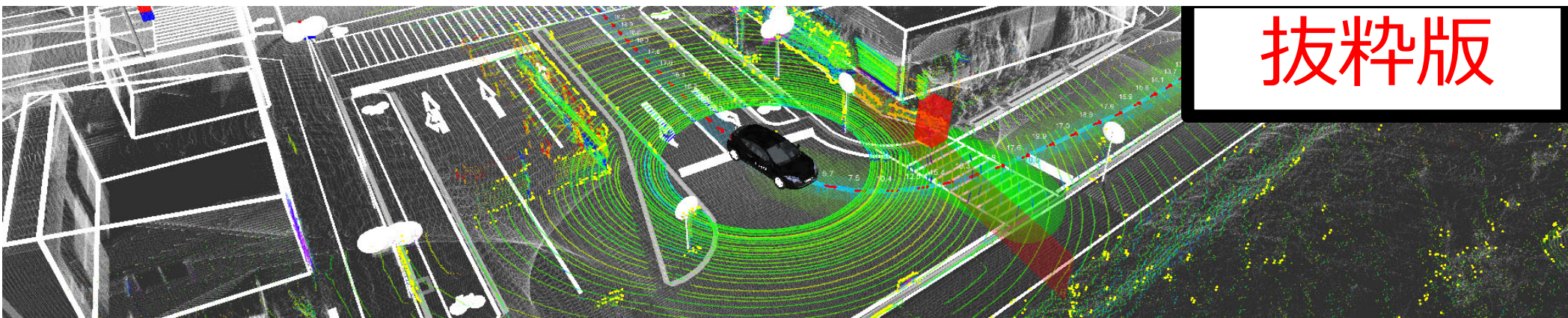


抜粋版



# 自動運転向けソフトウェアAutowareと MATLAB<sup>®</sup>/Simulink<sup>®</sup>の連携

安積卓也

埼玉大学大学院理工学研究科



# 自己紹介

名前：安積 卓也（あづみ たくや）

- 出身研究室：名古屋大学大学院 情報科学研究科
  - 2006-2009：博士後期課程 高田研：組込みリアルタイム  
2006-2007：未踏ソフトウェア（代表）
  - 2008-2010：学振 特別研究員 DC→PD
- 立命館大学情報理工学部・情報システム学科・助教
  - 2010年4月～2014年2月：西尾研究室：ユビキタス
  - 2011年9月～2012年3月：  
カリフォルニア大学アーバイン校：客員研究員

帰国後：自動運転向けシステムソフトウェアの研究開発開始

- 大阪大学 大学院基礎工学研究科 助教
  - 2014年3月～：潮研究室：制御・人工知能
  - 2016年9月～：株式会社エンブフォー：最高技術責任者
  - 2017年10月～：JSTさきがけ研究員（兼任）
- 埼玉大学大学院理工学研究科
  - 2018年4月～：准教授



# 自動運転のレベル

※1 官民ITS構想RM用語対応表  
 操舵：ハンドル（ステアリング）  
 加速：アクセル  
 制動：ブレーキ

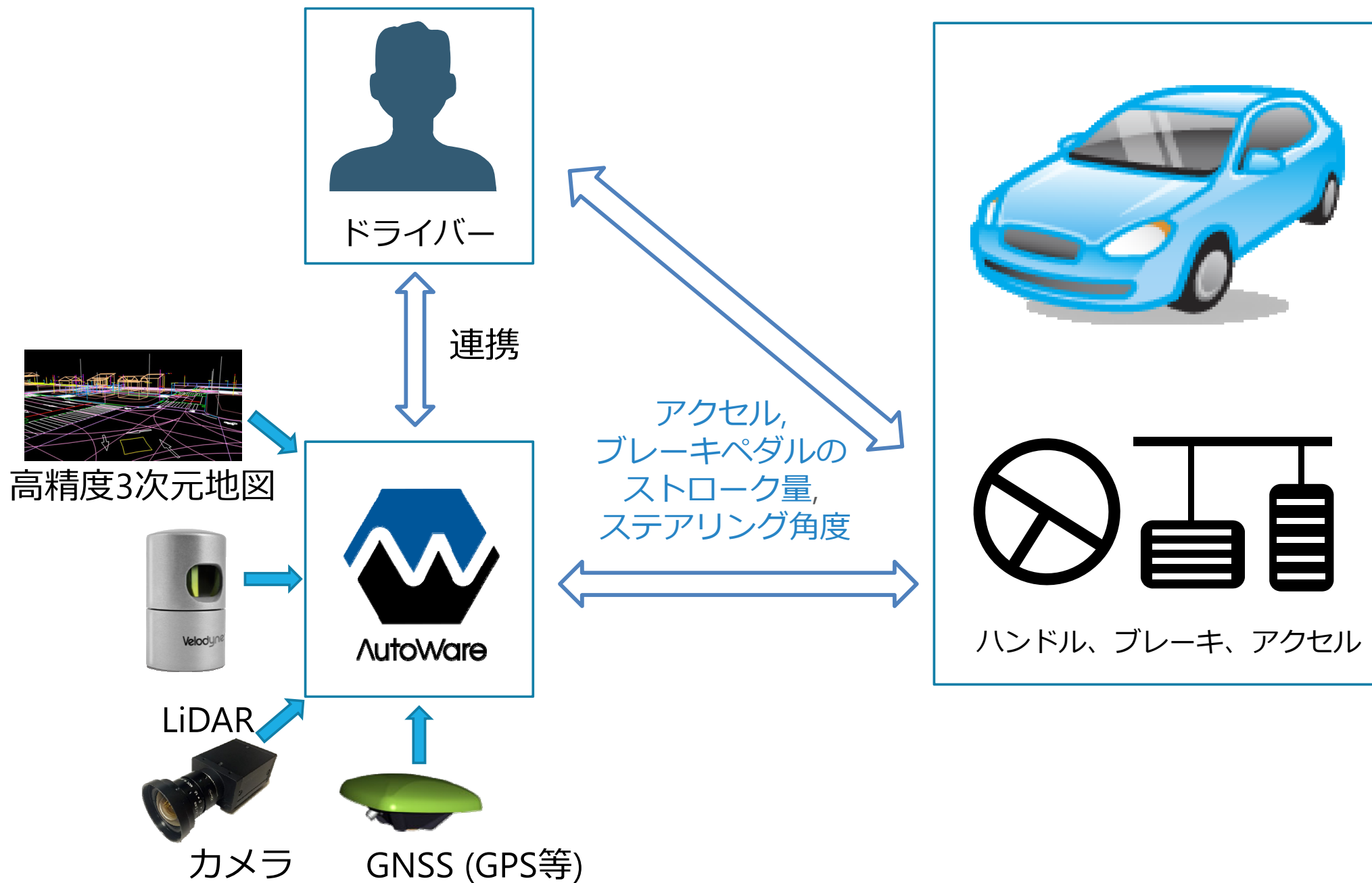
レベル	システム：※1 ハンドル・アクセル・ブレーキ	ドライバー	場所	
レベル1	いずれか一つ	主体		自動ブレーキ
レベル2	複数	主体		クルーズ コントロール
レベル3	すべて (危険回避はドライバー)	あり		公道実験
レベル4	すべて	なし	限定	自動運転バス
レベル5	すべて	なし	全て	自動運転 タクシー

