

自動運転領域における AI、IoT活用事例

Tier IV 技術本部 関谷英爾

概要

- 会社紹介
- AI活用事例
- IoT活用事例

自己紹介

- Tier4 技術本部
 - 関谷 英爾
- 経験
 - 分析基盤, 機械学習 (レコメンド, 強化学習, 組合せ最適化)
 - Tier IV 2018/03～
 - データ基盤・機械学習基盤
 - FMS開発マネジメント



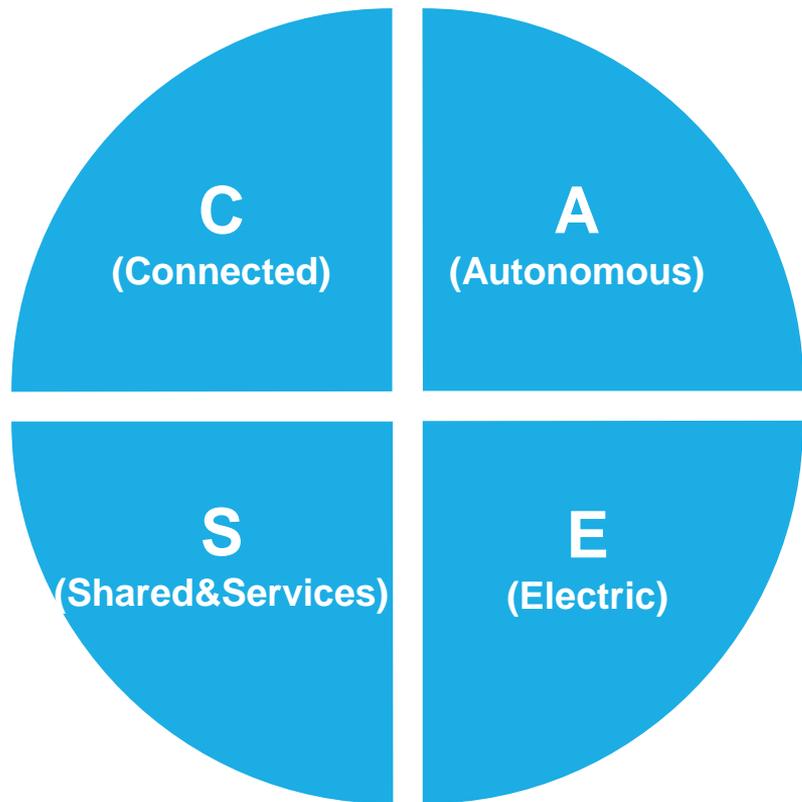
@eratostennis

株式会社ティアフォー (Tier IV, Inc.)

