

自動運転ソフトウェアプラットフォーム入門 ~要素技術と動作の仕組み~

安積卓也

大阪大学大学院基礎工学研究科



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
高田研究室：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：兼任
- 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

主査



アイサンテクノロジー株式会社

副主査

名古屋大学

株式会社 I I C

アイサンテクノロジー株式会社

アイシン・エイ・ダブリュ株式会社

イーソル株式会社

インクリメント P 株式会社

インテル株式会社

SBドライブ株式会社

キーノ コンサルティング

株式会社 ZMP

測位衛星技術株式会社

株式会社 ティアフォー

HERE Japan 株式会社

北陽電機株式会社

名古屋大学

長崎大学

大阪大学

産業技術総合研究所

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
レベル1	いずれか一つ	主体
レベル2	複数	主体
レベル3	すべて (危険回避はドライバー)	あり
レベル4	すべて	なし

自動ブレーキ

クルーズ
コントロール

公道実験

完全自動運転



レベル4を2つに分ける動きも：※SAE J3016 (2016)

レベル	システム	ドライバ	場所
レベル4	すべて	なし	限定
レベル5	すべて	なし	全て

自動運転バス

自動運転
タクシー

レベル4：一般道での完全自動運転の難しさ

専用道



走行レーン
が**明確**



同じ進行
方向



対向車の
分離

一般道の難しさ

交通ルールが多様



ルールが**複雑**

様々な移動体が混在



歩行者



様々な移動体

走行環境が整備されていない



車線が**無い**



路面状態

課題傾向

網羅的認識



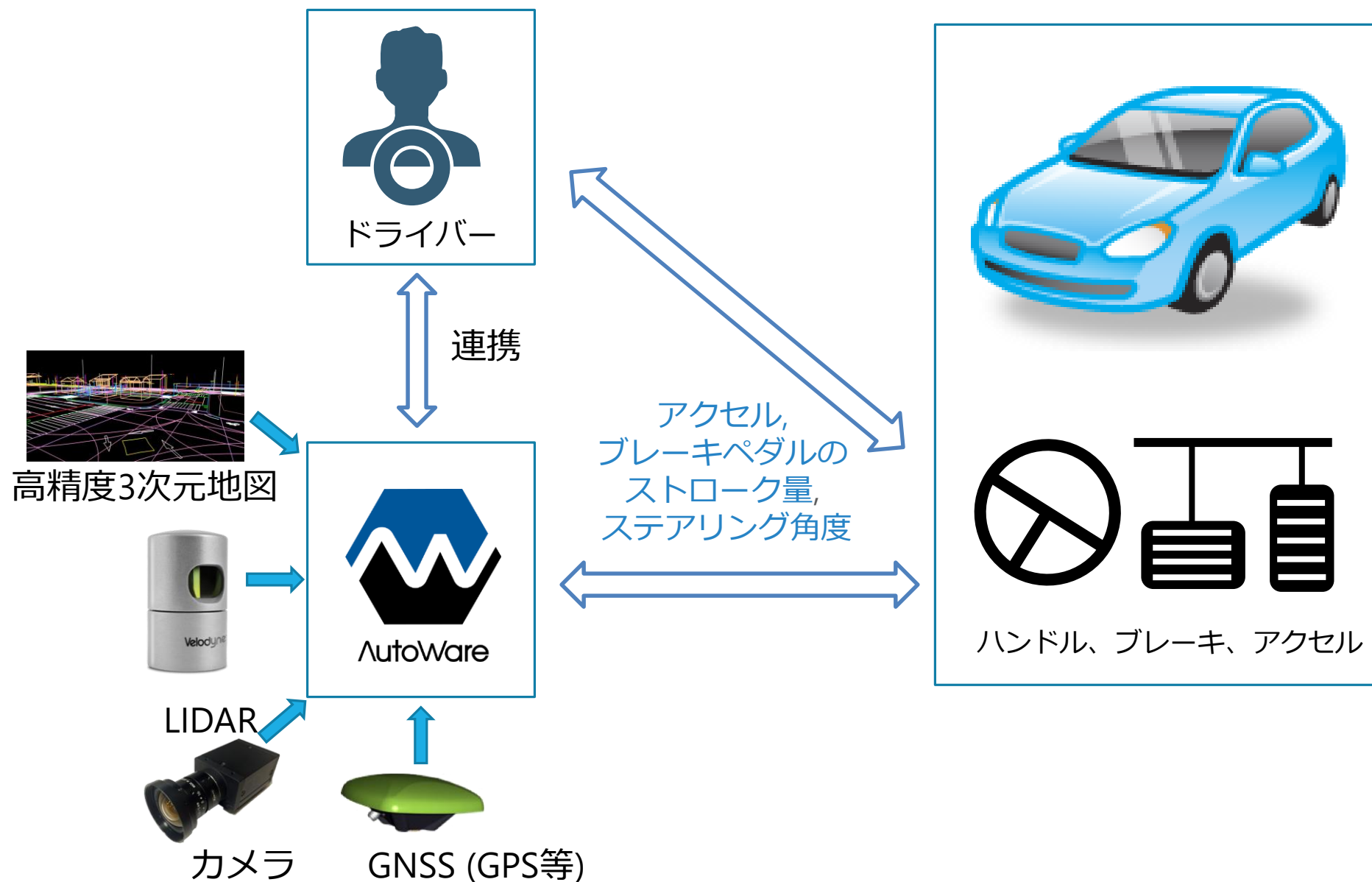
多様な環境



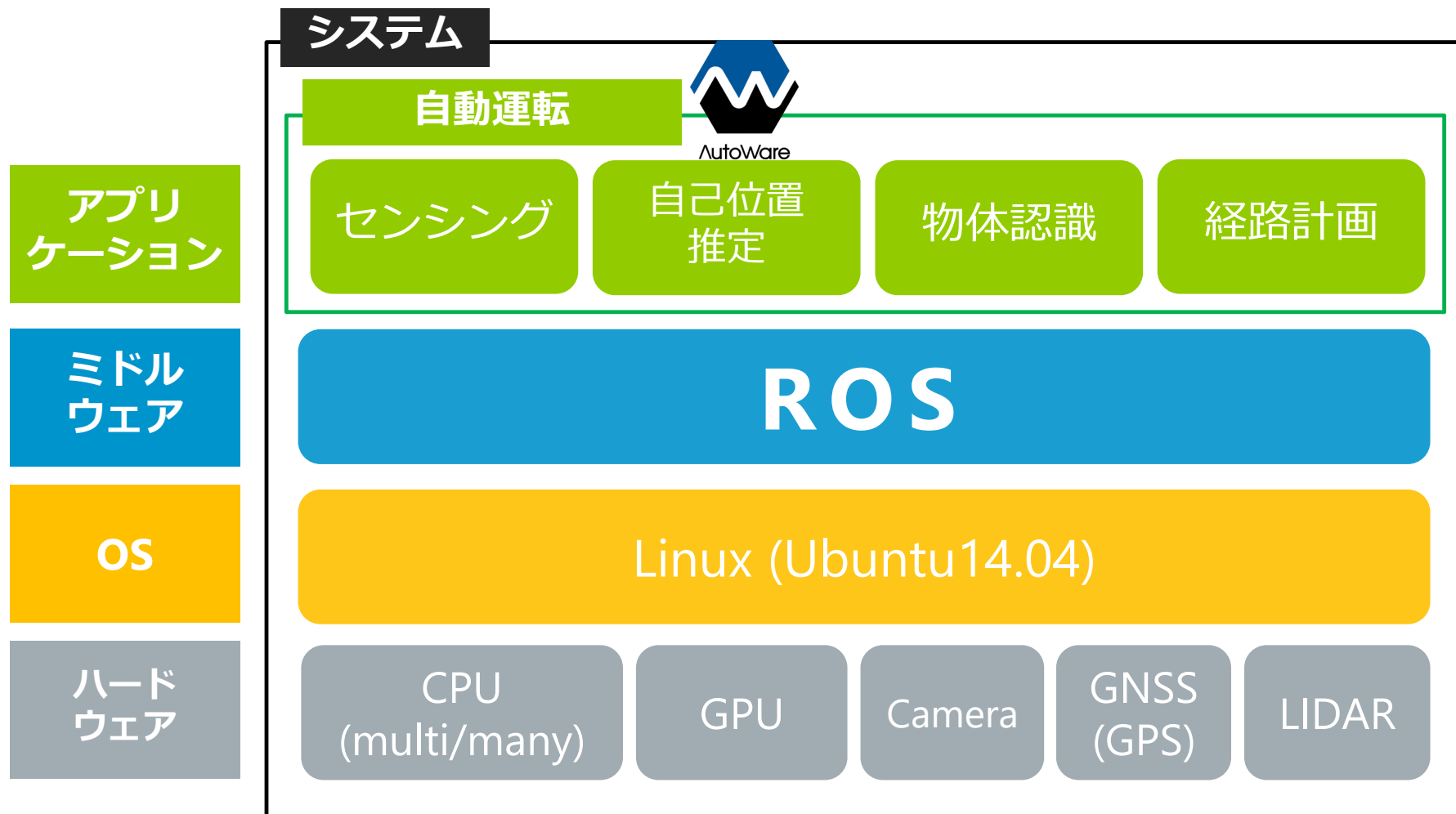
賢い判断



自動運転ソフトウェアの役目

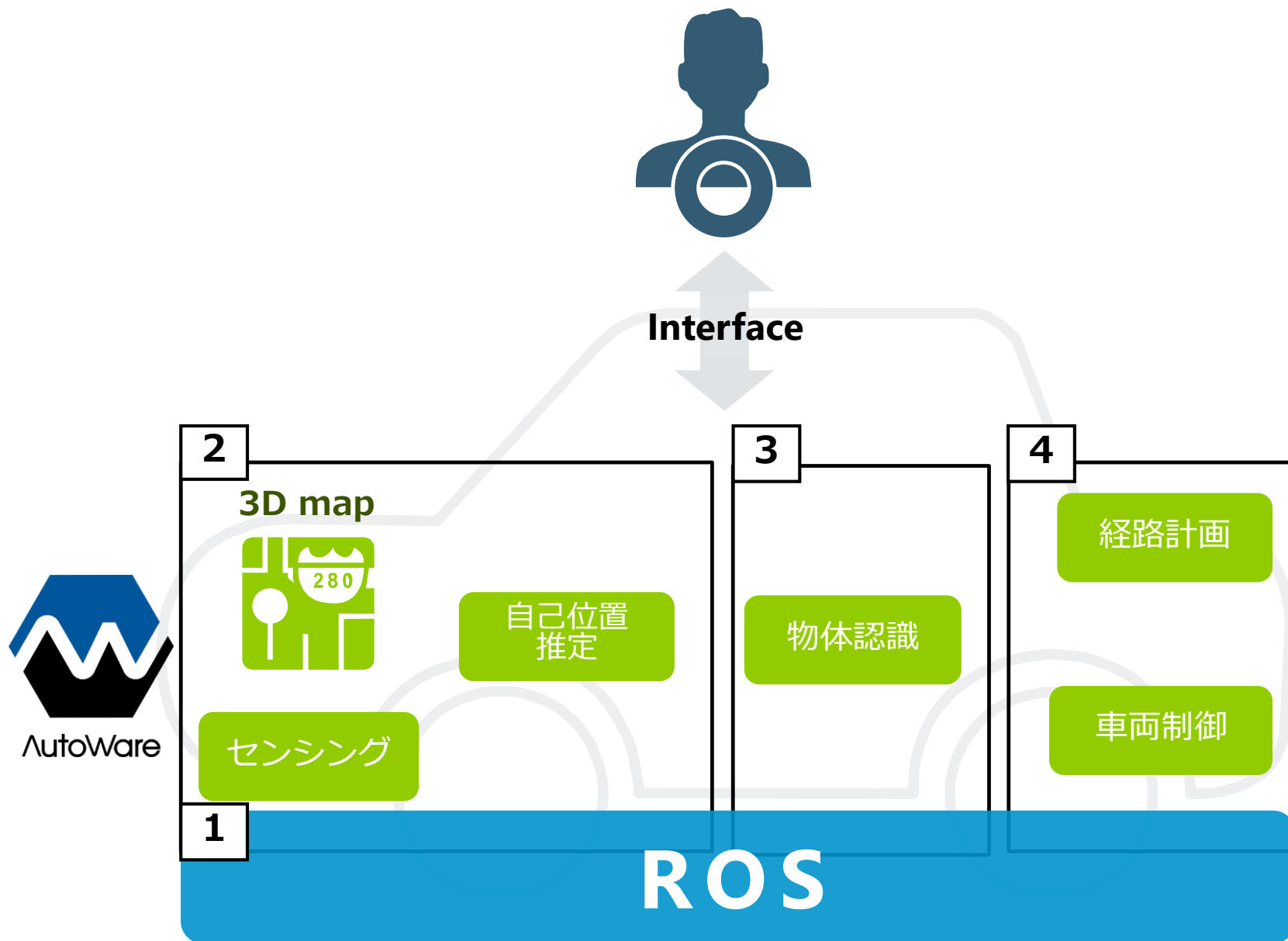


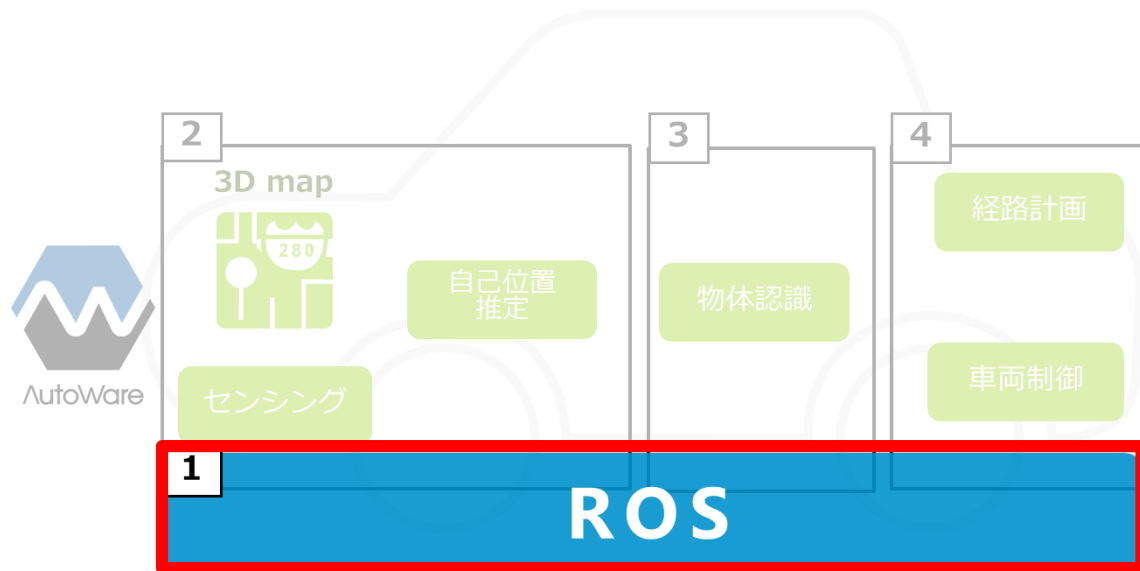
Autowareの構成



<https://github.com/CPFL/Autoware>

本日の講演の範囲





ROS (Robot Operating System)

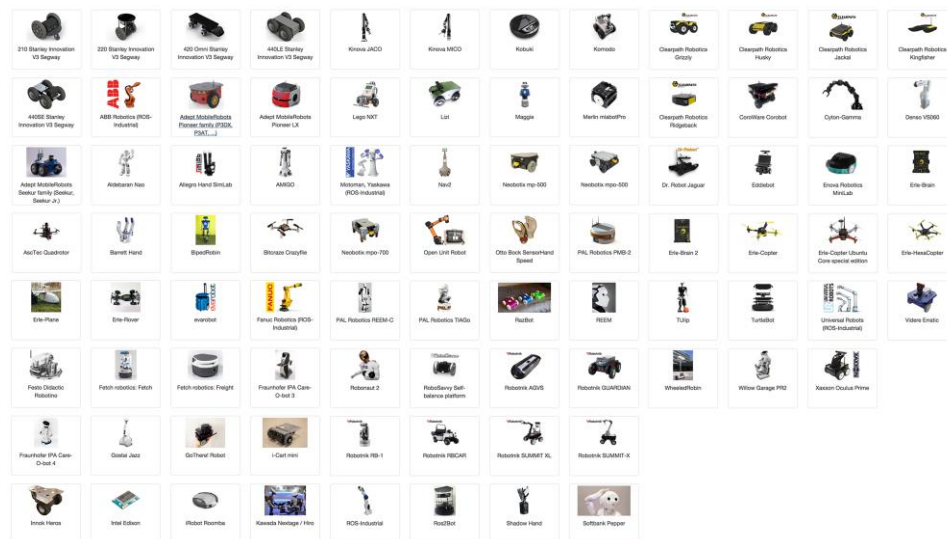
ROS (Robot Operating System)

: 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)