完全自動運転

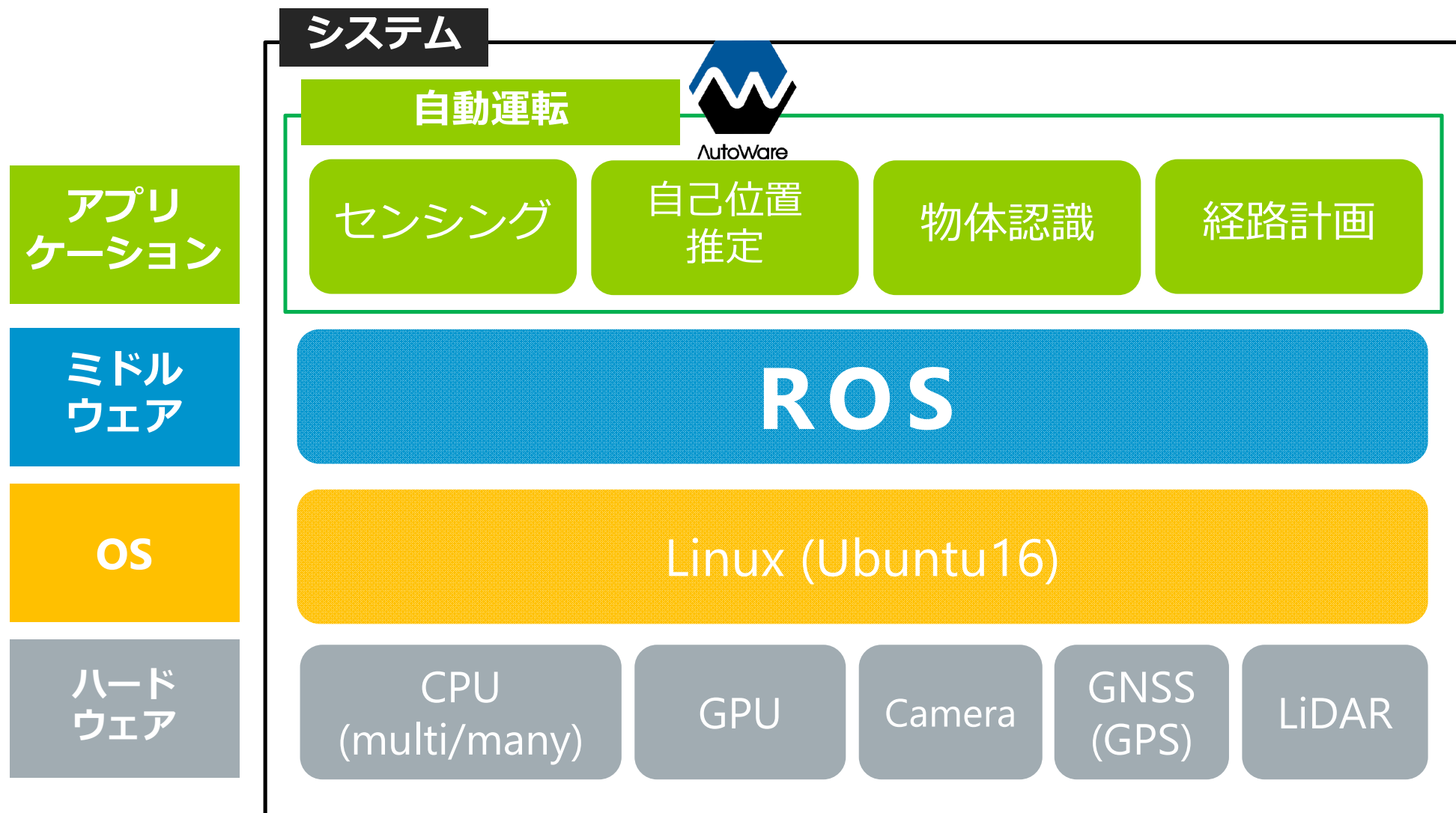
※SAE J3016 (2016)



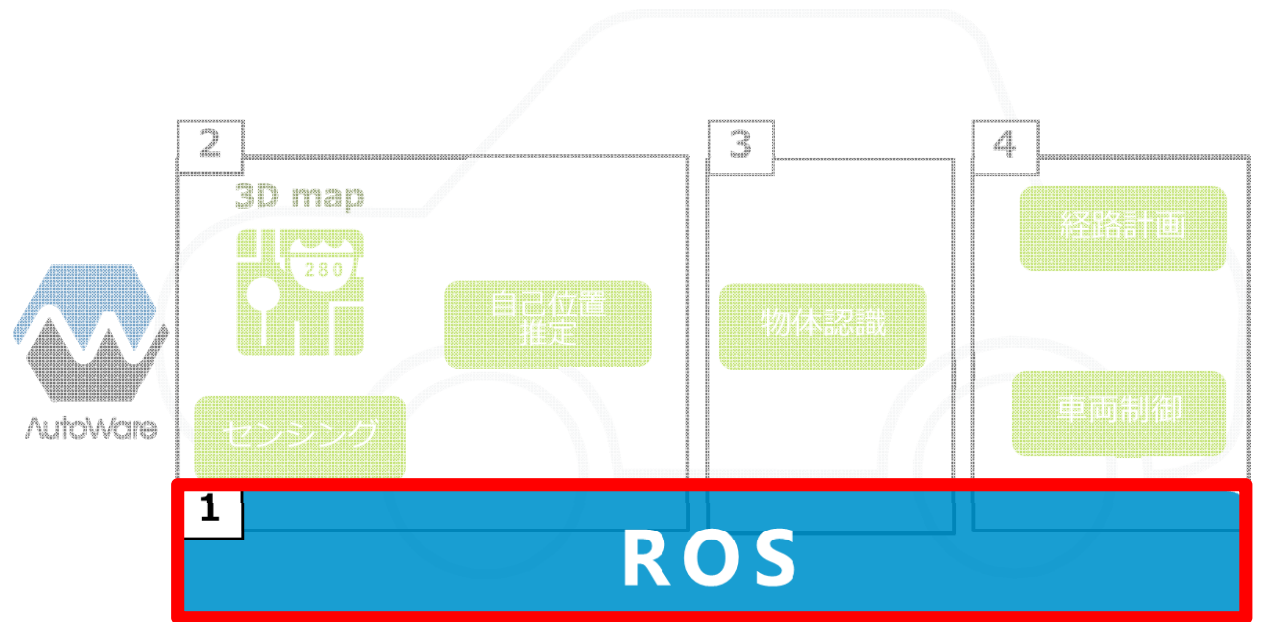
# 自動運転ソフトウェアの役目



# Autowareの構成



<https://github.com/autowarefoundation/autoware>



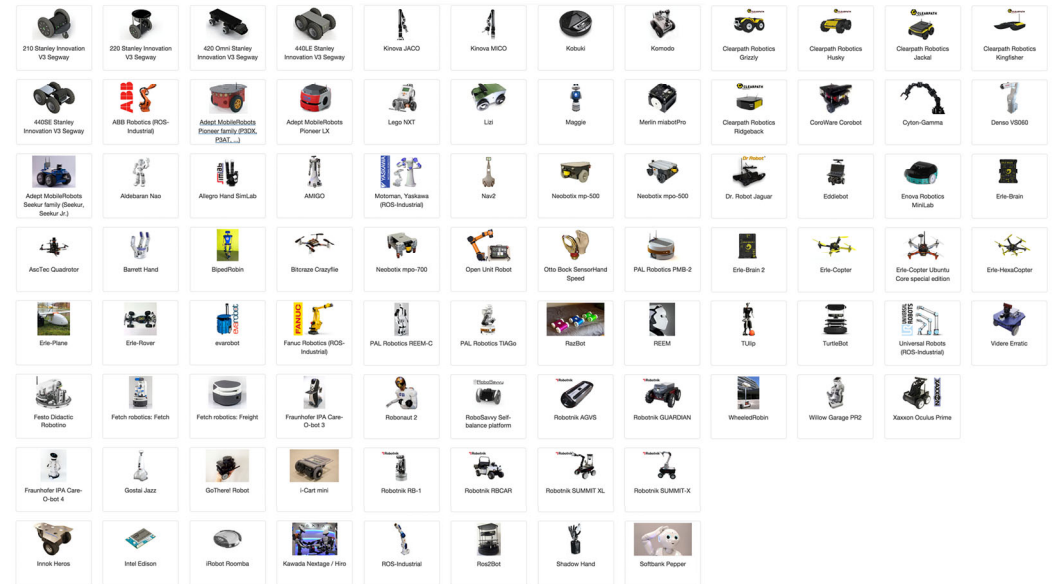
ROS (Robot Operating System)

## ROS (Robot Operating System)

: ロボット開発におけるライブラリやツールを提供  
ハードウェアの抽象化、デバイスドライバ、ライブラリ、視覚化ツール、  
データ通信、パッケージ管理 ...etc

## 特長

- 世界で最も利用されているロボットミドルウェア
- 豊富な対応ロボット・センサ
- オープンソース
- サポート言語 : C++, Python
- 管理団体 : OSRF
- 対応OS : Linux