- 自動運転ソフトウェア開発 (Autoware*)
- 自動運転技術を活用した物流サービスの開発
- 近距離低速自動運転サービスの開発

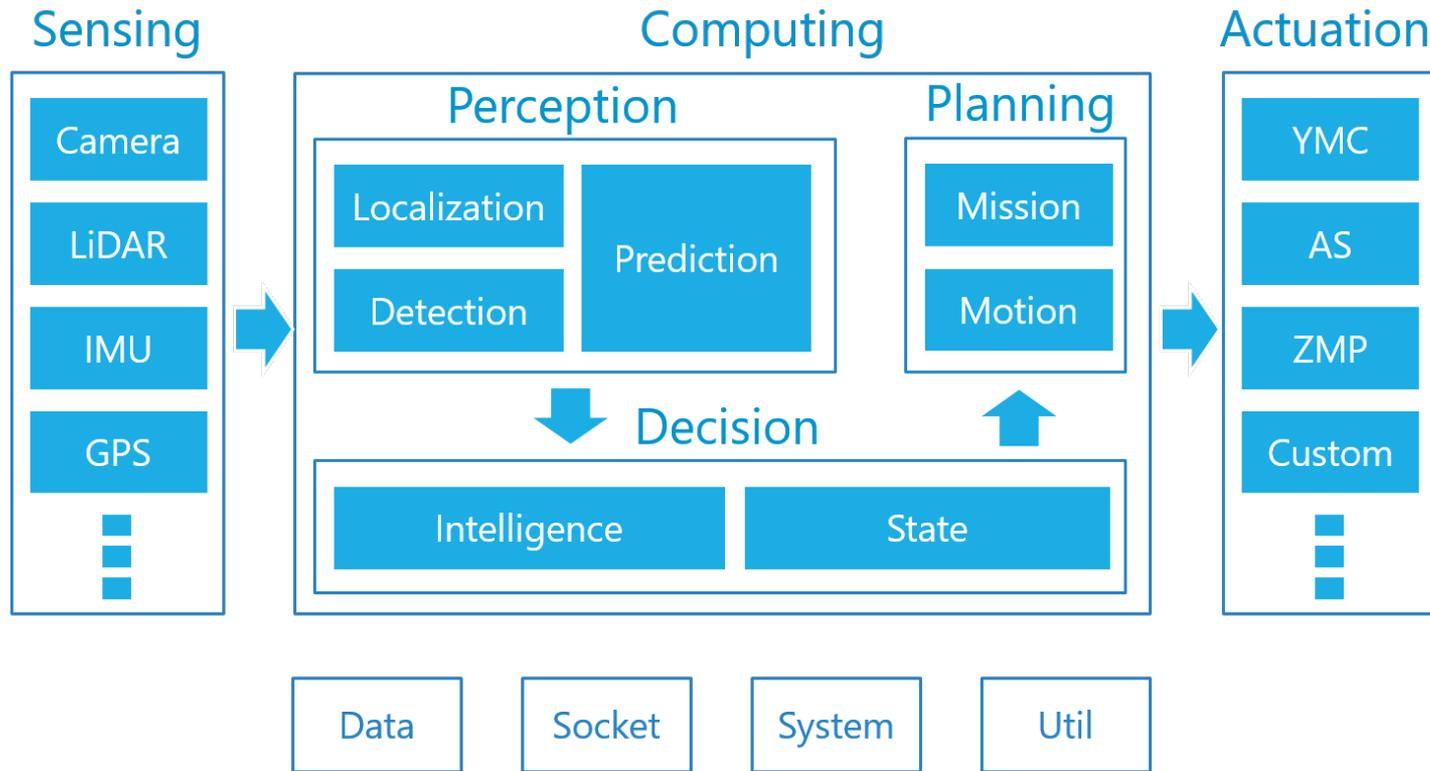


CASE



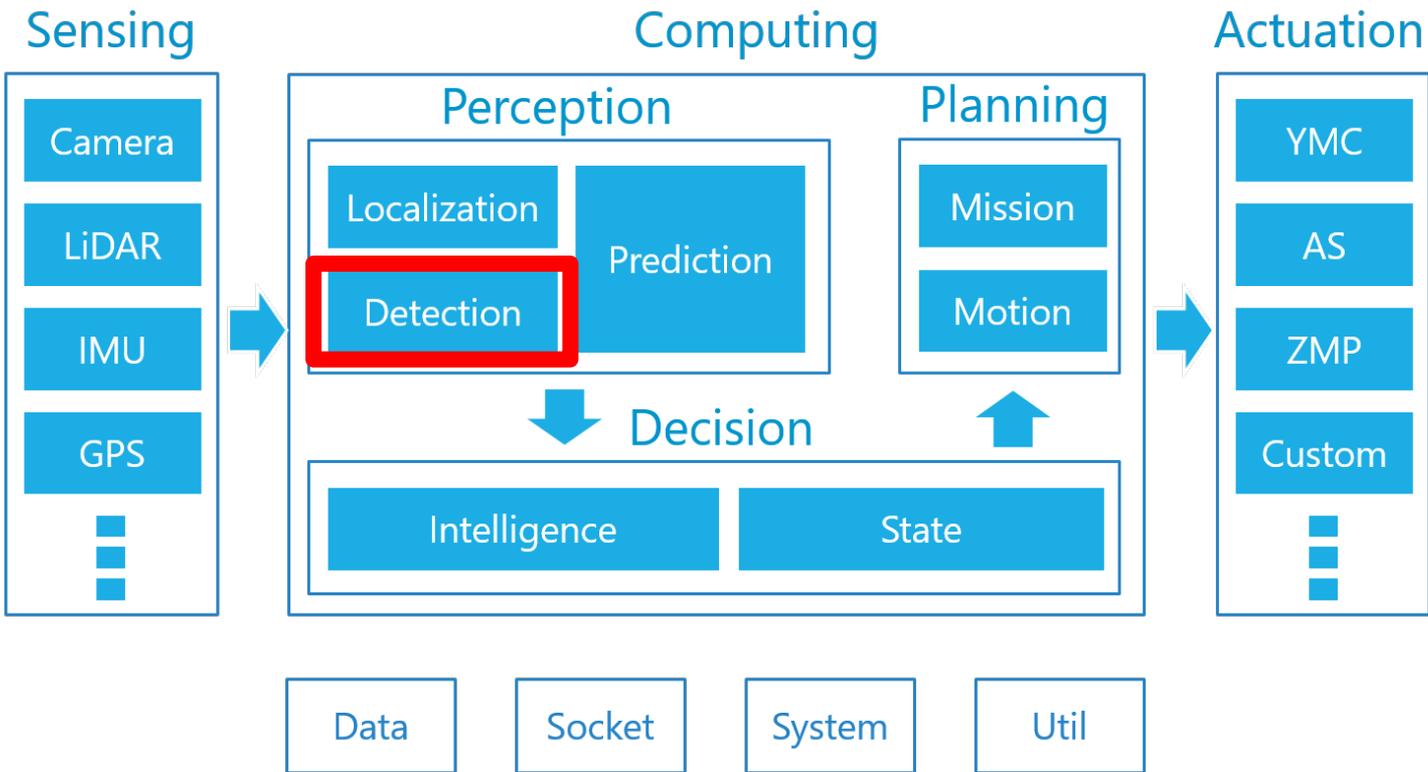
| 自動運転とAI

自動運転OSS Autoware



Retrieved from <https://gitlab.com/autwarefoundation/autoware.ai/autoware/wikis/Overview>

自動運転OSS Autoware



Retrieved from <https://gitlab.com/autwarefoundation/autoware.ai/autoware/wikis/Overview>

自動運転に組み込まれているDeep Learning

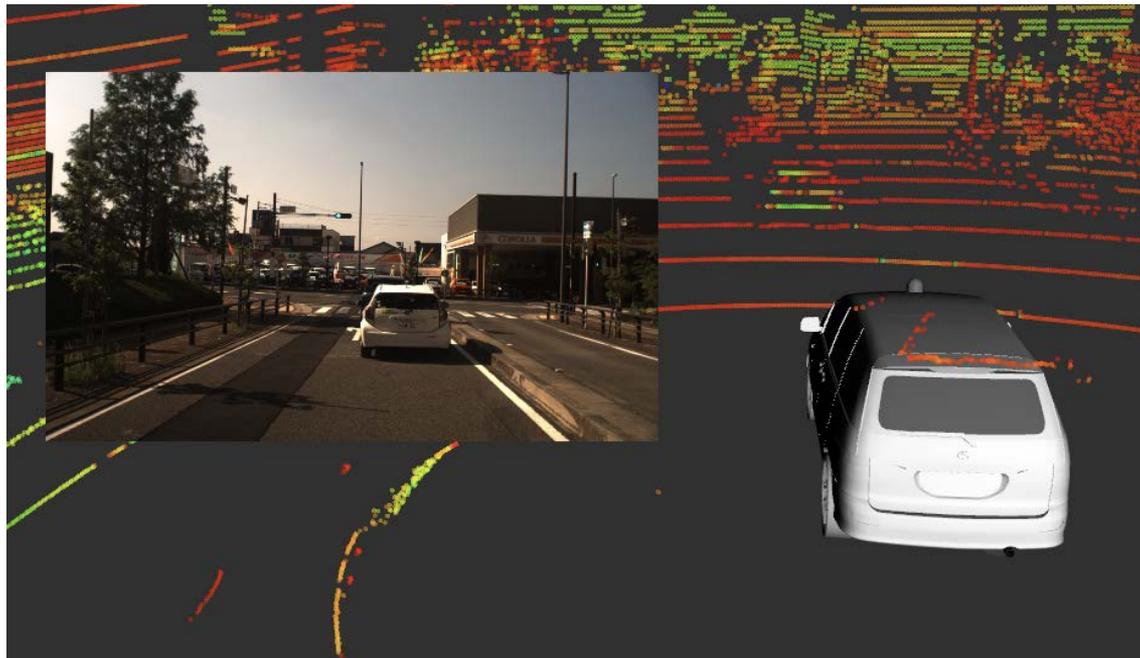
- 検出タスク (2D Bounding Box)
 - Yolov3, etc.
- 検出タスク (3D Bounding Box)
 - VoxelNet, etc.
- 分類タスク (信号色認識)
 - DenseNet
- etc.

