課題解決に向けた IoTエッジデバイス化



2019年8月9日

株式会社コア エンベデッドソリューションカンパニー 八尋 崇泰



目次



- 1. 会社紹介
- 2. 基盤システムについて
- 3. システムの要素技術
- 4. システムの課題
- 5. インテリジェント・単眼カメラ
- 6. 今後について



1. 会社紹介

1. 会社紹介



■株式会社コアのご紹介

社名 株式会社コア

東京証券取引所 市場第一部[2359]

設立 1969年12月(資本金4.4億円)

売上 204.6億円

従業員数 1,540名(連結) 1,100名(単体)

関連企業 関連会社16社

"コア"の由来

「情報サービス産業の核(CORE)になる」 「特定分野で業界トップを目指す」

事業セグメント

- ・SIビジネス
- ・ソリューションビジネス







■ 基盤システムの課題を解決してIoTエッジデバイス化

以前、交通インフラ情報収集のためのシステムを構築したが、様々な課題があった。 これら課題を解決することを目的に、不要な機能をそぎ落としつつ小型化、省電力化など の改善を行い、IoTエッジデバイスとして「インテリジェント・単眼カメラ」を製造した。

元になった基盤システムの課題をどのように解決したか紹介する。



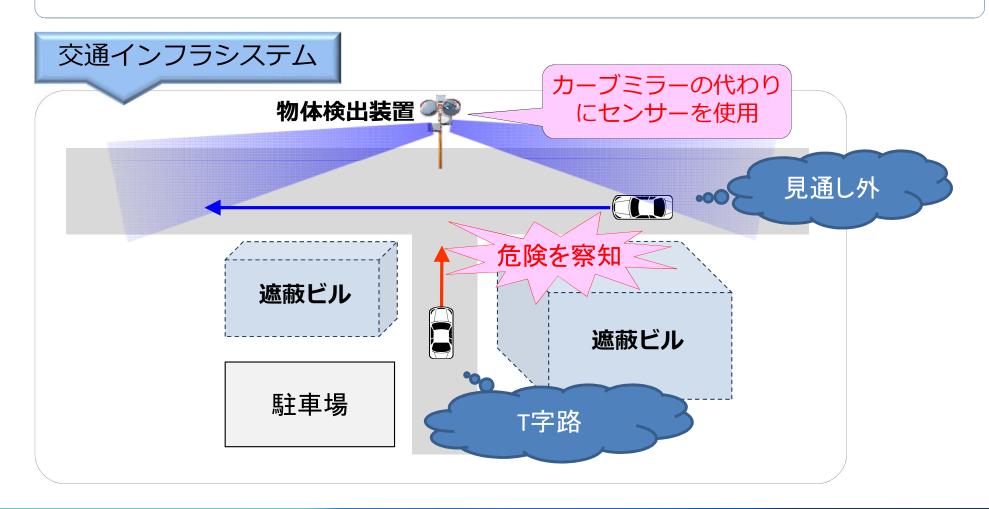


地図マッピング



■元になった基盤システムについて

元になった基盤システムは、ステレオカメラで検出した車や人を地図にマッピングし、自車で道路状況を把握することで、将来の自動運転における交通インフラシステムに活用。

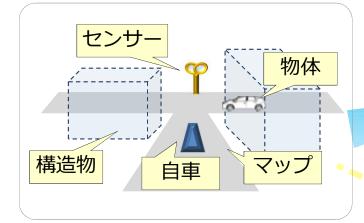




■元になった基盤システムについて

元になった基盤システムは以下のように構成されている。

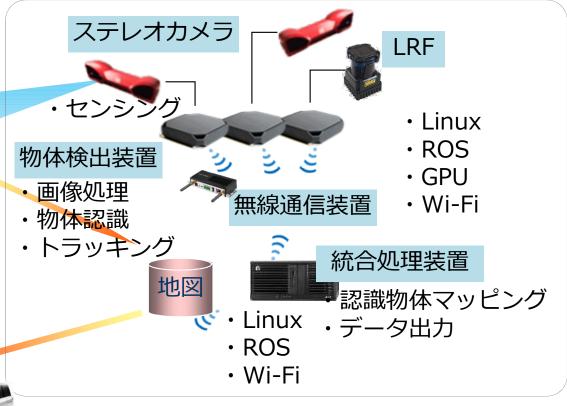
■ 検証環境



表示装置



物体検出装置

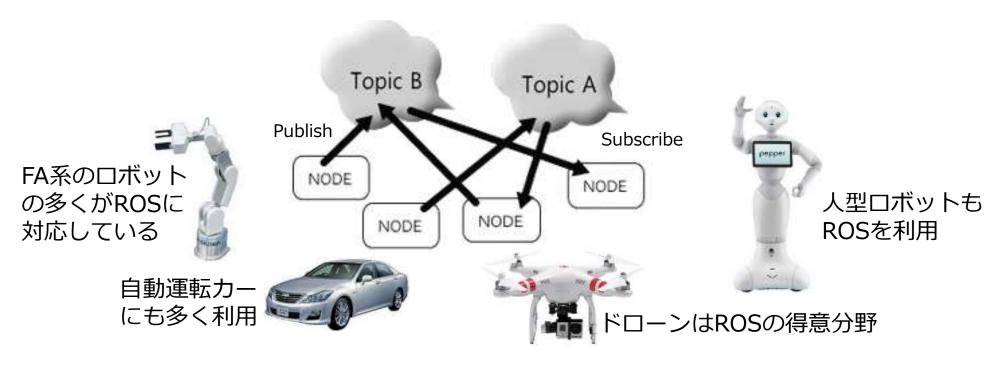






ROS (Robot Operation System)

