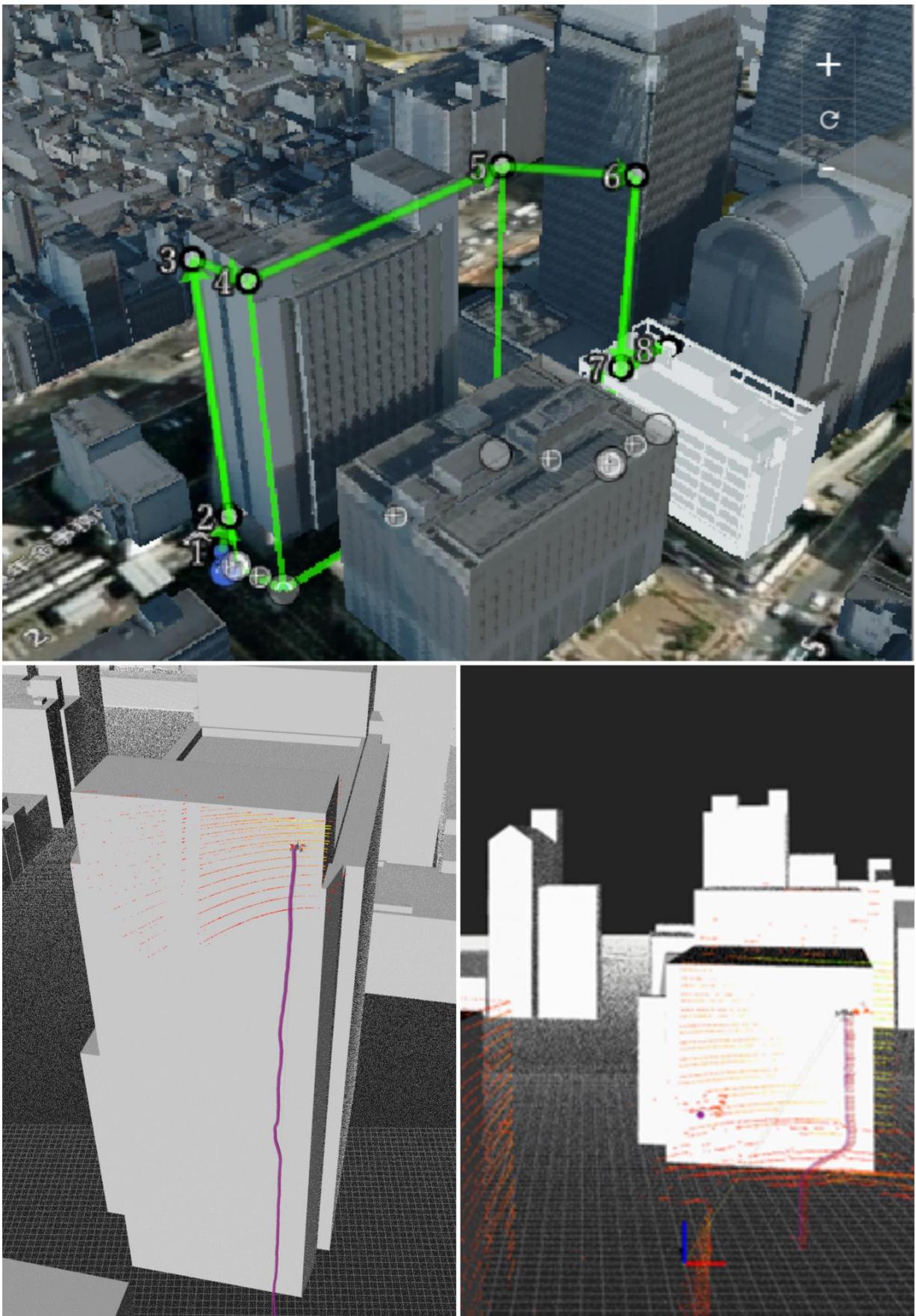




PLATEAU
by MLIT

PLATEAU Technical Report
3D都市モデル活用のための技術資料



3D都市モデルとBIMを活用したモビリティ自律運航

series
No. 86

システム（ドローン）v2.0 技術検証レポート

Technical Report on Autonomous Mobility Operation System Utilizing 3D City Models and BIM Models (Part: UAV) v2.0

目次

1. ユースケースの概要	- 1 -
1-1. 現状と課題	- 1 -
1-2. 課題解決のアプローチ	- 2 -
1-3. 創出価値	- 3 -
1-4. 想定事業機会	- 4 -
2. 実証実験の概要	- 5 -
2-1. 実証仮説	- 5 -
2-2. 実証フロー	- 6 -
2-3. 検証ポイント	- 6 -
2-4. 実施体制	- 7 -
2-5. 実証エリア	- 8 -
2-6. スケジュール	- 9 -
3. 実証システム	- 10 -
3-1. アーキテクチャ	- 10 -
3-1-1. システムアーキテクチャ	- 10 -
3-1-2. データアーキテクチャ	- 14 -
3-1-3. ハードウェアアーキテクチャ	- 16 -
3-2. システム機能	- 22 -
3-2-1. システム機能一覧	- 22 -
3-2-2. 利用したソフトウェア・ライブラリ	- 24 -
3-2-3. 開発機能の詳細要件	- 26 -
3-3. アルゴリズム	- 56 -
3-3-1. 利用したアルゴリズム	- 56 -
3-3-2. 開発したアルゴリズム	- 60 -
3-4. データインタフェース	- 64 -
3-4-1. ファイル入力インタフェース	- 64 -
3-4-2. ファイル出力インタフェース	- 64 -
3-4-3. 内部連携インタフェース	- 67 -
3-4-4. 外部連携インタフェース	- 72 -
3-5. 実証に用いたデータ	- 75 -
3-5-1. 活用したデータ一覧	- 75 -
3-5-2. 生成・変換したデータ	- 78 -
3-6. ユーザーインタフェース	- 80 -
3-6-1. 画面一覧	- 80 -
3-6-2. 画面遷移図	- 82 -
3-6-3. 各画面仕様詳細	- 83 -

3-7. 実証システムの利用手順	- 97 -
3-7-1. 実証システムの利用フロー	- 97 -
3-7-2. 各画面操作方法	- 98 -
4. 実証技術の検証	- 104 -
4-1. 自己位置推定機能の検証	- 104 -
4-1-1. 検証目的	- 104 -
4-1-2. KPI	- 104 -
4-1-3. 検証方法と検証シナリオ	- 104 -
4-1-4. 検証結果	- 107 -
4-2. 測位情報の統合アルゴリズムの検証	- 115 -
4-2-1. 検証目的	- 115 -
4-2-2. KPI	- 115 -
4-2-3. 検証方法と検証シナリオ	- 115 -
4-2-4. 検証結果	- 119 -
4-3. ビジュアルマイナーシャルオドメトリによる高さ方向への移動の検証	- 121 -
4-3-1. 検証目的	- 121 -
4-3-2. KPI	- 121 -
4-3-3. 検証方法と検証シナリオ	- 121 -
4-3-4. 検証結果	- 123 -
5. BtoBビジネスでの有用性検証	- 126 -
5-1. 検証目的	- 126 -
5-2. 検証方法	- 128 -
5-3. 被験者	- 129 -
5-4. ヒアリング・アンケートの詳細	- 130 -
5-4-1. アジェンダ・タイムテーブル	- 130 -
5-4-2. アジェンダの詳細	- 130 -
5-4-3. 検証項目と評価方法	- 131 -
5-4-4. 実証実験の様子	- 132 -
5-5. 検証結果	- 139 -
6. 成果と課題	- 145 -
6-1. 本実証で得られた成果	- 145 -
6-1-1. 3D都市モデルの技術面での優位性	- 145 -
6-1-2. 3D都市モデルのビジネス面での優位性	- 145 -
6-2. 実証実験で得られた課題と対応策	- 147 -
6-3. 今後の展望	- 148 -
7. 用語集	- 149 -

1. ユースケースの概要

1-1. 現状と課題

都市部における建設工事では資材運搬等による交通渋滞が発生するため、自律運航可能なドローン等の活用が期待されている。一方で、GPS を利用したドローンの自律運航は GPS 受信状況によっては安全性に課題が生じることから、環境地図の利用による汎用的な位置測位技術の確立など、ロバスト性（頑強性）のある自律運航システムの開発が求められている。

上記課題の解決に向け、2022 年度の「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」では、3D 都市モデルと BIM を統合したマップを点群化し、これを用いた LiDAR による自己位置測位と GPS 測位の結果を統合することで、高精度、低コスト、スケーラブルな屋内外でのドローンの自律運航システムを開発した。このシステムを用いた実証により、統合マップを用いた屋外から屋内へのドローンの自律運航が可能なことが確認され、一定の成果を得た。

一方で、LiDAR の性質上、均一な特徴点を持つ建築物（例えば、同じ形状の窓が並ぶ高層ビルなど）を垂直方向に長距離移動する場合などは、特徴点をとらえられず飛行精度が低下するという飛行面の課題や、自律運行システムの操作に専門知識が必要であることからユーザビリティ面での課題を残した。

1-2. 課題解決のアプローチ

今回の実証実験では、2022年度の「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」として開発した、3D都市モデルとBIMの統合マップを利用したLiDAR自己位置測位とGPS測位の統合自己位置測位システム（Edge LiDAR Autonomous System（ELAS））の課題を克服するため、ビジュアルイナーシャルオドメトリ（カメラと慣性計測装置を用いた経路推定手法）技術を組み合わせることで、飛行の安定性を更に高めたロバストな自律運航システムの開発を目指す。

これにより、統合マップを用いたLiDAR自己位置測位、GPS測位に3つ目の新たな測位情報を加え、これらに最適な統合処理を行うことで、それぞれの不得意な領域を補い合う統合的な自己位置測位システムを実現する。

また、専門知識を持たない一般的なユーザーが利用できるUIを開発することで、より多くのユーザーが自動運航システムの操作が可能となる。そのため、ユーザビリティの向上を目的として、株式会社センシンロボティクス製のGCSアプリ「SENSYN CORE Pilot」と接続することで、ドローン自律飛行経路の計画やドローン飛行中の飛行指示、及びドローン情報のリアルタイム表示等をより簡易に行える仕組みを構築する。

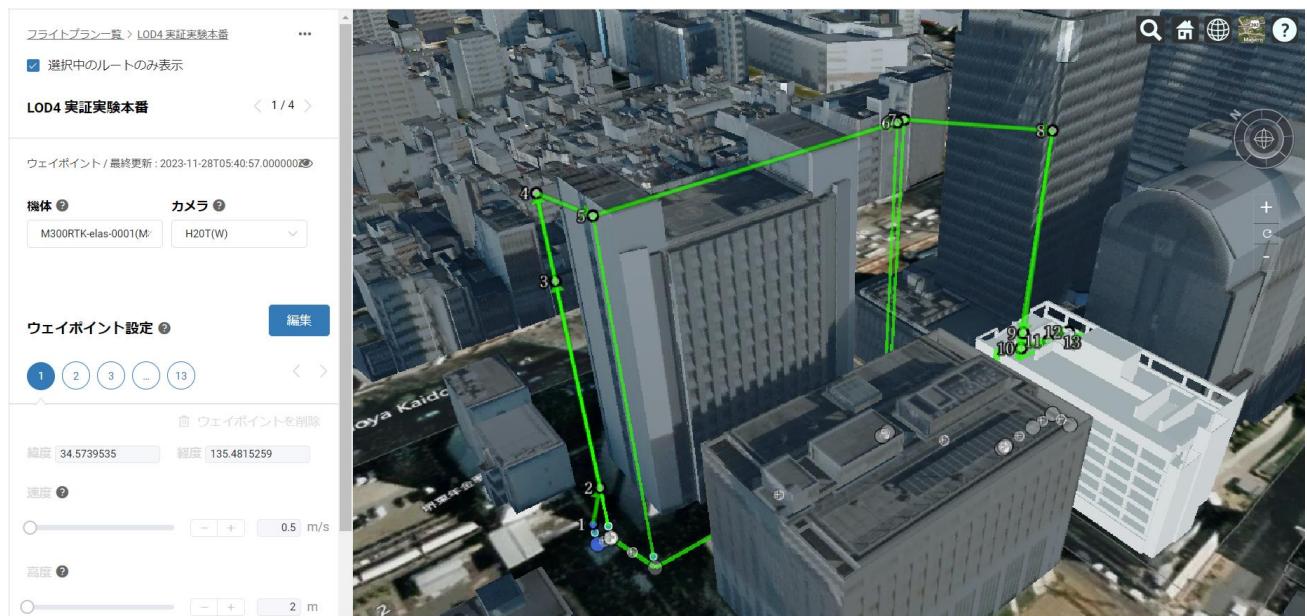


図 1-1 SENSYN CORE Pilot 上でのドローン飛行ルート設計画面

1-3. 創出価値

都市部における建設工事では、建設工事の資材運搬等によって発生する交通渋滞の問題や、様々な大きさ・重量の資材を屋外・屋内へ移動させる「ラストワンマイル」を効率化することの難しさが課題として挙げられている。これらの課題は、工事自体の遅延を引き起こすだけではなく、対象区画で通行止めの必要を生じさせるなど、近隣住民の生活や都市機能にも影響を及ぼす。

このため、自律運航型ドローンを資材運搬等に活用する方法が研究されているが、GPSによる位置測位の不安定性や、SLAMによる地図作成技術のコストの高さや精度の不足など、都市部における正確で安全な飛行の実現には課題が多い。

今回の実証実験で開発したシステムを用いて、都市部の屋外空間、屋内空間、建築物に沿った垂直空間など、特徴の異なる空間をシームレスに移動可能な自律運航型ドローンシステムを開発し、高精度で安定的なドローン自律運航システムを実現することで、新たなモビリティサービスとしてのドローンの社会実装を目指す。

1-4. 想定事業機会

表 1-1 想定事業機会

項目	内容
利用者	<ul style="list-style-type: none"> ● 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
サービス仮説	<ul style="list-style-type: none"> ● ドローン自律運航による業務支援サービス ● 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 <ul style="list-style-type: none"> ➢ 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
提供価値	<ul style="list-style-type: none"> ● 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"> ➢ ドローンでは GPS 測位のみ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 ➢ 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を活用した運行システムを開発し提供する

2. 実証実験の概要

2-1. 実証仮説

- 今回の実証実験では、昨年度のシステムを基礎として、「ビジュアルマイナーシャルオドメトリ」（連続的に取得されたカメラ画像の変化を捉えることでドローンの移動量を算出する技術）を用いた自己位置測位システムを追加する改修を加える。これにより、統合マップを用いたLiDARによる自己位置測位、GPS測位に3つ目の自己位置測位情報を加え、これらを最適に統合する処理を行うことで、それぞれの不得意な領域を補い合う統合的な自己位置測位システムを実現する。
- 3D都市モデルを活用したドローンの経路計画について、ドローンの自律運航システムを技術者ではない者でもオペレーション可能なGUIを含むWebアプリケーション（株式会社センシンロボティクス製のGCSアプリ「SENSYN CORE Pilot」）上に追加開発することにより、専門的な知識を持たないユーザーでも手軽に本システムを利用できる。

2-2. 実証フロー

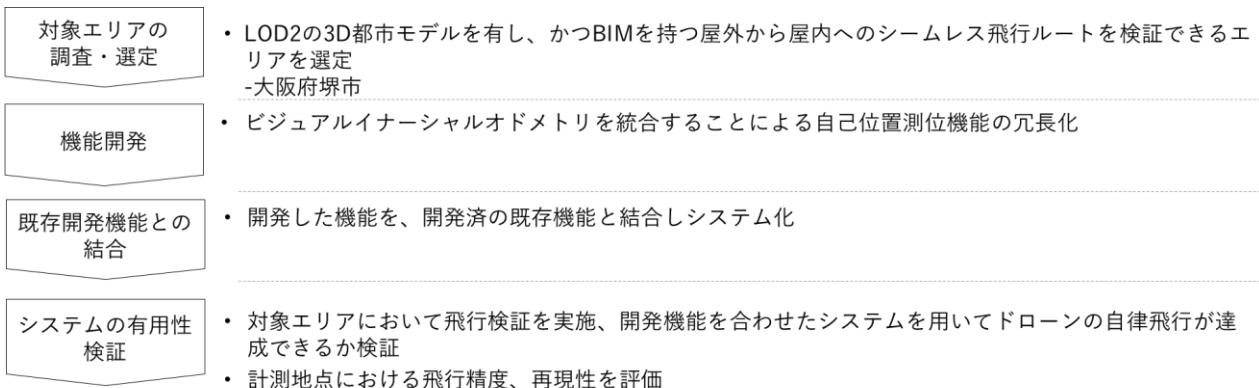


図 2-1 実証フロー

2-3. 検証ポイント

- ビジュアルイナーシャルオドメトリの使用による自己位置推定アルゴリズムの機能向上
 - 3D LiDAR ローカリゼーションで地上面を測定できなくなる 35m 以上の高さでビジュアルイナーシャルオドメトリを使用した場合と使用しない場合の自己位置推定の精度を比較することで、今回開発したアルゴリズムの優位性を確認する。
- 測位情報の統合アルゴリズムを活用した高精度なドローン自律飛行システムの確立
 - 3D 都市モデルを事前地図として活用することで、事前の撮影業務を行わずに精度の高い事前地図を作成し、測位情報の統合アルゴリズムによる高い自己位置推定精度を実現したドローン自律飛行システムが確立できることを確認する。
- ビジュアルイナーシャルオドメトリ単体での高さ方向への飛行ロバスト性
 - 昨年度の実証結果を踏まえ、LiDAR ローカリゼーションでは自己位置推定が難しかった均一な特徴点を持つ建築物（例えば、同じ形状の窓が並ぶ高層ビルなど）において、ビジュアルイナーシャルオドメトリを使用した垂直方向への移動に対する自己位置推定の有用性を確認する。

上記 3 点の検証ポイントについては、【4 章：実証技術の検証】にて検証結果を記載

- 今回の実証実験で開発したドローン自律運航システムの有用性検証
 - 「SENSYN CORE Pilot」を使用して構築した今回のシステムが、専門的な知識を持たないオペレーターでも手軽に使用可能なユーザビリティの高いシステムになっているかを確認する。
 - 昨年度開発した「Rviz」と比較し、ドローンのルート設計から飛行状況の確認及び緊急停止指示などのオペレーションまで利便性や操作性高く使用できるかを確認する。

上記 1 点の検証ポイントについては、【5 章：BtoB ビジネスでの有用性検証】にて検証結果を記載

2-4. 実施体制

表 2-1 実施体制

役割	主体	詳細
全体管理	国土交通省 都市局	プロジェクト全体ディレクション
	アクセンチュア	プロジェクト全体マネジメント
実施事業者	竹中工務店	ユースケース実証におけるプロジェクト全体管理・運営 ユースケース実証において利用するBIMデータの作成
	センシンロボティクス	ユースケース実証におけるドローン側システム構築の企画・開発・検証・運営
実施協力	大阪府堺市	実証実験場所に関する情報提供 飛行敷地提供
	大阪国税局堺税務署	飛行敷地提供

2-5. 実証エリア

表 2-2 実証エリア

項目	内容
実証地	大阪府堺市
面積	8,550 m ²
マップ (対象エリア は赤枠内)	

図 2-2 実証エリア

2-6. スケジュール

表 2-3 スケジュール

実施事項	2023年											2024年		
	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都市モデルの活用方策・要件定義・実証手法の検討	←			→										
2. BIM、3D都市モデルの整備		←			→									
3. ビジュアルライナーシャルオドメトリシステム開発			←		→									
4. ユーザーインターフェース開発			←		→									
5. ドローンによる飛行の実施					←	→								
6. ヒアリング・アンケート実施										↔				
7. 成果とりまとめ										←	→			

3. 実証システム

3-1. アーキテクチャ

3-1-1. システムアーキテクチャ

今回の実証実験では、ドローン自律運航システムのロバスト性の更なる向上を目的として LiDAR と GNSS（衛星測位システム）による自己位置推定に加え、新たにビジュアルイナーシャルオドメトリによる自己位置測位を活用した自己位置推定システムを構築した。この自己位置推定システムおよびドローン制御システムを兼ね備えたソフトウェアパッケージを Edge LiDAR Autonomous System (ELAS) と呼び、ELAS では、建築物モデル (LOD2) と BIM モデルを統合した屋内外の 3D マップを環境地図として用い、自己位置推定を行う。また、ルート設計とドローン監視のための GCS (※) として、センシンロボティクス社製の GCS アプリ 「SENSYN CORE Pilot」を活用した。SENSYN CORE Pilot を用いることで、ユーザビリティを向上させ、運航経路の計画やドローンへの飛行指示をより簡易に入力・実行できる仕組みとした。

これら二つのサブシステムを統合し、ドローン自律運航システムとして構築している。

※Ground Control System：無人航空機を人間が制御するためのシステム

「自己位置推定システム」は、ドローンの飛行状況に応じて複数の自己位置測位情報を統合又は利用する自己位置測位情報を切り替える機能を持つ。LiDAR と GNSS を用いた機能 (FUSION) は昨年度の開発 (<https://www.mlit.go.jp/plateau/use-case/uc22-024/>) により実装済みのため、今年度の開発では、これにビジュアルイナーシャルオドメトリを用いた自己位置測位情報を付加する開発を行った。

ビジュアルイナーシャルオドメトリとは、カメラで撮像 (データを活用/記録するために画像を取得すること) した画像と IMU (※) の変位からドローンや移動ロボットの姿勢および移動量を算出する技術である。この技術を用いることで、GNSS が利用できない環境下で、LiDAR が不得手である均一な特徴点を持つコース (例えば、同じ構造が連続する高層ビルの壁面を垂直方法に移動するケースなど) において自己位置推定の精度向上させることができる。

今回の実証実験では、ドローン本体の内蔵型前方カメラを用いて 1 秒間に 30 枚の画像を撮像し、限られたドローン内のリソースでもリアルタイムな処理が可能な FAST アルゴリズム (画像中から特徴点を検出するアルゴリズムであり、特徴点を検出する際に、画像を細かいエリアに分割し、エリアの処理の必要性を 4 つのピクセルのデータから先行して判断することで、画像全体の処理を高速に行う) を利用することで撮像した画像の特徴点の変化を移動量に変換している。

FUSION によってビジュアルイナーシャルオドメトリから取得した移動量、GPS・LiDAR 自己位置推定からの位置情報、IMU からの加速度・角速度など各センサデータによる運動モデル全体の予測を繰り返すことにより測位情報が統合され、各センサ単体よりもロバストで正確な自己位置推定を行うことが可能になる。

これらの処理はドローンに搭載されたエッジコンピューター上で行われ、その結果は ROS メッセージを用いて送信される。このため、機体側に USB 接続の LTE ドングルを通信デバイスとして搭載した。

システム全体の構築にあたってはロボット・アプリケーションの作成支援のためのライブラリ・ツール群である ROS (Robot Operating System) を利用している。

LiDARとGNSSにビジュアルマイナーシャルオドメトリを利用した測位情報を加えることで、それぞれ不得意な領域を補い合うロバストな自己位置測位システムを構築した。

※Inertial Measurement Unit：加速度、角速度など3次元の慣性運動を計測する装置

GCSアプリ「SENSYN CORE Pilot」は、ドローンの飛行ルートの設定と、飛行中のドローン位置の表示や飛行指示を行うためのオペレーティングツールであり、直感的なドローン運航を目的として利用している。本システムでは、「自己位置推定システム」から取得した自己位置測位情報を「SENSYN CORE Pilot」上でリアルタイムに表示しオペレーターの飛行操作を支援するため、地形データ切替え機能、PLATEAU-Terrain 対応機能、ROS メッセージ双方向変換機能、非常停止機能を開発。「SENSYN CORE Pilot」とドローンの双方向の通信システムを確立している。

ドローンの自律飛行（自己位置推定）に用いる3Dマップとしては、大阪府堺市堺区の3D都市モデルの建築物モデル（LOD2）に堺市役所立体駐車場のBIMを「FMEを用いたIFCからCityGMLへの変換フォーマット（<https://github.com/Project-PLATEAU/IFCtoCityGML>）」で変換した建築物モデル（LOD4）をマージし一つのCityGMLファイルにし、Trimeshライブラリを用いCityGMLの表面をサンプリングすることで点群を生成した。また、ドローンの経路計画およびドローン飛行中の機体位置の表示に用いる3Dマップのうち、大阪府堺市堺区の3D都市モデルの建築物モデル（LOD2）は「PLATEAU配信サービス」を利用し、堺市役所立体駐車場の建築物モデル（LOD4）はCesium ionを利用することで、Cesium ion地図上に建築物モデルをそれぞれ表示しドローン経路計画等に利用した。

本システムのシステム・アーキテクチャは下図の通りである。

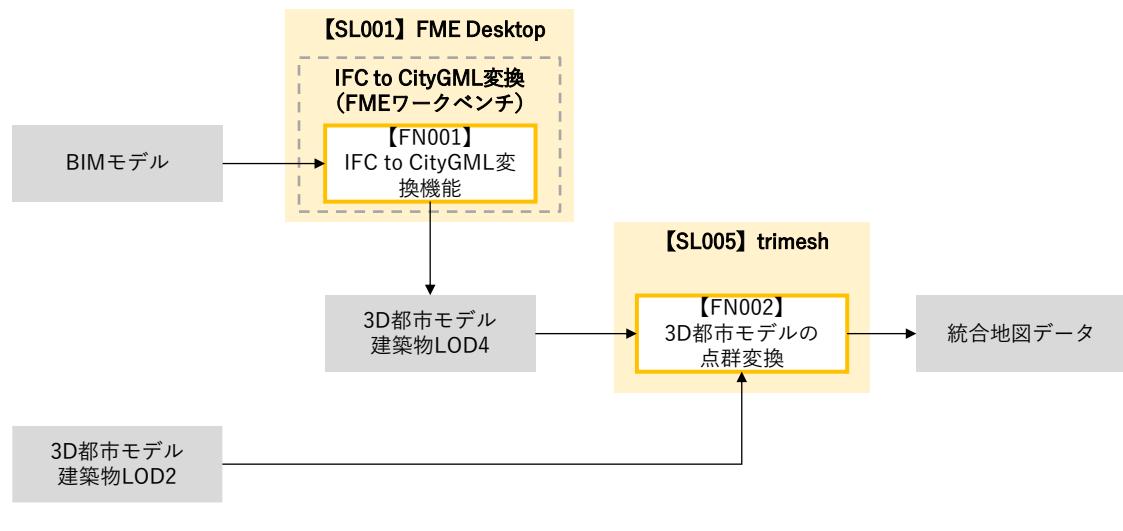
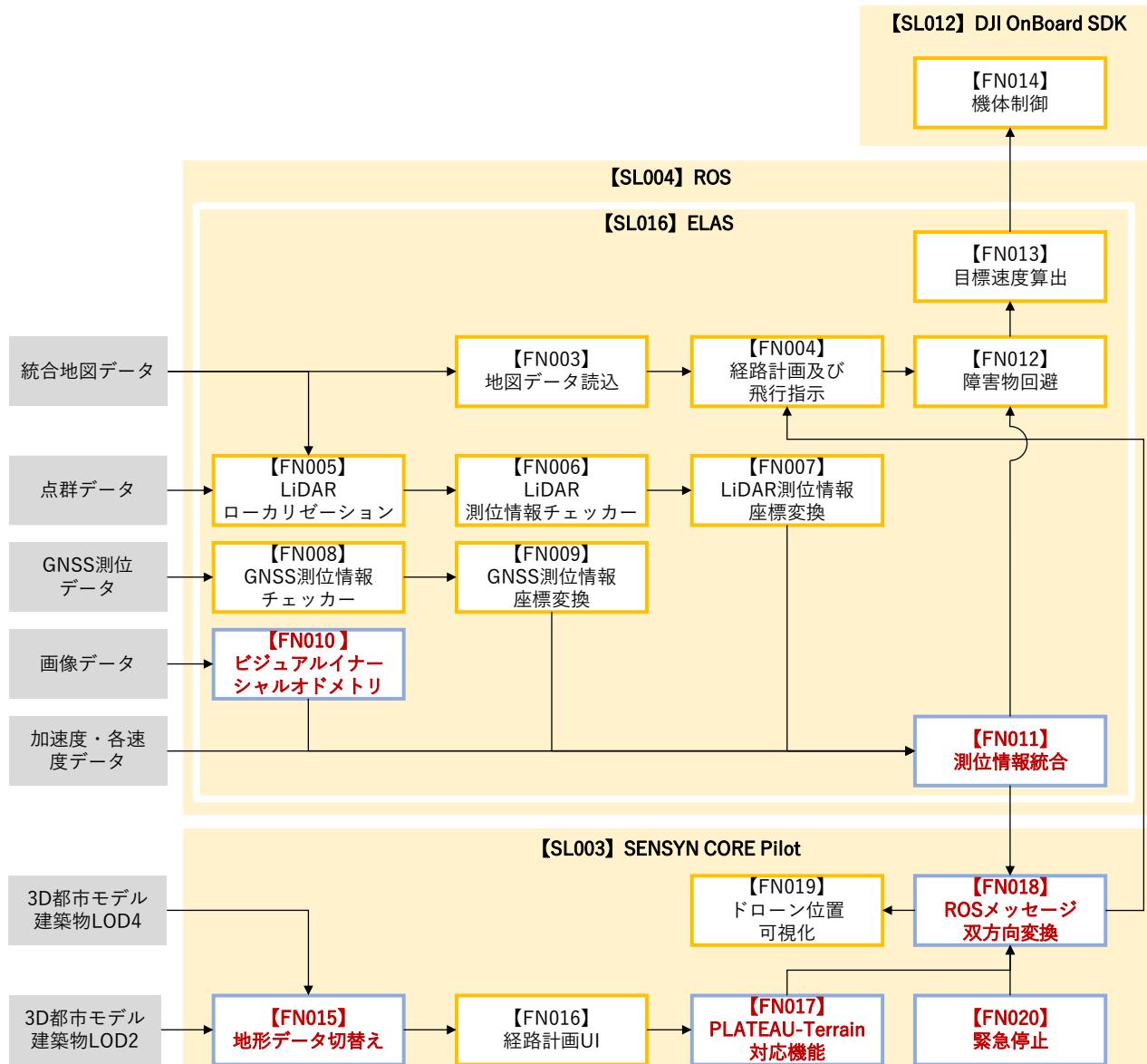


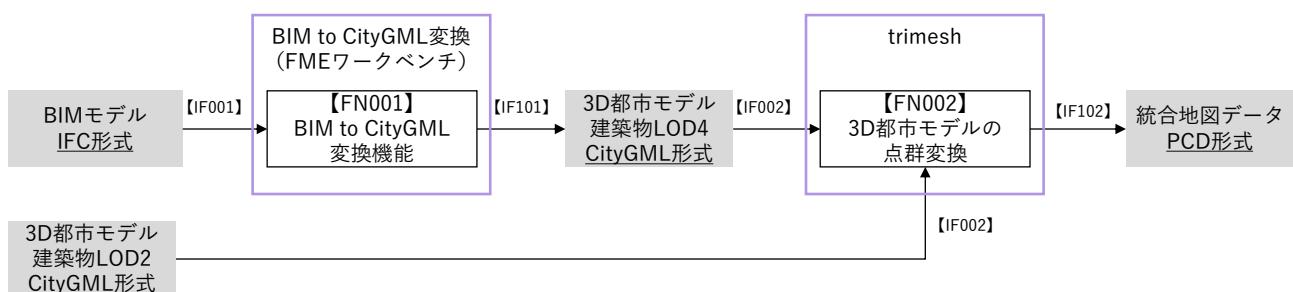
図 3-1 システムアーキテクチャ① (統合地図データ生成)



凡例	既存の ソフトウェア	開発した ソフトウェア	既存機能	開発した機能	データ	ファイル ストレージ	データベース
----	---------------	----------------	------	--------	-----	---------------	--------

図 3-2 システムアーキテクチャ② (飛行計画～ドローン自律運航)

3-1-2. データアーキテクチャ



凡例

UI入力

データ

データベース

ファイル
ストレージ

ソフトウェア

データ
処理

図 3-3 データアーキテクチャ① (統合地図データ生成)