自動運転に組み込まれているDeep Learning

- 検出タスク (2D Bounding Box)
 - Yolov3, etc.
- 検出タスク (3D Bounding Box)
 - VoxelNet, etc.
- 分類タスク (信号色認識)
 - DenseNet
- etc.

3Dの物体検出の特徴

- 点群データの特徴
 - 3D, Sparse



3Dの物体検出

クラスタリング手法

- サンプリング
- 地面除去
- Clipping
- Clustering
 - Euclidean Cluster

Deep Learningによる手法

- とりあえず前処理はほぼかけずにDeep Learningに食わせる
 - MV3D
 - VoxelNet
 - PointPillars

3Dの物体検出

クラスタリング手法

- サンプリング
- 地面除去
- Clipping
- Clustering
 - Euclidean Cluster

Deep Learningによる手法

- とりあえず前処理はほぼかけずにDeep Learningに食わせる
 - MV3D
 - VoxelNet
 - PointPillars

Euclidean Cluster



単純に距離の近いものを同一クラスとみなす
Euclidean Clusterで障害物検出と十分な計算量削減が可能
ただし、Semantic情報があれば、さらにトラッキング精度なども上がってくる
ちゃんと物体単位で検出したい

3Dの物体検出

クラスタリング手法

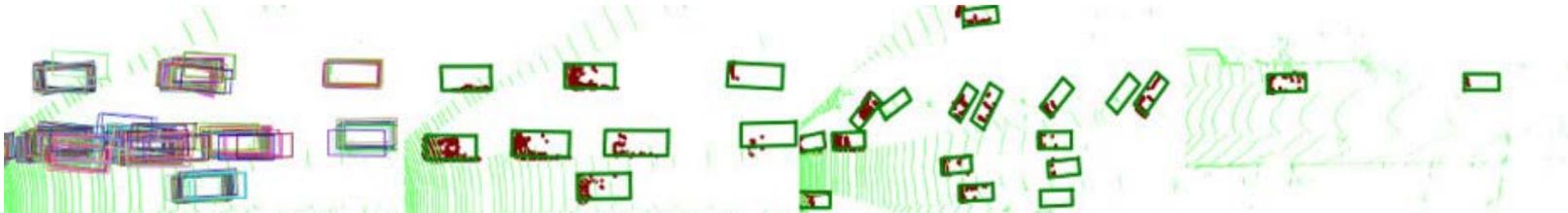
- サンプリング
- 地面除去
- Clipping
- Clustering
 - Euclidean Cluster

Deep Learningによる手法

- とりあえず前処理はほぼかけずにDeep Learningに食わせる
 - MV3D
 - VoxelNet
 - PointPillars

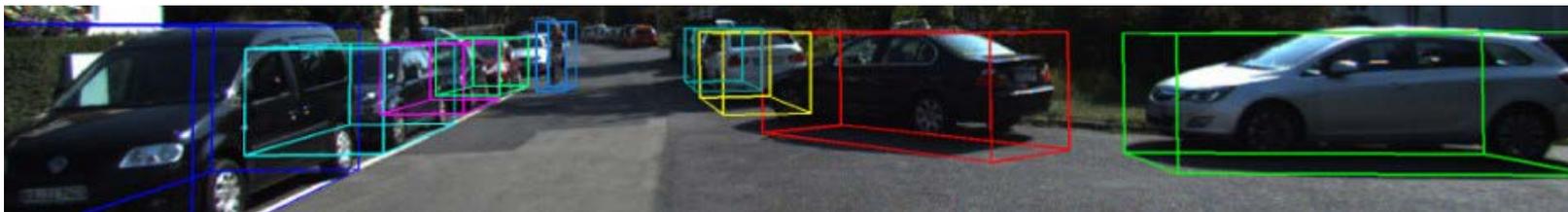
3D物体検出の精度評価

Bird's Eye View (BEV)



The KITTI Vision Benchmark Suite. Retrieved from http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=bev

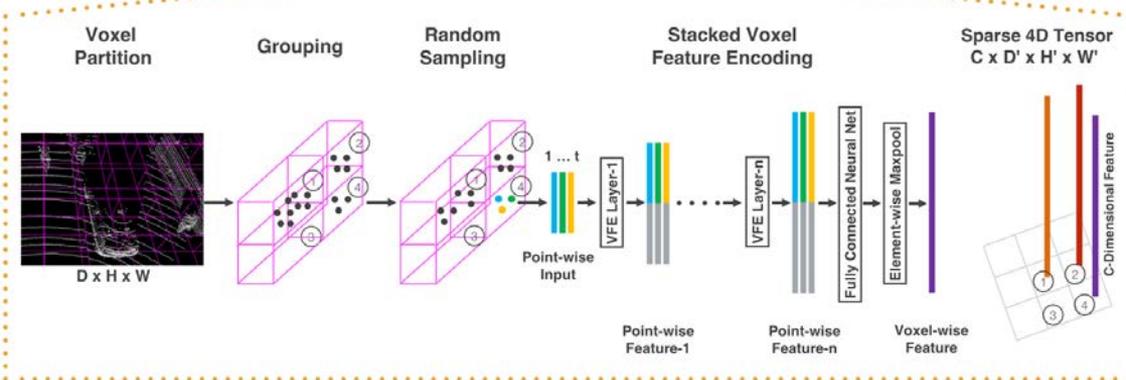
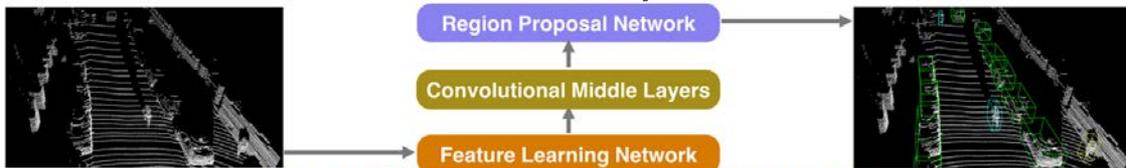
3D Bounding Box



The KITTI Vision Benchmark Suite. Retrieved from http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

● 特徴

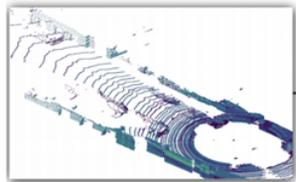
- 3D 物体検出初のEnd-to-end学習
- Convolution Middle Layers(は3D CNNで遅い (4.4Hz)