ROSはlinuxベースのロボット向け通信プラットフォームで、オープンソースである。ロボット開発で必要な様々な分散通信、アルゴリズム、画像処理ライブラリ、座標変換ライブラリ、視覚シュミレータ、再生可能なログなどがある。



ネットワークで繋がっていればノードは複数の プロセッサ上に分散して配置することが出来る



ROS (Robot Operation System)

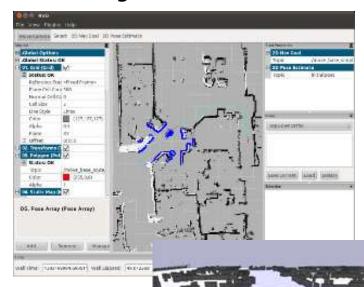
ROSにはロボット開発に必要な多種多様なライブラリが含まれている。 そのため、簡単なロボット開発ならソースコードを組む必要がない。

※サンプルやチュートリアルにはほとんどコードがない

代表的なライブラリ

- ・メッセージ内データの視覚化ツール rviz, rqt_plot
- ・メッセージ記録・再生ツール rqt_bag
- ・ロボット用モデリング言語 xacro, urdf
- ・物理シミュレータ gazebo
- ・画像処理ライブラリ OpenCV
- ・ナビゲーションライブラリ
- ・座標変換ライブラリ tf

Rvizなど視覚化用の物理シュミレータが充実していて、記録したRosBagを再生できる



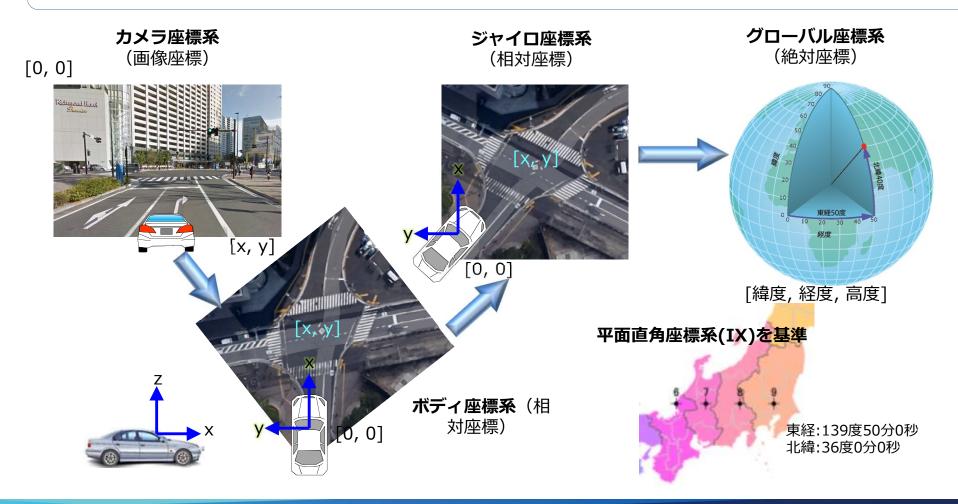
シュミレータやNavigationライ ブラリを使えばリアルタイムに 画面から操作が可能



12

■ TF (ROS座標変換ライブラリ)

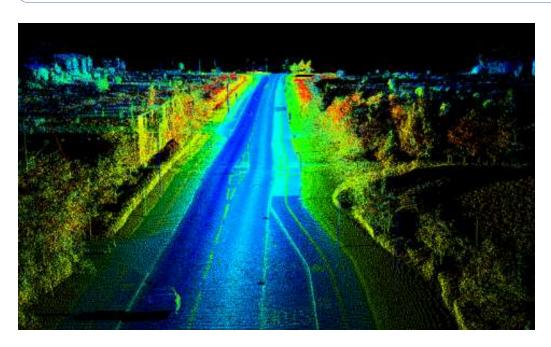
センサ情報は様々な座標系で構成され、出力される周期やタイミングがバラバラである。 ROSではtfを利用することで座標系を意識することなく開発が可能になる。





PCL (Point Cloud Library)

