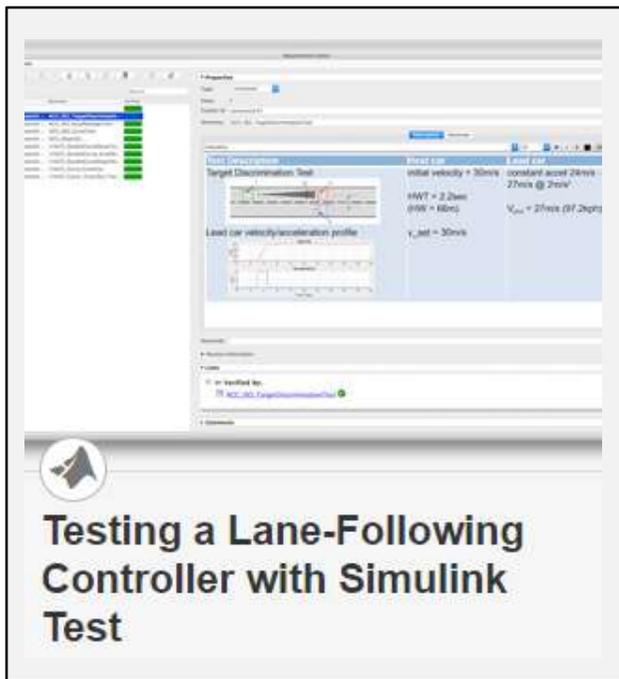
車両&テスト環境

- 車両ダイナミクス
- 運転シナリオとセンサーモデル

レーンキープ&先行車両追従制御

衝突検知

Simulink Testによるテストの管理・自動化



TESTS

Run

Filter tests by name or tags, e.g. tags: test

- Scenario
 - ACC_ISO_TargetDiscriminationTest
 - ACC_ISO_AutoRetargetTest
 - ACC_ISO_CurveTest
 - ACC_StopnGo
 - LFACC_DoubleCurve_DecelTarget
 - LFACC_DoubleCurve_AutoRetarget
 - LFACC_DoubleCurve_StopnGo
 - LFACC_Curve_CutInOut
 - LFACC_Curve_CutInOut_TooClose

事前定義済シナリオ

ACC_ISO_TargetDiscriminationT...

DESCRIPTION*

REQUIREMENTS*

scenarioId #1: ACC_ISO_TargetDiscriminationTest (LaneFollowingTestRequirements#1)

SYSTEM UNDER TEST*

Model: LaneFollowingTestBenchExample

TEST HARNESS

SIMULATION SETTINGS OVERRIDES*

PARAMETER OVERRIDES

CALLBACKS*

PRE-LOAD

POST-LOAD*

Runs after the model loads and the model PostLoadFcn callback

```

1 scenarioId = 1;
2 helperLFSetup;
    
```

シナリオIDの定義、データ初期化

CLEANUP*

Runs after simulations and all model callback

```

1 plotLFResults(slttest_simout.logout);
    
```

要求仕様へのリンク

Simulinkモデル

結果のプロット、合否判定

PROPERTY	VALUE
Name	ACC_ISO_TargetDiscri...
Type	Simulation Test
Model	LaneFollowingTestBenchEx...
Simulation Mode	Normal
Location	C:\02_ADST\2018b\Demo...
Enabled	<input checked="" type="checkbox"/>
Hierarchy	LaneFollowingTestScenario...
Tags	Type comma or space separat

