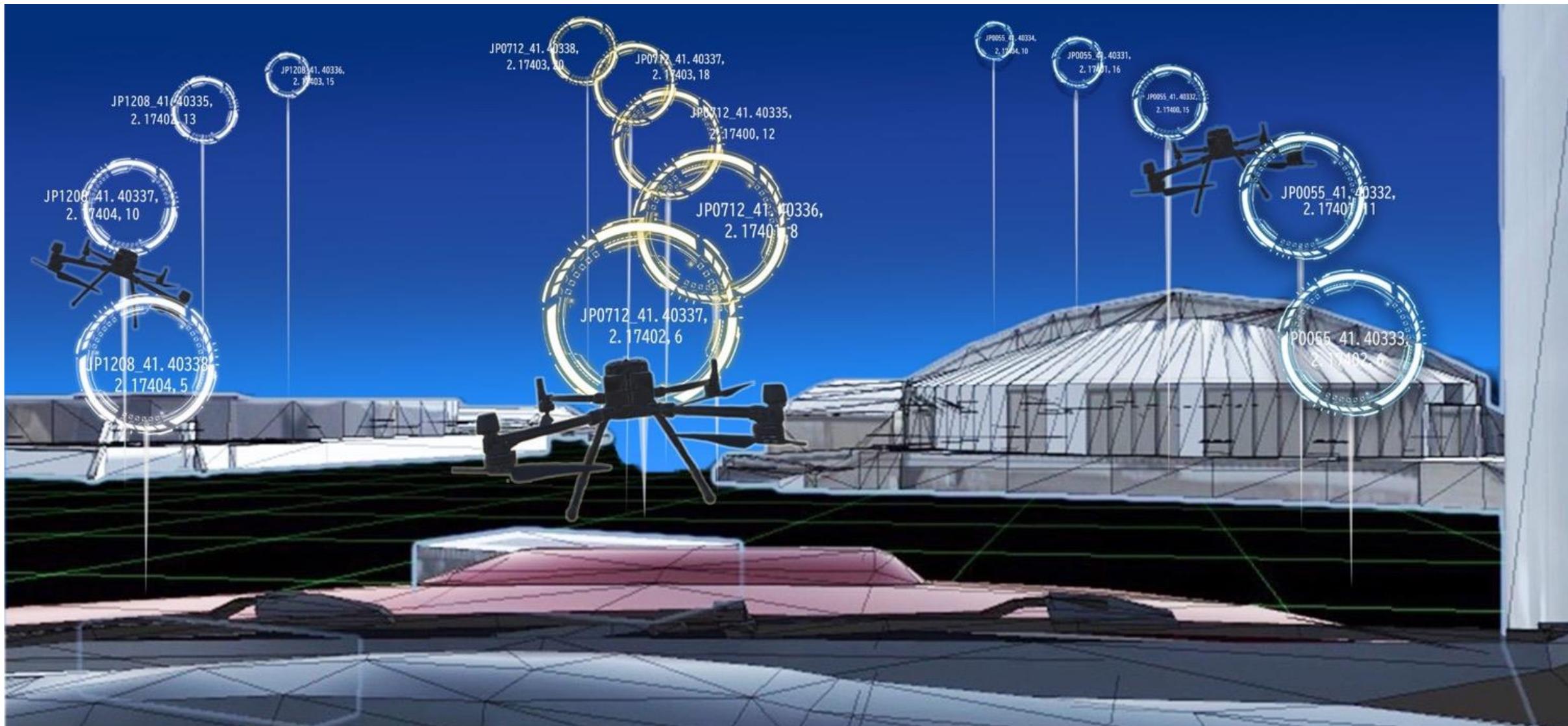


# 3D都市モデルとBIMを活用したモビリティ自律運行システム技術検証レポート (ドローン編)

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



# 目次

<b>I. 実証概要</b>	
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
<b>II. 実証技術の概要</b>	
1. 活用技術	9
2. LiDARセンサー	10
3. LiDARローカリゼーション	11
4. ROS	13
5. ArUcoマーカー	14
<b>III. 実証システム</b>	
1. 実証フロー	16
2. 想定事業機会	17
3. アーキテクチャ全体図	18
4. システム機能	22
5. アルゴリズム	37
6. データ	
① 活用データ	50
② データ処理	56
③ 出力データ	59
7. ユーザインタフェース	60
8. システムテスト結果	64
<b>IV. 実証技術の検証</b>	
1. 自己推定位置精度の検証	
① 検証内容	67
② 検証結果	72
<b>V. 成果と課題</b>	
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	75
② 3D都市モデルによるビジネス面での優位性	76
2. 今後の取り組みに向けた課題	77
<b>用語集</b>	78

# 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（ドローン）
実施場所	神奈川県川崎市川崎区扇町地域
目標・課題 ・創出価値	<ul style="list-style-type: none"><li>● 建設工事の資材運搬時の交通渋滞を解決する。<ul style="list-style-type: none"><li>- 交通渋滞は工事の遅延を引き起こすだけでなく近隣住民の生活や都市機能への影響も大きい</li></ul></li><li>● 建設現場への「ラストワンマイル運搬」の課題を解決する。<ul style="list-style-type: none"><li>- 「ラストワンマイル運搬」とは、具体的に資材置き場から工事現場の作業員への資材運搬を指すが、この運搬手段の課題を自律飛行ドローンによって解決する</li></ul></li><li>● 将来はGPSが入りにくい環境下である都市部や屋外・屋内のシームレスな自律飛行や、有人地帯でのドローンの正確で安全な目視外飛行を目指す。</li></ul>
ユースケース の概要	<ul style="list-style-type: none"><li>● 現在のドローン飛行に関わる下記課題を解決し、資材運搬時等の自律飛行を可能とする運行システムを開発する。本システムはLiDARやGPS等のセンサーと3D都市モデルを利用した自己位置測位を組み合わせて構成する<ul style="list-style-type: none"><li>- ドローンでは、GPS測位のための飛行では、受信状況が悪いビルの間や屋内などでは正確で安全な飛行を担保できない課題がある</li><li>- 飛行に必要な地図情報として民間事業者が提供する3Dマップを利用せざるを得ず、精度担保やデータ連携、カバレッジ等の点で課題がある</li></ul></li></ul>

# I. 実証概要 > 1. 全体概要

## 全体概要 (2/2)

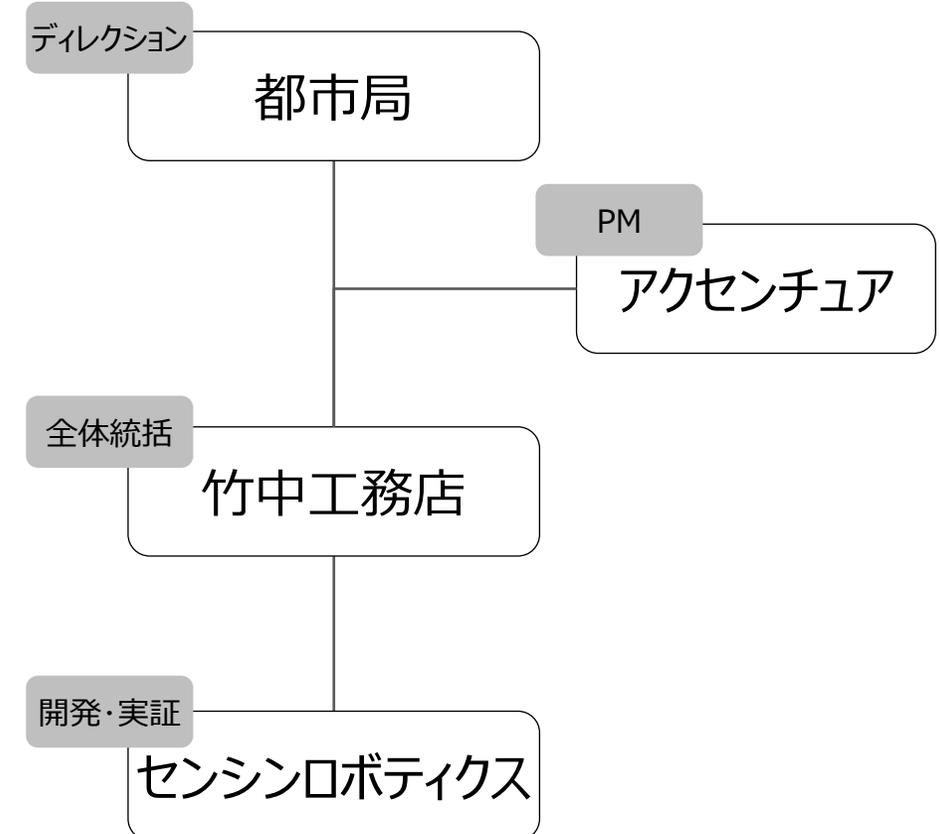
<b>実証仮説</b>	<ul style="list-style-type: none"><li>• LiDARによる自己位置推定に必要な作業の効率化：通常LiDAR SLAMでドローンの自己位置推定を行う場合、同一のエリアにおいて一度ドローンを飛行させドローンの環境地図を作成（スキヤニング）する必要があるが、3D都市モデル（CityGML）を点群化しドローンの環境地図に利用することが出来ると、事前の飛行（スキヤニング）工数を削減できる。</li><li>• ドローンの自己位置測位機能の冗長化、高精度化：上述の方法で得たマップを用いてローカリゼーションを行ったLiDAR自己位置推定と、GPS（RTK）データから得られる自己位置測位を統合・取捨選択することで、測位機能が冗長化され都市部における安全な飛行が可能となる。</li><li>• 屋外/屋内のシームレスな飛行を実現することで、建設現場の「ラストワンマイル運搬」の課題解決が可能となる。</li></ul>
<b>検証ポイント</b>	<ul style="list-style-type: none"><li>• GPSデータ・3D都市モデルデータ・センサーからの情報を統合・取捨選択して自己位置推定を行うドローンが、3D都市モデル（CityGML）とBIMから作成したマップをもとに自律飛行する際の精度の検証<ul style="list-style-type: none"><li>- MAP上に飛行ルートを設定し同じルートを飛行できるか（再現性）</li><li>- MAP上に設定したウェイポイントへの到達した際の位置精度は実用に十分か（精度）</li></ul></li></ul>

# 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
センシロボティクス	• ドローン対応システム作成 • ドローン自動飛行実証
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：神奈川県 川崎市 川崎区 扇町地域 (約26,000m<sup>2</sup>)



# 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. ドローン自動飛行システム開発			←→											
4. ドローンユースケース実証				←→										
5. 報告書の作成										←→				

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

項目	内容
LiDARセンサー	<ul style="list-style-type: none"><li>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、点群データを取得する技術</li></ul>
LiDARローカリゼーション	<ul style="list-style-type: none"><li>LiDARセンサーで移動体が周囲の環境をセンシング、マップを用いて自己位置推定を行う技術</li></ul>
ROS	<ul style="list-style-type: none"><li>ロボット・アプリケーション作成を支援するライブラリとツール、ロボット/モビリティの自律運行を実現する技術</li></ul>
ArUcoマーカー	<ul style="list-style-type: none"><li>姿勢推定に使用できるバイナリの正方形基準マーカー</li><li>QRコードに似た正方形マーカーにより、堅牢・迅速・簡単にカメラとマーカーの相対位置を検出することが可能な技術</li></ul>

# II. 実証技術の概要 > 2. LiDARセンサー

## LiDARセンサー

LiDAR (Light Detection and Ranging)

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

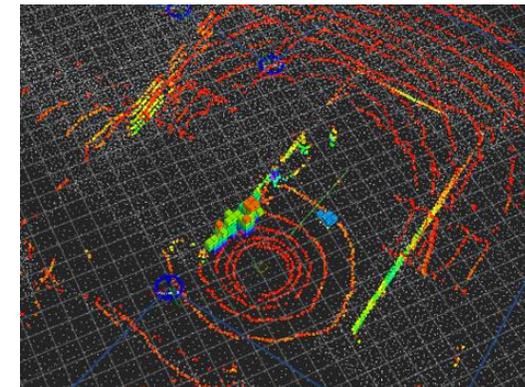
### 技術概要

項目	内容
名称	LiDAR (Light Detection and Ranging) センサー
概要	<ul style="list-style-type: none"> <li>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</li> </ul>
特徴	<ul style="list-style-type: none"> <li>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</li> </ul>
本ユースケースでの用途	<ul style="list-style-type: none"> <li>自己位置推定 (ローカリゼーション) のための点群取得に利用</li> </ul>

### LiDARセンサーと出力データイメージ



Ouster社  
LiDARセンサー OS-0 (オーエスゼロ)



Ouster OS0から出力される点群  
ファイル形式 .las/.pcd など

# II. 実証技術の概要 > 3. LiDAR ローカリゼーション

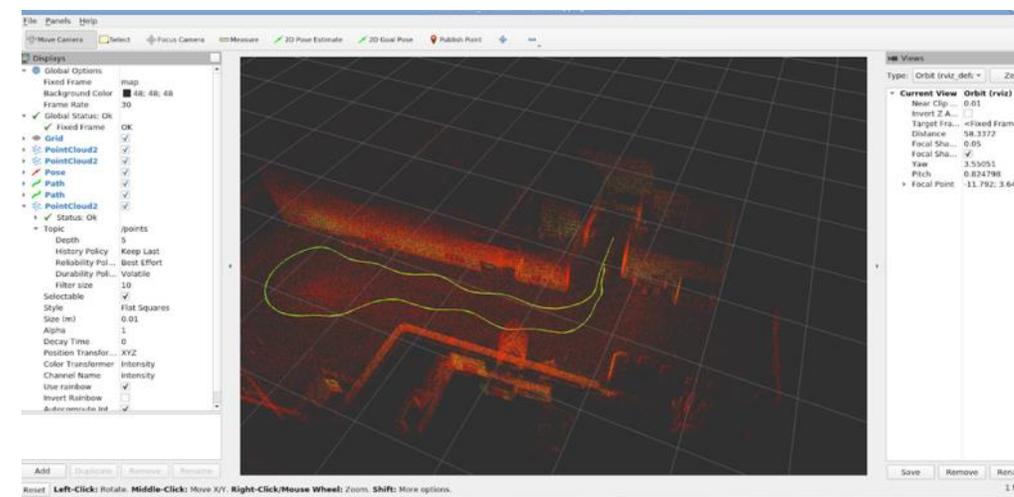
## LiDARローカリゼーション (1/2)

LiDARセンサーで移動体が周囲の環境をセンシングして、マップをもとに自己位置推定を行う技術

### 技術概要

### LiDAR SLAMのスキャンマップとローカリゼーション画面

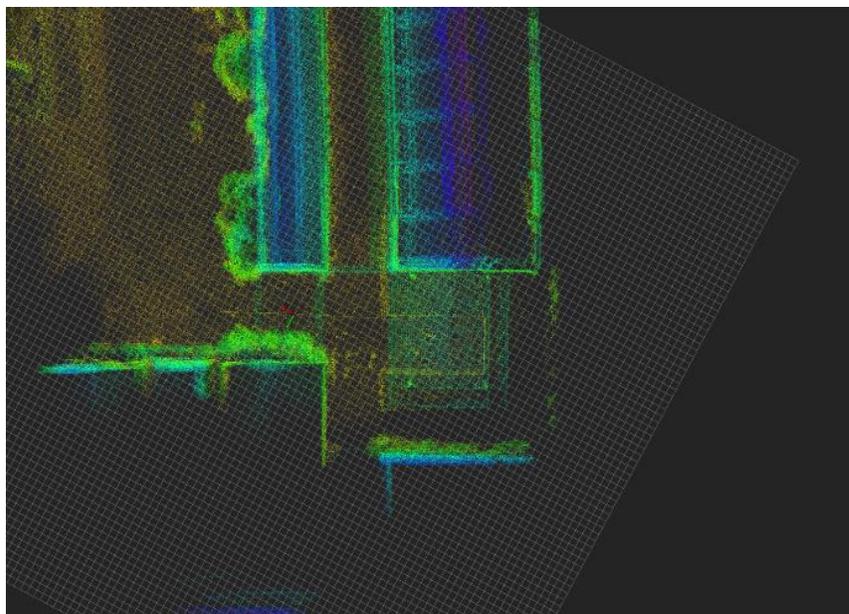
項目	内容
名称	LiDARローカリゼーション
概要	<ul style="list-style-type: none"> <li>LiDARセンサーによって得られた周辺の形状でデータ（点群）をエリアマップと比較することで、現在の自己位置を推定する手法</li> </ul>
主な機能	<ul style="list-style-type: none"> <li>自己位置推定（Localization）</li> </ul>
利用する機能	<ul style="list-style-type: none"> <li>自己位置推定（3D都市モデルを事前マップとして利用）</li> </ul>



## Ⅱ. 実証技術の概要 > 3. LiDAR ローカリゼーション LiDARローカリゼーション (2/2)

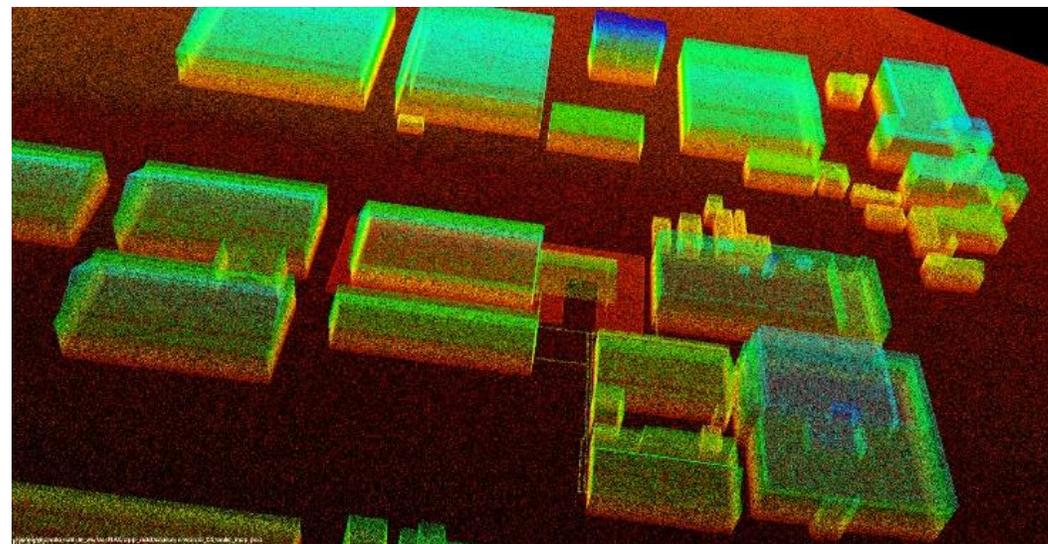
事前のマップ作製を行わず3D都市モデル/BIMからマップを生成して、LiDARローカリゼーションを実施