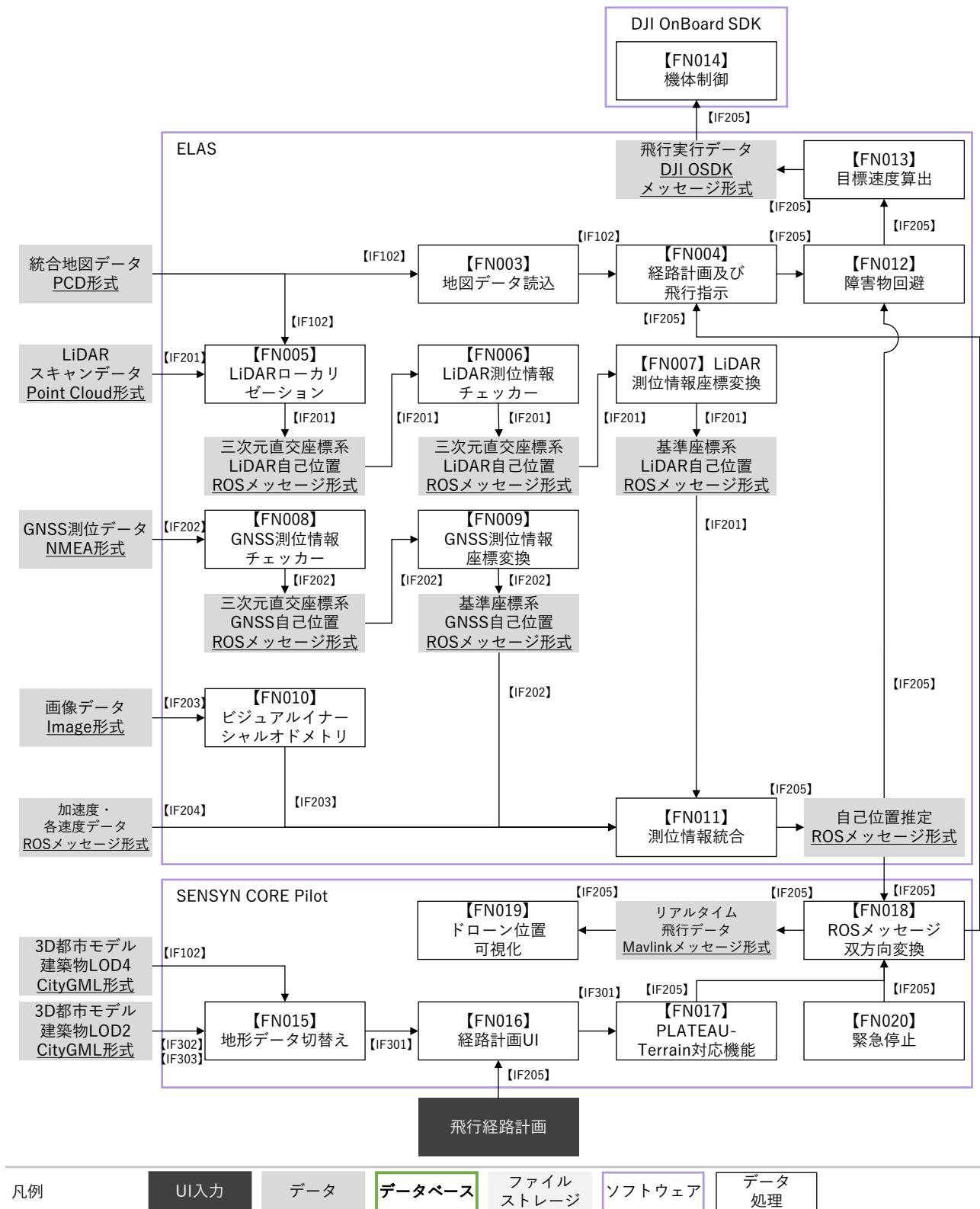


図 3-4 データアーキテクチャ② (飛行計画～ドローン自律運航)

3-1-3. ハードウェアアーキテクチャ

3-1-3-a. 利用したハードウェア一覧



図 3-5 ハードウェアアーキテクチャ① (点群マップ生成)

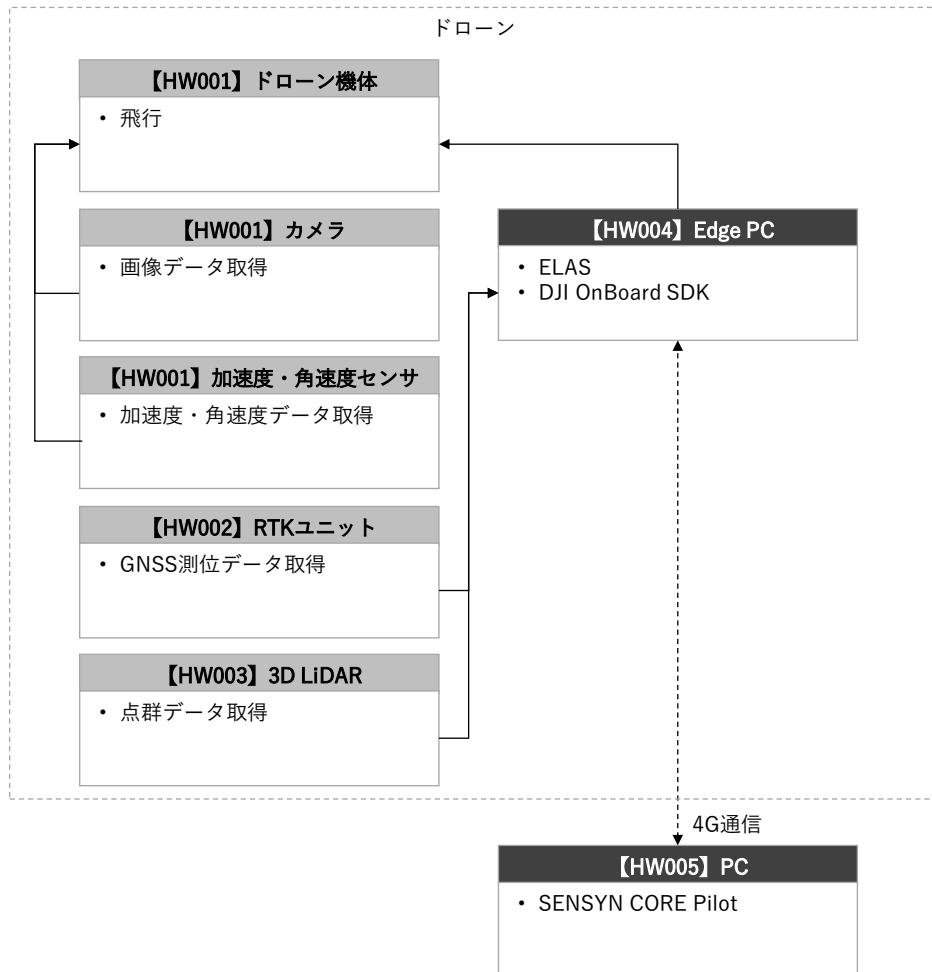


図 3-6 ハードウェアアーキテクチャ② (飛行計画～ドローン自律運航)

表 3-1 利用したハードウェア一覧

ID	種別	品番	用途
HW001	ドローン	DJI Matrice 300	3D LiDAR 及び Edge PC を搭載したドローン飛行のため に利用。
HW002	RTK ユニット	1 に標準搭載され たものを使用	GNSS 測位による自己位置推定のために利用。
HW003	3D LiDAR	Ouster OS0-64	LiDAR センサを用いた自己位置推定（ローカリゼーション） のための点群取得に利用。
HW004	Edge PC	PICO-WHU4W2-A11-0004	各センサ情報を処理し、LiDAR ローカリゼーションやビ ジュアルイナーシャルオドメトリ、GNSS 測位を用いた FUSION のために利用。
HW005	PC	Windows PC	点群マップの生成、および Web ブラウザ上で株式会社セ ンシンロボティクス社製 GCS アプリ「SENSYN CORE Pilot」を使用するために利用。

3-1-3-b. 利用したハードウェア詳細

1) 【HW001】 ドローン : DJI Matrice 300 RTK

- 選定理由

- Edge PC と LiDAR を搭載しやすい設計
- 外部 SDK を具備
- ペイロードが十分ある

- 仕様・スペック

- サイズ: 810 × 670 × 430mm
- 最大積載量: 2.7 kg
- 最大風圧抵抗: 15 m/秒
- 最大飛行時間: 55 分

- イメージ



図 3-7 DJI Matrice 300¹

2) 【HW002】 RTK ユニット : 1 に標準搭載されたもの

- 選定理由

- 機体にインテグレーション済

- 仕様・スペック

- GPS+GLONASS+BeiDou+Galileo
- 1cm+1ppm (水平方向)
- 1.5cm+1ppm (垂直方向)

¹ 公式 HP より抜粋 : <https://enterprise.dji.com/jp/matrice-300>

3) 【HW003】3D LiDAR : Ouster OS0-64

● 選定理由

- 超広視野角
- 軽量 (450g)

● 仕様・スペック

- 反射率10%のターゲットで30m以上
- 最大測定範囲50m
- 垂直視野90度
- ビーム本数64本の解像度
- 520万ポイント/秒

● イメージ

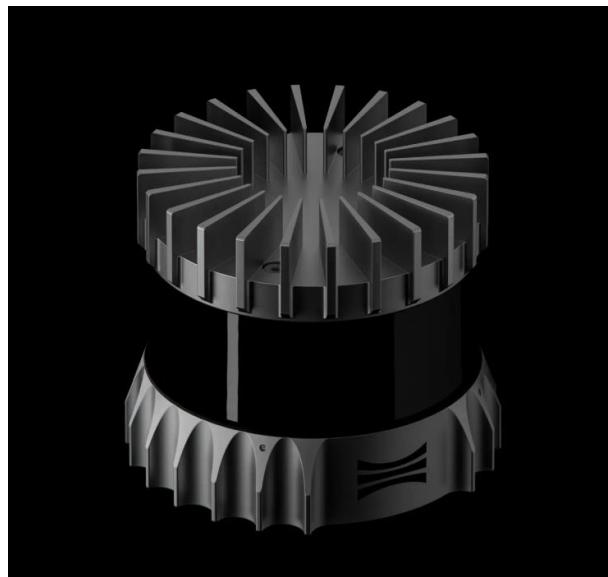


図 3-8 Ouster OS0-64²

² 公式HPより抜粋：<https://ouster.com/jp/products/scanning-lidar/os0-sensor/>

4) 【HW004】Edge PC : PICO-WHU4W2-A11-0004

● 選定理由

- 筐体が小型であり、各センサ情報を処理するのに十分なCPU処理性能を持つこと

● 仕様・スペック

- サイズ：縦100mm×横72mm

- CPU : Intel® i7-8665UE

- Storage : 256GB

● イメージ



図 3-9 PICO-WHU4W2-A11-0004（オモテ面）³



図 3-10 PICO-WHU4W2-A11-0004（ウラ面）⁴

³ 公式HPより抜粋：<https://www.aaeon.com/en/p/pico-itx-boards-pico-whu4>

⁴ 公式HPより抜粋：<https://www.aaeon.com/en/p/pico-itx-boards-pico-whu4>

5) 【HW005】PC : Lenovo 21B4S0C700

- 選定理由
 - SENSYN CORE Pilot アプリを利用できるのに十分なスペックである
- 仕様・スペック
 - サイズ: 約 305×218×17.3mm
 - CPU : Intel® i7-1255U 1.70 GHz
 - OS : Windows 10 Business 64bit
 - メモリ : 16GB
 - ストレージ : 512GB
- イメージ



図 3-11 Lenovo 21B4S0C700⁵

⁵ 公式HPより抜粋：[https://www.lenovo.com/jp/ja/p/laptops/thinkpad/thinkpad-l-series/thinkpad-l13-gen-3-\(13-inch-intel\)/len101t0031?orgRef=https%253A%252F%252Fwww.google.com%252F](https://www.lenovo.com/jp/ja/p/laptops/thinkpad/thinkpad-l-series/thinkpad-l13-gen-3-(13-inch-intel)/len101t0031?orgRef=https%253A%252F%252Fwww.google.com%252F)

3-2. システム機能

3-2-1. システム機能一覧

1) 機能一覧

表 3-2 機能一覧

※赤文字：既存改修・新規開発

大分類	小分類	ID	機能名	機能説明
統合地図データの作成	CityGML to Robot Map	FN001	BIM to CityGML 変換機能	<ul style="list-style-type: none"> 「3D都市モデル整備のためのBIM活用マニュアル」に従って変換 BIM(IFC)から3D都市モデル(LOD4)へ変換する
		FN002	3D都市モデルの点群変換	<ul style="list-style-type: none"> 3D都市モデルを点群(Point Cloud)を生成する
自己位置測位	ELAS	FN003	地図データ読み込み	<ul style="list-style-type: none"> 自己位置推定用の地図として利用するために点群マップを読み込む
		FN004	経路計画及び飛行指示 ・ウェイポイント(基準座標系) ・緊急停止指示	<ul style="list-style-type: none"> ユーザーが入力した経路計画(ウェイポイント/アクション)を受け取る
		FN005	LiDAR ローカリゼーション	<ul style="list-style-type: none"> ドローンに搭載した3D LiDARで取得したデータと3D都市モデルから変換した点群マップを重ね、自己位置を推定する
		FN006	LiDAR 測位情報チェック	<ul style="list-style-type: none"> 3D LiDARによる測位情報の信頼性評価を行う機能
		FN007	LiDAR 測位情報座標変換	<ul style="list-style-type: none"> 3D LiDARによる測位情報を基準座標系へ変換する機能
		FN008	GNSS 測位情報チェック	<ul style="list-style-type: none"> GNSSによる測位情報の信頼性評価を行う機能
		FN009	GNSS 測位情報座標変換	<ul style="list-style-type: none"> GNSSによる測位情報を基準座標系へ変換する機能
		FN010	ビジュアルライナーシャルオドメトリ	<ul style="list-style-type: none"> ドローン内臓カメラで撮影した、連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU情報を加味して移動量を推定する機能

		FN011	測位情報統合	● 複数の測位情報を統合する機能
		FN012	障害物回避	● 障害物回避機能
		FN013	目標速度算出	● 自己位置と目標位置からドローンの移動速度を決定/指示する機能
		FN014	機体制御【DJI Onboard SDK】	● 飛行指示に従い機体を制御する機能
ドローン制御ソフトウェア SENSYN CORE Pilot	経路計画	FN015	地形データ切替え	● 3D都市モデルとBIMの統合マップをSENSYN CORE Pilot上で表示するために地形データの切替えを行う機能
		FN016	経路計画UI	● 経路計画を行う画面を表示
		FN017	PLATEAU-Terrain 対応機能	● SENSYN CORE Pilotを用いてドローンの経路計画を行う際に、SENSYN CORE Pilotで用いられている標高データと異なる標高データ(PLATEAU-Terrain)を用いても意図どおりの飛行経路を作成する機能
	HW接続	FN018	ROSメッセージ双方向変換	● SENSYN CORE PilotとROSにより動作するロボットの間で、情報を双方向にやりとりするのに適切な形式に変換する機能
		FN019	ドローン位置可視化	● ドローンの現在位置をリアルタイムで表示
	HW制御	FN020	緊急停止	● ドローンを緊急停止させる指令をELASに送る機能

3-2-2. 利用したソフトウェア・ライブラリ

利用したソフトウェア、ライブラリを表に示す。

表 3-3 利用したソフトウェア・ライブラリ

赤文字：新規利用

ID	項目	内容
SL001	FME Desktop	<ul style="list-style-type: none"> Safe Software 社が提供するソフトウェア BIM (IFC データ) を 3D 都市モデルに変換する FME を用いた IFC から CityGML への変換フォーマット
SL002	Ubuntu	<ul style="list-style-type: none"> Canonical 社が提供する Linux Distribution
SL003	SENSYN CORE Pilot	<ul style="list-style-type: none"> SENSYN ROBOTICS 社の自社製 GCS アプリ。無人航空機の経路計画、制御、機体位置の可視化などを行う
SL004	ROS	<ul style="list-style-type: none"> Open Robotics が提供するロボットオペレーションシステム
SL005	trimesh	<ul style="list-style-type: none"> オープンソースとして提供されている Python ライブラリ 3D オブジェクトを多角形の集合で表現し、三角メッシュの読み込み操作・分析が可能
SL006	Point cloud library	<ul style="list-style-type: none"> オープンソースとして提供されている 3 次元点群処理が可能なライブラリ
SL007	proj	<ul style="list-style-type: none"> オープンソースとして提供されてる複数の測地系を扱うことを可能とするライブラリ
SL008	OpenCV	<ul style="list-style-type: none"> オープンソースとして提供されている画像処理アルゴリズムなどの機能を提供するライブラリ
SL009	Octomap	<ul style="list-style-type: none"> オープンソースとして提供されているセンサ情報から 3D 環境認識ができるライブラリ
SL010	ompl	<ul style="list-style-type: none"> Kavraki Lab (Rice University) が提供する Open Motion Planning Library 経路計画アルゴリズムのライブラリ
SL011	Boost	<ul style="list-style-type: none"> オープンソースとして提供されている C++ の準標準ライブラリ
SL012	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、オンラインボードコンピューターに対応するライトコントローラーに接続可能
SL013	Cesium Geocode API	<ul style="list-style-type: none"> Cesium 社が提供するジオコード API ジオコードとは、住所情報を数値形式のコードに変換したもの
SL014	egm96-universal	<ul style="list-style-type: none"> アメリカ国家地球空間情報局 (National Geospatial-Intelligence

		Agency、NGA) が提供する EGM96 のジオイド高を取得するライ ブライ
SL015	MAVLink toolchain	<ul style="list-style-type: none"> ● MAVLink はオープンソースとして提供されており、ドローンと GCS の間で機体の状態やルート情報などのデータを送受信するメッセージフォーマット ● MAVLink toolchain によりカスタムメッセージフォーマットを XML で定義することで C++ や Python のヘッダファイルを自動的に生成する
SL016	ELAS	<ul style="list-style-type: none"> ● SENSYN ROBOTICS 社が提供する Edge LiDAR Autonomous System の略称 ● 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 ● 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/指示を与えて、自律移動を行う技術 ● 「事前マップ」について、本実証実験では 3D 都市モデルを点群化したデータを事前マップとして活用することで、SLAM による事前マップ作成手法と比較して低コスト、高精度、スケーラブルなマップ作成を実現している ● 「自己位置測位」について、具体的には LiDAR と GNSS (衛星測位システム) による自己位置推定に加え、本実証実験で新たにビジュアルインーシャルオドメトリによる自己位置測位を活用できるよう改修を行った
SL017	cityjson	<ul style="list-style-type: none"> ● オープンソースとして提供されているデータフォーマット ● CityGML 形式の 3D 都市モデルを OBJ データに変換

3-2-3. 開発機能の詳細要件

開発機能の詳細要件を記す。なお、本業務において新規開発した要素（機能名）を赤字で示す。

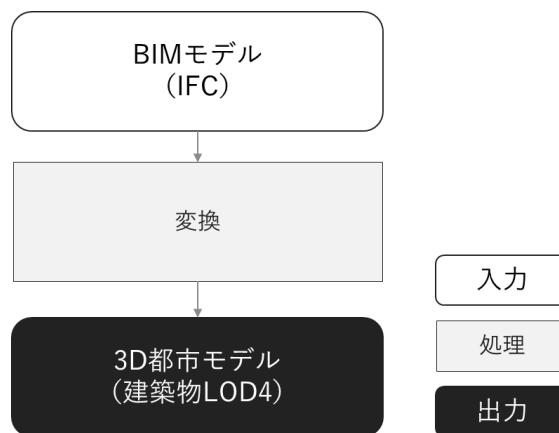
1) 機能一覧

1. 【FN001】BIM to CityGML 変換機能

- 機能概要

- 「3D都市モデル整備のためのBIM活用マニュアル」に従って変換
- BIMモデル(IFC)から3D都市モデル(LOD4)へ変換する

- フローチャート



- データ仕様

- 入力

- ❖ BIMモデル(IFC)
 - 内容
 - 堺市役所立体駐車場のIFCデータ
 - 形式
 - IFC形式
 - データ詳細
 - ファイル入力インターフェース【IF001】を参照

- 出力

- ❖ 3D都市モデル(建築物LOD4)
 - 内容
 - 3D都市モデルのCityGMLファイル
 - 形式
 - CityGML形式
 - データ詳細
 - ファイル出力インターフェース【IF101】を参照

- 機能詳細
 - IFC形式のデータをCityGML形式のデータに変換する
 - ✧ 処理内容
 - IFC形式のデータをCityGML形式のデータに変換
 - ✧ 利用するライブラリ
 - FME Desktop(ソフトウェア・ライブラリ【SL001】を参照)
 - ✧ 利用するアルゴリズム
 - なし

2. 【FN002】3D都市モデルの点群変換

- 機能概要
 - Trimeshのライブラリを用いて開発した機能
 - 3D都市モデルから点群(Point Cloud)を生成する
- フローチャート

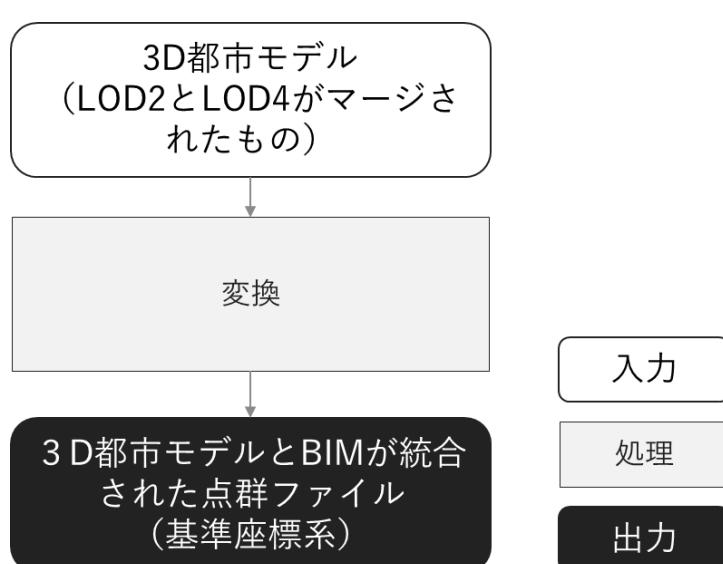


図 3-13 3D都市モデルの点群変換機能のフローチャート

- データ仕様
 - 入力
 - ✧ 3D都市モデル (LOD2とLOD4がマージされたもの)
 - 内容
 - 堺市役所・堺地方合同庁舎周辺の建築物 LOD2データと堺市役所立体駐車場のBIMモデルをLOD4データに変換したものマージした3D都市モデル (LOD2、LOD4)
 - 形式
 - CityGML形式
 - データ詳細
 - ファイル入力インターフェース【IF002】を参照

- 出力
 - ✧ 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）
 - 内容
 - 3D都市モデルの建築物LOD2、BIMモデル由来の建築物LOD4の統合点群データ
 - 形式
 - Point Cloud形式
 - データ詳細
 - ファイル出力インターフェース【IF102】を参照
- 機能詳細
 - CityGML形式の3D都市モデルマップから点群（Point Cloud）を生成
 - ✧ 処理内容
 - CityGML形式のマップをobjファイルに変換したのち点群に変換する
 - ✧ 利用するライブラリ
 - Trimesh（ソフトウェア・ライブラリ【SL005】を参照）
 - cityjson（ソフトウェア・ライブラリ【SL017】を参照）
 - ✧ 利用するアルゴリズム
 - なし

3. 【FN003】地図データ読み込み

- 機能概要
 - ドローン用の点群マップをELASに読み込む
- フローチャート

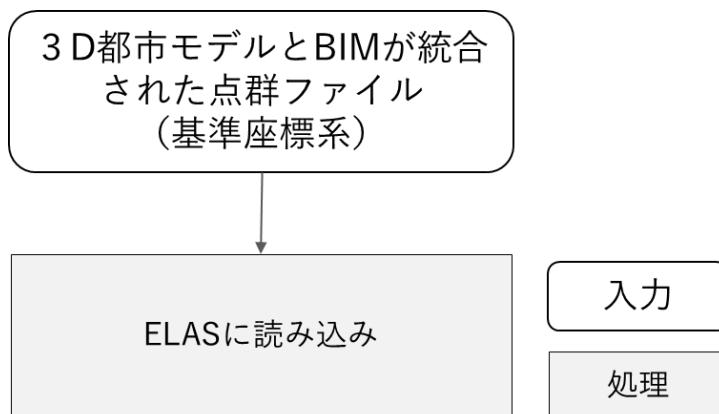


図 3-14 地図データ読み込み機能のフローチャート

- データ仕様
 - 入力
 - ✧ 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）
 - 内容
 - 3D都市モデルの建築物LOD2、BIMモデル由来の建築物LOD4の統合点群データ
 - 形式

- Pont Cloud 形式
- データ詳細
 - ファイル出力インターフェース【IF102】を参照
- 出力
 - ✧ なし
- 機能詳細
 - 点群データ読み込み
 - ✧ 処理内容
 - 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）をELASに読み込み、ドローンマップとして用いる
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

4. 【FN004】経路計画及び飛行指示・ウェイポイント（基準座標系）・緊急停止指示

- 機能概要
 - 3D都市モデルを地図上に表示させたSENSYN CORE Pilotの経路計画UI（【SC005】および【SC006】）上でユーザーが入力した経路計画（ウェイポイント）や、機体テレメトリー画面（【SC010】）上の緊急停止ボタン押下による緊急停止指示を受け取る機能
- フローチャート

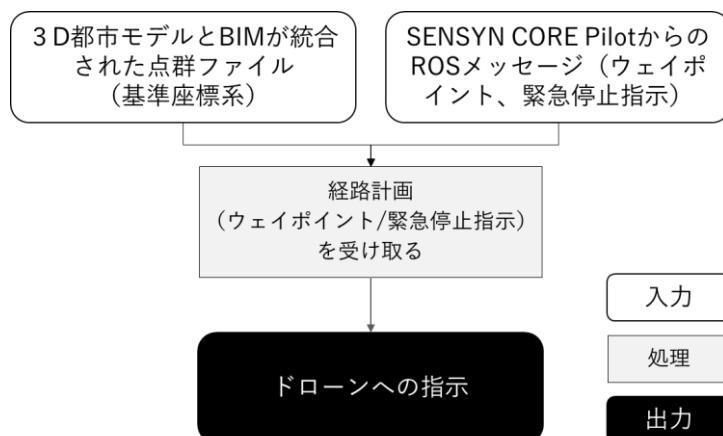


図 3-15 経路計画及び飛行指示機能のフローチャート

- データ仕様
 - 入力
 - ✧ 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）
 - 内容
 - 3D都市モデルの建築物LOD2、BIMモデル由来の建築物LOD4の統合点群データ
 - 形式

- Point Cloud 形式
- データ詳細
 - ファイル出力インターフェース【IF102】を参照
- ✧ SENSYN CORE Pilot からの ROS メッセージ（ウェイポイント、緊急停止指示）
 - 内容
 - 経路計画（ウェイポイント）情報
 - 緊急停止指示
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
- 出力
 - ✧ ドローンへの指示
 - 内容
 - ドローンへの飛行及びアクションの指示
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 外部連携インターフェース【IF304】を参照
- 機能詳細
 - 経路計画（ウェイポイント）や緊急停止指示の受け取り
 - ✧ 処理内容
 - 経路計画（ウェイポイント）や緊急停止指示を ROS メッセージで受け取る
 - 受け取った指示および点群マップを基にドローンに各種指示を送る
 - ✧ 利用するライブラリ
 - DJI Onboard SDK（ソフトウェア・ライブラリ【SL012】を参照）
 - ✧ 利用するアルゴリズム
 - なし
- 5. 【FN005】LiDAR ローカリゼーション
 - 機能概要
 - ドローンに搭載した 3D LiDAR で取得したデータと 3D 都市モデルから変換した点群マップを重ね、自己位置を推定する機能
 - フローチャート

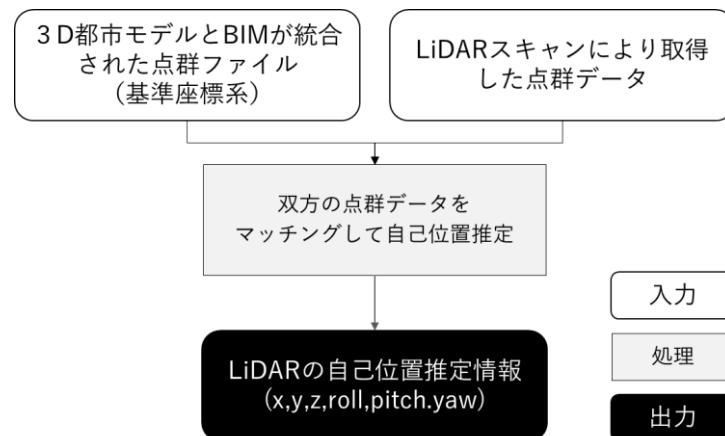


図 3-16 LiDAR ローカリゼーション機能のフローチャート

- データ仕様
 - 入力
 - ✧ 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）
 - 内容
 - 3D 都市モデルの建築物 LOD2、BIM モデル由来の建築物 LOD4 の統合点群データ
 - 形式
 - Point Cloud 形式
 - データ詳細
 - ファイル出力インターフェース【IF102】を参照
 - ✧ LiDAR スキャンにより取得した点群データ
 - 内容
 - LiDAR スキャンにより取得した、現実世界の点群データ
 - 形式
 - Point Cloud 形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - 出力
 - ✧ LiDAR の自己位置推定情報(x,y,z,roll,pitch,yaw)
 - 内容
 - LiDAR 機器の自己位置を推定した情報(三次元直交座標系)
 - 形式
 - ROS メッセージ形式 ※pose 形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - 機能詳細
 - LiDAR ローカリゼーションによる自己位置の推定
 - ✧ 処理内容

- 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）とLiDARスキャンにより取得した点群データ双方の点群データをマッチングして自己位置推定を行う
 - ✧ 利用するライブラリ
 - Point cloud library（ソフトウェア・ライブラリ【SL006】を参照）
 - ✧ 利用するアルゴリズム
 - NDT Registration（アルゴリズム【AL001】を参照）

6. 【FN006】LiDAR測位情報チェック

- 機能概要
 - 3D LiDARによる測位情報の信頼性評価を行う機能
- フローチャート

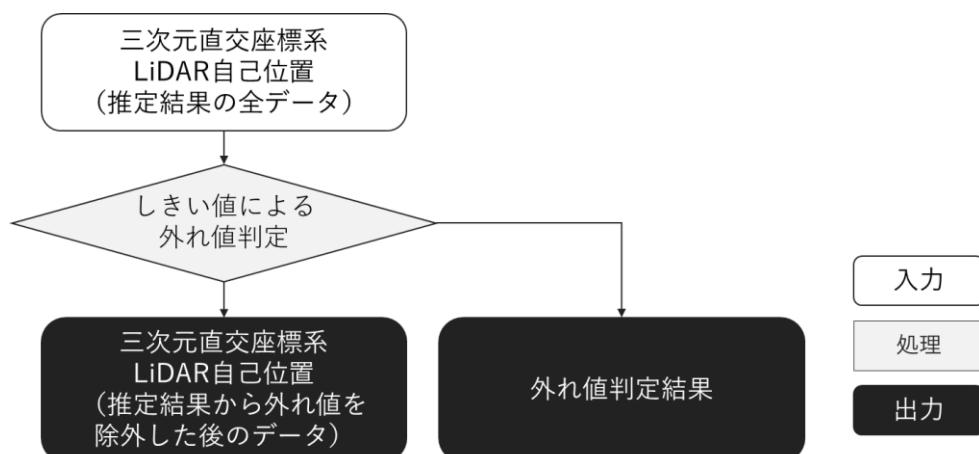


図 3-17 LiDAR測位情報チェック機能のフローチャート

- データ仕様
 - 入力
 - ✧ 三次元直交座標系 LiDAR 自己位置（推定結果の全データ）
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(三次元直交座標系)
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - 出力
 - ✧ 三次元直交座標系 LiDAR 自己位置（推定結果から外れ値を除外した後のデータ）
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(三次元直交座標系)
 - 形式