Open Source Robotics Foundation

# ROS の 特長

## ROS (Robot Operating System)

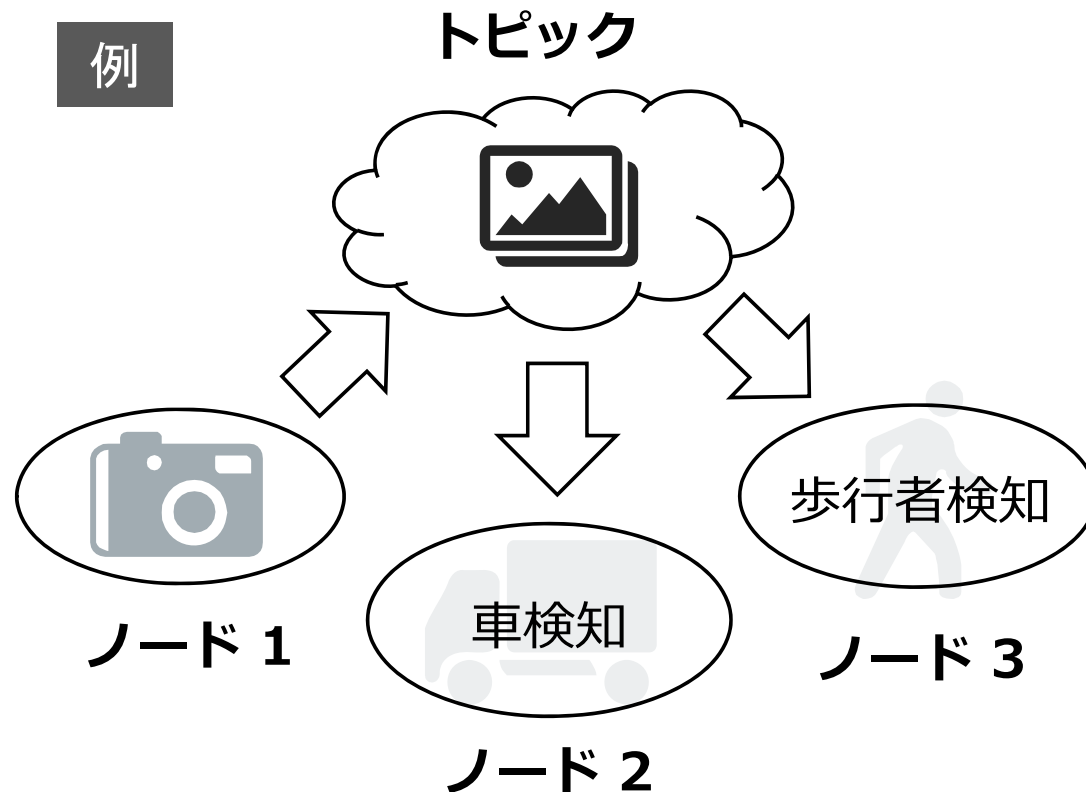
: ロボット開発におけるライブラリやツールを提供



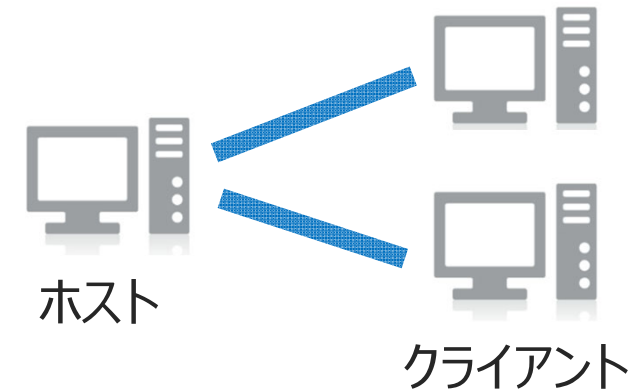
### Publish / Subscribe モデル

- ノードの集合としてシステムを構築
- トピックを介してデータをやり取り

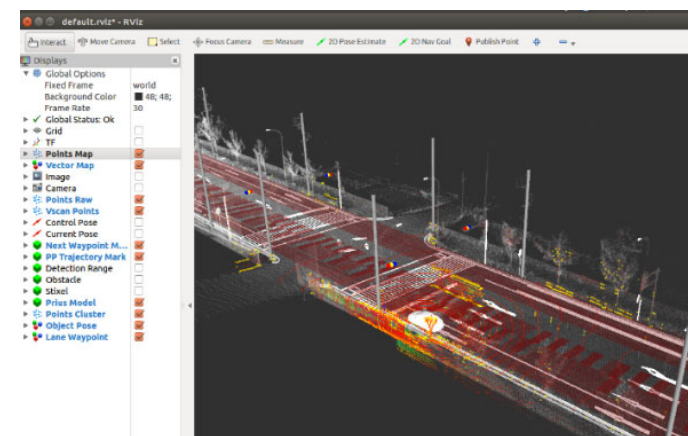
例