LiDARデバイスにより生成された点群マップ



事前にスキャンを行いマップを作成することが一般的

3D都市モデル/BIMにより生成された点群マップ



今回の実証においては、3D都市モデル/BIMから生成したマップを利用

## II. 実証技術の概要 > 4. ROS

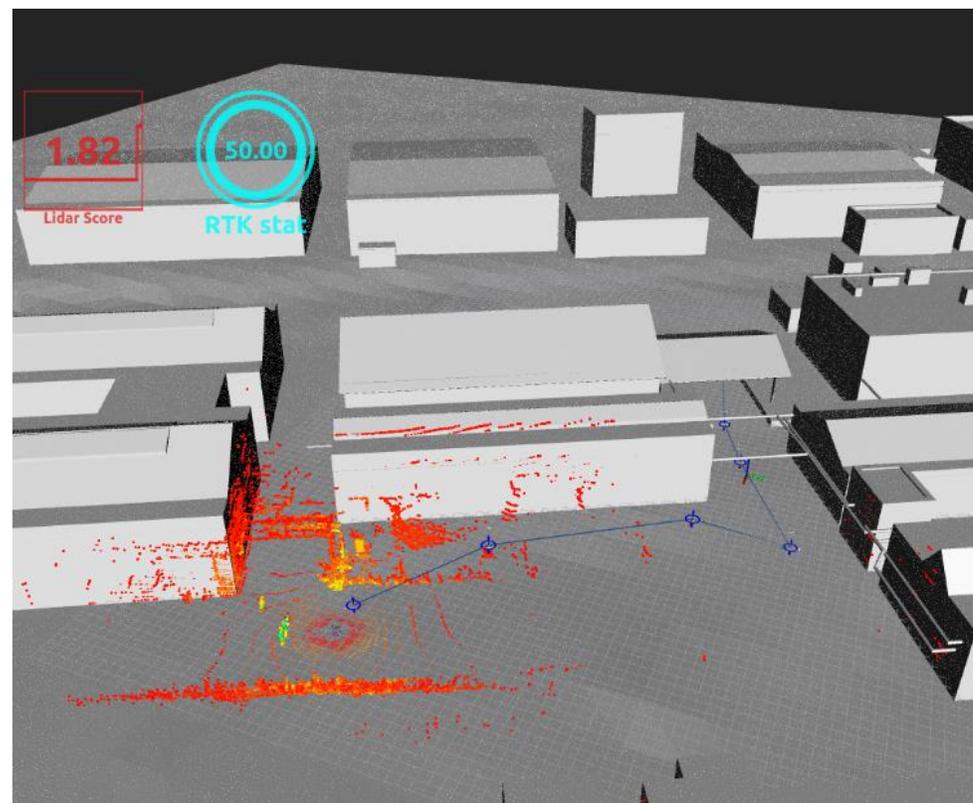
# ROS (ROBOT OPERATING SYSTEM)

ロボット・アプリケーション作成を支援するライブラリとツールを含みロボットモビリティの自律運行を実現するシステムを容易に開発することが可能。汎用性の高いROSを用いて今回実証するシステムの構築を行う

ライブラリ概要

項目	内容
名称	ROS (Robot Operating System) 
概要	<ul style="list-style-type: none"> <li>● ロボット・アプリケーション作成を支援するライブラリとツールのパッケージ</li> <li>● オープンソースで提供</li> </ul>
主な機能	<ul style="list-style-type: none"> <li>● ハードウェア抽象化</li> <li>● デバイスドライバ</li> <li>● ライブラリ</li> <li>● 視覚化ツール</li> <li>● メッセージ通信</li> <li>● パッケージ管理、など</li> </ul>
利用する機能	<ul style="list-style-type: none"> <li>● LiDARセンサーを利用するためのデバイスドライバ</li> <li>● 各データのやり取り</li> <li>● 視覚化ツール</li> </ul>

ROSベース視覚化ツールRvizを用いた表示画面



## II. 実証技術の概要 > 5. ArUcoマーカー

# ArUcoマーカー

QRコードに似た正方形のマーカーにより、迅速・簡単にカメラとマーカーの相対位置を検出することが可能

### 技術概要

項目	内容
名称	ArUcoマーカー
概要	<ul style="list-style-type: none"> <li>姿勢推定に利用可能な正方形の基準マーカー</li> <li>検出スピードが速く、実装も容易に可能</li> <li>シンプルなマーカーはノイズ・エラーに強いという特徴も持つ</li> </ul>
主な機能	<ul style="list-style-type: none"> <li>マーカーとカメラ間の姿勢・位置の検出</li> </ul>
利用する機能	<ul style="list-style-type: none"> <li>ドローンにカメラを搭載し、地上のマーカーとの相対位置 (X,Y,Z) 座標を取得する</li> </ul>

### ArUcoマーカーの利用イメージ

ドローン下部に取り付けたカメラ



Intel社  
デプスカメラ リアルセンスD455

ArUcoマーカー

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、3D都市モデルとBIMを合わせた地図データをドローンのLiDARローカリゼーションに利用しGNSS（RTK）測位と合わせて自己位置測位機能を冗長化し、屋外から屋内（非GNSS測位環境）へのシームレス飛行を実現し、飛行精度、再現性を検証する

### 対象エリアの 調査・選定

- ドローンの実証実験が継続的に可能であり、LOD2以上の3D都市モデルを有し、かつBIMを持つ屋外から屋内へのシームレス飛行ルートを検証できるエリアを選定  
-川崎市川崎区扇町

### 機能開発

- 3D都市モデルとBIMをドローン用のマップとする機能
- 任意の飛行経路（ウェイポイントを順次飛行）を作成、リアルタイムでドローン位置を確認するUIの開発

### 既存開発機能との 結合

- 開発した機能を、開発済の機能と結合しシステム化

### システムの 有用性検証

- 対象エリアにおいて飛行検証を実施、開発機能を合わせたシステムを用いてドローンの自律飛行が達成できるか検証
- 計測地点における飛行精度、再現性を評価



## Ⅲ. 実証システム > 2. 想定事業機会 想定事業機会

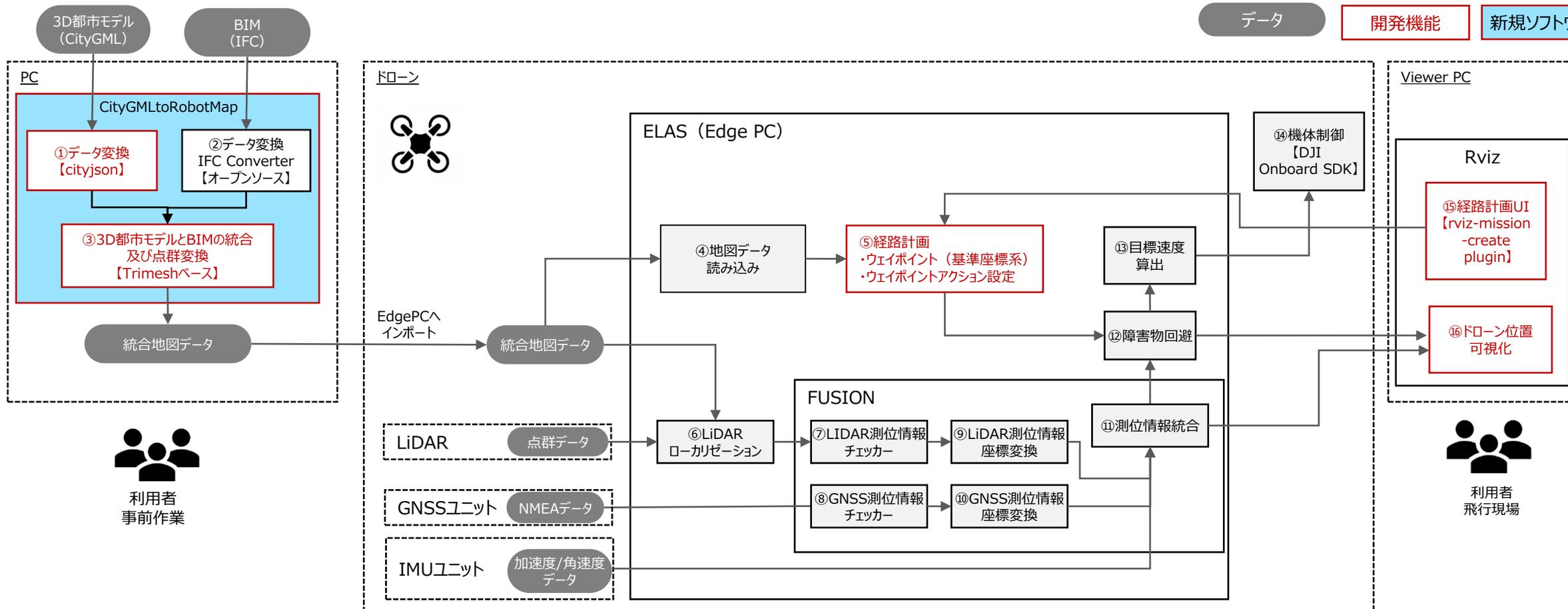
3D都市モデルとセンサーを活用した自己位置測位により、GPS測位のみでは安定性が担保されない場所でのドローンの飛行に対して、安定的な自律飛行を可能にする運行システムの提供

項目	内容
利用事業者	<ul style="list-style-type: none"><li>ゼネコン、運輸サービス事業者、ビルメンテナンス会社、警備会社、マスコミ・報道、消防、警察</li></ul>
提供価値	<ul style="list-style-type: none"><li>精度担保された3D都市モデルを活用した、ドローンの自律飛行を可能とする運行システムの提供<ul style="list-style-type: none"><li>- ドローンではGPS測位のみでの飛行では安全な飛行を担保できず、民間事業者が提供する3Dマップデータは精度担保やデータ連携、カバレッジ等の点の課題がある</li><li>- そのためオープンデータである3D都市モデルと汎用センサーを活用した運行システムを開発し提供する</li></ul></li></ul>
サービス仮説	<ul style="list-style-type: none"><li>ドローン自律飛行による業務支援サービス<ul style="list-style-type: none"><li>- 利用事業者の省力化につながる自律飛行システムを使った業務支援サービスを行う</li><li>- 例) 資材運搬や物流における自律ドローンによるラストワンマイルの輸送支援</li></ul></li></ul>

# Ⅲ. 実証システム > 3. アーキテクチャ全体図

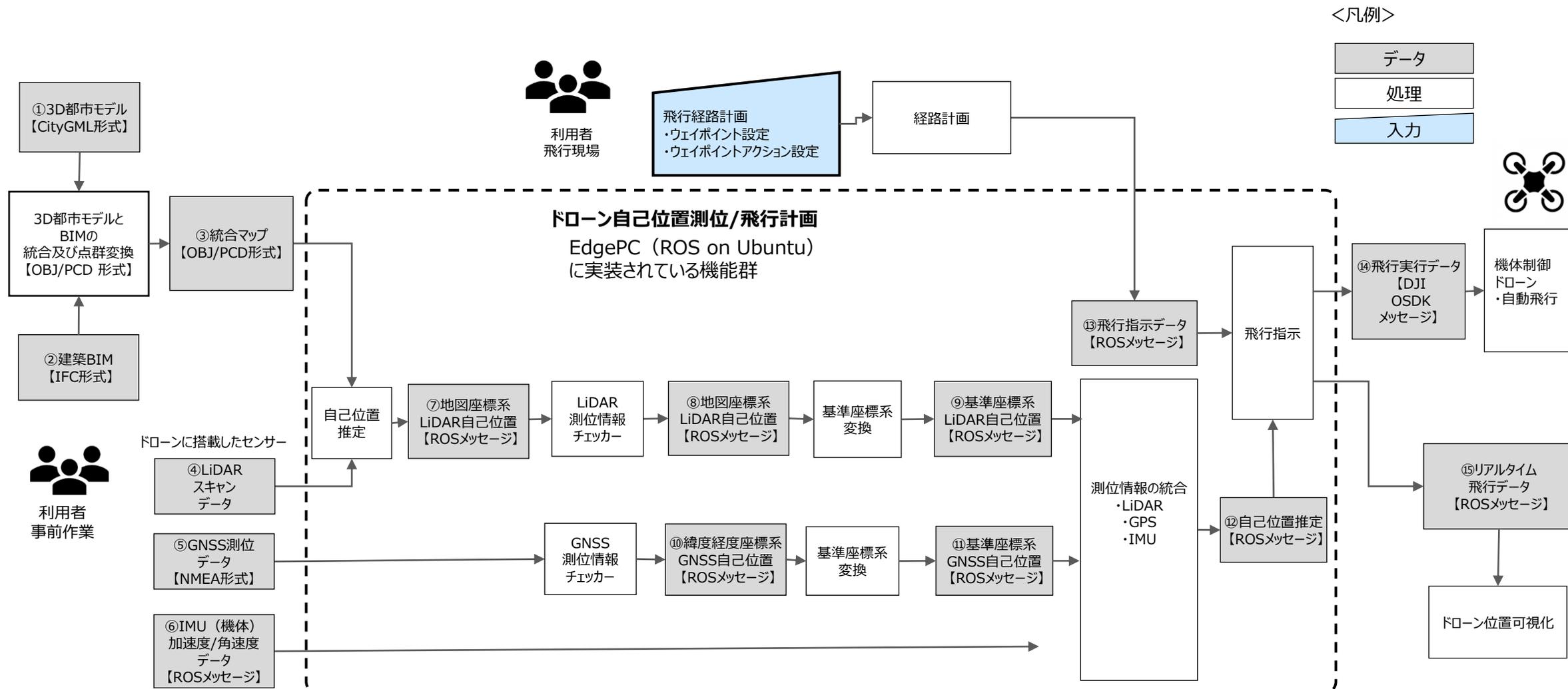
## システムアーキテクチャ全体図

<凡例>

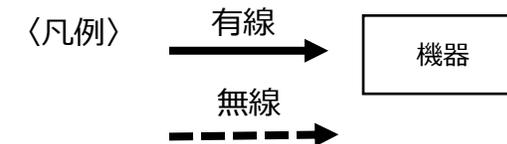


# Ⅲ. 実証システム > 3. アーキテクチャ全体図

## データアーキテクチャ全体図

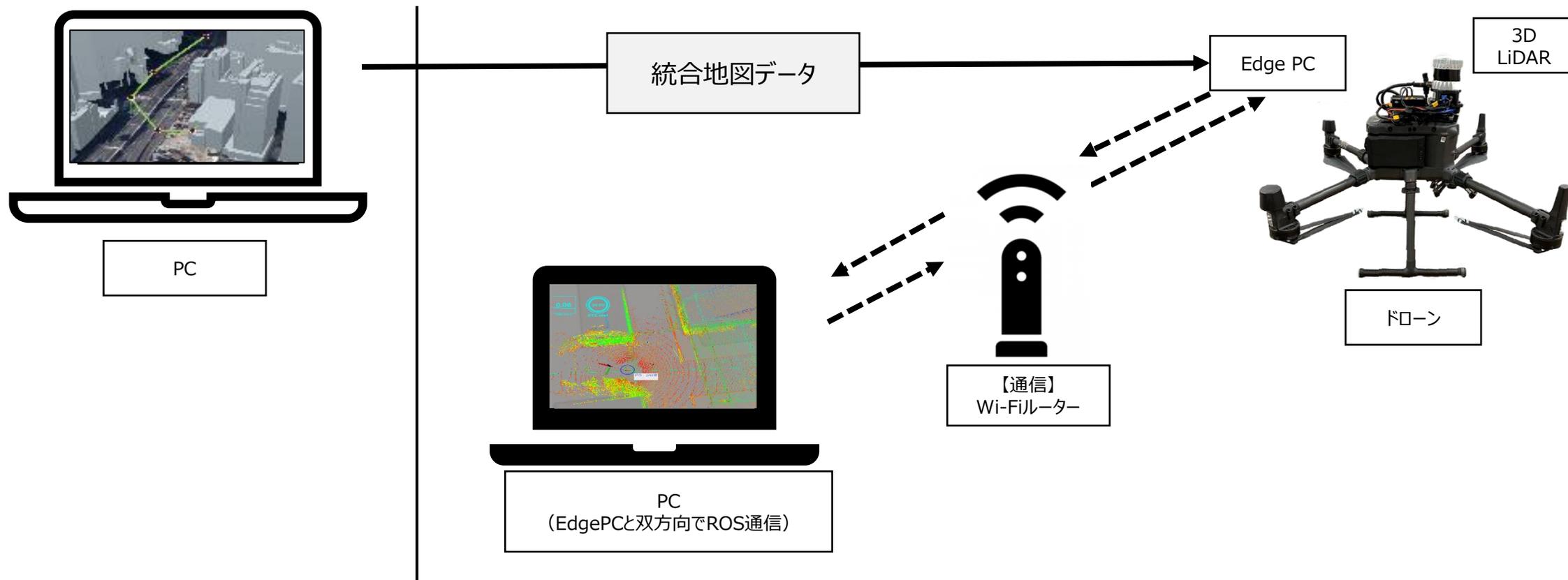


# Ⅲ. 実証システム > 3. アーキテクチャ全体図 ハードウェアアーキテクチャ全体図



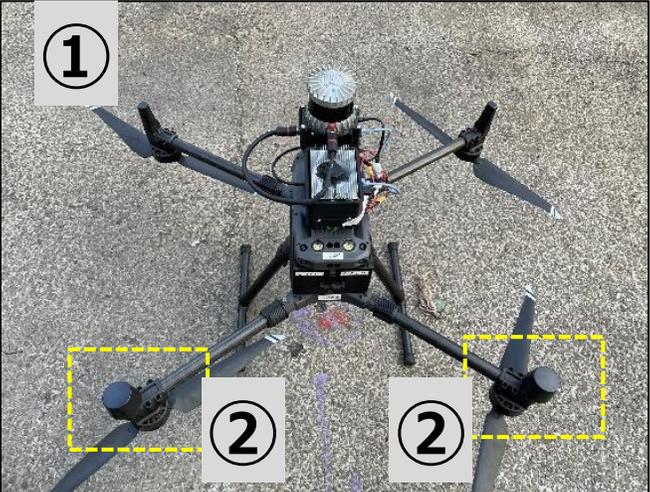
利用者事前作業

利用者飛行現場



# Ⅲ. 実証システム > 3. アーキテクチャ全体図 利用するハードウェア

		品番	選定理由	仕様
①	ドローン	DJI Matrice300	<ul style="list-style-type: none"> <li>EdgePCとLiDARを搭載しやすい設計</li> <li>外部SDKを具備</li> <li>ペイロードが十分ある</li> </ul>	<ul style="list-style-type: none"> <li>サイズ:810×670×430mm</li> <li>最大積載量:2.7 kg</li> <li>最大風圧抵抗:15 m/秒</li> <li>最大飛行時間:55分</li> </ul>
②	RTKユニット	①に搭載されたものを使用	<ul style="list-style-type: none"> <li>機体にインテグレーション済</li> </ul>	<ul style="list-style-type: none"> <li>GPS+GLONASS+BeiDou+Galileo</li> <li>1cm+1ppm (水平方向)</li> <li>1.5cm+1ppm (垂直方向)</li> </ul>
③	3D LiDAR	Ouster OS0	<ul style="list-style-type: none"> <li>超広視野角</li> <li>軽量 (450g)</li> </ul>	<ul style="list-style-type: none"> <li>反射率10%のターゲットで30m以上</li> <li>最大測定範囲100m以上</li> <li>垂直視野90度</li> <li>ビーム本数128本の解像度</li> <li>520万ポイント/秒</li> </ul>
④	EdgePC	PICO-WHU4W2-A11-0004	<ul style="list-style-type: none"> <li>小型であり、且つCPUが高性能</li> </ul>	<ul style="list-style-type: none"> <li>サイズ : 縦100mm×横72mm</li> <li>CPU: Intel®i7-8665UE</li> <li>Storage : SATAIII x 1mSATA x1</li> </ul>