- ROS メッセージ形式
- データ詳細
 - 内部連携インターフェース【IF201】を参照
- ✧ 外れ値判定結果
 - 内容
 - 【AL001】NDT Registration で算出した信頼性スコアが 2.0 を上回る値の場合に外れ値として設定する。
 - 信頼性スコアは、現実世界のスキャン点群と点群マップを比較し、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た値(0以上)を示す。本判定では昨年度の実証実験を踏まえ、複数の飛行試験データを分析・検証した結果、2.0 未満であればスキャンマッチングが動作していることが確認できたため、閾値は 2.0 とした。
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - なし
- 機能詳細
 - LiDAR 測位情報チエッカー
 - ✧ 処理内容
 - LiDAR ローカリゼーションの結果を測位情報統合機能に入力するかどうかの判定を、しきい値に合わせて行う
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

7. 【FN007】LiDAR測位情報座標変換

- 機能概要
 - 3D LiDARによる測位情報を基準座標系へ変換する機能
- フローチャート

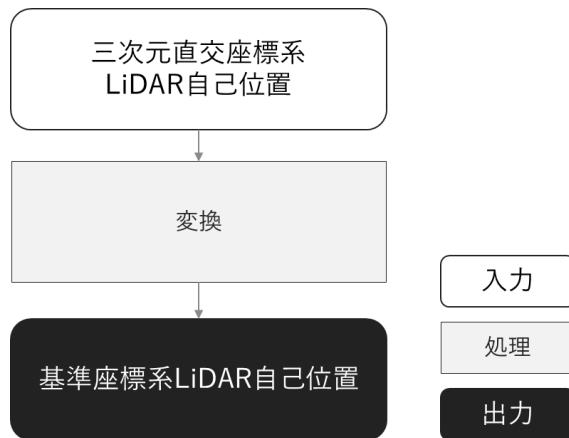


図 3-18 LiDAR測位情報座標変換機能のフローチャート

- データ仕様
 - 入力
 - ✧ 三次元直交座標系 LiDAR 自己位置
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(三次元直交座標系)
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - 出力
 - ✧ 基準座標系 LiDAR 自己位置
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を基準座標系に直したデータ
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF206】を参照
 - 機能詳細
 - 座標系の変換
 - ✧ 処理内容
 - LiDAR の自己位置推定情報の座標系を、三次元直交座標系から基準座標系に変換する

- ✧ 利用するライブラリ
 - proj (ソフトウェア・ライブラリ【SL007】を参照)
 - ✧ 利用するアルゴリズム
 - なし
8. 【FN008】GNSS測位情報チェック
- 機能概要
 - GNSSによる測位情報の信頼性評価を行う機能
 - フローチャート

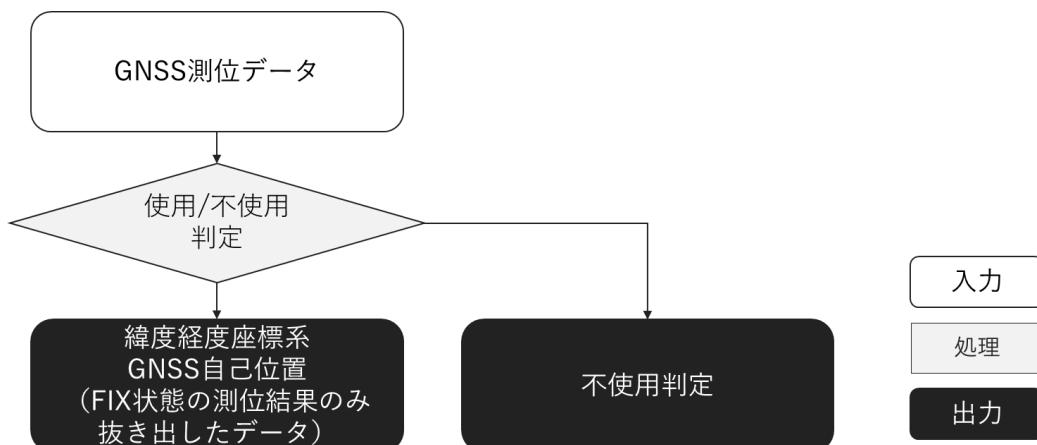


図 3-19 GNSS測位情報チェック機能のフローチャート

- データ仕様
 - 入力
 - ✧ GNSS測位データ
 - 内容
 - GNSS測位で得た緯度経度高さ情報
 - 形式
 - NMEA形式
 - データ詳細
 - 内部連携インターフェース【IF202】を参照
 - 出力
 - ✧ 緯度経度座標系 GNSS自己位置 (測位結果から外れ値を除外した後のデータ)
 - 内容
 - GNSS測位で取得した自己位置 (緯度経度座標系)
 - 形式
 - ROSメッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF202】を参照
 - ✧ 不使用判定

- 内容
 - GNSS 測位で取得した自己位置（緯度経度座標系）の信頼性スコアに対して、信頼性スコアが 50 を「使用」、50 以外の値を全て「不使用」と判定するよう設定する。
 - GNSS の信頼性は一般的に Fix、Float、Single で分類されており、徐々に位置精度が減少していることを示している。この値は内部的にそれぞれ 50、48、32 という値で表現されている。
 - 本システムでは、GNSS の状態が Fix の場合のみ【FN011】測位情報統合機能に GNSS 測位結果を使用するため、GNSS 信頼性スコア 50 以外の数値を全て「不使用」と判定した。
- 形式
 - ROS メッセージ形式
- データ詳細
 - なし
- 機能詳細
 - GNSS 測位情報チェック
 - ✧ 処理内容
 - GNSS 測位データを測位情報統合機能に入力するかどうかの判定を、しきい値に合わせて行う
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし
- 9. 【FN009】GNSS 測位情報座標変換
- 機能概要
 - GNSS による測位情報を基準座標系へ変換する機能
- フローチャート

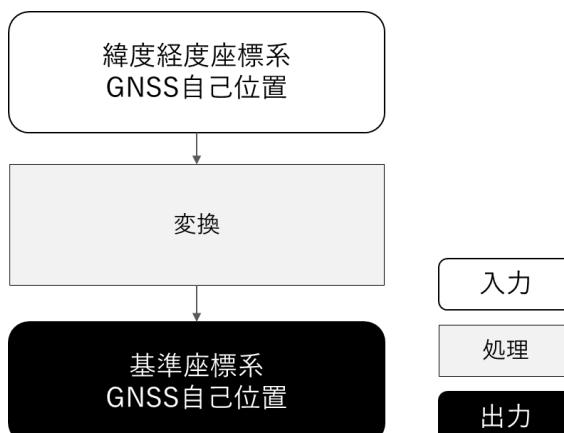


図 3-20 GNSS 測位情報座標変換機能のフローチャート

- データ仕様

- 入力

- ✧ 緯度経度座標系 GNSS 自己位置

- 内容

- GNSS 測位で取得した自己位置（緯度経度座標系）

- 形式

- ROS メッセージ形式

- データ詳細

- 内部連携インターフェース【IF202】を参照

- 出力

- ✧ 基準座標系 GNSS 自己位置

- 内容

- GNSS 測位で取得した自己位置（基準座標系）

- 形式

- ROS メッセージ形式

- データ詳細

- 内部連携インターフェース【IF202】を参照

- 機能詳細

- 座標系の変換

- ✧ 処理内容

- GNSS 測位データの座標系を、緯度経度座標系から基準座標系に変換する

- ✧ 利用するライブラリ

- proj（ソフトウェア・ライブラリ【SL007】を参照）

- ✧ 利用するアルゴリズム

- なし

10. 【FN010】ビジュアルイナーシャルオドメトリ (VIO)

- 機能概要

- ドローン内臓カメラで撮影した、連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU情報を加味して移動量を推定する機能

- フローチャート

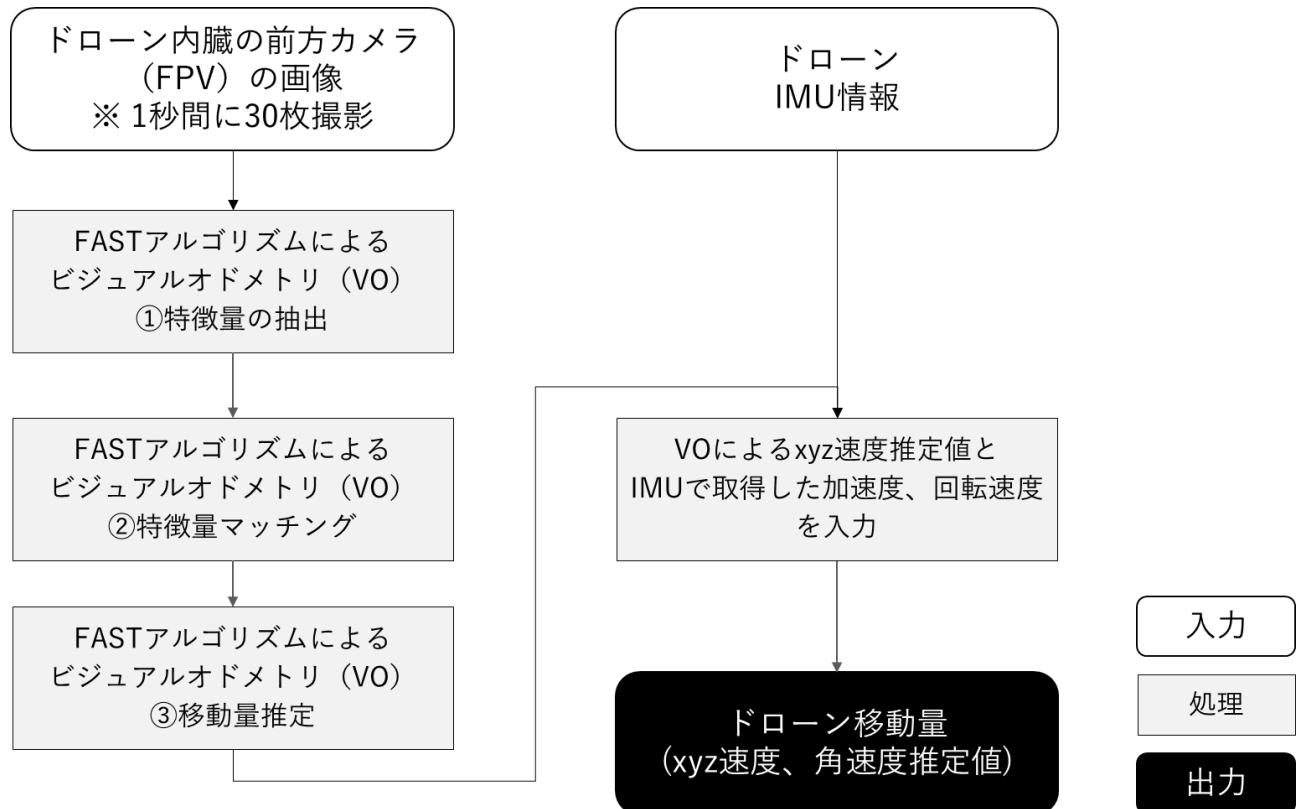


図 3-21 ビジュアルイナーシャルオドメトリ (VIO) 機能のフローチャート

- データ仕様

- 入力

- ❖ 画像データ
 - 内容
 - ドローン内臓の前方カメラ (FPV) の画像 ※ 1 秒間に 30 枚撮影
 - 形式
 - ROS メッセージ形式 ※sensor message / image 形式
 - データ詳細
 - 内部連携インターフェース 【IF203】を参照
- ❖ IMU 情報
 - 内容
 - IMU で取得した加速度、角速度
 - 形式
 - ROS メッセージ形式 ※sensor message / IMU 形式
 - データ詳細

- 内部連携インターフェース【IF204】を参照
- 出力
 - ✧ ドローン移動量
 - 内容
 - 1つ前のフレームからの移動量(xyz速度、角速度推定値)
 - 形式
 - ROSメッセージ形式 ※twist message形式
 - データ詳細
 - 開発したアルゴリズム【AL101】を参照
 - 内部連携インターフェース【IF206】を参照
 - 機能詳細
 - ビジュアルオドメトリ(VO)
 - ✧ 処理内容
 - ①特徴量の抽出
 - カメラで撮影した画像中から特徴量を検出する
 - ②特徴量マッチング
 - ある時点に撮影された画像で得た特徴量と、その1枚後に撮影された画像から新しく得られた特徴量の対応関係を求める
 - ③移動量推定
 - “②特徴量マッチング”で特定した特徴点のペアを用い、カメラの姿勢や移動量を求める。
 - ✧ 利用するライブラリ
 - OpenCV(ソフトウェア・ライブラリ【SL008】を参照)
 - ✧ 利用するアルゴリズム
 - FASTアルゴリズム(アルゴリズム【AL002】を参照)
 - ビジュアルイナーシャルオドメトリ(VIO)
 - ✧ 処理内容
 - EKF(Extended Kalman Filter)にビジュアルオドメトリにより推定された速度と、IMUの加速度、回転速度を入力し、1つ前のフレームからの移動量(xyz速度、角速度推定値)を出力する
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - FASTアルゴリズム(アルゴリズム【AL002】を参照)
 - EKF(Extended Kalman Filter)(アルゴリズム【AL003】を参照)
 - ビジュアルイナーシャルオドメトリ(アルゴリズム【AL101】を参照)

11. 【FN011】測位情報統合

- 機能概要
 - 複数の測位情報を統合する機能
 - 今回の実証実験における開発では、2022年度に開発した既存機能に【FN010】ビジュアルライナーシャルオドメトリ（VIO）から出力されたデータを統合し、自己位置推定のロバスト性を向上させた
- フローチャート

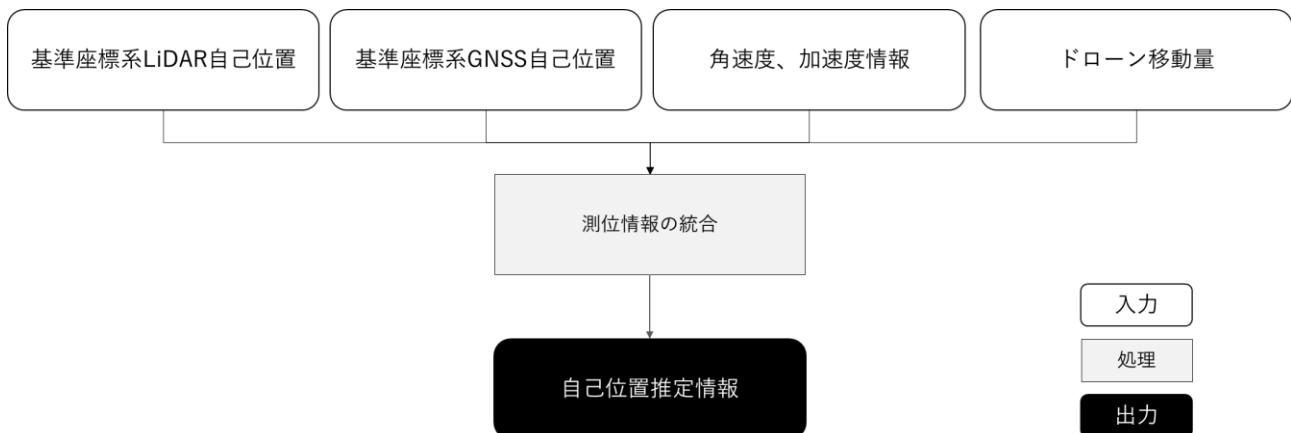


図 3-22 測位情報統合機能のフローチャート

- データ仕様
 - 入力
 - ✧ 基準座標系 LiDAR 自己位置
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を基準座標系に直したデータ
 - 形式
 - ROS メッセージ形式 ※pose 形式
 - データ詳細
 - 内部連携インターフェース【IF206】を参照
 - ✧ 基準座標系 GNSS 自己位置
 - 内容
 - GNSS 測位で取得した自己位置（基準座標系）
 - 形式
 - ROS メッセージ形式 ※pose 形式
 - データ詳細
 - 内部連携インターフェース【IF202】を参照
 - ✧ 角速度、加速度情報
 - 内容
 - IMU からの角速度、加速度情報
 - 形式

- ROS メッセージ形式 ※IMU 形式
- データ詳細
 - 内部連携インターフェース【IF204】を参照
- ❖ ドローン移動量
 - 内容
 - ビジュアルインサルオドメトリ(VIO)より得られる移動量(xyz速度、角速度推定値)
 - 形式
 - ROS メッセージ形式 ※twist message 形式
 - データ詳細
 - 内部連携インターフェース【IF206】を参照
 - 開発したアルゴリズム【AL101】を参照
- 出力
 - ❖ 自己位置推定情報
 - 内容
 - 各種測位情報が統合された結果、推定される自己位置情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム【AL102】を参照
- 機能詳細
 - 測位情報の統合
 - ❖ 処理内容
 - LiDAR ユニット、GNSS ユニット、IMU ユニット、カメラユニット(ビジュアルインサルオドメトリ(VIO))から得られた複数の測位情報を統合する
 - ❖ 利用するライブラリ
 - なし
 - ❖ 利用するアルゴリズム
 - EKF (Extended Kalman Filter) (アルゴリズム【AL003】を参照)
 - 測位情報の統合アルゴリズム (アルゴリズム【AL102】を参照)

12. 【FN012】障害物回避

- 機能概要
 - 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
- フローチャート

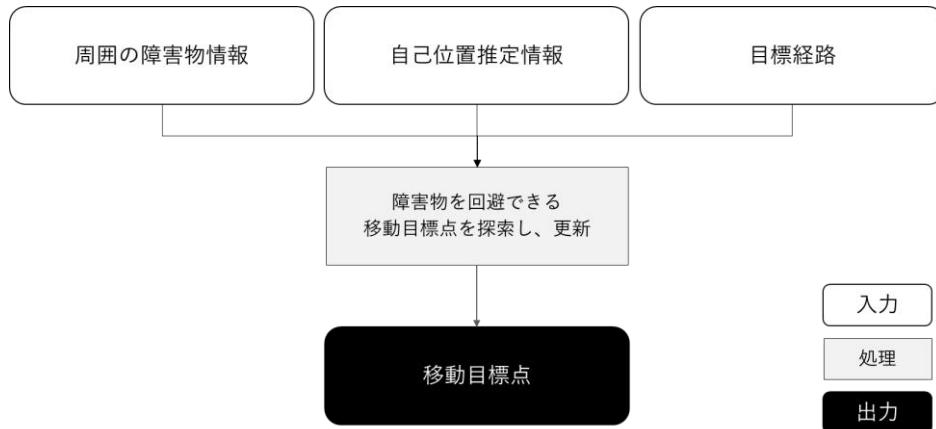


図 3-23 障害物回避機能のフローチャート

- データ仕様
 - 入力
 - ✧ 周囲の障害物情報
 - 内容
 - 機体に搭載した 3D LiDAR から得られるスキャン結果を元に得られた周囲の障害物情報
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - ✧ 機体の自己位置
 - 内容
 - 測位情報統合の結果得られた自己位置推定情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム【AL102】を参照
 - ✧ 目標経路
 - 内容
 - ドローンが飛行する目標経路
 - 形式
 - ROS メッセージ形式

- データ詳細
 - 内部連携インターフェース【IF205】を参照
- 出力
 - ✧ 更新された移動目標点
 - 内容
 - ドローンの自己位置と目標経路から決めた移動目標点を、障害物を回避するように更新する
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
- 機能詳細
 - 障害物判定
 - ✧ 処理内容
 - 機体に搭載した3D LiDARから得られるスキャン結果を元に、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。
 - ✧ 利用するライブラリ
 - Octomap（ソフトウェア・ライブラリ【SL009】を参照）
 - ompl（ソフトウェア・ライブラリ【SL010】を参照）
 - ✧ 利用するアルゴリズム
 - なし
 - 障害物回避
 - ✧ 処理内容
 - 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
 - ✧ 利用するライブラリ
 - Octomap（ソフトウェア・ライブラリ【SL009】を参照）
 - ompl（ソフトウェア・ライブラリ【SL010】を参照）
 - ✧ 利用するアルゴリズム
 - なし

13. 【FN013】目標速度算出

- 機能概要
 - 自己位置と目標位置からドローンの移動速度を決定/指示する機能
- フローチャート

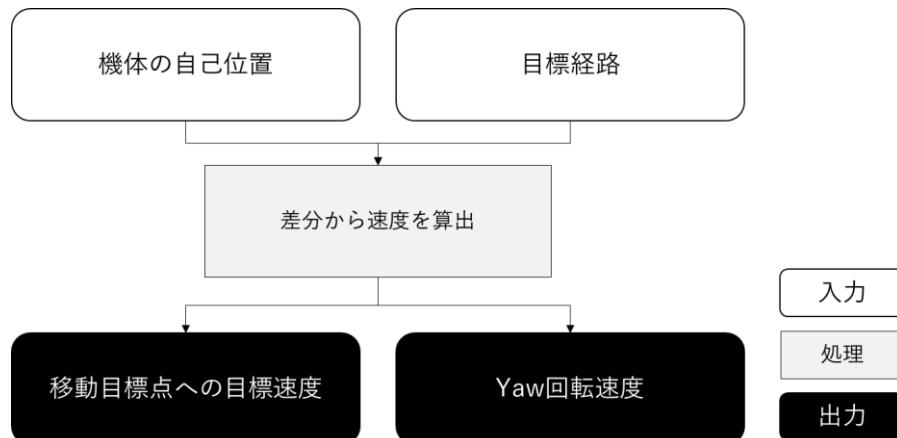


図 3-24 目標速度算出機能のフローチャート

- データ仕様
 - 入力
 - ✧ 機体の自己位置
 - 内容
 - 測位情報統合の結果から得られた自己位置推定情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム 【AL102】を参照
 - ✧ 目標経路
 - 内容
 - ドローンが飛行する目標経路
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インターフェース 【IF205】を参照
 - 出力
 - ✧ 移動目標点への目標速度
 - 内容
 - ドローンの自己位置と移動目標点の差分から定まった、目標方向に進む進行方向への速度
 - 形式
 - ROS メッセージ形式

- データ詳細
 - 内部連携インターフェース【IF205】を参照
- ✧ Yaw回転速度
 - 内容
 - ドローンの自己位置と移動目標点の差分から定めた、目標方向に進む角速度
 - 形式
 - ROSメッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF204】を参照
- 機能詳細
 - 目標速度の算出
 - ✧ 処理内容
 - 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
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

14. 【FN014】機体制御【DJI Onboard SDK】

- 機能概要
 - 飛行指示に従い、機体を制御する
- フローチャート

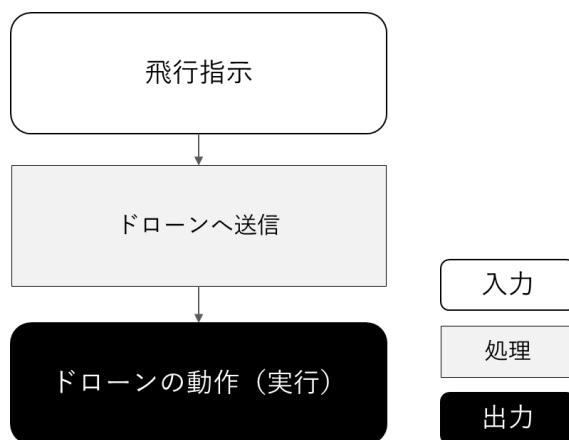


図 3-25 機体制御【DJI Onboard SDK】機能のフローチャート

- データ仕様
 - 入力
 - ✧ 飛行指示
 - 内容
 - ELASからのドローンに送られる飛行指示

- 形式
 - DJI OSDK メッセージ形式
- データ詳細
 - 内部連携インターフェース【IF205】を参照
- 出力
 - ✧ ドローンの動作（実行）
 - 内容
 - 現実世界でのドローンの動作（自律飛行、緊急停止）
 - 形式
 - なし
 - データ詳細
 - なし
 - 機能詳細
 - 機体の制御
 - ✧ 処理内容
 - 飛行指示に従い機体を制御する
 - ✧ 利用するライブラリ
 - DJI Onboard SDK（ソフトウェア・ライブラリ【SL012】を参照）
 - ✧ 利用するアルゴリズム
 - なし

15. 【FN015】地形データ切り替え

- 機能概要
 - 3D都市モデルとBIMの統合マップをSENSYN CORE Pilot画面の3Dマップ(Cesium)上で表示するために、3D都市モデルが存在する場所では利用する地形データをCesiumのデフォルトの地形データからPLATEAU-Terrainが持つ地形データへ切り替えを行う機能
- フローチャート

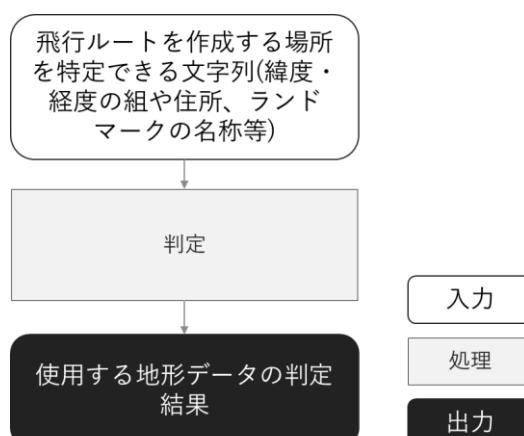


図 3-26 地形データ切り替え機能のフローチャート

- データ仕様

- 入力

- ✧ 飛行ルートを作成する場所を特定できる文字列(緯度・経度や住所、ランドマークの名称等)

- 内容

- 緯度・経度の組や住所、ランドマークの名称等、飛行ルートを作成する場所を示す情報

- 住所や建物名はジオコーディングし、最終的には緯度・経度の組として扱う

- 形式

- 文字列形式

- データ詳細

- 外部連携インターフェース【IF301】【IF302】を参照

- 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベースを利用（外部連携インターフェース【IF303】を参照）

- 出力

- ✧ 使用する地形データの判定結果

- 内容

- Cesium のデフォルト地形データと PLATEAU-Terrain のどちらを使用するかを意味する列挙値

- 形式

- 文字列形式又は数値形式

- データ詳細

- なし

- 機能詳細

- 地形データ切り替え

- ✧ 処理内容

- 3D都市モデルとBIMの統合マップをSENSYN CORE Pilot上で表示するために地形データの切り替えを行う

- Cesium のデフォルトの地形データ上に 3D 都市モデルを表示させると、対地高度の前提となる標高データが異なり、3D 都市モデルが地面に埋まる又は地表から浮いた状態で表示される場合があるため、この問題を解決するために Cesium 上で利用する地形データを適切に置き換える

- SENSYN CORE Pilot 上で PLATEAU-Terrain を使用せずに作成したルートに影響を与えないようにする

- ただし、PLATEAU-Terrain が持つ地形情報は日本国内のみであるため、日本国外で飛行経路を作成する場合は Cesium のデフォルト地形データを利用する必要がある

- この切り替えを自動で行う

- ✧ 利用するライブラリ

- Cesium Geocode API（ソフトウェア・ライブラリ【SL013】を参照）

◆ 利用するアルゴリズム

- 日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム（アルゴリズム【AL103】を参照）
- PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム（アルゴリズム【AL104】を参照）

16. 【FN016】経路計画 UI

- 機能概要
 - ドローンの経路計画を行う
- フローチャート

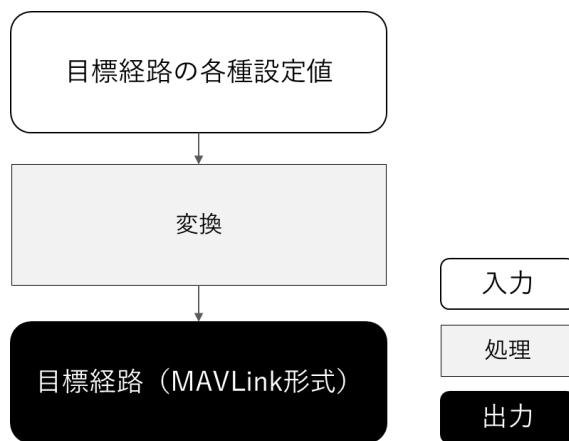


図 3-27 経路計画 UI のフローチャート

- データ仕様
 - 入力
 - ◆ 目標経路の各種設定値
 - 内容
 - 目標経路（飛行ルート）の情報（各ウェイポイントの緯度経度高さ、各ウェイポイント間の移動速度）
 - 形式
 - 数値形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
 - 出力
 - ◆ 目標経路（MAVLink 形式）
 - 内容
 - 目標経路（飛行ルート）のルート情報
 - 形式
 - MAVLink 形式 ※waypoints
 - データ詳細

- 内部連携インターフェース【IF205】を参照

- 機能詳細

- 経路計画UI

- ✧ 処理内容

- ユーザーがUI上で入力した飛行ルートの各情報をMAVLink形式に変換する

- ✧ 利用するライブラリ

- MAVLink toolchain(ソフトウェア・ライブラリ【SL015】を参照)

- ✧ 利用するアルゴリズム

- なし

17. 【FN017】PLATEAU-Terrain 対応機能

- 機能概要

- SENSYN CORE Pilotを用いてドローンの経路計画を行う際に、SENSYN CORE Pilotで用いられている標高データと異なる標高データ(PLATEAU-Terrain)を用いても意図どおりの飛行経路を作成する機能

- フローチャート

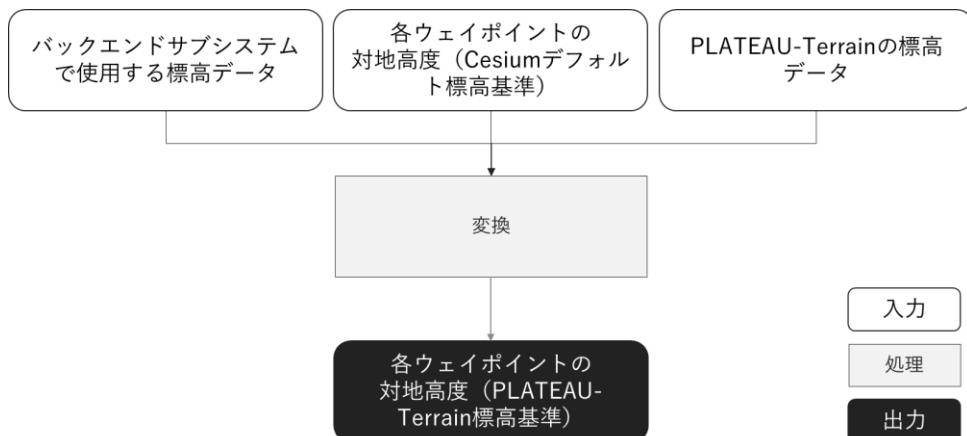


図 3-28 PLATEAU-Terrain 対応機能のフローチャート

- データ仕様

- 入力

- ✧ 各ウェイポイントの対地高度(PLATEAU-Terrain 標高基準)

- 内容

- PLATEAU-Terrainの標高データを基準とする、各ウェイポイントの対地高度

- 形式

- 数値形式

- データ詳細

- 外部連携インターフェース【IF302】を参照

- ✧ PLATEAU-Terrainの標高データ

- 内容

- PLATEAU-Terrain が持つ標高データ
- 形式
 - PLATEAU-Terrain 形式
- データ詳細
 - 外部連携インターフェース【IF302】を参照
- ✧ バックエンドサブシステムが利用している標高データ
 - 内容
 - バックエンドサブシステムが持つ標高データ
 - 形式
 - quantized-mesh 形式
 - データ詳細
 - バックエンドサブシステムで使用する標高データ ※SENSYN CORE Pilot ベースで開発を行うため、詳細は開示不可
- 出力
 - ✧ 各ウェイポイントの対地高度（バックエンドサブシステムの標高基準）
 - 内容
 - バックエンドサブシステムが利用している標高データを基準とする、飛行経路上の各ウェイポイントの対地高度
 - 形式
 - 数値形式
 - データ詳細
 - 外部連携インターフェース【IF301】を参照
- 機能詳細
 - 各ウェイポイントの高度変換
 - ✧ 処理内容
 - 3D都市モデルとBIMの統合マップ（PLATEAU-Terrainに合わせたデータ）を基にSENSYN CORE Pilotを用いてドローンの経路計画を行う際に、バックエンドサブシステム上で用いられている標高データとPLATEAU-Terrainの標高データを整合させる
 - SENSYN CORE Pilotにおいて経路計画（ルート作成）時に利用するバックエンドサブシステムの標高データ（TP高度ではない、SRTM等の標高データ）はPLATEAU-Terrainのデータ（TP高度）とは異なる
 - 利用する標高データ間の差分はドローンの飛行に影響を与えるため、各ウェイポイントの高度を適切に変換し、異なる標高データを用いても意図どおりの高度を飛ばせるようにする処理を行う
 - ✧ 利用するライブラリ
 - PLATEAU-Terrainの標高データ（外部連携インターフェース【IF302】を参照）
 - ✧ 利用するアルゴリズム
 - なし

