

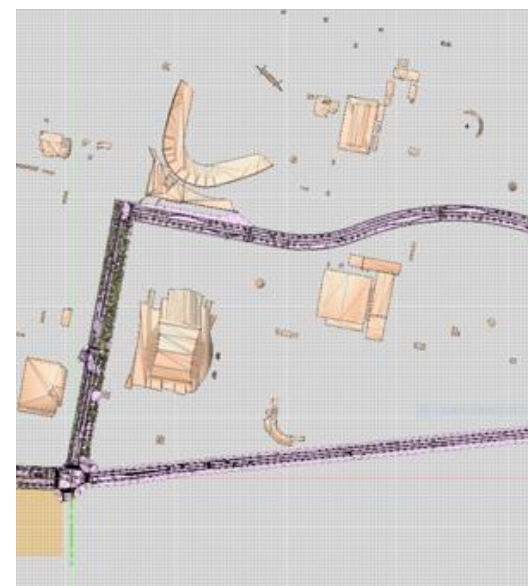
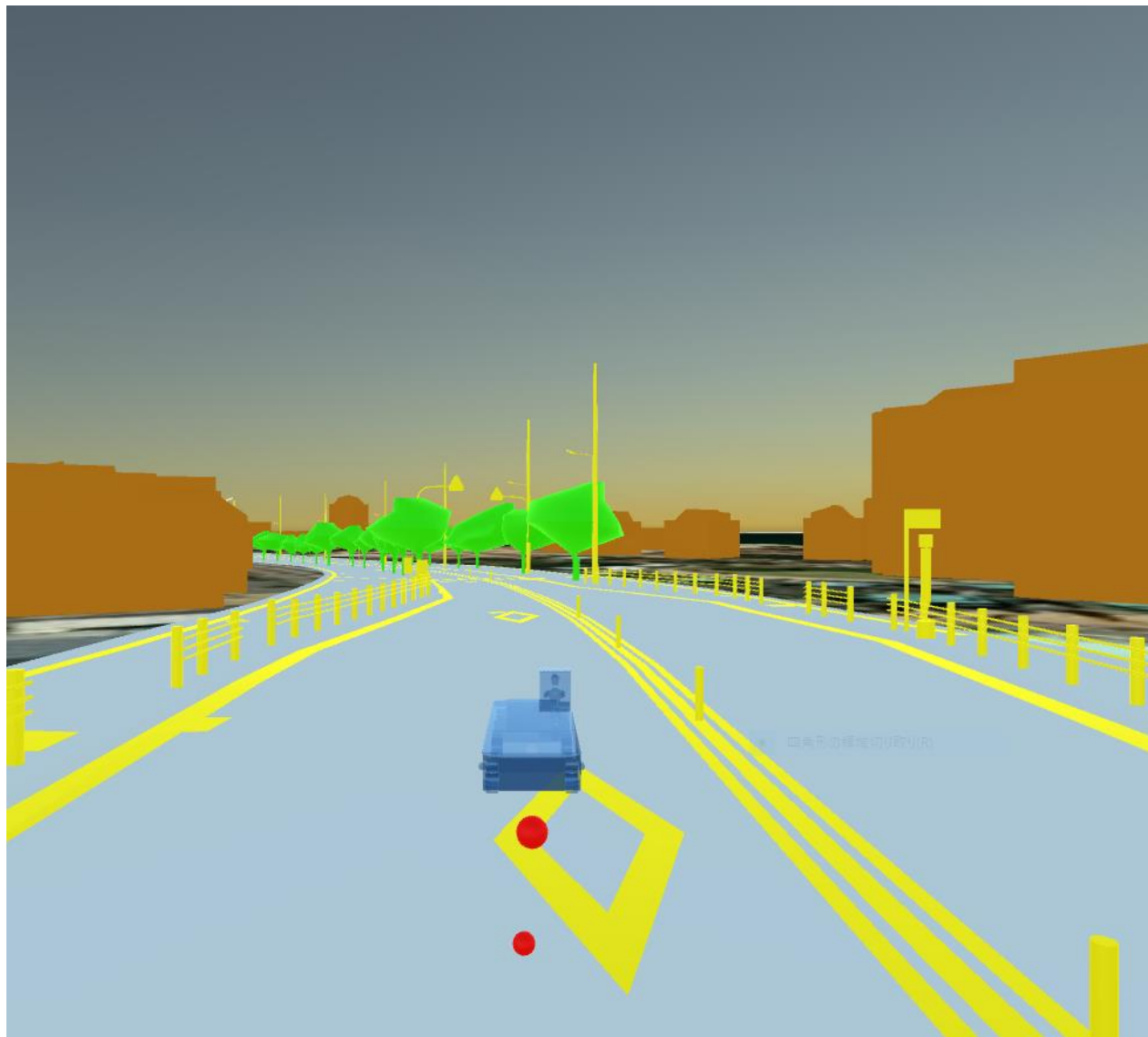
3D都市モデルとBIMを活用したモビリティ自律運行システム技術検証レポート

Technical Report for Mobility autonomous operation system using 3D City Models and BIM Models (Part:AGV)

(無人搬送車両編)



PLATEAU
by MLIT



目次

I. 実証概要	
1. 全体概要	3
2. 実施体制	5
3. 実証エリア	6
4. スケジュール	7
II. 実証技術の概要	
1. 活用技術	9
2. LiDARセンサー	10
3. LiDAR Localization	11
4. FZKViewer	12
5. Autodesk Fusion	13
6. Warpner	14
7. Warpner Viewer	15
8. Autoware.Universe	16
III. 実証システム	
1. 実証フロー	19
2. 想定事業機会	20
3. アーキテクチャ全体図	21
4. システム機能	24
5. アルゴリズム	45
6. データ	
① 活用データ	50
② データ処理	55
③ 出力データ	58
7. ユーザインタフェース	60
8. システムテスト結果	63
IV. 実証技術の検証	
1. 自己位置推定の精度検証	
① 検証内容	65
② 検証結果	68
V. 成果と課題	
1. 今年度の実証で得られた成果	
① 3D都市モデルによる技術面での優位性	79
② 3D都市モデルによるビジネス面での優位性	80
2. 今後の取り組みに向けた課題	81
用語集	82

I. 実証概要

II. 実証技術の概要

III. 実証システム

IV. 実証技術の検証

V. 成果と課題

I. 実証概要 > 1. 全体概要

全体概要 (1/2)

ユースケース名	3D都市モデルとBIMを活用したモビリティ自律運行システム（搬送車両）
実施場所	大阪府大阪市舞洲地区
目標・課題・創出価値	<ul style="list-style-type: none">• 都市部での大規模再開発に伴う建設工事による交通への影響は、現代社会において深刻な問題となっている。この問題に対処するために、都市部や屋外・屋内のシームレスな自律運行の実現が期待される。• 上記の実現には、3Dマップの活用が必要となる。但し、精度担保やデータカバレッジ等、活用に向けた技術的なハードルが存在している状況である。<ul style="list-style-type: none">- 加えて自動運転車が交通量の多い都市部で安全に運行するためには、リアルタイムの3Dマップが必要である。• 3D都市モデルを活用し安全かつ効率的に自律運行の実現するシステムの構築を目指す。
ユースケースの概要	<ul style="list-style-type: none">• 現在の搬送車両に係る下記課題を解決し、資材運搬時などの自律運行を可能とする運行システムを開発する<ul style="list-style-type: none">- 自律運行では、走行ルートの点群マップ生成が必要不可欠である。その点群マップを3D都市モデルを用いて生成し、自律走行させ、その精度を計測する。- 自律運行の場合、遠隔でのモニタリングが必要となるケースがある。そのために、車両情報をリアルタイムで監視し、3D都市モデルと連携したリアルタイムモニタリングシステムを開発する。

I. 実証概要 > 1. 全体概要

全体概要 (2/2)

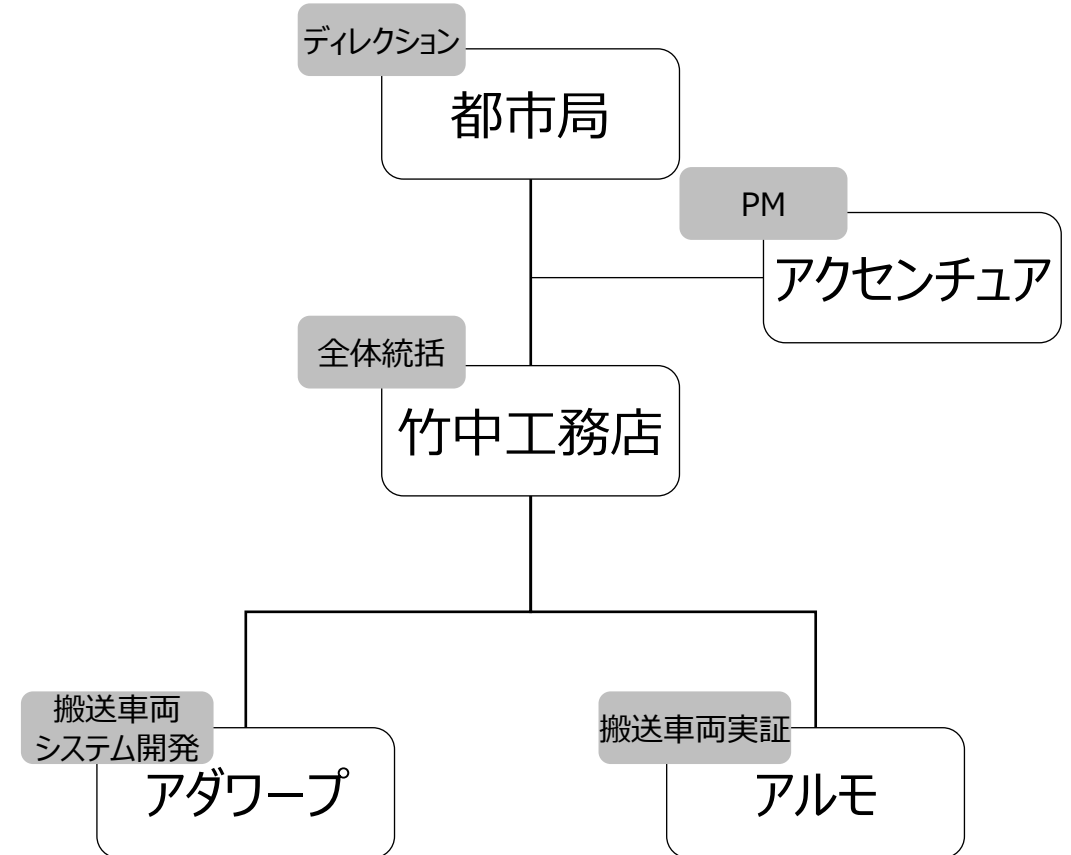
実証仮説	<ul style="list-style-type: none">• 3D都市モデルを応用し自動運転に必要な点群マップを生成することで、開発のコスト削減が可能となる。<ul style="list-style-type: none">- 点群マップには、道路の形状や標識、信号、建物などが3次元の点群データとして表現され、自動運転車は点群マップを基に、自分自身の位置や周囲の環境を正確に把握することができる。
検証ポイント	<ul style="list-style-type: none">• LOD3のモデルを使って自動運転マップを作製した場合の自己位置推定の精度。• 正確な推定が難しいと考えられる下記エリアでの自己位置推定の精度。<ul style="list-style-type: none">- 建物がなくその他の地物（植生、道路、等）のみで推定が必要なエリア- 現実と3D都市モデルの提供データ量や粒度に差異があるエリア

I. 実証概要 > 2. 実施体制 実施体制

各主体の役割

主体	役割
竹中工務店	全体統括
アダワープ	搬送車両対応システム作成
アルモ	搬送車両自律走行実証
アクセンチュア	プロジェクトマネジメント

実施体制図



I. 実証概要 > 3. 実証エリア 実証エリア

対象エリア：大阪府 大阪市 舞洲地域 (2.2km²)



I. 実証概要 > 4. スケジュール スケジュール

実施事項	令和4年										令和5年			
	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	
1. 3D都市モデルの実証方法の検討	←→			←→										
2. 3D都市モデルデータ整備			←→											
3. 搬送車両自律走行システム開発			←→											
4. 搬送車両自律走行実証				←→										
5. 報告書作成									←→					

I. 実証概要

II. 実証技術の概要

III. 実証システム

IV. 実証技術の検証

V. 成果と課題

Ⅱ. 実証技術の概要 > 1. 活用技術 活用技術一覧

項目	内容
LiDARセンサー	レーザー光の反射によって対象物との距離や形状を図るLiDAR技術を用いた測距センサーで、点群データを取得可能にする技術
LiDAR Localization	LiDARセンサーから出力される点群データとエリアマップをマッチングすることで、自身の位置を推定する技術
FZKViewer	CityGMLを視覚化するためのソフトウェア
Autodesk Fusion	Autodeskによって開発された、3D CAD、CAM、CAEの機能を統合したクラウドベースの製品設計ソフトウェア
Warpner	LTE経由で低遅延の映像を送信するAndroidアプリケーション
Warpner Viewer	Warpnerアプリから映像を受信するためのブラウザアプリケーション
Autoware.Universe	自動運転のオープンソースソフトウェアプラットフォーム

Ⅱ. 実証技術の概要 > 2. LiDARセンサー

LiDARセンサーとは

レーザー光の反射によって対象物との距離や形状を図るLiDAR技術を用いた測距センサーで、点群データを取得可能

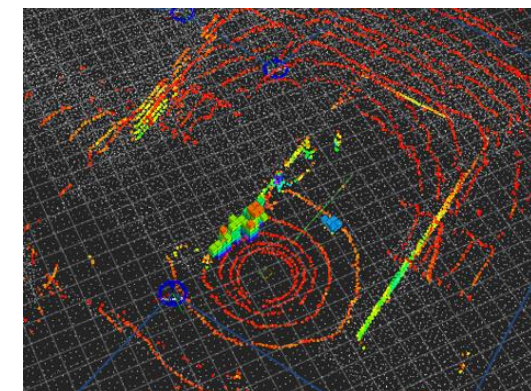
技術概要

項目	内容
名称	LiDAR (Light Detection and Ranging、Laser Imaging Detection and Ranging) センサー
概要	<ul style="list-style-type: none"> レーザー光の反射によって、対象物との距離や形状を図る技術を用いたセンサー
特徴	<ul style="list-style-type: none"> LiDAR技術を使ったLiDARセンサーVCSELレーザーアレイ、デジタルレシーバSoC (ASIC)、マイクロ光学系という3つの技術を統合し、霧や雪等の障害物のある環境下でも、生物・木・金属など物体の種類も含めて識別する点群データを取得することが可能
本ユースケースでの用途	<ul style="list-style-type: none"> 自己位置推定のための点群取得に利用

LiDARセンサーと出力データイメージ



Ouster社LiDARセンサー OS0



Ouster OS0から出力される点群

Ⅱ. 実証技術の概要 > 3. LiDAR Localization

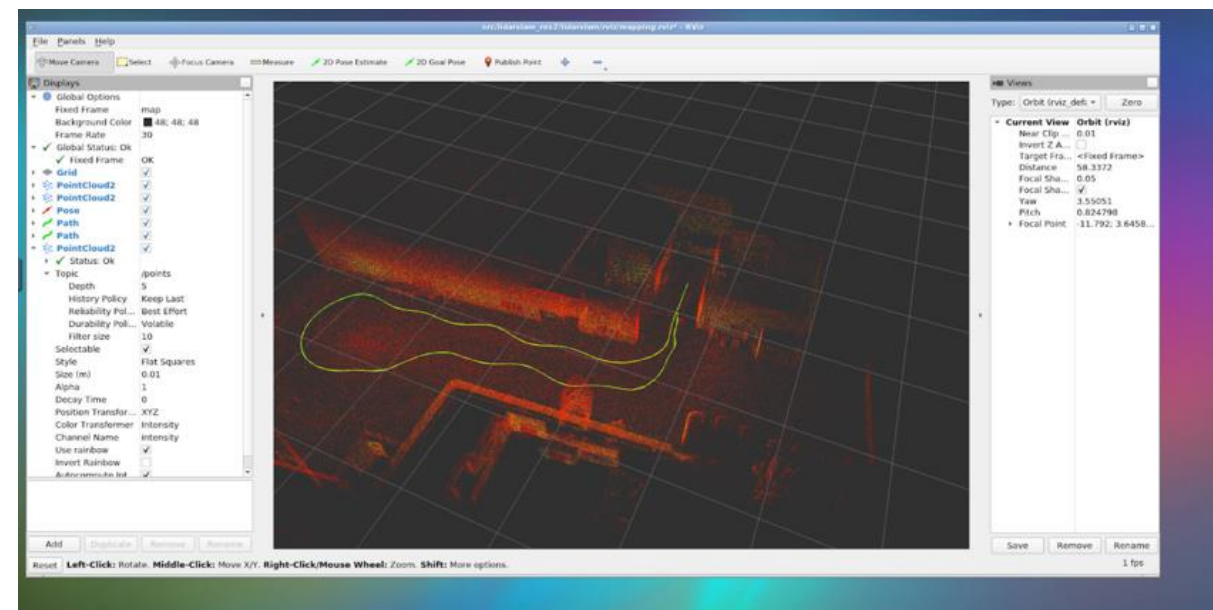
LiDAR Localizationとは

LiDAR (Light Detection and Ranging) と呼ばれるレーザーを用いたセンサーから取得したデータを利用して、車両の正確な位置情報を推定する技術

技術概要

項目	内容
名称	LiDAR Localization
概要	<ul style="list-style-type: none"> LiDARセンサーによって得られた周辺の形状でデータ（点群）をエリアマップと比較することで、現在の自己位置を推定する手法
主な機能	<ul style="list-style-type: none"> 自己位置推定
利用する機能	<ul style="list-style-type: none"> 3D都市モデルを事前マップとして自己位置推定機能を利用

LiDAR Localizationによる自己位置推定例



Ⅱ. 実証技術の概要 > 4. FZKViewer

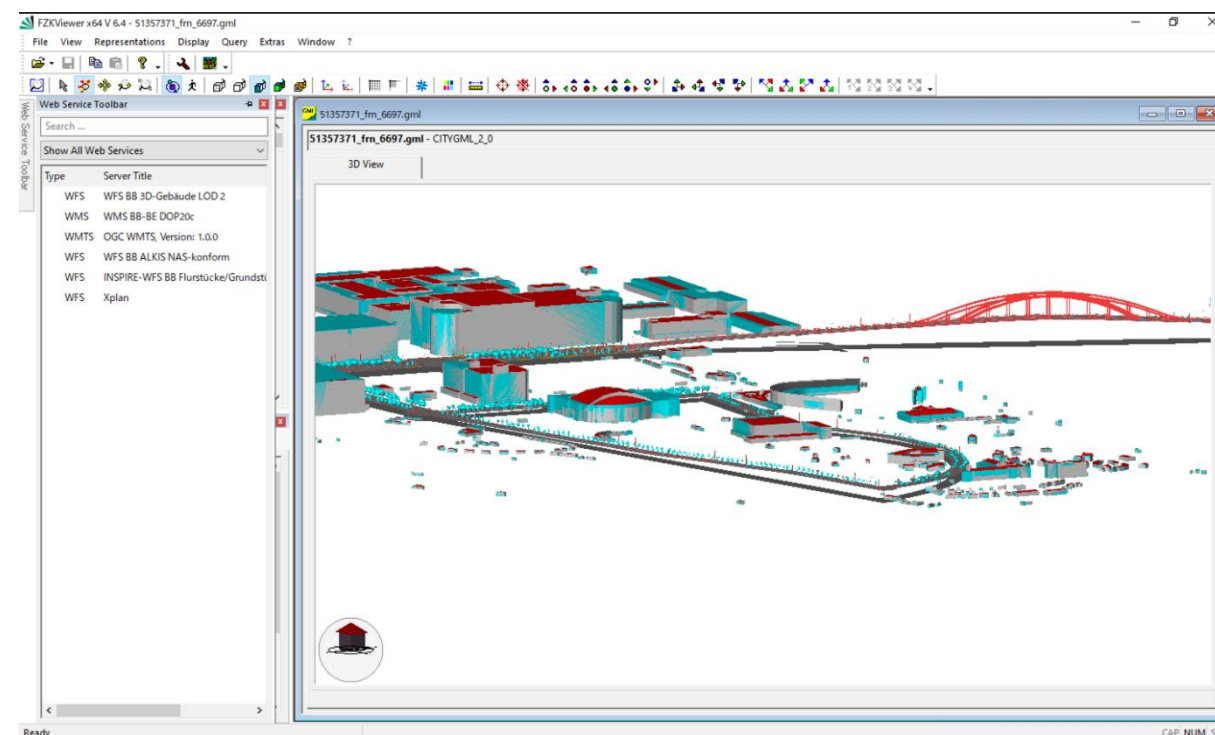
FZKViewerとは

建築情報モデリング、地理情報システムのデータモデルを可視化するためのソフトウェア

技術概要

項目	内容
名称	FZKViewer
概要	<ul style="list-style-type: none"> 建築情報モデリング（BIM）や地理情報システム（GIS）の分野からのセマンティックデータモデル（現実世界の情報や関係性を含めた情報）を可視化するためのソフトウェア
主な機能	<ul style="list-style-type: none"> CityGMLの可視化 BIMの可視化 STL、FBXなどの3Dモデルを表現するための形式へ変換
利用する機能	<ul style="list-style-type: none"> CityGMLをSTLに変換するために利用