ポイントクラウド(点群)を処理するためのライブラリ。主に2次元/3次元の点群処理を行う。元々はROSのロボットビジョンに含まれていたプロジェクトが独立したもの。



PCLではKinect用のインタフェースや 3次元モデリングに利用可能な機能が 開発されてきた。

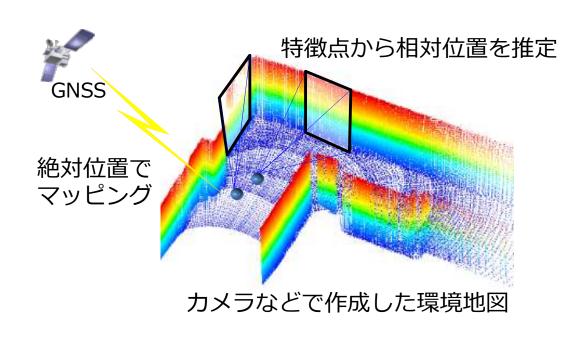


フィルタリング、特徴推定、表面再構成、イメージレジストレーション、モデルフィッティング、セグメンテーションなどのアルゴリズムを扱える。

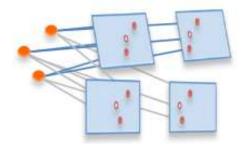


SLAM (Simultaneous Localization And Mapping)

カメラやセンサーを用いて自己位置推定と地図作成を同時に行う技術。GNSSでは電波状況によって位置が取れなかったり誤差が生じたりするため、SLAMを使うことが多い。



Visual Odometry



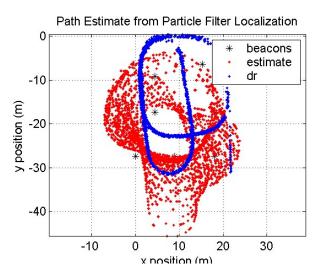
複数視点の画像から類似している特徴点を見つけて相対位置を推定する技術。



■ Tracking (追跡)

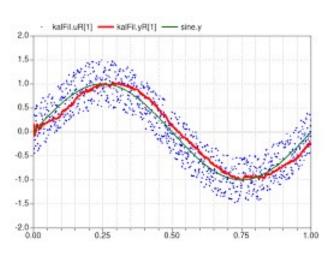
車両や人物を認識するだけでは動きを予測できないため、Trackingが必要である。 追跡用アルゴリズムはParticle FilterやKalman Filterなど多くの手法がある。

PF(Particle Filter)



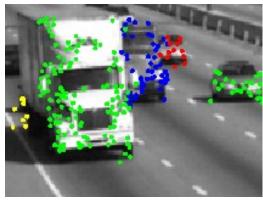
物体を粒子として捉え、 動きをシミュレーション することで追跡(予測)す る。

KF(Kalman Filter)



誤差のある観測から時々 刻々と時間変化する物体 の位置と速度を推定する。

KLT Tracker (Kanade Lucas Tomasi)



前の画像のある点が次の画像のどこへ動いたのかを特徴点のオプティカルフローを計算することでトラッキングする。

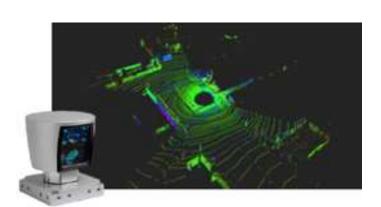


■ NDT (Normal Distributions Transform)

3次元点群地図と3次元LIDARデータを入力として、アルゴリズムをベースとしたスキャンマッチングを行うことで、自車位置を10cm程度の誤差で推定可能。



環境地図データ(ポイントクラウド)

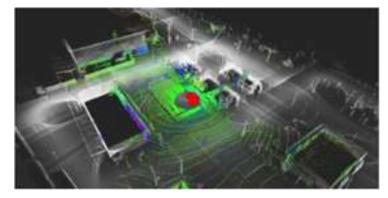


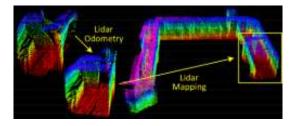
スキャンデータ(LIDAR/LRF)

スキャンマッチング

高精度点群地図データとLIDARから得られ た点群データを用いて位置を推定





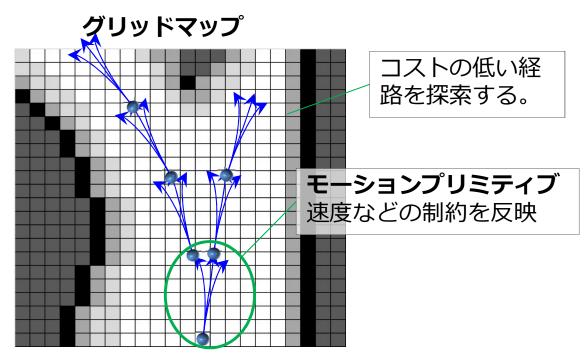


※NDTはLinux標準のPCLに標準搭載



SBPL (Search-based Planning Library)

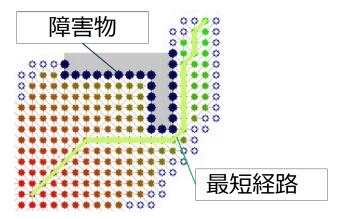
A*(A-star)などの探索アルゴリズムを利用してグリッドマップ上のコストの低い(障害物のない)経路を探索する。また、ロボットの運動モデルに則した経路を計算する。