18. 【FN018】ROS メッセージ双方向変換

● 機能概要

- SENSYN CORE Pilot と ROS により動作するロボットの間で、情報を双方向にやりとりするのに適切な形式に変換する機能

● フロー チャート

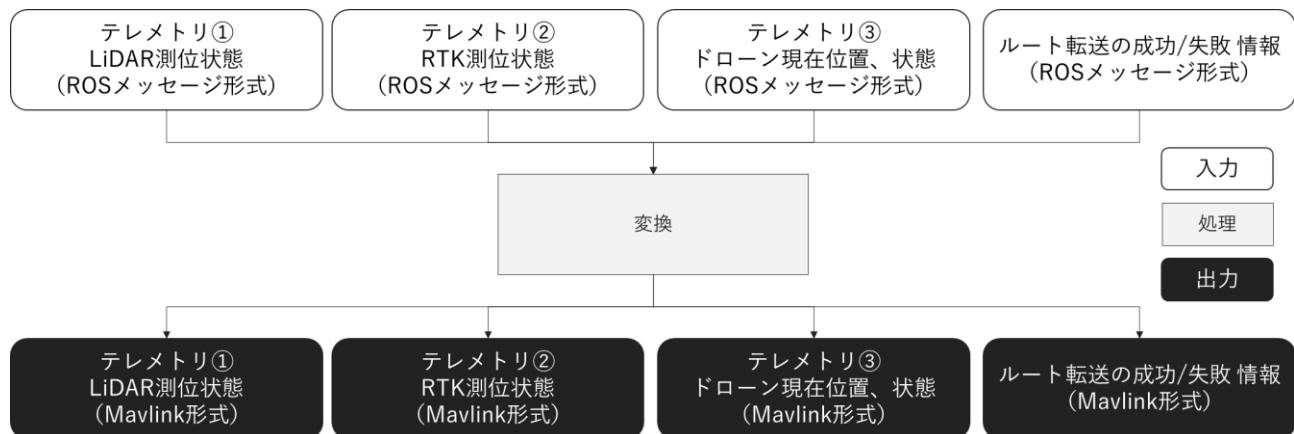


図 3-29 ROS メッセージ双方向変換機能のフロー チャート（ドローンから SENSYN CORE Pilot への入力）

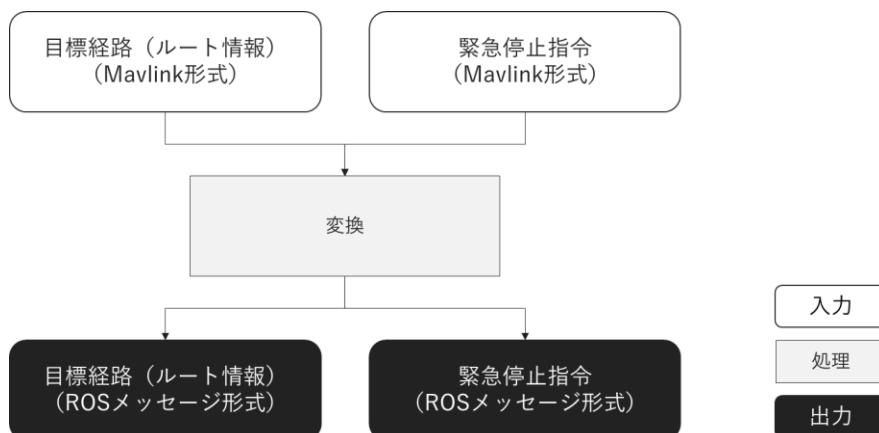


図 3-30 ROS メッセージ双方向変換機能のフロー チャート（SENSYN CORE Pilot からドローンへの入力）

● データ仕様

➢ 入力

❖ ドローン (ELAS) からの入力

● 内容

- テレメトリー① LiDAR 測位状態
- テレメトリー② RTK 測位状態
- テレメトリー③ ドローン現在位置、状態
- ルート転送の成功/失敗 情報

● 形式

- ROS メッセージ形式

- データ詳細
 - 内部連携インターフェース【IF205】を参照
- ✧ SENSYN CORE Pilotからの入力
 - 内容
 - 目標経路（ルート情報）
 - 緊急停止指令
 - 形式
 - MAVLink形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
- 出力
 - ✧ ドローン（ELAS）への出力
 - 内容
 - 目標経路（ルート情報）
 - 緊急停止指令
 - 形式
 - ROSメッセージ形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
 - ✧ SENSYN CORE Pilotへの出力
 - 内容
 - テレメトリー①LiDAR測位状態
 - テレメトリー②RTK測位状態
 - テレメトリー③ドローン現在位置、状態
 - ルート転送の成功/失敗 情報
 - 形式
 - MAVLink形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
- 機能詳細
 - ROSメッセージ双方向変換
 - ✧ 処理内容
 - SENSYN CORE PilotとROSにより動作するロボットの間で情報を双方向にやりとりするため、送信するメッセージのメッセージ形式を送信先に合わせてROSメッセージ形式、MAVLink形式のそれぞれ適切な形式に変換する
 - ✧ 利用するライブラリ
 - Boost（ソフトウェア・ライブラリ【SL011】を参照）
 - ✧ 利用するアルゴリズム

- テレメトリーについては、辞書ベースのデータマッピングアルゴリズム（アルゴリズム【AL004】を参照）
- ルート転送については、State machine ベースの変換アルゴリズム（アルゴリズム【AL005】を参照）

19. 【FN019】ドローン位置可視化

- 機能概要
 - ドローンの現在位置をリアルタイムで表示する
- フローチャート

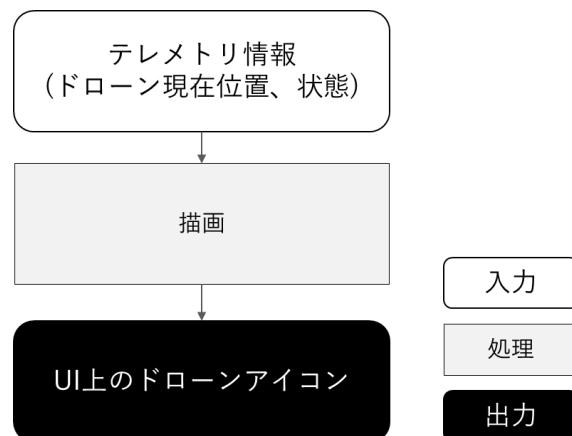


図 3-31 ドローン位置可視化機能のフローチャート

- データ仕様
 - 入力
 - ✧ テレメトリー情報
 - 内容
 - ドローン現在位置、状態を示すテレメトリー情報
 - 形式
 - MAVLink 形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
 - 出力
 - ✧ SENSYN CORE Pilot の地図上でのドローンアイコン描画
 - 内容
 - ドローン位置を示すアイコンによりドローンの現在位置を Cesium 上で描画する
 - 形式
 - GLB 形式
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
 - 機能詳細

- ドローン位置の可視化
 - ✧ 処理内容
 - ドローン（ELAS）から受信したテレメトリー情報を、UI上に描画する
 - ✧ 利用するライブラリ
 - PLATEAU-Terrain の標高データ（外部連携インターフェース【IF302】を参照）
 - egm96-universal（ソフトウェア・ライブラリ【SL014】を参照）
 - ✧ 利用するアルゴリズム
 - なし

20. 【FN020】緊急停止

- 機能概要
 - ドローンを緊急停止させる指令を ELAS に送る機能
- フローチャート

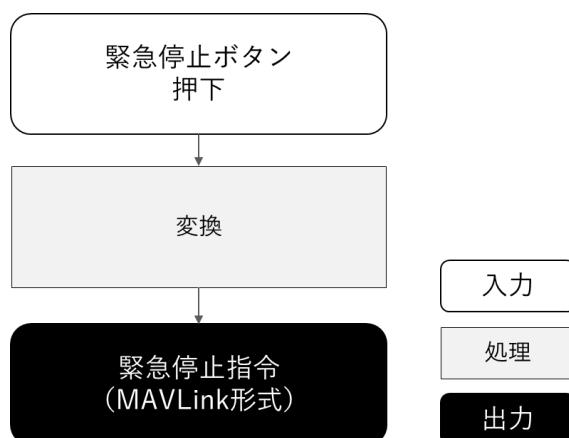


図 3-32 緊急停止機能のフローチャート

- データ仕様
 - 入力
 - ✧ ユーザーによる UI からの入力（ボタンクリック）
 - 内容
 - UI 上のドローン緊急停止ボタンを押下
 - 形式
 - なし
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
 - 出力
 - ✧ 緊急停止指令（MAVLink 形式）
 - 内容
 - ドローンに緊急停止を指示する命令
 - 形式

- MAVLink 形式
- データ詳細
 - 内部連携インターフェース【IF205】を参照
- 機能詳細
 - 緊急停止
 - ✧ 処理内容
 - UI 上でのボタン押下を MAVLink 形式の命令に変換する
 - ✧ 利用するライブラリ
 - SENSYN CORE Pilot ベースで開発を行うため、詳細は開示不可。
 - ✧ 利用するアルゴリズム
 - なし

3-3. アルゴリズム

3-3-1. 利用したアルゴリズム

表 3-4 利用したアルゴリズム一覧

ID	アルゴリズムを利用した機能	名称	説明	選定理由
AL001	FN005	NDT Registration	<ul style="list-style-type: none"> 異なる二つの点群間の位置合わせに相応しい移動・回転量を見つけるためのアルゴリズム 事前に準備した点群地図と LiDAR から得られるスキャンを位置合わせすることで点群地図における LiDAR の位置を算出する 	<ul style="list-style-type: none"> 高速にスキャン同士のマッチングが可能であり、自律移動のようなリアルタイムアプリケーションに適しているため。 大規模な地図データを用いても高速に動作するため
AL002	FN010	FAST アルゴリズム	<ul style="list-style-type: none"> 画像中からコーナー特徴量を検出するアルゴリズムで、非常に高速に動作する 	<ul style="list-style-type: none"> リソースの限られたデバイスなどデモンストレーション実行が容易で、ドローンや移動ロボットなどのリアルタイム性が要求されるアプリケーションに適しているため。
AL003	FN010 FN011	拡張カルマンフィルタ：EKF (Extended Kalman Filter)	<ul style="list-style-type: none"> 複数の測位情報の統合を行うことで真値に近くノイズに強い結果を得ることができるアルゴリズム 	<ul style="list-style-type: none"> 他の情報統合アルゴリズムなどに比べて高速に実行することができるため
AL004	FN018	辞書ベースのデータマッピングアルゴリズム	<ul style="list-style-type: none"> 複数システム間でのプロトコル変換を行うアルゴリズム ROS メッセージで表 	<ul style="list-style-type: none"> 多種類のデータを高速にマッピングするのに適しているため 状態や順番を考慮せ

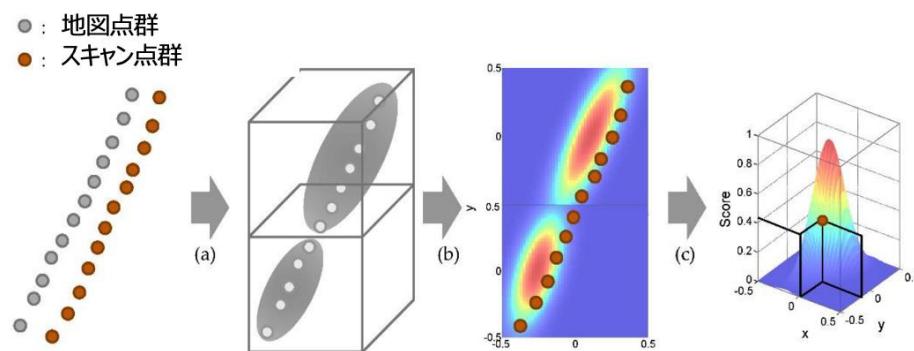
			現されたテレメトリー情報を MAVLink に変換するために利用	利⽤できるアルゴリズムであるため
AL005	FN018	State machine ベースの変換アルゴリズム	<ul style="list-style-type: none"> 複数システム間でのプロトコル変換を行うアルゴリズム MAVLink で表現されている飛行ミッションを ROS メッセージに変換するために利用 	<ul style="list-style-type: none"> 辞書ベースのマッピングではできない、複数のメッセージをまとめるため 状態や順番を考慮できるアルゴリズムであるため

1) 【AL001】NDT Registration

● 計算量

- 地図点群が持つ点の数を M としたとき地図点群をボクセルに分割し各セル内のポイントから正規分布を割り出すための計算量は $O(M)$
- スキャン点群が持つ点の数を N とし、ボクセルの数を v としたとき、また反復的な最適化が行われる回数が κ であったときの計算量は $O(\kappa C)$ となる。このとき $C = Nv$ 。
- また反復ごとに収束したかチェックを行うがこのときの計算量は $O(1)$
- 全体を通しての NDT Registration の計算量は $O(M + \kappa C)$ となる

● イメージ

図 3-33 NDT Registration による各点が持つ一致度評価のイメージ⁶

2) 【AL002】FAST アルゴリズム

● 計算量

- 画像中のコーナーをターゲットとして検出する高速動作する特徴量検出アルゴリズム。画素周辺の

⁶ 出典 : A Review of Point Cloud Registration Algorithms for Laser Scanners: Applications in Large-Scale Aircraft Measurement (<https://www.mdpi.com/2076-3417/12/20/10247>)

明暗さのパターンによりコーナーを検出する。

- 画像サイズが $N \times M$ である場合計算量は $O(NM)$

- イメージ

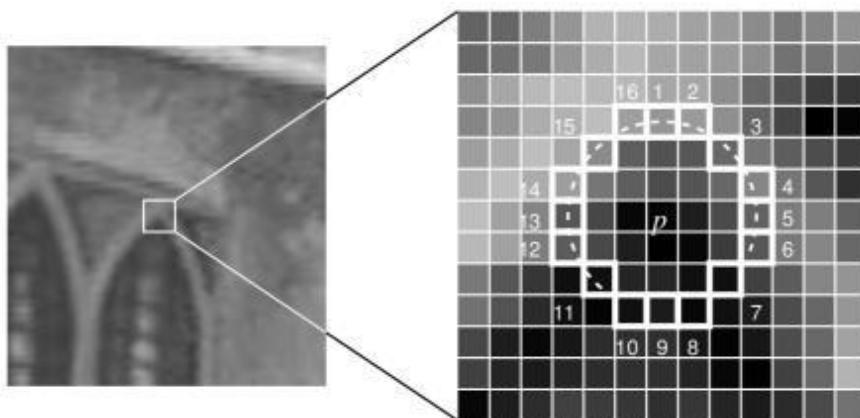


図 3-34 FAST アルゴリズムによるビジュアルオドメトリのイメージ⁷

3) 【AL003】拡張カルマンフィルタ：EKF (Extended Kalman Filter)

- 計算量

- ノイズを含む複数の情報を統合し、センサの弱点を補い合うことでより真値に近い情報へとフィルタリングを行う。
- 状態ベクトルの次元数を N としたとき $O(N^3)$

- イメージ

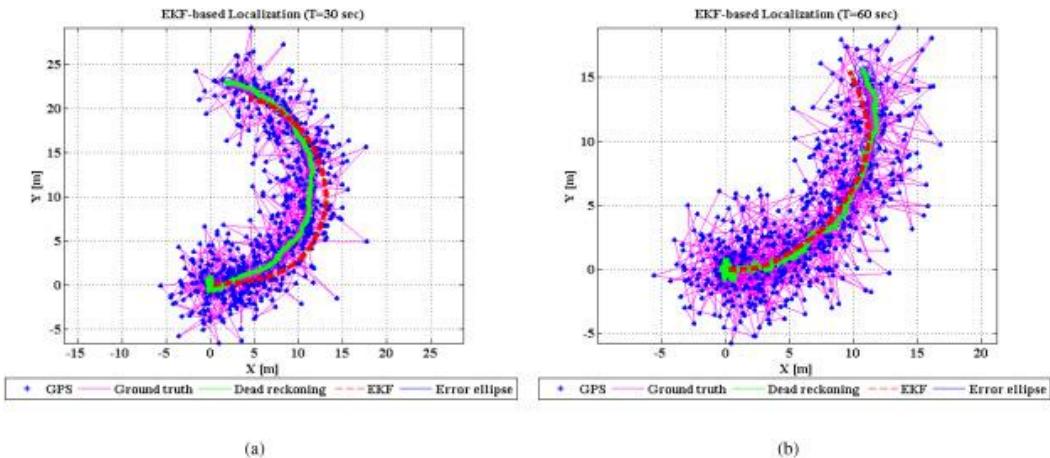


図 3-35 EKF(Extended Kalman Filter)による自己位置推定軌道（赤色点）のイメージ⁸

⁷ 出典：Edward Rosten and Tom Drummond, “Machine learning for high speed corner detection” in 9th European Conference on Computer Vision, vol. 1, 2006, pp. 430–443.

⁸ 出典：Extended Kalman Filter-based localization algorithm by edge computing in Wireless Sensor Networks (<https://www.sciencedirect.com/science/article/pii/S2352864820302601#fig5>)

4) 【AL004】辞書ベースのデータマッピングアルゴリズム

- 計算量

- あるROSメッセージについて、辞書を参照し1対1で対応するMavlinkメッセージを特定し、データマッピングを行う（逆方向のMavlinkメッセージ to ROSメッセージについても同様）
- 1対1で対応する相手方メッセージを検索するため、その計算量は $O(1)$
- 変換を行うメッセージの数をnとしたとき $O(n)$

- イメージ

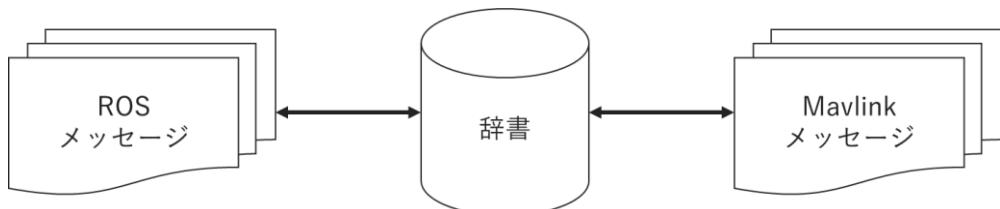


図 3-36 辞書ベースのデータマッピングアルゴリズムによるROSメッセージ変換のイメージ

5) 【AL005】State machine ベースの変換アルゴリズム

- 計算量

- あるウェイポイント情報を構築するために必要な各種情報が複数のMavlinkメッセージで送られてくるため、そのメッセージを受信し変換、統合することで1つのウェイポイント情報を作成する
- 1つのウェイポイントを作成する際に必要なMavlinkメッセージの数をnとしたとき $O(n)$

- イメージ

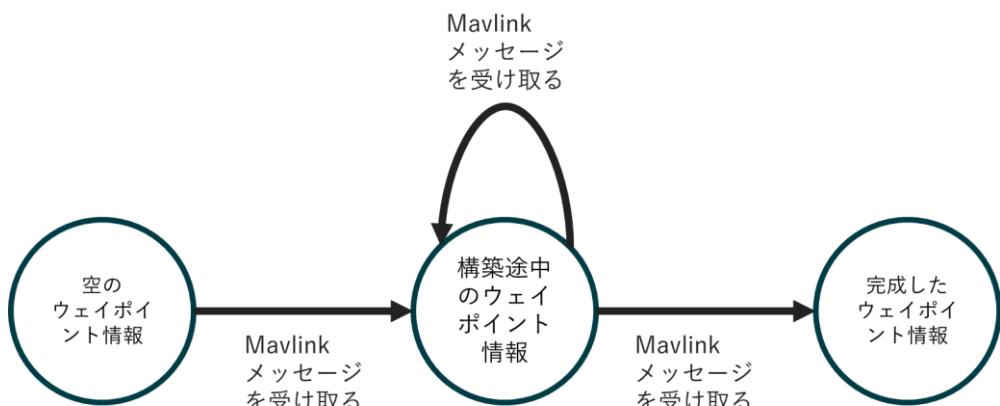


図 3-37 State machine ベースの変換アルゴリズムによるROSメッセージ変換のイメージ

3-3-2. 開発したアルゴリズム

1) 【AL101】ビジュアルイナーシャルオドメトリ (VIO)

- 本アルゴリズムを利用した機能
 - 【FN010】ビジュアルイナーシャルオドメトリ

● アルゴリズムの詳細

ドローン自律運航システムのロバスト性の更なる向上を目的として、LiDARとGNSS（衛星測位システム）による既存の自己位置推定技術に加え、新たにビジュアルイナーシャルオドメトリ (VIO) の開発を行った。ビジュアルイナーシャルオドメトリ (VIO) とは画像中からコーナー特微量を検出するアルゴリズムで、非常に高速に動作する FAST アルゴリズム (AL002) により抽出した各画像に対する特徴点と IMU 情報により、局所的なカメラの移動量を推定するアルゴリズムである。

先行研究として、Anand George "Analysis of Visual-Inertial Odometry Algorithms for Outdoor Drone Applications"⁹などが挙げられる。また、既存技術としては FAST アルゴリズムの他にスケール不編性をもち異なる、距離やズーム状態で取得した画像間でも特微量の精密なマッチングが可能な SIFT アルゴリズム、速度と精度のバランスが良いと言われている。SIFT に近い高精度な特微量抽出が可能であり、計算速度が比較的高速な AKAZE アルゴリズムなどがあるが、本ユースケースではドローン飛行を行うため、ドローン飛行に必要なリアルタイム性を満たす、高速な FAST アルゴリズムを採用した。

ステップ

1. カメラにより取得した単一画像に対しコーナー特微量を検出する。この際 FAST アルゴリズム (AL002) を用いる。あるピクセルを p としそのピクセルの値を I_p とした時、円形の周囲 16 ピクセルを調査する。もし調査したピクセルの値が 10 個連続して I_p より低いまたは連続して値が高い場合は画素 p をコーナーとして検出する。
2. 前回画像フレームで得た特微量と新しく得られた画像フレーム内の特微量の対応関係を求める。具体的には前回得た特微量の周囲数ピクセルの変化を勾配として捉え、各特徴点周辺の移動量を算出し、画像全体としての画像内での平行移行移動、回転を決定する。また、特微量の新しい位置を決定する。
3. マッチングで特定した特徴点のペアを用い、カメラの相対的な姿勢や移動量（平行移動・回転量）を求める。この時 5 点以上の対応点を用い基本行列(Essential matrix)を最適化により求める。ただしこの際実際の距離に関しては不確定となる。
4. 積算した IMU の加速度と角速度情報を積算することにより、画像フレーム間での実際の移動量を算出。特微量追跡により得られた平行移動量・回転量に対してスケールの調整を行い正確な移動量を出す。

⁹ 出典：<https://aaltodoc.aalto.fi/server/api/core/bitstreams/a0655bd9-d0bd-45ec-a57bf709f43fe0d8/content>

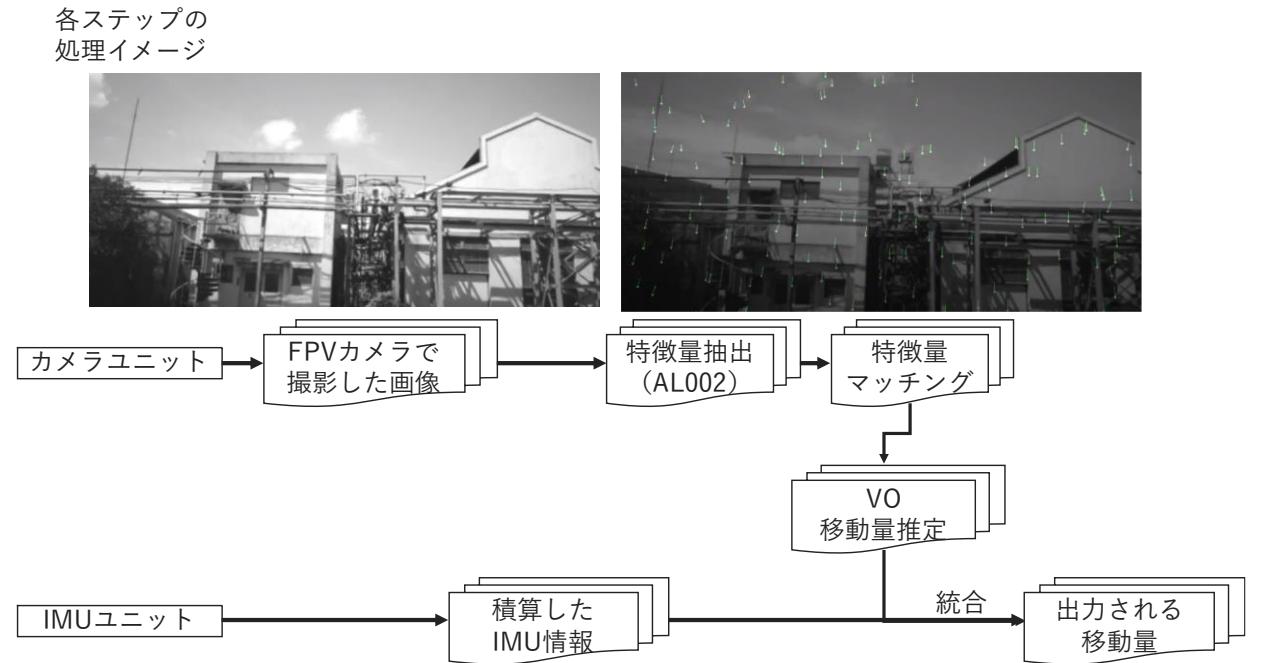


図 3-38 アルゴリズム詳細

2) 【AL102】測位情報の統合アルゴリズム

- 本アルゴリズムを利用した機能
 - 【FN011】測位情報統合

● アルゴリズムの詳細

測位情報・センサ情報の統合・取捨選択を行うアルゴリズムを開発した。

具体的には GPS、LiDAR 自己位置推定からの位置情報、IMU からの加速度・角速度、VIO からの移動速度など複数の異なる観測情報を統合し、各センサ単体よりもロバストな自己位置推定を実現する。また、統合される直前で測位情報に対し信頼できるかどうかの検査が行われ信頼できる場合にのみ統合に用いる技術も用いられておりシステムの堅牢性向上に寄与している。

1. 運動モデル定義： 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を定義する。ドローンの運動モデルは 3 次元位置と 3 次元姿勢を考慮するため、6 自由度。このときの状態ベクトルは以下
 $[X, Y, Z, roll, pitch, yaw, X^{'}, Y^{'}, Z^{'}, roll^{'}, pitch^{'}, yaw^{'}, X^{''}, Y^{''}, Z^{''}]$
2. 観測モデル定義： 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）を予測する方程式を決める。このときの観測ベクトルは以下
 $[x_{linear k}, y_{linear k}, z_{linear k}, r_{linear k}, p_{linear k}, y_{linear k}, x_{pos k}, y_{pos k}, z_{pos k}, \omega_{roll k}, \omega_{pitch k}, \omega_{yaw k}, x_k, a_{y k}, a_{z k}]$
3. 予測ステップ： 運動モデルに基づいて、現在の状態ベクトルから次の時刻の状態ベクトルを予測。過去の観測データから現在、又は未来の自己位置などを予測し利用する。更新式は以下

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

x_t は時刻 t における状態ベクトル

u_t は時刻 t における制御入力。ただし $u_t=0$

4. 更新ステップ：観測データを用いて、状態変数及び共分散行列の予測値修正を更新ステップで行う。

この際各センサや測位情報が入力されるが、使用される測位情報の信頼度が低い場合は信頼度の低い情報は無視される。状態ベクトルの更新は以下のように行われる。

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

z_t は時刻 t の実際の観測値

z_t は時刻 t の実際の観測値

H_t は時刻 t での観測モデルのヤコビ行列

5. 3と4を繰り返す

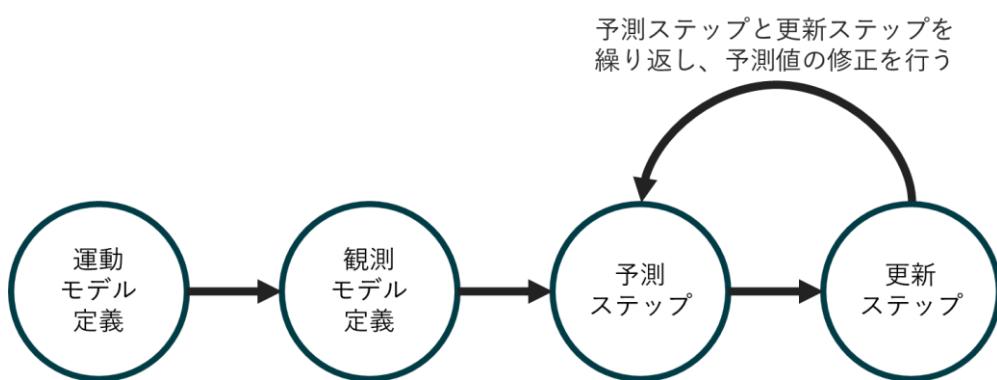


図 3-39 アルゴリズム概要

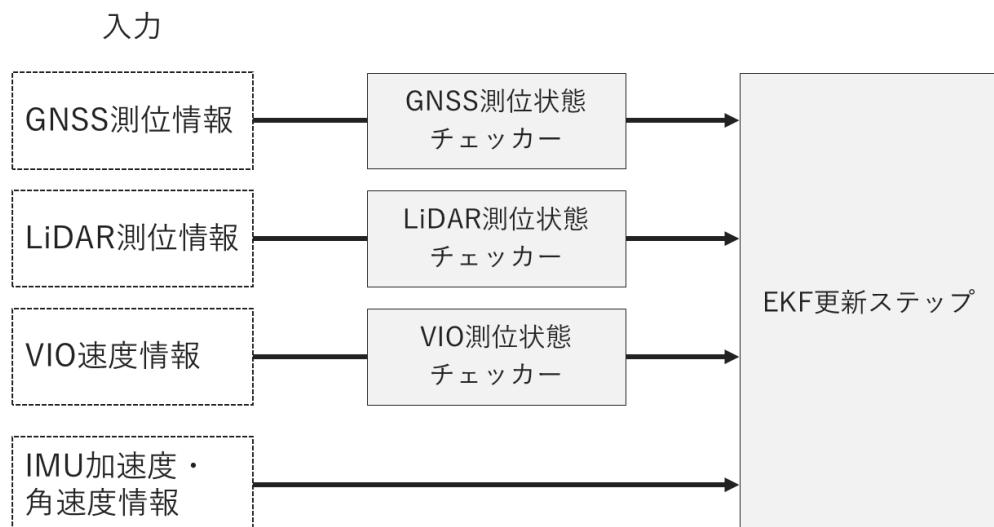


図 3-40 EKF 更新ステップ入力概要

- 3) 【AL103】日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム

- イメージ

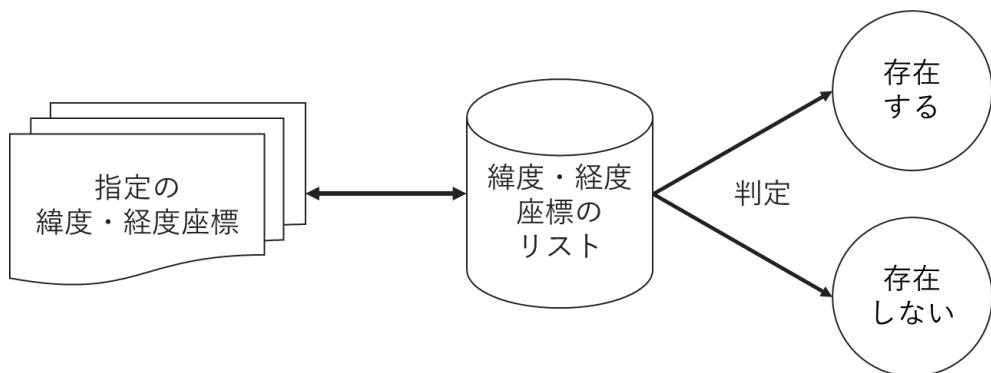


図 3-41 日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズムによる判定のイメージ

- 本アルゴリズムを利用した機能

➢ 【FN015】地形データ切り替え

- アルゴリズムの詳細

指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム

- 4) 【AL104】PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム

- イメージ

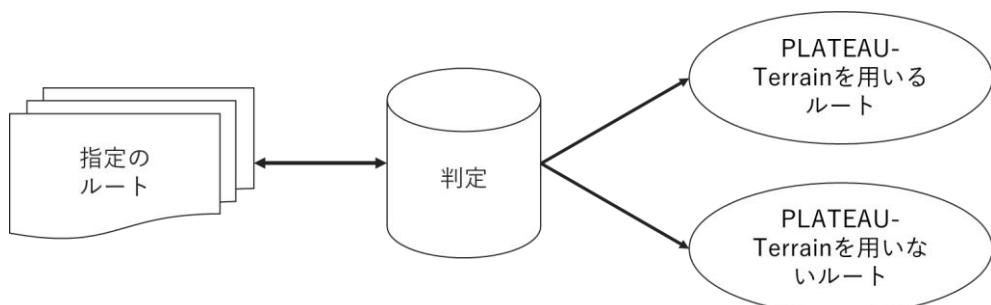


図 3-42 PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズムによる判定のイメージ

- 本アルゴリズムを利用した機能

➢ 【FN015】地形データ切り替え

- アルゴリズムの詳細

SENSYN CORE Pilot で作成した飛行ルートについて、PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム

3-4. データインターフェース

3-4-1. ファイル入力インターフェース

1) 【IF001】BIMモデル IFCファイル入力

- 本インターフェースを利用した機能：【FN001】

表 3-5 BIMモデル IFCファイル入力

入力ファイル名称	形式	詳細
堺市役所立体駐車場.ifc	IFC 形式	堺市役所立体駐車場の IFC データ

2) 【IF002】3D都市モデル CityGMLファイル入力

- 本インターフェースを利用する機能：【FN002】

表 3-6 3D都市モデル CityGMLファイル入力

入力ファイル名称	形式	詳細
51356388_bldg_6697_op.gml	CityGML 形式	堺市役所・堺地方合同庁舎周辺の建築物 LOD2 データと 堺市役所立体駐車場の LOD4 データを マージした 3D 都市モデル

3-4-2. ファイル出力インターフェース

1) 【IF101】BIMモデル変換インターフェース CityGMLファイル出力

- 本インターフェースを利用した機能

- 【FN001】

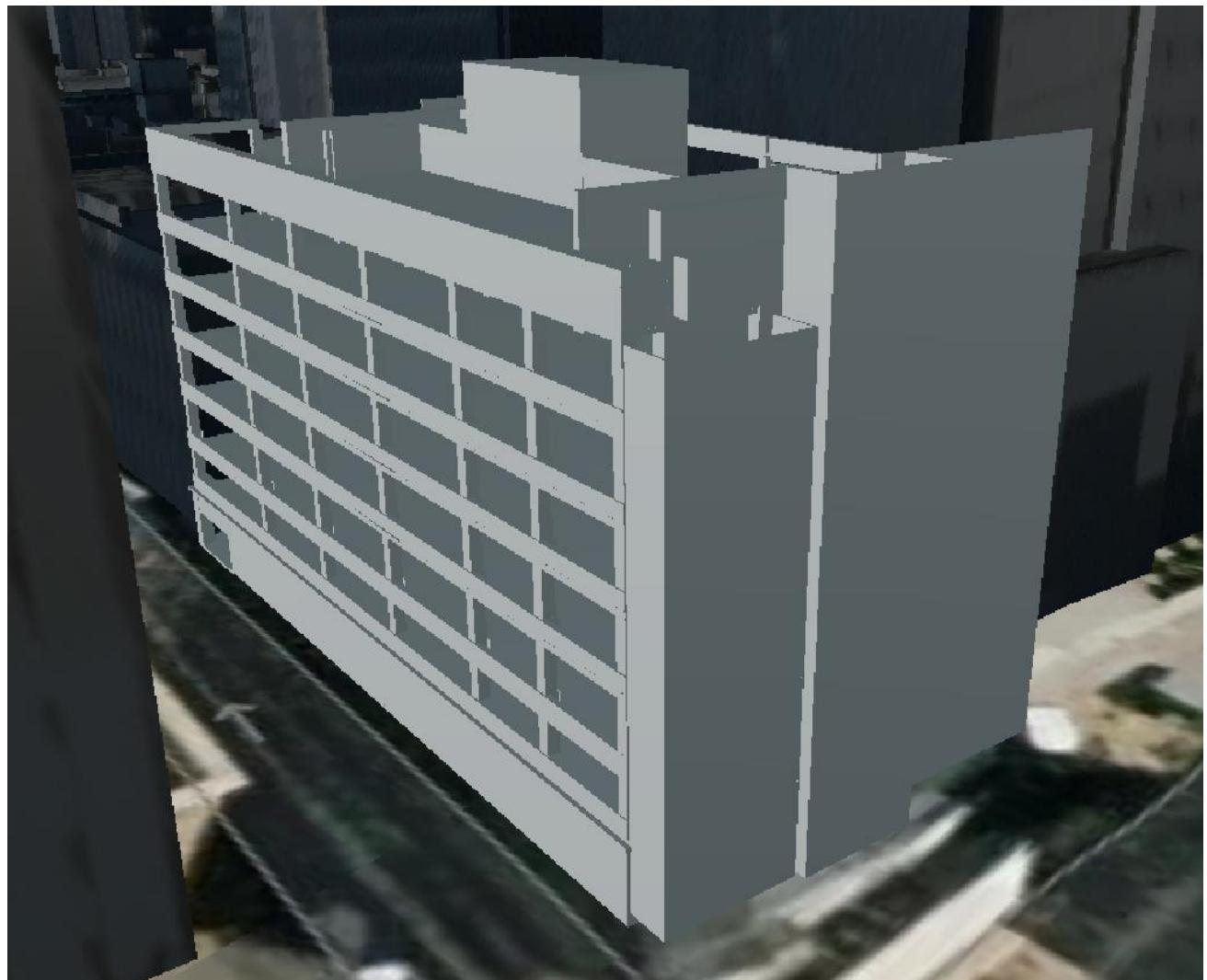


図 3-43 BIM モデル変換インターフェース CityGML ファイル出力で出力した CityGML イメージ図

表 3-7 BIM モデル変換インターフェース CityGML ファイル出力

出力ファイル名称	形式	詳細
堺市役所立体駐車場.gml	CityGML 形式	堺市役所立体駐車場の IFC データを CityGML 形式に変更したもの

2) 【IF102】3D 都市モデルの点群変換 PCD ファイル出力

- 本インターフェースを利用した機能
 - 【FN002】【FN003】【FN004】【FN005】

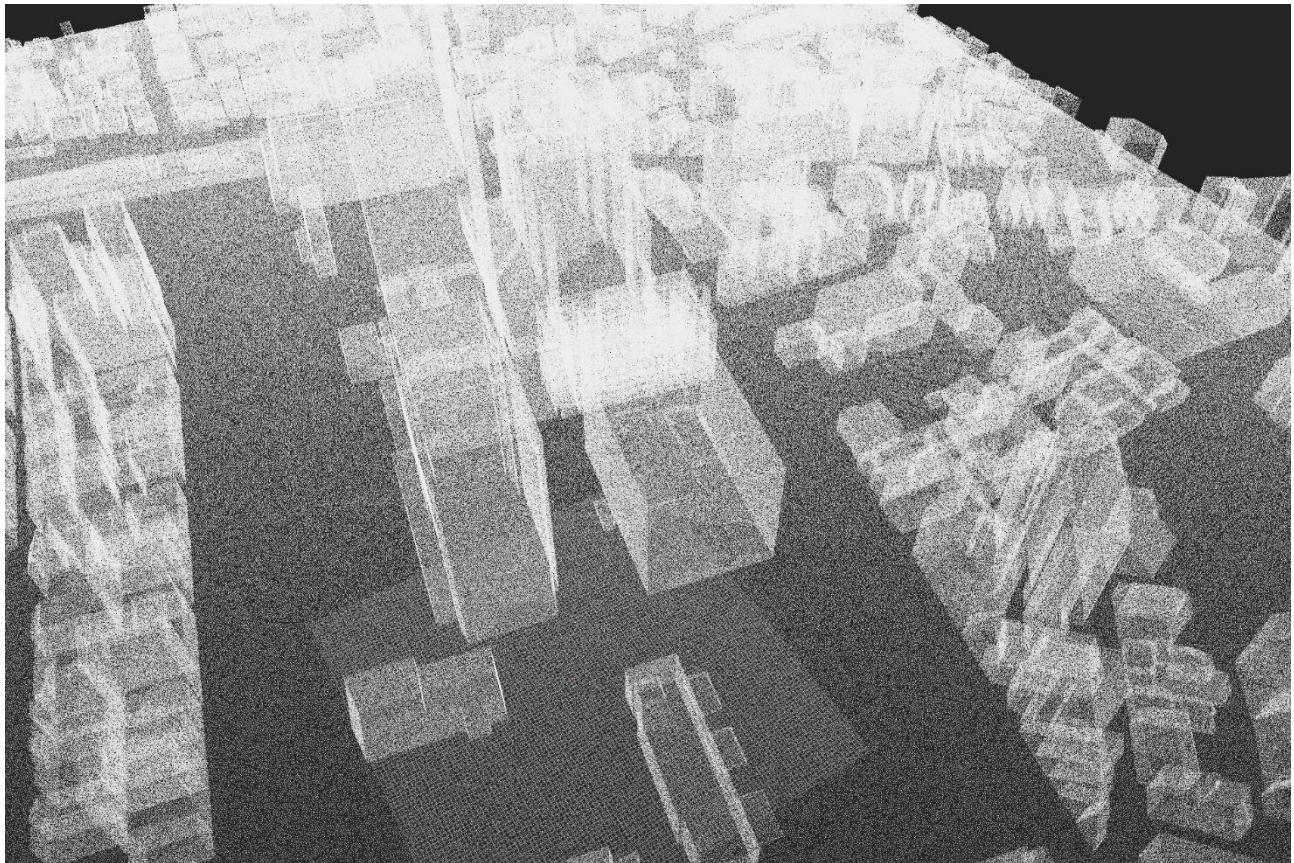


図 3-44 3D 都市モデルの点群変換 PCD ファイル出力で出力した PCD ファイルイメージ図

表 3-8 3D 都市モデルの点群変換 PCD ファイル出力

出力ファイル名称	形式	詳細
elas_map.pcd	PCD 形式	大阪府堺市全域の建築物 LOD2 データから抜き出した、堺市役所・堺地方合同庁舎周辺が含まれる 1 メッシュ分の建築物 LOD2 データと堺市役所立体駐車場の建築物 LOD4 データをマージした 3D 都市モデルを、PCD 形式に変更したもの。 PCD 形式に変換する際、各建物についてそれぞれ一番 LOD のレベルが高いデータを点群化する。これにより、建築物 LOD4 が唯一存在する堺市役所立体駐車場は LOD4 をベースに、他の建築物は LOD2 をベースとした点群データ (pcd ファイル) が出力される。

3-4-3. 内部連携インターフェース

データの入出力はROSメッセージ等により直接行う。

1) 【IF201】LiDARスキャンデータの連携

- インタフェースの詳細
 - LiDARユニットで取得したスキャンデータをELAS側で受け取る
- 本インターフェースを利用した機能
 - 【FN005】 【FN006】 【FN007】 【FN011】 【FN012】
- プロトコル
 - ROSメッセージ
- メソッド
 - GET
- パス
 - /fusion/lidar
- リクエストパラメータ

表 3-94 リクエストパラメータ

パラメータ名	説明	値	必須
token	トークン	-	-

- レスポンス

表 3-10 レスポンス

ステータスコード	説明
200	処理に成功
400	パラメータ不正

- レスポンスボディ

- AnswerForm

表 3-11 レスポンスボディ (AnswerForm)

フィールド	説明	値
lidar_scan	LiDARスキャンデータ	-

表 3-12 LiDARスキャンデータの連携

名称	形式	タイミング・頻度
LiDARスキャンデータ	Point Cloud 形式	20 Hz (1秒間に20回取得)

2) 【IF202】GNSS測位座標の連携

- インタフェースの詳細
 - GNSSユニットで取得した測位情報をELAS側で受け取る
- 本インターフェースを利用した機能
 - 【FN008】【FN009】【FN011】
- プロトコル
 - ROSメッセージ
- メソッド
 - GET
- パス
 - /fusion/gnss
- リクエストパラメータ

表 3-135 リクエストパラメータ

パラメータ名	説明	値	必須
token	トークン	-	-

- レスポンス

表 3-14 レスポンス

ステータスコード	説明
200	処理に成功
400	パラメータ不正

- レスポンスボディ

- AnswerForm

表 3-15 レスポンスボディ (AnswerForm)

フィールド	説明	値
gnss	GNSS測位座標データ	-

表 3-16 GNSS測位座標の連携

名称	形式	タイミング・頻度
GNSS測位座標	NMEA形式	5 Hz (1秒間に5回取得)

3) 【IF203】カメラ画像データの連携

- インタフェースの詳細
 - カメラユニットで取得した画像をELAS側で受け取る
- 本インターフェースを利用した機能
 - 【FN010】
- プロトコル
 - ROSメッセージ
- メソッド
 - GET
- パス
 - /fusion/camera
- リクエストパラメータ

表 3-176 リクエストパラメータ

パラメータ名	説明	値	必須
token	トークン	-	-

- レスポンス

表 3-18 レスポンス

ステータスコード	説明
200	処理に成功
400	パラメータ不正

- レスポンスボディ

- AnswerForm

表 3-19 レスポンスボディ (AnswerForm)

フィールド	説明	値
fpv_img	カメラ映像データ	-

表 3-20 カメラ画像データの連携

名称	形式	タイミング・頻度
カメラ画像データ	Image 形式	30 Hz (1秒間に30回取得)

4) 【IF204】IMU（機体）加速度/角速度の連携

- インタフェースの詳細
 - IMU ユニットで取得した加速度/角速度を ELAS 側で受け取る
- 本インターフェースを利用した機能
 - 【FN010】 【FN011】 【FN013】
- プロトコル
 - ROS メッセージ
- メソッド
 - GET
- パス
 - / fusion/ imu
- リクエストパラメータ

表 3-217 リクエストパラメータ

パラメータ名	説明	値	必須
token	トークン	-	-

● レスポンス

表 3-22 レスポンス

ステータスコード	説明
200	処理に成功
400	パラメータ不正

● レスポンスボディ

- AnswerForm

表 3-23 レスポンスボディ (AnswerForm)

フィールド	説明	値
acceleration	加速度	-
angular_velocity	角速度	-

表 3-24 IMU（機体）加速度/角速度の連携

名称	形式	タイミング・頻度
IMU（機体）加速度/角速度	ROS メッセージ形式	200 Hz (1 秒間に 200 回取得)

5) 【IF205】 ELAS、SENSYN CORE Pilot間連携インターフェース

- 本インターフェースを利用した機能
 - 【FN004】 【FN012】 【FN013】 【FN014】 【FN016】 【FN018】 【FN019】 【FN020】
- プロトコル
 - ROS メッセージ

表 3-25 ELAS、SENSYN CORE Pilot間連携インターフェース

#	インターフェース名	内容
1	目標経路（飛行ルート）の送信	<ul style="list-style-type: none"> ● 飛行ルートを送信する <ul style="list-style-type: none"> ➢ 実行する飛行経路のウェイポイント (緯度・経度・高度) ➢ 各ウェイポイントにおけるドローンの速度 ➢ 飛行ルートの送信結果（送信成功/失敗）表示
2	飛行開始指示	<ul style="list-style-type: none"> ● 機体に送信された飛行ルートでのミッションを開始し、離陸する指示を送信する
3	緊急停止指示	<ul style="list-style-type: none"> ● 進行中のミッションを一時停止し、その場でホバリングする指示を送信する
4	飛行再開指示	<ul style="list-style-type: none"> ● 一時停止した、進行中のミッションを再開する
5	テレメトリー情報の表示	<ul style="list-style-type: none"> ● 機体の各種テレメトリー情報を SENSYN CORE Pilot に送信する <ul style="list-style-type: none"> ➢ 機体名、カメラ名 ➢ 機体の接続状況 ➢ 機体の位置（緯度・経度・高度） ➢ 機体の進行速度（m/s） ➢ 機体の自己位置推定ステータス（GNSS 測位状態、LiDAR 測位状態） ➢ 機体のバッテリー残量 ➢ 飛行モード ➢ 飛行予定時間

6) 【IF206】自己位置推定情報/ドローン移動量推定情報の連携

- 本インターフェースを利用した機能

➢ 【FN007】【FN010】【FN011】

表 3-26 自己位置測位/推定情報の連携

#	インターフェース名	内容
1	LiDAR 自己位置推定情報の連携	<ul style="list-style-type: none"> ● LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を【FN007】LiDAR 測位情報座標変換機能で基準座標系に直したデータを【FN011】測位情報統合に連携する
2	ドローン移動量推定情報の連携	<ul style="list-style-type: none"> ● 【FN010】ビジュアルイナーシャルオドメトリ (VIO) より得られるドローン移動量推定値 (xyz 速度、角速度推定値) を【FN011】測位情報統合に連携する

3-4-4. 外部連携インターフェース

1) 外部連携インターフェース

① 【IF301】Cesium、SENSYN CORE Pilot サーバー間

- インタフェースの概要
 - Cesium と SENSYN CORE Pilot 間で通信を行う
- 本インターフェースを利用した機能
 - 【FN015】【FN017】
- プロトコル
 - HTTPS

表 3-27 Cesium、SENSYN CORE Pilot サーバー間インターフェース

インターフェース名	内容
地図データリクエスト	Cesium から地図データを読み込む
3D オブジェクトデータリクエスト	Cesium から 3D オブジェクトのデータを読み込む

② 【IF302】PLATEAU 配信サービス、SENSYN CORE Pilot サーバー間

- インタフェースの概要
 - PLATEAU 配信サービスと SENSYN CORE Pilot 間で通信を行う
- 本インターフェースを利用した機能
 - 【FN015】【FN017】【FN019】

- プロトコル
 - HTTPS
- その他
 - PLATEAU配信サービスについて、詳しくはGitHubドキュメントを参照¹⁰

表 3-28 PLATEAU配信サービス、SENSYN CORE Pilotサーバー間インターフェース

インターフェース名	内容
地形データリクエスト	PLATEAU配信サービスからPLATEAU-Terrainデータを読み込む
建築物LOD2データのリクエスト	PLATEAU配信サービスから建築物LOD2データを読み込む

(3) 【IF303】場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース

- インタフェースの概要
 - Cesium Geocode API
 - 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース
- 本インターフェースを利用した機能
 - 【FN015】
- その他
 - 詳しくはCesium公式ドキュメントを参照¹¹
- プロトコル
 - HTTPS

表 3-29 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース

Name	Type
container	Element string
Scene	Scene
geocoderServices	Array
autoComplete	boolean
flightDuration	number
destinationFound	Geocoder.DestinationFoundFunction

(4) 【IF304】ELAS、DJIドローン間

¹⁰<https://github.com/Project-PLATEAU/plateau-streaming-tutorial>¹¹<https://cesium.com/learn/cesiumjs/ref-doc/Geocoder.html>

- インタフェースの概要
 - ELAS と DJI ドローン間で通信を行う
- 本インターフェースを利用した機能
 - 【FN004】
- プロトコル
 - シリアル通信 (RS232)

表 3-30 ELAS、DJI ドローン間インターフェース

インターフェース名	内容
RTK-GNSS 測位情報リクエスト	RTK-GNSS によるドローン自己位置測位情報をリクエスト
FPV 撮影画像リクエスト	ドローン内臓の前方カメラ (FPV) で撮影した画像をリクエスト
ドローン制御リクエスト	ドローンの制御をリクエスト

3-5. 実証に用いたデータ

3-5-1. 活用したデータ一覧

1) 利用した3D都市モデル

- 年度：2022年度
- 都市名：堺市
- ファイル名：27140_sakai-shi_2022_citygml_1_op
- メッシュ番号：513563-51356388（インデックスマップで黄色いの箇所）

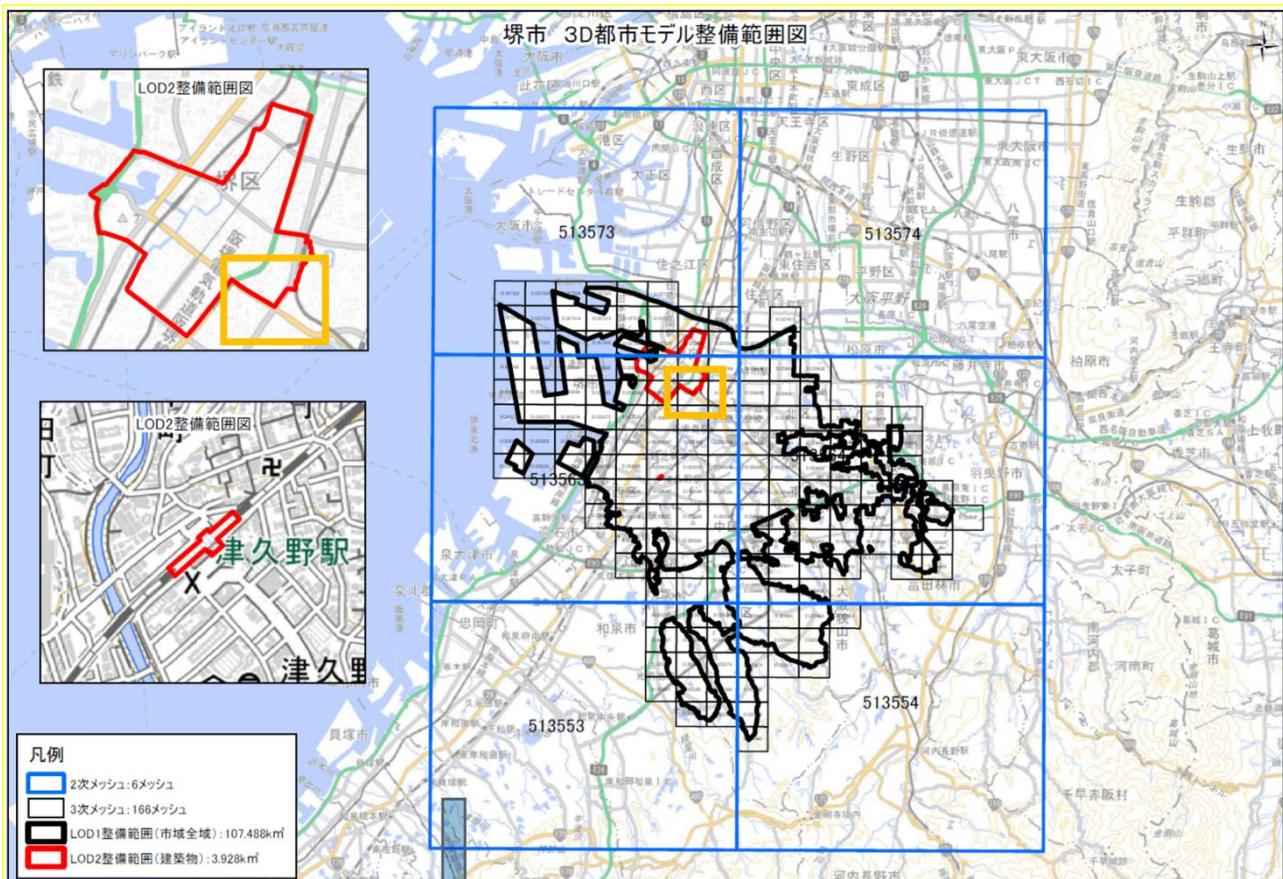


図 3-42 インデックスマップ（堺市）

表 3-31 利用した3D都市モデル

地物	地物型	属性区分	ID	属性名	内容	データを利用した機能(ID)
建築物 LOD2	bldg:Building	空間属性	DT001	bldg:lod2Solid	建築物のLOD2の立体	FN002
建築物 LOD4	bldg:Building	空間属性	DT002	bldg:lod4Solid	建築物のLOD4の立体	FN002

2) 利用したその他のデータ

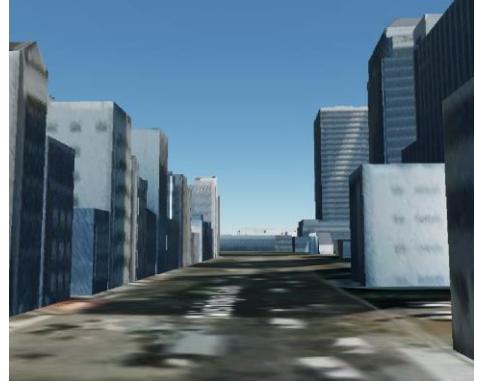
1. データ一覧

表 3-32 利用したその他データ(一覧)

ID	エリア (都市)	活用データ	内容	データ形式	出所	データを利用した機能 (ID)
DT101	堺市	BIM(IFC)	堺市役所立体駐車場の設計図面を基に、本ucのため新規作成したBIMモデル	IFC	堺市 株式会社竹中工務店	FN001
DT102	堺市	PLATEAU-Terrain	地形モデルのTerrain dbデータ	Terrain db	PLATEAU 配信サービス FN015 FN016 FN017	FN002 FN015 FN016 FN017
DT103	堺市	PLATEAU-3DTiles	建築物モデル等の3D Tilesデータ	3D Tiles	PLATEAU 配信サービス	FN016

2. データサンプル（イメージ）

表 3-33 利用したその他データ（サンプル）

ID	活用データ	サンプル・イメージ
DT101	BIM (IFC)	
DT102	PLATEAU-Terrain	
DT103	PLATEAU-3DTiles	

3-5-2. 生成・変換したデータ

表 3-34 生成・変換したデータ

ID	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)	データを利用した機能 (ID)
DT201	3D都市モデル (CityGML 形式)	ドローン用マップ作成のため 3D都市モデルビューワでの表示のため	<ul style="list-style-type: none"> BIM (IFC ファイル) を「3D都市モデル整備のための BIM 活用マニュアル」に従って建築物 LOD4 に変換 「3D都市モデル整備のための BIM 活用マニュアル」に従って建築物 LOD4 を建築物 LOD2 (埠市該当地域) にマージして、ドローン用マップに利用 	FME Desktop (https://github.com/Project-PLATEAU/IFCtoCityGML?tab=readme-page)	BIM モデル (IFC 形式)	FN001 FN002 FN005 FN016
DT202	LiDAR スキャンデータ (ROS メッセージ形式)	自己位置推定のため	<ul style="list-style-type: none"> 3D LiDAR センサで計測 【FN005】 LiDAR ローカリゼーション機能で自己位置推定処理を行う 	【FN005】 LiDAR ローカリゼーション	Point Cloud	FN005 FN011
DT203	GNSS 測位座標 (NMEA 形式)	自己位置測位のため	<ul style="list-style-type: none"> GNSS を利用して、ドローンの位置座標を取得 【FN008】 GNSS 測位情報チェック機能、 【FN009】 GNSS 測位情報 	【FN008】 GNSS 測位情報 チェック機能 【FN009】 GNSS 測位情報	GNSS 測位 座標 (NMEA 形式)	FN008 FN009 FN011

			測位情報座標変換機能でそれぞれ自己位置推定に必要な処理を行う	報座標変換機能		
DT204	IMU（機体） 加速度/角速度 (ROS メッセージ形式)	自己位置推定のため	<ul style="list-style-type: none"> ● ジャイロと加速度計によって3次元の加速度/角速度を計測 ● 【FN011】測位情報統合機能で、自己位置推定のために利用 	【FN011】 測位情報統合機能	加速度/角速度 (ROS メッセージ形式)	FN011
DT205	カメラ画像データ (ROS メッセージ形式)	自己位置推定のため	<ul style="list-style-type: none"> ● ドローン内臓カメラによって画像を記録 ● 【FN010】ビジュアルインナーシャルオドメトリ(VIO)機能で、自己位置推定のために利用 	【FN010】 ビジュアルインナーシャルオドメトリ(VIO)機能	画像データ (Image 形式)	FN010 FN011

3-6. ユーザーインターフェース

3-6-1. 画面一覧

1) PC 画面

表 3-35 PC 画面一覧

ID	連携 (ID)	画面名	画面説明	画面を表示した機能 (ID)
SC001	-	フライトプラン	<ul style="list-style-type: none"> 自動飛行を実行するエリアの名称を規定し、新規ルートの作成やインポートした CityGML から生成したオブジェクトモデルをひも付けための登録画面 	FN016
SC002	-	新規ルートの作成	<ul style="list-style-type: none"> フライトプラン登録後呼び出したエリアにおいて、新規の飛行ルートを作成する際に必要な情報・飛行方法を決定する画面 	FN016
SC003	SC004,005 ,006,007	メイン画面 (飛行前)	<ul style="list-style-type: none"> モデルの地図表示、機体の選択、自動飛行ルートの作成と保存までを一元表示 	FN016
SC004	-	機体表示	<ul style="list-style-type: none"> リスト形式で、今回利用する機体及びカメラを選択 	FN016
SC005	006	アクション表示	<ul style="list-style-type: none"> 機体のアクションを規定、ウェイポイント、高度、速度の設定を行う 高度、速度はスライダーバー又は、数値を直接入力 飛行ルートを保存する 	FN016
SC006	005	地図画面	<ul style="list-style-type: none"> 3D 形式で任意の方向、アングルの変更、縮尺の変更が可能 ウェイポイントの指定時は、マウスのダブルクリックで順次新規のウェイポイントを作成する 作成したルートがリアルタイムで反映される 	FN016 FN019
SC007	-	ルートアップロード	<ul style="list-style-type: none"> 作成したルートを保存し、飛行開始画面に遷移 	FN016
SC008	-	飛行開始画面	<ul style="list-style-type: none"> 機体の飛行開始時に利用 	FN016
SC009	010	メイン画面 (飛行中)	<ul style="list-style-type: none"> 飛行中の機体の位置を地図画面に表示 サブウインドウにテレメトリーとして、現在位置座標、高度、速度、LiDAR 測位状態、バ 	FN019 FN020

			バッテリー情報等の表示 ● 緊急時の停止、再開、帰還の選択	
SC010	009	機体テレメトリー	● 現在位置座標、高度、速度、GNSSRTK測位状態、LiDAR測位状態、バッテリー情報等を表示	FN019 FN020
SC011	-	緊急停止	● 飛行の一時停止実行	FN019 FN020

2) 管理者画面

表 3-36 管理者画面一覧

ID	連携 (ID)	画面名	画面説明	画面を表示した機能 (ID)
SC101	-	動作選択	● メイン画面の3点リーダーから、リスト選択で新規3Dオブジェクトの読み込みを選択	FN016
SC102	-	新規3Dオブジェクトの登録	● CityGMLやBIMから変換したオブジェクトをSENSYN COREへインポートする ● ID入力で自動的に反映	FN016
SC103	-	3Dオブジェクト表示	● 地図画面にインポートしたオブジェクトを表示	FN016