FZKViewerでの3D都市モデル表示例



Ⅱ. 実証技術の概要 > 5. Autodesk Fusion

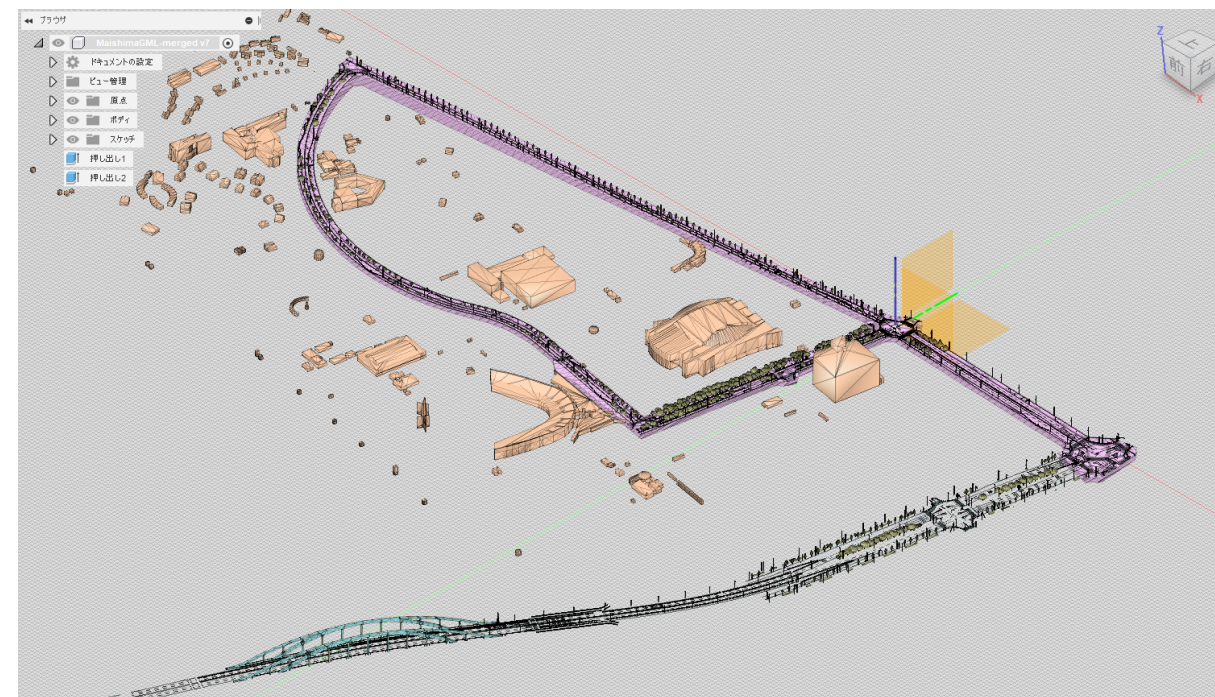
Autodesk Fusionとは

クラウドベースの高機能 3D CAD、CAM、CAE、PCB 統合システム

技術概要

項目	内容
名称	Autodesk Fusion
概要	<ul style="list-style-type: none"> 3DモデリングやCAD、シミュレーション、CAMなどの機能を備えた、クラウドベースの総合設計ソフトウェア
主な機能	<ul style="list-style-type: none"> 3Dモデリング 3Dモデル変換 シミュレーション等
利用する機能	<ul style="list-style-type: none"> STL形式のモデルをOBJ形式に変換

Autodesk Fusionでの3D都市モデル表示例



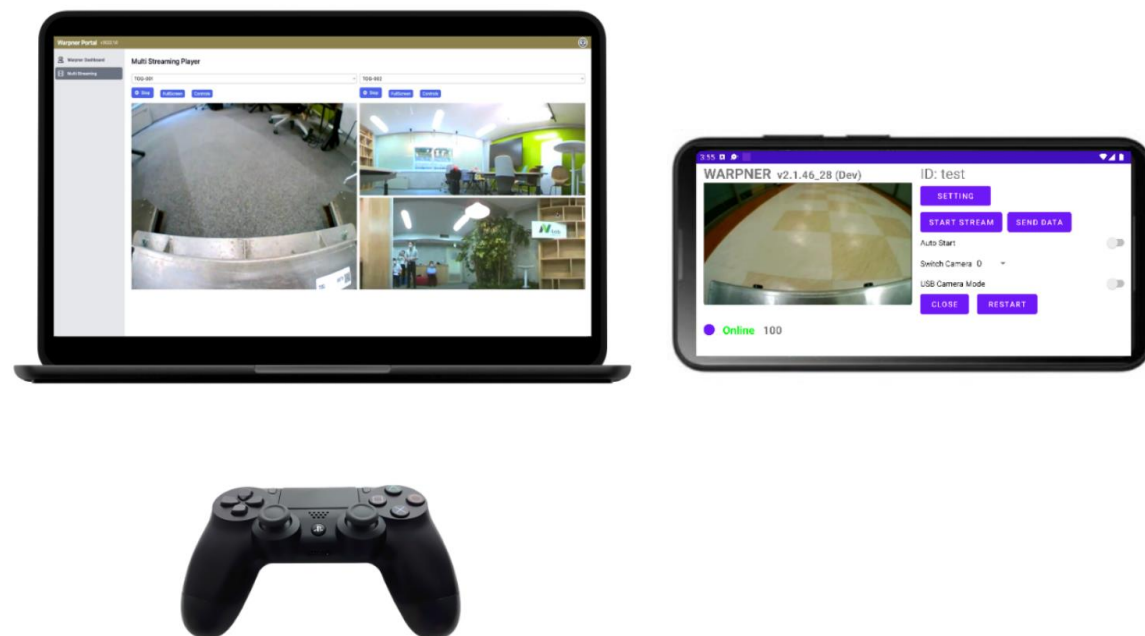
Ⅱ. 実証技術の概要 > 6. Warpner Warpnerについて

LTE経由で低遅延の映像を送信するAndroidアプリケーション

技術概要

項目	内容
名称	Warpner Android
概要	<ul style="list-style-type: none"> スマートフォンを対象物に取り付けるだけで遠隔操作を可能にするAndroidアプリ 映像はブラウザアプリのWarpner Viewerにて受信
主な機能	<ul style="list-style-type: none"> 映像伝送 操作信号伝送
利用する機能	<ul style="list-style-type: none"> 搬送車両に取り付け、前方後方、左右、ドライバーの監視を行う

Warpnerの最小構成図（ブラウザアプリ、Androidアプリ）



Ⅱ. 実証技術の概要 > 7. Warpner Viewer Warpner Viewerについて

Warpnerアプリから映像を受信するためのブラウザアプリケーション

技術概要

項目	内容
名称	Warpner Viewer
概要	<ul style="list-style-type: none"> Warpnerアプリから映像を受信するブラウザアプリケーション
主な機能	<ul style="list-style-type: none"> 映像受信 音声送受信
利用する機能	<ul style="list-style-type: none"> 公道走行時の車両の映像監視 遠隔監視室にて、四画面で車両の前方後方、左右、ドライバーの映像をリアルタイムで監視

遠隔監視室の様子



Ⅱ. 実証技術の概要 > 8. Autoware.Universe

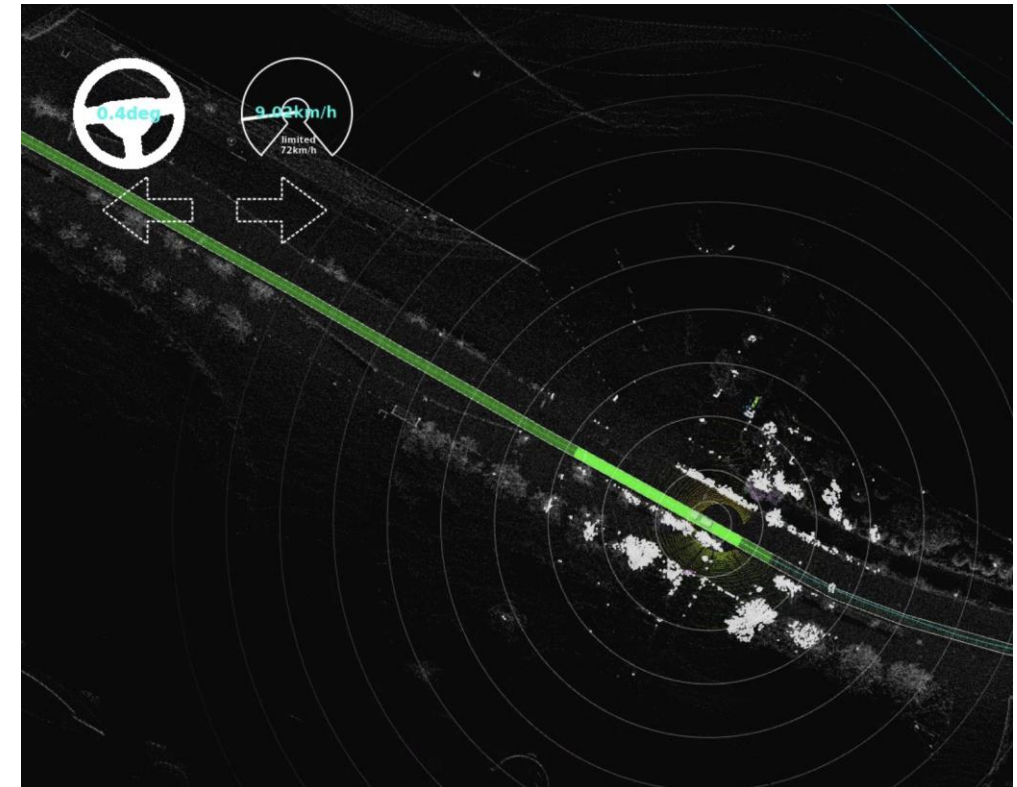
Autoware.Universeについて

自動運転のオープンソースソフトウェアプラットフォーム

技術概要

項目	内容
名称	Autoware.Universe
概要	<ul style="list-style-type: none">自動運転車両の開発に必要な機能を提供する、複数のモジュールやパッケージの集合体。ローカライゼーション、知覚、予測、計画、制御、シミュレーションなど、さまざまなソフトウェアコンポーネントが含まれる。
主な機能	<ul style="list-style-type: none">自己位置推定人物検知経路計画行動計画
利用する機能	<ul style="list-style-type: none">3DLiDARを用いた自己位置推定搬送車両を公道で走行する際の経路計画、行動計画

Autoware.Universe開発画面



II. 実証技術の概要 > 8. Autoware.Universe

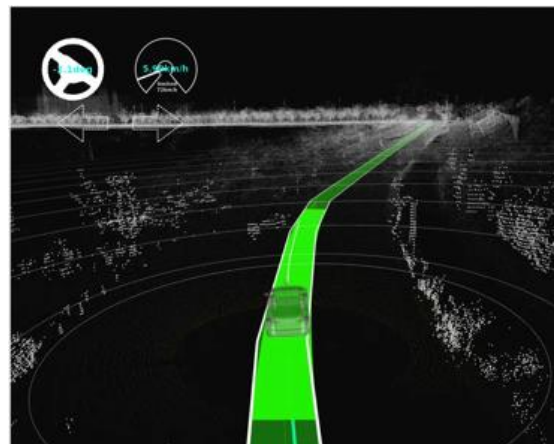
Autoware.Universeの構成要素

LinuxとROS2をベースとした自動運転システム用オープンソースソフトウェアで、レーザレーダ、カメラ、GNSSなどの環境センサを利用し、自車位置や周囲物体を認識しながら、走行ルート上を自律走行する機能を有する

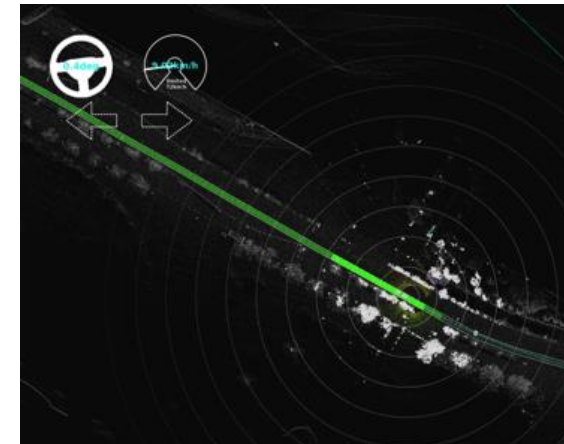
Autoware.Universeの構成要素

Autoware-sensing	<ul style="list-style-type: none"> • LiDAR、RTK測位デバイス、カメラなどのセンシングデータをAutowareに取り込む
Autoware-localization	<ul style="list-style-type: none"> • sensingで得られたデータ、また与えられた点群マップより自己位置推定を行う
Autoware-control	<ul style="list-style-type: none"> • planningから出力される指示情報を基に、車両を走行させる指示を出す
Autoware-planning	<ul style="list-style-type: none"> • localizationで得られた自己位置を基に、経路を計画する
Autoware-perception	<ul style="list-style-type: none"> • sensingで得られたデータから、対向車、人物、障害物を検知する（本実証では利用なし）

システムでの画面



Autowareで自己位置推定をしている様子



上空ビュー

I. 実証概要

II. 実証技術の概要

III. 実証システム

IV. 実証技術の検証

V. 成果と課題



Ⅲ. 実証システム > 1. 実証フロー

実証フロー

実証実験では、実務と同様の手法による算出結果との比較を通じて自己位置推定の精度を検証するとともに、公道での自動走行を通じてその有用性を明らかにする

対象エリアのルート
調査・選定

- 3D都市モデルをもとに、車両が走行可能なルートを調査する
- 各ルートの距離や所要時間、交通量、信号機のタイミングを調査する

公道走行申請

- 警察署に対して、公道走行の申請を行う
- 申請には、車両の詳細な仕様、走行ルート、安全対策、運行時間帯、走行距離などが含まれる

システム開発

- 3D都市モデルのデータ変換、点群マップの生成、遠隔監視に必要なソフトウェアの構築を行い、並行して3D LiDAR、RTK測位、カメラなどのハードウェアの構築を行う
- 構築したシステムを利用して、自動運転車が走行するエリアのセンサーデータを収集する

システムの精度検証

- 自己位置推定の精度を検証するために、「1. 手動走行時により作成された点群マップ」、「2. 3D都市モデル由来の点群マップ」の2種類のマップを利用し、点群マッチングの精度を検証する

システムの有用性検証

- 想定コースのうち、安全性の基準を満たすエリアに関して自動走行を行い、遠隔監視室でカメラ映像、車両情報をリアルタイムで確認することで、本システムの実務上の有用性を評価する



Ⅲ. 実証システム > 2. 想定事業機会 想定事業機会

搬送車両を自律走行させるためのシステムや遠隔モニタリングを行うシステムをゼネコンや運輸サービス事業者に提供し、車両運行管理の省力化を実現する

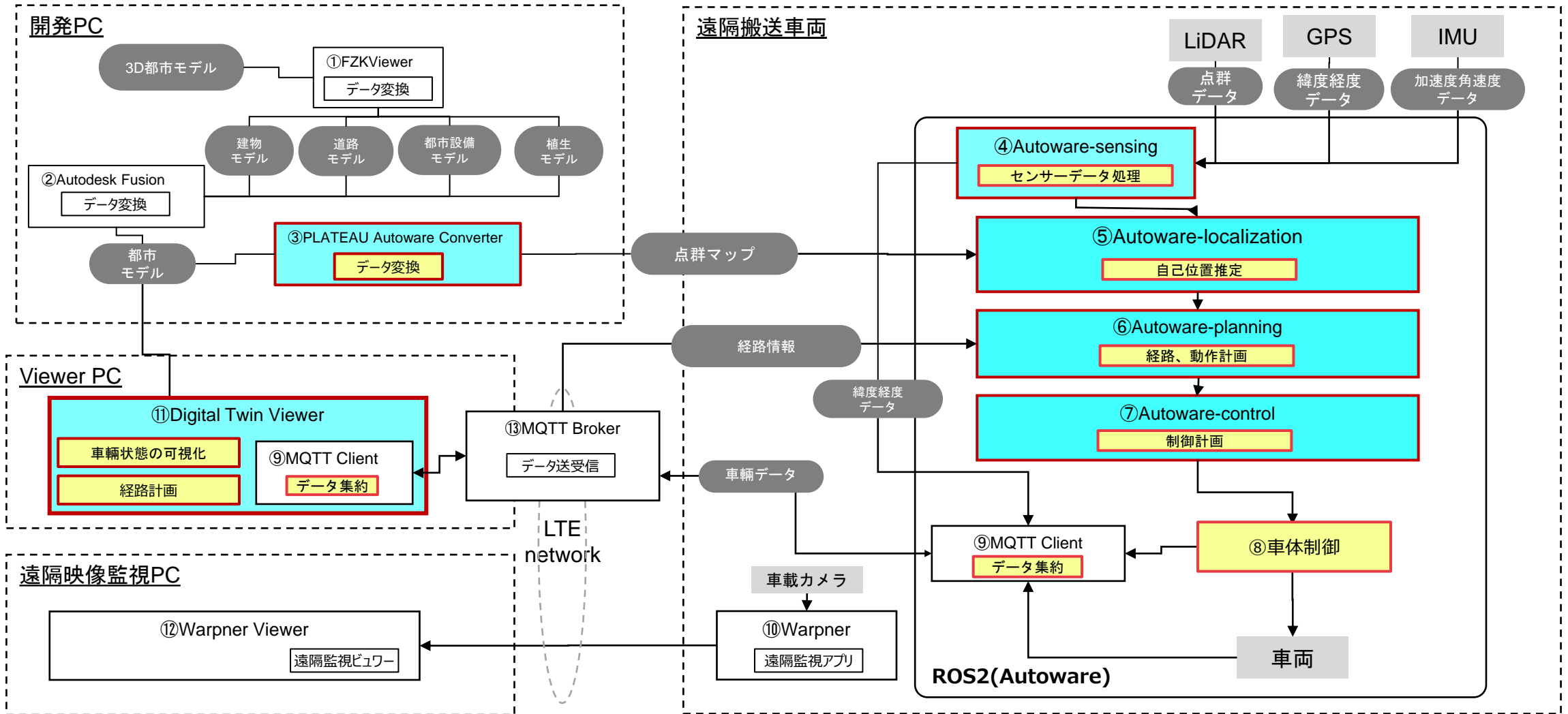
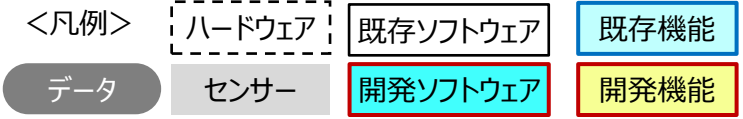
項目	内容
利用事業者	<ul style="list-style-type: none">● ゼネコン● 運輸サービス事業者
提供価値	<ul style="list-style-type: none">● GPS測位のみでは安全な公道自律走行が担保できない搬送車両に対して、3Dマップデータを活用した安全な自律走行システムを提供する<ul style="list-style-type: none">- 資材運搬や物流等を担う搬送車両の自律走行を可能とするため、LiDARやGPS等のセンサーと3D都市モデルを利用したGPSのみに依存しない自己位置測位を組み合わせたシステムとする● 建築・物流事業者が利用可能な、デジタルツインビューを組み合わせたユーザインターフェースを開発し、提供する
サービス仮説	<ul style="list-style-type: none">● 自律走行の搬送車両による業務支援サービス<ul style="list-style-type: none">- 資材運搬・物流等の省力化につながる自律走行システムによる業務支援サービスを提供する● 搬送車両の自律走行車両の遠隔モニタリングサービス<ul style="list-style-type: none">- 自己位置推定とデジタルツインビューによる自律走行支援型遠隔車両の管制サービスの提供を行う