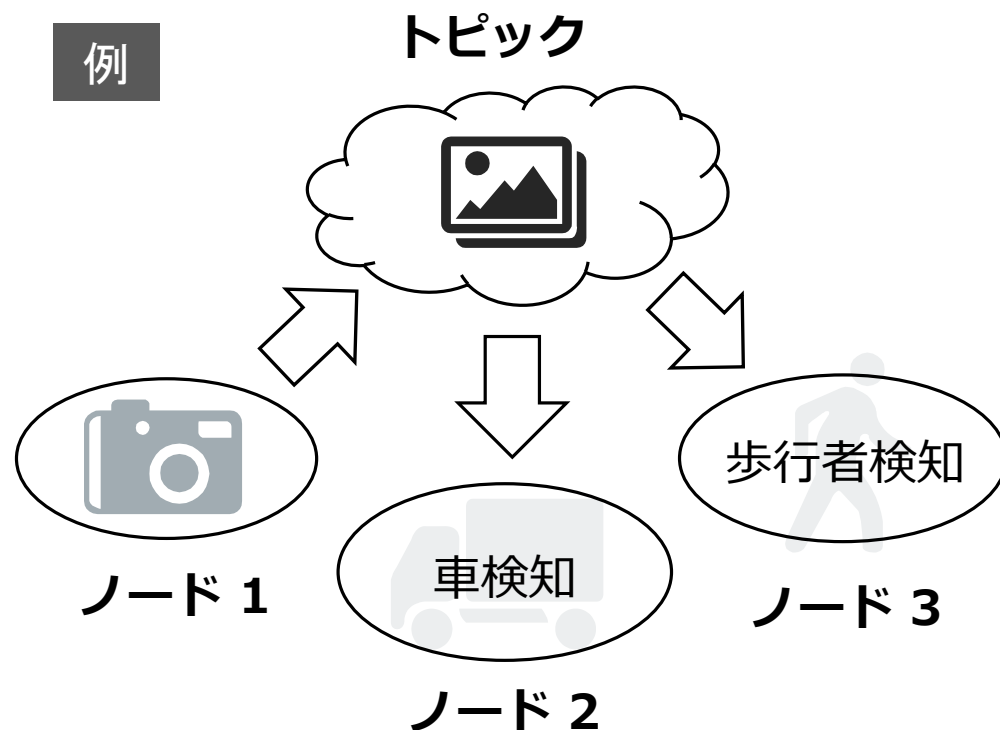
: ロボット開発におけるライブラリやツールを提供

ROS

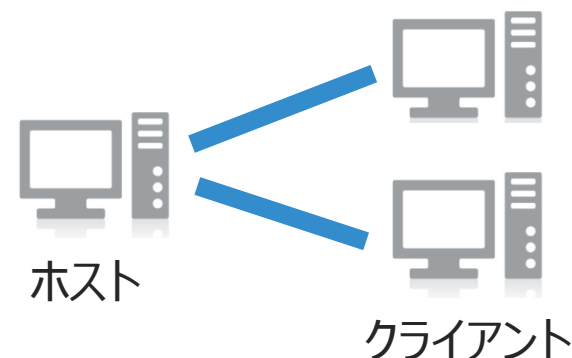
Publish / Subscribe モデル

- 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

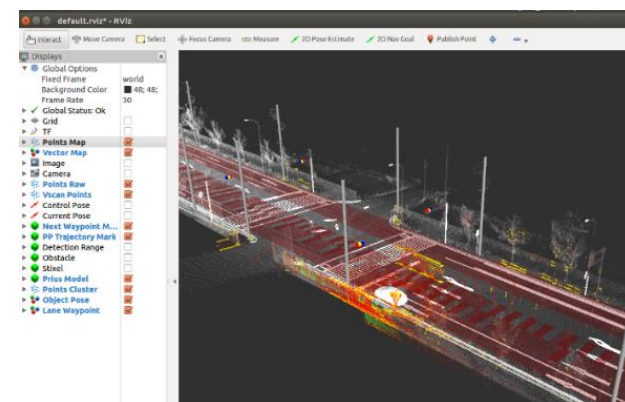
例



分散システム



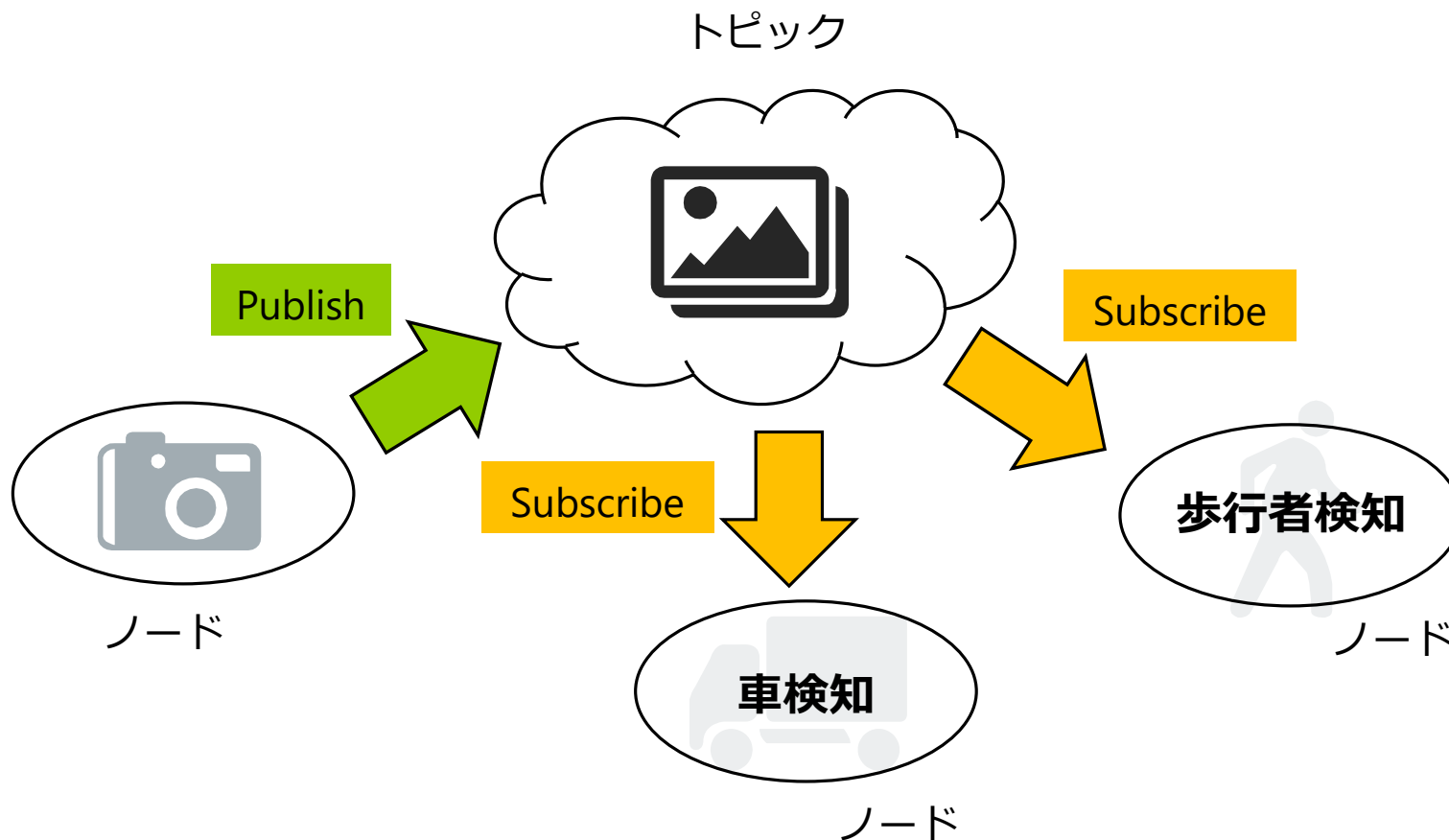
視覚化・シミュレーション



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。

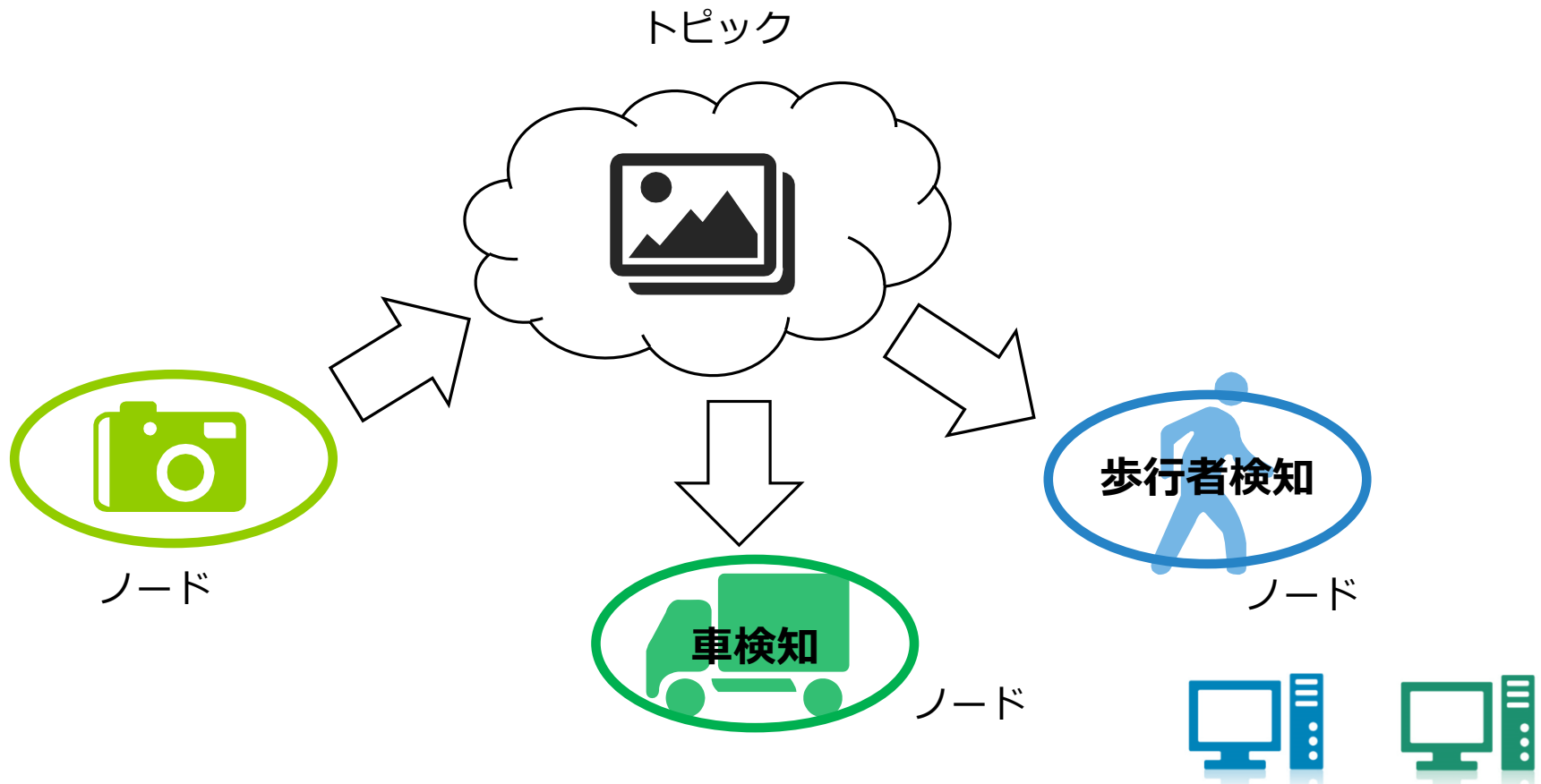
➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