3-6-2. 画面遷移図

1) PC 画面

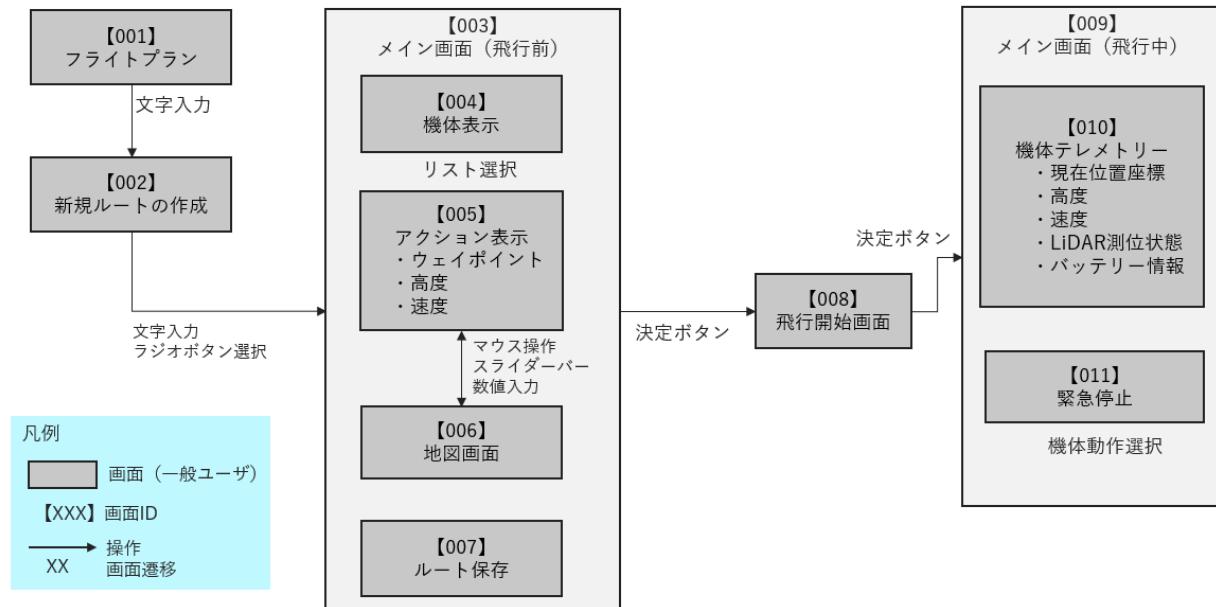


図 3-43 PC 画面遷移図

2) 管理者画面

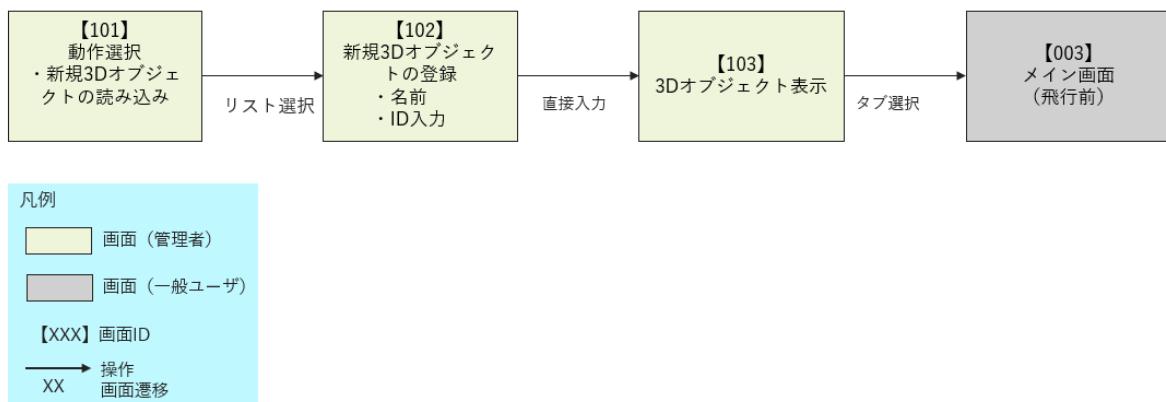


図 3-44 管理者画面遷移図

3-6-3. 各画面仕様詳細

1) PC 画面

1. 【SC001】 フライトプラン

● 画面の目的・概要

- 事前にインポートした3Dモデルがマージされているエリアを選択
- フライトプランを命名し、配下に新規ルートを複数登録することを可能にする

● 画面イメージ

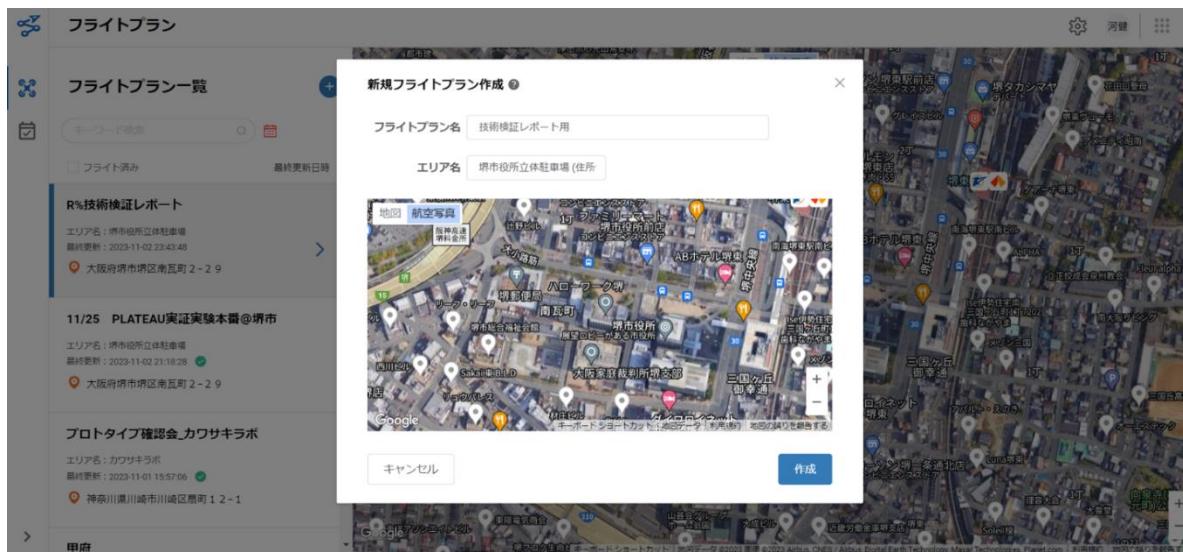


図 3-45 フライトプランの画面イメージ

2. 【SC002】新規ルートの作成

- 画面の目的・概要
 - フライトプランで設定された対象エリアにおけるルート作成の初期画面
 - ルート名を入力
 - 飛行方法の選択

※今回の検証機体はウェイポイント飛行のみ対応
- 画面イメージ



図 3-46 新規ルートの作成画面イメージ

3. 【SC003】メイン画面（飛行前）

- 画面の目的・概要
 - 地図画面と機体の選択、アクションの入力画面に大きく分けられ、ユーザーは、地図を見ながら飛行ルートを作成
 - 編集ボタンから、飛行ルートの修正やアクションの追加が可能
- 画面イメージ

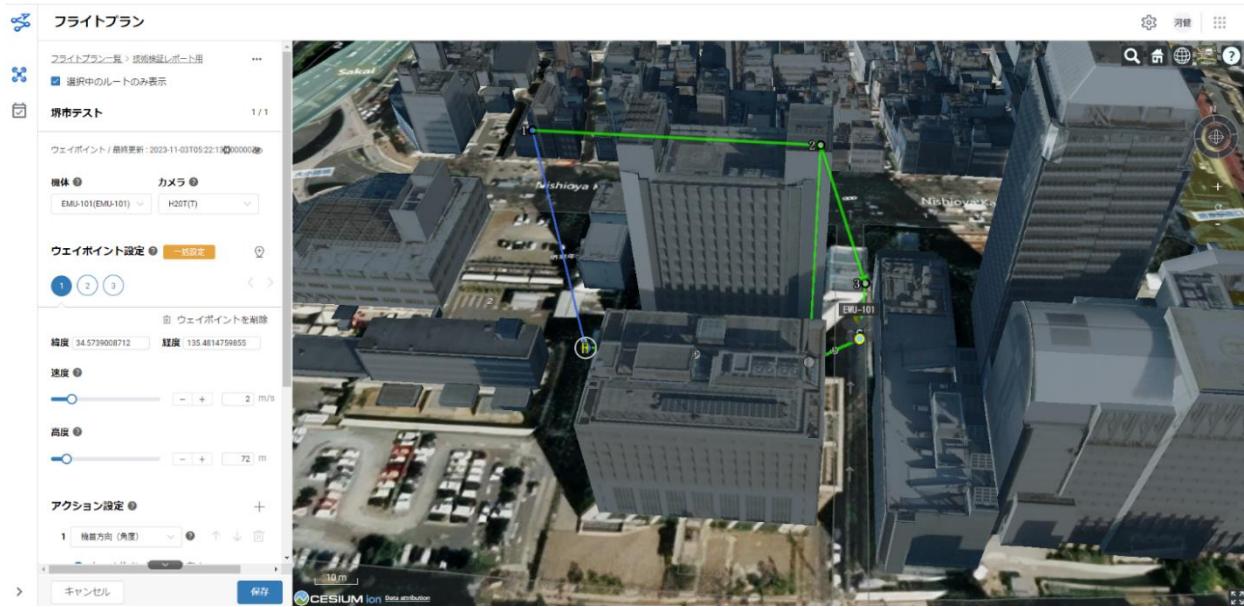


図 3-47 メイン画面（飛行前）のイメージ

4. 【SC004】機体表示

- 画面の目的・概要
 - 飛行させる機体を選択する ※機体の事前登録は管理者が実施
 - 搭載しているカメラを選択する
- 画面イメージ



図 3-48 機体選択画面イメージ

5. 【SC005】アクション表示

● 画面の目的・概要

- 緯度経度を直接入力して、ウェイポイントを設定
※地図画面でポインタ操作により任意の地点でダブルクリックすることデモンストレーション可能
- ドローンの飛行速度を設定
※スライダーバー又は数値の入力
- 高度の設定
※スライダーバー又は数値の入力
- ルートの保存

● 画面イメージ



図 3-49 アクション表示画面イメージ

6. 【SC006】地図画面

- 画面の目的・概要
 - 3D形式で任意の方向、アングルの変更、縮尺の変更が可能
 - ウェイポイントの指定時は、マウスのダブルクリックで順次新規のウェイポイントを作成する
 - 作成したルートがリアルタイムで反映される
- 画面イメージ

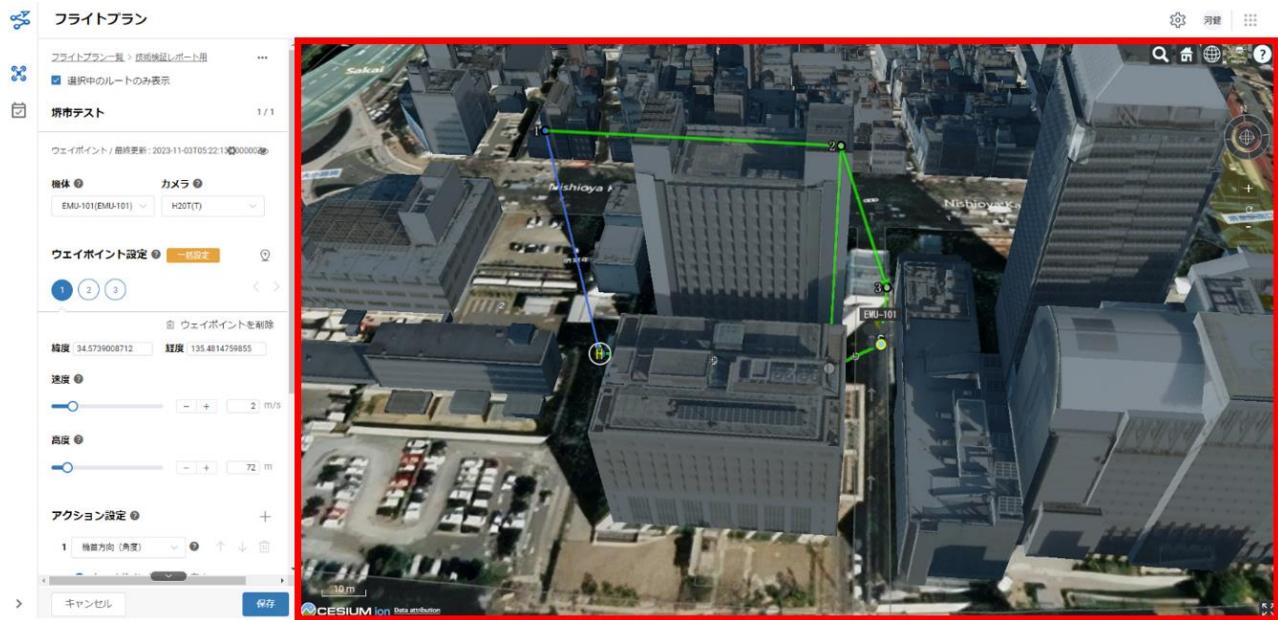


図 3-50 アクション表示画面イメージ

7. 【SC007】ルートアップロード

● 画面の目的・概要

- 飛行を選択すると、ポップアップ画面が表示
- ポップアップ画面には、機体から得たテレメトリー表示
※バッテリー残量/バッテリー電圧/GPS測位状態/飛行時間/距離
- ステータスを確認後、「フライト確認画面に進む」を選択
- 更にポップアップされ、フライト終了時の機体動作を選択
- 「ルートアップロードをロード」ボタンを選択すると機体にルートがアップロードされる

● 画面イメージ

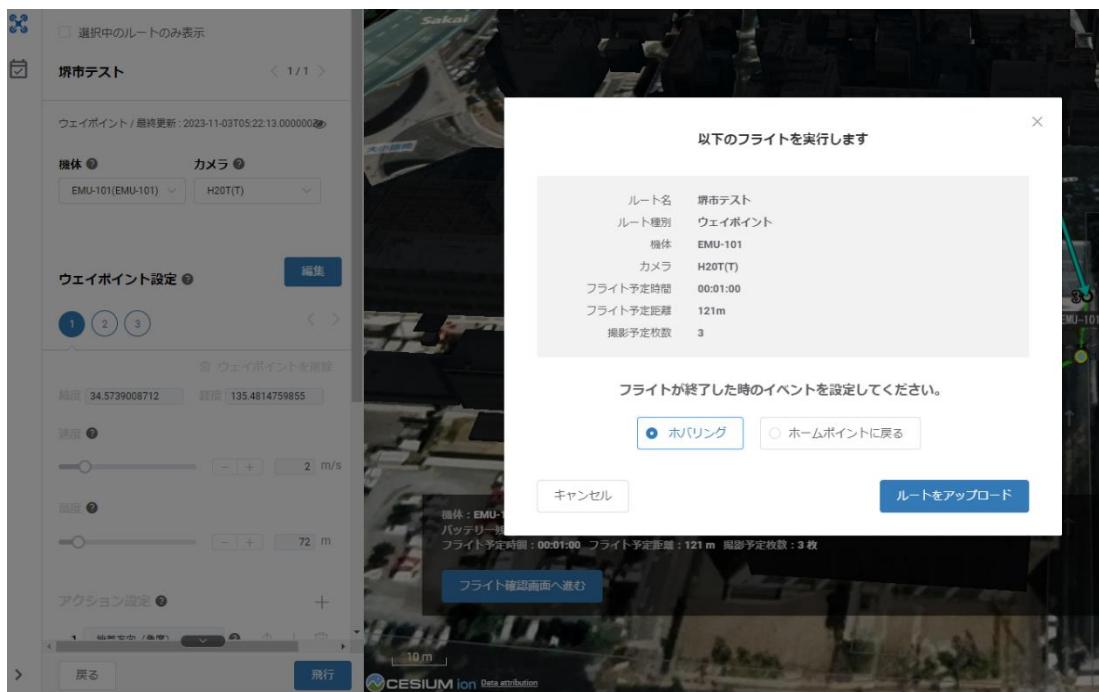


図 3-51 ルートアップロード画面イメージ

8. 【SC008】飛行開始画面

- 画面の目的・概要
 - ルートアップロード後にすぐに自動で飛び立たないために安全の観点で設ける
 - ポップアップ画面には、離陸前の注意事項を改めて提起
 - 周辺状況と機体の最終チェック後、飛行開始ボタンを選択すると、機体が自動で離陸
- 画面イメージ

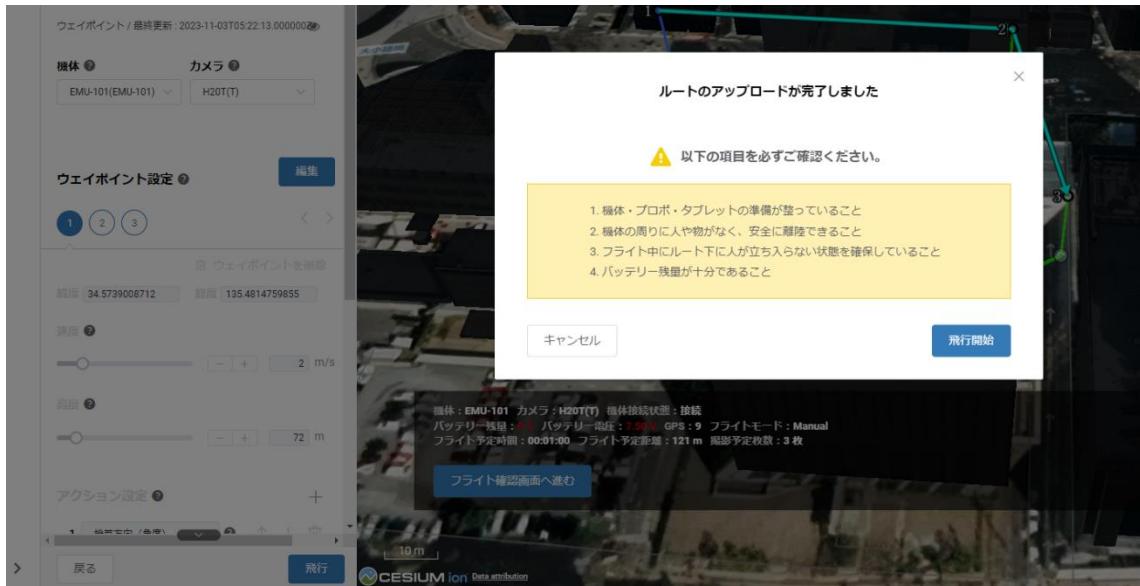


図 3-52 新規ルートの作成画面イメージ

9. 【SC009】メイン画面（飛行中）

- 画面の目的・概要
 - 自動航行を行っている際の監視画面、機体の現在位置を3次元で地図画面に表示
 - サブウインドウにテレメトリーとして、現在位置座標/高度/速度/LiDAR測位状態/バッテリー情報等の表示
 - 緊急時の停止、再開、帰還の選択
- 画面イメージ

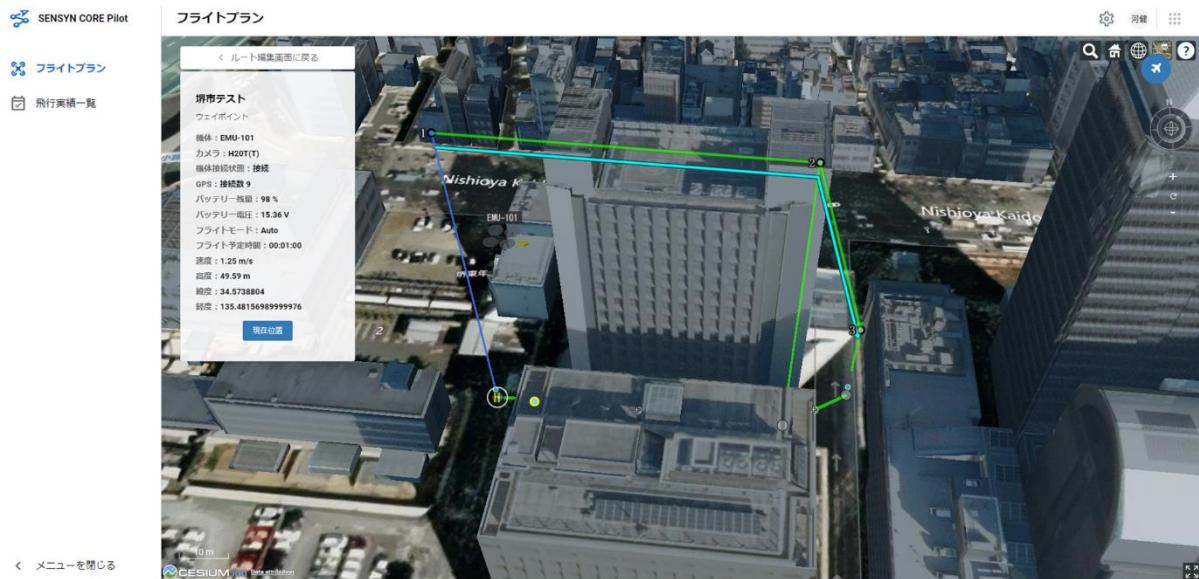


図 3-53 メイン画面（飛行中）イメージ

10. 【SC010】機体テレメトリー画面

- 画面の目的・概要
 - 緯度経度座標、高度、速度、GNSSRTK測位状態、LiDAR測位状態、バッテリー情報等を表示
- 画面イメージ

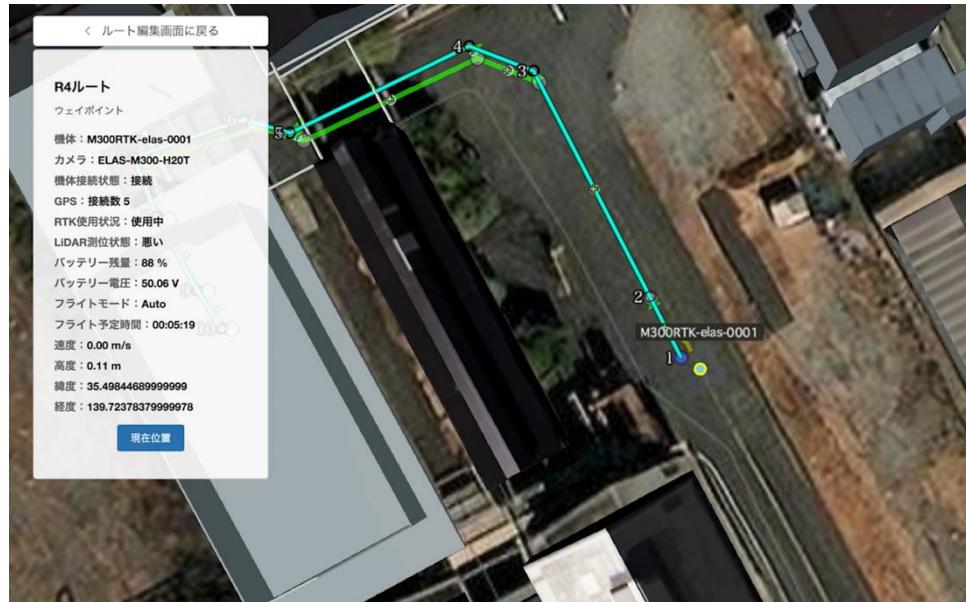


図 3-54 機体テレメトリーの画面イメージ

11. 【SC011】緊急停止画面

- 画面の目的・概要
 - 自動航行中に何らかの影響で機体を緊急停止させる際に利用
 - 機体の一時停止のほか、飛行再開と離陸地点への自動帰還の選択を具備
 - 飛行中メイン画面の右上の航空機マークを選択するとポップアップ
- 画面イメージ



図 3-55 緊急停止の画面イメージ

2) 管理者画面

1. 【SC101】動作選択

- 画面の目的・概要
 - メイン画面の 3 点リーダーから、リスト選択で新規 3D オブジェクトの読み込みを選択
- 画面イメージ



図 3-56 動作選択の画面イメージ

2. 【SC102】新規3Dオブジェクトの登録

- 画面の目的・概要
 - CityGMLやBIMから変換したオブジェクトをSENSYN COREへインポートする
 - ID入力で自動的に反映
- 画面イメージ

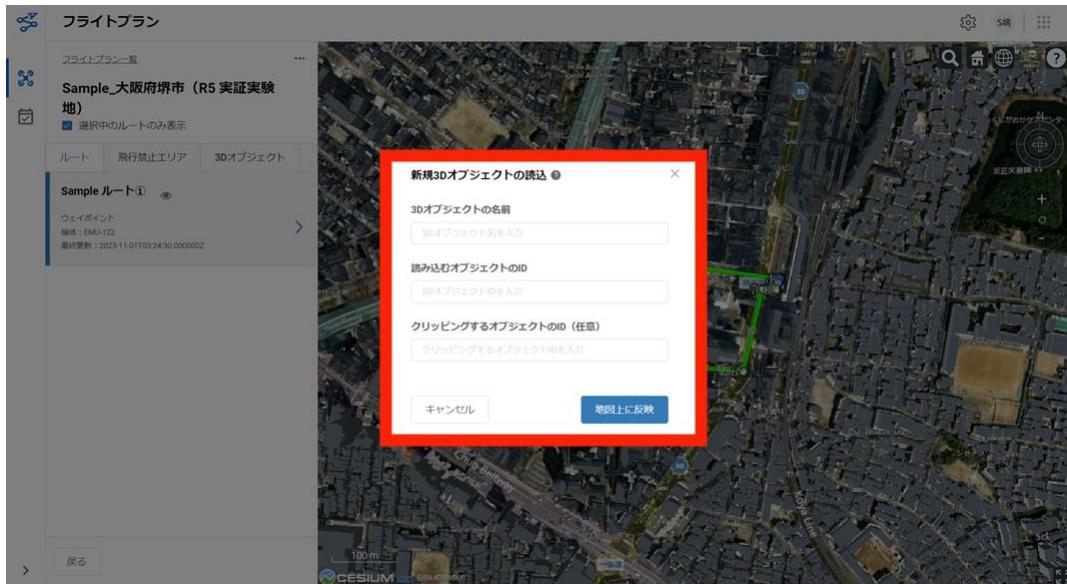


図 3-57 新規3Dオブジェクト読み込みの画面イメージ

3. 【SC103】3Dオブジェクトの表示

- 画面の目的・概要
 - 地図画面にインポートしたオブジェクトを表示
- 画面イメージ

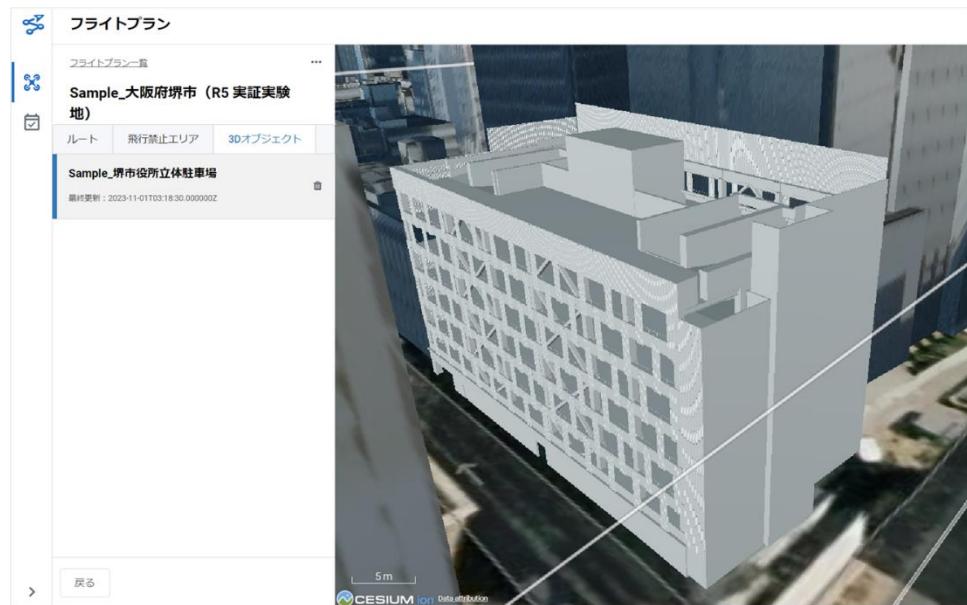


図 3-58 3D オブジェクトの表示画面イメージ

3-7. 実証システムの利用手順

3-7-1. 実証システムの利用フロー

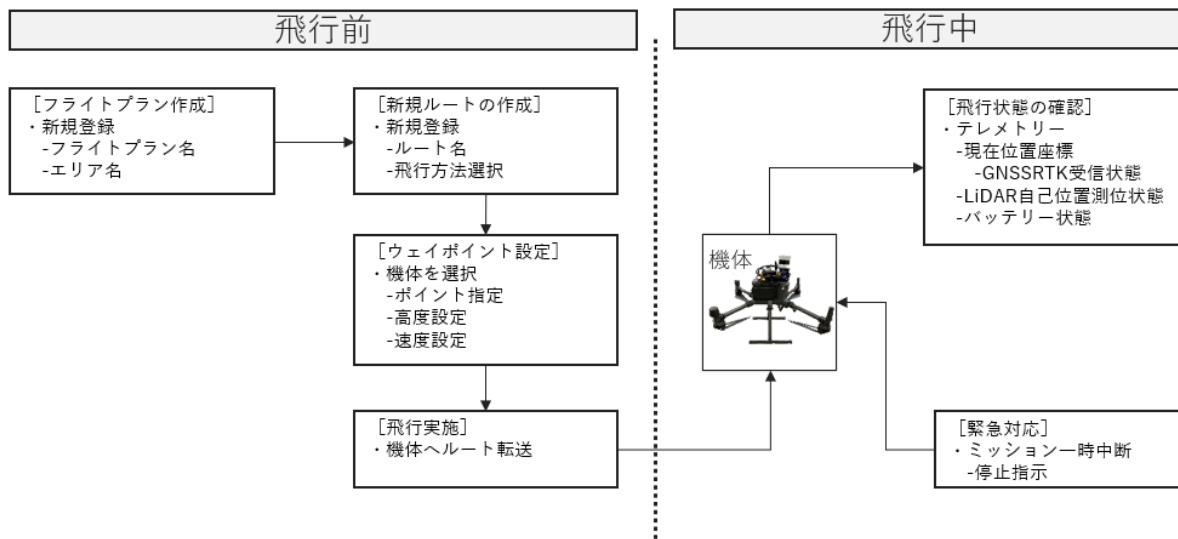


図 3-59 システムの利用フロー

- 登録したモデルに命名し、フライトプランを作成、新規ルート作成時はルート命名して機体を選択、各ウェイポイントを指定後、高度と速度を決定し機体ヘルートを転送
- 飛行中は、機体のテレメトリー情報をリアルタイムで取得
- 万一の場合は、ミッション一時中断して緊急停止