Testing a Lane Following Controller with Simulink Test

Simulink Test™

Automated Driving Toolbox™

Model Predictive Control Toolbox™

MATLAB EXPO 2019

目的に応じて選択できる2つの環境

ゲーミングエンジン
連携

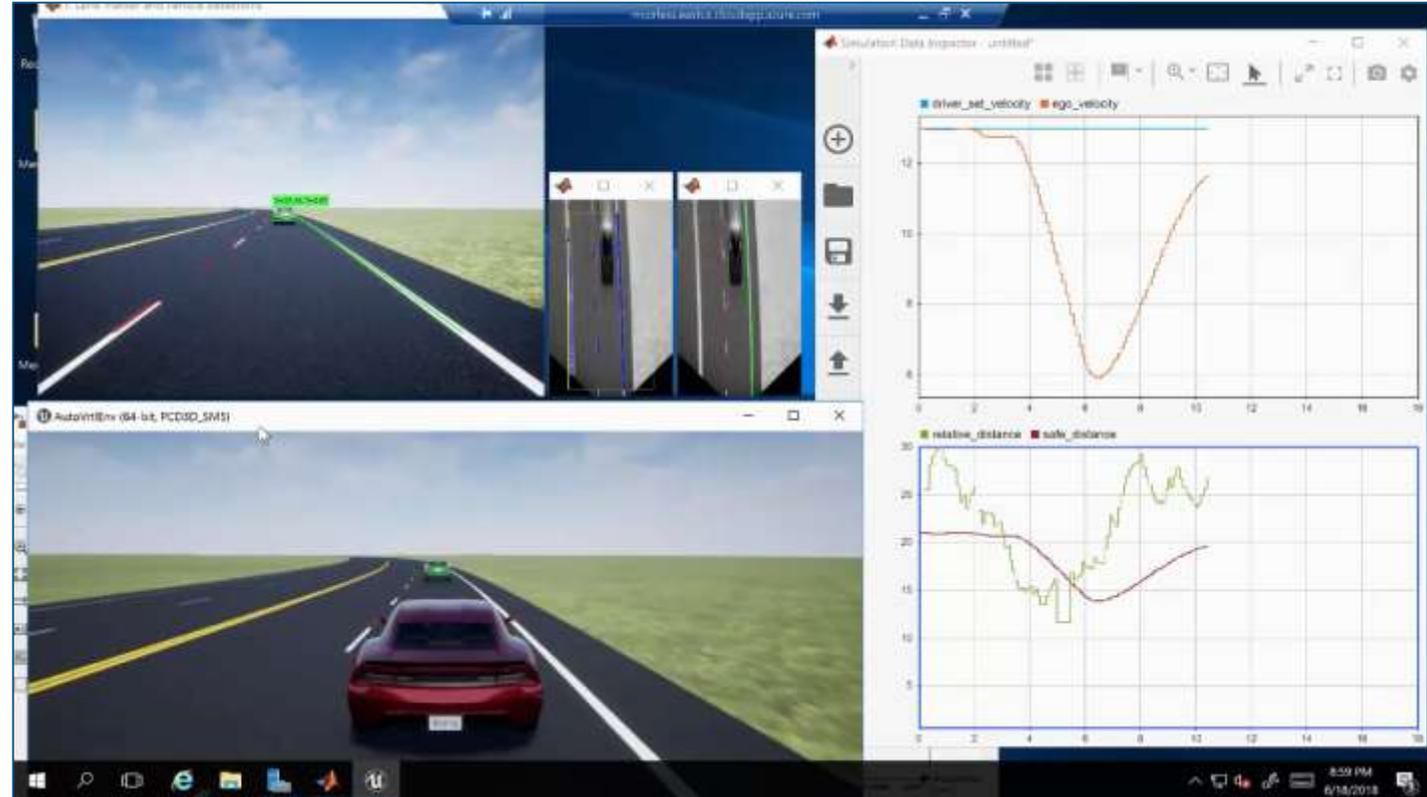


認識系アルゴリズム必要



より“リアル”な環境、
物理センサーモデル

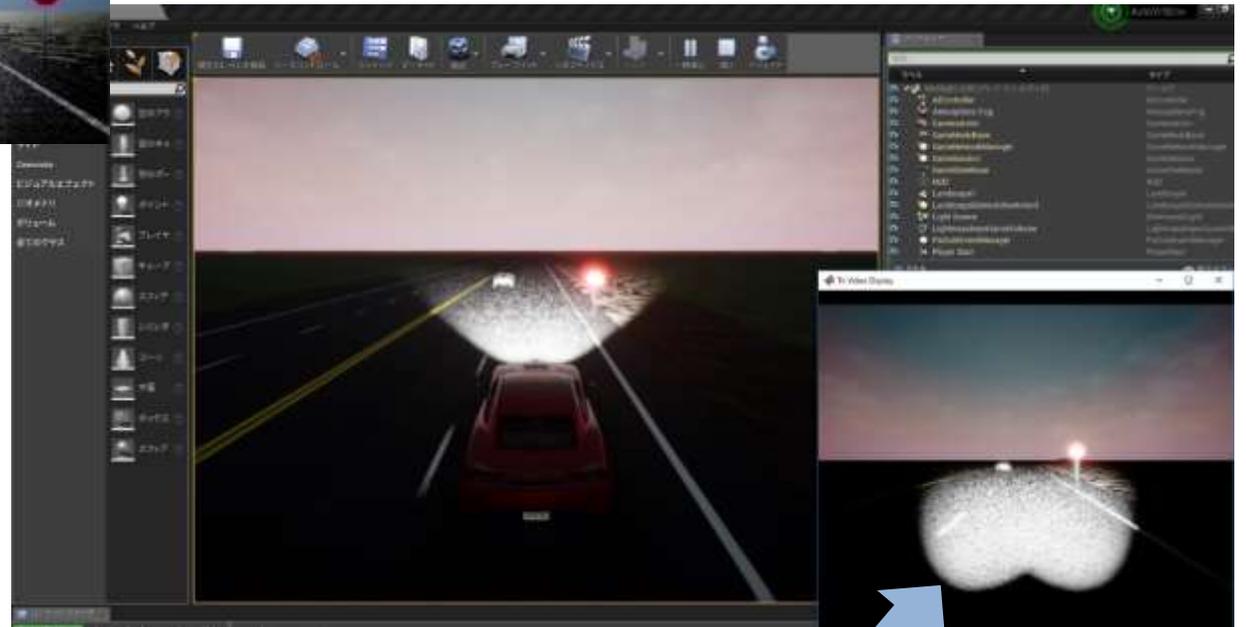
- 認識系アルゴリズムを含む、より現実に近いシミュレーション
- 実行時間は低速、環境構築のため専門知識が必要