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

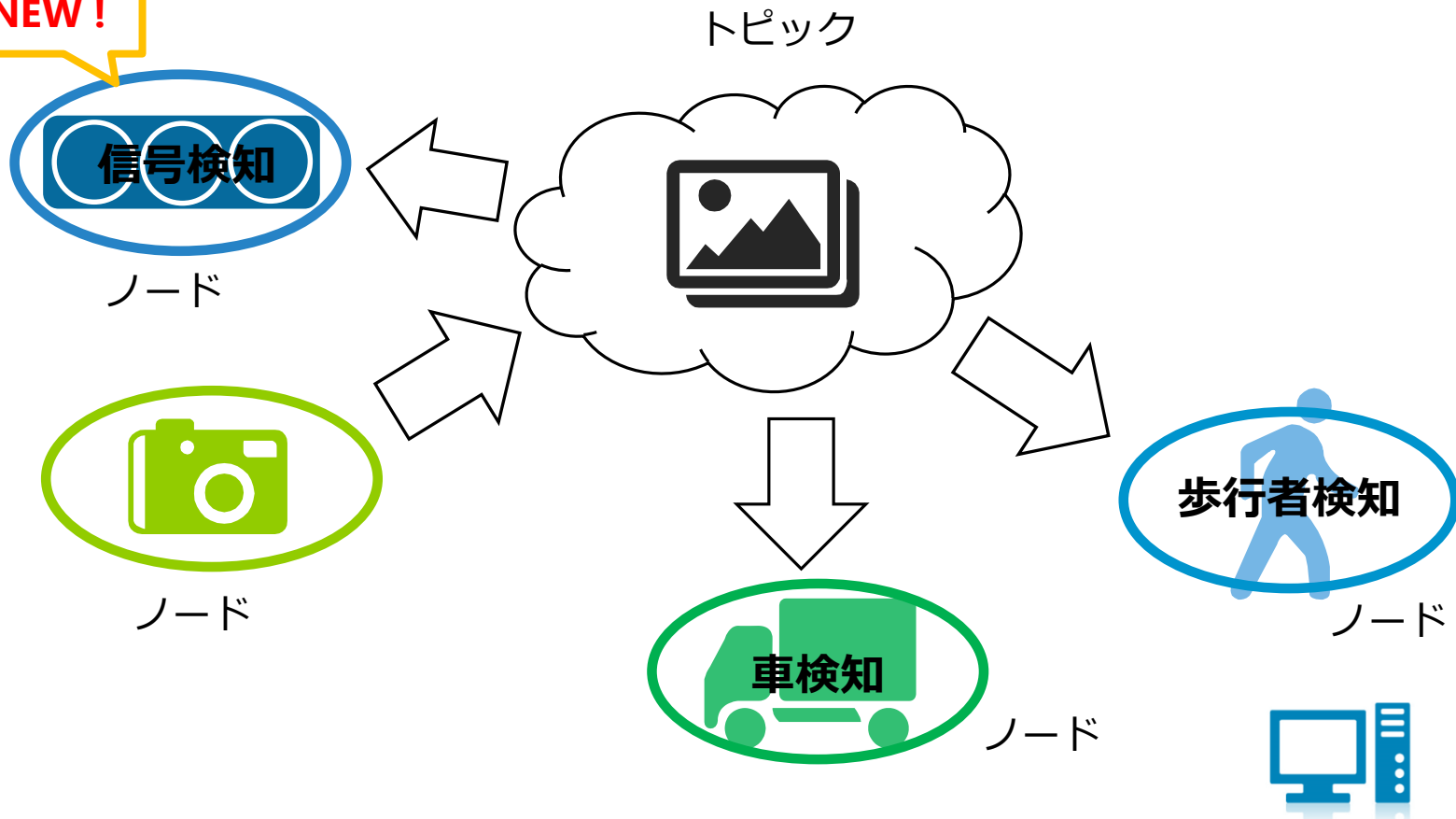


Publish / Subscribe モデル

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

NEW !



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

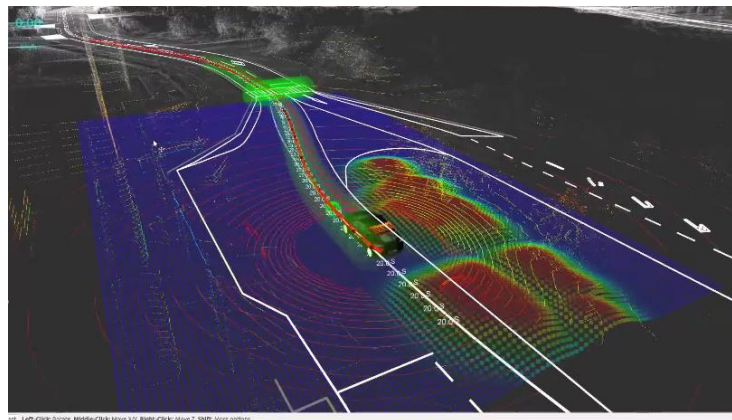


RViz: 3D視覚化ツール

簡単にシステム状態を視覚化可能

[再生データ]

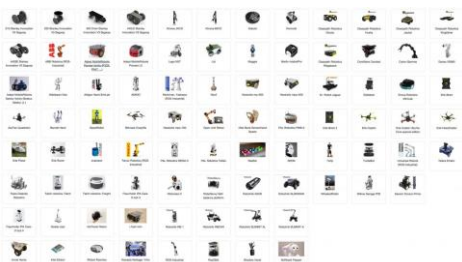
- 記録したセンサデータ（**rosbag** ファイル）
- 指定した値のデータ



ROS の 特長

ハードウェア

様々なロボットやセンサをサポート



パッケージ

2,000を超えるソフトウェアパッケージで効率的開発

ライブラリ

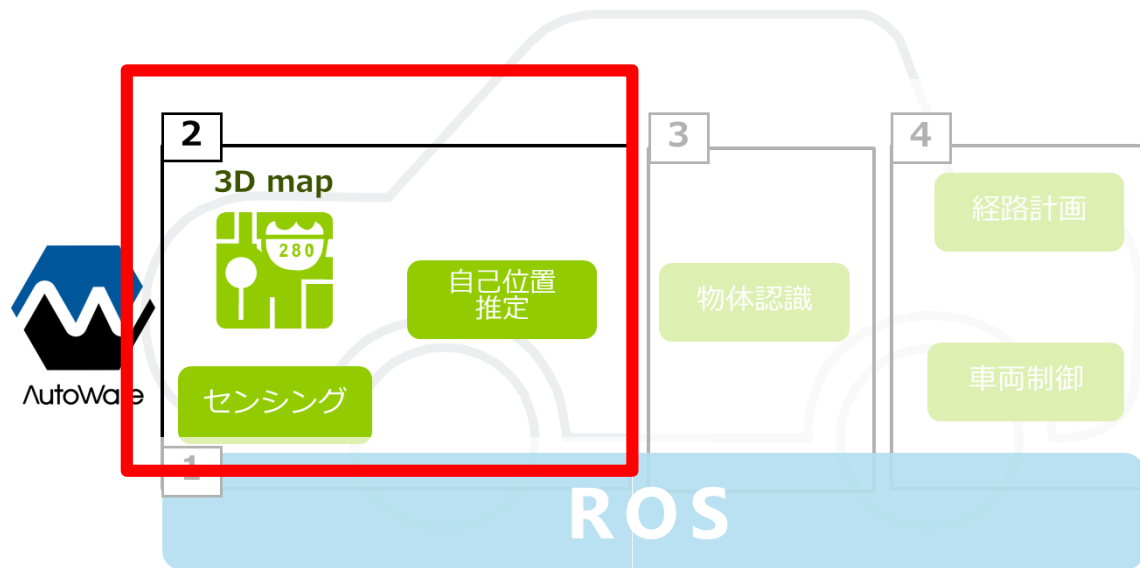
座標変換・画像処理・点群処理
など豊富にサポート



多くのソフトウェアが
オープンソースで

豊富なパッケージ
(デバイスドライバやライブラリ)