# Ⅲ. 実証システム > 4. システム機能

## システム機能一覧 (1/2)

機能一覧①～⑦

<凡例> **赤太字**：新規開発機能

#	ソフトウェア名称	機能	詳細
①	<b>CityGMLtoRobotMap</b>	<b>データ変換 (3D都市モデル)</b>	<ul style="list-style-type: none"> <li>cityjsonのライブラリーを用いて開発した変換機能</li> <li>3D都市モデル (3D tiles)を.objへ変換する</li> </ul>
②	IFC Converter	データ変換 (BIM)	<ul style="list-style-type: none"> <li>BIMを属性情報を省き.objへ変換する機能</li> </ul>
③	<b>CityGMLtoRobotMap</b>	<b>3D都市モデルとBIMの統合及び点群変換</b>	<ul style="list-style-type: none"> <li>Trimeshのライブラリーを用いて開発した機能               <ul style="list-style-type: none"> <li>3D都市モデルとBIMを同一マップに合成して.objと点群 (Point Cloud)を生成する</li> <li>両モデルの位置調整の機能も持つ</li> </ul> </li> </ul>
④	ELAS	地図データ読み込み	<ul style="list-style-type: none"> <li>経路計画を行うために外部より点群マップを読み込む</li> </ul>
⑤		<b>経路計画</b>	<ul style="list-style-type: none"> <li>ウェイポイント (基準座標系)</li> <li>ウェイポイントアクション設定</li> </ul>
⑥		自己位置推定	<ul style="list-style-type: none"> <li>ドローンに搭載した3D LiDARで取得したデータと3D都市モデルとBIMを統合した点群マップを重ね、自己位置を推定する(LiDARではマッピングを行わない)</li> </ul>
⑦		LiDAR測位情報チェッカー	<ul style="list-style-type: none"> <li>測位情報の取舍選択</li> </ul>

# Ⅲ. 実証システム > 4. システム機能

## システム機能一覧 (2/2)

機能一覧⑧～⑯

<凡例> **赤太字**：新規開発機能

#	ソフトウェア名称	機能	詳細
⑧	ELAS	GNSS測位情報チェッカー	● 測位情報の取捨選択
⑨		LiDAR測位情報変換	● 基準座標系へ変換
⑩		GNSS測位情報変換	● 基準座標系へ変換
⑪		測位情報統合	● 複数の測位情報を統合
⑫		障害物回避	● 障害物回避
⑬		目標速度算出	● 目標位置からドローンの移動速度を決定/指示
⑭		DJI Onboard SDK	機体制御
⑮	Rviz	<b>経路計画UI</b>	● 経路計画を行う画面を表示
⑯		<b>ドローン位置可視化</b>	● ドローンの現在位置をリアルタイムで表示

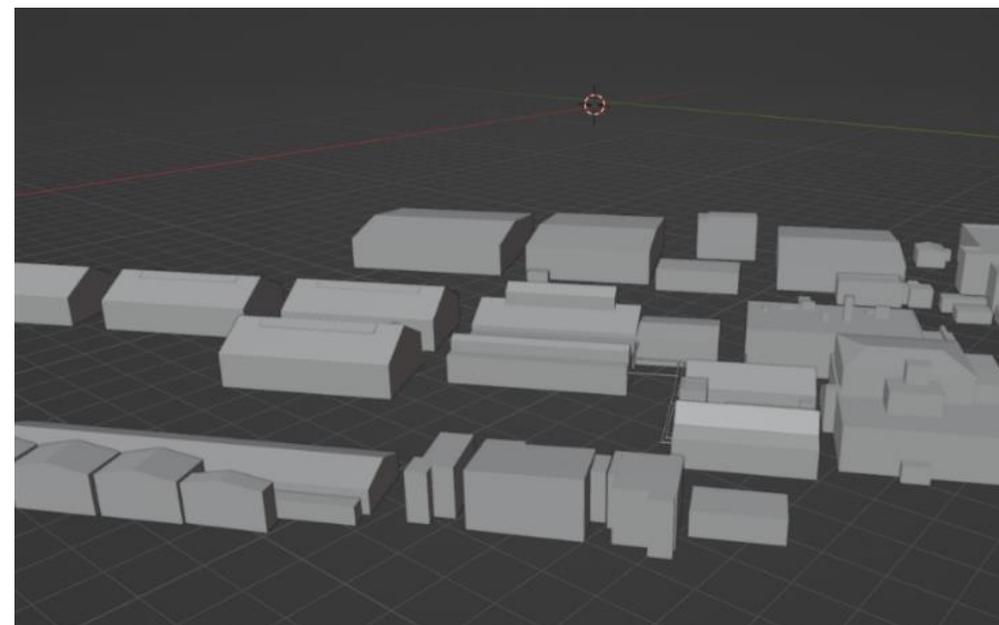
# Ⅲ. 実証システム > 4. システム機能 データ変換（3D都市モデル）

CityGML形式のデータをOBJデータに変換する

## 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>CityGML形式の3D 都市モデルをOBJデータに変換</li> </ul>
入力データ仕様	<ul style="list-style-type: none"> <li>CityGML (JGD2011)</li> </ul>
出力データ仕様	<ul style="list-style-type: none"> <li>3D都市モデルのOBJ (基準座標系)</li> </ul>
利用するライブラリ	cityjson

## 変換モデル（川崎市川崎区扇町）



# Ⅲ. 実証システム > 4. システム機能

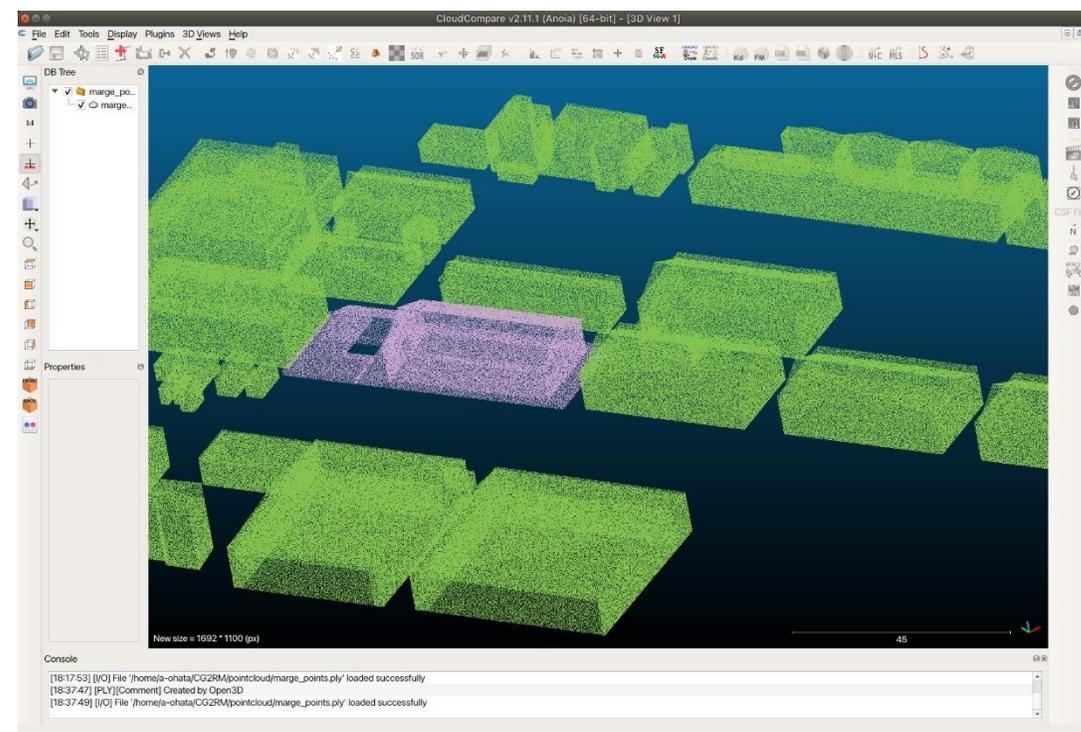
## 3D都市モデルとBIMの統合及び点群生成

3D都市モデル上にBIMを配置し統合された点群を生成する

概要

統合し点群化したモデル

項目	内容
機能概要	<ul style="list-style-type: none"> <li>CityGML形式の3D都市モデルから生成したOBJデータとBIMから生成したOBJデータを統合する機能</li> <li>OBJデータを指定した密度で点群に変換する機能</li> </ul>
入力データ仕様	<ul style="list-style-type: none"> <li>3D都市モデルから生成したOBJ(基準座標系)</li> <li>BIMから生成したOBJ(モデル固有の座標系)</li> </ul>
出力データ仕様	<ul style="list-style-type: none"> <li>3D都市モデルとBIMが統合された点群ファイル(基準座標系)</li> <li>点群の面積あたりの密度:30 points/m<sup>2</sup></li> <li>データフォーマット: pcd</li> </ul>
利用するライブラリ	Trimesh



緑点群はCityGMLより生成  
ピンク色の部分はBIMより生成

# Ⅲ. 実証システム > 4. システム機能

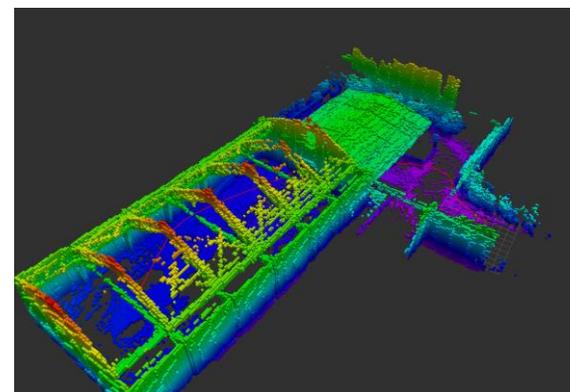
## ELAS

事前マップを用いて自己位置測位、かつ経路計画を行い、ロボットに移動経路/指示を与える技術を用いて自律移動を行うソフトウェアパッケージ

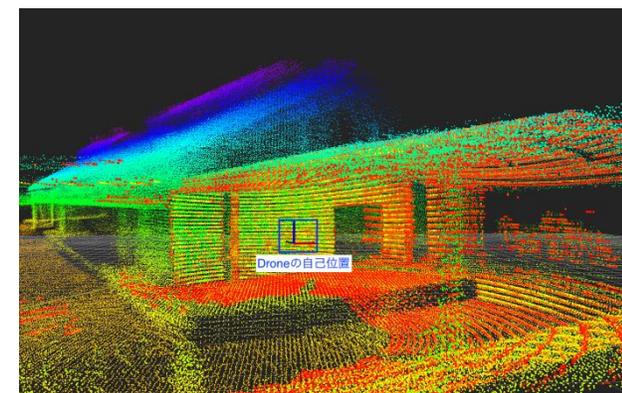
### 技術概要

項目	内容
名称	ELAS (Edge LiDAR Autonomous System )
概要	<ul style="list-style-type: none"> <li>非GPS・GPS環境におけるロボットの自律移動に必要な、認識、計画、制御 機能を有したソフトウェアパッケージ</li> </ul>
主な機能	<ul style="list-style-type: none"> <li>自己位置推定</li> <li>LiDAR測位情報チェッカー</li> <li>GNSS測位情報チェッカー</li> <li>LiDAR測位情報座標変換</li> <li>GNSS測位情報座標変換</li> <li>測位情報統合</li> <li>障害物回避</li> <li>目標速度算出</li> </ul>

### イメージ



自律移動時のイメージ



# Ⅲ. 実証システム > 4. システム機能

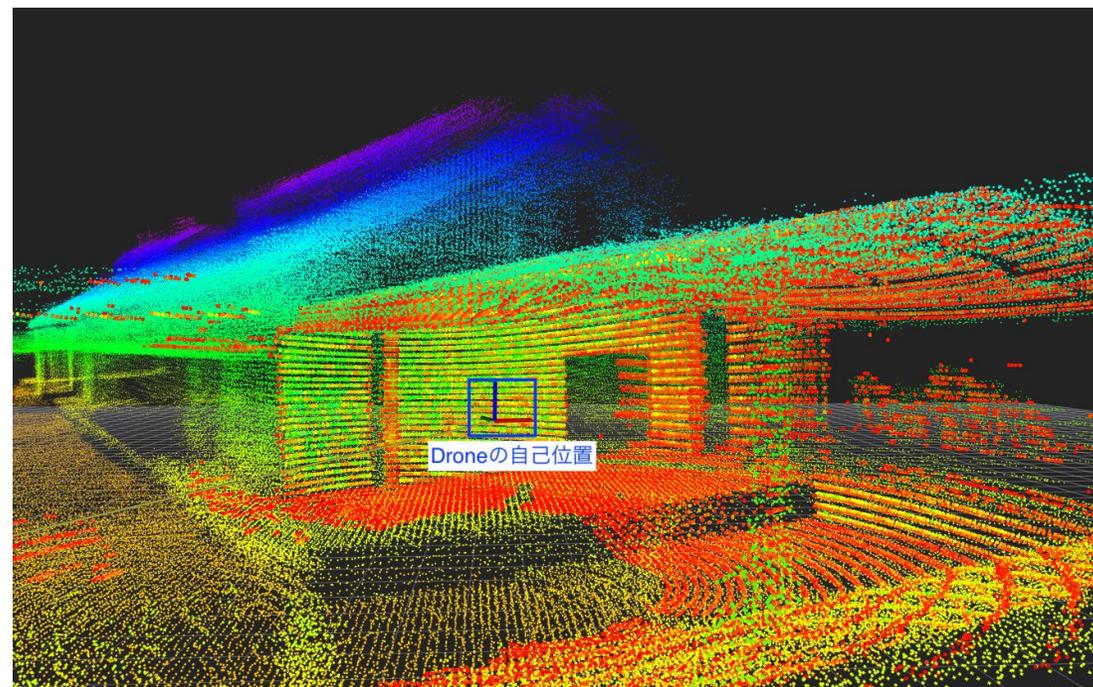
## ELAS | 自己位置推定

対象環境の点群とリアルタイムに取得されるLiDAR スキャン点群をマッチングすることで自己位置を推定

### 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>点群化したデータとスキャンデータをマッチングして自己位置推定を行う機能</li> </ul>
入力データ	<ul style="list-style-type: none"> <li>環境点群地図</li> <li>LiDARスキャン点群データ</li> </ul>
出力データ	<ul style="list-style-type: none"> <li>LiDARの自己位置情報 (x,y,z,roll,pitch,yaw)</li> </ul>
利用するライブラリ	Point cloud library
利用するアルゴリズム	LiDARローカリゼーション (NDT Registration)

### 川崎市川崎区扇町でのドローンによる自己位置推定の様子

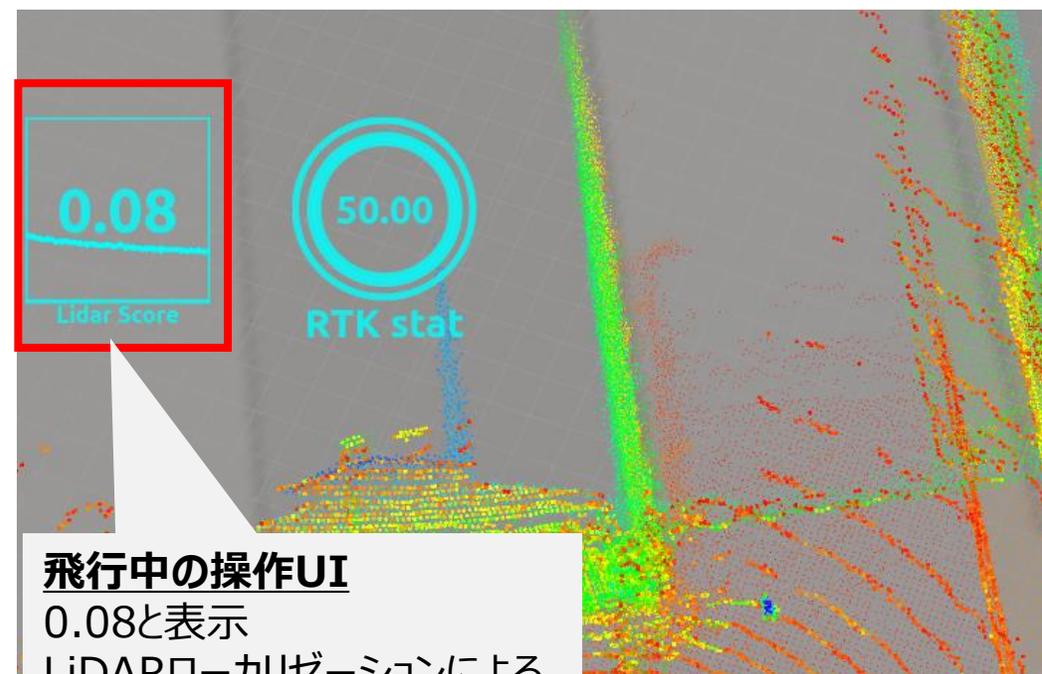


# Ⅲ. 実証システム > 4. システム機能 ELAS | LiDAR測位情報チェッカー

概要

イメージ

項目	内容
機能概要	<ul style="list-style-type: none"><li>• LiDARによる自己位置推定の結果が使用可・不可の判断を信頼度を用いて行う</li><li>• 信頼値が2.0以下であれば使用可</li></ul>
入力データ	<ul style="list-style-type: none"><li>• LiDAR測位情報</li><li>• 信頼度</li></ul>
出力データ	<ul style="list-style-type: none"><li>• LiDAR測位情報</li></ul>
利用するライブラリ	(スクラッチ開発)



## 飛行中の操作UI

0.08と表示

LiDARローカリゼーションによる自己位置推定が利用可能であることを示す

LiDARローカリゼーションによって得た、自己位置推定の信頼値

# Ⅲ. 実証システム > 4. システム機能 ELAS | GNSS測位情報チェッカー

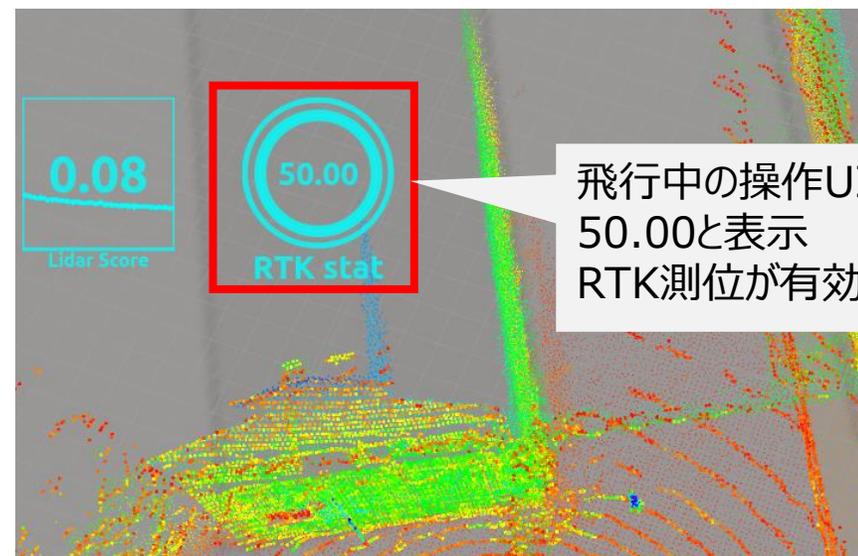
## 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>RTK-GNSSによる測位結果が使用可・不可の判断をステータスにより行う</li> </ul>
入力データ	<ul style="list-style-type: none"> <li>RTK-GNSS測位情報</li> <li>RTK-GNSS測位ステータス情報</li> </ul>
出力データ	<ul style="list-style-type: none"> <li>RTK-GNSS測位情報</li> </ul>

RTK Status Checker ステータスバリュー		
Status	DJI Onboard SDK定義の数値*	期待位置精度
<b>FIX</b>	<b>50</b>	<b>2-3cm</b>
FLOAT 2	32	20-40cm
FLOAT 1	16	2-3 m
No solution	0	なし