天候・照明の有無による認識率の変化を再現



画像処理(Deep Learning)により
標識を検出



反射の影響で物体をロスト

Agenda

- 運転シナリオ生成&シミュレーション概要
- Road Network作成
 - 地図データへのアクセス
 - 道路情報のインポート
- 交通参加者の定義
 - センサーデータ(LiDAR)の取り扱い
 - 点群からの物体認識ワークフロー
- シナリオ自動生成とその活用
- まとめ

Key Takeaways : ADAS・自動運転アルゴリズム検証のためのシナリオ生成とシミュレーション

- ✓ 地図データへのアクセス、OpenDRIVE形式でのロードネットワークの取り込み
- ✓ LiDAR等センサーデータの取り扱いやセンサーフュージョンアルゴリズム開発用の高度なライブラリ群
- ✓ 交通量や車両軌跡の修正、簡易ドライバーモデルの追加等カスタマイズ
- ✓ Simulink Testによるテスト項目の一括評価

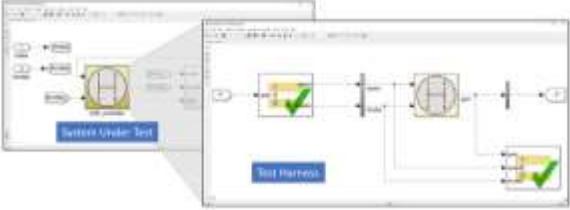
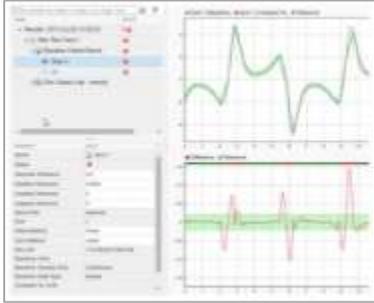
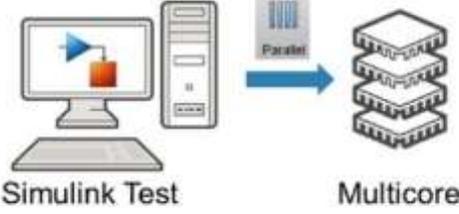
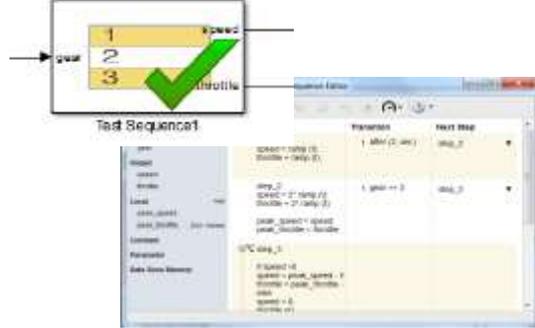
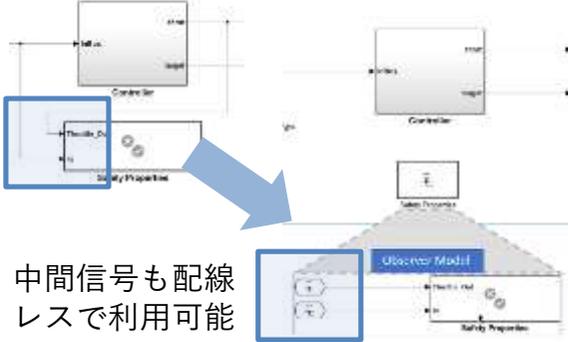


既存データからのシナリオ生成・活用により
より効率的なADAS/ADアルゴリズムの開発・検証のフローを実現



© 2019 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Appendix : Simulink Test でテスト作業やテスト管理の改善・省力化

テストハーネス	テストマネージャ	並列・分散テスト	テストシーケンス
<p>モデルやサブシステムにテスト専用モデルを作成・関連付けて管理できます</p>	<p>MIL/SIL/PILでテストを自動実行、可否レポートを作成できます</p>	<p>複数のテストを並列・分散処理で高速化できます</p>	<p>入力パターンや評価式を状態遷移表で簡潔に表現できます</p>
<p>親モデルとは独立してテストハーネスに入出力やプラントを追加できます</p> 	<p>信号値・時間ずれの評価が可能です</p> 	 <p>Parallel Computing Toolbox / MATLAB Parallel Serverが必要です</p>	
Excelテストデータ読み込み	マルチリリーステスト	オブザーバ	要求に基づくテスト評価
<p>Excelテンプレートに沿って入出力データを記述、簡単に取り込めます</p>	<p>同一モデルを複数バージョンで比較・評価できます</p>	<p>面倒な配線無しで信号データを表示・評価できます</p>	<p>要求を自然言語風テンプレートに沿って記述、その成否を評価します</p>
		 <p>中間信号も配線レスで利用可能</p>	<ul style="list-style-type: none"> 信号値の上下限・範囲内チェック ～秒経過したら～する 特定条件が～回発生後～になる、等 