### 分散システム



### 視覚化・シミュレーション

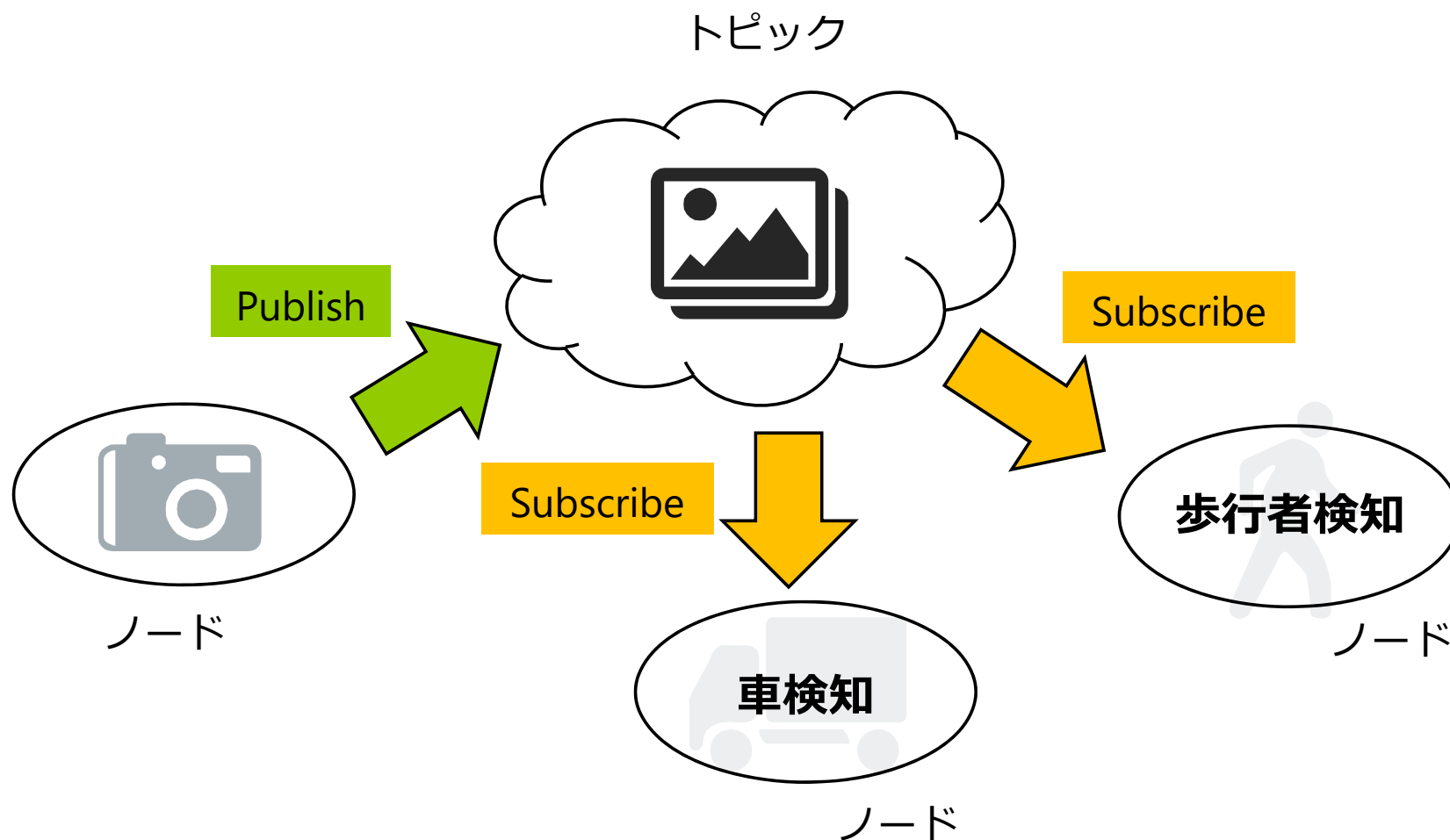




# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

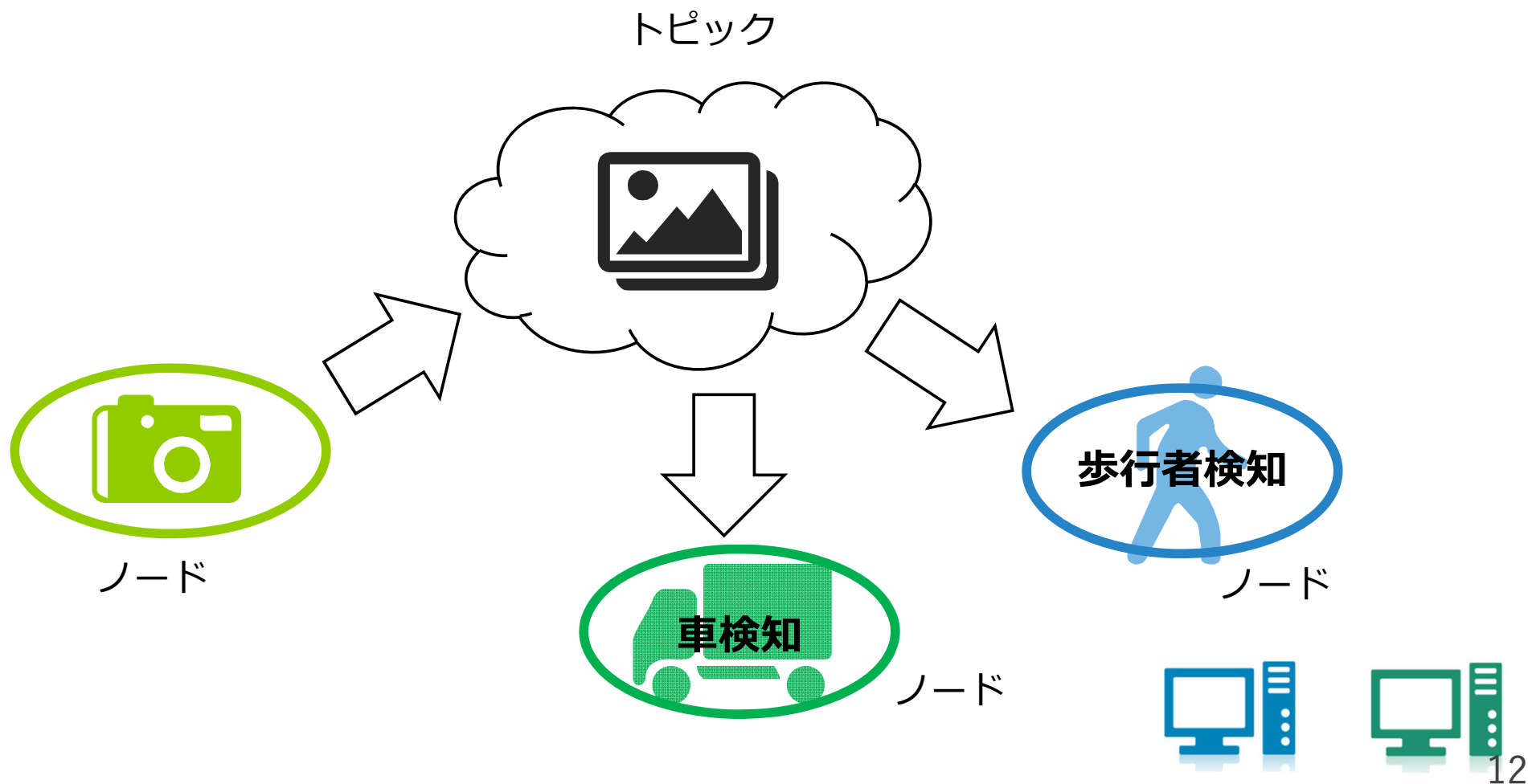
➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

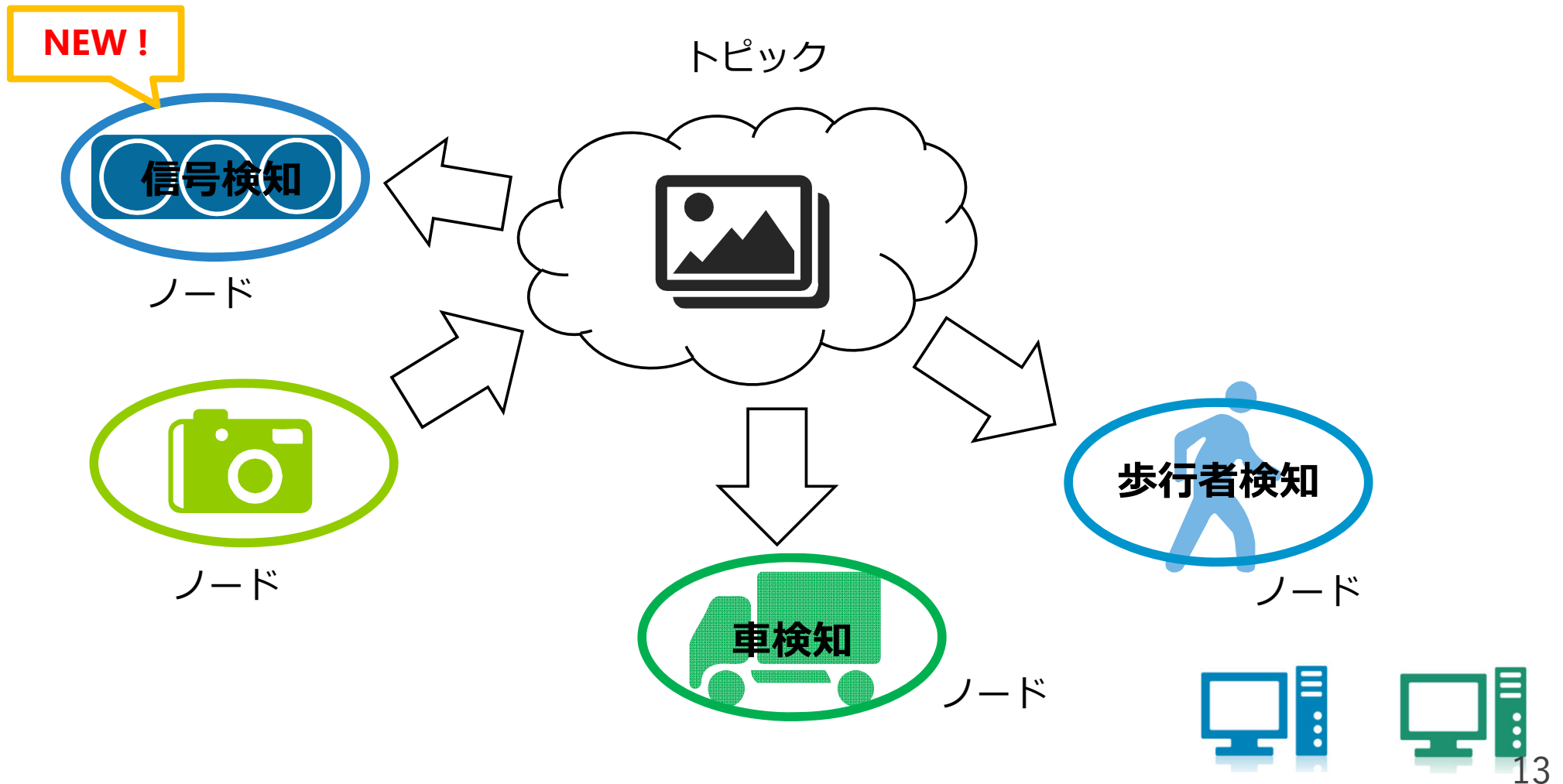
➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