Ⅲ. 実証システム > 3. アーキテクチャ全体図

システムアーキテクチャ全体図

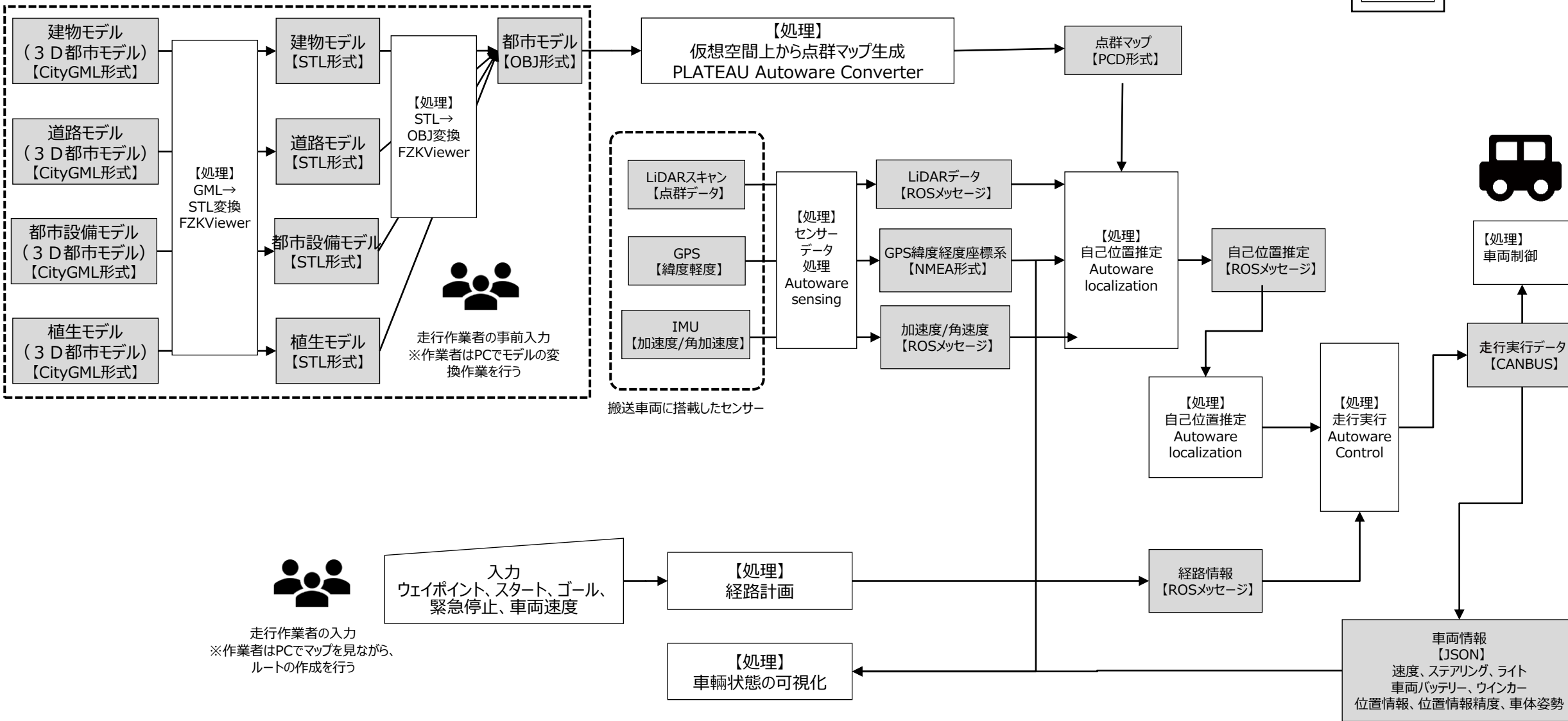
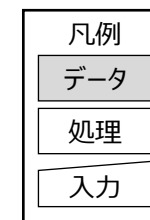


PLATEAU
by MLIT



Ⅲ. 実証システム > 3. アーキテクチャ全体図

データアーキテクチャ全体図



Ⅲ. 実証システム > 3. アーキテクチャ全体図

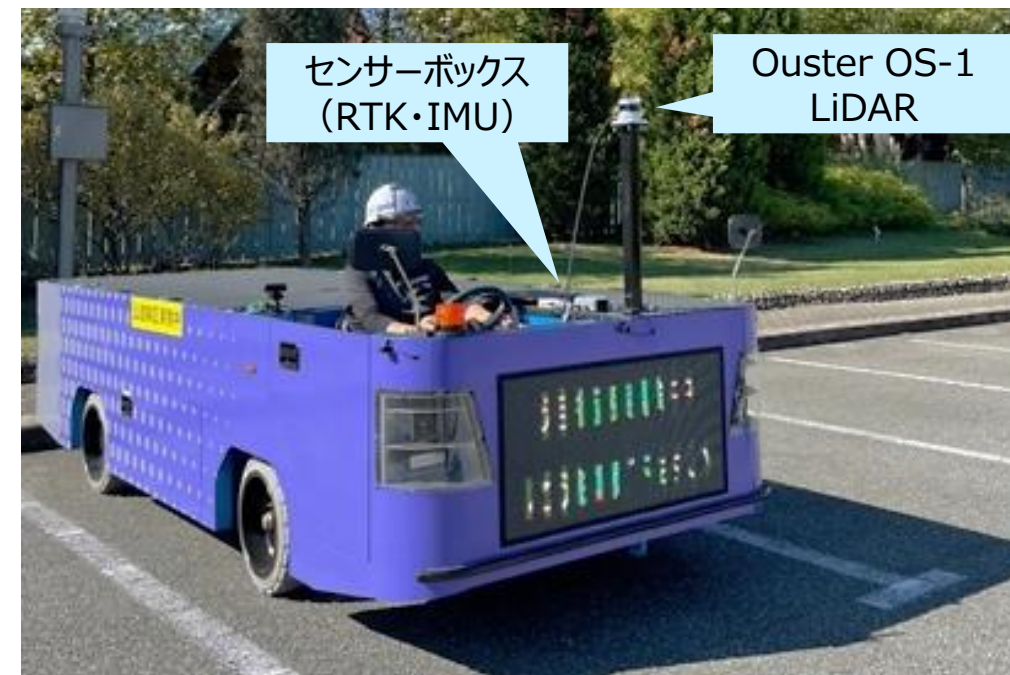
搬送車両のハードウェアスペック

本実証実験に向け、資材運搬が可能な自動搬送車両を開発した

搬送車両の諸元表

No.	項目	単位	規格	備考
1	積載荷重	Kg	4,000	
2	荷台寸法	L	mm	4,695
3		W	mm	1,660
4		H	mm	800
4	総重量	Kg	1,600	空車重量
5	本体素材		4.5t Checkered Plate	模様鋼板
6	駆動方式		後輪	
7	速度	積載なし	Km/h	15
8		積載あり	Km/h	13
9	登坂能力	積載なし	deg	20 (37%)
		積載あり	deg	8.5 (15%)
10	最大走行距離	Km	40	平地10km/h定速走行
11	車両寸法	L	mm	4,695
12		W	mm	1,660
13		H	mm	800
14	左右タイヤ間	ホイールベース	mm	2,095
15		前輪	mm	1,450
16	寸法	後輪	mm	1,450
17	最小旋回半径	M	4.6	
18	車高	mm	150	

搬送車両の外観



Ⅲ. 実証システム > 4. システム機能 システム機能一覧 (1/2)

<凡例> **赤太字** : 開発要素

	ソフトウェア名	機能名	詳細
①	FZKViewer	データ変換	3D都市モデル (CityGML) をSTLに変換
②	Autodesk Fusion	データ変換	STLをOBJに変換
③	PLATEAU Autoware Converter	データ変換	3D都市モデル (OBJ) を仮想空間上に読み込み、仮想車両を走行させ、点群マップを生成
④	Autoware-sensing	センサーデータ処理	3D LiDAR (Ouster) から取り込んだデータをAutoware (ROS2) で扱えるように変換、ダウンサンプル処理に利用
⑤	Autoware-localization	自己位置推定	3D LiDARの点群データと、点群マップを元に、点群マッチングを行い、自己位置の推定に利用
⑥	Autoware-planning	経路計画	自己位置を基に経路追従、動作計画を定めるのに利用
⑦	Autoware-control	動作計画	経路追従、動作計画から命令される動作指示を搬送車両に伝える

Ⅲ. 実証システム > 4. システム機能 システム機能一覧 (2/2)

<凡例> **赤太字** : 開発要素

	ソフトウェア名	機能名	詳細
⑧	ROS2	車体制御	CANBUSを通じて車体の速度、ステアリングの制御を行う 既存ソフトウェア (ROS2 Autoware) を本車輛のハードウェアに対応させるためのカスタマイズ (開発) を実施
⑨	MQTT Client	データ集約	車両データをJSON化し、MQTT Brokerへ送信
⑩	Warpner	遠隔監視モジュール	遠隔監視者がリアルタイムで搬送車両の映像を見ることができるアプリケーション
⑪	Digital Twin Viewer	車輛状態の可視化	3D都市モデル (CityGML) を運搬車が走行するリアルタイムモニタリングシステム
⑫	Digital Twin Viewer	経路計画	自動運転の経路を計画し、車体に経路を送信
⑬	Warpner Viewer	搬送車両搭載のカメラ映像ビューワー	車両の映像をリアルタイムで監視できるブラウザベースのアプリケーション
⑭	MQTT Broker	データ送受信	JSON化された車両データを受信し、遠隔監視用のモニターに送信するサーバー

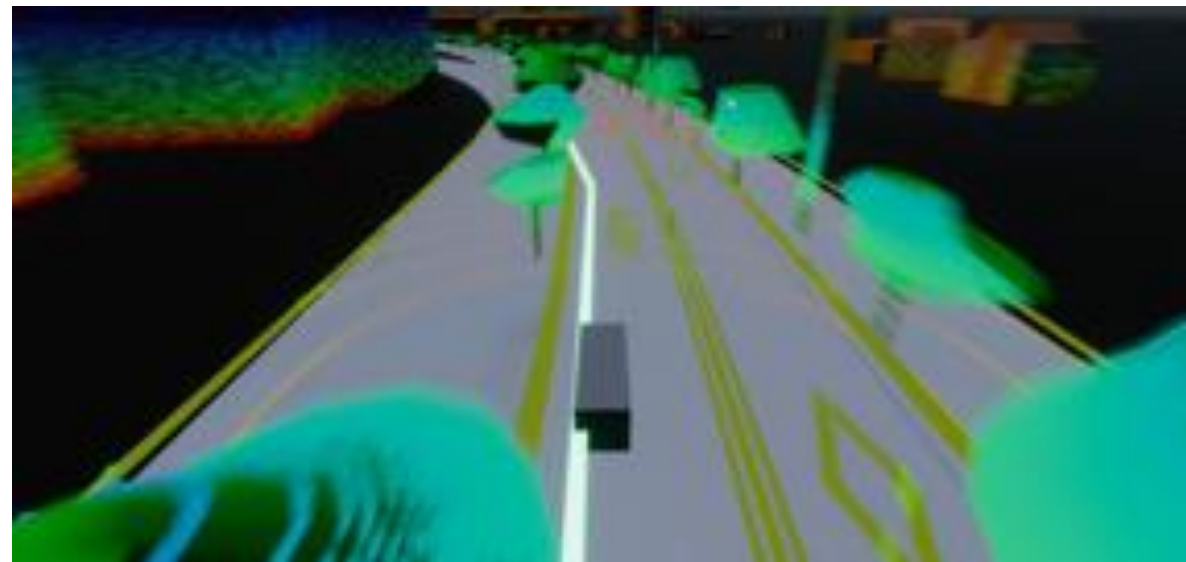
Ⅲ. 実証システム > 4. システム機能

③データ変換 (PLATEAU Autoware Converter)

概要

3D都市モデルを仮想車両が走行、点群マップを擬似的に生成

項目	詳細
機能名	データ変換 (PLATEAU Autoware Converter)
機能概要	<ul style="list-style-type: none">3D都市モデルを仮想空間上に読み込み、仮想車両を走行させ、点群マップを生成。
入力データ仕様	<ul style="list-style-type: none">3D都市モデル (obj形式)
出力データ仕様	<ul style="list-style-type: none">点群マップ (pcd形式)
利用するライブラリ	<ul style="list-style-type: none">Unityros2-for-unityRobtoec-GPU-LiDARAWSIM



Ⅲ. 実証システム > 4. システム機能

③ データ変換 | 処理フロー

3D都市モデルを点群データにするために、仮想空間内で仮想LiDARによる点群取得を行う点群データコンバーターを開発

PLATEAU Autoware Converterによる3D都市モデルの点群変換処理フロー

OBJ形式の都市モデルを
読み込み

- OBJ形式かされた3D都市モデルをUnityのシーンに読み込み

仮想LiDARの
放射パターンを実施

- OBJをPLATEAU Autoware Converterに読み込み、点群化に必要なパラメータを設定
- 仮想空間内の3Dモデルに仮想LiDARによる放射を行う

ノイズ除去
(ガウシアンフィルター)

- ガウシアンフィルターにより放射ノイズを除去し、仮想空間での点群取得品質を高める

レイトレース
(光源のシミュレート)

- 光源から放射された光の軌跡を1本ずつトレースし反射した位置を点として取得する
- これを繰り返し、点群データを取得する

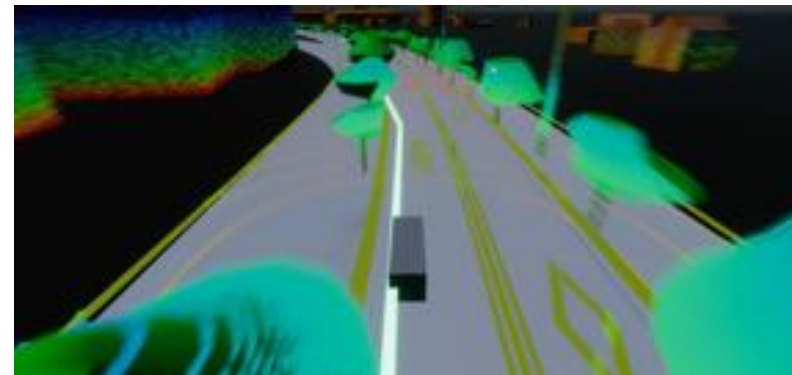
点群のダウンサンプル

- 得られた点群データを必要な点群密度に間引き処理

点群データ生成

- PCD形式で点群データを書き出し

仮想空間上を仮想のLiDARと車両で走行し点群マップを生成



出力された点群マップ



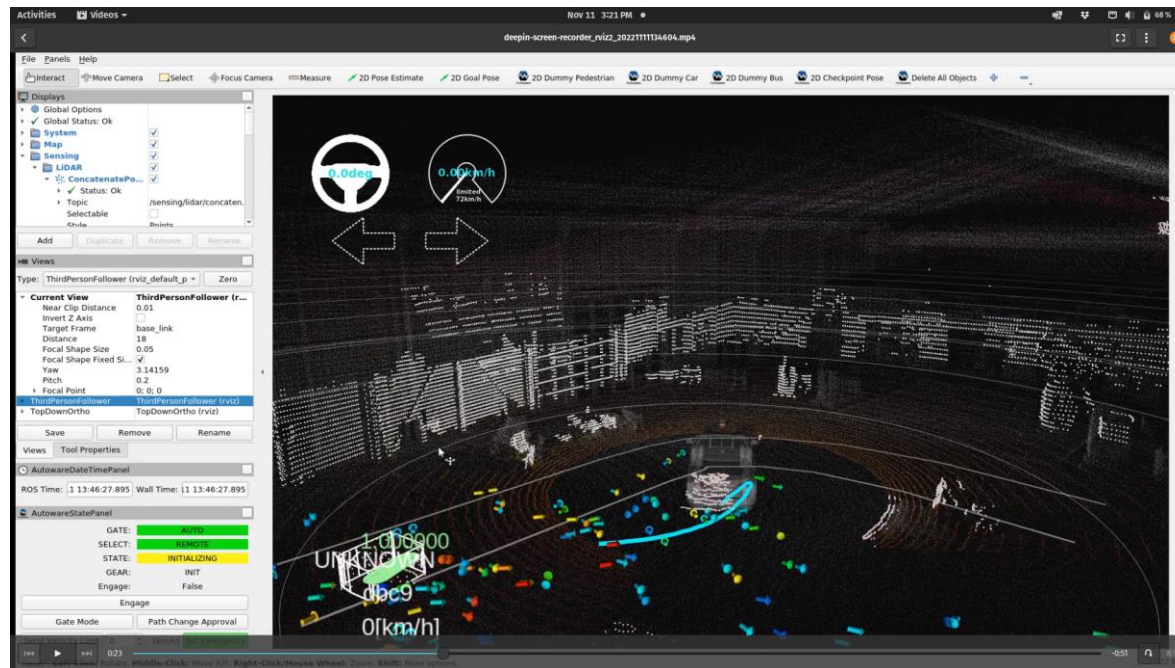
Ⅲ. 実証システム > 4. システム機能

④ センサーデータ処理 (Autaware-sensing)

概要

Autaware-sensingにより、LiDARから点群を可視化

項目	詳細
機能名	<ul style="list-style-type: none"> センサーデータ処理
機能概要	<ul style="list-style-type: none"> Lidar, RTK測位デバイス, IMUなどのセンサからのデータの取得と処理
入力データ仕様	<ul style="list-style-type: none"> LiDAR (PCD形式) RTK測位 (NMEA形式) IMU 加速度、角加速度 (float形式)
出力データ仕様	<ul style="list-style-type: none"> top/lidar/pointcloud (ROSメッセージ)
利用するアルゴリズム	<ul style="list-style-type: none"> なし

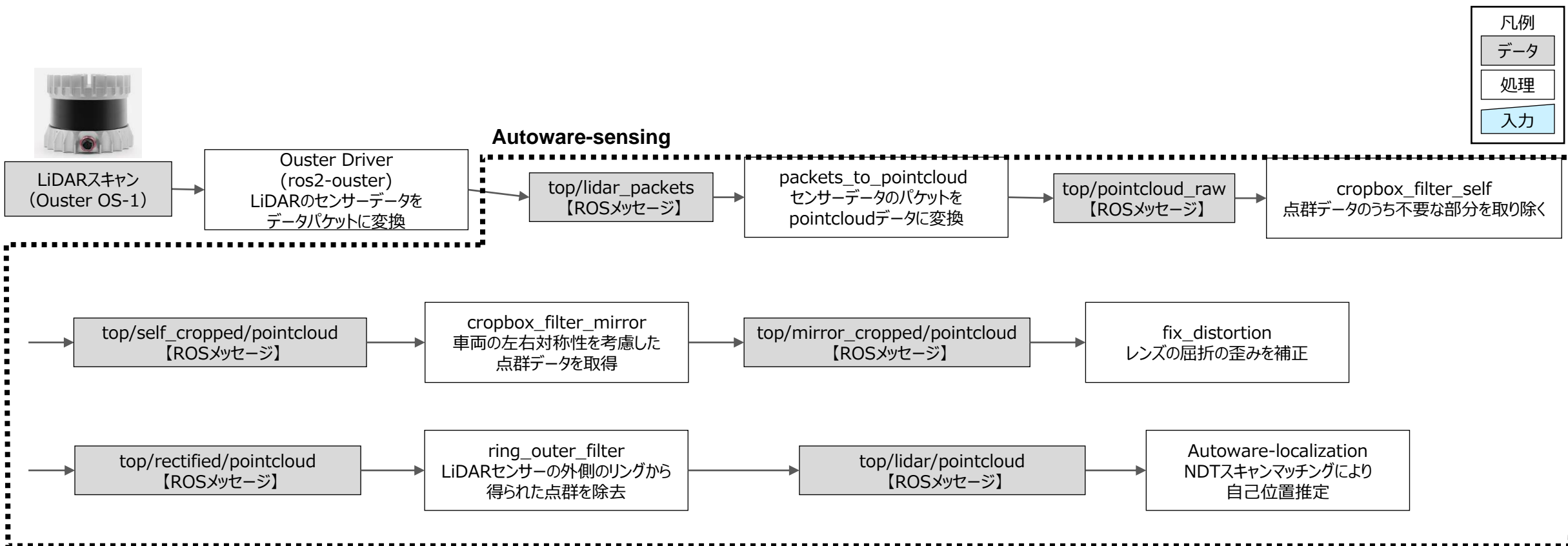


Ⅲ. 実証システム > 4. システム機能

④ センサーデータ処理 | 処理フロー

自動運転システムにおいて必要な様々なセンサーからのデータを処理するためのソフトウェアモジュール

Autoware-sensingでのLiDARデータの処理フロー



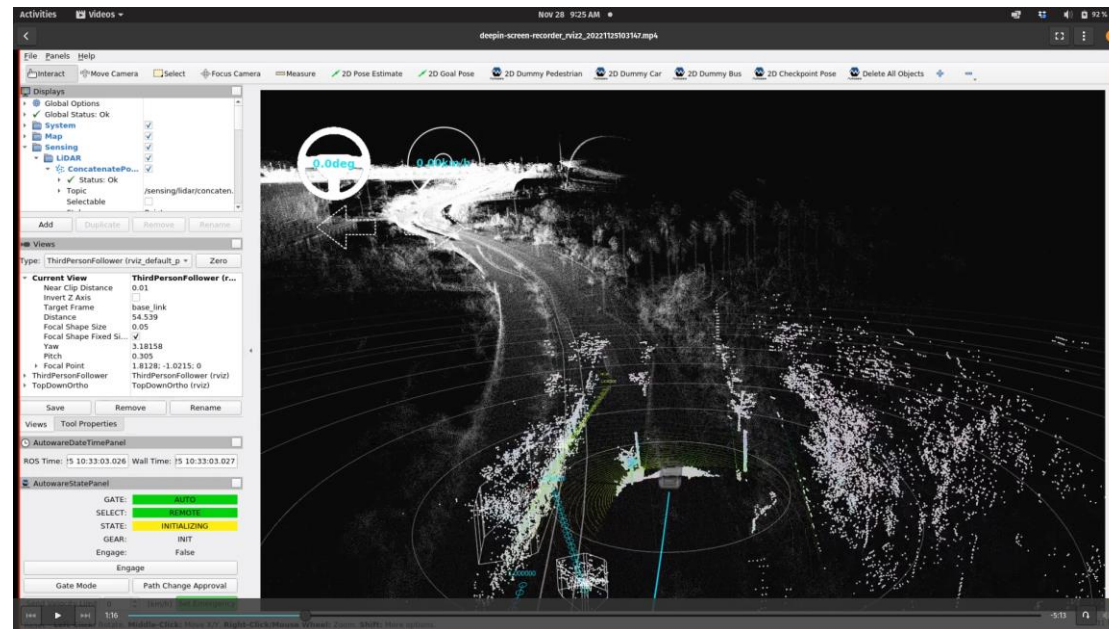
Ⅲ. 実証システム > 4. システム機能

⑤ 自己位置推定 (Autoware-localization)