\*DJI Onboard SDK定義の数値は測位環境や衛星数を表しているわけではなく、あくまで測位信頼度を表現するラベルとして定義されている

## イメージ



飛行中の操作UI  
50.00と表示  
RTK測位が有効



飛行中の操作UI  
0.00と表示時はRTK  
測位無効

# Ⅲ. 実証システム > 4. システム機能

## ELAS | LiDAR測位情報座標変換

### 概要

項目	内容
機能概要	<ul style="list-style-type: none"><li>LiDAR測位情報を基準座標系上の自己位置情報に変換する</li></ul>
入力データ	<ul style="list-style-type: none"><li>LiDAR測位情報（緯度経度高さ・姿勢）</li></ul>
出力データ	<ul style="list-style-type: none"><li>基準座標系上のLiDAR自己位置</li></ul>
利用するライブラリ	(スクラッチ開発)

### イメージ



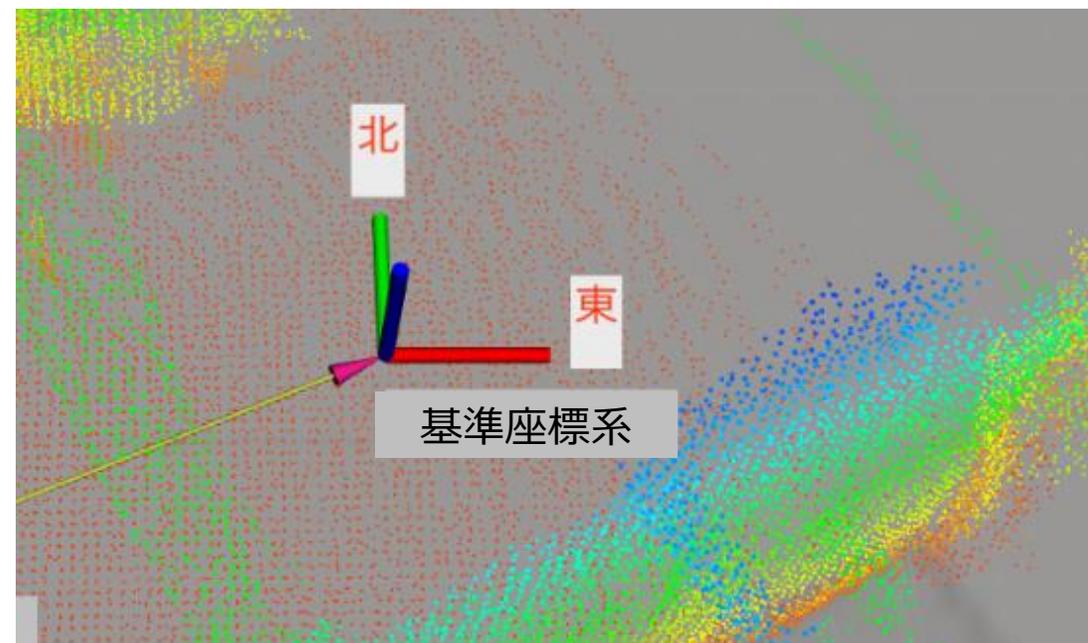
- 基準座標系と地図座標系との相対関係から、地図座標系で得た位置を基準座標系に変換する
- 図中の点群は事前取得した点群地図

# Ⅲ. 実証システム > 4. システム機能 ELAS | GNSS測位情報座標変換

## 概要

項目	内容
機能概要	<ul style="list-style-type: none"><li>• RTK-GNSS測位情報を基準座標系上の自己位置情報に変換する</li></ul>
入力データ	<ul style="list-style-type: none"><li>• RTK-GNSS測位情報（緯度経度高さ・向き）</li></ul>
出力データ	<ul style="list-style-type: none"><li>• 基準座標系上のRTK-GNSS自己位置</li></ul>
利用するライブラリ	proj

## イメージ



- 基準座標系は、任意の緯度経度高さにより定めた、ENU座標系
- (East, North, Up) の3軸が (x, y, z) 座標軸に対応する

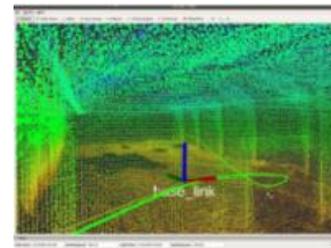
# Ⅲ. 実証システム > 4. システム機能 ELAS | 測位情報統合

## 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>複数の測位・計測情報を統合する機能</li> </ul>
入力データ	<ul style="list-style-type: none"> <li>GNSS測位情報</li> <li>LiDAR測位情報</li> <li>IMUの加速度角速度</li> </ul>
出力データ	<ul style="list-style-type: none"> <li>自己位置情報</li> </ul>
利用するライブラリ	Robot localization
利用するアルゴリズム	測位情報の統合

## イメージ

測位情報A LiDARローカリゼーション



測位情報B GNSS (RTK)



良好	良好
良好	不良
不良	良好
不良	不良

- GNSS (RTK) が良いときはGNSS (RTK) を利用、LiDARローカリゼーションが良いときはLiDARローカリゼーションを利用、また両方良いときは両方を使う取捨選択を行う
- 両方とも不良の場合は、飛行の中止判断をする

# Ⅲ. 実証システム > 4. システム機能

## ELAS | 障害物回避

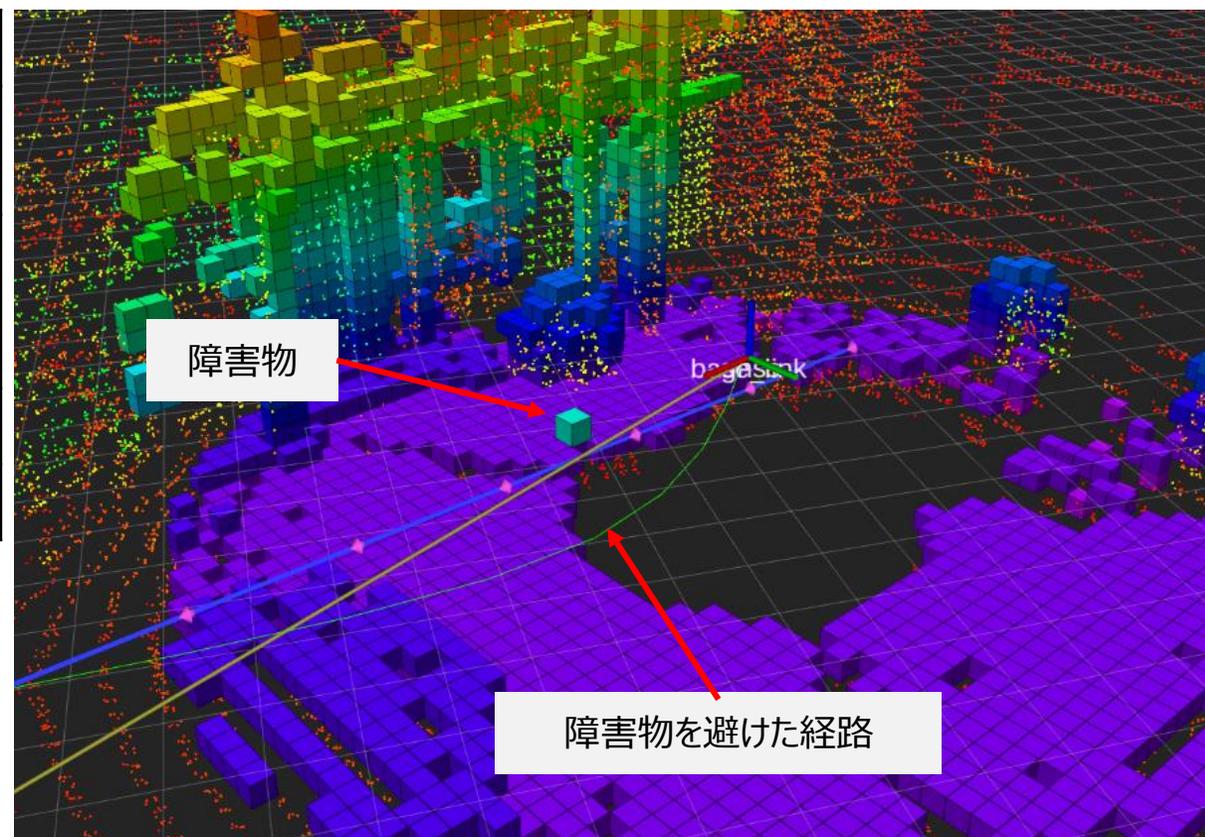
障害物が存在する場合は障害物を避ける経路を生成し、移動目標点を更新する機能

### 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>障害物が存在する場合は障害物を避ける経路を生成し、移動目標点を更新する機能</li> </ul>
入力データ	<ul style="list-style-type: none"> <li>周囲の障害物情報</li> <li>機体自己位置</li> <li>目標経路</li> </ul>
出力データ	<ul style="list-style-type: none"> <li>経路</li> </ul>
利用するライブラリ	Octomap, ompi

- 機体に搭載した3D LiDARから得られるスキャンを元に、30cm四方に区切った空間内（ボクセル）に点群が存在するか判定を行う
- 点群を検知した場合は障害物がある空間として扱う

### 障害物により新たな経路（緑）を生成した様子



# Ⅲ. 実証システム > 4. システム機能

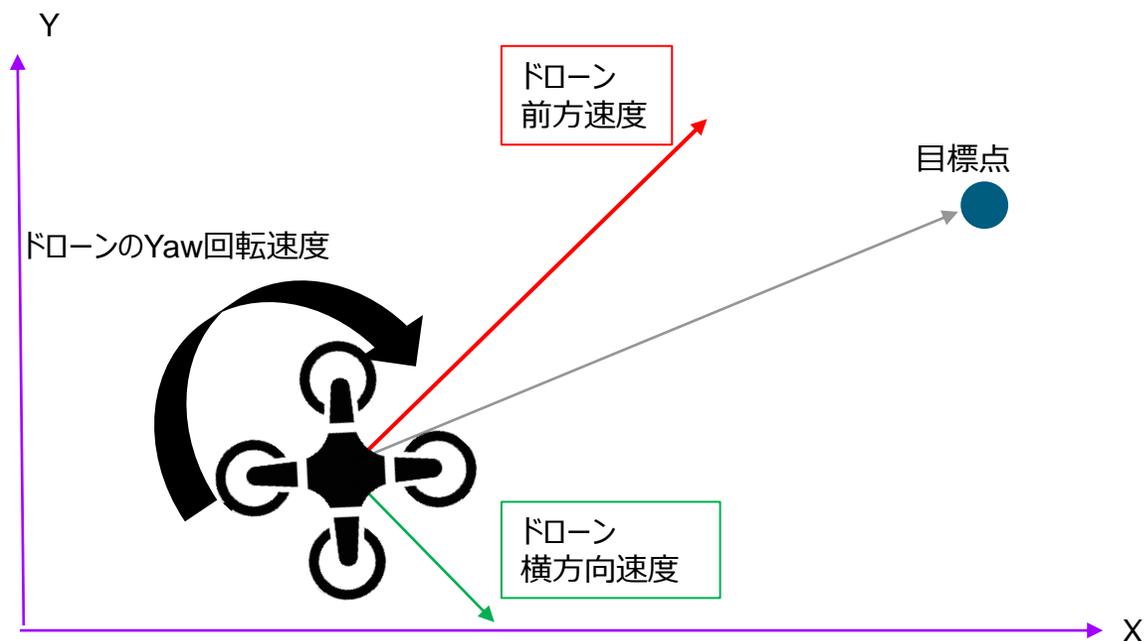
## ELAS | 目標速度算出

ドローンの自己位置と目標経路から移動目標点を求め、目標速度を算出する機能

### 概要

項目	内容
機能概要	<ul style="list-style-type: none"> <li>ドローンの自己位置と目標経路から移動目標点を求め、目標速度を算出する機能</li> </ul>
入力データ	<ul style="list-style-type: none"> <li>機体自己位置</li> <li>目標経路</li> </ul>
出力データ	<ul style="list-style-type: none"> <li>目標速度</li> <li>Yaw回転速度</li> </ul>
利用するライブラリ	(スクラッチ開発)

### 目標速度算出平面 (XY面) 概念図



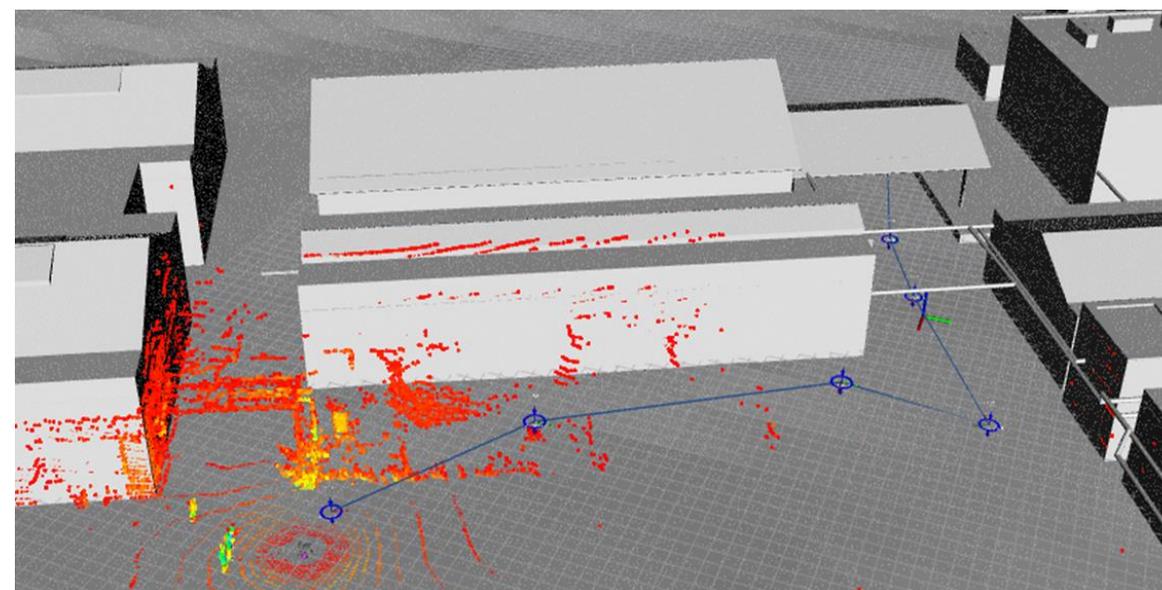
- ドローンの自己位置と目標地点の差分から、 $x, y, z$ 成分ごとに差分を埋め、目標方向に進むような速度を定める

# Ⅲ. 実証システム > 4. システム機能 経路計画UI

## 概要

項目	内容
機能概要	<ul style="list-style-type: none"><li>• リモートコンピュータからドローンに飛行目標経路指示を送信する機能</li><li>• 指示位置・向きなどで構成される飛行目標経路を作成することができる機能</li></ul>
入力データ	<ul style="list-style-type: none"><li>• 画面上のマウスクリックによる指定点</li></ul>
出力データ	<ul style="list-style-type: none"><li>• 経路</li></ul>
利用するライブラリ	(スクラッチ開発)

## UI/ウェイポイントの詳細

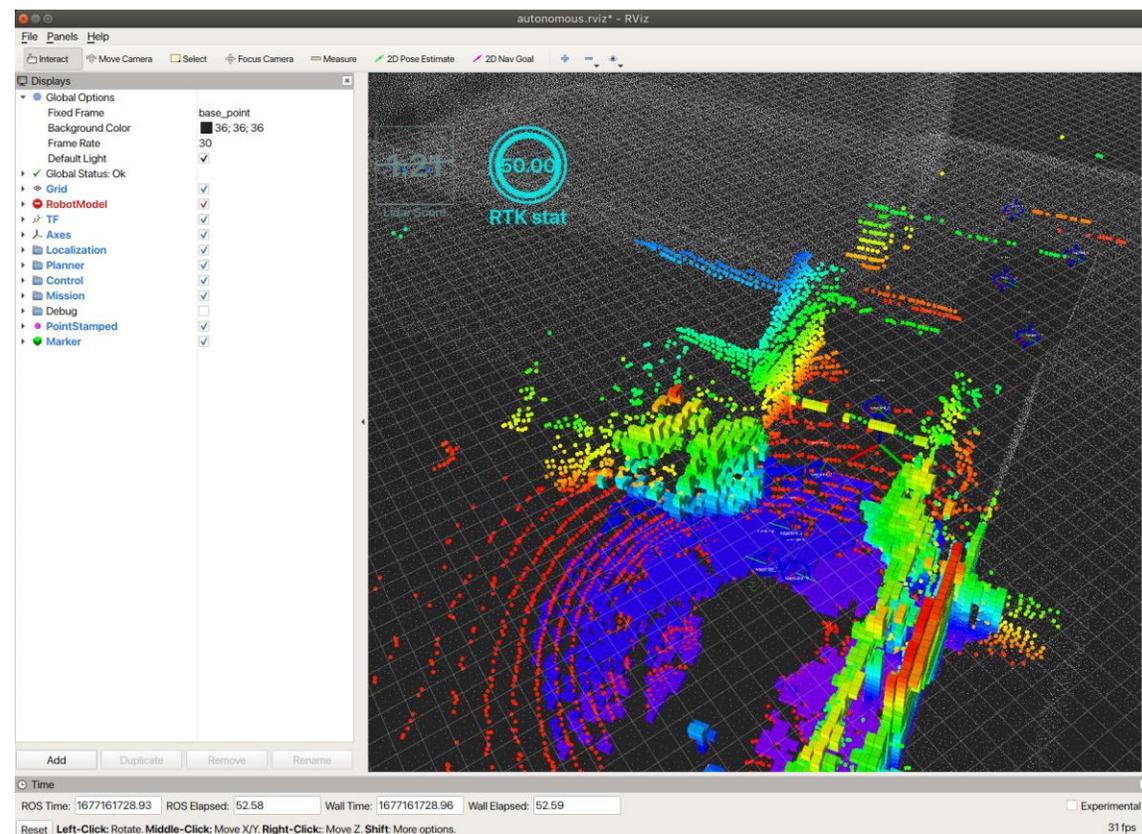


# Ⅲ. 実証システム > 4. システム機能 ドローン位置可視化

## 概要

項目	内容
機能概要	<ul style="list-style-type: none"><li>ドローンの自己位置推定結果や周囲の認識状況をPC上で表示する機能</li></ul>
入力データ	<ul style="list-style-type: none"><li>機体自己位置</li><li>周囲環境計測情報</li></ul>
出力データ	<ul style="list-style-type: none"><li>描画データ</li></ul>
利用するライブラリ	(スクラッチ開発)

Rviz上でドローン自己位置や周囲の認識状況が確認できる様子



# Ⅲ. 実証システム > 5. アルゴリズム アルゴリズム一覧

名称	説明
LiDAR ローカリゼーション	点群地図とLiDARセンサーデータによる自己位置推定アルゴリズム
測位情報の統合	複数の測位・計測情報を統合する機能

# Ⅲ. 実証システム > 5. アルゴリズム LiDAR ローカリゼーション

## フロー

### ①点群地図の正規分布表現への変換

- 事前に準備した点群地図を対象環境地図として読み込む。地図の正規分布表現を生成する

### ②初期位置の設定及びスキャン一致度の評価

- RTK-GNSSが取得可能な場合は自己位置の初期値とし、RTK-GNSSが取得できない場合は手動で初期位置を付与
- 初期位置を与えることで一致度を定義可能となる