● 特徴

- 3D CNNを使わずに高速化 (62Hz)
- Single Shot Detectorによる位置の回帰とクラス分類

Point cloud

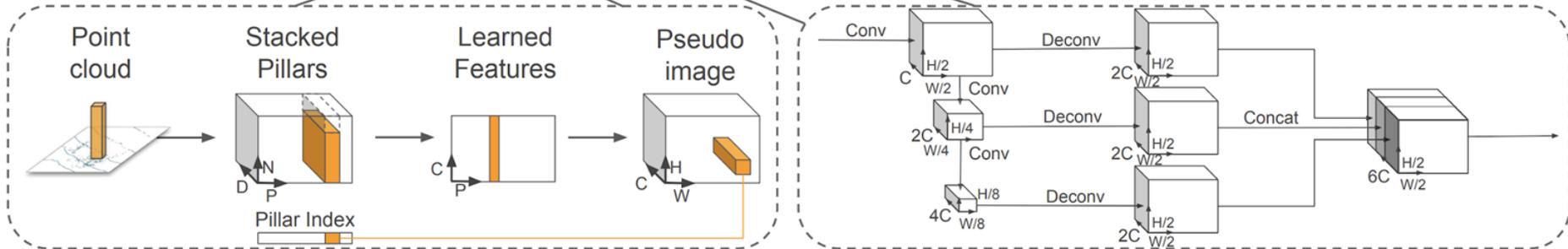
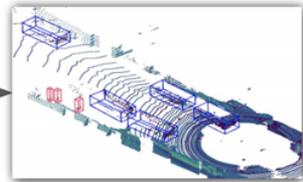


Pillar Feature Net

Backbone (2D CNN)

Detection Head (SSD)

Predictions



Performance

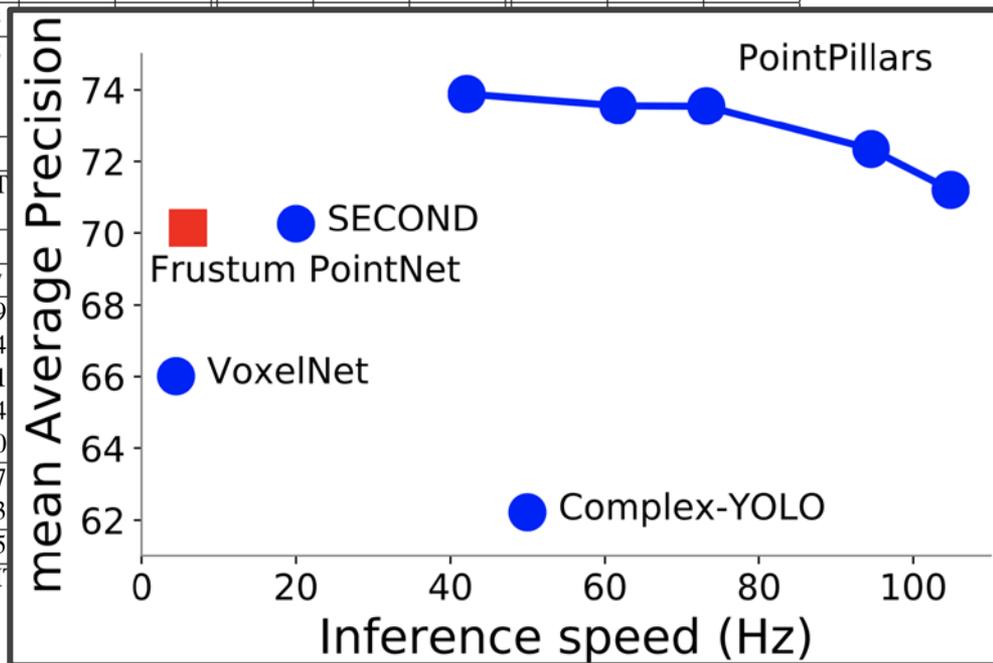
Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, Oscar Beijbom. 2018.
 PointPillars: Fast Encoders for Object Detection from Point Clouds
 arXiv:1812.05784.
 Retrieved from <https://arxiv.org/abs/1812.05784>

Method	Modality	Speed (Hz)	mAP			Pedestrian			Cyclist			
			Mod.	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MV3D [2]	Lidar & Img.	2.8	N/A	86.02	76.90	68.49	N/A	N/A	N/A	N/A	N/A	N/A
Cont-Fuse [15]	Lidar & Img.	16.7	N/A	88.81	85.83	77.33	N/A	N/A	N/A	N/A	N/A	N/A
Roarnet [25]	Lidar & Img.	10	N/A	88.20	79.41	70.02	N/A	N/A	N/A	N/A	N/A	N/A
AVOD-FPN [11]	Lidar & Img.	10	64.11	88.53	83.79	77.90	58.75	51.05	47.54	68.09	57.48	50.77
F-PointNet [21]	Lidar & Img.	5.9	65.39	88.70	84.00	75.33	58.09	50.22	47.20	75.38	61.96	54.68
HDNET [29]	Lidar & Map	20	N/A	89.14								
PIXOR++ [29]	Lidar	35	N/A	89.38								
VoxelNet [31]	Lidar	4.4	58.25	89.35								
SECOND [28]	Lidar	20	60.56	88.07								
PointPillars	Lidar	62	66.19	88.35								

Table 1. Results on the KITTI

Method	Modality	Speed (Hz)	mAP	
			Mod.	Easy
MV3D [2]	Lidar & Img.	2.8	N/A	71.09
Cont-Fuse [15]	Lidar & Img.	16.7	N/A	82.54
Roarnet [25]	Lidar & Img.	10	N/A	83.71
AVOD-FPN [11]	Lidar & Img.	10	55.62	81.94
F-PointNet [21]	Lidar & Img.	5.9	57.35	81.20
VoxelNet [31]	Lidar	4.4	49.05	77.47
SECOND [28]	Lidar	20	56.69	83.13
PointPillars	Lidar	62	59.20	79.05