DWA(Dynamic Window Approach)

DWAは、現在の位置情報と速度情報、障害物情報、そしてゴールの位置情報から、ロボットの運動モデルに則した制御入力を計算する。

A*(A-star)探索アルゴリズム



スタートノードから、ある ノード n を通って、ゴール ノードまでたどり着くとき の最短経路を求める。



■Autoware(名古屋大学、長崎大学、産総研による共同成果)

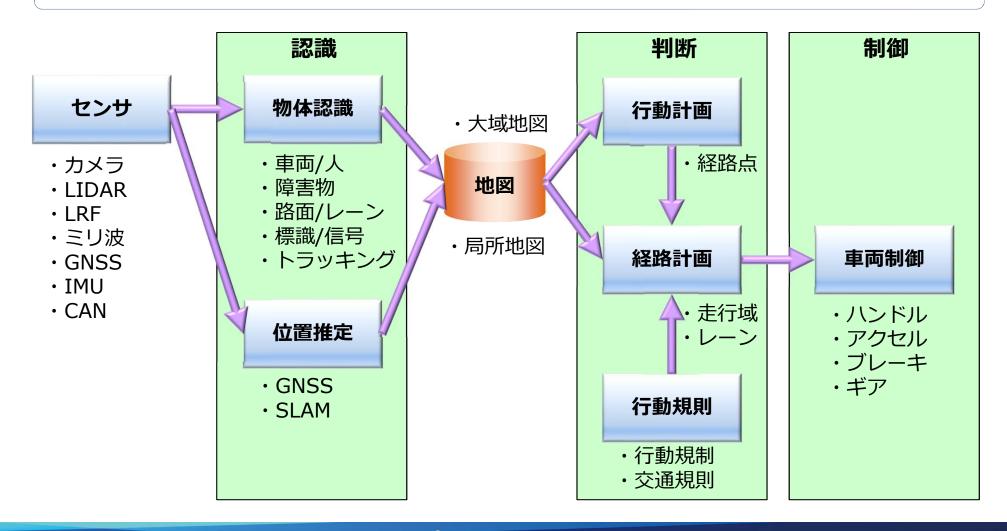
自律走行に必要な、物体認識、位置推定、走行計画、車両制御の基本機能を持ち、 LIDAR/LRF、カメラ、CAN、GNSSなどの入力から自動で経路を計算し自律走行する。





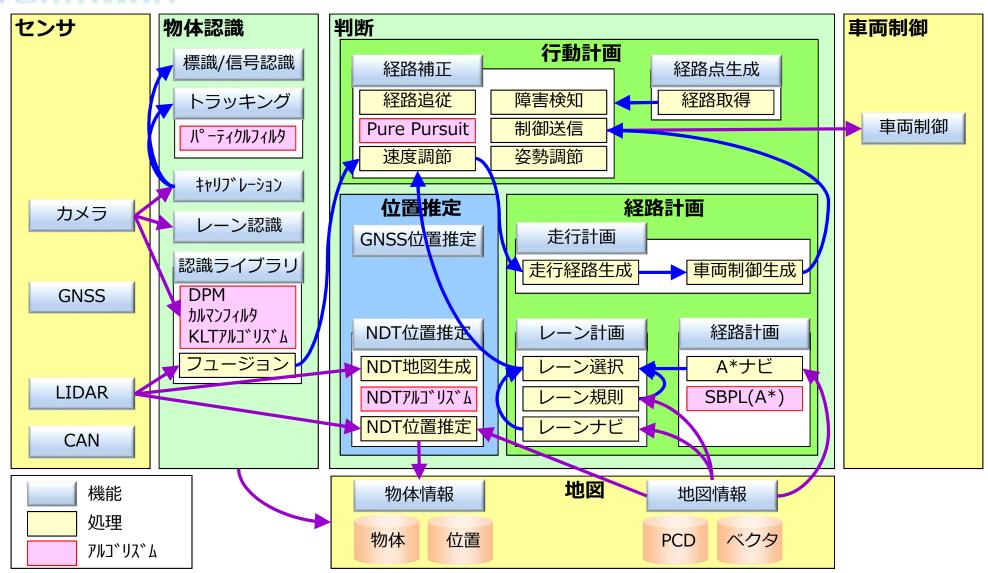
■ Autoware (ROSベースのOSS)