### ③LiDAR スキャンによる自己位置推定

- LiDARからのスキャンを0.1mの粒度でボクセル化、初期位置周辺で最も地図と形状が一致する箇所がどこかを判定し最も一致度が高い位置・向きを自己位置とする

### ④結果の評価・初期位置の更新

- 自己位置推定結果の信頼度を評価する
- 最新のLiDARスキャンによる自己位置推定の結果を次回の初期位置とする

### ⑤

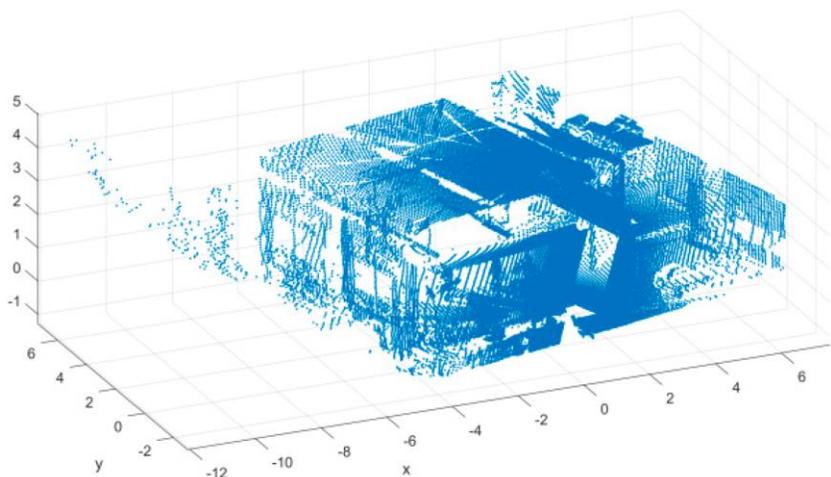
③と④のプロセスを新しくスキャンが得られるごとに繰り返す

# Ⅲ. 実証システム > 5. アルゴリズム

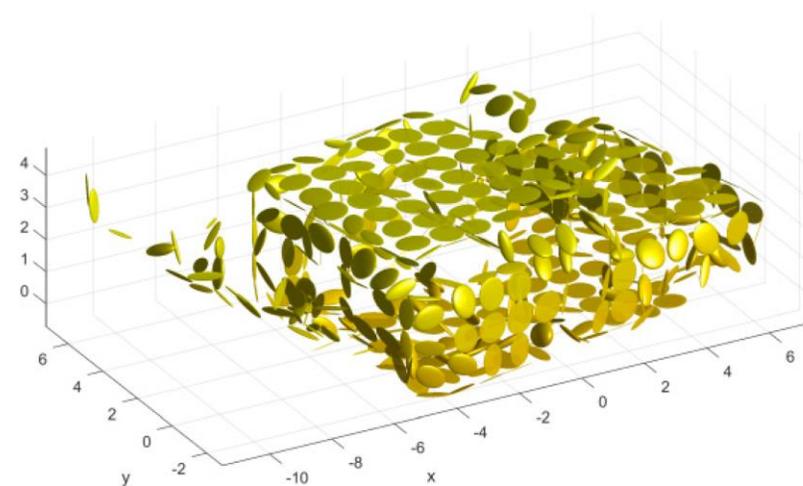
## ① 点群地図の正規分布表現への変換

点群地図の読み込みと、ボクセル内点群を正規分布表現に変換

- ① 地図点群を環境地図として読み込み環境地図を指定サイズのボクセルで分割する
- ② 各ボクセル内部の点群の分布から $x, y, z$ 成分を持った正規分布を求め環境地図を正規分布表現に変換する



点群地図



正規分布

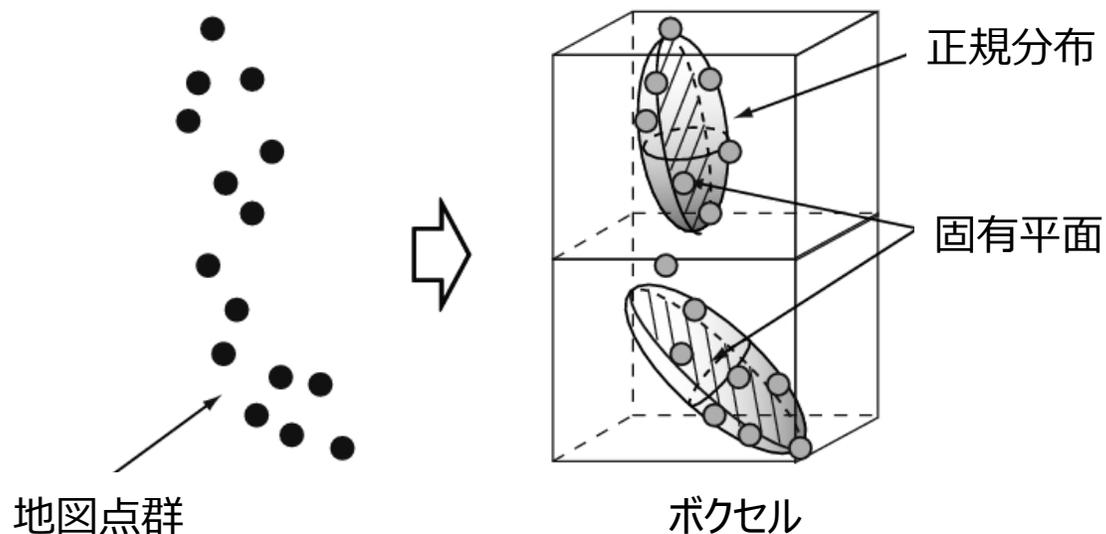
## Ⅲ. 実証システム > 5. アルゴリズム

# ① 点群地図の正規分布表現への変換 | 手法概要

### 点群地図の正規分布表現への変換手法

#### NDT (Normal Distributions Transform) 法

- 2つの点群のマッチング手法の一つ
- 地図空間をボクセルに区切り、ボクセル毎の点の集合を正規分布で近似
- 計算量はスキャンデータに依存、地図データの大きさには依存しない



地図点群をボクセル分割した際、各ボクセル内にある3次元点群を  $P = \{p_i \in \mathbb{R}^3 | i = 1, \dots, n\}$  としたとき、以下のように求められる  $\mu$ 、 $\Sigma$  により特徴付けられる正規分布で地図内の点分布を表現したもの

- $\mu \in \mathbb{R}^3$  : 平均
- $\Sigma \in \mathbb{R}^3 \times 3$  : 分散共分散

$$\mu = \frac{1}{n} \sum_{i=1}^n p_i \quad \Sigma = \frac{1}{n-1} \sum_{i=1}^n (p_i - \mu)(p_i - \mu)^T$$

# Ⅲ. 実証システム > 5. アルゴリズム

## ② 初期位置の設定及びスキャン一致度の評価

### 初期位置設定及びスキャン一致度の評価

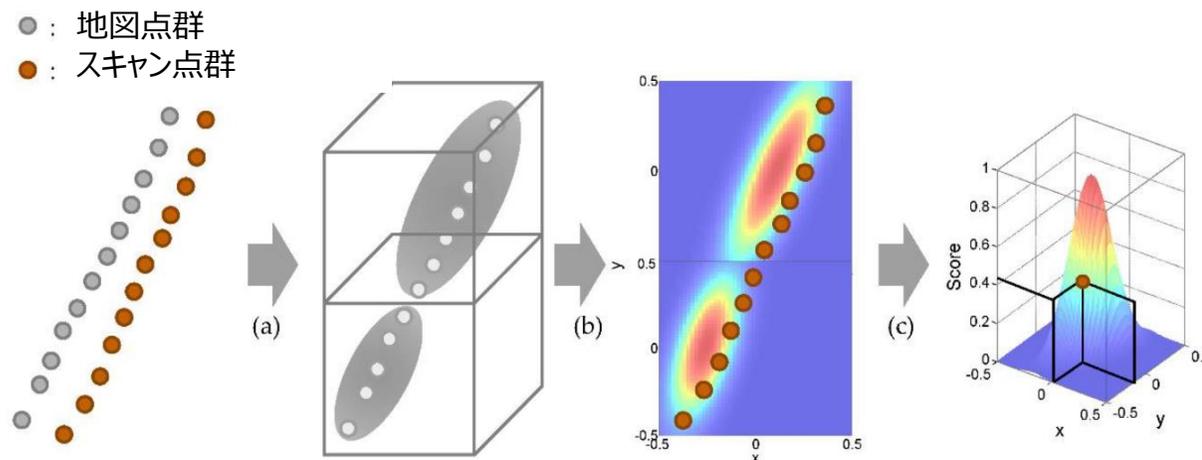
- LiDAR 初期位置が定まると、スキャンと正規分布表現を持つ地図との一致度評価を行うことが可能となる
- RTK-GNSSが取得可能な場合はRTK-GNSSの自己位置を利用、取得不可の場合は手動で初期位置を与える

### スコア算出（スキャン一致度の評価）

スキャン点群の初期位置が与え、下図（b）のように、地図内にある各ボクセルが持つ正規分布上の値を各スキャン点に付与する

この時  $score(p) = \sum_i \exp\left(\frac{-(x'_i - q_i)^t \Sigma_i^{-1} (x'_i - q_i)}{2}\right)$ . をスキャンが持つスコアとして定義

地図点群とスキャン点群の形状の一致度が高いほど高いスコアが得られる



# Ⅲ. 実証システム > 5. アルゴリズム

## ③ LiDAR スキャンによる自己位置推定

### マッチングによる位置更新のプロセス

#### スキャン位置更新前

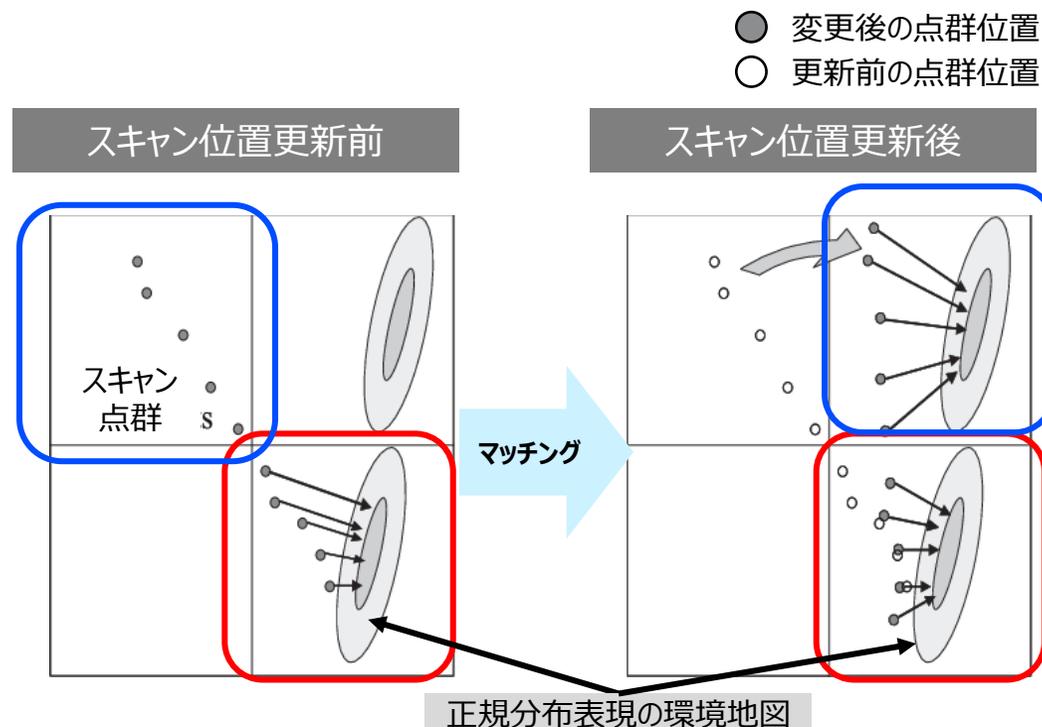
- ボクセル内の点群が持つスコアの合計値が大きくなるようにスキャン位置の更新を行う
  - 青囲**: 現時点で、地図の正規分布表現がないためスコアを持たない
  - 赤囲**: 正規分布表現により一点一点スコアを持つ

#### スキャン位置更新後

- スキャン位置更新によりボクセル内の正規分布と点群との関係が変化
  - 青囲**: 新たに点群にスコアが付与され点群全体のスコアも上がる
  - 赤囲**: 所属する点群のスコアが上がり、より環境地図に近づく

スコア（スキャン各点を持つ値の合計）が最大化するような座標変換とスコアの更新を何度も繰り返すことで、地図とスキャン点群が一致する位置を推定

### マッチングによる位置更新のイメージ



- 等高線で描かれているものは地図が持つ正規分布表現
- 点と正規分布表現の間の線はスコア増加方向を表現する

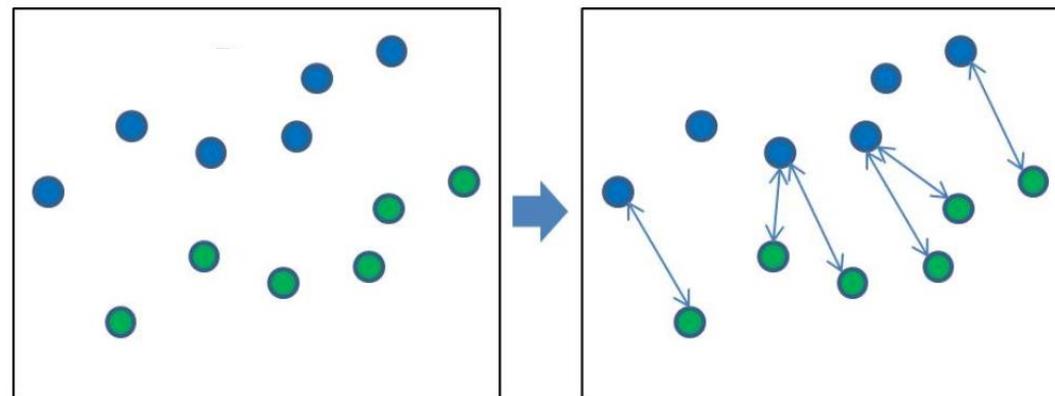
# Ⅲ. 実証システム > 5. アルゴリズム

## ④ 結果の評価・初期位置の更新

### LiDARローカリゼーションの信頼性評価基準

- 信頼性とその評価基準
  - 位置合わせ済みスキャン点群と地図点群とを比較し信頼度を算出
  - スキャン点群の各点から最も近い点群マップ内の点までの距離の平均値を算出しそれを信頼値とし、**信頼値が2.0未満を飛行中に自己位置推定の結果を使用可とする条件とした**
    - 複数の飛行試験データを分析・検証した際に信頼値が2.0未満であればスキャンマッチングがうまく動作することが判明
- 初期位置の更新
  - 信頼できる自己位置が取得できた場合は、次回スキャンが得られた際のLiDARローカリゼーションの初期位置として入力

### スキャン点群と地図点群の対応イメージ



- スキャン点群
- 地図点群

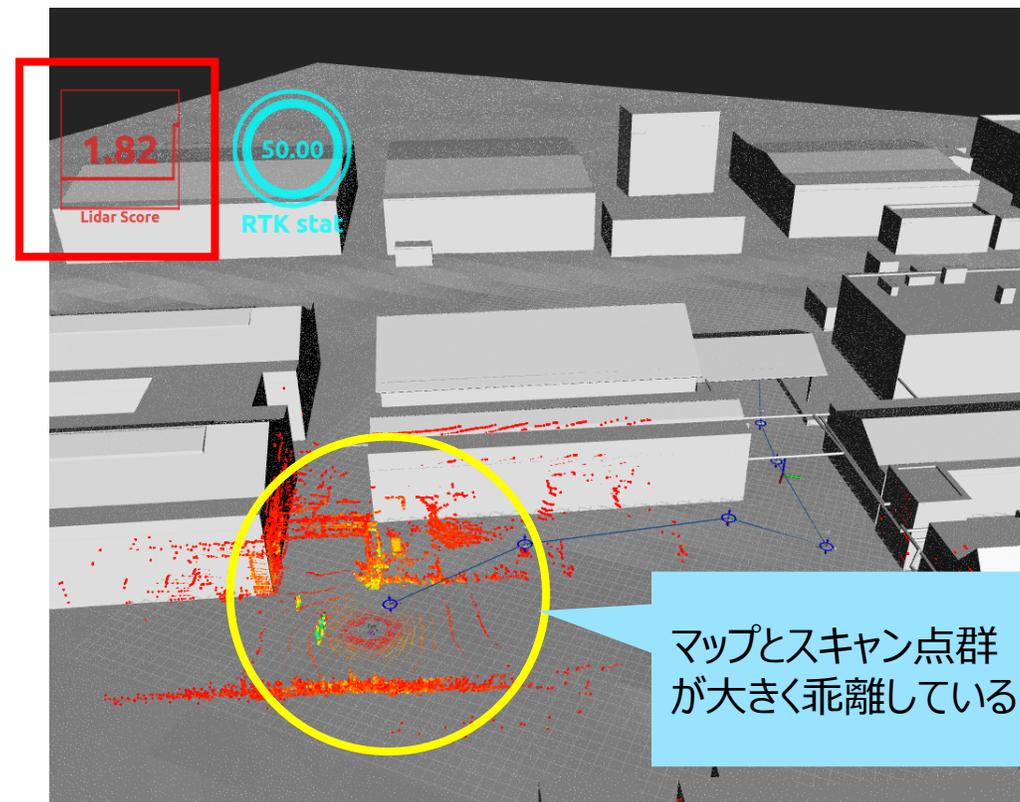
# Ⅲ. 実証システム > 5. アルゴリズム

## LiDAR ローカリゼーション信頼性評価の条件

### スキャンが不十分な際の信頼性評価

- ドローンが上空を飛行する際に**地面や周囲の建築物との距離が遠くなることでスキャン点群の数が減少する場合**において、実際には自己位置推定がうまくいっていないにもかかわらず**信頼値が高いと誤判断するケースあり**
  - このようなケースを避けるためには**周囲を広範囲に計測できていることが保証される必要あり**
    - LiDARにより一部の周囲環境のみを計測できた場合（例えば建築物の壁の一面のみがスキャン点群として得られた等）地図上のどこに位置合わせをしても良い結果が得られる
- ↓
- 簡易的対策としてボクセル化した後のLiDARスキャン点群が**1万ポイント以上の点を持つ場合のみ、信頼値を算出する**
    - 1万ポイントの根拠：実データを観察・解析した結果数千ポイント以下になると上記問題が起こることが判明し、安全側に寄せて1万ポイントを設定