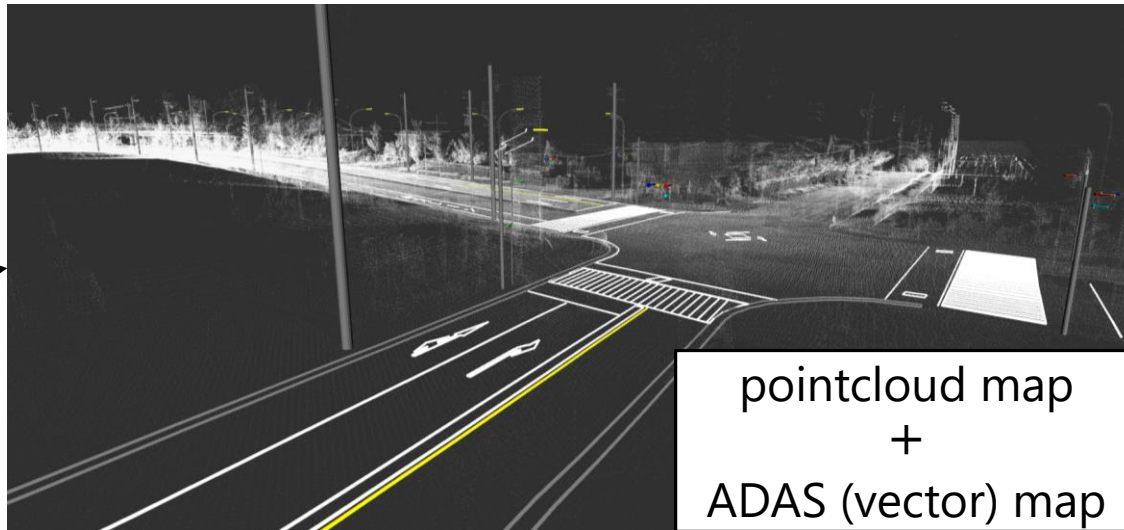
自己位置推定技術

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

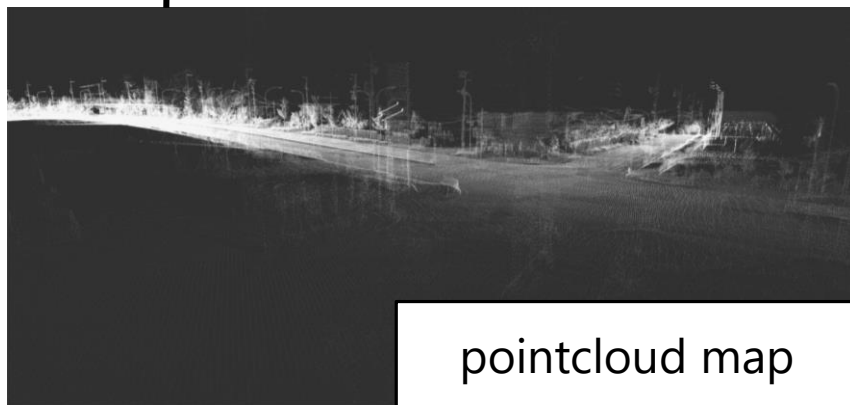


● ポイントクラウド地図

- ✓ 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

LIDAR – Light Detection and Ranging

レーザーを対象物に照射し、散乱光を測定することにより、
対象物までの距離(や性質)を取得



レーザー照射部・受信部

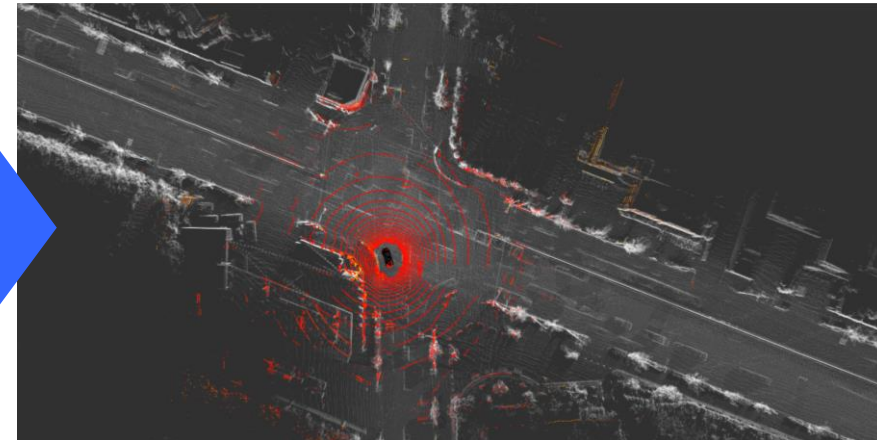
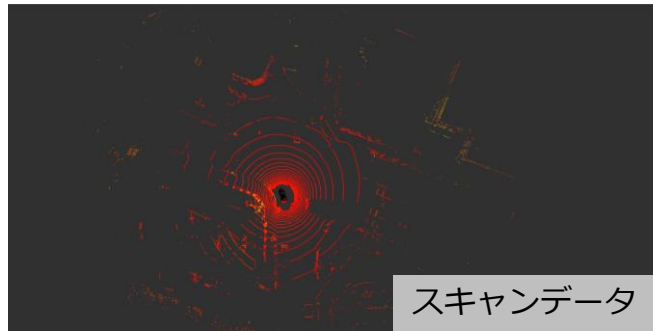
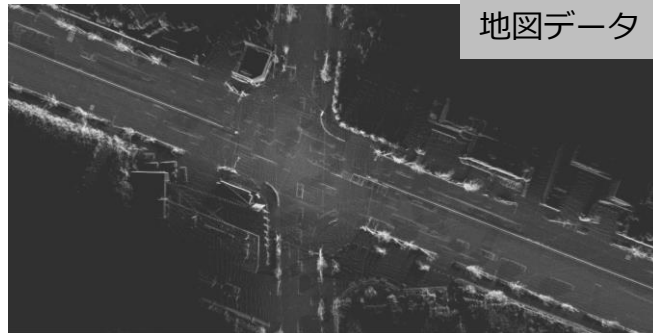


対象物までの距離、位置、反射強度を取得

※ Rader – Radio Detection and Ranging レーザーではなく電波（波長が長い）を用いる

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



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

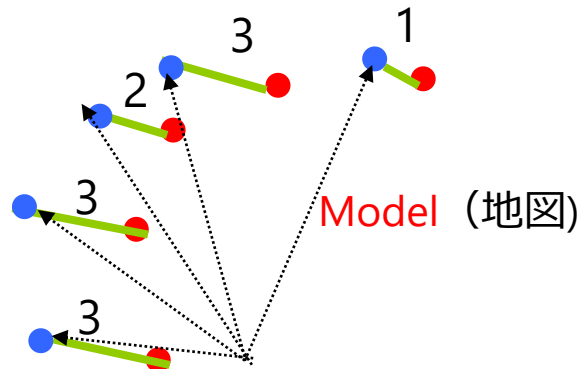
- ICP (Iterative Closest Point) - P.J. Besl et al. (1992)
- 2D-NDT (Normal Distributions Transform) - P. Biber et al. (2003)
- 3D-NDT - E. Takeuchi et al. (2006) , M. Magnusson et al. (2007)