## データの保存 : rosbag

実データ（トピック情報）を保存可能

## 視覚化・シミュレーション

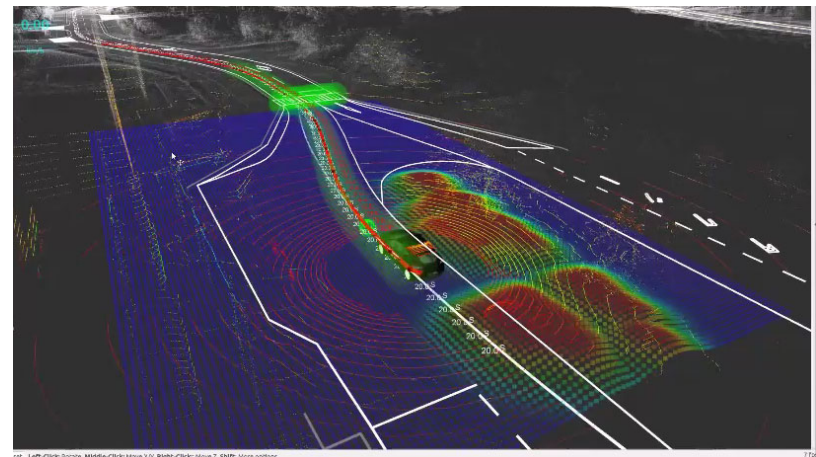


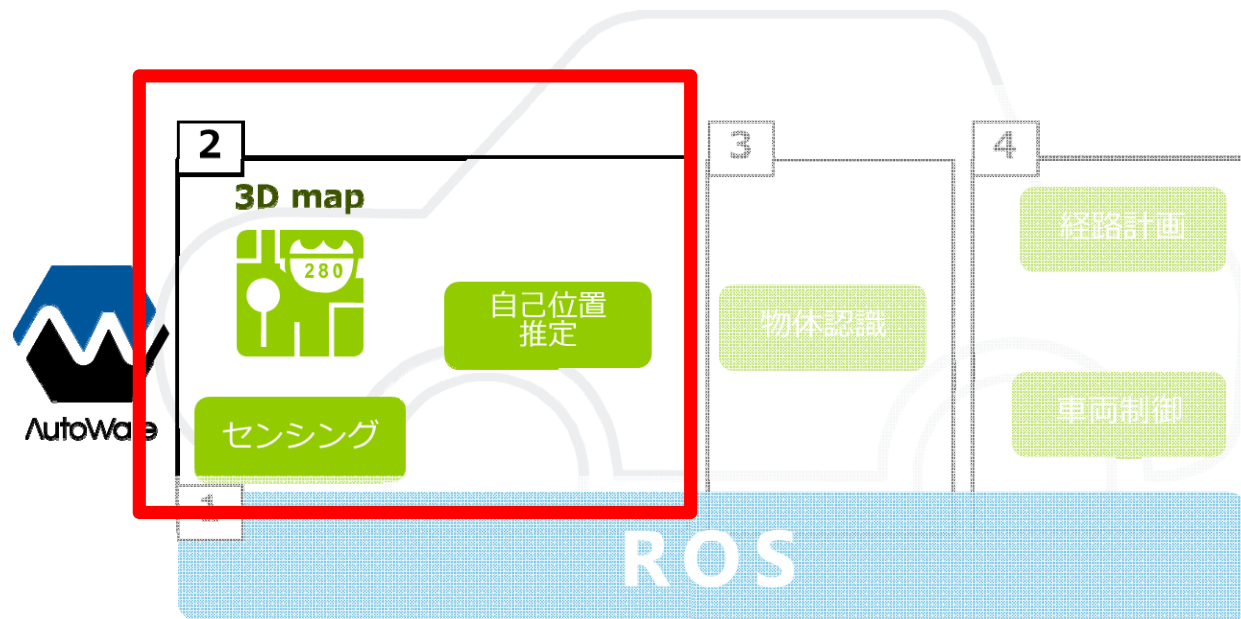
## RViz: 3D視覚化ツール

簡単にシステム状態を視覚化可能

[再生データ]