概要

Autoware-localizationで点群マッチングを行なっている様子

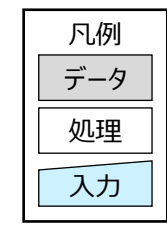
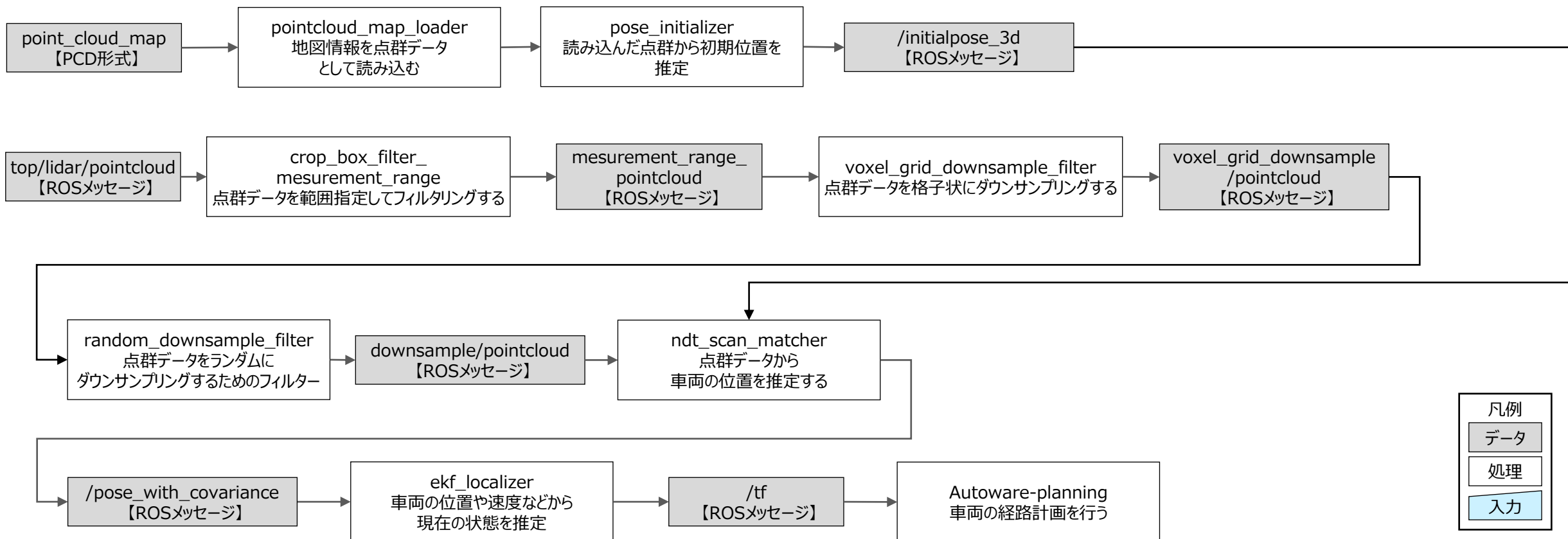
項目	詳細
機能名	<ul style="list-style-type: none"> 自己位置推定
機能概要	<ul style="list-style-type: none"> 地図情報から、車両が進むことのできる道路や交差点などの情報を取得する 取得したセンサーデータを元に、車両の位置を推定する
入力データ仕様	<ul style="list-style-type: none"> point_cloud_map (pcd形式) lidar/pointcloud (ROSメッセージ)
出力データ仕様	<ul style="list-style-type: none"> /tf (ROSメッセージ)
利用するアルゴリズム	<ul style="list-style-type: none"> NDTスキャンマッチング



Ⅲ. 実証システム > 4. システム機能

⑤ 自己位置推定 | 処理フロー

Autaware-localizationでの処理フロー



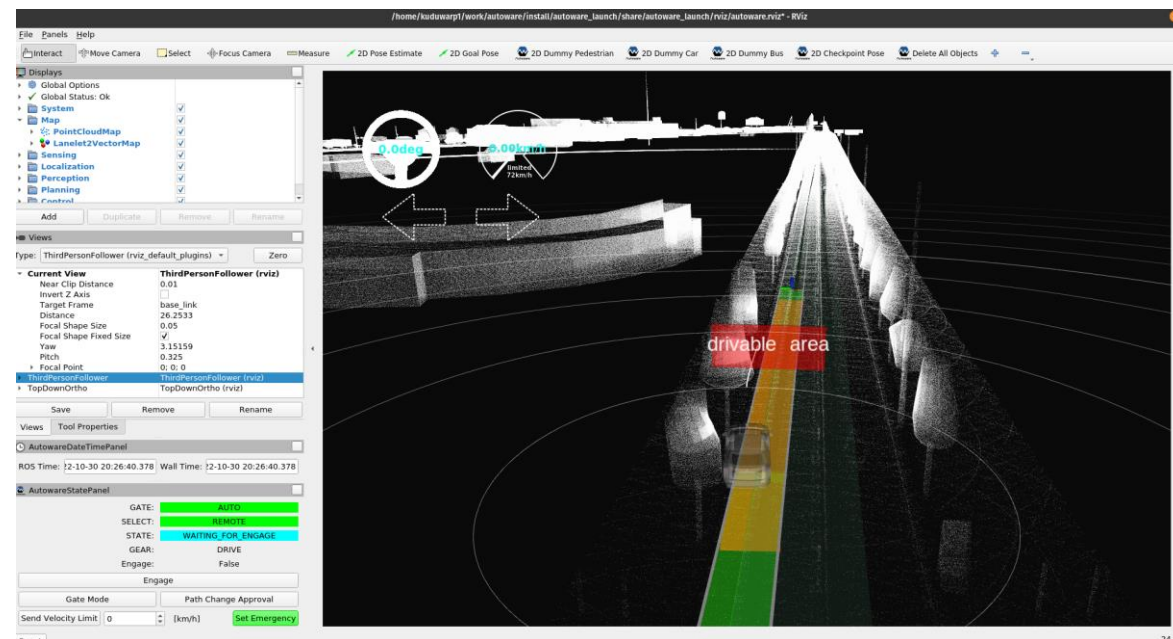
Ⅲ. 実証システム > 4. システム機能

⑥ 経路計画 (Autoware-planning)

概要

項目	詳細
機能名	<ul style="list-style-type: none"> 経路計画
機能概要	<ul style="list-style-type: none"> 自動運転車両の現在位置や周囲の環境情報から、次の移動目標地点を決定する 車両の動作に応じた経路計画を生成する 車両の速度や加速度、制動力などの動的特性を考慮した制御指令を生成する
入力データ仕様	<ul style="list-style-type: none"> /goal/pose (ROSメッセージ)
出力データ仕様	<ul style="list-style-type: none"> /scenario_planning/trajectory (ROSメッセージ)
利用するアルゴリズム	<ul style="list-style-type: none"> なし

Autoware-planningにより、動作経路を計画



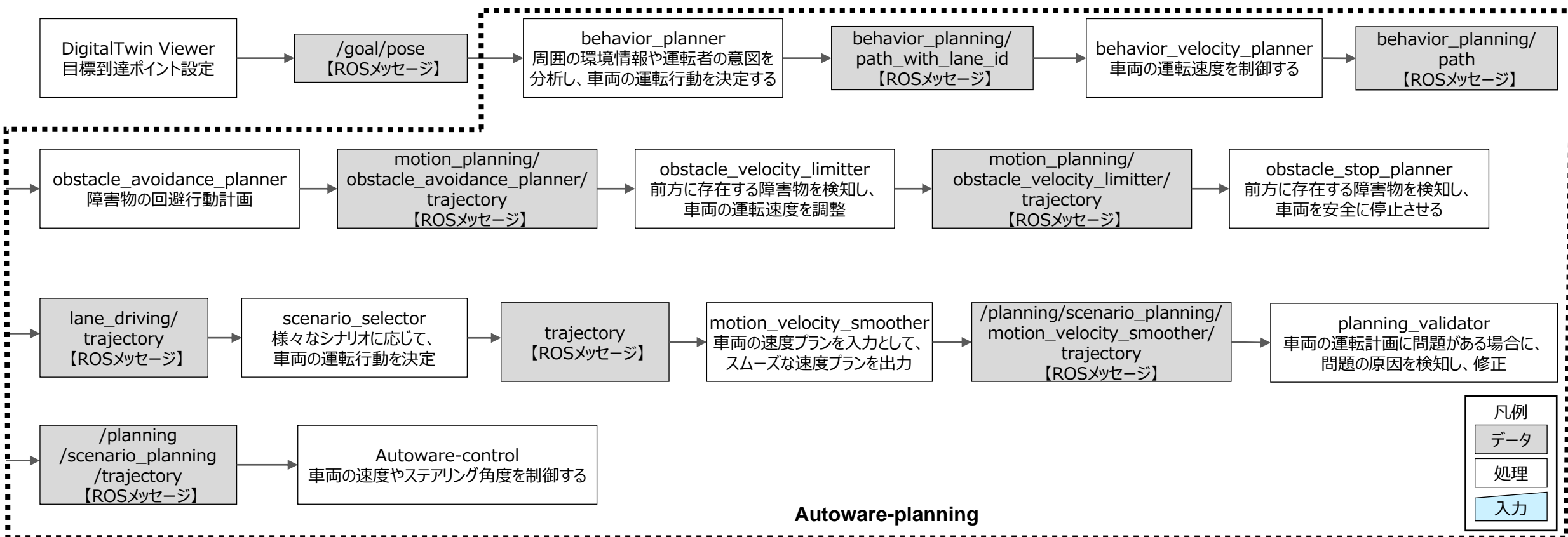


Ⅲ. 実証システム > 4. システム機能

⑥ 経路計画 | 処理フロー

自動運転車が自己位置推定や障害物検知を基に、安全かつ効率的に目的地に到達するためのルートプランニングを行うためのソフトウェアモジュール

Autaware-planningでの処理フロー



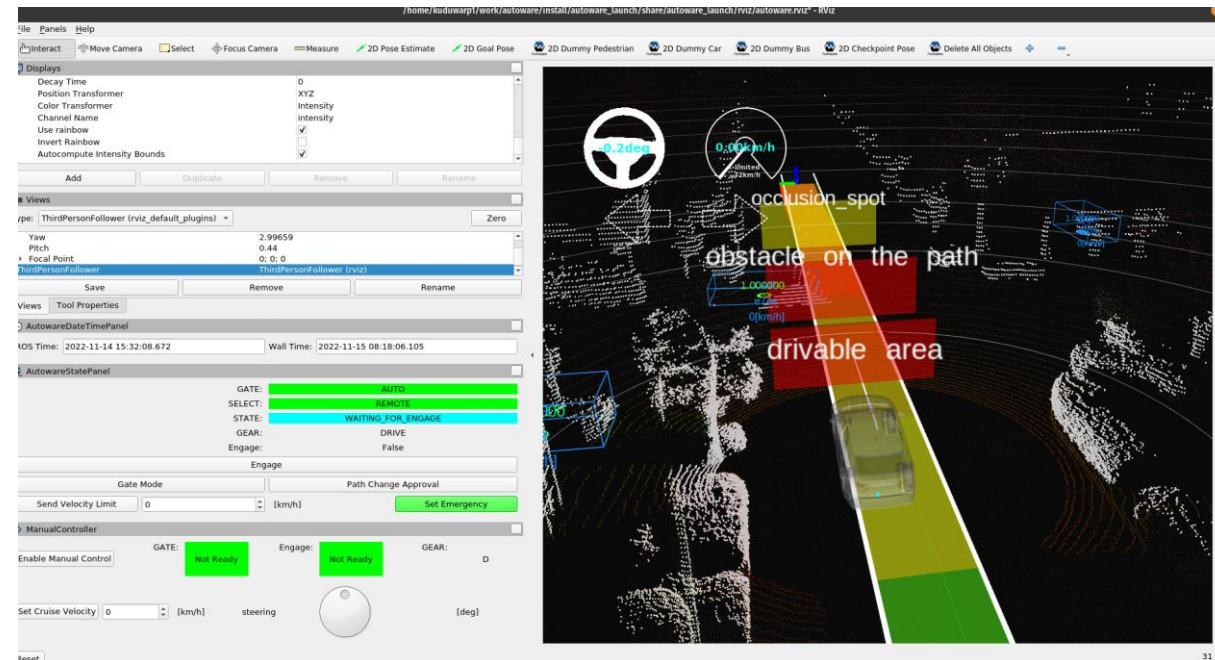
Ⅲ. 実証システム > 4. システム機能

⑦ 動作計画 (Autoware-control)

概要

Autoware-controlにより車両の速度、ステアリング角の動作を決定

項目	詳細
機能名	<ul style="list-style-type: none"> 動作計画
機能概要	<ul style="list-style-type: none"> 自動運転車両の動作状態をモニタリングする 車両の速度や加速度、制動力などの動的特性を制御する 車両の制御に必要な制御情報を生成する
入力データ仕様	<ul style="list-style-type: none"> /scenario_planning/trajectory (ROSメッセージ)
出力データ仕様	<ul style="list-style-type: none"> vehicle_cmd_gate (ROSメッセージ)
利用するアルゴリズム	<ul style="list-style-type: none"> なし

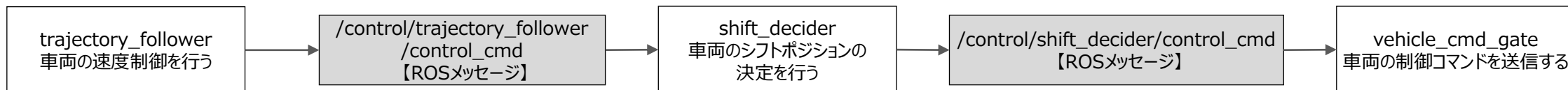


Ⅲ. 実証システム > 4. システム機能

⑦ 動作計画 | 処理フロー

自動運転車の動作計画を行うためのソフトウェアモジュール

Autoware-controlでの制御フロー



Ⅲ. 実証システム > 4. システム機能

⑧ 車体制御 (ROS2)

概要

項目	詳細
機能名	<ul style="list-style-type: none"> • 車体制御
機能概要	<ul style="list-style-type: none"> • Autowareから計算される動作指示を、搬送車両に伝える
入力データ仕様	<ul style="list-style-type: none"> • vehicle_cmd_gate (ROSメッセージ)
出力データ仕様	<ul style="list-style-type: none"> • CANBUSコマンド
利用するアルゴリズム	<ul style="list-style-type: none"> • なし

車両のCANBUSコマンドをゲームコントローラーで確認している様子

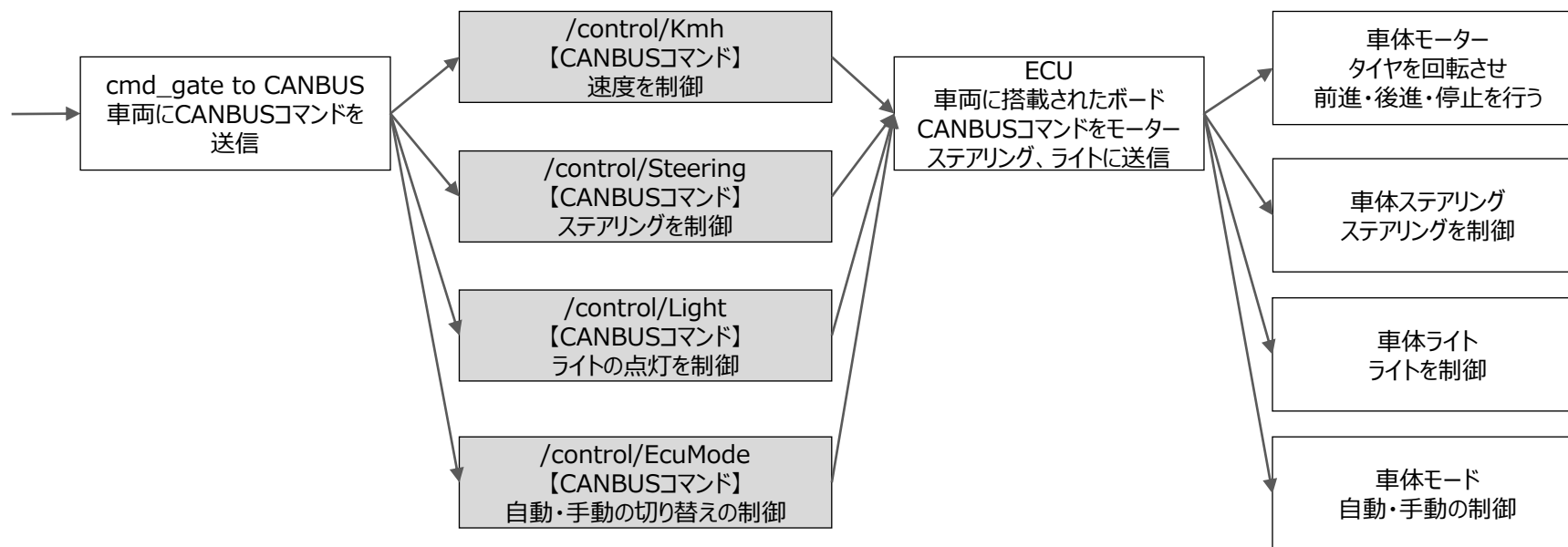


Ⅲ. 実証システム > 4. システム機能

⑧ 車体制御 | 制御フロー

自動運転車の走行制御を行うためのソフトウェアモジュール

車体制御フロー



Ⅲ. 実証システム > 4. システム機能

⑨データ送受信 (MQTT Broker)

概要

項目	詳細
機能名	<ul style="list-style-type: none"> データ送受信
機能概要	<ul style="list-style-type: none"> IoT (Internet of Things) やM2M (Machine-to-Machine) などの分野で広く使用されている、軽量なメッセージングプロトコル MQTTのClientは、Brokerにメッセージを送信し、MQTTのBrokerは、メッセージを受信して適切なクライアントに配信するサーバーの役割を持つ
入力データ仕様	<ul style="list-style-type: none"> トピック (文字列) メッセージ (JSON形式)
出力データ仕様	<ul style="list-style-type: none"> トピック (文字列) メッセージ (JSON形式)
利用するアルゴリズム	<ul style="list-style-type: none"> なし
利用したライブラリ	<ul style="list-style-type: none"> mqtt_bridge (ROS MQTTクライアント) MQTTNet (Unity MQTTクライアント)