### LiDAR自己位置推定時の信頼値が悪化しているケース



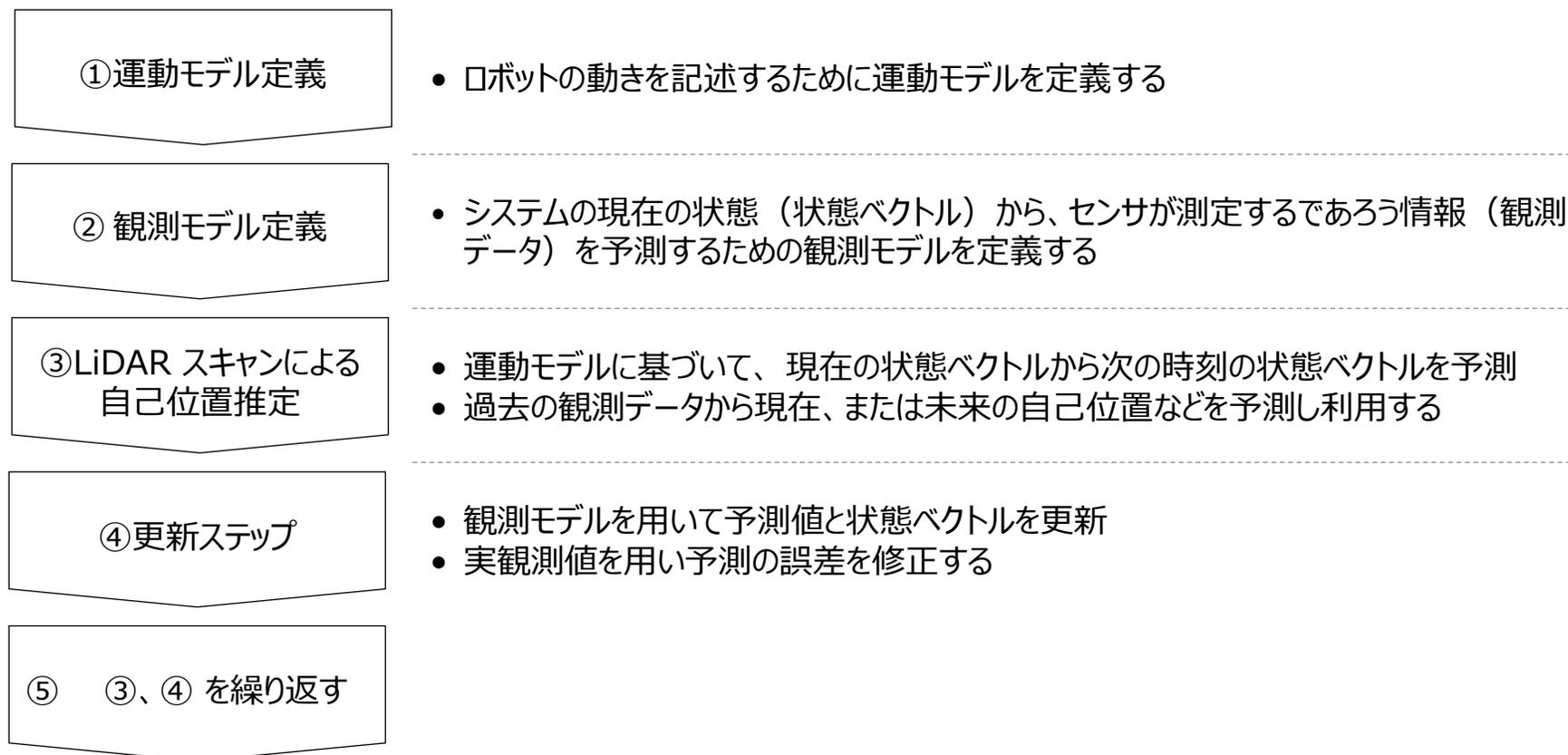


# Ⅲ. 実証システム > 5. アルゴリズム

## 測位情報の統合

ドローンの自己位置推定において、センサデータ（GNSS、LiDAR、IMUなど）を統合して精度を向上させるため、拡張カルマンフィルタ（EKF：Extended Kalman Filter）を利用する

### EKF フロー



## Ⅲ. 実証システム > 5. アルゴリズム

### ① 運動モデル定義

#### 運動モデルについて

運動モデルとはロボットの動きを記述するために定義されるシステムモデル

- ドローンの運動モデルは、6自由度（3次元位置と3次元姿勢）を考慮する
  - ロボットの動きは非線形であり、状態変数（位置、速度、姿勢）および入力変数（位置、速度、加速度）を含む方程式で表現
  - 状態ベクトル  
 $[X, Y, Z, roll, pitch, yaw, X', Y', Z', roll', pitch', yaw', X'', Y'', Z'']$   
 それぞれ位置、姿勢、速度、角速度、加速度を表す

#### ドローンの運動モデル

##### 1. 位置の更新:

$$\begin{aligned} X_k &= X_{k-1} + X'_{k-1}\Delta t + 0.5X''_{k-1}\Delta t^2 \\ Y_k &= Y_{k-1} + Y'_{k-1}\Delta t + 0.5Y''_{k-1}\Delta t^2 \\ Z_k &= Z_{k-1} + Z'_{k-1}\Delta t + 0.5Z''_{k-1}\Delta t^2 \end{aligned}$$

##### 2. 姿勢の更新:

$$\begin{aligned} roll_k &= roll_{k-1} + roll'_{k-1}\Delta t \\ pitch_k &= pitch_{k-1} + pitch'_{k-1}\Delta t \\ yaw_k &= yaw_{k-1} + yaw'_{k-1}\Delta t \end{aligned}$$

##### 3. 速度の更新:

$$\begin{aligned} X'_k &= X'_{k-1} + X''_{k-1}\Delta t \\ Y'_k &= Y'_{k-1} + Y''_{k-1}\Delta t \\ Z'_k &= Z'_{k-1} + Z''_{k-1}\Delta t \end{aligned}$$

##### 4. 角速度の更新:

$$\begin{aligned} roll'_k &= roll'_{k-1} \\ pitch'_k &= pitch'_{k-1} \\ yaw'_k &= yaw'_{k-1} \end{aligned}$$

##### 5. 加速度の更新:

$$\begin{aligned} X''_k &= X''_{k-1} \\ Y''_k &= Y''_{k-1} \\ Z''_k &= Z''_{k-1} \end{aligned}$$

$\Delta t$ はサンプリング間隔（前回の状態からの経過時間）を表す

# Ⅲ. 実証システム > 5. アルゴリズム

## ② 観測モデル定義

### 観測モデルについて

観測モデルは、システムの現在の状態（状態ベクトル）から、センサが測定するであろう情報（観測データ）を予測する方程式

- 1 : IMUからは加速度・角速度
- 2 : GNSSからは基準座標系に変換された後の位置
- 3 : LiDARローカリゼーションからは位置と姿勢

- 観測モデルから求められる観測方程式は右。また観測情報全体を示す観測ベクトルは以下

$[x_{\text{lieark}}, y_{\text{lieark}}, z_{\text{lieark}}, r_{\text{lieark}}, p_{\text{lieark}}, y_{\text{lieark}}, x_{\text{posk}}, y_{\text{posk}}, z_{\text{posk}}, \omega_{\text{rollk}}, \omega_{\text{pitchk}}, \omega_{\text{yawk}}, x_k, a_{yk}, a_{zk}]$

- LiDAR自己位置の位置・向き
- GNSSによる自己位置
- IMUによる角速度・加速度

### 観測モデル

IMU IMUデータはロボットの線形加速度と角速度を測定（観測モデルは以下）

加速度:

$$a_{xk} = \ddot{X}_k + v_{axk},$$

$$a_{yk} = \ddot{Y}_k + v_{ayk},$$

$$a_{zk} = \ddot{Z}_k + v_{azk}$$

角速度:

$$\omega_{\text{rollk}} = \text{roll}'_k + v_{\text{rollk}},$$

$$\omega_{\text{pitchk}} = \text{pitch}'_k + v_{\text{pitchk}},$$

$$\omega_{\text{yawk}} = \text{yaw}'_k + v_{\text{yawk}}$$

ここで、 $v_{axk}, v_{ayk}, v_{azk}, v_{\text{rollk}}, v_{\text{pitchk}}, v_{\text{yawk}}$ はそれぞれ加速度と角速度の観測ノイズを表す

GNSS GNSSデータは、ロボットの3次元位置（ $X, Y, Z$ ）を直接測定

$$x_{\text{posk}} = X_k + v_{x\text{posk}},$$

$$y_{\text{posk}} = Y_k + v_{y\text{posk}},$$

$$z_{\text{posk}} = Z_k + v_{z\text{posk}}$$

ここで、 $v_{x\text{posk}}, v_{y\text{posk}}, v_{z\text{posk}}$ はそれぞれ $X, Y, Z$ の観測ノイズを表す。

LiDAR LiDARによる自己位置推定は、ロボットの位置（ $X, Y, Z$ ）および姿勢（ $\text{roll}, \text{pitch}, \text{yaw}$ ）を測定する

位置:

$$x_{\text{lieark}} = X_k + v_{x\text{lieark}},$$

$$y_{\text{lieark}} = Y_k + v_{y\text{lieark}},$$

$$z_{\text{lieark}} = Z_k + v_{z\text{lieark}}$$

姿勢:

$$r_{\text{lieark}} = \text{roll}_k + v_{r\text{lieark}},$$

$$p_{\text{lieark}} = \text{pitch}_k + v_{p\text{lieark}},$$

$$y_{\text{lieark}} = \text{yaw}_k + v_{y\text{lieark}}$$

ここで、 $v_{x\text{lieark}}, v_{y\text{lieark}}, v_{z\text{lieark}}, v_{r\text{lieark}}, v_{p\text{lieark}}, v_{y\text{lieark}}$ はそれぞれ $X, Y, Z, \text{roll}, \text{pitch}, \text{yaw}$ の観測ノイズを表す

# Ⅲ. 実証システム > 5. アルゴリズム

## ③ 予測ステップ

### 運動モデルによる予測ステップ

予測ステップでは、運動モデルを使用してある時刻までの状態ベクトルと観測結果から、次のタイムステップにおける状態ベクトルを予測する

目的となる状態ベクトルは

$[X, Y, Z, roll, pitch, yaw, X', Y', Z', roll', pitch', yaw', X'', Y'', Z'']$

- 予測ステップでは、運動モデルを元に次の時刻状態を予測する
- 次の時刻状態を予測するための状態方程式は以下  
この時の時刻は $t-1$ であり次のステップ時刻である時刻 $t$ の状態に関して予測を行う

$$x_t = f(x_{t-1}, u_{t-1})$$

- $x_t$  は時刻 $t$ における状態ベクトル
- $u_t$  は時刻 $t$ における制御に入力。ただし  $u_t=0$ とする

$f()$  は状態遷移関数。  $u_t=0$ のため先述の運動モデルと同じになる

### 予測結果から状態誤差共分散行列を更新

- 状態方程式による予測を行ったのち、状態遷移関数のヤコビ行列を計算し、状態誤差共分散行列を更新する

- 状態誤差共分散行列の更新は以下のように行われる

$$P_t = F_t P_{t-1} F_t^T + Q_t$$

- $P_t$  は時刻 $t$ における時刻 $t$ における状態誤差共分散行列
- $F_t$  は時刻 $t$ におけるヤコビ行列
- $Q_t$  は時刻 $t$ におけるプロセスノイズ行列

## Ⅲ. 実証システム > 5. アルゴリズム

### ④ 更新ステップ

#### 更新ステップ

実観測値を用いた状態予測誤差の補正が目的に、観測データを用いて、状態変数および共分散行列の予測値修正を更新ステップで行う

- ① 更新ステップでは、まずカルマンゲイン（実データと予測データから生じた差分をどの程度状態ベクトルに反映するかの影響度を定めるパラメータ）を計算。併せて、観測ベクトルに対するヤコビ行列を計算する

$$[x_{lieark}, y_{lieark}, z_{lieark}, r_{lieark}, p_{lieark}, y_{lieark}, x_{posk}, y_{posk}, z_{posk}, \omega_{rollk}, \omega_{pitchk}, \omega_{yawk}, x_k, a_{yk}, a_{zk}]$$

- ② カルマンゲインと観測予測誤差を使用して状態ベクトルの更新を行う  
観測モデルと予測された状態ベクトルに基づき、観測データの予測値を計算し、実観測データとの差分から観測予測誤差を計算
- ③ カルマンゲインを用いて状態誤差共分散行列の更新を行う

#### 更新ステップの計算式

- ① カルマンゲインの計算

$$K_t = P_t|_{t-1} H_t^T (H_t P_t|_{t-1} H_t^T + R_t)^{-1}$$

- $P_t|_{t-1}$  は時刻  $t-1$  における共分散行列の予測値
- $H_t$  は時刻  $t$  での観測モデルのヤコビ行列
- $R_t$  は時刻  $t$  での観測ノイズ共分散行列

- ② 状態ベクトルの更新

$$y_t = z_t - H_t x_t|_{t-1}$$

- $z_t$  は時刻  $t$  の実際の観測値、
- $x_t|_{t-1}$  は時刻  $t-1$  における状態ベクトルの予測値)

- ③ 共分散行列の更新

$$P_t|_t = (I - K_t H_t) P_t|_{t-1}$$

- $P_t|_t$  は時刻  $t$  における更新された共分散行列
- $I$  は単位行列

これにより、観測データが状態ベクトルと共分散行列に反映され、反映された値が次の予測ステップで使用される

# Ⅲ. 実証システム > 6. データ > ①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
地形情報LOD1	dem:TINRelief	空間属性	dem:tin	地面を構成する各点の座標

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

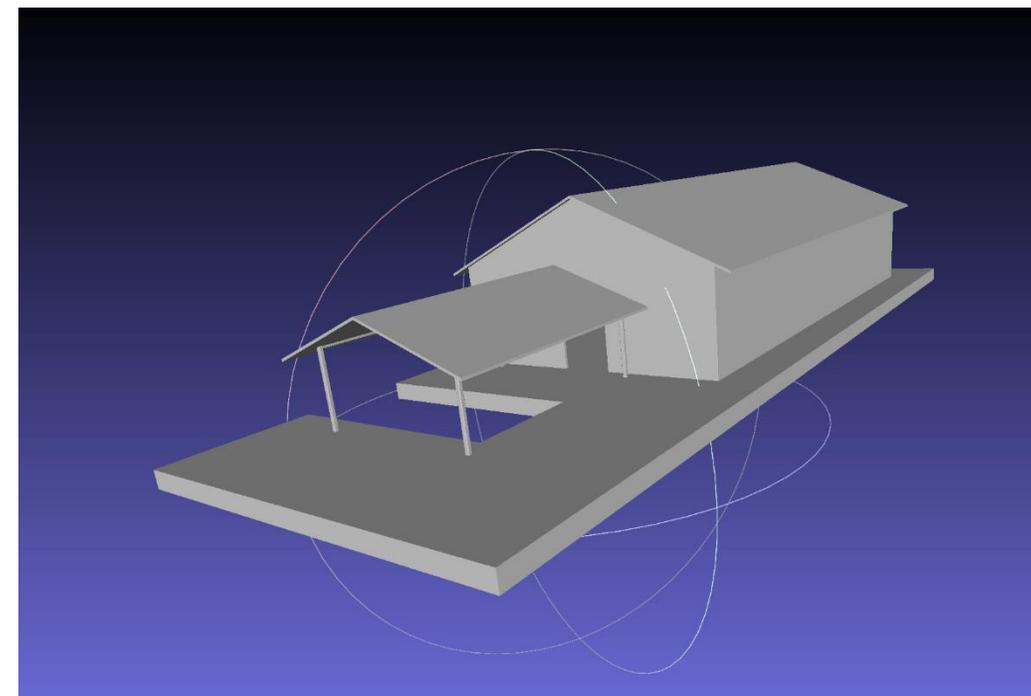
活用データ	内容	データ形式	出所
建築BIM	建物3Dモデル	IFC	株式会社センシンロボティクス
LiDARスキャンデータ	LiDARデバイスが計測したスキャンデータ	ROSメッセージ	3DLiDARセンサー
GNSS測位座標	WGS (World Geodetic System) 84測地系での測位データ	NMEA	GNSSユニット
IMU (機体) 加速度/角速度	3軸のジャイロと3方向の加速度計によって得られる、3次元の角速度と加速度	ROSメッセージ	IMUユニット

# Ⅲ. 実証システム > 6. データ > ①活用データ 建物データ

## 概要

項目	内容
対象建物	ENEOS川崎事業所 倉庫1棟 (神奈川県川崎市川崎区扇町)
作成ソフトウェア	Revit
データ形式	IFC (2022年7月センシロボティクス作成)

## イメージ



# Ⅲ. 実証システム > 6. データ > ①活用データ LiDARスキャンデータ

## 概要

項目	内容
メーカー	OUSTER社
モデル・型式	OS-0 64
選定理由	<ul style="list-style-type: none"><li>2022年時点において、ドローンに搭載可能なサイズ</li><li>重量 445g</li><li>同クラスの機種の中で最も視野角が広い(90°)</li></ul>
使用詳細	<ul style="list-style-type: none"><li>垂直FoV : 90°</li><li>測距レンジ : 50m</li><li>精度 : ±1.0-5.0cm</li><li>1秒当たりの回転数 : 10-20回</li><li>最大点群取得数/秒 : 1310720</li></ul>

## センサーイメージ/取得点群イメージ



サンプル：取得点群

# Ⅲ. 実証システム > 6. データ > ① 活用データ GNSS測位座標

## 概要

項目	内容
メーカー	DJI社
モデル・型式	MATRICE300RTK 付属GNSS (RTKユニット)
捕捉衛星	<ul style="list-style-type: none"><li>• GPS</li><li>• GLONASS</li><li>• BeiDou</li><li>• Galileo</li></ul>
RTKが有効でFIXステータスの場合の精度	1 cm + 1 ppm* (水平方向) 1.5 cm + 1 ppm (垂直方向)

## センサーイメージ



測位ユニット

\*ppmは 100万分の1をあらわす単位、1 ppmは、測距 1 kmの場合について 1 mmを表す

# Ⅲ. 実証システム > 6. データ > ①活用データ IMU（機体）加速度/角速度

## データ概要

項目	内容
メーカー	DJI社
モデル・型式	MATRICE300RTK 付属IMUユニット
取得可能データ	<ul style="list-style-type: none"><li>• 加速度</li><li>• 角速度</li></ul>

## センサーの諸元



IMUセンサーユニットは、  
機体内部に組み込み

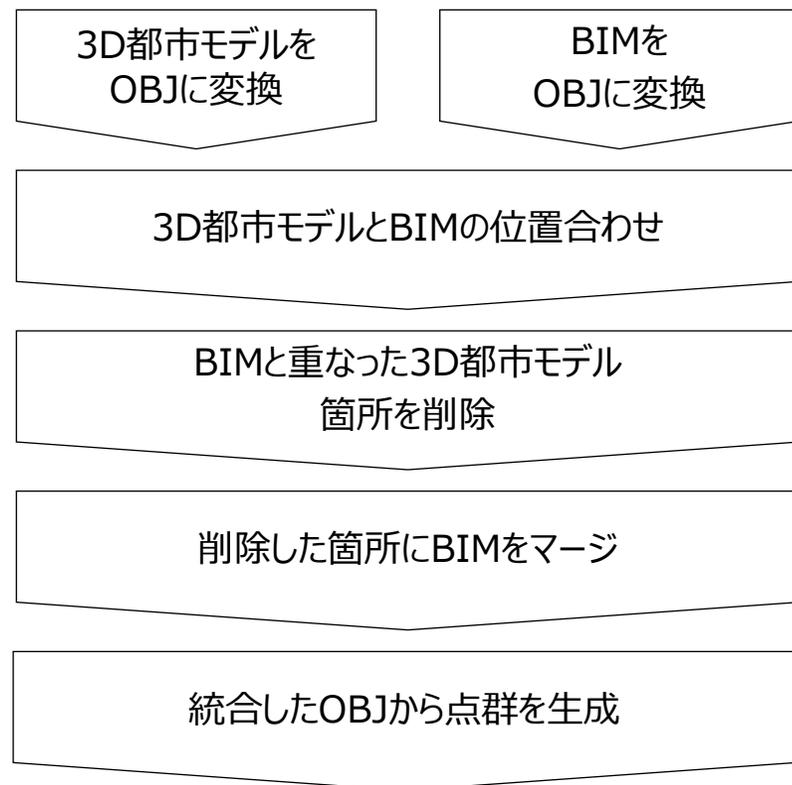
# Ⅲ. 実証システム > 6. データ > ②データ処理 データ処理 | 一覧

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)
統合マップ (obj/pcd形式)	経路計画処理への入力 LiDARローカリゼーションへの 入力	<ul style="list-style-type: none"><li>• CityGML形式/IFC形式から obj/pcd形式への変換</li></ul>	CityGMLtoRobot Map	<ul style="list-style-type: none"><li>• 3D都市モデル (CityGML形式)</li><li>• 建築BIM (IFC)</li></ul>