- 記録したセンサデータ（**rosbag** ファイル）
- 指定した値のデータ





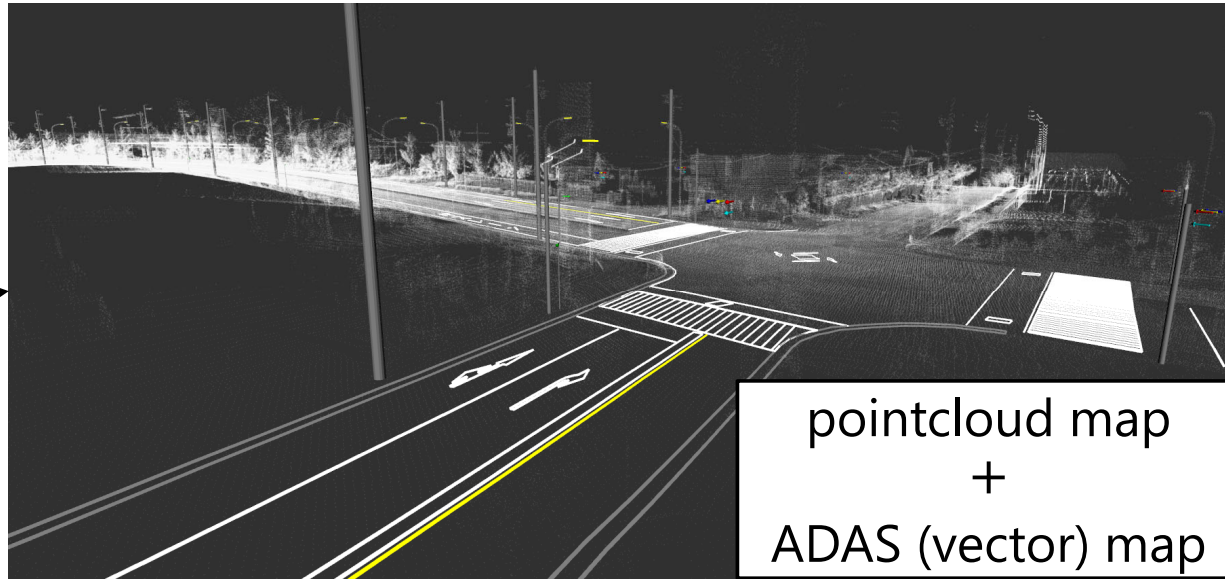
## 自己位置推定技術

高精度3次元地図

位置推定手法

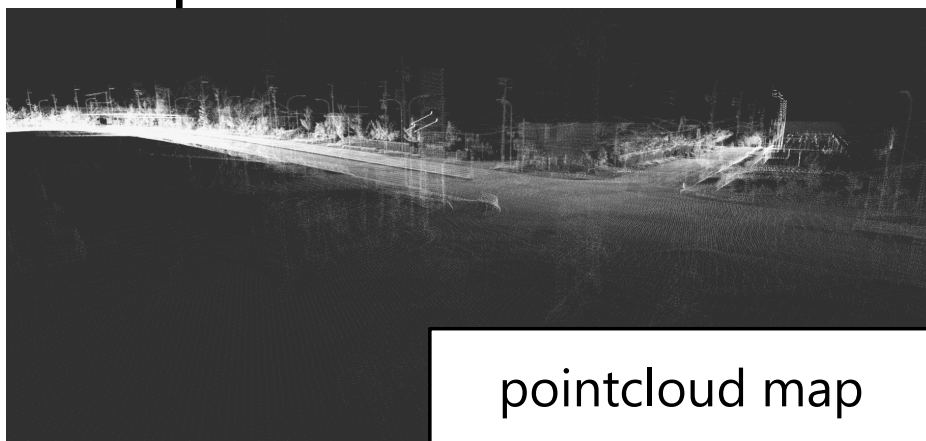
LiDARによる自己位置推定

# 高精度 3 次元地図



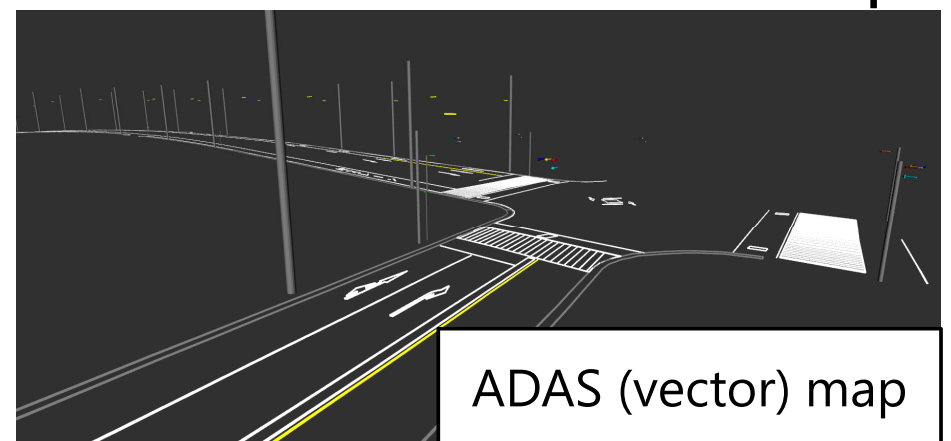
## ● ポイントクラウド地図

- ✓ 3次元座標(緯度・経度・標高)
- ✓ RGB値

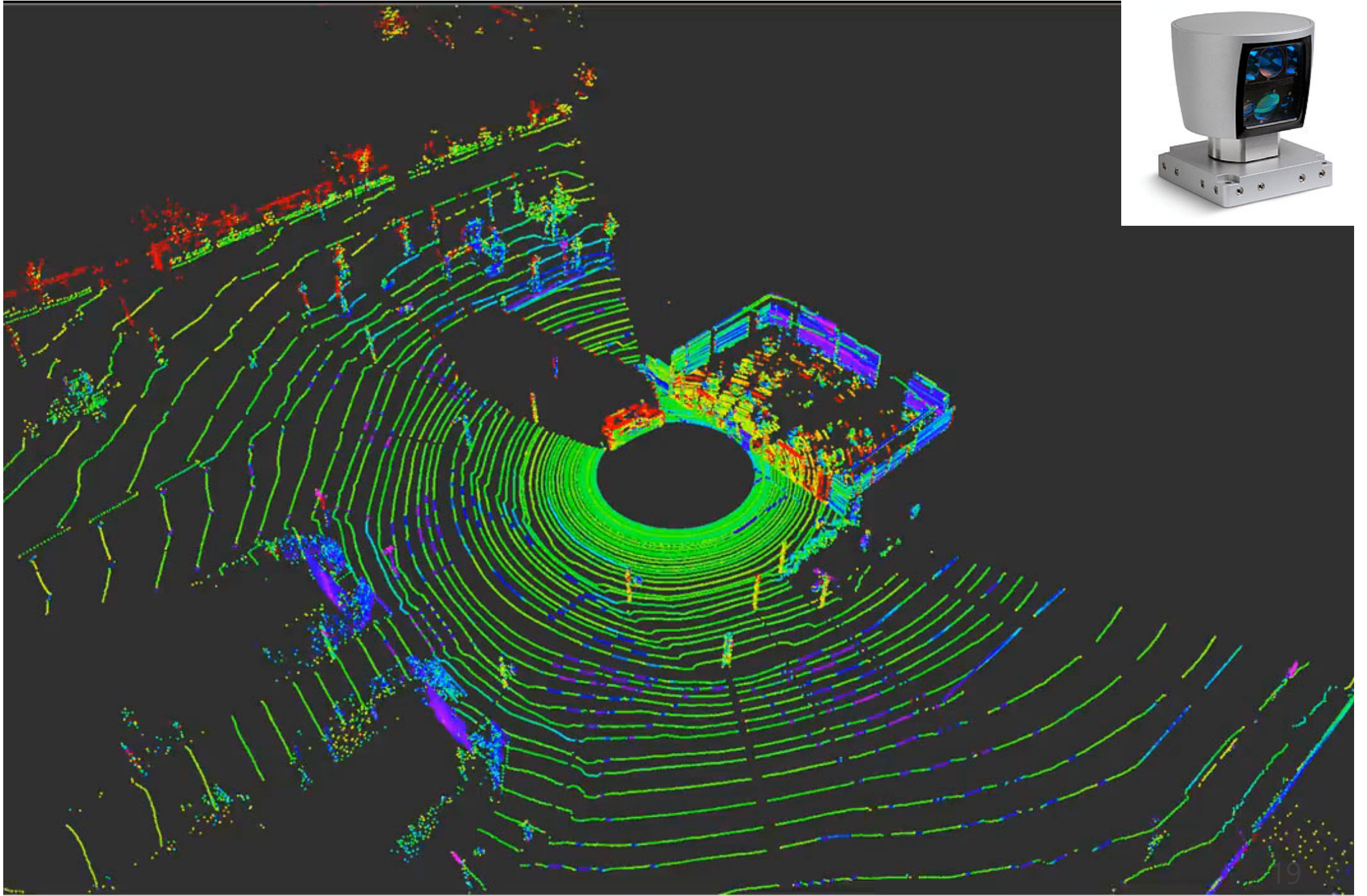


## ● ADAS地図 - 点群地図から地物を抽出

- ✓ 信号、路面標示 etc.

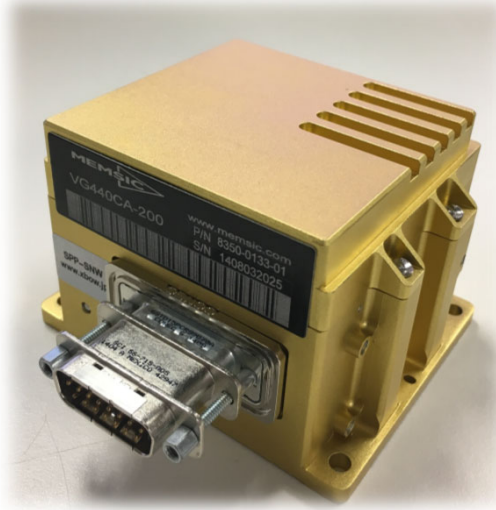
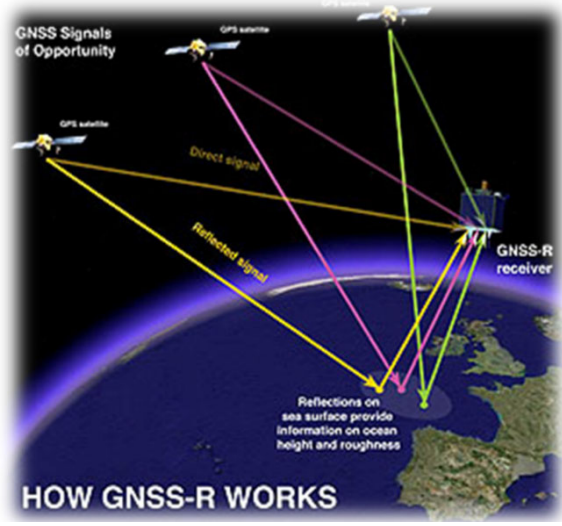
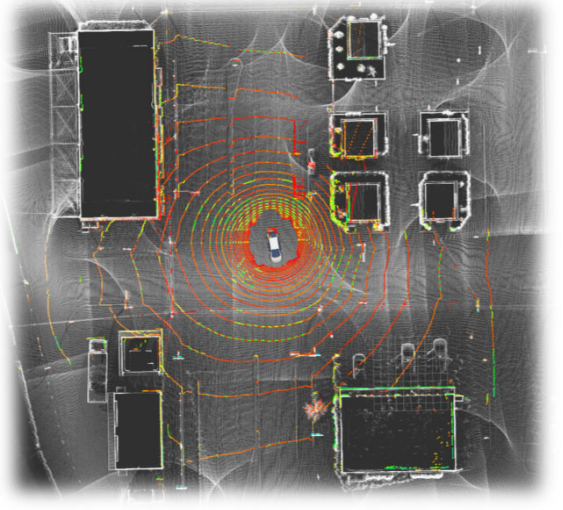


# | センシング : LiDAR



# 位置推定手法

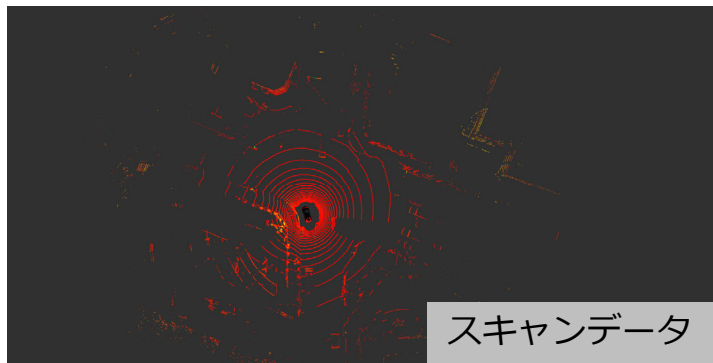
位置推定には様々な手法があり、用いられるセンサも異なる

	デッドレコニング (自律航法)	GNSS (Global Navigation Satellite System)	スキャンマッチング
用いられる センサ	IMU (Inertial Measurement Unit) ホイールエンコーダ等	GNSS受信機	LiDAR
利点	どこでも使える	地図がなくても 使える	高精度な 位置推定が可能
欠点	誤差の蓄積大	<ul style="list-style-type: none"> <li>信号の受信状況に影響</li> <li>トンネル内では使えず</li> </ul>	<ul style="list-style-type: none"> <li>地図データが必須</li> <li>特徴のないエリアでは使えず</li> </ul>
		 <p><a href="http://noc.ac.uk/conference/ocean-observation-workshop">http://noc.ac.uk/conference/ocean-observation-workshop</a></p>	