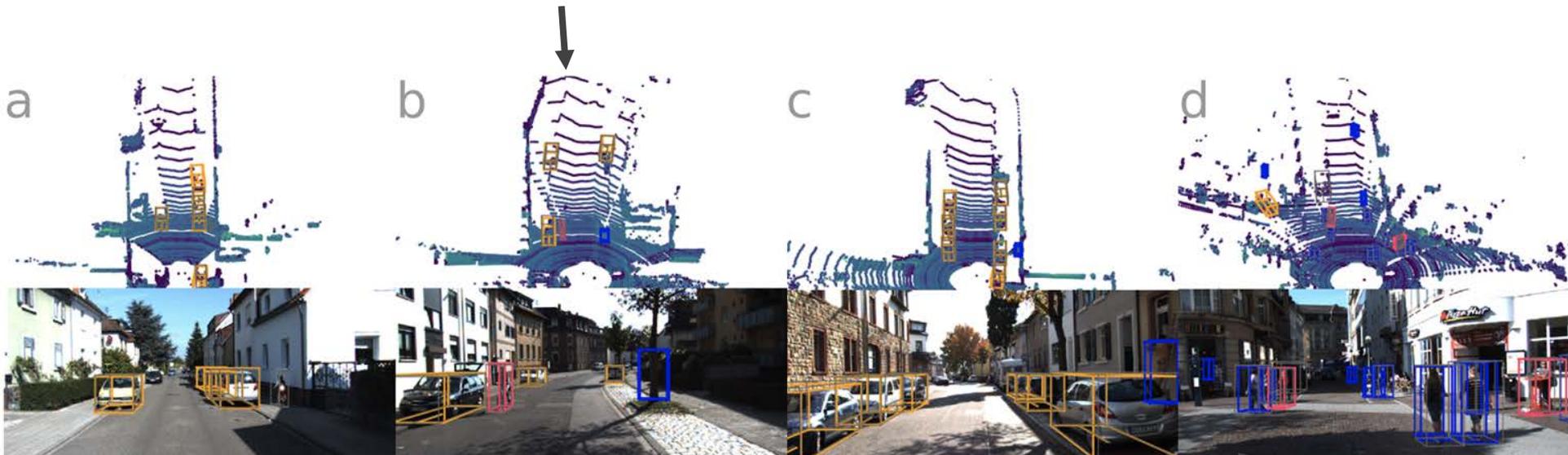
Table 2. Results on the KITTI



Failure Case

Alex H. Lang, Sourabh Vora, Holger Caesar, Luning Zhou, Jiong Yang, Oscar Beijbom. 2018.
PointPillars: Fast Encoders for Object Detection from Point Clouds
arXiv:1812.05784.
Retrieved from <https://arxiv.org/abs/1812.05784>

街路樹を歩行者と分類



歩行者とサイクリストの誤分類

3D物体検出との付き合い方

- 精度が上がって来たが、まだまだ3D物体検出単体で使うにはいまいち
- 使い方
 - 画像とFusionする
 - Deep Learningで行う必要はない
 - 地図のレーン情報など他に持っているデータも利用する
- Trackingの精度向上
 - 単純な2Dだけで推測するより距離などは正確
 - 物体の運動モデルがだいたい分かっていたら役に立つ

Edgeでの推論について

- エッジのリソース制約
 - メモリ、電力
- モデルの軽量化
- リアルタイム性
- テスト手法

Edgeでの推論について

- エッジのリソース制約
 - メモリ、電力
- モデルの軽量化
- リアルタイム性
- テスト手法

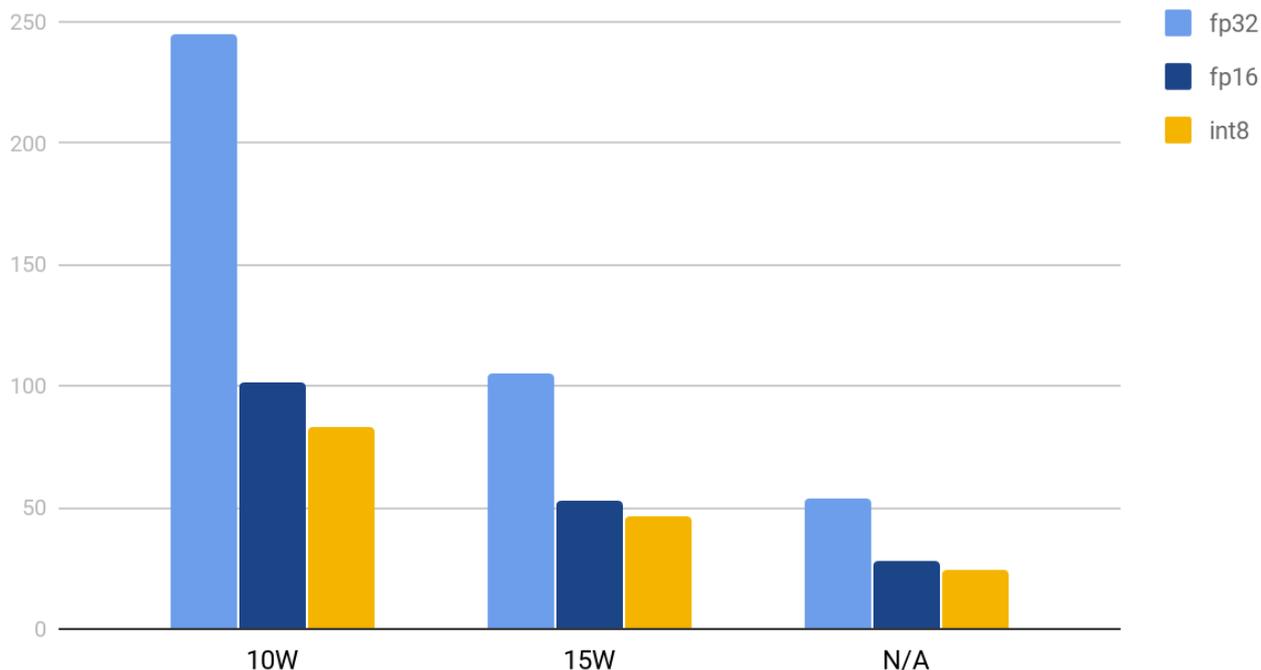
NVIDIA Jetson AGX, Jetson Nano

- NVIDIAが発売した組み込みAI向けデバイス

	AGX	Nano
サイズ	100×87mm	70×45mm
CPU	8-Core ARM v8.2 64-Bit CPU, 8MB L2 + 4MB L3	クアッドコア ARM Cortex- A57 MP Core
GPU	512-Core Volta GPU with 64 Tensor Cores	NVIDIA Maxwell
メモリ	16GB	4GB

Jetson AGX (Tensor RT, yolov3)