ICPスキャンマッチングのアルゴリズム

- 地図・スキャンの最近傍点 (Nearest Neighbor) を求める

距離の2乗の合計値が小さいところを探す

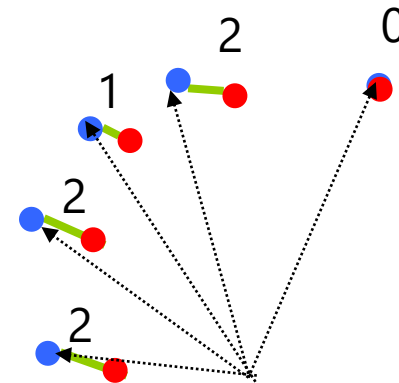
$$3^2 + 3^2 + 2^2 + 3^2 + 1^2 = 30$$



並進: T 回転: R



$$2^2 + 2^2 + 1^2 + 2^2 + 0^2 = 13$$

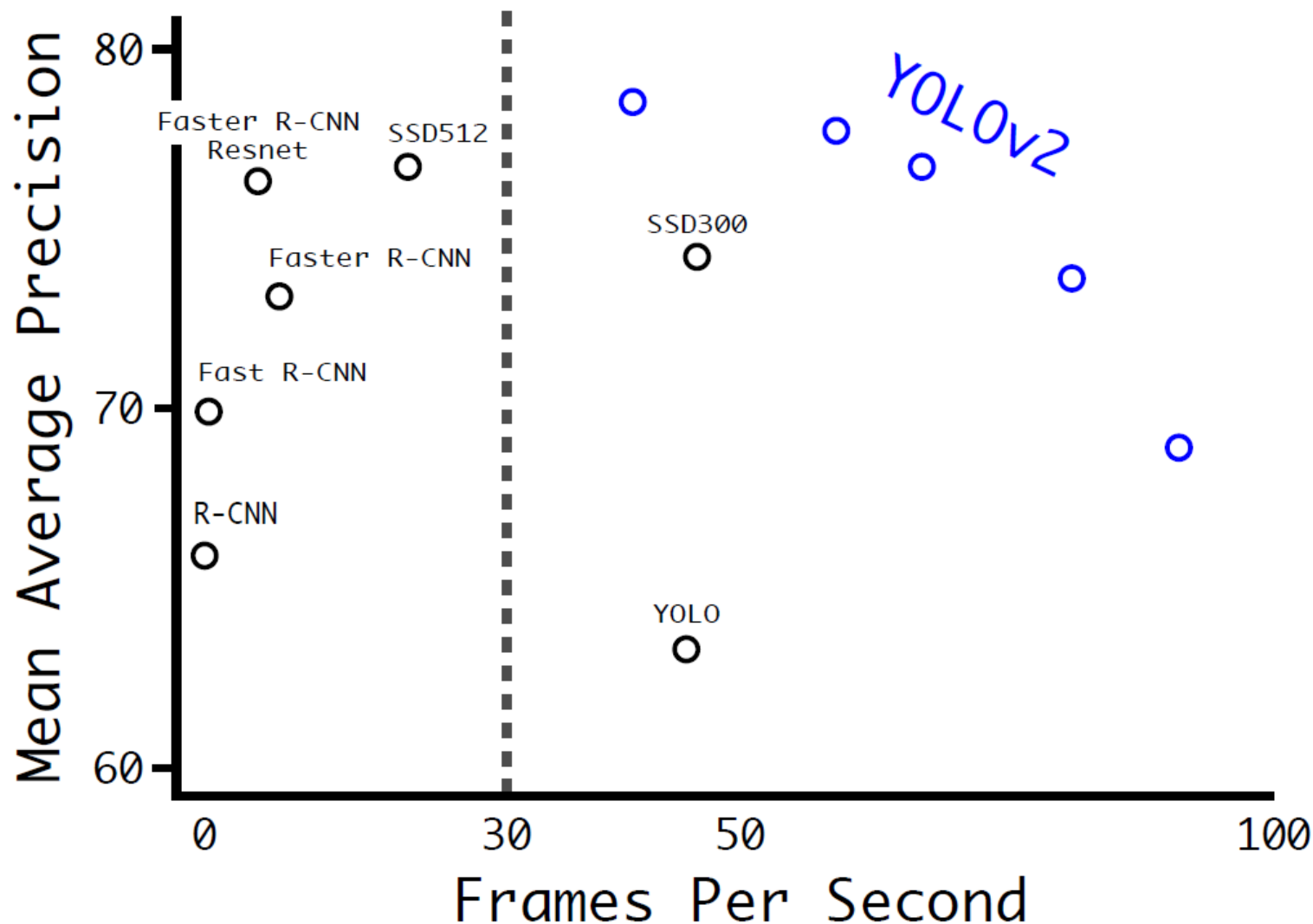


※最近ではICPをベースとした確率密度を利用した方法 (NDT) を採用し簡略化している



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

Autowareの認識の例 (SSD)

The screenshot displays the Autoware ROS interface. The left panel shows a 3D point cloud visualization of a street scene. The right panel shows a 2D camera view with object detection results. The interface includes a top toolbar with various tools like Move Camera, Select, Focus Camera, Measure, 2D Pose Estimate, 2D Nav Goal, and Publish Point. The right panel has a 'Views' section with a dropdown menu set to 'XYOrbit (rviz)'. Below this, there's a table of view parameters:

Current View	
Near Clip Distance	0.01
Target Frame	map
Distance	75.4521
Yaw	2.93859
Pitch	0.880203
Focal Point	11.716; -0.86031; -1.1921e-07

Below the view parameters, there are buttons for 'Save', 'Remove', and 'Rename'. The 'ImageViewerPlugin' section shows the 'Image Topic' as '/image_raw' and the 'Object Rect Topic' as '/obj_car/image_obj'. The main image shows a street scene with several cars detected and labeled with their confidence scores:

- car:1
- car:0.56
- car:0.53
- car:0.82
- car:0.9
- car:0.96
- car:0.75
- car:0.82

The bottom status bar shows the following information:

- Time: 1427157680.73
- ROS Elapsed: 30.33
- Wall Time: 1481899375.02
- Wall Elapsed: 45.17

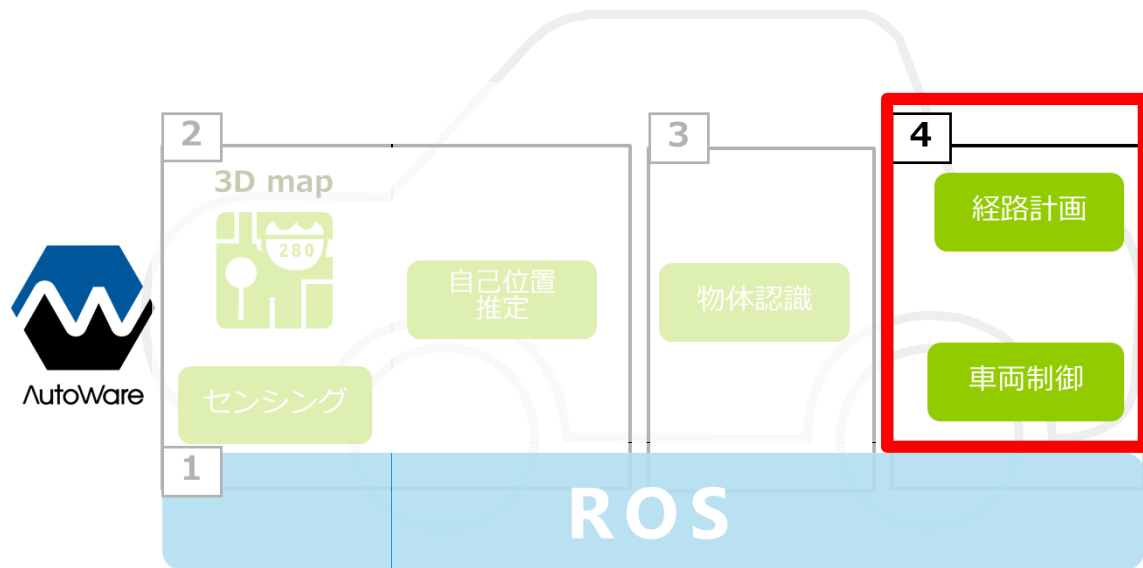
There is also an 'Experiment' checkbox on the far right.