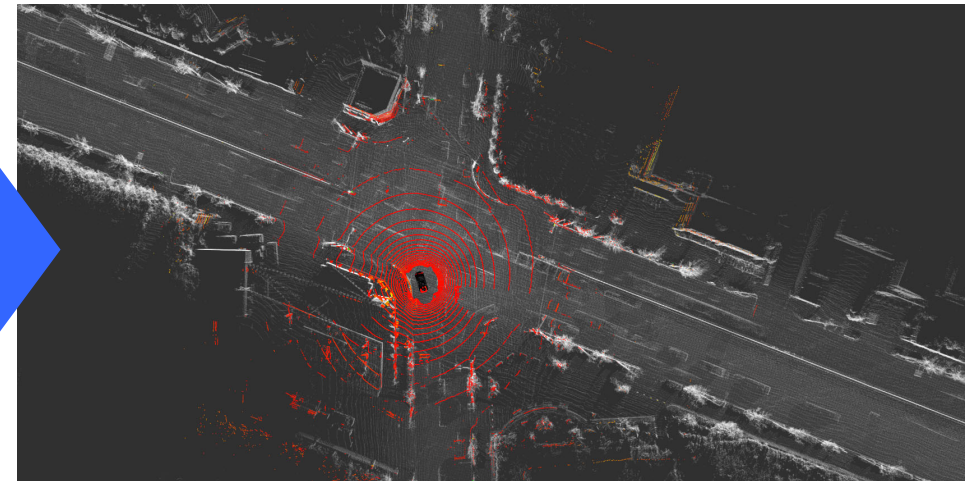


# LiDARによる自己位置推定

地図データとスキャンデータがきれいに重なる座標変換を計算し、  
地図内の位置・向きを算出



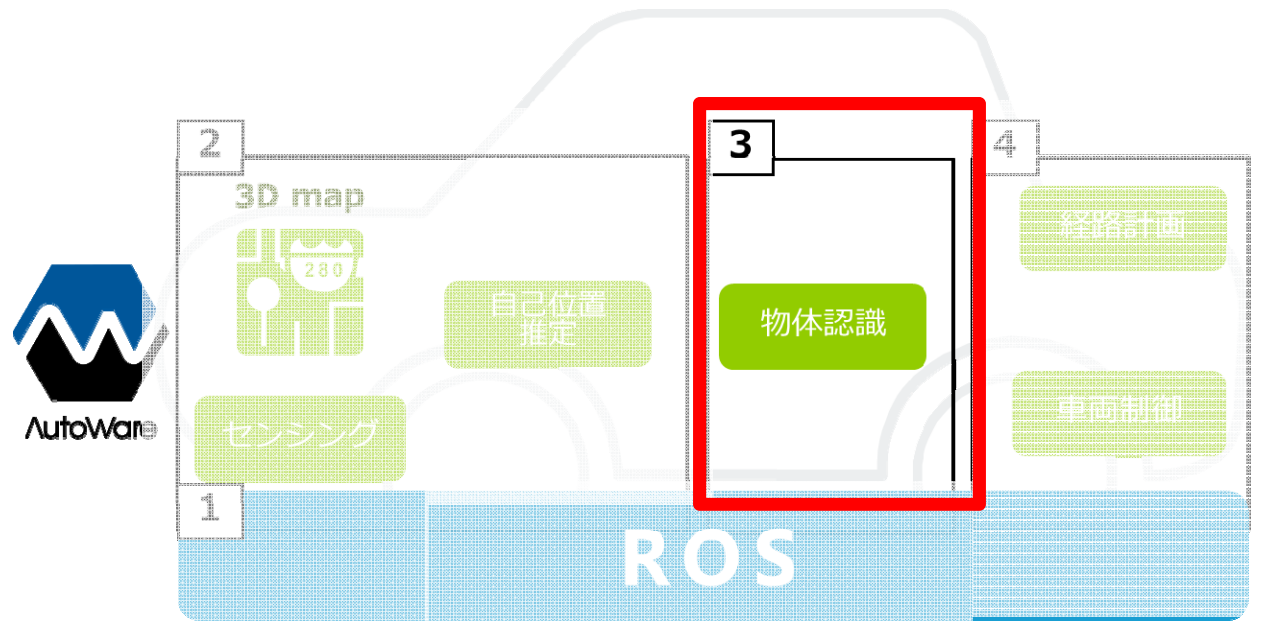
スキャン  
マッチング



3次元地図とスキャンデータの座標変換を計算  
車両の位置・向き

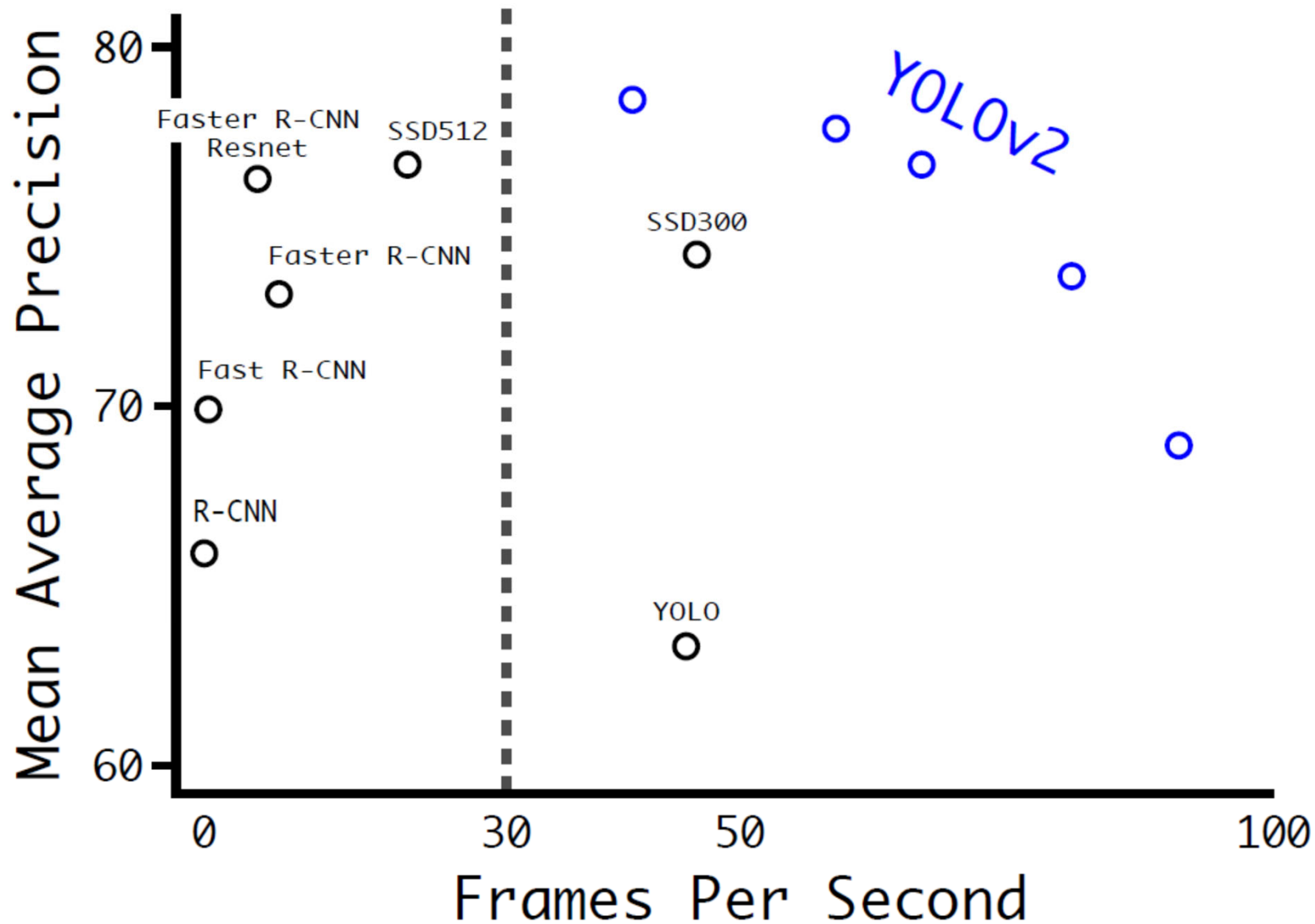
## 代表的なスキャンマッチングのアルゴリズム

- ICP (Iterative Closest Point) - P.J. Besl et al. (1992)
- 2D-NDT (Normal Distributions Transform) - P. Biber et al. (2003)
- 3D-NDT - E. Takeuchi et al. (2006) , M. Magnusson et al. (2007)