# Ⅲ. 実証システム > 6. データ > ②データ処理 統合マップの作製 (1/2)

3D都市モデルとBIMをマージ、LiDARローカリゼーションのためのマップとして利用

CityGMLtoRobotMapによる処理フロー



概要

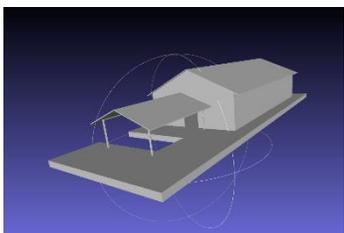
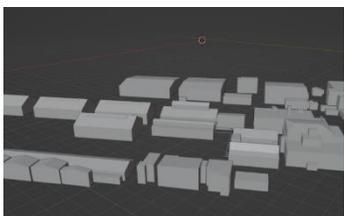
- 開発したCityGMLtoRobotMapを用いて、ドローンのローカリゼーションに用いるマップを作製
- CityGMLとBIMを統合することでCityGMLには反映されていない開口部などが再現、屋内外のシームレス飛行時に、屋外から屋内への侵入が可能となる

フロー中の処理と使用ツールの対応表

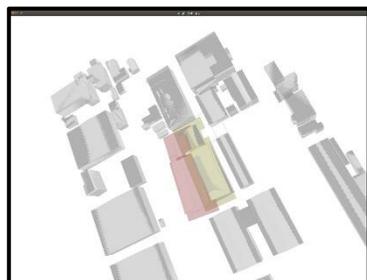
処理	使用ソフトウェア・ライブラリ
3D都市モデルをOBJに変換	CityGMLtoRobotMap
BIMをOBJに変換	IfcConverter
3D都市モデルとBIMの位置合わせ	CityGMLtoRobotMap
BIMと重なった3D都市モデル箇所を削除	Blender
削除した箇所にBIMをマージ	CityGMLtoRobotMap
統合したOBJから点群を生成	CityGMLtoRobotMap

# Ⅲ. 実証システム > 6. データ > ②データ処理 統合マップの作製 (2/2)

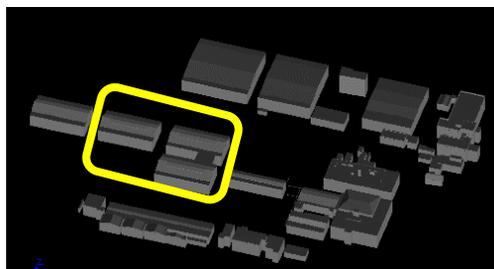
3D都市モデル/BIMを  
OBJへ変換



3D都市モデルとBIMの  
位置合わせ



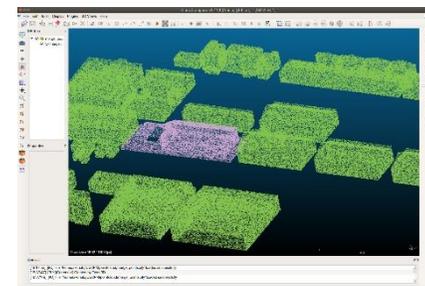
BIMと重なった3D都市モデル  
箇所を削除



削除した箇所にBIMをマージ



統合したOBJから点群を生成

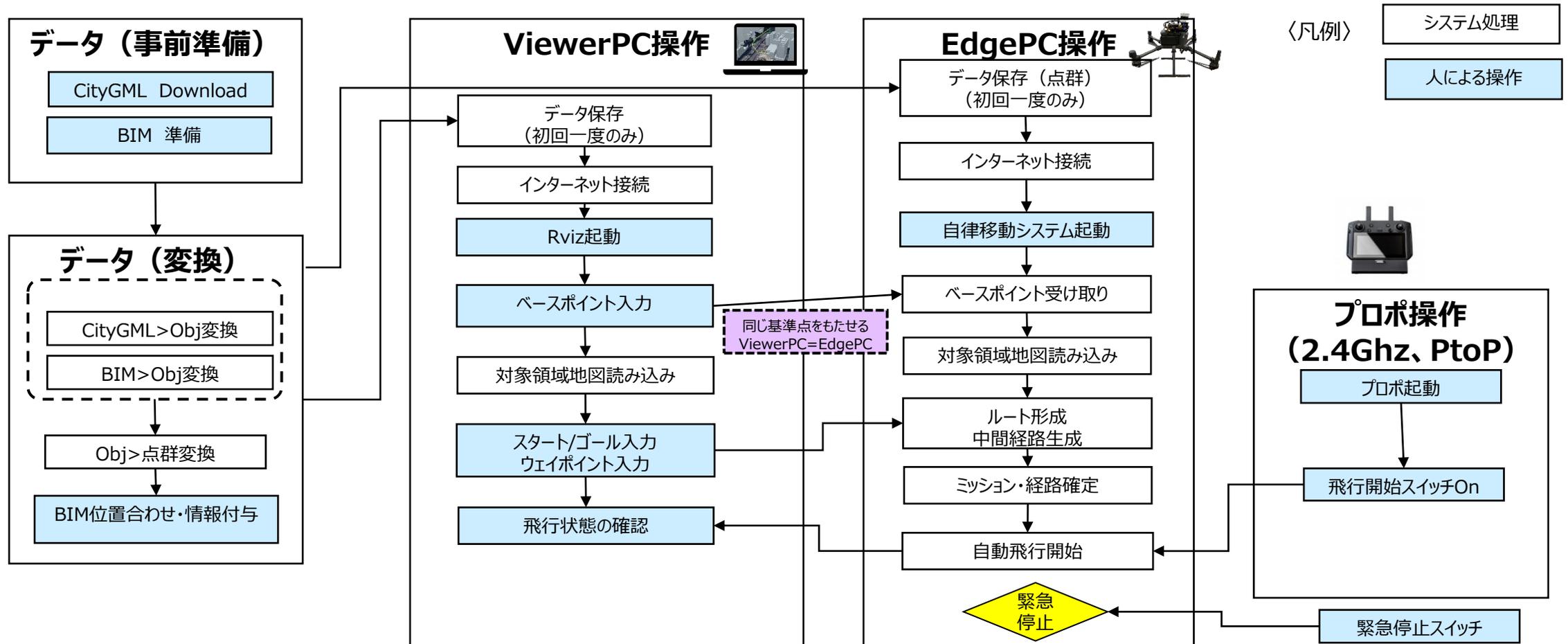


# Ⅲ. 実証システム > 6. データ > ③出力データ

## ③出力データ | 一覧

出力データ	内容	データ形式
-	-	-

# Ⅲ. 実証システム > 7. ユーザインタフェース 自律飛行実施のフロー

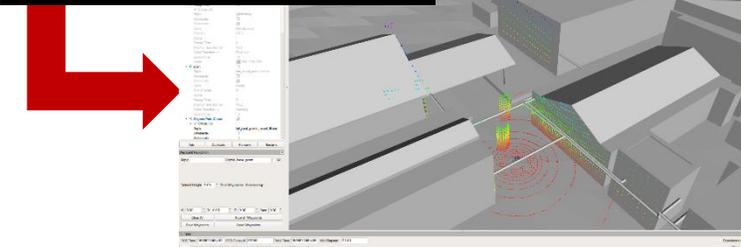


# Ⅲ. 実証システム > 7. ユーザーインターフェース 自律飛行ルートの作成手順 (Rviz)

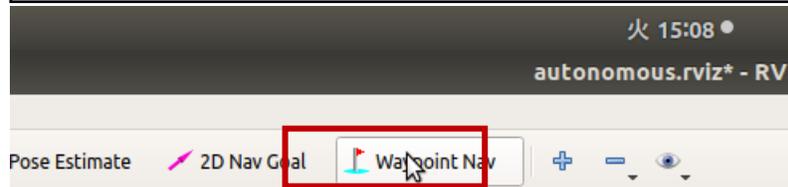
Rvizを利用してウェイポイント（経由点）を設置し、複数設置したポイントを経由しながらドローンが自律飛行するように飛行ルートを作成する

## ① : Rvizを起動

```
$ rosrn rviz rviz
```

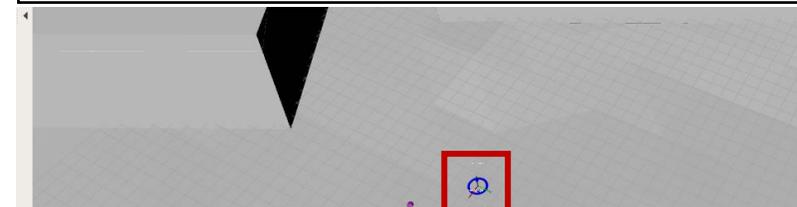


## ② : ウェイポイント（経由点）設定



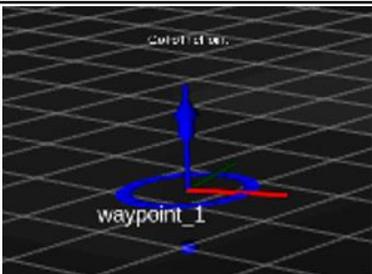
メニューの【Waypoint Nav】をクリックする

## ③ : ウェイポイント設置



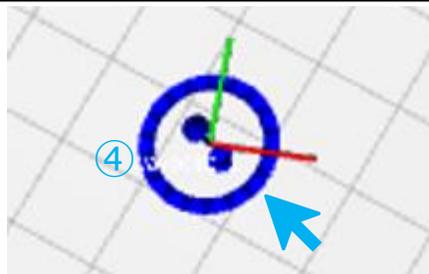
MAP上の任意の点をクリックすると青いリングが表示され、ウェイポイントが設置される

## ④ : 高度調整



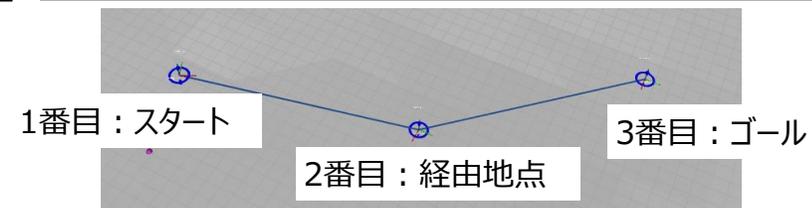
ウェイポイントの縦方向のピラーを上下にドラッグさせることで高度の調整を行う

## ⑤ : 平面移動



ウェイポイント（青のリング）をドラッグして移動させる

## ⑥ : 複数ウェイポイント設定



②～⑤を繰り返すことで、複数のウェイポイントが設定、作成順に飛行順序となる

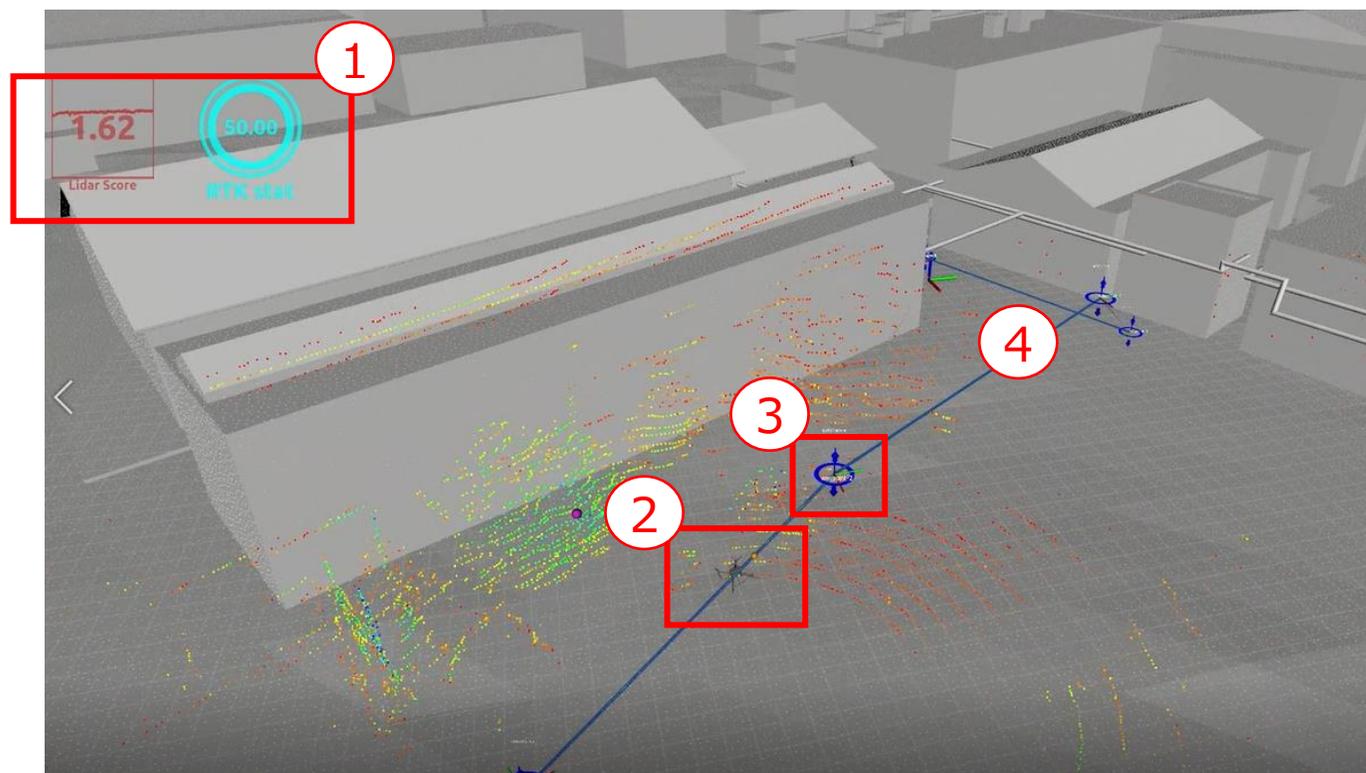
# Ⅲ. 実証システム > 7. ユーザインタフェース ドローン位置可視化

屋外の表示

表示詳細

番号	内容
1	GNSS・LiDARステータス
2	ドローン現在位置
3	ウェイポイント
4	青線 飛行ルート

Rviz画面



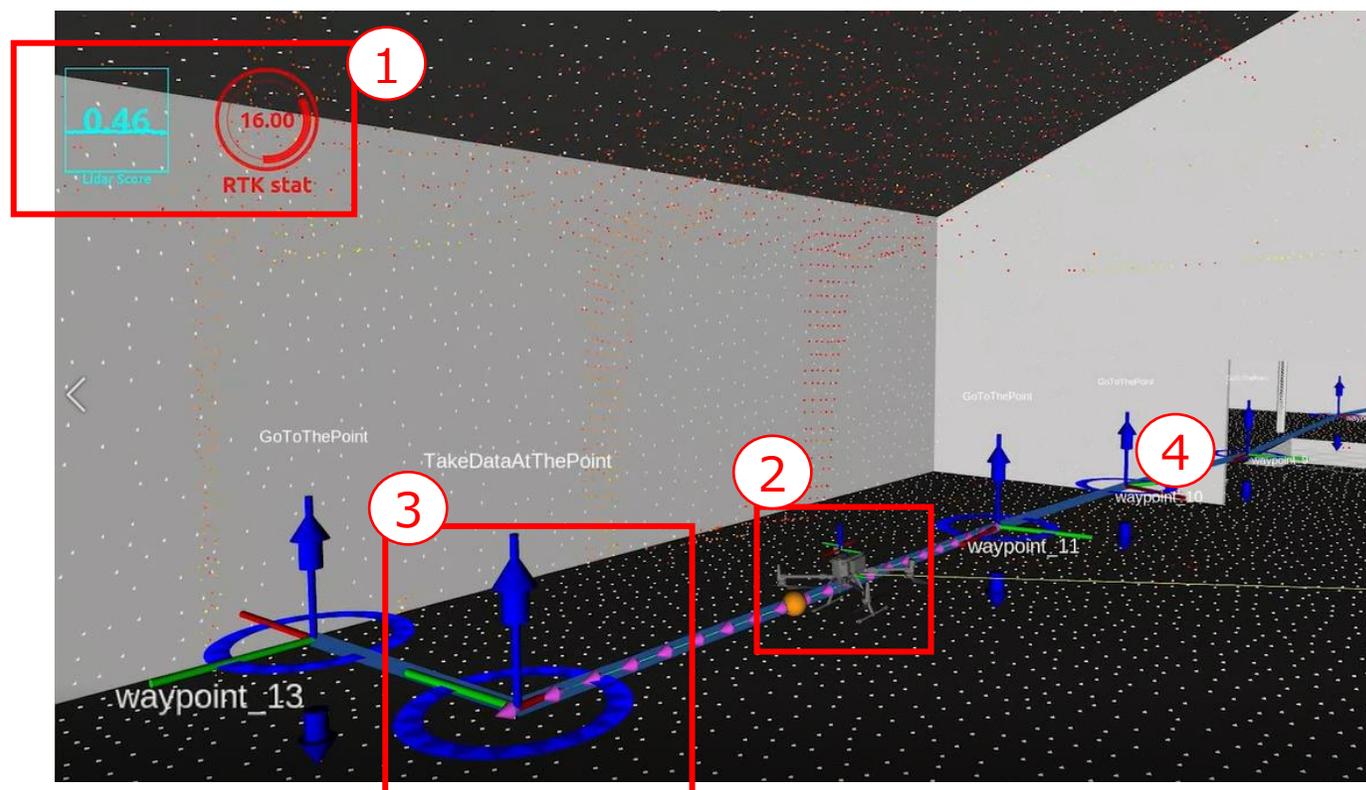
# Ⅲ. 実証システム > 7. ユーザインタフェース ドローン位置可視化

屋内の表示

表示詳細

番号	内容
1	GNSS・LiDARステータス
2	ドローン現在位置
3	ウェイポイント
4	青線 飛行ルート

Rviz画面



# 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
obj形式に変換	ENEOS川崎事業所のCityGMLのLOD2とENEOS川崎事業所内になる建屋1棟のBIMモデルをobjファイル形式に変換できるか	OK
データの統合	上記で変換したobjを統合して1つのMAPにできるか	OK
RTK-GNSSの位置測位	RTK-GNSSを使った位置計測ができるか	OK
RTK-GNSSのステータスチェック	RTK-GNSSのステータスが正しく取得できているかRviz画面上にて確認できるか	OK
LiDAR点群取得とステータスチェック	LiDARを起動させた状態で点群とステータスが正しく取得できているかの確認ができるか	OK
シミュレーションテスト	MAPとウェイポイントを設定して一連の自律飛行をRviz画面にてシミュレーション飛行させて確認できるか	OK
フライト成功の検証	経路計画で作成した飛行ルートを実行させ、スタート地点から自律飛行にて、ゴール地点まで到達できるか	OK

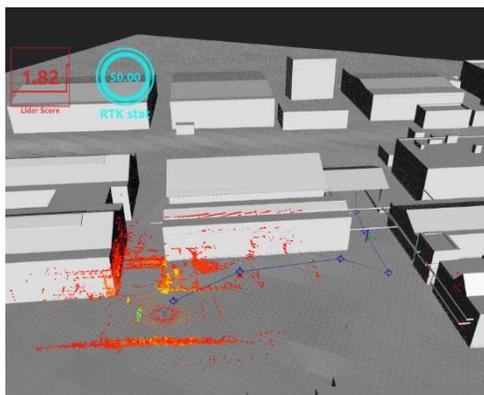


# Ⅲ. 実証システム > 8. システムテスト結果 フライトテスト結果

合計13か所のウェイポイントを設定して飛行するフライトテストを7回実施したところ、全件で自律飛行でゴール地点まで到達、事前のマップ作製を行わずに3D都市モデル/BIMから作成するマップを用いた自律飛行システムの優位性が実証