<https://www.youtube.com/watch?v=EjamMJkjBA&feature=youtu.be>

Autowareの信号認識



<https://www.youtube.com/watch?v=KmOdBms9r2w>



経路計画・車両制御

経路計画・軌道生成

スタートからゴールまでの
大まかな経路を計画
経路計画



車両の運動特性を考慮して
すぐ近くまでの**なめらかな経路**を生成
軌道生成



生成された経路を追従するための
速度・角速度を車両へ伝達

ルート探索

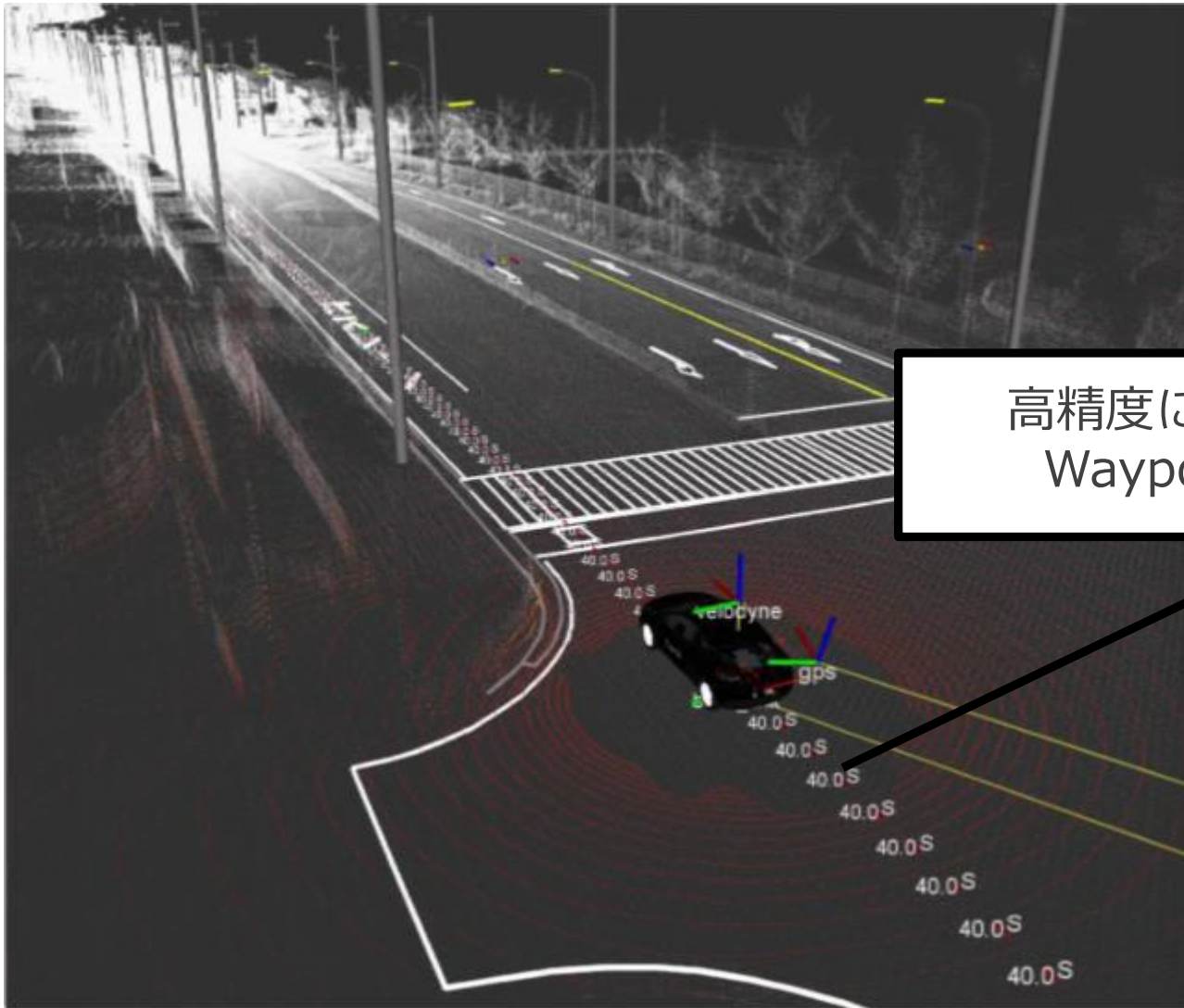
障害物回避

ステアリング制御

アクセルブレーキ制御

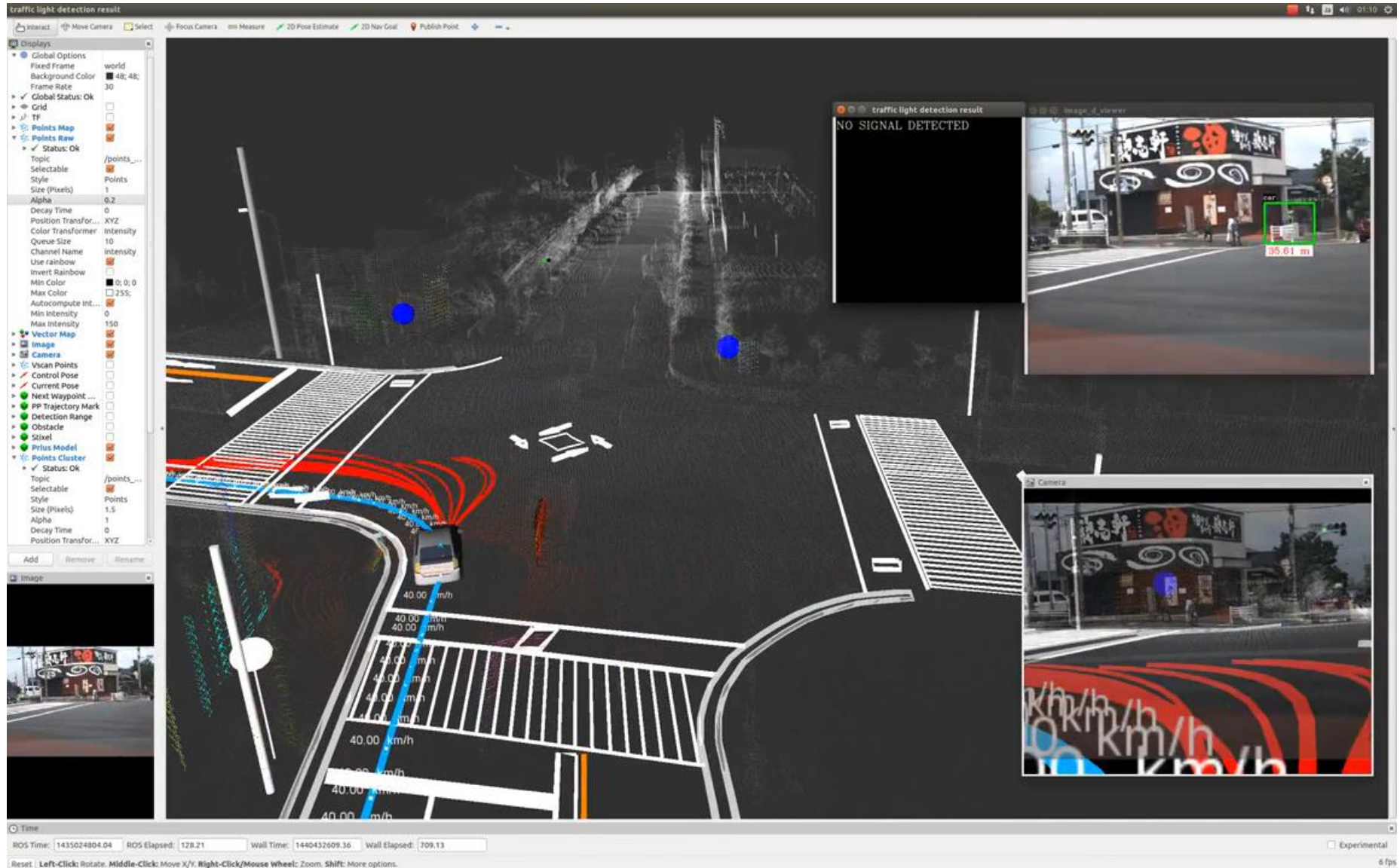
経路生成フロー (3/3)