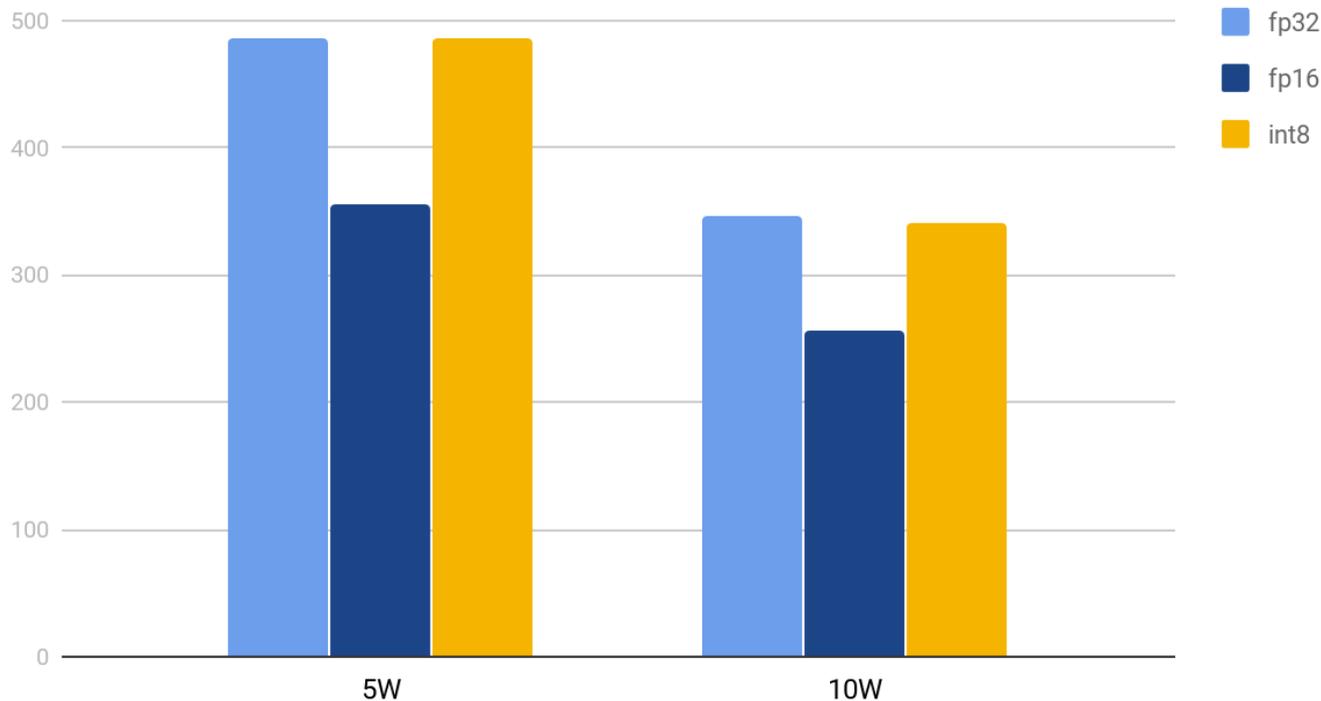
Execution Time [ms] - Power Budget



消費電力をあげればあげるほど速度改善
15Wで20Hzとかなり高速で動作

Jetson Nano (Tensor RT, yolov3)

Execution Time [ms] - Power Budget



Maxwell世代でint8の計算で実行時間が悪化
工夫をしていかなければ10Hz以上の速度がでない

| 運行管理システムとIoT

IoT活用事例

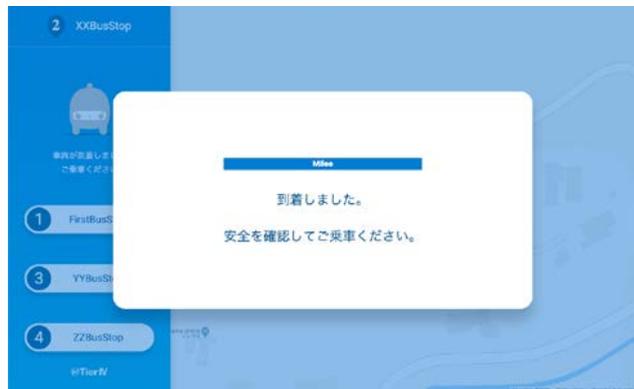
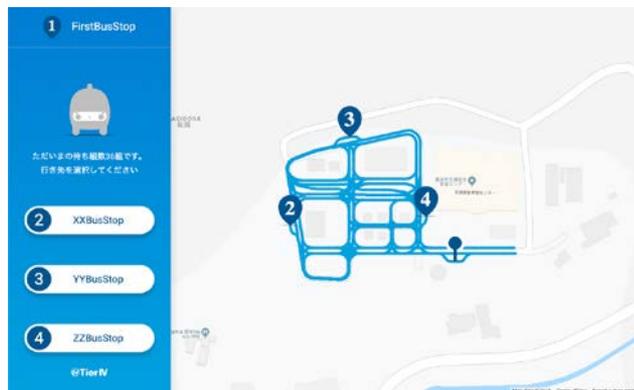
- 運行管理・配車サービス
- 設備連携 (V2I)
- フィールドテスト

Autoware FMS

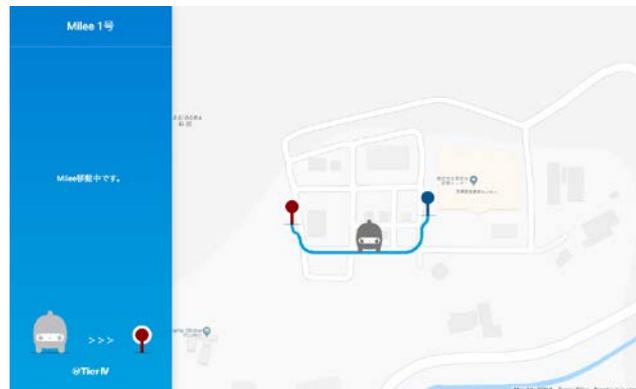
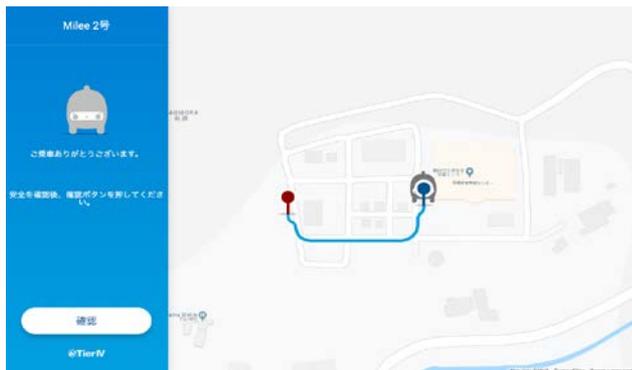
- Autowareで動く自動運転車両の運行管理システム
- 機能
 - モニタリング
 - スケジューリング
 - 経路計画
 - 運行過去実績管理



配車アプリ



車載アプリ



コンソール

- 位置や車両状態などのモニタリング
- エラー・オフライン時の通知
- スケジュールの予定・走行実績確認

The screenshot displays the 'メインコンソール' (Main Console) interface for FMS. The top navigation bar includes 'サービス検索' (Service Search) and 'ログアウト' (Logout). The left sidebar contains navigation options: 'メインコンソール', '車両詳細' (Vehicle Details), '車両管理' (Vehicle Management), and '日程' (Schedule). The main area is divided into a map and a data table.