車両の情報をMQTT Brokerから受信したデータ例

```
{ topic: 'fix' }
{
  message: '{"header": {"stamp": {"sec": 1670380518, "nanosec": 73145291},
"frame_id": "gnss_link"}, "status": {"status": 2, "service": 1}, "latitude":
34.664093253333334, "longitude": 135.39285443, "altitude": 44.397999999999996,
"position_covariance": [0.00012544000000000005, 0.0, 0.0, 0.0,
0.00012544000000000005, 0.0, 0.0, 0.0, 0.00200704000000000007],
"position_covariance_type": 1}'
}
{ topic: '/vehicle/status/steering_status' }
{
  message: '{"stamp": {"sec": 1670380531, "nanosec": 151323427},
"steering_tire_angle": 0.015446163713932037}'
}
{ topic: '/vehicle/status/steering_status' }
{
  message: '{"stamp": {"sec": 1670380531, "nanosec": 201267592},
"steering_tire_angle": 0.01424188632518053}'
}
```

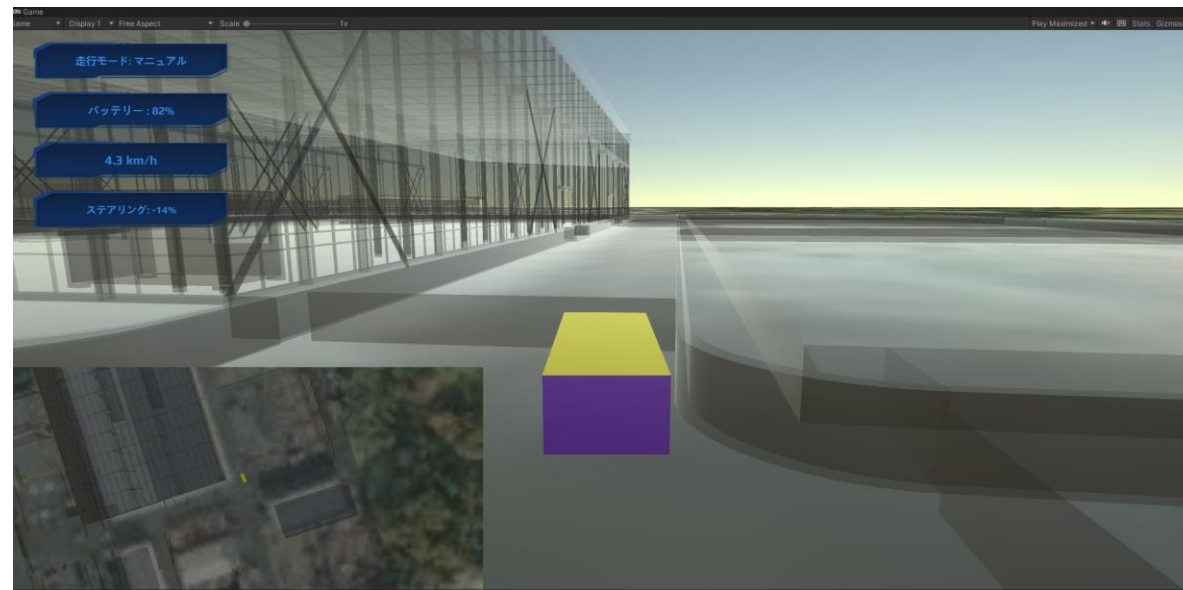
Ⅲ. 実証システム > 4. システム機能

⑨ データ集約 (MQTT Client)

概要

項目	詳細
機能名	<ul style="list-style-type: none"> データ集約
機能概要	<ul style="list-style-type: none"> MQTTクライアントは、IoTやM2Mなどの分野で広く使用される、軽量で効率的なメッセージングプロトコルであるMQTTのクライアントアプリケーション MQTTクライアントは、MQTTブローカーに接続してメッセージを送受信が可能 また、トピックを使用してメッセージのルーティングを行うことが可能
入力データ仕様	<ul style="list-style-type: none"> トピック (文字列) メッセージ (JSON形式)
出力データ仕様	<ul style="list-style-type: none"> トピック (文字列) メッセージ (JSON形式)
利用するアルゴリズム	<ul style="list-style-type: none"> なし

MQTT Clientが受信した情報を表示する開発画面



Ⅲ. 実証システム > 4. システム機能

⑨ データ集約 (MQTT Client) | データフォーマット

デジタルツインビューから搬送車両への指示データ

活用データ	送信されるトピック	ROS2データ型
制限速度	/planning/scenario_planning/max_velocity_default	tier4_planning_msgs.msg:VelocityLimit
自動運転開始	/autoware/engage	autoware_auto_vehicle_msgs.msg:Engage
初期位置設定	/initialpose	geometry_msgs.msg:PoseWithCovarianceStamped
目標位置設定	/planning/mission_planning/goal	geometry_msgs.msg:PoseStamped

搬送車両からのデジタルツインビューへの通信データ

活用データ	送信されるトピック	ROS2データ型
RTK測位データ	/sensing/gnss/fix	sensor_msgs.msg:NavSatFix
自己位置推定精度 (TP値)	/localization/pose_estimator/transform_probability	tier4_debug_msgs.msg:Float32Stamped
運転モード	/kentora/status/EcuMode	std_msgs.msg:Int8
速度	/vehicle/status/velocity_status	autoware_auto_vehicle_msgs.msg:VelocityReport
ステアリング	/vehicle/status/steering_status	autoware_auto_vehicle_msgs.msg:SteeringReport
バッテリー	/kentora/status/BatteryLevel	std_msgs.msg:Int8

Ⅲ. 実証システム > 4. システム機能

⑪ 車両情報の可視化 | Digital Twin Viewer (1/2)

概要

車両の情報を表示

項目	詳細
機能名	<ul style="list-style-type: none"> 車両情報の可視化
機能概要	<ul style="list-style-type: none"> 車両情報の表示 経路計画、経路情報の送信 車両位置情報の表示 (2D、3D)
入力データ仕様	<ul style="list-style-type: none"> NMEAデータ (JSON形式) 車両データ (JSON形式)
出力データ仕様	<ul style="list-style-type: none"> 経路情報 (ROSメッセージ)
利用するライブラリ	<ul style="list-style-type: none"> MQTTNet
利用するアルゴリズム	<ul style="list-style-type: none"> なし



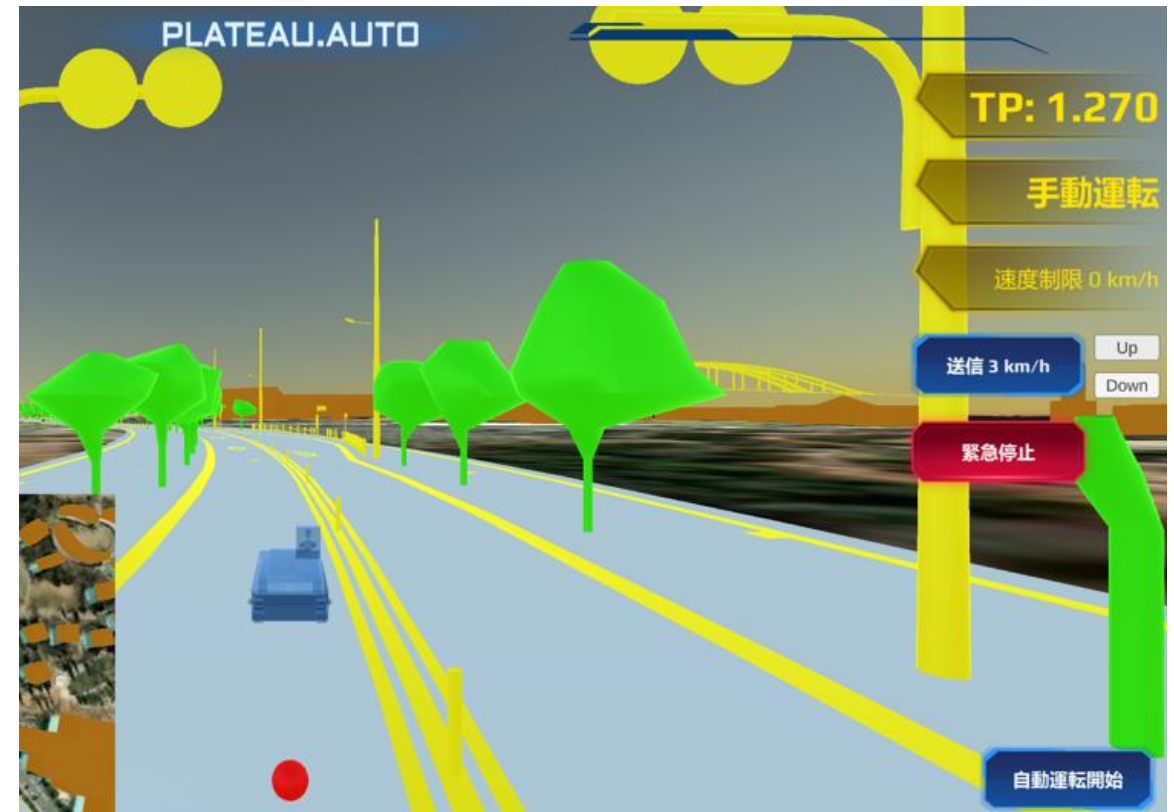
Ⅲ. 実証システム > 4. システム機能

⑪ 車両情報の可視化 | Digital Twin Viewer (2/2)

概要

イメージ

項目	詳細
機能名	● 車両制御
機能概要	● 車両の制限速度を設定 ● 車両の緊急停止 ● 車両の自動運転開始
入力データ仕様	● なし
出力データ仕様	● 速度指令 (ROSメッセージ) ● 停止指令 (ROSメッセージ) ● 自動運転開始指令 (ROSメッセージ)
利用するライブラリ	● MQTTNet
利用するアルゴリズム	● なし



Ⅲ. 実証システム > 4. システム機能

⑫ 経路計画 (Digital Twin Viewer)

概要

走行時の上空ビュー ポイントをクリックして経路情報を送信

項目	詳細
機能名	● 経路計画
機能概要	● 自動運転の経路を計画 ● 経路を送信
入力データ仕様	● なし
出力データ仕様	● 経路情報 (ROSメッセージ)
利用するライブラリ	● MQTTNet
利用するアルゴリズム	● なし



Ⅲ. 実証システム > 4. システム機能

⑭ データ送受信 (MQTT Broker) | メッセージ仕様

メッセージ仕様

項目	詳細
メッセージに含まれる項目	<ul style="list-style-type: none"> トピック メッセージ本体 メッセージID

メッセージデータ (JSON形式) のサンプル

```
'{"header": {"stamp": {"sec": 1670380703, "nanosec": 474754963},
"frame_id": "gnss_link"}, "status": {"status": 2, "service": 1}, "latitude":
34.664332163333334, "longitude": 135.39580569666666, "altitude":
43.863, "position_covariance": [0.00013459999999999999, 0.0, 0.0,
0.0, 0.00013459999999999999, 0.0, 0.0, 0.0,
0.00215295999999999998], "position_covariance_type": 1}'
```

```
'{"stamp": {"sec": 1670380719, "nanosec": 152854170},
"steering_tire_angle": 0.018587756901979446}'
```

Ⅲ. 実証システム > 5. アルゴリズム

NDTスキャンマッチングとは

NDTスキャンマッチングの特徴（ICPスキャンマッチングと比較した際の優位性・適合性）

	本ユースケースで利用した技術 NDT*スキャンマッチング	比較対象：ICPスキャンマッチング
アルゴリズム概要	<ul style="list-style-type: none"> 環境地図をボクセルごとに区切り正規分布を計算し、3次元LiDARから得られた点群データとマッチングを行う手法 	<ul style="list-style-type: none"> 地図データ郡群とセンサデータ群の2つの点群間の最近傍点を対応点とし、対応関係にある点間の距離計が最短になる箇所をマッチングポイントとする方法
計算速度	<ul style="list-style-type: none"> 高速（点群データが多い場合でも有用）： 点群データを正規分布に基づいたヒストグラムに変換することで、高速な位置姿勢推定を実現でき、リアルタイム性の要求が高いロボットや自動運転車などのアプリケーションに適任 	<ul style="list-style-type: none"> 高速 （但し、点群データが多いと低速になる）
ノイズ	<ul style="list-style-type: none"> 強い： 点群データに含まれるノイズや不良点に対しても比較的頑健な処理を行うことが可能なため、実際のセンサーから取得した点群データにも適用可能 	<ul style="list-style-type: none"> 弱い （不良点があると計算時間が長くなる）
計算量	<ul style="list-style-type: none"> 多い： 正規分布を用いたフィッティングにより大域的な最適解を見つけ易く、異なる観測角度から得られる点群データのマッチングにも有効 	<ul style="list-style-type: none"> 点と点のマッチングのため、少ない （但し、点群量が増えたり不良点が増えると計算量が増大）
精度	<ul style="list-style-type: none"> 高い（但し、初期値に依存）： 地図データと比較して、点群データのマッチングの誤差を減らすことができるため、高精度な位置姿勢推定を実現可能 	<ul style="list-style-type: none"> 高い （但し、不良点があると下がる）

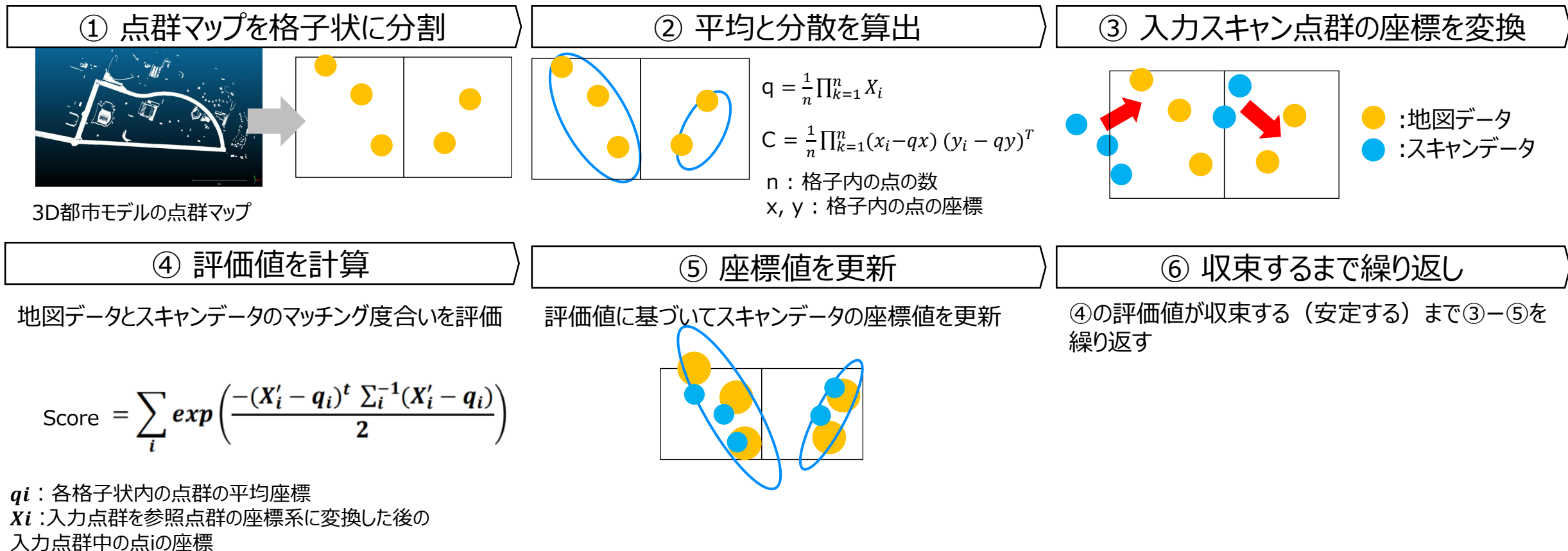
*Normal Distribution Transformの略称

Ⅲ. 実証システム > 5. アルゴリズム

NDTスキャンマッチングのフロー

点と分散のマッチングを行うNDTスキャンマッチングにより、自己位置推定を行う。

Autaware-localizationでの自己位置推定方法：NDTスキャンマッチング詳細



Ⅲ. 実証システム > 5. アルゴリズム

NDTスキャンマッチングとT.P.値

NDTスキャンマッチングとTransformation Probabilityの関係性

- Transformation Probabilityとは
 - NDTスキャンマッチングでは、Transformation Probability（変換確率）値（T.P.値）が重要な役割を果たす。この確率は、スキャンデータと地図データの間に対応関係の評価し、最適な変換パラメータを見つけるために使用される
- NDTスキャンマッチングとTransformation Probabilityの関係性
 - NDTスキャンマッチングでは、参照データと入力データが与えられたとき、Transformation Probabilityが最大となるような変換行列を求める。これにより、車両の現在位置や地図作成に役立つ正確な位置合わせが実現される
- 最適なパラメータを求めるために
 - NDTスキャンマッチングアルゴリズムでは、誤差関数を最小化することで、最適な変換パラメータを求める。その結果、Transformation Probabilityが最大化された場合に、スキャンデータと地図データの間に対応関係が最も良好となる。このアプローチにより、ロボットやセンサーの位置推定が改善され、より正確なナビゲーションや地図作成が可能になる

Transformation Probabilityに関連する主要な要素

項目	説明
Transformation	点群データ間の相対的な位置と姿勢変換を表す行列。これにより、点群データの位置合わせが行われる
Probability	点群データ間の位置合わせの精度を表す尺度。値が高いほど、位置合わせが正確であると評価される
NDTセル	3D点群データを、局所的なガウス分布で近似するために使用される空間区分。各セルは、平均座標と共分散行列を持つ
局所ガウス分布	点群データを近似するために使用される確率分布。各NDTセルは、平均座標と共分散行列を持つ3Dガウス分布で表される
点群データ	3D環境を表す点の集合。これは、センサー（例：LIDAR）から取得され、NDTスキャンマッチングに使用される
参照データ	位置合わせのベースとなる点群データ。通常、事前に作成された地図や過去のスキャンデータが使用される

Ⅲ. 実証システム > 5. アルゴリズム

NDTスキャンマッチングのパラメーター (1/2)

項目	説明	特徴	設定値	型
trans_epsilon	<ul style="list-style-type: none"> 自己位置推定において現在の姿勢と前回の姿勢の差分を示すパラメーター 	<ul style="list-style-type: none"> trans_epsilonの値は、車両の速度やセンサーの精度、環境条件などによって異なる より速度が速い車両では、より小さい値が必要となる場合がある また、センサーのノイズが多い場合には、より大きな値を設定することが推奨される 	0.01	double型
step_size	<ul style="list-style-type: none"> ロバストな自己位置推定のためのステップサイズを設定するためのパラメーター 	<ul style="list-style-type: none"> ステップサイズは、車両の速度や環境条件によって異なる より高速な車両では、より大きなステップサイズが必要だが、環境が複雑である場合には、より小さなステップサイズが必要となる 	0.1	double型
resolution	<ul style="list-style-type: none"> 自己位置推定に使用される地図の解像度を表すパラメーター 	<ul style="list-style-type: none"> 解像度が高いほど、地図が細かく分割され、自己位置推定の精度が向上する ただし、解像度が高すぎると、計算コストが高くなり、リアルタイムの自己位置推定が困難になる可能性がある 	2.0	double型
max_iterations	<ul style="list-style-type: none"> アルゴリズムが最大で何回まで反復処理を行うかを指定するパラメーター 	<ul style="list-style-type: none"> max_iterations パラメーターは、車両の速度や環境条件に応じて調整する必要がある 高速で移動する車両では、より少ない反復回数で収束する必要があるため、max_iterations を小さくすることが推奨される 	30	int型

Ⅲ. 実証システム > 5. アルゴリズム

NDTスキャンマッチングのパラメーター (2/2)