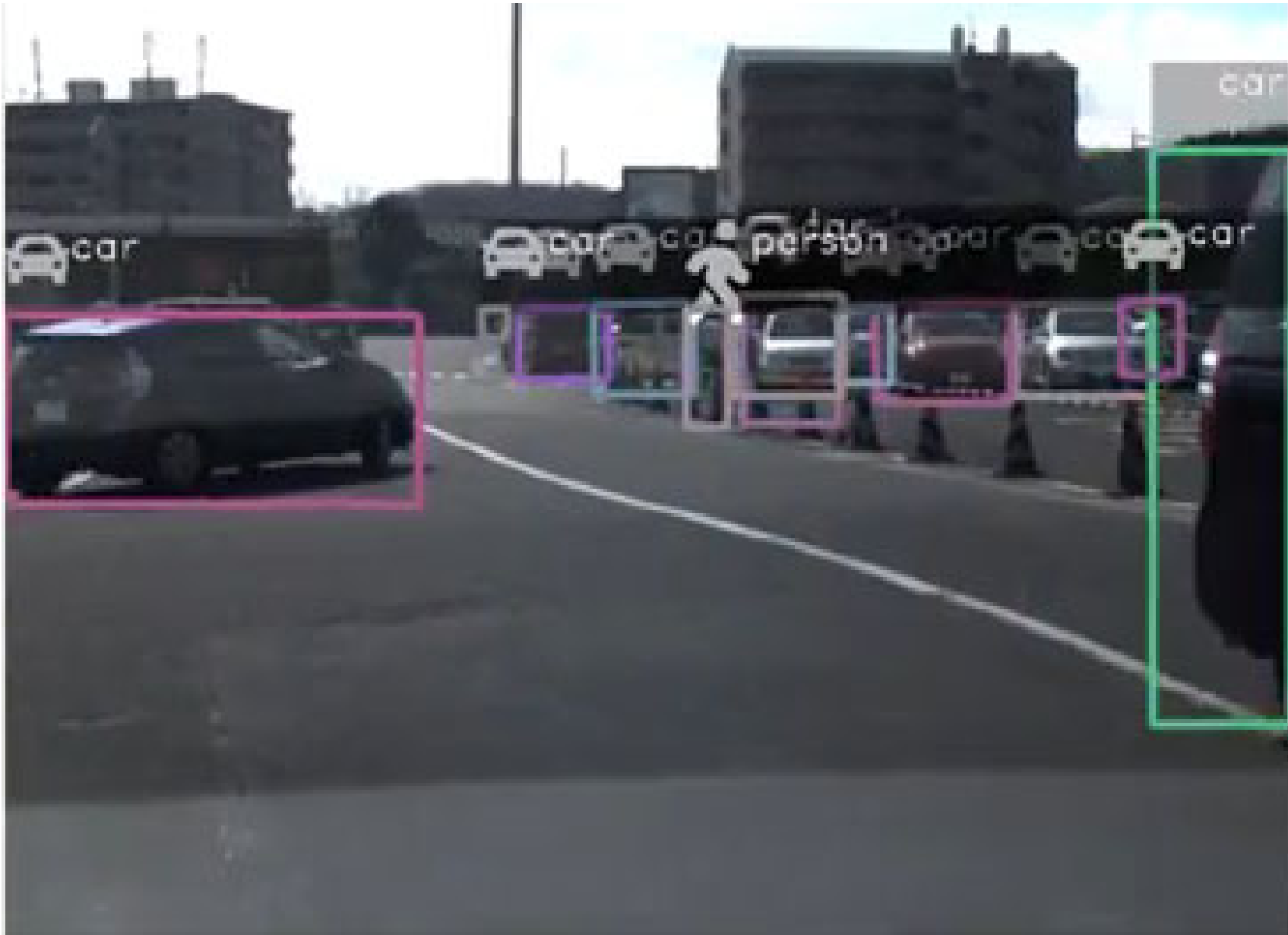
# 物体認識

# 最近のディープラーニング手法の傾向



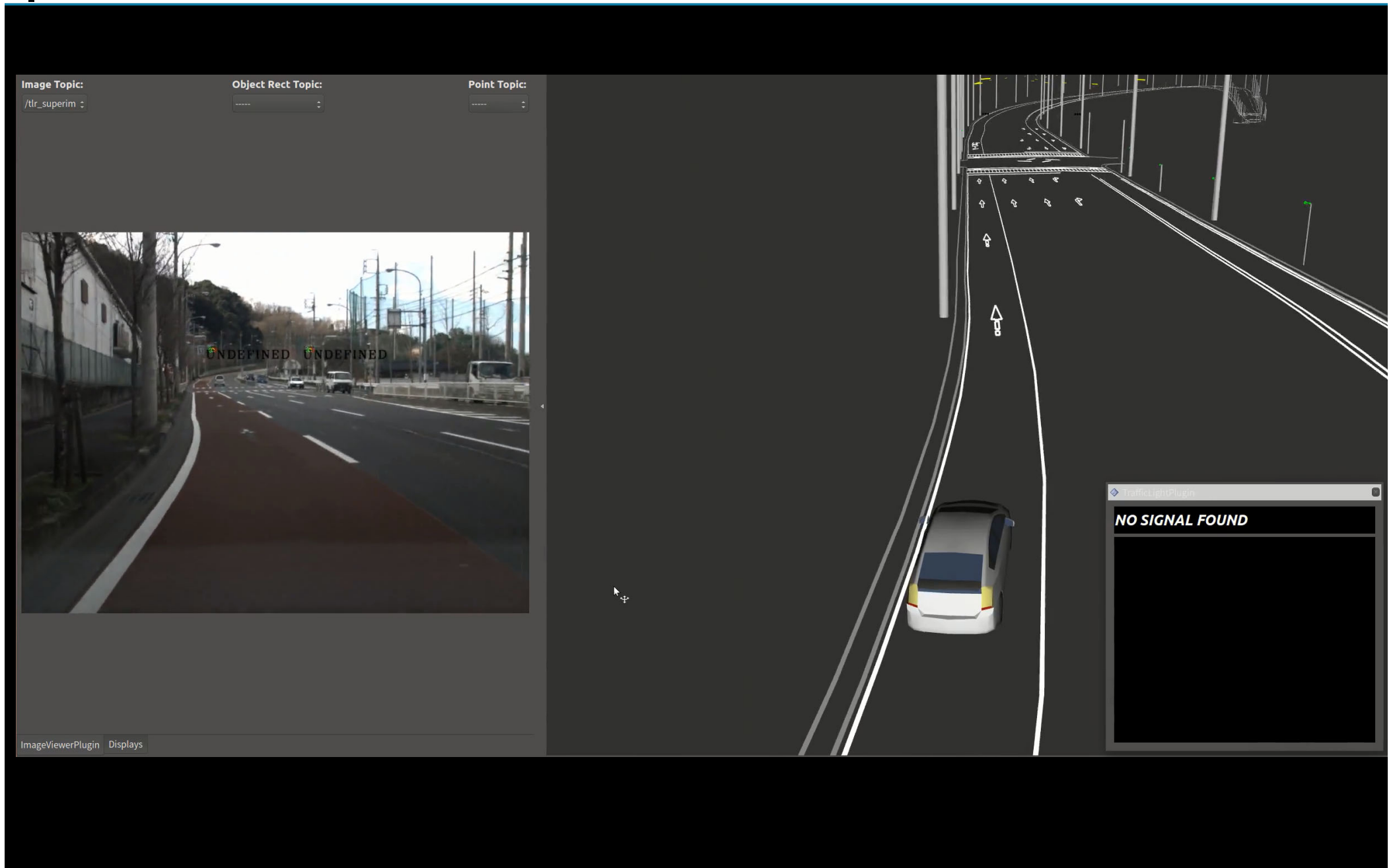
J. Redmon, and A. Farhadi, "YOLO9000: Better, Faster, Stronger",  
*arXiv preprint arXiv:1612.08242*, 2016

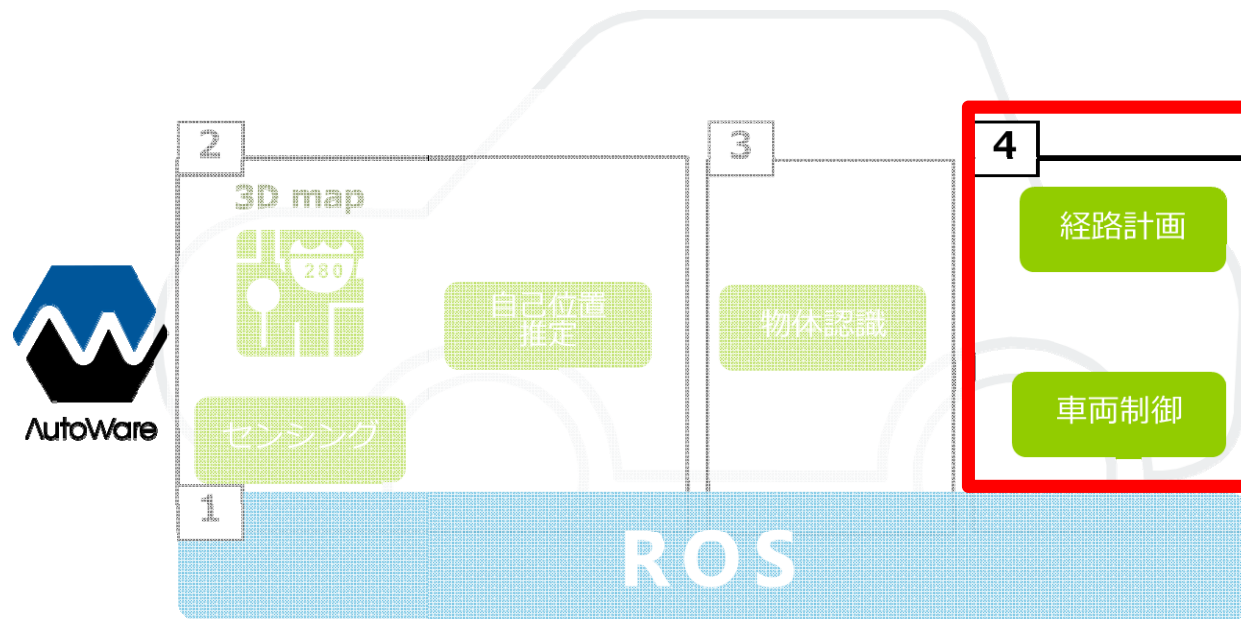
# ディープラーニング (YOLOv3) を用いた物体認識：最新



<https://www.youtube.com/watch?v=KFfD3Mkkz4Y>

# Autowareの信号認識





## 経路計画・経路追従

# Pure Pursuit

※ R Craig Coulter. "Implementation of the Pure Pursuit Path Tracking Algorithm".  
Technical Report CMU-RI-TR-92-01, Robotics Institute, Pittsburgh, PA, January 1992.

経路上の目標点と自己位置の情報を基に車両制御信号を計算