## フライトテスト概要

- 事前準備：経路計画作成
  - スタート/経由地点/ゴールで合計13か所のウェイポイントを設定し飛行ルートを作成する
- 検証内容
  - 作成した経路計画を使用して自律飛行を複数回実施し成功回数を計測する
- 成功条件
  - スタート地点から自律飛行にて、ゴール地点まで到達できれば成功、到達できない場合は失敗とする



ルートを設定した画面

## フライトテスト結果

回	スタート時刻	ゴール到達時刻	主な参加者	場所	結果
1	2022年11月18日 15時52分	2022年11月18日 15時58分	センシンロボティクス	ENEOS川崎事業所内	成功
2	2022年11月18日 16時10分	2022年11月18日 16時15分	センシンロボティクス	ENEOS川崎事業所内	成功
3	2022年11月22日 16時10分	2022年11月22日 16時16分	センシンロボティクス	ENEOS川崎事業所内	成功
4	2022年11月22日 16時10分	2022年11月22日 16時16分	センシンロボティクス	ENEOS川崎事業所内	成功
5	2022年11月28日 11時45分	2022年11月28日 11時50分	竹中工務店 ・センシンロボティクス	ENEOS川崎事業所内	成功
6	2022年11月28日 13時40分	2022年11月28日 13時46分	国土交通省・アクセンチュア・竹中工務店・センシンロボティクス	ENEOS川崎事業所内	成功
7	2022年11月28日 14時00分	2022年11月28日 14時06分	国土交通省・アクセンチュア・竹中工務店・センシンロボティクス	ENEOS川崎事業所内	成功

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 検証実施概要

目的	ドローンの自己位置推定精度を複数回の飛行を試行し、指定位置到達の再現性を検証する
実施期間	2022年11月22日/11月28日
実施場所	神奈川県川崎市川崎区扇町 ENEOS川崎事業所
実施内容	指定ポイントにArUcoマーカを設置し、ArUcoマーカ的位置 (x,y,z) を計測する • ArUcoマーカ上にて10秒間ホバリングを行う • ドローン搭載カメラにて1秒間に15枚の画像を撮影 • 合計150枚の画像から、x,y,zそれぞれの平均値を算出する • 1回目、2回目の平均値を比較し再現性を確認する

# 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

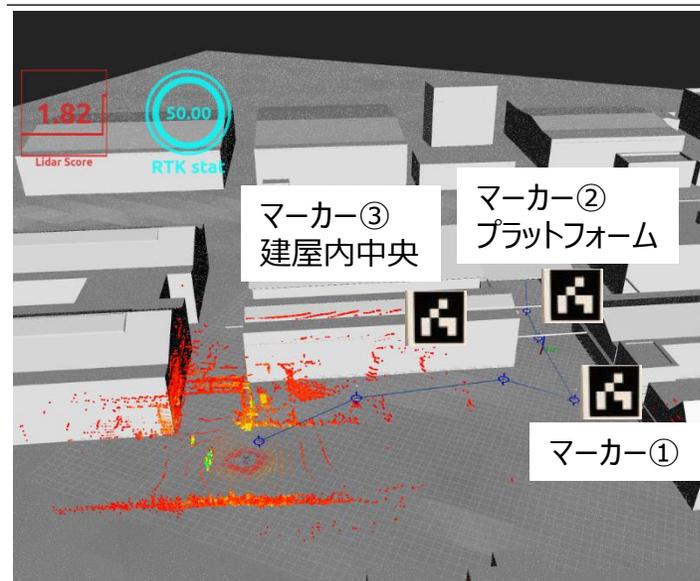
## 自己位置推定の精度検証

ドローンの自己位置推定精度を複数回の飛行による位置の再現性にて検証を行う

### 検証フロー

マーカー設置	飛行ルートのウェイポイント3か所にマーカーを設置する (①~③)
自律飛行1回目	1回目の自律飛行を実施
相対位置計測	3か所のマーカー上空で10秒間ホバリングして、マーカーとドローンとの相対位置を計測する
自律飛行2回目	2回目の自律飛行を実施
相対位置計測	3か所のマーカー上空で10秒間ホバリングして、マーカーとドローンとの相対位置を計測する
比較検証	1回目と2回目のArUcoマーカー3つとそれぞれ比較してセンチメートル単位の差分を算出する

### UI画面/機体飛行中の画像



ArUcoマーカー上空で10秒間ホバリングして計測

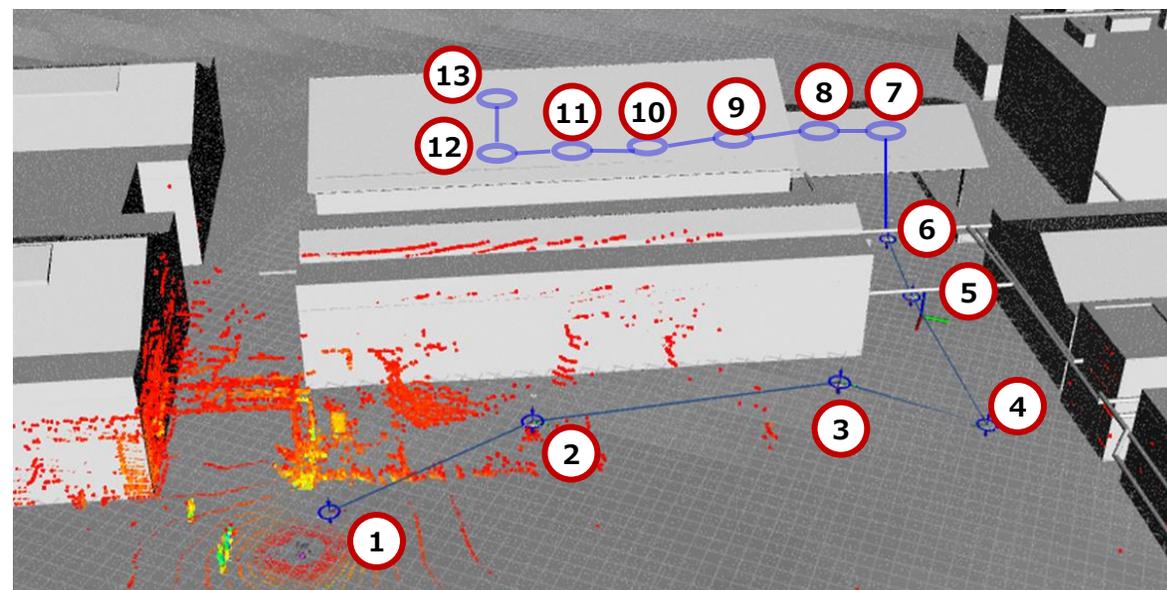
# IV. 実証技術の検証 > 1. 自己推定位置精度の検証 > ①検証内容 各ウェイポイント詳細

屋外から屋内へシームレスに飛行するルートを設定（①-⑤は屋外、⑥-⑧屋根下、⑨-⑬は屋内）

ウェイポイント概要

#	場所	設置目的
①	屋外	スタート地点
②		第1ウェイポイントから上昇して高度8mに到達するポイント
③		第2ウェイポイントから高度8mを維持して水平飛行を行わせるため
④		高度2m 左へ方向転換するポイント
⑤		高度2m 地上8mに配管があり、配管の下を飛行させる
⑥	屋根下	建屋の屋根の下に到達しRTK-GNSSの受信が途絶えるポイント
⑦		建屋開口部（高さ3.5m幅3.5m）を通過するにあたりドローンの侵入経路を確保するため
⑧		開口部を正面としてLiDARから情報を得るため
⑨	建屋内部	開口部を通過した先の目標ポイントとして
⑩		RTK-GNSSが受信できない為、⑫までの間に設置
⑪		RTK-GNSSが受信できない為、⑫までの間に設置
⑫		RTK-GNSSが受信できない環境下での方向転換
⑬		ゴール地点

ウェイポイントの設定位置



# 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 ウェイポイントでの検証点

屋外での自律飛行



#①～⑤

屋根下の自律飛行



#⑥～⑧

開口部を通過



#⑨～⑬

ウェイ  
ポイント

検証点

RTKと3D都市モデルを使ったLiDARを併せた自律飛行が可能か

RTKの受信環境が悪い状況で、3D都市モデルを使ったLiDARのみで自律飛行が可能か

3D都市モデルとBIMデータの接続部においてLiDARのみで自律飛行が可能か



# 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

## ArUcoマーカーによる精度の算出方法

ArUcoマーカーによって地上に設置したマーカーとドローンの測定位置を3か所の測定ポイントで計測し、2回のフライト間の差分から、精度の評価をXYZ別に評価を行う

ArUcoマーカーによる精度の算出方法詳細

測定方法の概要	<ul style="list-style-type: none"> <li>ArUcoマーカーを使い、自己位置推定によって自律飛行するドローンの位置精度を設定した各ウェイポイントで評価</li> </ul>
測定ポイント	ルート上に3か所
同ポイント上でのホバリング時間	10秒
同ポイント上での計測回数	150回
マーカーとドローンの相対位置の算出方法	<ul style="list-style-type: none"> <li>150回の計測結果から、それぞれX/Y/Zの結果を集計して、平均値を算出</li> </ul>
精度の算出方法	<ul style="list-style-type: none"> <li>2回のフライトの相対位置の算出結果から得られた位置情報の差分を算出</li> <li>3つの測定ポイントの結果を平均し、精度（再現性）を算出</li> </ul>

測定ポイントにおける精度の算出結果の一例

	1回目	2回目	差分
X	179mm	159mm	20mm
Y	75mm	86mm	11mm
Z	1,336mm	1,356mm	20mm

※数値の範囲は路面に設置したマーカーのサイズ、カメラの解像度、カメラのレンズ画角によって設定する必要がある。今回はXYZそれぞれ1m以上3m以内で設定

# 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 飛行再現性検証サマリ

複数回の飛行を実施

ArUcoマーカを設置したポイント①②③にて算出した平均値と差分を計算

KPI		達成度・結果 (差分の平均値)	示唆
事前のLiDARスキャンによるマップを用いずに、3D都市モデル/BIMを用いてLiDARローカリゼーションによる自己位置推定を用いた自律飛行の再現性	X方向	35mm	<ul style="list-style-type: none"> <li>3D都市モデル（LOD2）とBIMを統合した自動飛行用のMAPを作成して活用することで、事前飛行によるMAP作成をすることなく限定された条件下で自動飛行ができることが分かった</li> <li>屋外から屋内への飛行ルート設定を同一インターフェース上で容易に設定できた</li> <li>Z方向はドローンの回転翼による推力方向と一致するため、制御性の高さから精度が高いことが見込まれる。よって、飛行動作を考慮しない自己位置推定のみの精度の高さはXYZともにこのZ方向の結果を超える精度が得られていることが期待される</li> </ul>
	Y方向	60mm	
	Z方向	9mm	



# 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

## 自己位置推定の精度結果

ArUcoマーカを使った検証により、XYZ方向それぞれの精度を評価した結果いずれも100mm未満の結果が得られることが分かり、中でもドローンの推力に平行なZ方向は10mm未満の精度が得られた

X方向の精度検証結果

Y方向の精度検証結果

Z方向の精度検証結果

	ウェイポイント		
	①	②	③
1回目	14mm	29mm	37mm
2回目	5mm	117mm	30mm
<b>差分</b>	9mm	88mm	7mm

	ウェイポイント		
	①	②	③
1回目	210mm	112mm	120mm
2回目	235mm	6mm	169mm
<b>差分</b>	25mm	106mm	49mm

	ウェイポイント		
	①	②	③
1回目	1,333mm	1,572mm	1,251mm
2回目	1,344mm	1,565mm	1,261mm
<b>差分</b>	11mm	7mm	10mm

X方向の精度の平均値：35mm

Y方向の精度の平均値：60mm

Z方向の精度の平均値：9mm

\*各測定値は、ArUcoマーカとドローンとの相対距離である。Z方向から撮影しているため、距離をとる必要があり、Z方向の距離はXYに比べて大きい

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

## 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"> <li>3D都市モデルを活用することで、現地での事前飛行によるMAP作成の必要なく、仮想空間上でドローン自律飛行用のMAPを作成することができる               <ul style="list-style-type: none"> <li>3D都市モデル（建物LOD2）とBIMを統合した自動飛行用のMAPを作成して活用することで、事前飛行によるMAP作成をすることなく限定された条件下で自動飛行ができることが分かった</li> </ul> </li> </ul>
自動運行ルート	<ul style="list-style-type: none"> <li>本実証にて、3D都市モデルおよびBIMを統合したMAPを用いて、屋外から屋内への飛行ルート設定を同一インターフェース上で容易に設定できた</li> </ul>

# V. 成果と課題 > 1. 今年度の実証で得られた成果

## 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
自律飛行の実現性	<ul style="list-style-type: none"> <li>● 3D都市モデルがあれば、事前の点群MAP作成をせずに、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"> <li>- 3D都市モデルが整備される以前は、自律飛行には事前運行による点群MAP作成が必要不可欠であるが、ビルの間などのGPS測位の難しいエリアでは目視のみでの飛行に頼る必要があった</li> </ul> </li> </ul>
入手・加工の容易性	<ul style="list-style-type: none"> <li>● 3D都市モデルが整備されさえいれば、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"> <li>- 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りうる</li> <li>- 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</li> </ul> </li> </ul>
オープンデータの信頼性	<ul style="list-style-type: none"> <li>● 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</li> </ul>

# 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
上空エリアでの推定精度向上	<ul style="list-style-type: none"> <li>● 上空利用のLTE回線は、一般のLTE回線より通信速度が遅くなるため自己位置推定精度が低下するおそれがある               <ul style="list-style-type: none"> <li>- 上空はドローンから地上へ送信される自己位置情報、点群データなどが遅延し、自己位置推定に影響を及ぼす</li> <li>- 今回は実証できていない10メートル以上の高度ではドローンに搭載したLiDARから周囲の点群情報が更に減少する可能性が高い</li> </ul> </li> </ul>
積載容量の増加	<ul style="list-style-type: none"> <li>● 建設資材等を搬送するためには本年活用したドローンの積載容量は不足していて、ハードウェア自体の技術開発が必要である               <ul style="list-style-type: none"> <li>- 実証では最大積載量2.7kgのドローンにRTK-GNSS、LiDAR、EdgePCを搭載し自動飛行を行うことができたが建築資材運搬には不十分である</li> </ul> </li> </ul>
システムのユーザーインターフェース向上	<ul style="list-style-type: none"> <li>● サービス化に向けてターゲットとするユーザー（資材運搬、運輸、等）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               <ul style="list-style-type: none"> <li>- 本ユースケースでは技術面での実証としたため、システムはシステム開発者で操作できるレベルに留めたが、想定ユーザーに使ってもらえる使い勝手のよいインターフェースを検討する必要がある</li> </ul> </li> </ul>

# 用語集 (1/4)

用語	内容
ア行	
ROS	<ul style="list-style-type: none"><li>• Robot Operating Systemの略</li><li>• 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</li><li>• 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</li></ul>
RTK	<ul style="list-style-type: none"><li>• Real Time Kinematicの略</li><li>• 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</li></ul>
Rviz	<ul style="list-style-type: none"><li>• ROSが提供しているデータの可視化を行うツール</li><li>• センサーから取得した映像や点群、各プログラムの出力結果を簡単に表示することができる</li></ul>
IFC	<ul style="list-style-type: none"><li>• Industry Foundation Classesの略</li><li>• 建物を構成する全てのオブジェクトの体系的な表現方法の仕様</li><li>• BIMのデータを流通させるためのファイル形式</li><li>• 異なるアプリケーション間でモデルデータを受け渡す際にIFCを介することによってデータの損失を低減することが可能</li></ul>
IFC Converter	<ul style="list-style-type: none"><li>• IFCの変換ツール (ライブラリ)</li><li>• IFC形式のファイルからobjファイルへ変換を行う</li></ul>
IMU	<ul style="list-style-type: none"><li>• Inertial Measurement Unitの略</li><li>• 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</li></ul>



# 用語集 (2/4)

用語	内容
ア行	
ArUcoマーカー	<ul style="list-style-type: none"><li>カメラの姿勢推定に使用できるバイナリの正方形の基準マーカー</li><li>主な利点は、検出が堅牢で、迅速かつ簡単であること</li></ul>
ichimill	<ul style="list-style-type: none"><li>Softbank社が提供している高精度測位サービス</li></ul>
Ubuntu	<ul style="list-style-type: none"><li>デスクトップPCやクラウド、インターネットに接続されたあらゆる機器まで、すべての環境において動作可能なオープンソースのソフトウェアオペレーティングシステム</li></ul>
ウェイポイント	<ul style="list-style-type: none"><li>経路上の地点情報</li><li>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</li><li>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</li></ul>
ウェイポイントアクション	<ul style="list-style-type: none"><li>ウェイポイントとして設定したポイント上にて、ドローンの方向転換などを行うアクション</li></ul>
SoC	<ul style="list-style-type: none"><li>System-on-a-chipの略</li><li>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</li></ul>
NMEA	<ul style="list-style-type: none"><li>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</li><li>NMEA0183というフォーマット（NMEAフォーマット）がある</li></ul>
NDT	<ul style="list-style-type: none"><li>Normal Distribution Transformの略</li><li>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</li></ul>
OBJ	<ul style="list-style-type: none"><li>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</li><li>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</li></ul>

# 用語集 (3/4)

用語		内容
カ行	Galileo	<ul style="list-style-type: none"> <li>EUによる全地球航法衛星システム</li> </ul>
	GLONASS	<ul style="list-style-type: none"> <li>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</li> </ul>
サ行	GNSS	<ul style="list-style-type: none"> <li>Global Navigation Satellite Systemの略</li> <li>「全球測位衛星システム」を意味</li> <li>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</li> </ul>
	自己位置推定	<ul style="list-style-type: none"> <li>ある場所にいる自分自身の位置を推定する技術</li> <li>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</li> </ul>
	自己位置測位	<ul style="list-style-type: none"> <li>自己位置を確定すること</li> <li>ドローン自己位置測位では、自己位置推定で得た位置をさらに精度を高めるために、GPS RTKで位置情報を取得した座標を追加することで、位置精度を高めている</li> </ul>
	準天頂衛星	<ul style="list-style-type: none"> <li>Quasi-Zenith Satellite Systemの別名</li> <li>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</li> <li>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</li> </ul>
	SLAM	<ul style="list-style-type: none"> <li>Simultaneous Localization and Mappingの略</li> <li>同時位置推定と地図作成を行う技術</li> <li>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</li> </ul>

# 用語集 (4/4)

用語		内容
タ行	DJI Onboard SDK	<ul style="list-style-type: none"> <li>• 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</li> <li>• 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</li> </ul>
	trimesh	<ul style="list-style-type: none"> <li>• 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</li> </ul>
ハ行	PCD	<ul style="list-style-type: none"> <li>• Point Cloud Dataの略</li> <li>• 点群を扱うファイル形式の一つ</li> </ul>
	VCSEL	<ul style="list-style-type: none"> <li>• 半導体レーザーの一種</li> <li>• 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</li> </ul>
	FIXステータス	<ul style="list-style-type: none"> <li>• LiDAR・GPS RTKの情報を使用可能な状態である場合のステータス</li> </ul>
ラ行	LiDAR	<ul style="list-style-type: none"> <li>• Light Detection And Rangingの略</li> <li>• 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</li> </ul>
	ラストワンマイル	<ul style="list-style-type: none"> <li>• 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</li> <li>• 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</li> </ul>

# 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（ドローン編）

**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。