項目	説明	特徴	設定値	型
converged_param_transform_probability	<ul style="list-style-type: none"> 自己位置推定アルゴリズムが収束したかどうかを表す確率の閾値を指定するためのパラメーター 	<ul style="list-style-type: none"> このパラメーターは、推定された自己位置が正確である確率を表す 閾値を超える場合、アルゴリズムは収束したと判断され、推定された自己位置が使用される 一方、閾値を下回る場合、アルゴリズムは収束していないと判断され、推定された自己位置は破棄される 	3.0	double型
converged_param_nearest_voxel_transformation_likelihood	<ul style="list-style-type: none"> 自己位置推定アルゴリズムが収束したかどうかを表す、最も近いボクセルの変換確率の閾値を指定するためのパラメーター 	<ul style="list-style-type: none"> 閾値を高く設定しすぎると、推定された自己位置が不正確になる可能性がある 逆に、閾値を低く設定しすぎると、アルゴリズムが収束するまでの時間が長くなり、リアルタイムの自己位置推定が困難になる可能性がある 	2.3	double型
initial_estimate_particles_num	<ul style="list-style-type: none"> 自己位置推定アルゴリズムで使用されるパーティクルフィルターの初期状態のパーティクル数を指定するためのパラメーター 	<ul style="list-style-type: none"> 高速で移動する車両では、より多くのパーティクルが必要になる場合がある また、環境条件が複雑な場合には、より多くのパーティクルが必要になる場合がある しかし、パーティクルの数を増やしすぎると、計算コストが増加し、リアルタイムの自己位置推定が困難になる可能性がある 	100	int型

Ⅲ. 実証システム > 6. データ > ①活用データ 活用データ（3D都市モデル）一覧

地物	地物型	属性区分	属性名	内容
建築物LOD2	bldg:Building	空間属性	bldg:lod2Solid	建築物のLOD2の立体
道路LOD3	tran:Road		tran:lod3MultiSurface	道路のLOD3の面（標高をもつ面として表現）
都市設備LOD3	frn:CityFurniture		frn:lod3Geometry	都市設備のLOD3の立体
植生LOD3	veg:SolitaryVegetation Object		veg:lod3Geometry	植生のLOD3の立体

Ⅲ. 実証システム > 6. データ > ①活用データ 活用データ一覧：その他

活用データ	内容	データ形式	出所
3Dスキャン点群データ	LiDARセンサーにて取得	PointCloud	OUSTERで計測
高精度測位データ	GPS (RTK-GNSS) にて取得	NMEA data	Ichimillで計測
加速度データ	IMU (慣性計測ユニット) にて取得	Sensor data	WitMotion WT901Cで計測

Ⅲ. 実証システム > 6. データ > ①活用データ

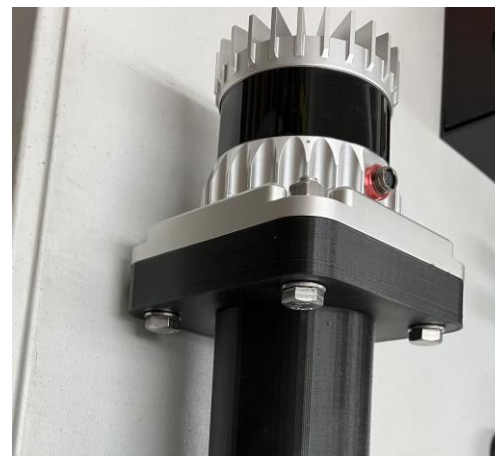
3Dスキャン点群データ

LiDARセンサーとしてOuster OS-1を利用し、点群データを取得した

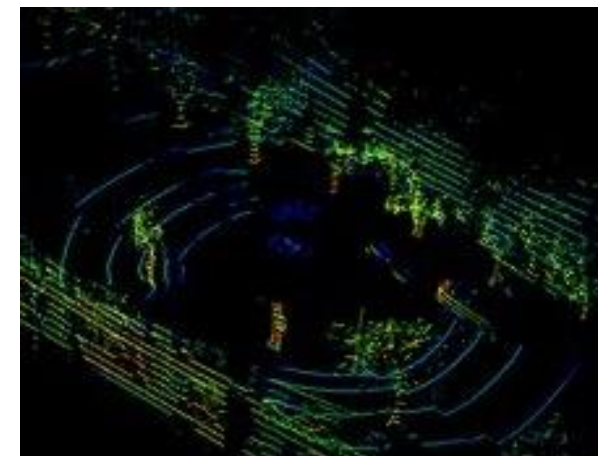
LiDARセンサーの概要

項目	スペック
製品名	Ouster OS-1
チャンネル数 (垂直)	64 channels
解像度 (水平)	512, 1024, or 2048 ※本検証では1024を利用
距離	120m
視野角 (垂直)	45° (±22.5°)
角度分解能 (垂直)	0.35° - 2.8°
精度	±1.5 - 5 cm
速度 (点群の取得数/秒)	1,310,720
回転速度	10 or 20 Hz ※本検証では20Hzを利用
電源	14 - 20 W
質量	455g
IPコード (防水防塵)	IP68, IP69K

イメージ



Ouster OS-1



Ouster OS-1から出力される点群

Ⅲ. 実証システム > 6. データ > ①活用データ

RTK-GNSSおよびIMU

測位手段としてRTK、搬送車両の姿勢・移動の検出にIMUを利用した

RTK-GNSS測位センサーの概要

項目	スペック
概要	<ul style="list-style-type: none"> 誤差数センチメートルの測位を可能とするRTK測位（固定局・移動局併用での測位）センサー
製品名	Ichimill
対応衛星	【2周波対応】 <ul style="list-style-type: none"> QZSS L1/L2、GPS L1/L2、GLONASS G1/G2、Galileo E1/E5、BeiDou B1/B2
サイズ	129 x 90 x 50mm

イメージ



Ⅲ. 実証システム > 6. データ > ①活用データ

加速度データ

測位手段としてRTK、搬送車両の姿勢・移動の検出にIMUを利用した

IMUセンサーの概要

イメージ

項目	スペック
概要	<ul style="list-style-type: none"> • 加速度計、ジャイロセンサーを含む慣性計測装置 • 3次元の姿勢や運動量を検出することが可能
製品名	WitMotion WT901C
センシング項目	<ul style="list-style-type: none"> • 3軸加速度 • 角度、角速度 • 磁場
サイズ	77 x 75 x 22 mm



Ⅲ. 実証システム > 6. データ > ②データ処理 データ処理一覧

システムに入力するデータ (データ形式)	用途	処理内容	データ処理ソフトウェア	活用データ (データ形式)
3D都市モデルデータ (STL形式)	3Dモデル変換	<ul style="list-style-type: none"> CityGML形式の3DモデルからSTL形式へ変換 	FZKViewer	対象地域の3D都市モデルデータ (CityGML形式)
3D都市モデルデータ (OBJ形式)	3Dモデル変換	<ul style="list-style-type: none"> STL形式の3DモデルからOBJ形式へ変換 	Autodesk Fusion	対象地域の3D都市モデルデータ (STL形式)
点群マップ (PCD形式)	点群マップの生成	<ul style="list-style-type: none"> 仮想空間内で仮想LiDARによる点群取得を行い、点群マップを出力 	PLATEAU Autoware Converter	対象地域の3D都市モデルデータ (OBJ形式)
位置情報 (ROSメッセージ)	自己位置推定	<ul style="list-style-type: none"> 点群マッチングによる自己位置推定のマップとして利用 	自己位置推定 (Autoware-localization)	3D都市モデルから生成された点群マップ (PCD形式)
センサー情報 (ROSメッセージ)	自己位置推定	<ul style="list-style-type: none"> 点群マッチングによる入力点群として利用 	センサーデータ処理 (Autoware-sensing)	車両に取り付けたLiDARの点群データ (PCD形式)
車両情報 (JSON形式)	遠隔監視	<ul style="list-style-type: none"> 車両の状態をJSON化して送信 	データ送受信 (MQTT Broker)	車両から得られるデータ (JSON形式)

Ⅲ. 実証システム > 6. データ > ②データ処理 3D都市モデルの変換 | FZK Viewer

変換処理のフロー

データ読み込み

- 対象となるCityGML形式のデータを読み込み

データ選択

- 読み込んだCityGMLデータから、対象の地域を選択

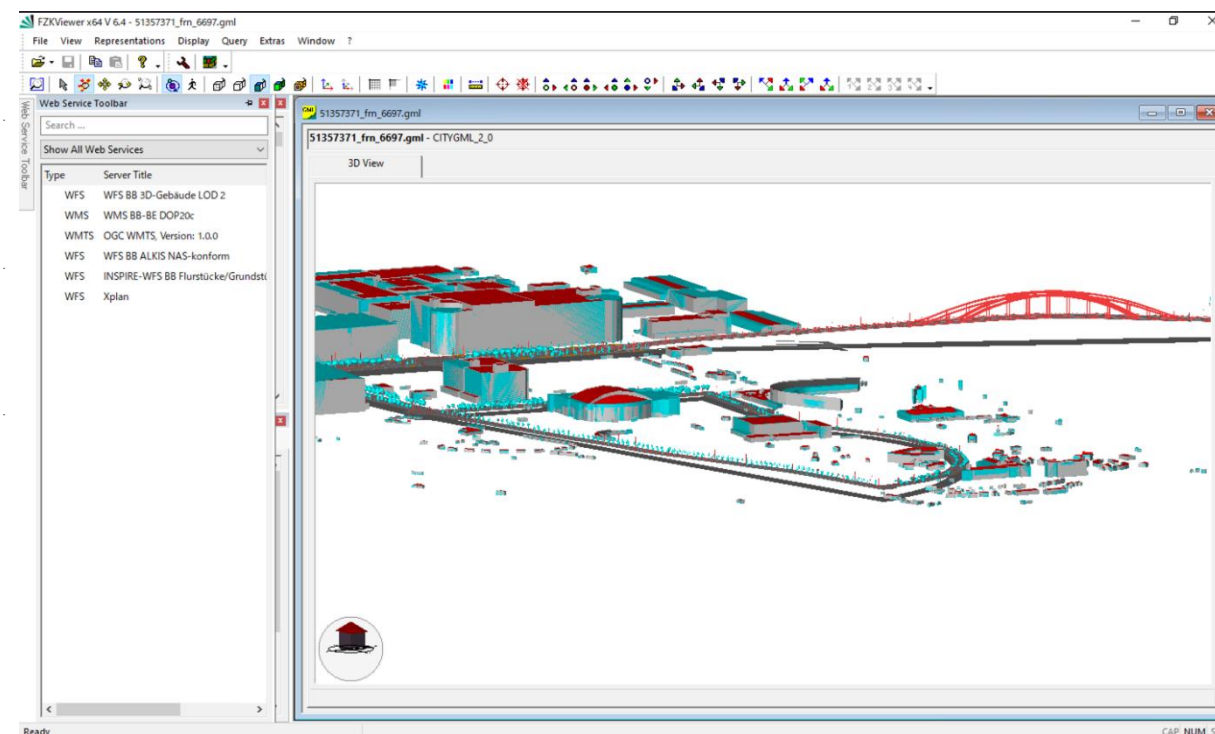
データ変換

- 出力先フォルダを選択、データ変換を行う

データ出力

- 指定フォルダにSTL形式のデータが出力される

変換後のイメージ

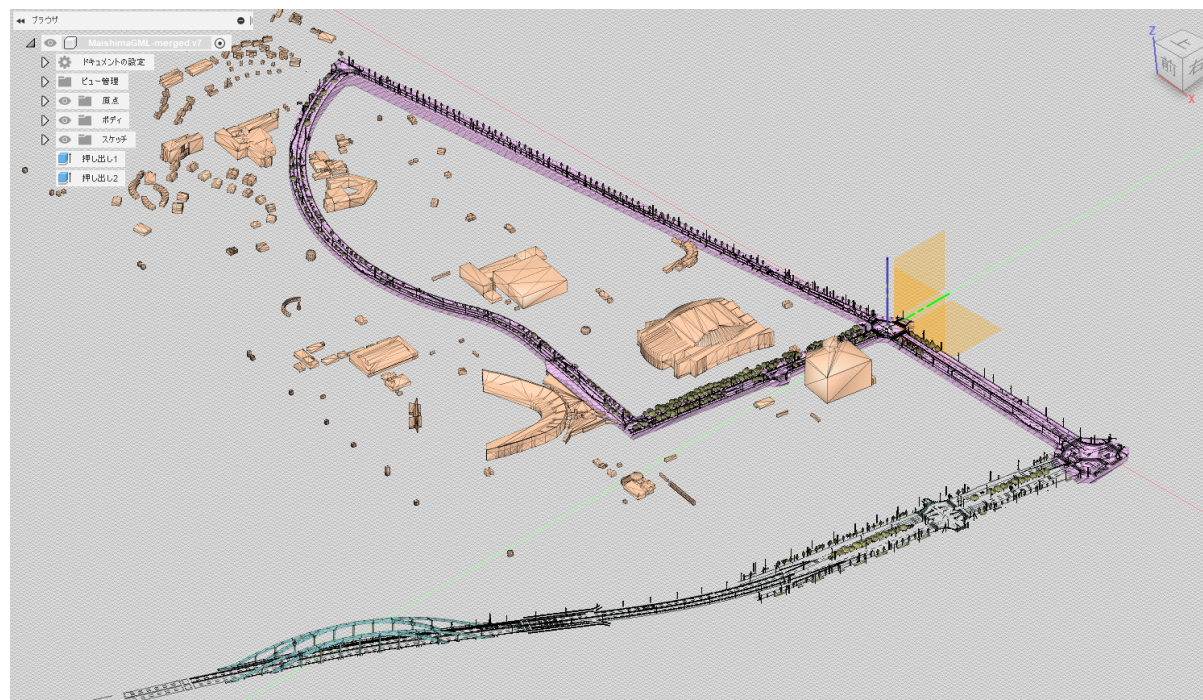


Ⅲ. 実証システム > 6. データ > ②データ処理 3D都市モデルの変換 | Autodesk Fusion

変換処理のフロー



変換後のイメージ



Ⅲ. 実証システム > 6. データ > ③出力データ 出力データ一覧

出力データ	内容	データ形式
走行実行データ (CANBUS形式)	<ul style="list-style-type: none">車両のステアリングやモーターを制御	搬送車両の走行実行データ (CANBUS)

Ⅲ. 実証システム > 6. データ > ③出力データ 走行実行データ (CANBUS形式)

走行実行データ

ゲームコントローラーでCANBUS信号を確認している様子

項目	詳細
名称	<ul style="list-style-type: none">CANBUS (Controller Area Network Bus)
概要	<ul style="list-style-type: none">自動車や産業用機械などの分野で使用されるシリアル通信プロトコルで、データを高速かつ信頼性の高い方法で転送することができる
データに含まれる項目	<ul style="list-style-type: none">ID (Identifier)データ (Data)コントロールフィールド (Control Field)CRC (Cyclic Redundancy Check エラーチェック)



Ⅲ. 実証システム > 7. ユーザインタフェース

Digital Twin Viewer | ユーザ操作画面

ユーザーはDigital Twin Viewer上でリアルタイムでのシミュレーション状況を確認しながら、制限速度やTP値等を変更させることができる

Digital Twin Viewerのユーザー操作画面



俯瞰ビュー（大画面）と上空ビュー（左下画面）で構成

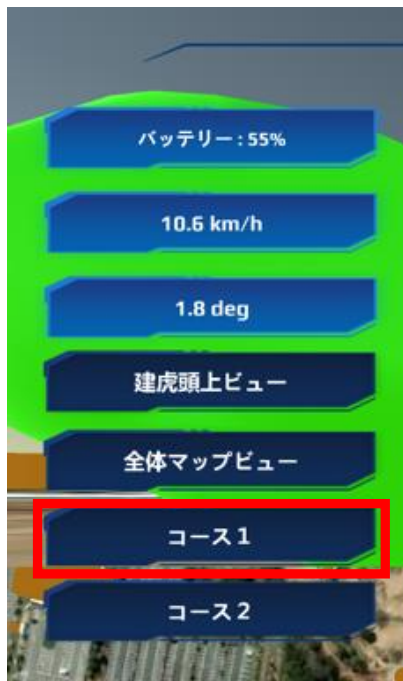
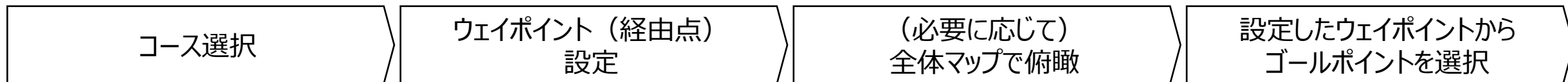
画面詳細と機能

バッテリー: 55% 10.6 km/h 1.8 deg	<ul style="list-style-type: none"> 現在の車両データ <ul style="list-style-type: none"> - バッテリー残量 - 走行速度 - ステアリング確度 	TP: 1.270	<ul style="list-style-type: none"> TP値（マッチング頻度）を示す
建虎頭上ビュー 全体マップビュー	<ul style="list-style-type: none"> ビューは建虎（車両）頭上と全体俯瞰の2つから切り替えが可能 	手動運転 速度制限 0 km/h	<ul style="list-style-type: none"> 手動は車両側の乗員が運転し、自動は本システムで運行
コース1 コース2	<ul style="list-style-type: none"> 俯瞰ビューをあらかじめ設定したそれぞれのコースにズームインする 	送信 3 km/h (Up/Down) 緊急停止	<ul style="list-style-type: none"> 制限速度は1km/h刻みで設定可能

Ⅲ. 実証システム > 7. ユーザインタフェース

Digital Twin Viewer | ウェイポイント設定

コースを選択することで、ウェイポイント（経由点）とゴールを設定が可能となる



あらかじめ設定したコースから
検証したいコースを選択する



2D地図を右クリックすると、
Waypointが追加される



ウェイポイントを左クリックし、
Goalを設定する

Ⅲ. 実証システム > 7. ユーザーインターフェース Warper Viewer

Warper ID選択

開始ボタンをクリック

映像受信

(必要に応じて)
マイク、スピーカー設定

Main Camera	kentora-1
Sub Camera	kentora-2

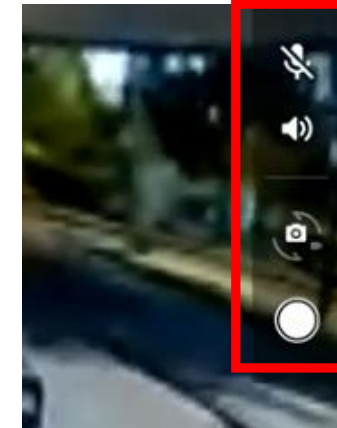
メインカメラ、
サブカメラを設定

キャンセル	開始する
-------	------

開始ボタンをクリック



リアルタイム映像を受信



マイク、スピーカーのミュート設定を行う

Ⅲ. 実証システム > 8. システムテスト結果

システムテスト結果一覧

試験項目	確認内容	結果
デジタルツインビューへ3D都市モデルの読み込み	対象エリアとなる舞洲の3D都市モデルをデジタルツインビューに読み込めたか	OK
Autowareの繋ぎ込み	Autowareによって搬送車両を制御できたか	OK
デジタルツインビューの表示確認	デジタルツインビュー上の表示内容が正しく車両と連動しているか • 走行モード：マニュアル or オート • バッテリー • 速度 • ステアリング • TP	OK
RTK-GNSSの位置測位	搬送車両に搭載したRTK-GNSSによって位置計測ができたか	OK
LiDAR事前計測点群を用いた自律走行	センシング用に搭載したLiDARセンサーを使って事前測量を行った点群を使って自律走行ができたか	OK
3D都市モデルを用いた自律走行	3D都市モデルから生成した点群を用いて自律走行ができたか	OK