3-7-2. 各画面操作方法

1) フライトプランの作成

- フライトプラン名、エリア名（住所又は緯度経度座標）を登録

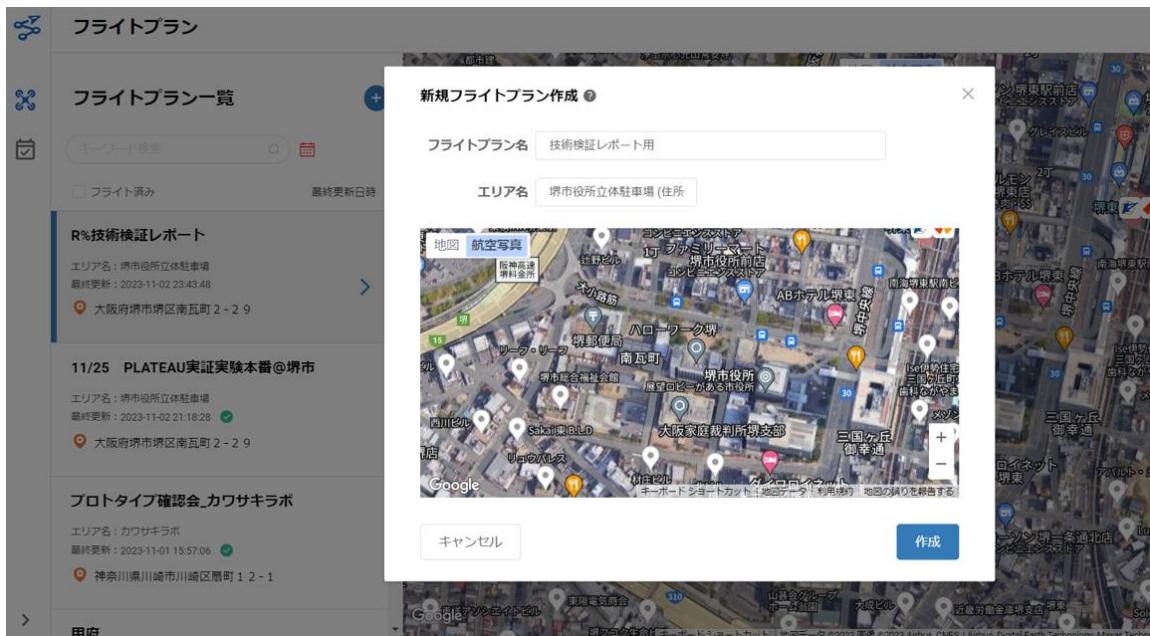


図 3-60 フライトプランの作成

2) 新規ルートの作成

- ルート名、飛行方法を選択



図 3-61 新規ルートの作成

3) ウェイポイント設定

- 機体を選択し、マウスを用いて任意のウェイポイントを設定
- 高度・速度設定

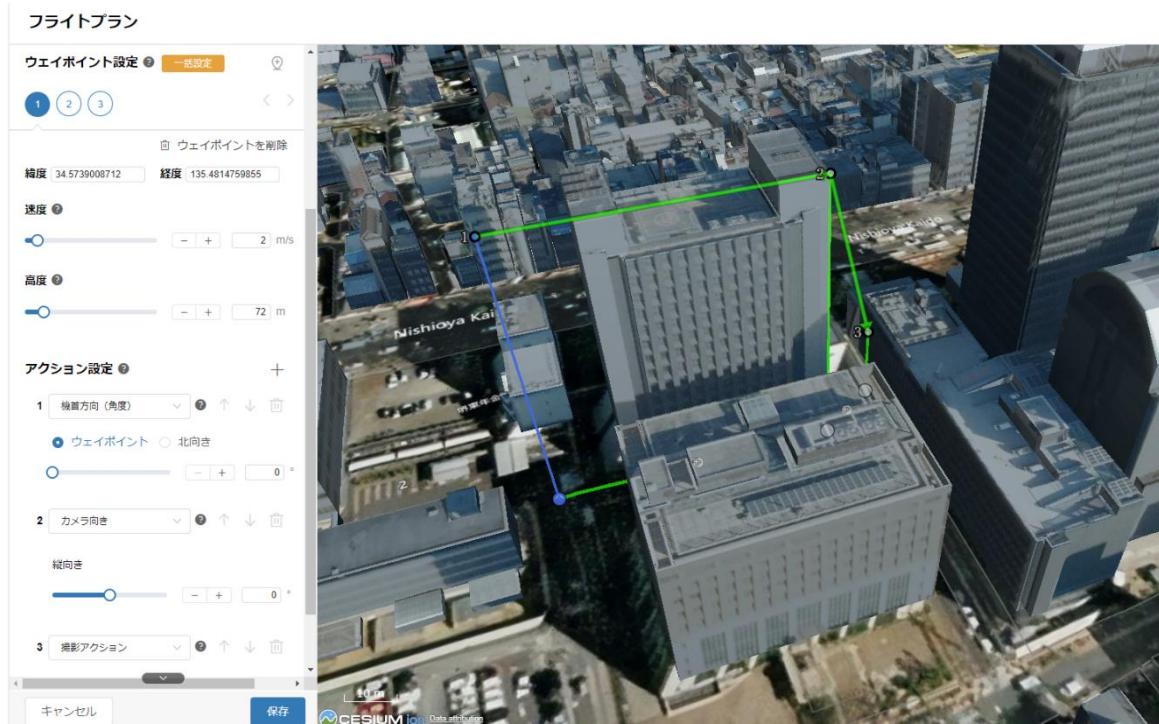


図 3-62 ウェイポイント設定

4) 飛行実施

- 機体にルートをアップロードし飛行開始

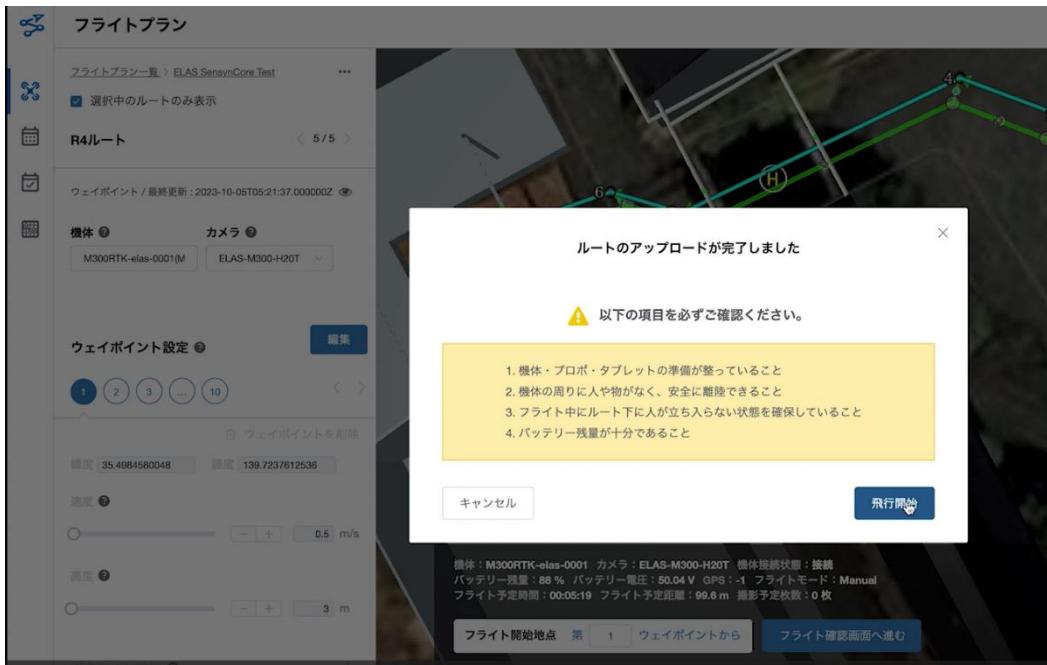


図 3-63 飛行実施

5) 飛行状態の確認

- テレメトリーとして、現在位置座標/GNSSRTK/LiDAR測位状態/バッテリー情報の表示

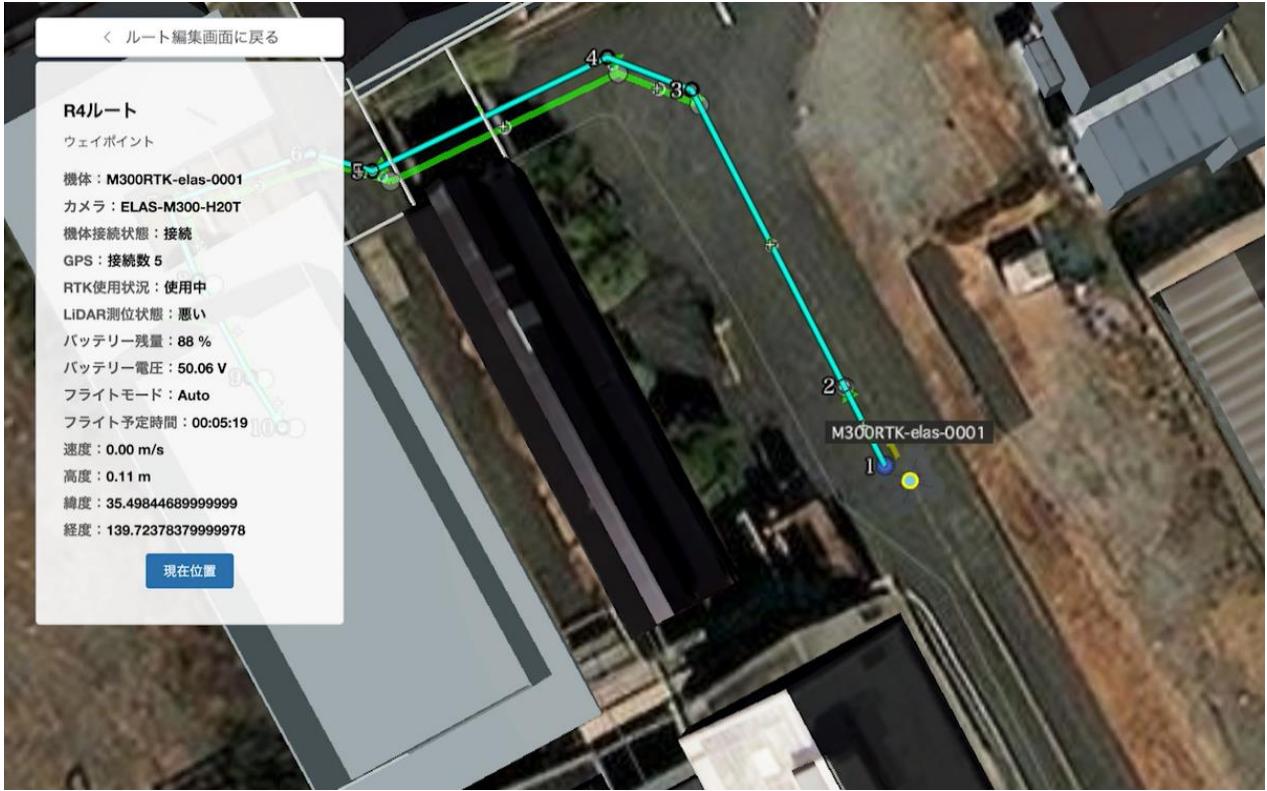


図 3-64 飛行状態の確認

6) 緊急対応

● ミッション一時中断、再開

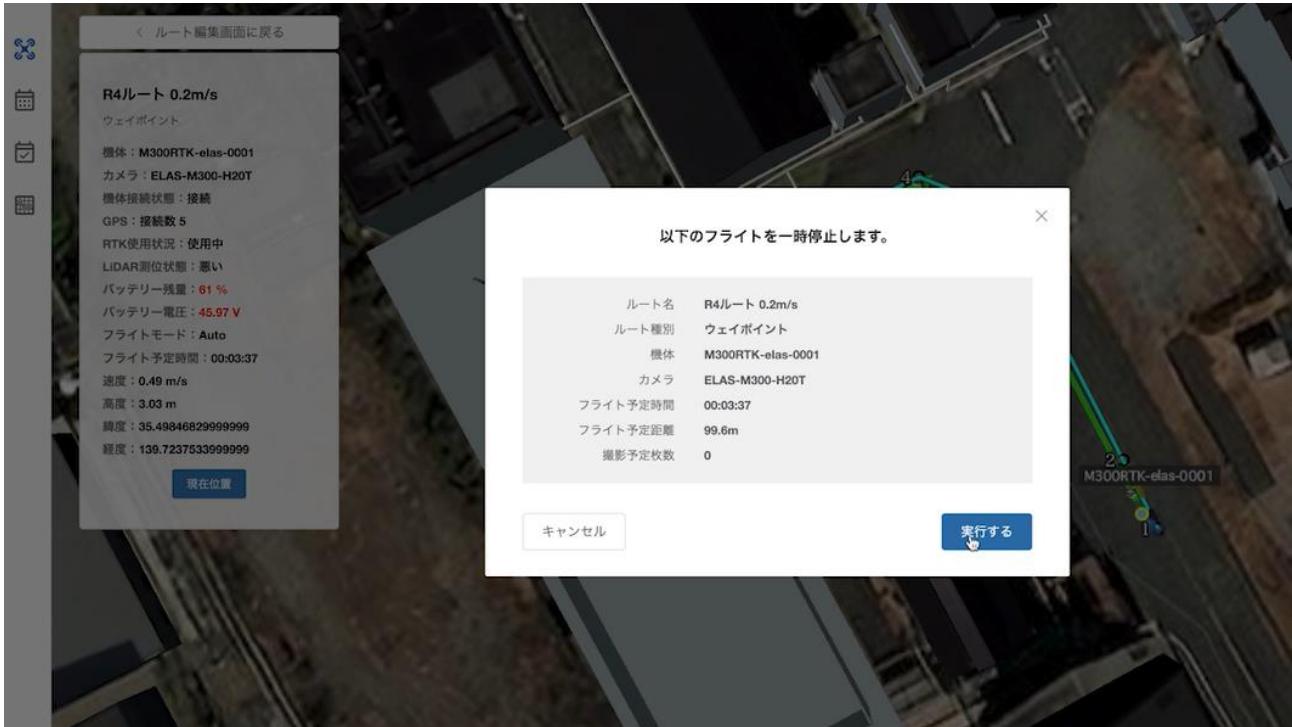


図 3-65 緊急停止の実行画面

4. 実証技術の検証

4-1. 自己位置推定機能の検証

4-1-1. 検証目的

- 3D LiDAR ローカリゼーションで地上面を測定できなくなる 35m以上の高さでビジュアルイナーシャルオドメトリ使用した場合と使用しない場合の自己位置推定の精度を比較することで、今回開発したアルゴリズムの優位性を確認する。

4-1-2. KPI

表 4-1 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法サマリー
1	3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ 35m以上の高さで自律飛行を継続達成	達成	<ul style="list-style-type: none"> 自律飛行を継続達成できたか（達成/未達）を評価する 昨年度実績を踏まえ、目標値を設定した 高さ：35m（3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ） 飛行距離：約 150m <p>【昨年実績】</p> <ul style="list-style-type: none"> 高さ：10m 飛行距離：75m 	<p>【前提条件】</p> <ul style="list-style-type: none"> 3D 都市モデル (CityGML) LEVEL : LOD2 天候：晴れ又は曇り 風速：5m 以下 大阪府堺市 堺市役所・堺地方合同庁舎周辺地域において、KPI 目標高度 35m を満たす経路で実施

4-1-3. 検証方法と検証シナリオ

● 検証シナリオ

- 1) 3D LiDAR ローカリゼーションで地上面を特定できなくなる 35m 以上の高さで自律飛行を継続達成

RTK による自己位置測位が使用できない状況において、LiDAR 自己位置測位及び VIO を使用した場合の自律飛行の達成/未達を KPI として設定する。また、自律飛行達成時は、RTK の測位情報を真値として比較した精度を確認する。

VIO を使用しない場合については、実際に飛行した際に記録したデータを使用し、VIO のデータを抜いた状態

で検証を行った。

RTKの測位情報を真値として比較した精度は以下の方法で算出する。

前提条件：堺市役所・堺地方合同庁舎周辺地域に作成したドローンルート（堺地方合同庁舎の壁面に沿って、一定の離隔を保ちつつ高度約70mまで直上するルート）でドローンを飛行させる。

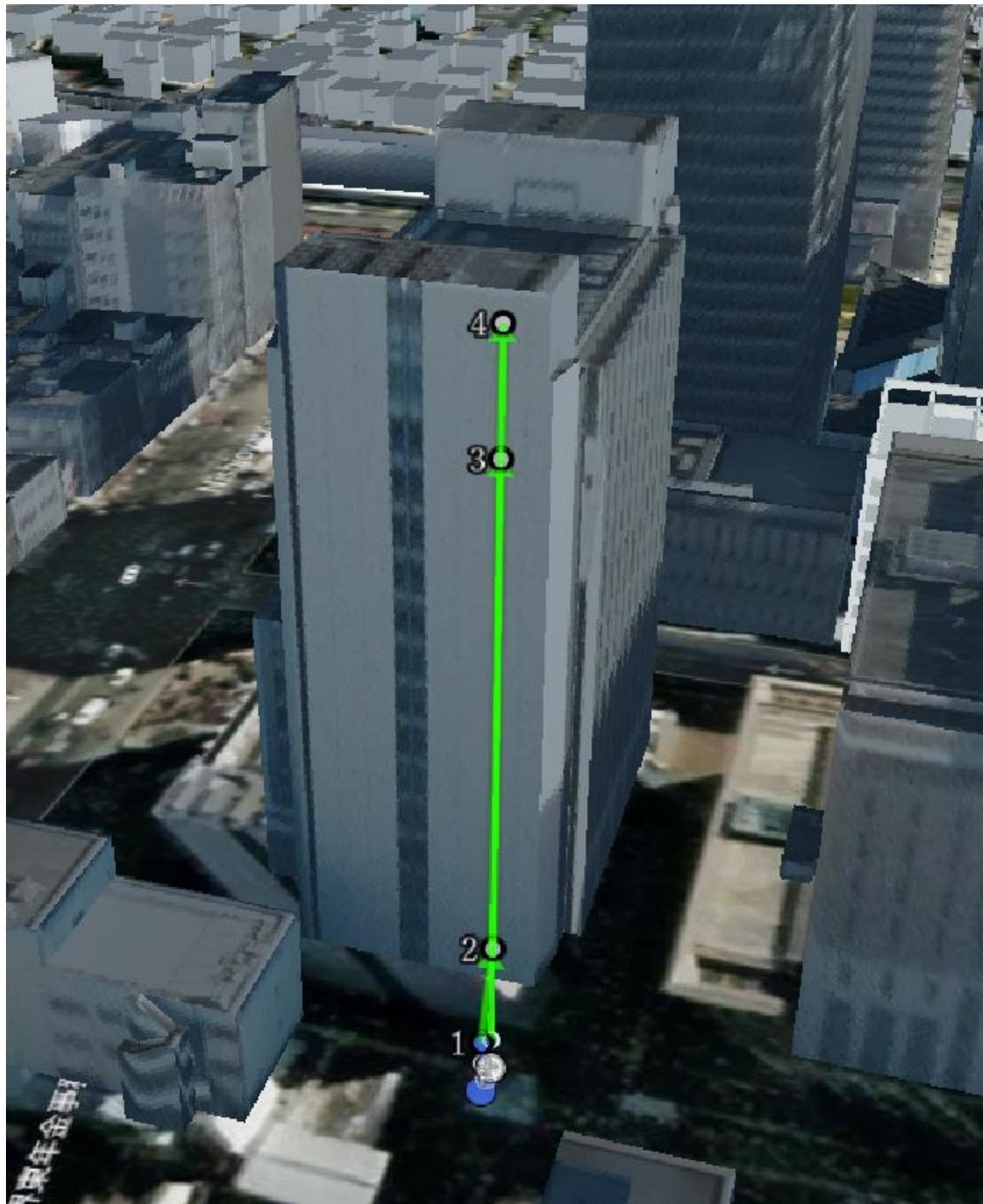


図 4-1 自己位置推定機能の検証で用いたドローンルート

正解データ (target) : RTK の測位情報

検証データ (result) : LiDAR 自己位置測位及び VIO を使用した場合の、自律飛行中のドローン飛行位置

精度は以下の方法で求める。

α_t : 離陸後の時間 t における、正解データ位置ベクトル (target)

β_t : 離陸後の時間 t における、検証データー位置ベクトル (result)

$$\text{正解データとの距離(m)} = \|\alpha_t - \beta_t\|$$

μ : 離陸後の各時間における t_x における正解データとの距離(m)の平均値

$$\text{正解データとの平均距離(m)} = \frac{1}{n} \sum_{t=0}^n (\|\alpha_t - \beta_t\|)$$

表 4-2 検証シナリオ一覧（精度）

No.	検証方法	正解データ	比較するデータ
1-1	正解データとの平均距離 (m)	RTK の測位情報	LiDAR 自己位置測位及び VIO を使用した場合の、自律飛行中のドローン飛行位置

- その他

- 実施日
 - ✧ 2023年11月25日
- 主な参加者
 - ✧ 全体統括、現場統括、現場進行、プロジェクトマネージャー：各1人
 - ✧ パイロット、操縦補助者、ソフトウェアエンジニア：各2人

4-1-4. 検証結果

1. 3D LiDAR ローカリゼーションで地上面を特定できなくなる 35m 以上の高さで自律飛行を継続達成

● 検証結果まとめ

- ビジュアルイナーシャルオドメトリ (VIO) を使用した場合、自律飛行を達成したが、使用しない場合、途中で自己位置を失い、自己位置推定に失敗することが分かった。また、自己位置推定に失敗したことで検証ルートから大きく外れ、自律飛行を達成できなかったため、ビジュアルイナーシャルオドメトリ (VIO) を利用した自己位置推定機能の優位性が示された。
- また、VIO あり状態の自律飛行の自己位置推定結果を RTK 測位情報と精度を比較した結果、平均して 1.38m の誤差が確認された。
 - ✧ このときの水平方向の誤差は 0.87m、高さ方向の誤差は 0.91m であった。水平方向の誤差は飛行する高さによる変化は見られなかつたが、高さ方向の誤差についてはドローンが上昇するにつれて誤差が大きくなる傾向が確認された。
 - ✧ この高さ方向の誤差の変化については、VIO による累積誤差の影響の他に、3D 都市モデル (LOD2.0) の建築物モデルの仕様に由来する差（現実の建物が LOD2.0 建築物モデルよりも高い）が影響していることが示唆された。LiDAR による自己位置推定では 3D 都市モデル (LOD2.0) を点群化したマップを利用しており、ドローンが上昇し現実と点群マップ間で誤差がある壁面最上部に近づくにつれて LiDAR ローカリゼーション機能で実際より低い位置の自己位置を推定している可能性がある。さらにその自己位置推定情報が測位情報として統合されたため、最終的な自己位置推定結果の誤差が上昇するにつれ大きくなつたと考えられる。

表 4-3 検証結果サマリー

黄セル：KPI 達成	青セル：KPI 未達
------------	------------

検証内容	評価指標・KPI	目標値	結果		示唆
			項目	評価	
3D LiDAR ローカリゼーションで地上面を特定できなくなる 35m 以上の高さで自律飛行	自律飛行を継続達成	達成	LiDAR 自己位置推定 + VIO あり	達成	VIO を使用しない場合、途中で自己位置を見失ってしまったことから、VIO は高高度ビル周辺での飛行において必須だと考えられる
			LiDAR 自己位置推定、VIO なし	未達	

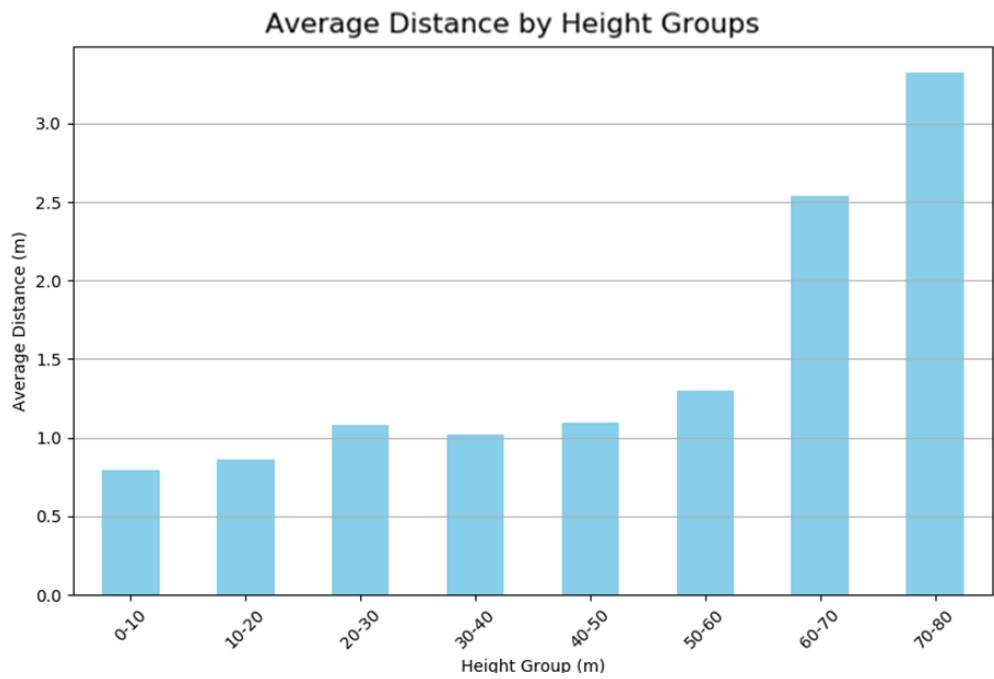


図 4-2 飛行高度ごとの距離誤差（平均値）

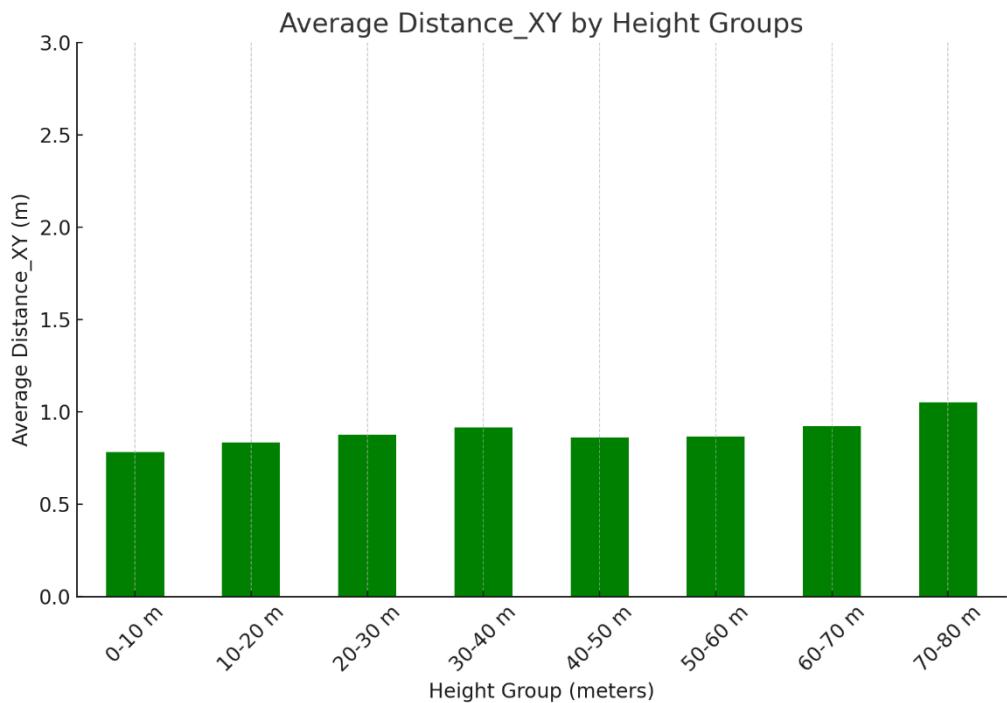


図 4-3 飛行高度ごとの水平距離誤差（平均値）

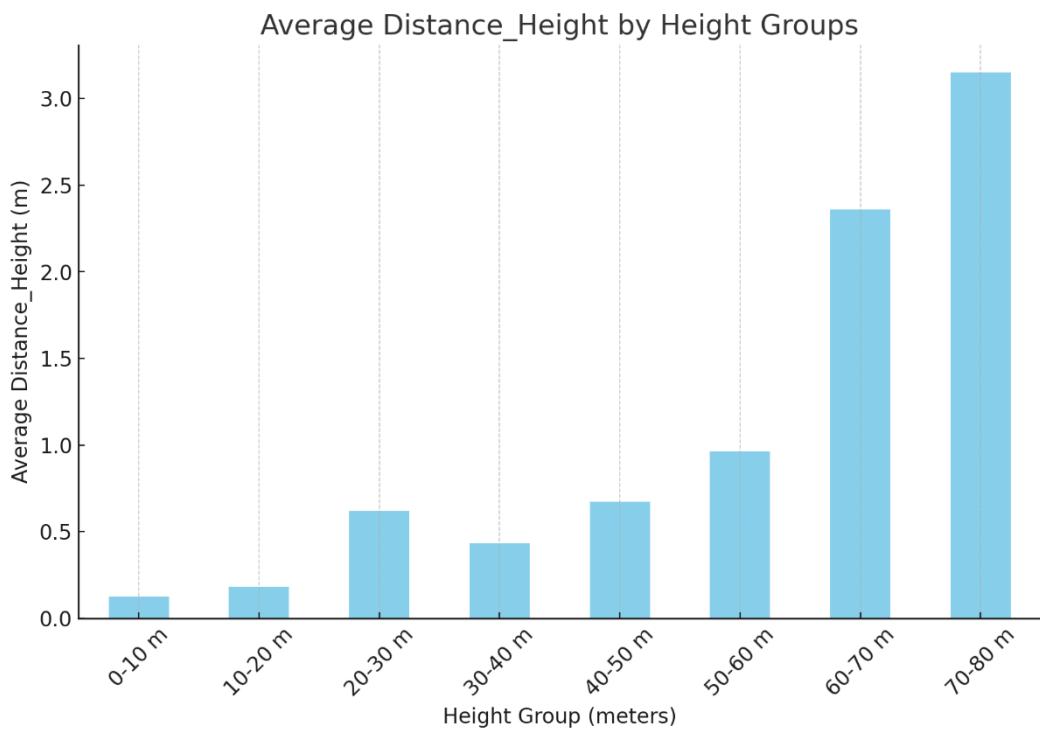


図 4-4 飛行高度ごとの高度誤差（平均値）

精度の検証について、全体的な距離誤差は飛行高度が高くなるにつれ大きくなっていることが示された（図 4-2）。このとき、水平方向の誤差は飛行高度による差は見られなかった（図 4-3）が、高さ方向の誤差については全体的な距離誤差と同様の傾向を持っている（図 4-4）ことが示された。この傾向については VIO による累積誤差が影響し高さ方向にズレが発生しているとも考えられるが、別の要因も影響していることが分かった。

図4-5はRTKを自己位置推定に取り入れた場合での高度70m付近での様子であるが、3D都市モデル(LOD2)の壁面の高さと、実際にLiDARで捉えた壁面の高さが一致しないことが見て取れる。

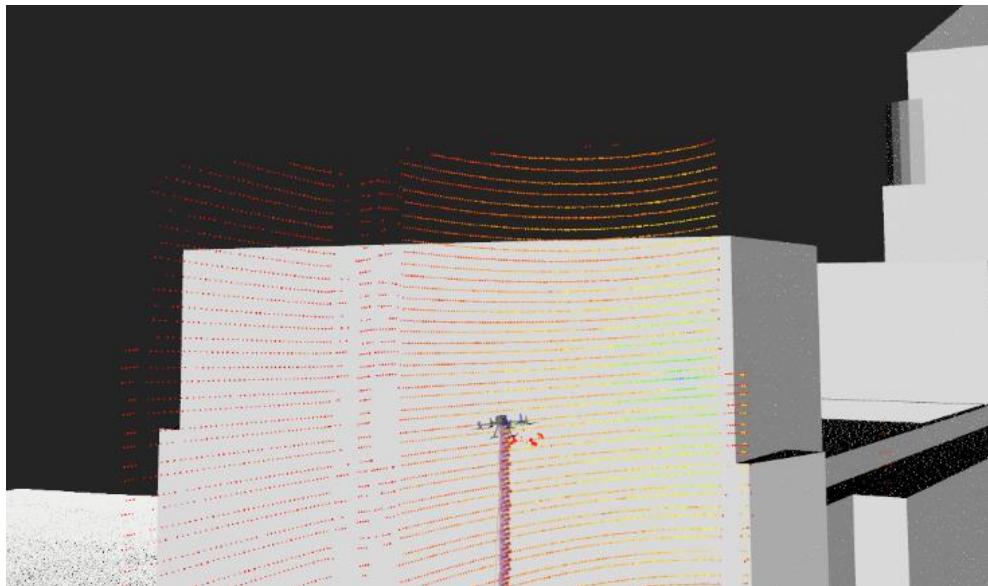


図4-5 モデルの壁面とLiDARによりとらえた壁面の高さがずれている様子
(LOD2.0の建築物より上部にLiDARの反射があることがわかる)

一方、地面から高さ数メートル地点における自己位置推定の場合は3D都市モデル(LOD2)の壁面の高さと、実際にLiDARで捉えた壁面の高さが一致している(図4-6)。