自律走行ロボットは基本的に以下のような構成で実現されている。





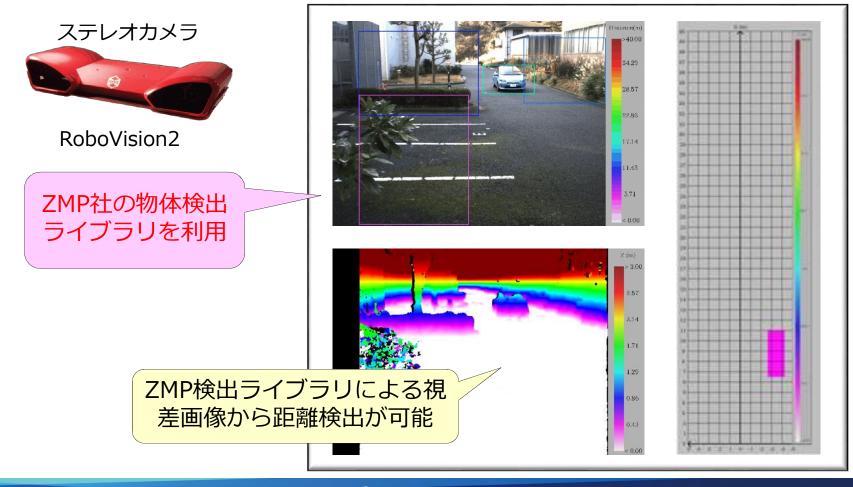
■ Autoware (ROSベースのOSS)





■ステレオカメラ(物体距離検出)

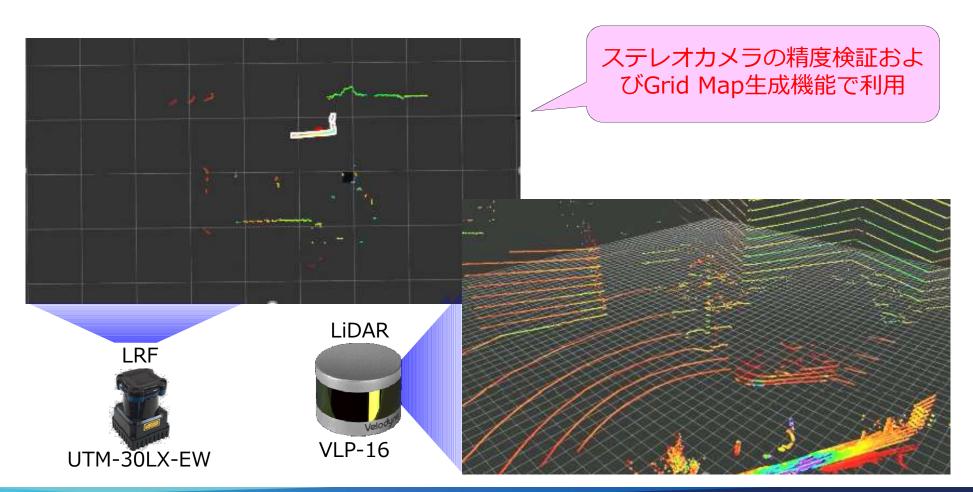
物体検出距離30m以上、距離誤差10%などを満たすため、ZMP社のRoboVision2を採用





■ LRF(北陽電機)、LiDAR(Velodyne)

ステレオカメラの精度を測定するため、レーザレンジファインダを採用





■ Blender(3Dモデリングツール)

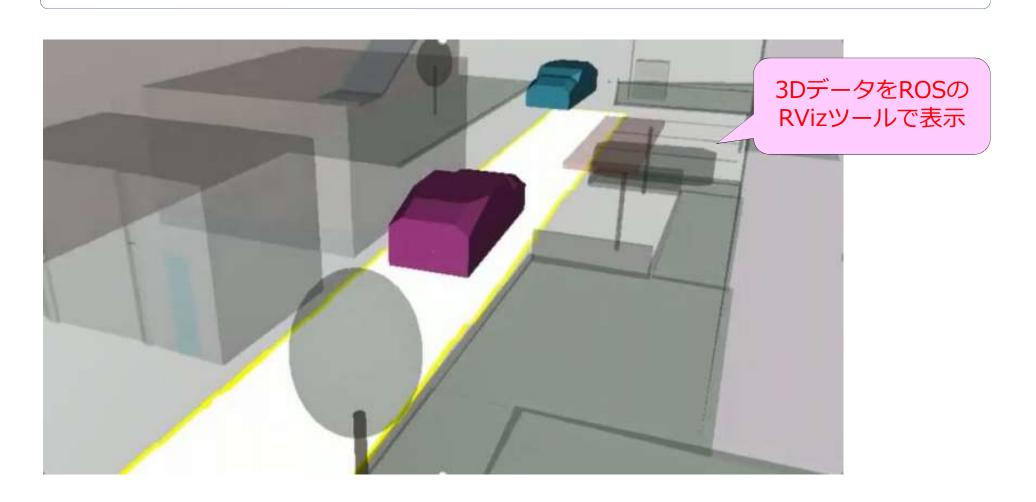
航空写真の地図などからBlenderで3D地図データを作成





■ RViz(ROSの3D表示ツール)

Blenderで作成した3DデータをROSのRvizツールで表示







- システムが大きくて移動が困難
- ステレオカメラでの認識では物体の判別が難しい
- 消費電力が大きくてバッテリーが持たない
- 設置後の位置合わせなどの設定が大変