I. 実証概要

II. 実証技術の概要

III. 実証システム

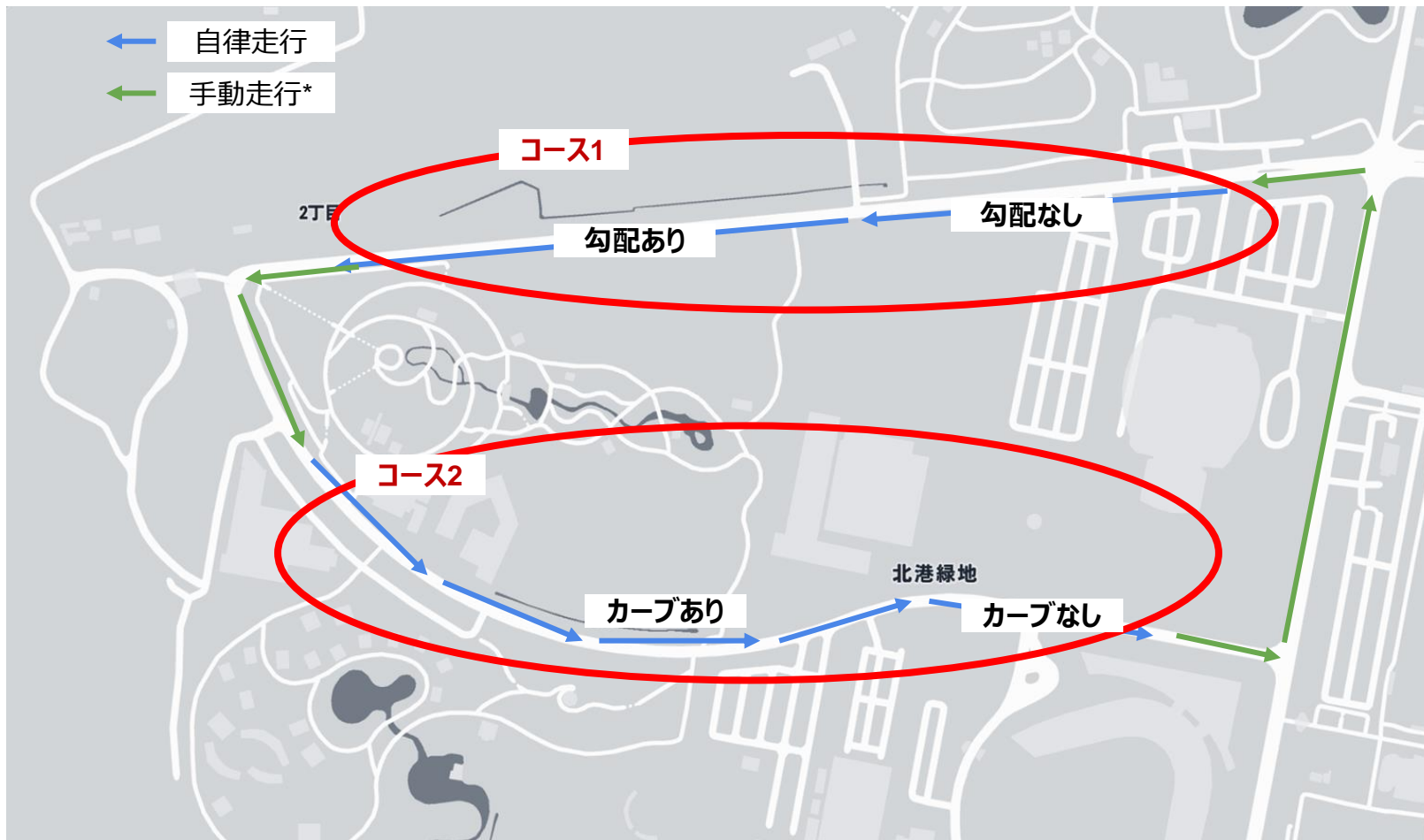
IV. 実証技術の検証

V. 成果と課題

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ① 検証内容 実証実験詳細

目的	事前取得 / 3D都市モデル由来 の点群マップを用いた搬送車両の自律走行
実施期間	事前取得の点群マップによる自律走行：2022年12月2日 10:00-12:00 3D都市モデル由来の点群マップによる自律走行：2022年12月8日 10:00-12:00
実施場所	遠隔監視室：ホテルロジック舞洲 2F会議室、車両走行ルート：舞洲地区
主な参加者	国土交通省・アクセンチュア・竹中工務店・アルモ・アダワープジャパン
実施内容	1. 事前取得した点群マップにより、コースを自律走行できるか検証（12月2日） 2. 3D都市モデル由来の点群マップにより、コースを自律走行できるか検証（12月8日）
事前取得点群マップの生成	1. 自己位置推定で利用するOuster OS1 64chを用いて、手動走行により点群マップを取得 2. 10km/h以下の速度で点群データを取得
気象条件	1. 事前取得マップ生成時 降水量0.0mm、平均気温 8.2度、最高気温12.0度、平均湿度56%、曇後晴 2. 3D都市モデル生成時 降水量0.0mm、平均気温11.5度、最高気温15.1度、平均湿度51%、晴後薄曇
評価方法	1. 事前取得マップを取得した際の天候状況と同じ状況で3D都市モデル由来の点群マップを用いた自律走行を行う 2. 事前取得点群マップ及び3D都市モデル由来の点群マップを用いたAutowareの自己位置推定から得られる X座標、Y座標、TP値を散布図、グラフにプロット 3. ルート整合率を算出してX座標、Y座標にプロットし、3D都市モデルの優位性を評価

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ① 検証内容 走行ルート



コース1：ストレートエリア
ゆるやかな坂が続く直線ルート



コース2：カーブエリア
ゆるやかなカーブが続くルート

*大阪府警よりご指摘頂き、一般車両のわき道を誘発するルートおよび交差点付近は手動走行

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ① 検証内容

取得点群マップ情報

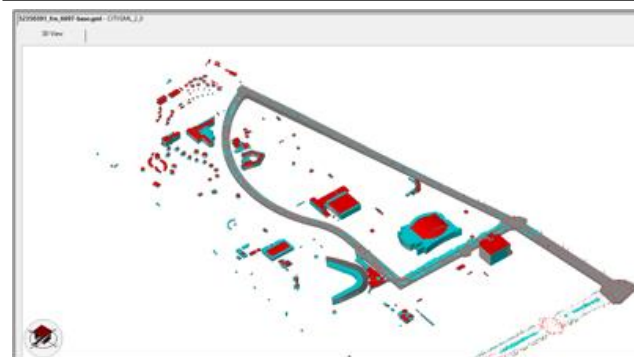
検証に先立って、3D都市モデル由来のデータと実際の道路上を事前走行して取得したデータのそれぞれから点群マップを作成

取得データの概要

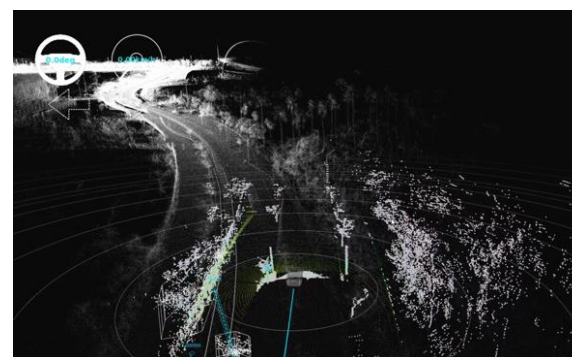
項目	3D都市モデル由来の点群マップ	事前走行から取得した点群マップ
総点群数	10,878,200	6,622,209
取得範囲	縦: 1250.41m 横: 800.64m 高さ: 90.80m	縦: 1078.76m 横: 661.604m 高さ: 71.12m
総点群密度*	11.966	13.046

*1m³に存在する点群の数 (点群数/m³)

イメージ図



舞洲エリアのCityGML (LOD2)



LiDAR (Ouster) で取得した舞洲エリアの点群データ



3D都市モデル由来の点群マップ
LOD2の建物がしっかりと表現される



事前走行による点群マップ
道路の周りの地物が多く取得される



IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ②検証結果 マッチング分析結果

精度検証の結果、3D都市モデル上に建物/植生データの不足しているエリアでは自己位置推定やマッチングに難が生じたものの、建物がメインのエリアでは自己位置推定精度が高く安定的に自律走行ができた

走行ルート	事前取得マップ		3D都市モデルマップ		要因
	マッチング可否	TP値	マッチング可否	TP値	
①ロジ舞洲前	○	3.5 ~ 4.0	×	0.5 ~ 1.0	<ul style="list-style-type: none"> 現実にある植生/建物の塀が3D都市モデルになく、マッチング精度が低く自己位置推定ができなかった。
②バッファローズ合宿所前	○	6.0 ~ 7.0	○	2.8 ~ 3.0	<ul style="list-style-type: none"> カーブエリアだが、両脇に建物があり、マッチング精度が高くなり、一番誤差の少ないエリアとなった。
③おおきにアリーナ舞洲前	○	6.0 ~ 7.0	○	2.8 ~ 3.2	<ul style="list-style-type: none"> 両脇に建物があり、その特徴点を見てマッチング精度が向上した。 路上駐車が多く、手動走行した。
④バッファローズスタジアム手前	○	6.0 ~ 7.0	×	1.4 ~ 1.9	<ul style="list-style-type: none"> 植え込みやスタジアムの柱が3D都市モデルになく、マッチング精度が減少した。
⑤バッファローズスタジアム通過後の坂道	○	6.0 ~ 7.0	○	2.8 ~ 3.2	<ul style="list-style-type: none"> 3D都市モデル上も現実にも建物がなく、マッチング精度が高くなったため自律走行ができた。



IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果

絶対位置精度分析結果

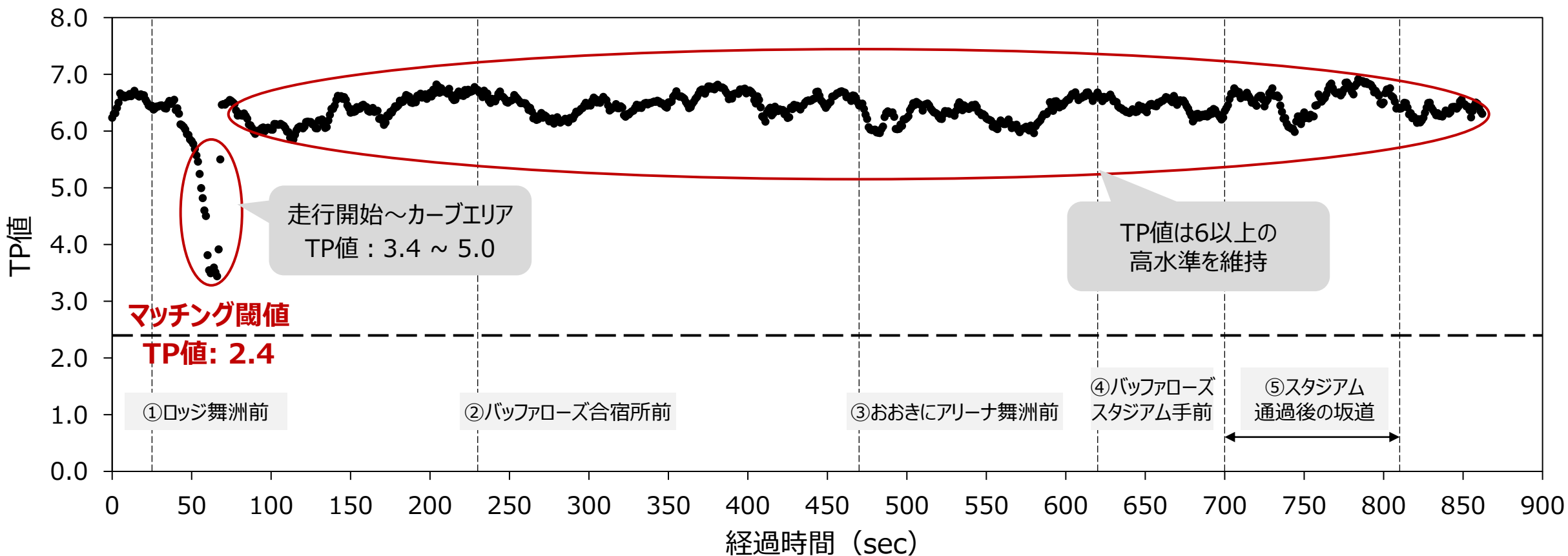
TP値が閾値を超える走行ルート上においてはRTKとの位置誤差が0.3m未満となっており、車幅1.66m、一般的な道路幅員3mを考えると十分に公道走行が可能と考えられる

走行ルート	マッチング可否	誤差	コメント
① ロッジ舞洲前	×	— 自己位置推定が合わず、測定不可	マッチングも不可で、誤差の計算のための座標も算出できず、自律走行は不可となった。
② バッファローズ合宿所前	○	0.19m	マッチングも自律走行も可能で誤差は0.19mとなり、今回の走行では一番誤差の少ないエリアとなった。
③ おおきにアリーナ舞洲前	○	— 手動走行のため、測定不可	マッチングは可能だったが、路上駐車が多く、手動にて走行した。
④ バッファローズスタジアム手前	×	0.88m	マッチングは減少したが、自律走行は可能で、誤差は0.88mとなり、車両の幅1.6mに近づいた。自動運転では一番誤差の開いたエリアとなった。
⑤ バッファローズスタジアム通過後の坂道	○	0.25m	マッチングは回復し、自律走行は可能であった。

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 マッチング分析：事前取得点群マップ

事前取得の点群マップを用いた自己位置推定の精度を検証すると、TP値は60～70秒時点では3.4と低いが、その後概ね6～7.0で推移している

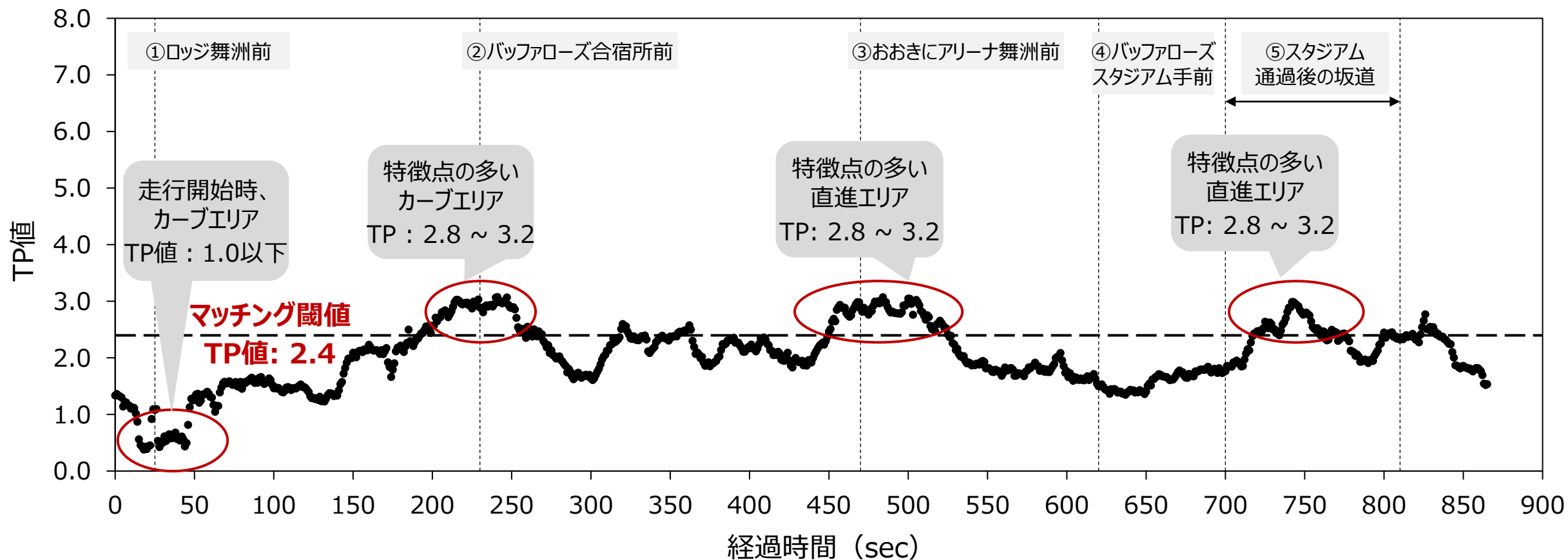
経過時間ごとのTP値のグラフ



IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 マッチング分析：3D都市モデル由来の点群マップ

3D都市モデル由来の点群マップを用いた自己位置推定の精度を検証すると、0~60秒付近でのTP値は1.0以下の値となっているが、その後は概ね1.6~3.0で推移している

経過時間ごとのTP値のグラフ

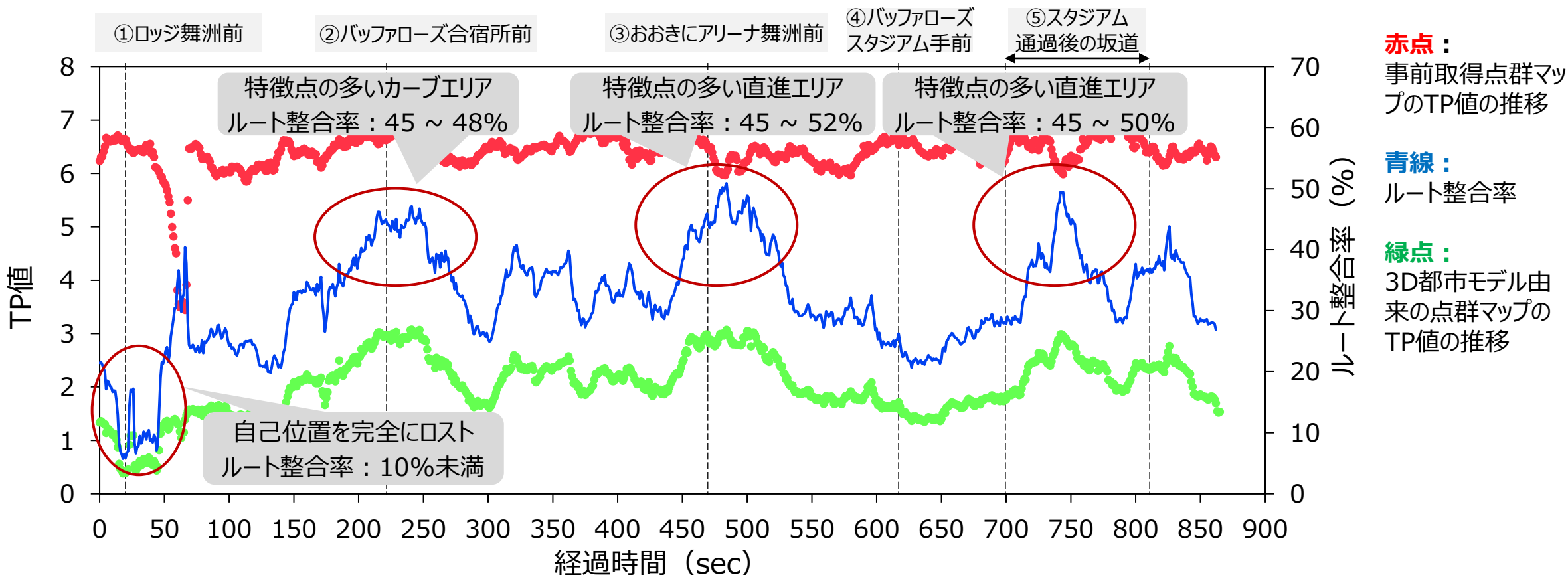


IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果

マッチング分析：マップ間の比較

事前取得点群マップのTP値及び3D都市モデル由来マップのTP値を散布図に重ね、折線グラフでルート整合率（ $TP_PLATEAU / TP_事前取得$ ）を示す

経過時間ごとのTP値のグラフ

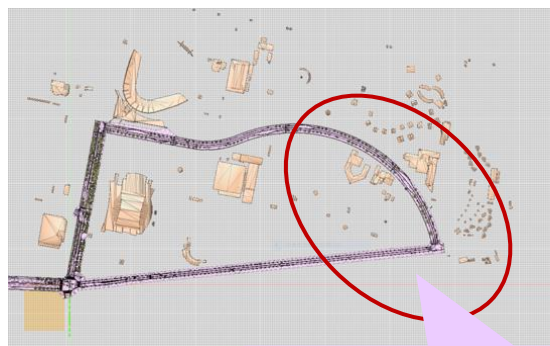


IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果

ルート上の特徴点 ① ロッジ舞洲前

3D都市モデル上に建物や植生データが再現されておらず、現地写真との乖離が大きい。そのため自己位置推定ができず自律走行ができなかった

測定ポイント



ロッジ舞洲前

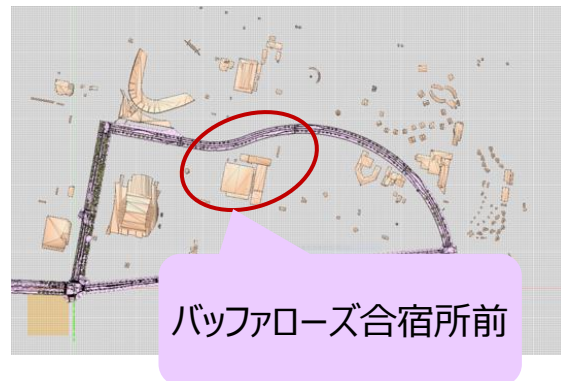
3D都市モデルと現地写真の比較



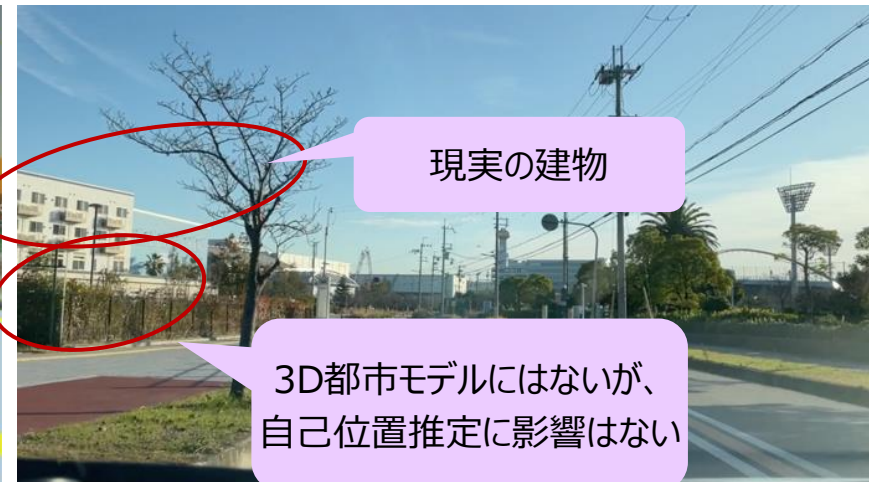
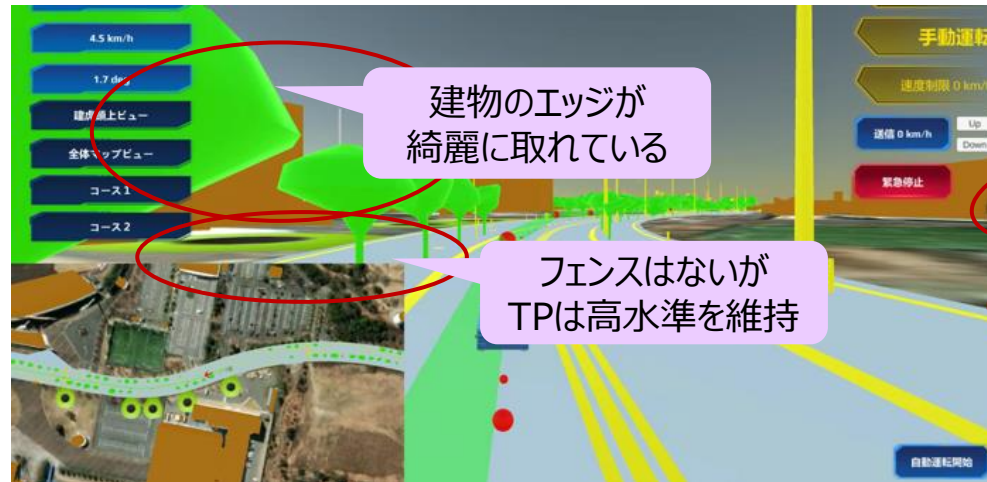
IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 ルート上の特徴点 ② バッファローズ合宿所前