高精度地図の車線情報を**経路 (Waypoint)** に**変換**する



高精度に道路中心線を通る
Waypointが取得できる

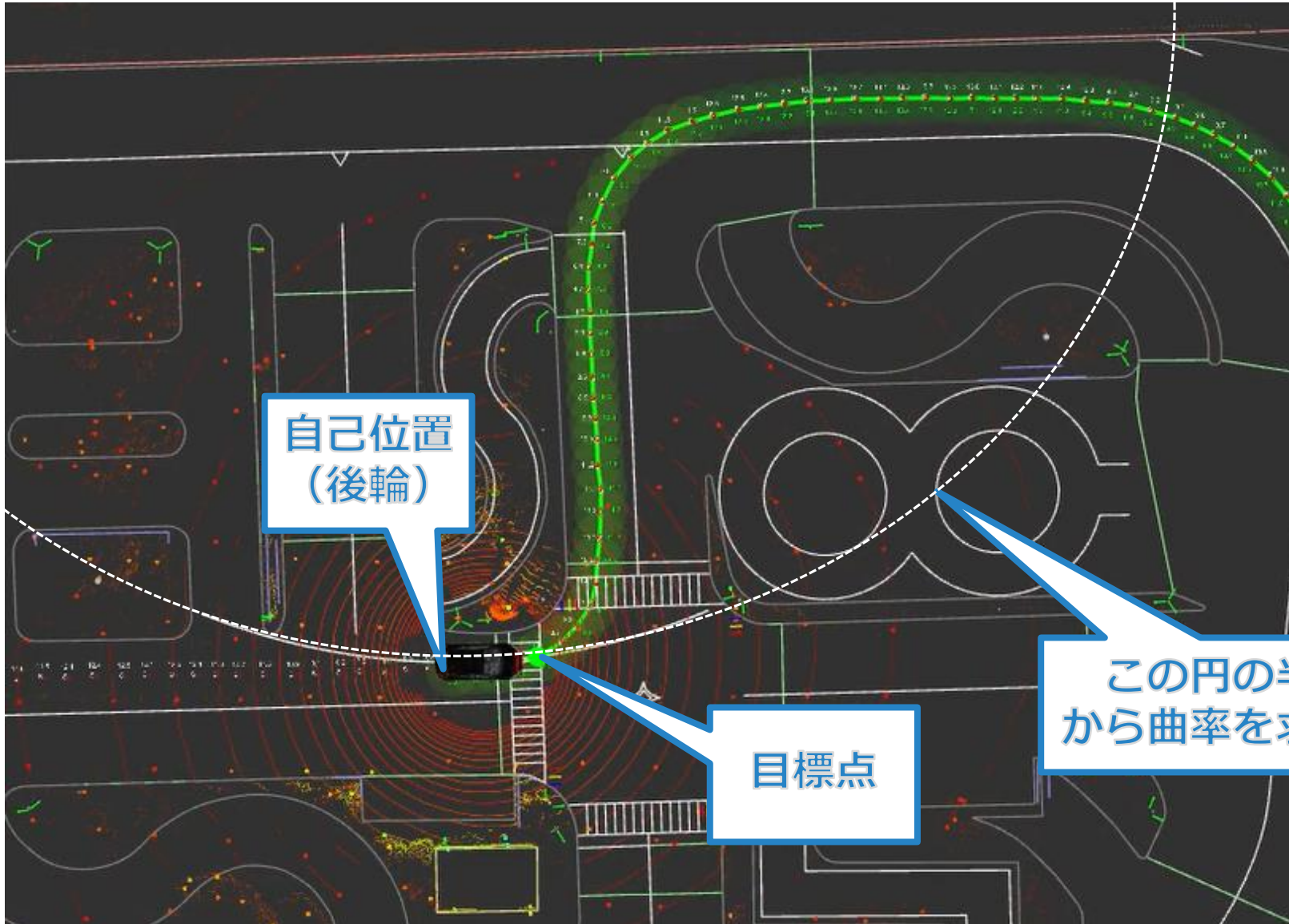
経路生成と経路追従



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.

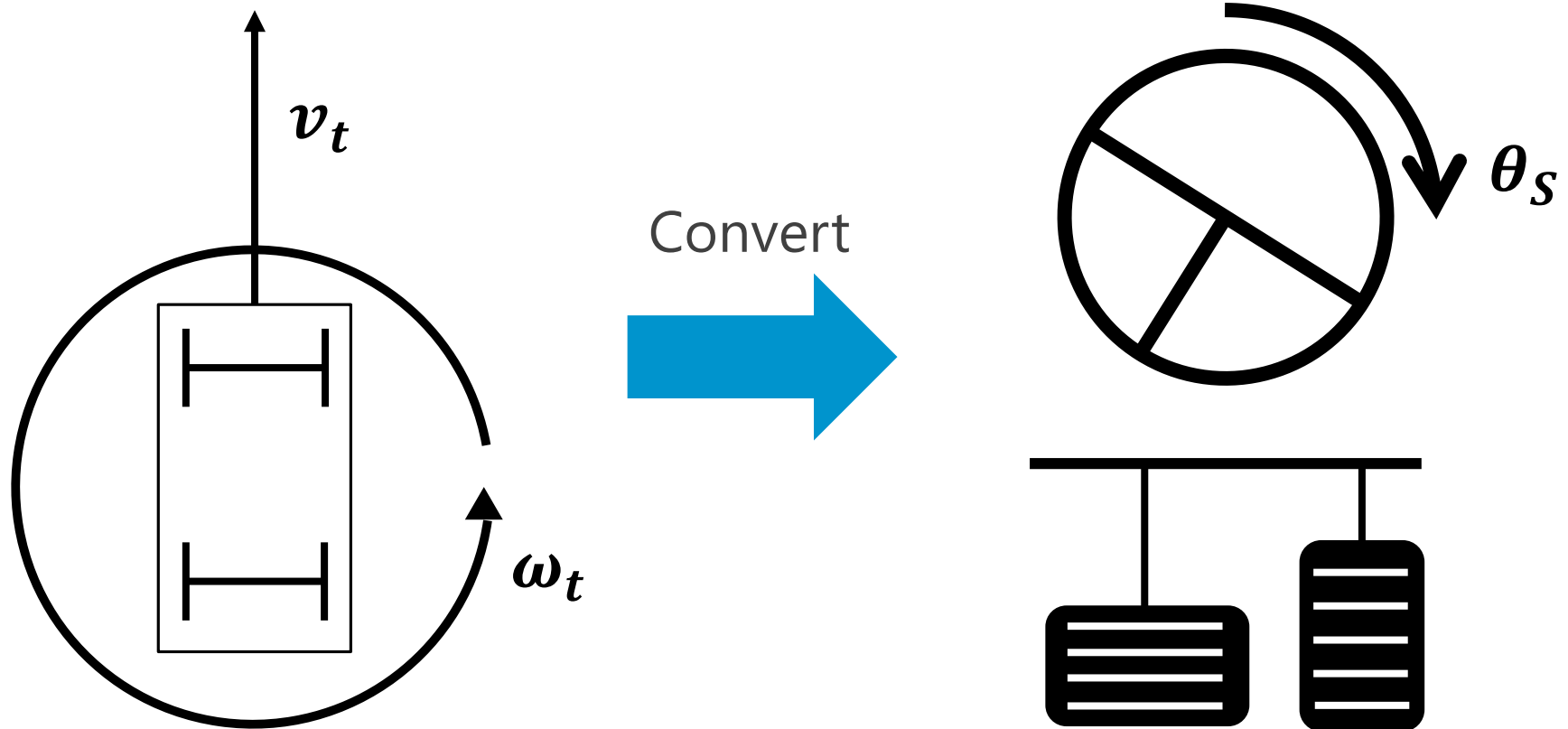
経路上の**目標点**と**自己位置**の情報を基に車両制御信号を計算



車両制御：等速円運動モデル

曲率から ω_t を求める

- 目標速度 v_t : アクセル, ブレーキペダルのストローク量に変換
- 目標角速度 ω_t : 目標ステアリング角度に変換

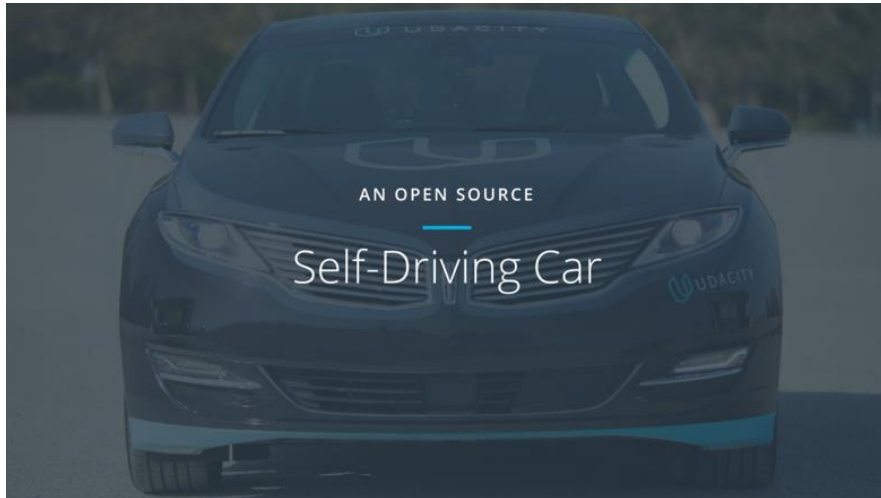


事例紹介

採用事例：Udacity

UdacityはSebastian Thrun、David Stavens、Mike Sokolskyが設立したオンライン教育機関。

UdacityのSelf driving carのベースにAutowareを採用



<https://github.com/udacity/self-driving-car>



Sebastian Thrun

https://www.ted.com/talks/sebastian_thrun_google_s_driverless_car

採用事例：ヤマハ発動機



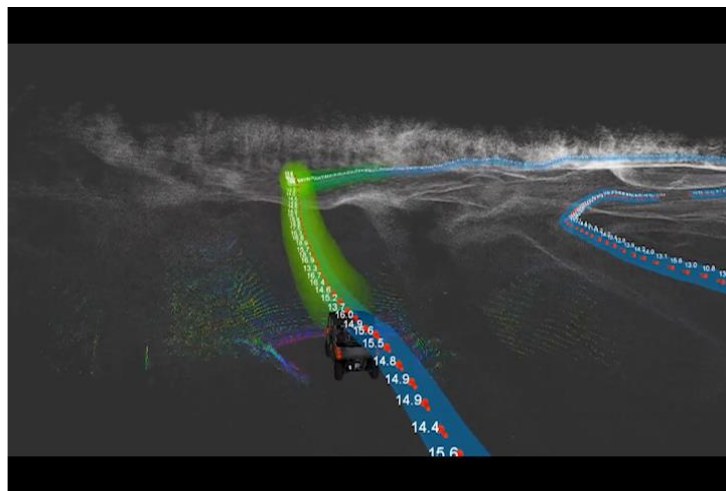
ROS
通信機能



知能化プラットフォーム
[認知] [判断] [操作]



自律車両プラットフォーム
[走る] [止まる] [曲がる]



引用元：2016年12月・第52号 - 技報 | ヤマハ発動機株式会社
自律ビークルの知能化プラットフォーム開発- ROS, Autowareの活用



ヤマハ発動機「Viking VI」をベースとしたバイワイヤ車両

まとめ

●自動運転ソフトウェアプラットフォーム入門 ~要素技術と動作の仕組み~

- ROS
- 自己位置推定技術
- 環境認識技術
- 経路計画・車両制御技術