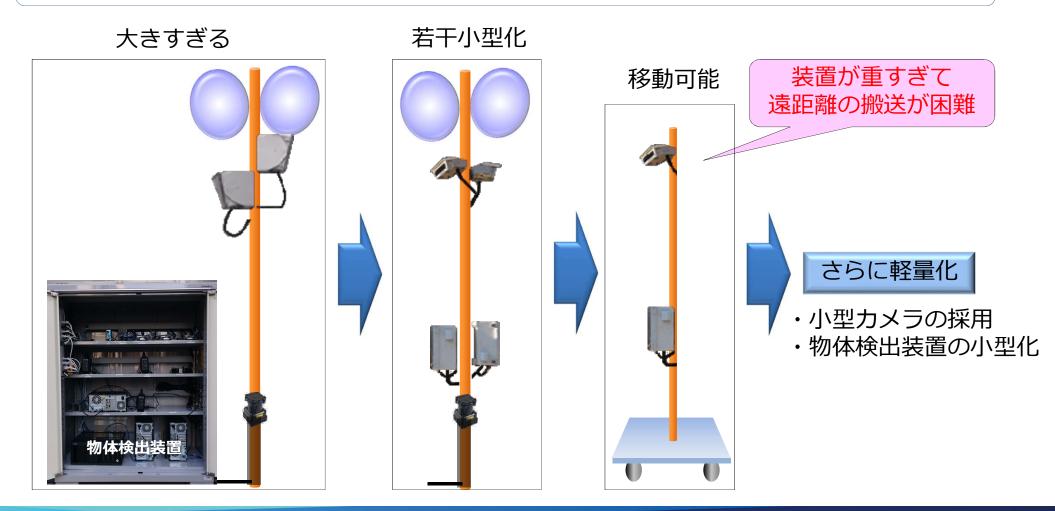


これら課題を解決するため システムをIoTエッジデバイス化



■システムが大きくて移動が困難

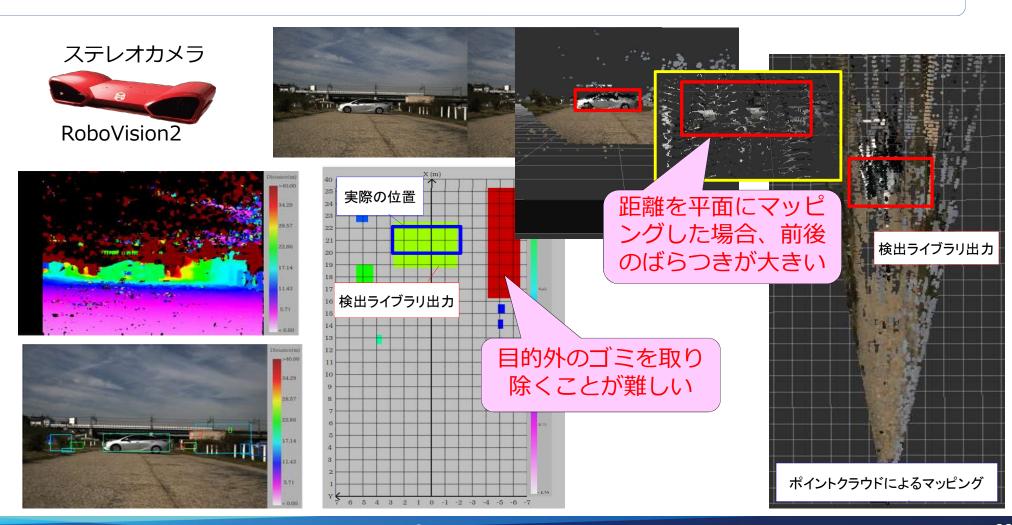
システムの小型化や移動可能にして改善したが、課題解決には不十分





■ ステレオカメラでの認識では物体の判別が難しい

ステレオカメラは物体の距離は計測できるが、車や人の認識が難しい





■ ステレオカメラでの認識では物体の判別が難しい

カメラ画像から物体認識し、対象物を特定・距離計算することで、効率よく位置推定が可能





■消費電力が大きくてバッテリーが持たない

電源の無い場所で稼働する場合、バッテリーでどれだけ動かせるかが重要



- ・ゲーミングPC
- GPU:Nvidia1080Ti
- ·消費電力:300w





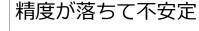


- ・高性能小型PC
- GPU:Nvidia1060

精度十分、電力大

・消費電力:100w

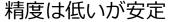






- RaspberryPi
- VPU:Movidius
- ・消費電力:10w







- · 超小型PC
- ・GPU:なし
- ・消費電力:20w



AIによる物体認識

model	mAP	FPS
FastR-CNN	70	0.5
SSD(300)	74.3	46
SSD(512)	76.8	19
YOLO	63.4	45
YOLOv2(416)	76.8	67