一般論ではカーブは先行きが見えずセンシングが困難だが、3D都市モデルにより建物の角が表現されていること自己位置推定に影響を与えるオブジェクトが少ないことから、精度高いマッチングで最も安定的な自律走行ができた

測定ポイント



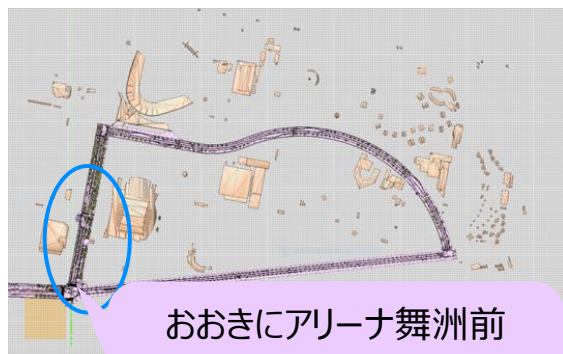
3D都市モデルと現地写真の比較



IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 ルート上の特徴点 ③ おおきにアリーナ舞洲前

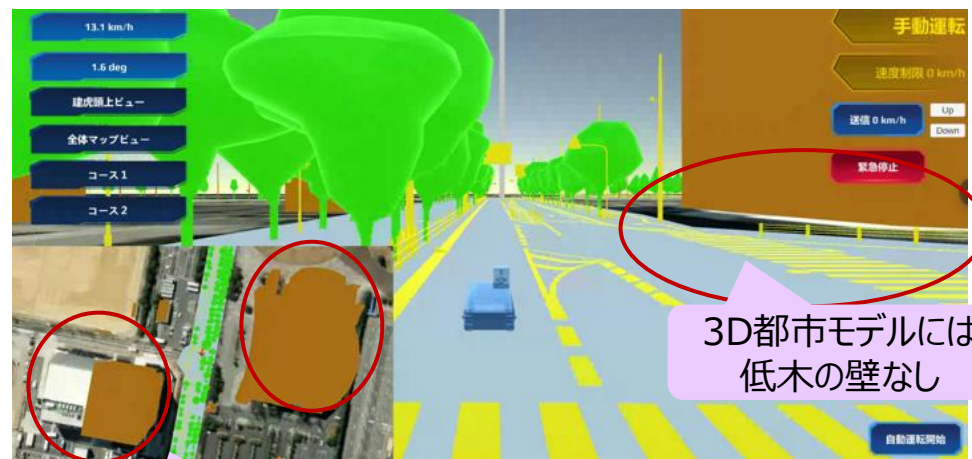
両脇に建物で特徴点を把握できたため精度の高いマッチングが実現し、安定した自律走行ができた

測定ポイント



おおきにアリーナ舞洲前
 自己位置推定は高水準だが
 路駐があるため、手動運転

3D都市モデルと現地写真の比較



3D都市モデルには
 低木の壁なし

3D都市モデルに
 特徴のある二つの
 建物モデルがある



現実世界の建物

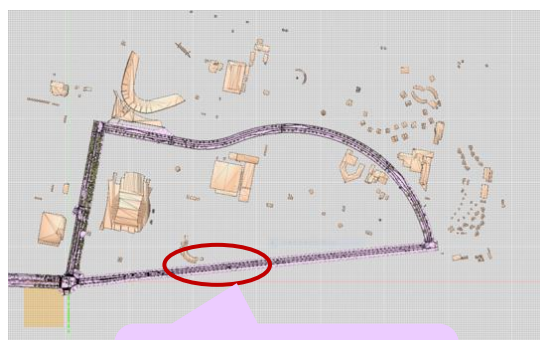
両脇に建物がある

現実には
 低木の壁があるが
 一度マッチングすると
 TPは高水準で推移

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 ルート上の特徴点 ④ バッファローズスタジアム手前

柱や低木のモデルが3D都市モデルとして再現されていないためマッチング精度が低下し、安定した自律走行ができなかった

測定ポイント



バッファローズ
スタジアム手前

3D都市モデルと現地写真の比較



3D都市モデルには
スタジアムのモデルはあるが
柱のモデルはなし
低木のモデルもなし



現実にはスタジアムの
バックネットの柱があり

現実には植え込みあり

IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ② 検証結果 ルート上の特徴点 ⑤ バッファローズスタジアム後の坂道

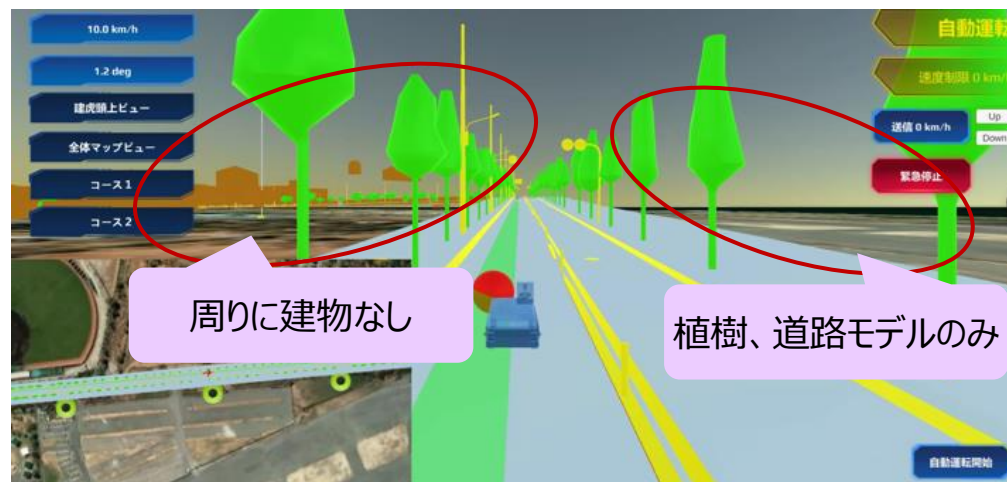
3D都市モデルと現地の乖離が少なかったため、自己位置推定ができ安定的な自律走行ができた

測定ポイント



バッファローズスタジアム
通過後 坂道

3D都市モデルと現地写真の比較



周りに建物なし

植樹、道路モデルのみ



フェンスなどもあるが、
自己位置推定が
下がる影響はなかった

現実にも、植樹しかない

I. 実証概要

II. 実証技術の概要

III. 実証システム

IV. 実証技術の検証

V. 成果と課題

V. 成果と課題 > 1. 今年度の実証で得られた成果

① 3D都市モデルによる技術面での優位性

項目	想定される技術面での優位性
モビリティ用MAP	<ul style="list-style-type: none"> ● 3D都市モデルを活用することで、仮想空間上で搬送車両自律走行用のMAPを作成することができ、現地での事前運行によるMAP作成が不要となった <ul style="list-style-type: none"> - 3D都市モデル（建物LOD2、道路LOD3）から自律走行用のMAPを作成してすることで、事前走行によるMAP作成をすることなく限定された条件下で自律走行ができることが分かった
自動運転監視	<ul style="list-style-type: none"> ● 通常の監視はカメラと2DMAPのみだが、3D都市モデル（LOD3）を用いて植樹、道路標識、地面、建物データ等の豊富なデータを活用した搬送車両のデジタルツインビューによる自動運転監視を行った <ul style="list-style-type: none"> - 道路中央のポールや道路標識などがモデル化されているため鮮明に確認できる - 車両の速度、ステアリング角度、自己位置推定の精度、など車両からの情報も付加することで、より搬送車両走行の監視にも役立つことが分かった

V. 成果と課題 > 1. 今年度の実証で得られた成果

② 3D都市モデルによるビジネス面での優位性

項目	想定されるビジネス面での優位性
自律走行の実現性	<ul style="list-style-type: none"> ● 3D都市モデルがあれば、事前の点群MAP作成をせずに自律走行用の点群マップが作成でき、従来よりもリードタイムが短く自動運行が可能となる <ul style="list-style-type: none"> - 3D都市モデルが整備される以前は、公道での自動運行のためには事前運行による点群MAP作成が必要不可欠であり、公道検証のハードルは高かった
入手・加工の容易性	<ul style="list-style-type: none"> ● 3D都市モデルが整備されていれば、誰でも容易に点群MAPが作成でき、モビリティサービスへの活用が可能となる <ul style="list-style-type: none"> - 自動運行のシミュレーションや点群MAPの生成などで、3D都市モデルは有効なプラットフォームになる - 3D都市モデルの整備エリアの拡大に伴い、モビリティサービスも拡大しやすくなる
オープンデータの信頼性	<ul style="list-style-type: none"> ● 3D都市モデルは公共測量を用いており、同精度であれば問題なく自動運行が可能となる

V. 成果と課題 > 2. 今後の取り組みに向けた課題 活用にあたっての課題

項目	活用にあたっての課題
自己位置推定	<ul style="list-style-type: none"> ● 3D都市モデル由来の点群MAPは現地測量を不要とする分、不得意とする領域がある <ul style="list-style-type: none"> - 3D都市モデルにない地物（植え込み等）や建物施設（柱等）とマッチングができず、自己位置推定に影響を与えている - 可視光カメラを使った道路LOD3の車線によるマッチングとの併用や、3D都市モデル由来の地図の調整手法の検討が必要である
ハードウェア	<ul style="list-style-type: none"> ● 公道で通常走行を行うためには、一般車両に近い走行速度での自己位置推定が求められるが、その点での精度担保が未了 <ul style="list-style-type: none"> - 実証では時速15km以下での自律走行が可能であるものの、より高速な走行速度での精度確保のためにはマッチングスピードを高めるためのアルゴリズム改修やハードウェアのアップグレード等が必要である
システムのユーザーインターフェース	<ul style="list-style-type: none"> ● ターゲットとするユーザー（資材運搬、運輸、等）の利用しやすいインターフェースとなっていない <ul style="list-style-type: none"> - サービス化に向けては、当該ユーザーが利用できるようなユーザーインターフェースの開発が必要である - 上記ユーザー向けにはトラブル発生時のシューティングや品質の安定化が必要である



用語集 (1/5)

用語		内容
ア行	ROS	<ul style="list-style-type: none">• Robot Operating Systemの略• ロボット用のソフトウェアプラットフォームRobot Operating Systemの略• ロボット開発において重要な役割を果たすオープンソースのロボット制御ソフトウェア、およびそれを包括するロボット開発プラットフォーム全体
	RTK	<ul style="list-style-type: none">• Real Time Kinematicの略• 固定局と移動局の2つの受信機を利用し、リアルタイムに2点間で情報をやりとりすることで、高精度での測位を可能にする手法
	IMU	<ul style="list-style-type: none">• Inertial Measurement Unitの略• 慣性計測装置のことで3次元の慣性運動（直行3軸方向の並進運動および回転運動）を検出する
	ICP	<ul style="list-style-type: none">• Iternative Closest Pointの略• 地図データ群とセンサデータ群の2つの点群間の最近傍点を対応点とし、対応関係にある点間の距離計が最短になる箇所をマッチングポイントとする方法
	ichimill	<ul style="list-style-type: none">• Softbank社が提供している高精度測位サービス
	ウェイポイント	<ul style="list-style-type: none">• 経路上の地点情報• 自動運航を行う際に、現在地から次のポイント（座標）を設定したポイント• 複数設定することで、ゴールまでの運航ルートを繋ぐこととなる
	AGV	<ul style="list-style-type: none">• Automated guided vehicleの略• 産業用途で多く使用される自動運転車の一種で、人間が運転操作を行わなくとも自動で走行できる搬送車



用語集 (2/5)

用語	内容
ア行	
SoC	<ul style="list-style-type: none">• System-on-a-chipの略• 集積回路の1個のチップ上に、プロセッサコアをはじめ一般的なマイクロコントローラが持つような機能の他、応用目的の機能なども集積し、連携してシステムとして機能するように設計されている、集積回路製品
STL	<ul style="list-style-type: none">• Standard Triangulated Languageの略• 三次元形状を表現するデータを保存するファイルフォーマットのひとつ
NMEA	<ul style="list-style-type: none">• GNSS受信機から測位結果として出力されるデータ形式の一つ• NMEA0183というフォーマット（NMEAフォーマット）がある
NDT	<ul style="list-style-type: none">• Normal Distribution Transformの略• 環境地図をボクセルごとに区切り正規分布を計算し3次元LiDARから得られた点群データとマッチングを行う手法
MQTT	<ul style="list-style-type: none">• メッセージ指向ミドルウェアのアプリケーション層で使用される、TCP/IPによるPub/Sub型データ配信モデルの軽量なデータ配信プロトコル• Message Queueing Telemetry Transportの略
Autoware.Universe	<ul style="list-style-type: none">• LinuxとROS2をベースとした自動運転システム用オープンソースソフトウェア• レーザレーダ、カメラ、GNSSなどの環境センサを利用し、自車位置や周囲物体を認識しながら、走行ルート上を自律走行する
Autodesk Fusion	<ul style="list-style-type: none">• Autodesk社の3Dモデリング、CAD、CAM、CAE、PCBソフトウェアが統合された、製品設計・製造向けのクラウドベースのプラットフォーム
OBJ	<ul style="list-style-type: none">• 3次元コンピュータグラフィックス（3DCG）で用いる物体の形状データを記録するファイル形式の一つ• 三次元空間における物体の形状を表すデータで、頂点の座標、物体表面を構成する面の情報、曲線や曲面を表すパラメータなどをテキスト形式で記述する

用語集 (3/5)

用語		内容
カ行	Galileo	<ul style="list-style-type: none"> EUによる全地球航法衛星システム
	GLONASS	<ul style="list-style-type: none"> 旧ソビエト連邦が開発し、現在はロシア航空宇宙軍の手によってロシア政府のために運用されている衛星測位システム
	小型特殊車両	<ul style="list-style-type: none"> 車両の長さ4.7m以下・車両の幅 1.7m以下・車両の高さ2.8m以下・最高速度時速15km以下の条件を満たしたものの
サ行	GNSS	<ul style="list-style-type: none"> Global Navigation Satellite Systemの略 「全球測位衛星システム」を意味 米国のGPS、日本の準天頂衛星（QZSS）、ロシアのGLONASS、欧州連合のGalileo等の衛星測位システムの総称
	自己位置推定	<ul style="list-style-type: none"> ある場所にいる自分自身の位置を推定する技術 GPSを使う手法やカメラ映像を使う手法、LiDARの超音波センサーを使用する手法などが存在する
	自己位置測位	<ul style="list-style-type: none"> 自己位置を確定すること
	準天頂衛星	<ul style="list-style-type: none"> Quasi-Zenith Satellite Systemの別名 日本及びアジア太平洋地域向けに利用可能とする航法衛星システム 全地球を対象とするグローバル・ポジショニング・システム（GPS）と異なり、局地的な位置情報サービスを目的とするために準天頂軌道を周回する人工衛星を利用する
	SLAM	<ul style="list-style-type: none"> Simultaneous Localization and Mappingの略 同時位置推定と地図作成を行う技術 ロボットなどの自律移動体が、自らがいる環境を探索しながら、自身の位置を推定しながら地図を作成する



用語集 (4/5)

用語		内容
タ行	DJI Onboard SDK	<ul style="list-style-type: none">• DJI社から提供されているドローン内のハードウェア制御のための開発ツール• ドローンに搭載したシリアルポート(TTLUART) を使用して、オンボードコンピューターに対応するフライトコントローラーに接続可能
	TP	<ul style="list-style-type: none">• Transformation Probabilityの略• マッピング点群と走行点群のマッチング度
	trimesh	<ul style="list-style-type: none">• 3Dオブジェクトを多角形の集合で表現することが可能な、三角メッシュを読み込み操作・分析するためのPythonライブラリ
ハ行	PCD	<ul style="list-style-type: none">• Point Cloud Dataの略• 点群を扱うファイル形式の一つ
	VCSEL	<ul style="list-style-type: none">• 半導体レーザーの一種• 表面実装した基板に対し垂直方向にレーザーを射出する
	PLATEAU Autoware Converter	<ul style="list-style-type: none">• CityGML形式のデータをAutowareに読み込める形式 (PCD形式) に変換するプログラム
ヤ行	Unity	<ul style="list-style-type: none">• Unity Technology社が開発・提供するゲームエンジンおよび開発環境• スマートフォン向けのアプリや家庭用ゲーム機・ウェブなどの様々なプラットフォームで実行可能なアプリケーションを開発することができる



用語集 (5/5)

用語	内容	
ラ行	LiDAR	<ul style="list-style-type: none">• Light Detection And Rangingの略• レーザー光を照射して、その反射光の情報をもとに、対象物までの距離や対象物の形などを計測する技術
	ラストワンマイル	<ul style="list-style-type: none">• 現在は、物流・交通業界において多く用いられ、「顧客にモノ・サービスが到達する最後の1マイル（1.6km程度）の区間・接点」を指す• 元々は通信業界に用いられていた用語で「生活者や企業に対し、通信接続を提供する最後の区間・接点」を意味していた
	ルート整合率	<ul style="list-style-type: none">• 事前取得の点群マップを用いた位置推定や3D都市モデル由来の点群マップによる自己位置推定から得られるTP値（Transformation Probability：マッピング点群と走行点群のマッチング度）
	WARPNER	<ul style="list-style-type: none">• WebRTC（音声やビデオ、データなどをWebブラウザ間でリアルタイムにやり取りするための通信規格）を利用した映像伝送システム

3D都市モデルとBIMを活用したモビリティ自律運行システム 技術検証レポート（無人搬送車両編）

令和5年3月 発行

委託者：国土交通省 都市局 都市政策課

受託者：株式会社竹中工務店・株式会社センシンロボティクス・アダワープジャパン株式会社・株式会社アルモ

本報告書は、株式会社竹中工務店・株式会社センシンロボティクス・アダワープジャパン株式会社・株式会社アルモが国土交通省との間で締結した業務委託契約書に基づき作成したものです。受託者の作業は、本報告書に記載された特定の手続や分析に限定されており、令和5年3月までに入手した情報にのみ基づいて実施しております。従って、令和5年4月以降に環境や状況の変化があったとしても、本報告書に記載されている内容には反映されておりません。