The map shows a vehicle labeled 'tsc02' with a green dashed outline, indicating a specific route or area. The table below the map provides a summary of vehicle status:

車両名	状態	異常有無	車両詳細へ
tsc02	発進指示待ち	異常あり	車両詳細へ

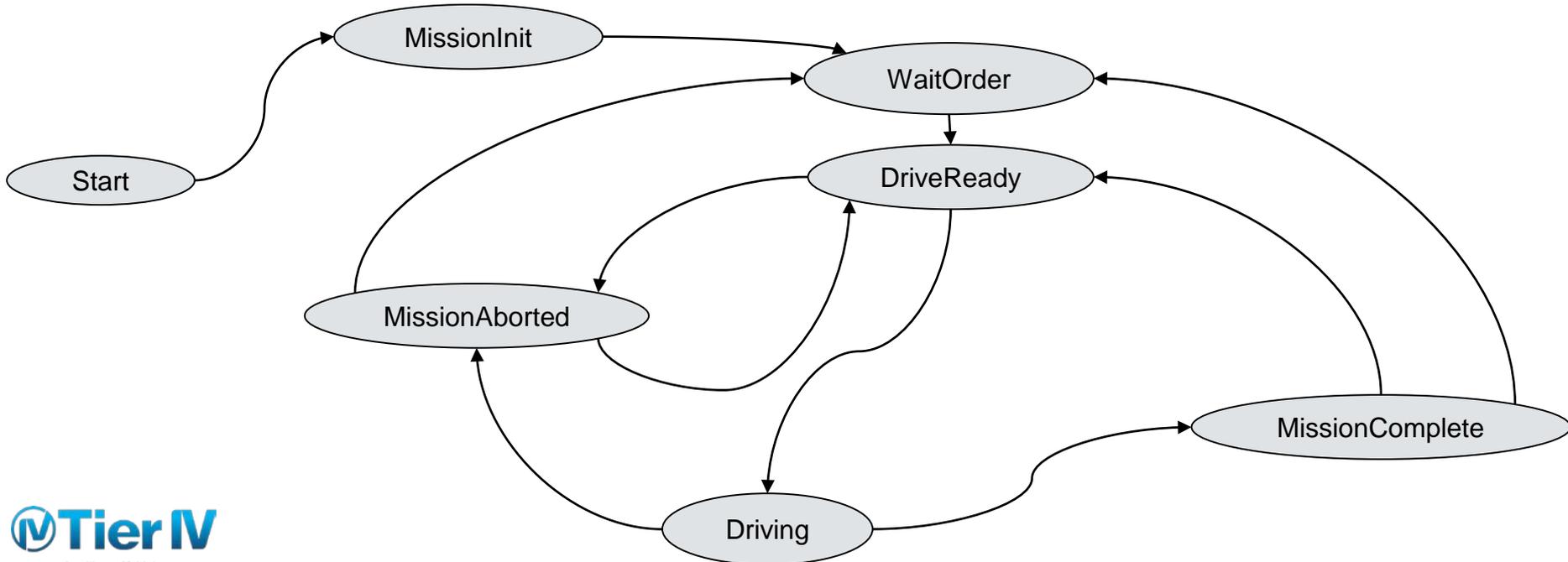
At the top of the table, there are summary counts: 1 車両 (1 Vehicle), 0 エラー車両 (0 Error Vehicle), and 1 通信遅延車両 (1 Communication Delay Vehicle). The table also includes a search bar for '車両検索...' and a '停止可能地点表示' (Show possible stop locations) button.

様々なデータの通信

- 車両 => FMS
 - 車両状態, 速度, バッテリー, 位置情報など
- 車両 <= FMS
 - ステートコマンド, 経路, 車両操作コマンド

スケジュール実行時の処理の流れ

- FMSに登録されている予定をマスターとし、Autowareのステートを変化させて行く



Autoware Drive (遠隔運転の取り組み)

Autoware Drive <<

Dashboard
Map
3DMap
Data
Camera
Monitor
Setting

Vehicle Info
Vehicle ID: 1
Speed: 0.83 km/h
Target Speed: 0 km/h
State:

Steering Position
Controller Steering Position
Controller Accel
Controller Brake

NORMAL
PROGRAM
AUTO
DEMO

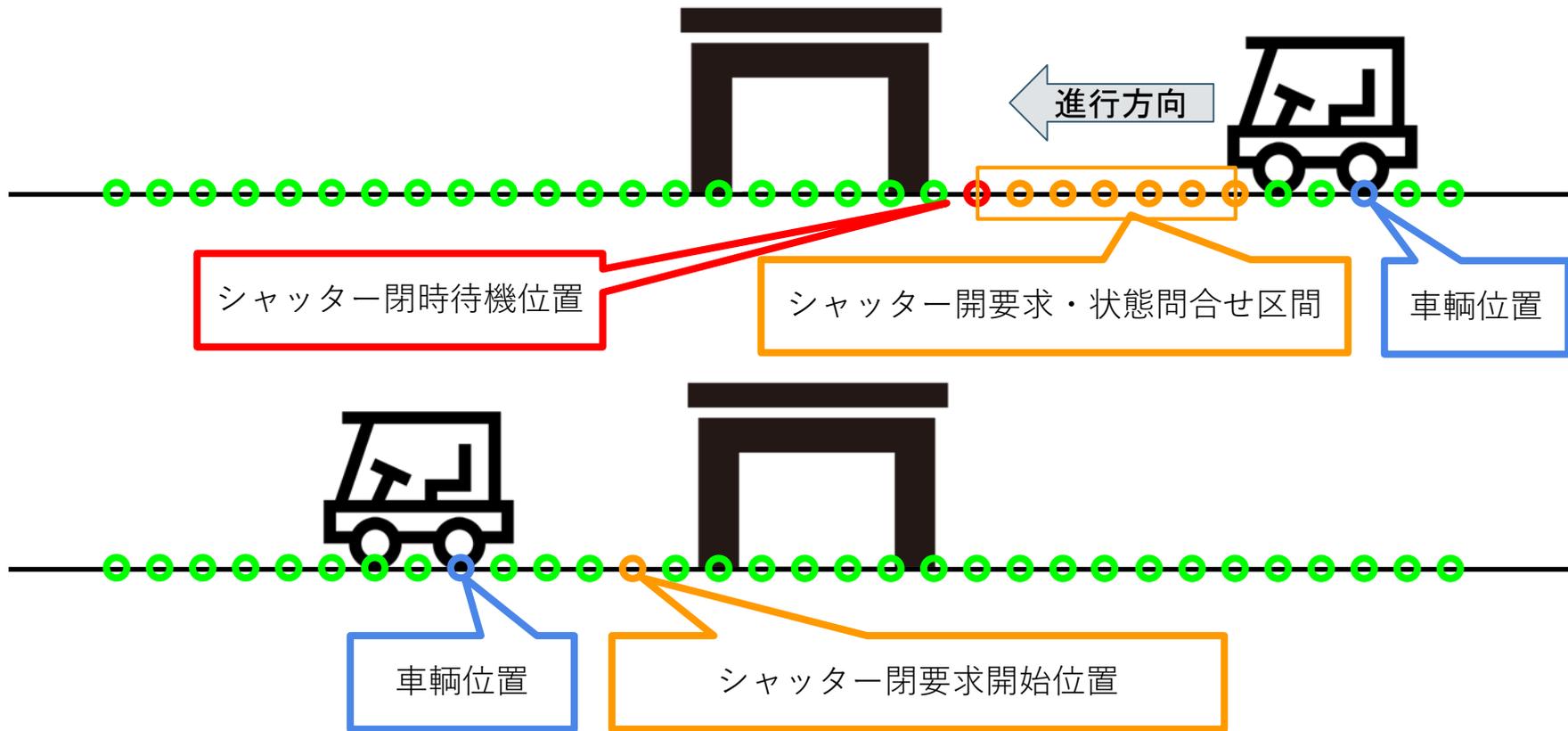
Autoware Drive (遠隔運転の取り組み)

- 安全・高速な通信が必要
- 様々な検討
 - AWS IoT
 - MQTT
 - QUIC

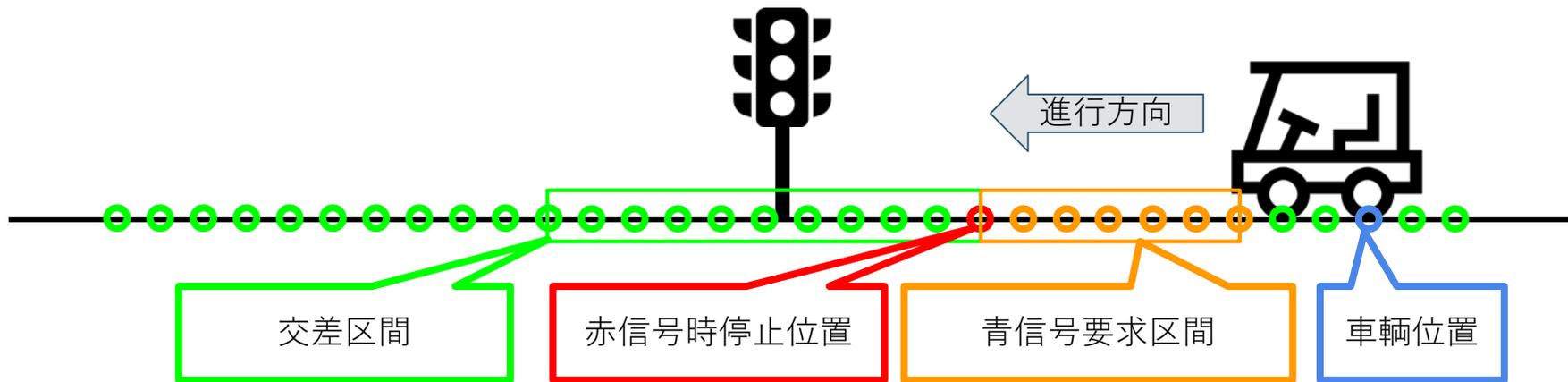
設備連携

- 完全自動運転になると人手を介入させない設計が必要
- 例
 - 交差点での複数車両調停
 - シャッター連携
 - エレベータ連携

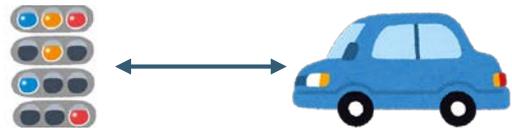
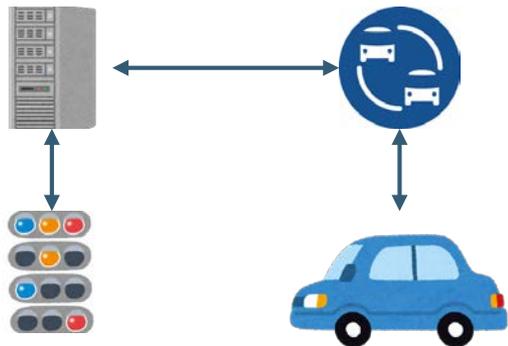
シャッター連携



信号機連携



システム連携



フィールドテスト準備

- 事前の品質向上
 - プログラムのテスト
 - シミュレータによるテスト

フィールドテスト

- フィールドでしか試せないことがある
 - 実車両・走行環境によって様々な調整が必要
 - 制御動作の確認, 通信環境の確認
- 決められた手順で調整できることが重要

Over the Air

- プログラムの更新
- ログレベルの変更
 - テスト時に遠隔から手厚いサポートをする場合に設定を変更
- 映像配信ビットレート
- 地図データ更新

ドラレコデータのリアルタイムアップロード

- Autowareが利用しているROSではログデータが大きくなりやすい
 - Autowareに異常があった時間に絞ってログを管理
 - 機械学習を用いた未知のデータの検出を行う研究開発

今後の取り組み

- AIの取り組み
 - モデル評価・品質・安全性の取り組み
 - デバイスエッジ最適化による省エネ・高速化
- IoTの取り組み
 - デバイスエッジでのデータ選択・圧縮
 - セキュアな環境を素早く構築