MobileNet SSDの採用 VPU(Mobidius)の採用



intel OpenVINO™ の採用

小型化、省電力、 GPU/FPGA未使用 中間表現(IR)で高速化



■ 消費電力が大きくてバッテリーが持たない

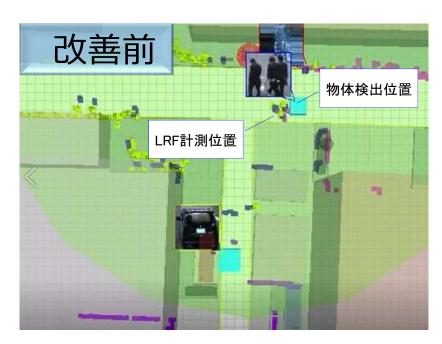
色々なボードとAIモデルの組み合わせで小型性能を比較

H/W	CPU/GPU/VPU	考察	小型 性能
ゲーミングPC	i7-8700 Nvidia1080Ti	YOLOv3で120FPS、消費電力300W⇒大き すぎ	×
高性能小型PC	i7-7700 Nvidia1060	YOLOv3で60FPS、消費電力 100W⇒200Whバッテリーで2-3時間	×
Jetson TX2	ARM A57 256基CUDA コア	YOLOv3で1FPS程度、SSDで5FPS程度、 7.5W⇒予想外に遅い、環境が複雑	Δ
RaspberryPi3B+	ARM Cortex-A53 VPU:Movidius	YOLOv3で1FPS程度、SSDで5FPS程度、 10W、5V入力なのでバッテリー利用可	0
LattePanda	Intel Atom X5- Z8300	RaspberryPi3Bと同程度だが、環境構築など×	Δ
Intel Compute Stick	Intel Core m3- 6Y30	OpenVino+MobileNet SSDで10FPS 15W程度、5V入力〇	0

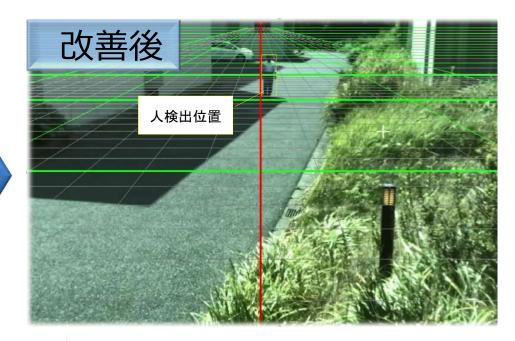


■ 設置後の位置合わせなどの設定が大変

カメラ設置後、高さや向きを実測しても実際とずれるため、キャリブレーションが大変



物体検出位置とLRF測定位置を地図上で 合わせこんでカメラの位置と姿勢を設定

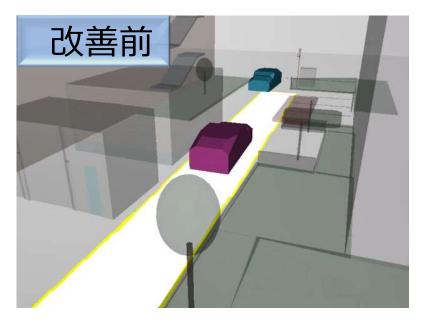


人が歩いた検出結果からカメラの高さ・ 姿勢を計算するアルゴリズムを構築



■ 設置後の位置合わせなどの設定が大変

計測場所が1ヵ所なら3Dマップを作ればいいが、別の場所での構成に手間がかかる



計測場所の3Dマップを作る必要があり、 非常に大変



改善後

ブラウザのマップ上に表示することで、 マップ作成の手間が省ける



インテリジェント・

単眼カメ

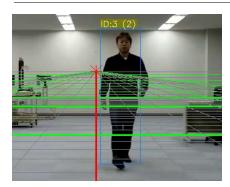
© CORE CORPORATION



■ IoTエッジデバイスとしてのカメラ



物体距離計算



ディープラーニング

による物体認識

物体認識モデル: SSD MobileNet





CPU高速演算

AIエンジン: intel OpenVINO™

小型化、省電力、GPU/FPGA未使用

ROS2メッセージ



OS: Windows ALinux 605





グローバル座標へ

マッピング

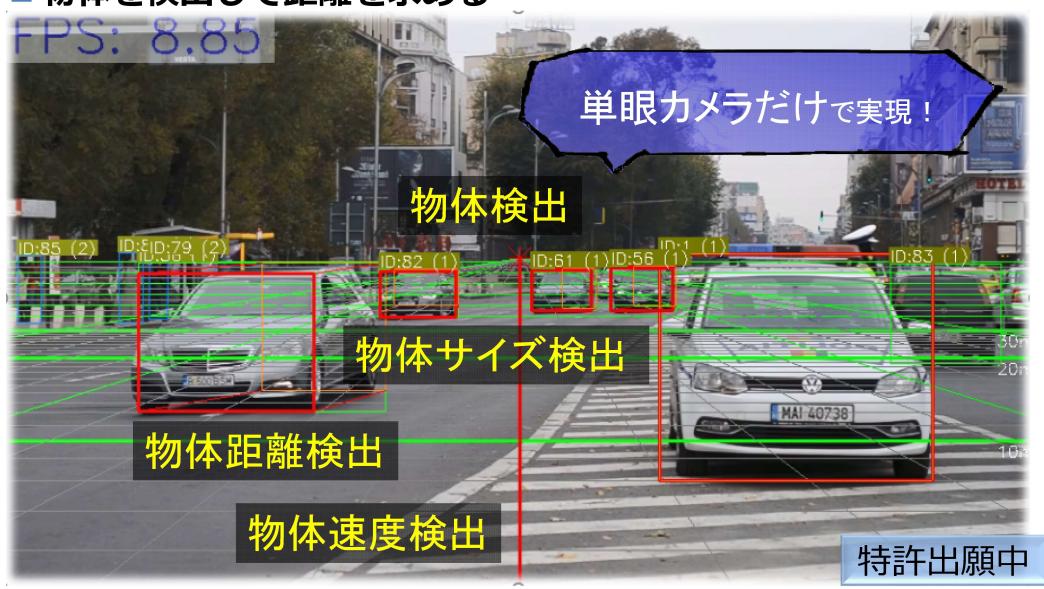


■ ET Westへの出展





■物体を検出して距離を求める



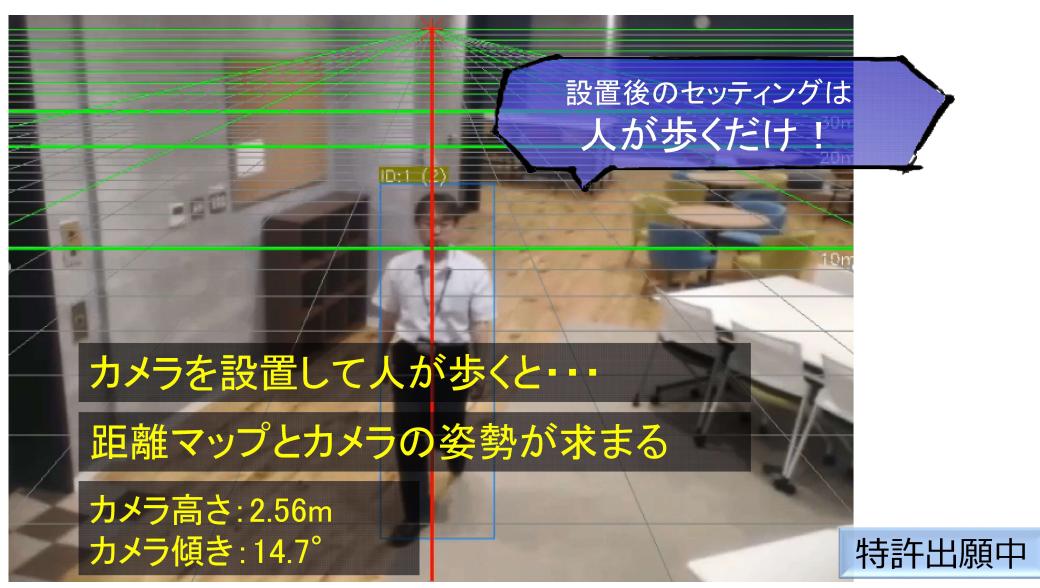


■検出した物体を地図に表示





■ 簡単なカメラセッティング





物体検出から表示までの流れ





センサー情報

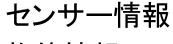
•緯度: 34.7053232

•経度:135.4946746

•方角:90度







物体情報



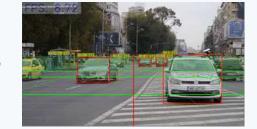




グローバル座標変換



物体検出



物体距離検出



センサー情報 物体情報

表示PC





データ統合処理



地図マッピング



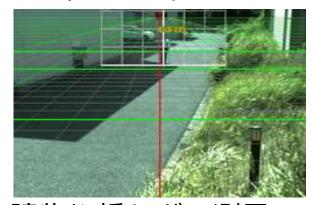
■色々な場面での利用

■交通インフラで利用 ■車や人の監視



カーブミラーの代わりに

■建築・土木 への応用



建物や橋などの測量



車両認識や人認証と連携

■GNSS との組み合わせ



■危険区域 侵入の監視



人だけの侵入を検知

■その他

モビリティへの 位置提供

認識物体の地図 マッピング



6. 今後について

6. 今後について



■更なる小型化に向けて



完全IoTエッジデバイス化

デバイス性能		
処理性能	約15~30fps	
通信	メッシュ無線	
防水	IP65	
消費電力	約1~5W	
バッテリー	37Wh	
太陽電池	20W~50W	

- ・数cm角程度のサイズ
- ・電源はソーラーパネル
- ・設置場所を問わない
- 低価格

一緒に作りませんか!?





■ 〒154-8552 東京都世田谷区三軒茶屋1-22-3 コアビル

C 03-5712-3706