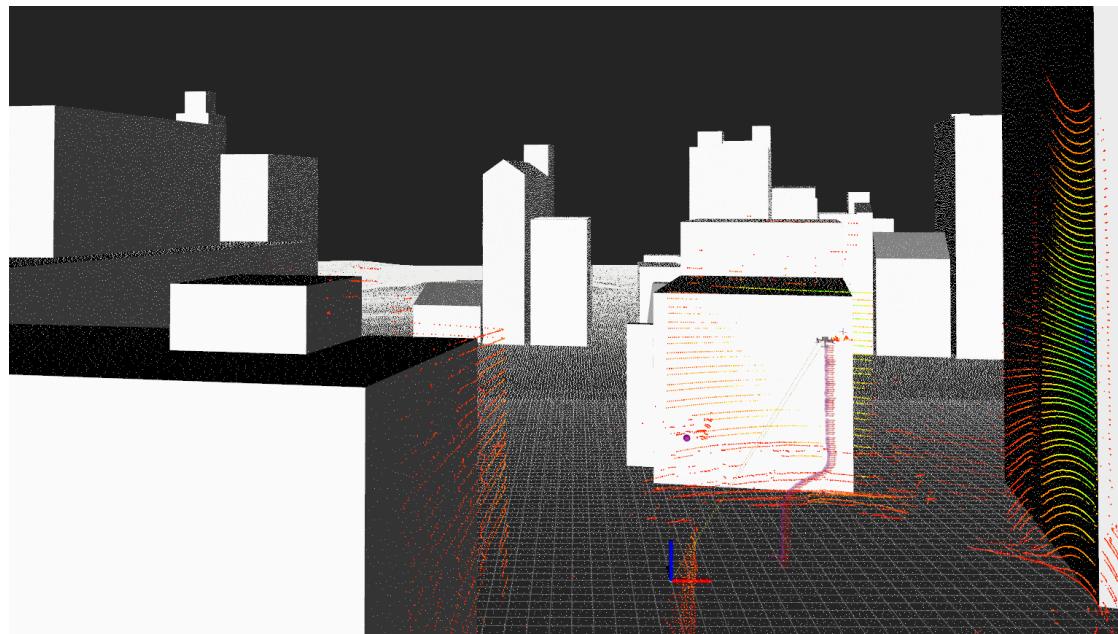


図4-6 モデルの壁面とLiDARによりとらえた壁面の高さが一致している様子
(画像手前のLOD2.0の建築物に対して、LiDARの反射の高さ一致していることがわかる)

のことから堺地方合同庁舎の3D都市モデル（LOD2）が現実の高さと一致しておらず、それにより70m付近でのRTK自己位置とVIOとLiDARによる自己位置推定結果の高さ方向の誤差が特に大きくなっていると考えられた。

上記結果および考察を踏まえ、堺地方合同庁舎の3D都市モデル（LOD2）が現実の高さと一致していない可能性について検証するため、RTK測位による建物の対地高度の実測値と3D都市モデル（LOD2）の対地高度との比較、およびモデル整備を行った事業者等に対して前述のLOD2の建築物モデルの対地高度の正確さについての確認を行った。その結果を表4-7に示す。

表4-4 各参照データにおける堺地方合同庁舎の対地高度一覧

No.	参照データ	地面に接する位置の標高 …①	建物最上部の標高 …②	建物最上部の対地高度 …(②) - (①)
1	設計図面	—	—	69.9m (「薄壁」含めた頂点)
2	3D都市モデル (LOD2)	5.51m	72.94m	67.43m
3	RTK測位情報	6.3m	75.3m	69.0m

上記の比較結果より、RTK測位による結果のほうが3D都市モデル（LOD2）よりも設計図面に近い対地高度を測位していることが示された。RTK測位情報と設計図面との0.9mの差については、設計図面と実際の施工結果との差分や、ドローン離陸地点の標高と建物の接地面の標高の差分等が原因で発生していると考えられる。

また、モデル整備を行った事業者等に問い合わせを行った結果、3D都市モデル（LOD2）の対地高度は正確であるが、図4-7で示した堺地方合同庁舎の壁面最上部=屋上部分から上に張り出している「薄壁」部分が3D都市モデル（LOD2.0）の建築物モデルでは表現出来ないため、設計図面との差が発生しているとの回答を得た。これにより、3D都市モデル上の堺地方合同庁舎壁面最上部の高さが「薄壁」の最上部ではなく屋上面の高さとなることで実際の壁面最上部よりも約1.0m~1.1m低く表現されたことが、後述の通りドローンの自己位置推定に影響を及ぼし自己位置推定結果の高さ方向の誤差に影響している可能性が示された。



図 4-7 堺地方合同庁舎屋上の「薄壁」(図中の赤枠部分)

※上記写真は実証実験で飛行した壁面と反対側の壁面を上から撮影したものであるが、

構造は左右対称のため実証実験で飛行した壁面と大きな違いはない

LiDAR による自己位置推定では 3D 都市モデル (LOD2.0) を点群化したマップを利用している。そのため、RTK を使用せず LiDAR およびビジュアルイナーシャルオドメトリ (VIO) のみを利用した飛行では、ドローンが上昇飛行し誤差がある壁面最上部に近づくにつれ、図 4-8 の赤線で示した LiDAR で捉えた壁面最上部とドローンとの距離 α と、図 4-8 の青線で示した点群化した 3D 都市モデル (LOD2.0) の壁面最上部と点群マップ内のドローン位置との距離 α' の差が最小化された位置を自己位置と推定する。このため、図 4-9 に示すように実際の飛行高度より低い位置の、誤った自己位置推定結果となったと考えられる。

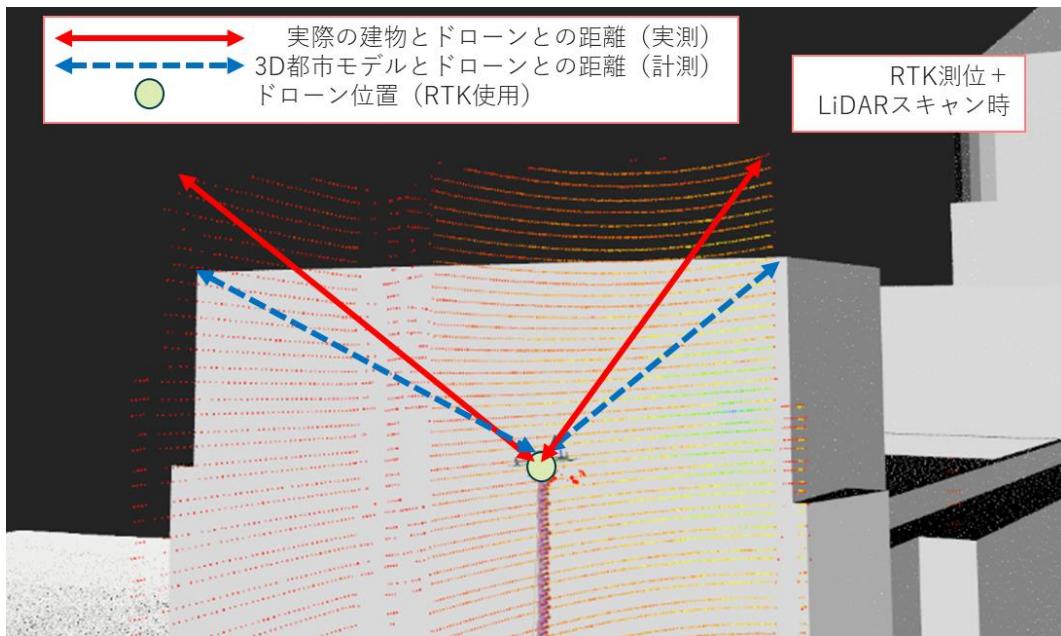


図 4-8 RTK 測位を自己位置推定に取り入れた場合での高度 70m 付近での様子

※RTK 測位により正しい自己位置を飛行できるため、LiDAR スキャン結果と 3D 都市モデル（LOD2）の壁面と LiDAR によりとらえた壁面の高さがずれている

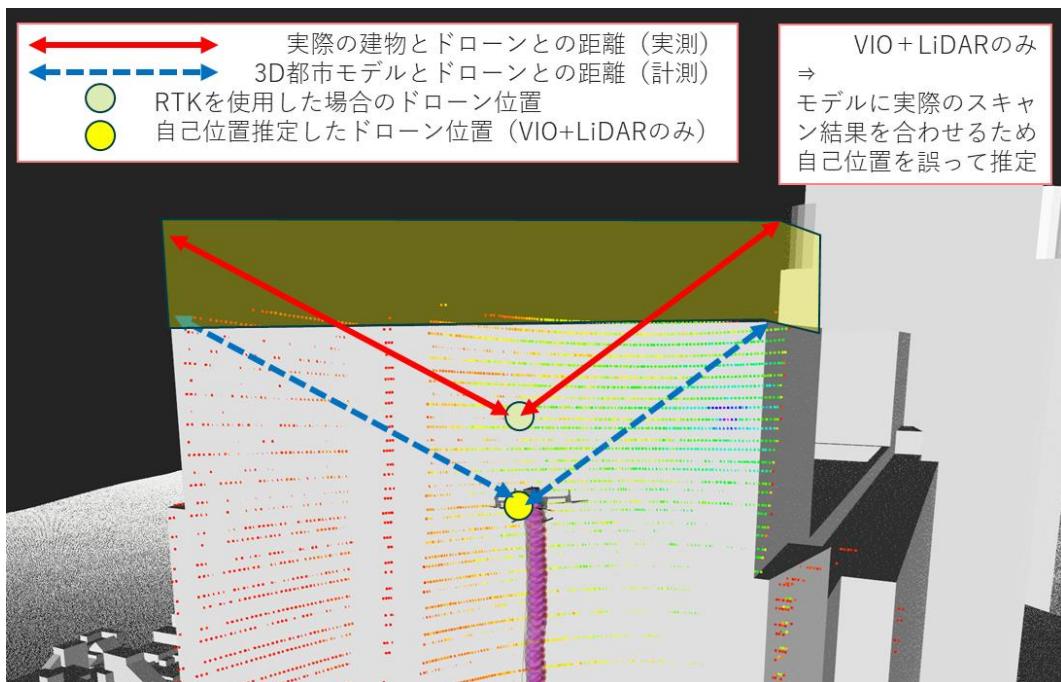


図 4-9 RTK を用いず、VIO と LiDAR のみで飛行した場合

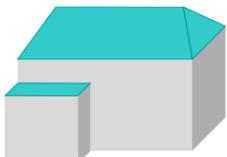
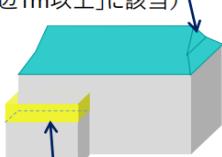
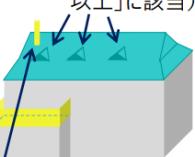
現実と差がある 3D 都市モデル（LOD2.0）を正として自己位置を推定した結果

誤った自己位置を推定している

※薄い黄色で示された面が 3D 都市モデル（LOD2.0）と実際の建物との差分

この「薄壁」は3D都市モデル（LOD2.1）では表現することができるため、LiDAR自己位置推定用の点群マップ作成時に、3D都市モデルLOD2.0ではなくLOD2.1を利用することで今回確認された高さ方向の誤差を小さくできる可能性が示唆された。

また、この3D都市モデル（LOD2.0）の建築物モデルにおける仕様に由来する高さの誤差は他の建築物でも同様に起きうるため、3D都市モデル（LOD2.0）の建築物モデルを利用して建物と近い距離を飛行するドローンルート設計をする際には、衝突回避の観点での注意する必要があることが分かった。

LOD	LOD2.0	LOD2.1	LOD2.2
取得例			
説明	屋根の主要な外形が再現される。LOD2.0では付属物は取得しないため、バルコニーも屋根として取得する。なお、LOD2では屋根面は詳細化されるが壁面は詳細化されないため、バルコニーの下部も建築物の一部として表現される。	小屋根のうち規模が大きいものが再現される。LOD2.0では切妻屋根として表現されたが、LOD2.1の条件を満たしたため、小屋根として表現された。また、LOD2.1の条件を満たすバルコニーが、付属物として区分される。	小屋根のうち規模の小さいものが再現される。LOD2.1では無視された屋根窓の屋根がLOD2.2の条件を満たしたため、この屋根形状が表現された。また、LOD2.2の条件を満たす屋根上の煙突が付属物として、さらに区分される。

■ RoofSurface ■ WallSurface ■ BuildingInstallation

図 4-10 建築物モデル（LOD2.1）のイメージ¹²

¹² 出典：3D都市モデル標準製品仕様書第3.5版

(https://www.mlit.go.jp/plateau/file/libraries/doc/plateau_doc_0001_ver03.pdf)

4-2. 測位情報の統合アルゴリズムの検証

4-2-1. 検証目的

- 3D都市モデルを事前地図として活用することで、事前の撮影業務を行わずに精度の高い事前地図を作成し、測位情報の統合アルゴリズムによる高い自己位置推定精度を実現したドローン自律飛行システムが確立できることを確認する。

4-2-2. KPI

表 4-5 KPI一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法サマリー
1	3D LiDAR ローカリゼーションで地上面を特定できる高さ 35m 以上の高さで自律飛行を継続達成	達成	<ul style="list-style-type: none"> 自律飛行を継続達成できたか（達成/未達）を評価する 昨年度実績を踏まえ、目標値を設定した 高さ：35m（3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ） 飛行距離：約 150m <p>【昨年実績】</p> <ul style="list-style-type: none"> 高さ：10m 飛行距離：75m 	<p>【前提条件】</p> <ul style="list-style-type: none"> 3D都市モデル（CityGML）LEVEL：LOD2 天候：晴れ又は曇り 風速：5m 以下 大阪府堺市 堺市役所・堺地方合同庁舎周辺地域において、KPI 目標高度 35m を満たす経路で実施

4-2-3. 検証方法と検証シナリオ

1) 飛行総距離

- 検証方法
 - 本実証実験では、大阪府堺市堺区にて堺市役所・堺地方合同庁舎及びその周辺部でのドローン自律飛行を行った。ドローン飛行に用いた点群マップは、堺市堺区の3D都市モデル（LOD2）と、BIMモデルから変換し生成した3D都市モデル（LOD4）を活用した。
 - 飛行ルートは、堺地方合同庁舎周辺部の屋外から離陸し、堺市役所敷地内の建物の屋内に着陸するように設計された、飛行最大高度約70m、飛行総距離238mの経路である。
- 検証に用いたルート
 - 以下2つのパート（長距離飛行検証パート、屋内外シームレス飛行検証パート）を合成し、1つの経路にした飛行ルートで検証を行った。

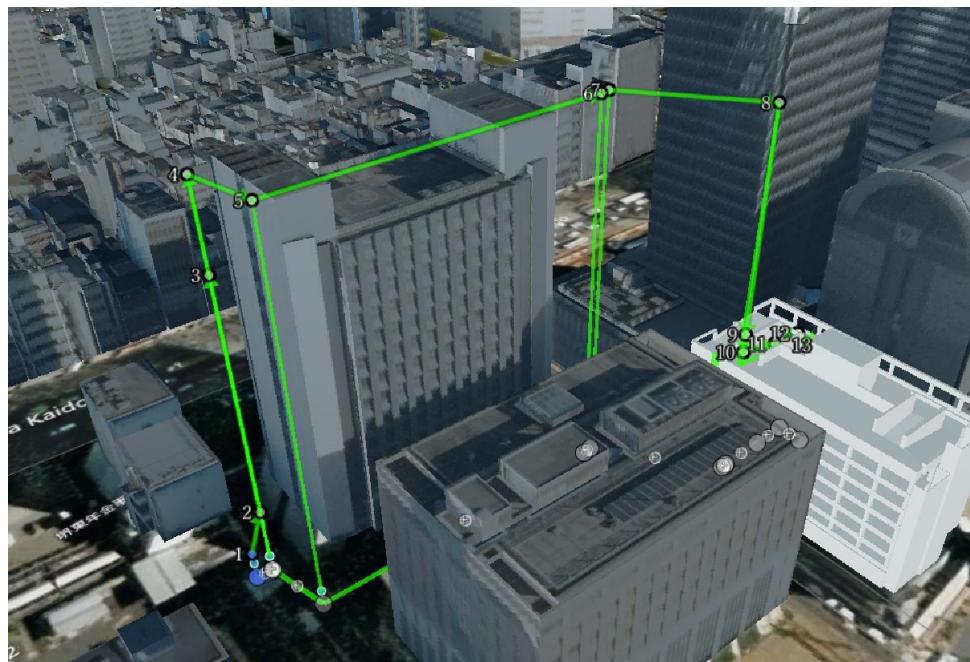


図 4-11 検証に用いたルート（全体）

- 長距離飛行検証パート
 - ✧ 着陸後直上し、建物に対して水平方向のルート部分
 - ✧ 建物の壁面から 5m 程度の離隔を保って飛行し、自律飛行が達成できるかを確認する

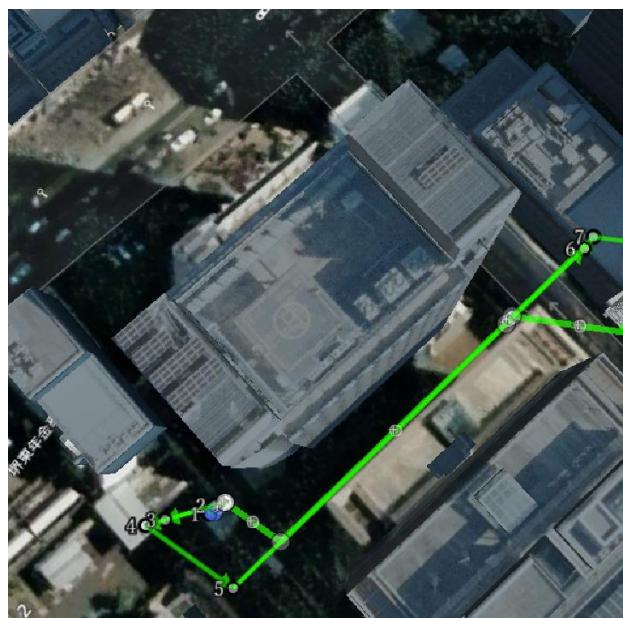


図 4-12 検証に用いたルート（長距離飛行検証パート）

緑線：飛行ルート

- 屋内外シームレス飛行検証パート
 - ✧ 屋外から屋内にシームレス飛行を行うルート部分
 - ✧ RTK 測位が途切れる屋内侵入時に、LiDAR 及びビジュアルインーシャルオドメトリによる自己位置測位が正常に作動し、着陸地点まで自律飛行を達成できるかどうかを確認する

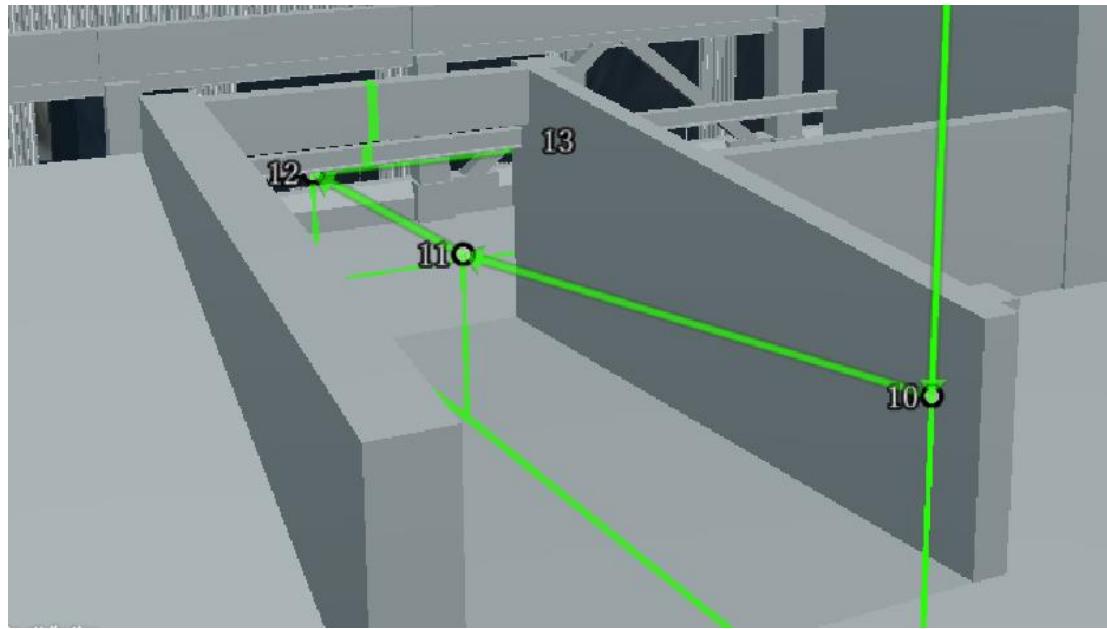


図 4-13 検証に用いたルート（屋内外シームレス飛行検証パート）

緑線：飛行ルート、横方向から見た様子

※ウェイポイント 12~13 の屋内部分はルート作成を行いやすくするため屋根が非表示になっている



図 4-14 検証に用いたルート（屋内外シームレス飛行検証パート）

緑線：飛行ルート、上方向から見た様子

※ウェイポイント 12~13 の屋内部分はルート作成を行いやすくするため屋根が非表示になっている

- 検証シナリオ

表 4-6 検証シナリオ一覧

No.	自己位置推定技術	事前地図	フライトルート
1-1-1	3D LiDAR	3D都市モデル（LOD2）	長距離飛行検証パート
1-1-2		3D都市モデル（LOD4）	屋内外シームレス飛行検証パート
1-2-1	IMU	なし	長距離飛行検証パート
1-2-2			屋内外シームレス飛行検証パート
1-3-1	ビジュアル	なし	長距離飛行検証パート
1-3-2	イナーシャル オドメトリ（VIO）		屋内外シームレス飛行検証パート

- その他

- 実施日
 - ✧ 2023年11月25日
- 主な参加者
 - ✧ 全体統括、現場統括、現場進行、プロジェクトマネージャー：各1人
 - ✧ パイロット、操縦補助者、ソフトウェアエンジニア：各2人

4-2-4. 検証結果

1. 飛行総距離

- 検証結果まとめ

- 本実証実験の結果、飛行最大高度約70m 飛行総距離238mの経路を、屋外の離陸地点から屋内の着陸地点に着陸するまでシームレスに自律飛行し、KPIを達成した。

表 4-73 検証結果サマリー

	黄セル：KPI達成	青セル：KPI未達
--	-----------	-----------

検証内容	評価指標・KPI	目標値	結果		示唆
			項目	評価	
3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ35m以上の高さで自律飛行	自律飛行を継続達成	達成	【屋外飛行時】 RTK 使用 【屋内飛行時】 LiDAR 自己位置推定 + VIO	達成	<ul style="list-style-type: none"> ● 屋外から屋内への侵入時、RTKが使用できなくなったタイミングで一時停止することなく、シームレスにLiDAR測位+ビジュアルインーシャルオドメトリ(VIO)による自己位置測位を行うことが示された。 ● これにより、測位情報の統合アルゴリズムの有用性が示された。

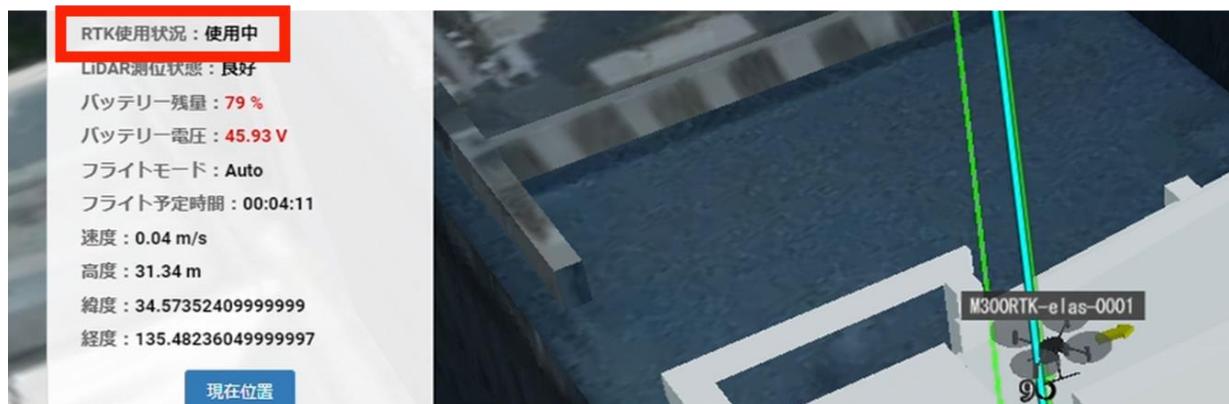


図 4-15 屋外飛行中のテレメトリー情報 (RTK 使用中)

uc23-17-1_技術検証レポート_3D 都市モデルと BIM を活用したモビリティ自律運航システム(ドローン)v2.0

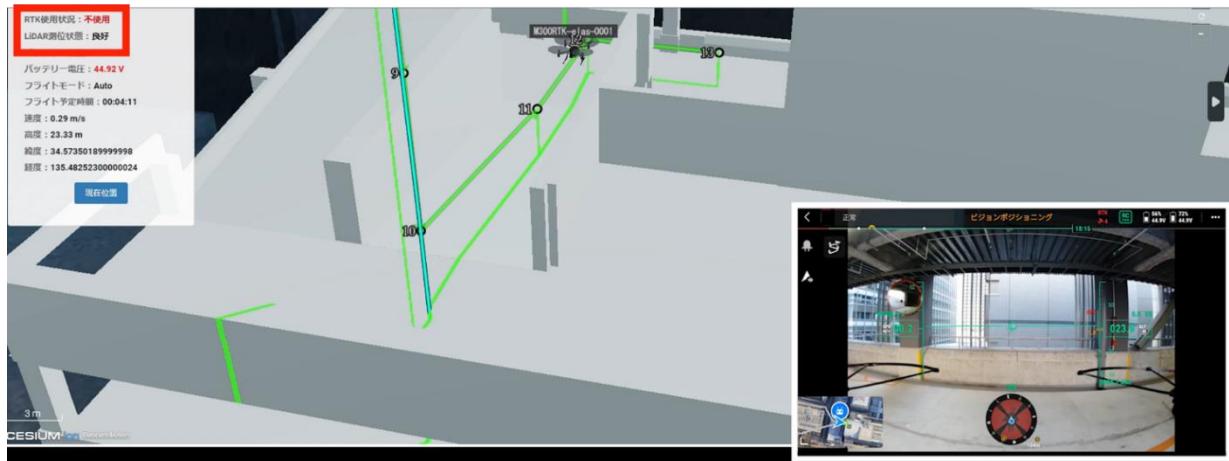


図 4-16 屋内飛行中のテレメトリー情報（RTK が使用できない場合、LiDAR 測位と VIO を使用）

4-3. ビジュアルライナーシャルオドメトリによる高さ方向への移動の検証

4-3-1. 検証目的

- 昨年度の実証では、LiDAR ローカリゼーションだけでは均一な特徴点を持つ建築物（例えば、同じ形状の窓が並ぶ高層ビルなど）を垂直方向に長距離移動する場合などは、特徴点をとらえられず飛行精度が低下するという飛行面の課題があった。この課題の克服のため本年度はビジュアルライナーシャルオドメトリ（VIO）を開発し、垂直方向の移動量推定のロバスト化を図った。
- RTK 測位情報を使用しない状態で、均一な特徴点を持つ高さ 35m 以上の建築物である堺地方合同庁舎の壁面を、垂直方向に 3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ 35m を超える高さまで自律飛行した際には垂直方向の自己位置推定が安定しないことが予想される。このような状況におけるビジュアルライナーシャルオドメトリ（VIO）の有用性を示す。

4-3-2. KPI

表 4-8 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法サマリー
1	3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ 35m 以上の高さで自律飛行を継続達成	達成	<ul style="list-style-type: none"> 自律飛行を継続達成できたか（達成/未達）を評価する 昨年度実績を踏まえ、目標値を設定した 高さ：35m（3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ） 飛行距離：約 150m（暫定） <p>【昨年実績】</p> <ul style="list-style-type: none"> 高さ：10m 飛行距離：75m 	<p>【前提条件】</p> <ul style="list-style-type: none"> 3D 都市モデル（CityGML）LEVEL : LOD2 天候：晴れ又は曇り 風速：5m 以下 大阪府堺市 堀市役所・堺地方合同庁舎周辺地域において、KPI 目標高度 35m を満たす経路で実施

4-3-3. 検証方法と検証シナリオ

1) 飛行総距離

● 検証方法

- 本実証実験では、大阪府堺市堺区にて堺市役所・堺地方合同庁舎及びその周辺部でのドローン自律飛行を行った。ドローン飛行に用いた点群マップは、堺市堺区の 3D 都市モデル（LOD2）を活用した。
- 飛行ルートは、堺地方合同庁舎周辺部の屋外から離陸し、堺地方合同庁舎の壁面から約 5m の離隔

を保って直上するように設計された、飛行最大高度約70m、飛行総距離約70mの経路である。

- 検証に用いたルート

- 堺地方合同庁舎の壁面から約5mの離隔を保って直上するように設計された飛行ルートで検証を行った

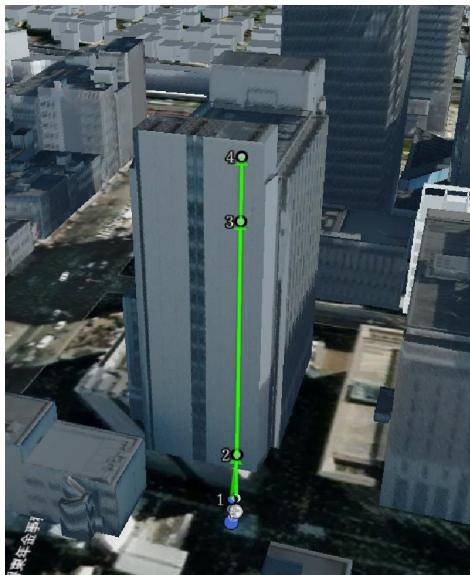


図 4-17 検証に用いたルート

- 検証シナリオ

表 4-9 検証シナリオ一覧

No.	検証方法	3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ	KPI 目標値
1-1	<p>検証場所： 堺地方合同庁舎の壁面沿い</p> <p>条件： RTK を送信機側で OFF に設定し、飛行中に RTK が使用できない (LiDAR 自己位置測位とビジュアルイナーシャルオドメトリが有効) 状態にする</p> <p>飛行方法： 堺地方合同庁舎の壁面沿いを、約5mの離隔を保つ直上するルートでドローンでの自律飛行を行う</p>	<p>35m</p> <p>根拠： LiDAR 照射距離が 50m であり、下方照射角度は最大 45° であることから、</p> <p>地上面を特定できる高さ(m)</p> $< 50(m) / \frac{1}{\sqrt{2}} \approx 35(m)$	自律飛行の達成

- その他

- 実施日
 - ✧ 2023年11月25日
- 主な参加者
 - ✧ 全体統括、現場統括、現場進行、プロジェクトマネージャー：各1人
 - ✧ パイロット、操縦補助者、ソフトウェアエンジニア：各2人

4-3-4. 検証結果

- 検証結果まとめ

- 3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ 35m の倍の高さ約 70m をゴールとするルートを自律飛行し、ゴールまでの飛行を達成した
- また、4-1 で示したとおり、ビジュアルイナーシャルオドメトリ (VIO) を使用しない場合は自己位置をロストし自律飛行を達成できなかったことから、ビジュアルイナーシャルオドメトリ (VIO) の結果を測位情報の統合アルゴリズムで使用する情報に追加することにより、LiDAR 自己位置測位のみの場合と比較してロバスト性が向上することが示された

表 4-10 検証結果サマリー

黄セル：KPI 達成	青セル：KPI 未達
------------	------------

検証内容	評価指標・ KPI	目標値	結果		示唆
			項目	評価	
3D LiDAR ローカリゼーションで地上面を特定できなくなる高さ 35m 以上の高さで自律飛行	自律飛行を継続達成	達成	LiDAR 自己位置 推定 + VIO あり	達成	● ビジュアルイナーシャルオドメトリの結果を測位情報の統合アルゴリズムで使用する情報に追加することにより、LiDAR 自己位置測位のみの場合と比較しロバスト性の向上が示唆された
			LiDAR 自己位置 推定、 VIO なし	未達	

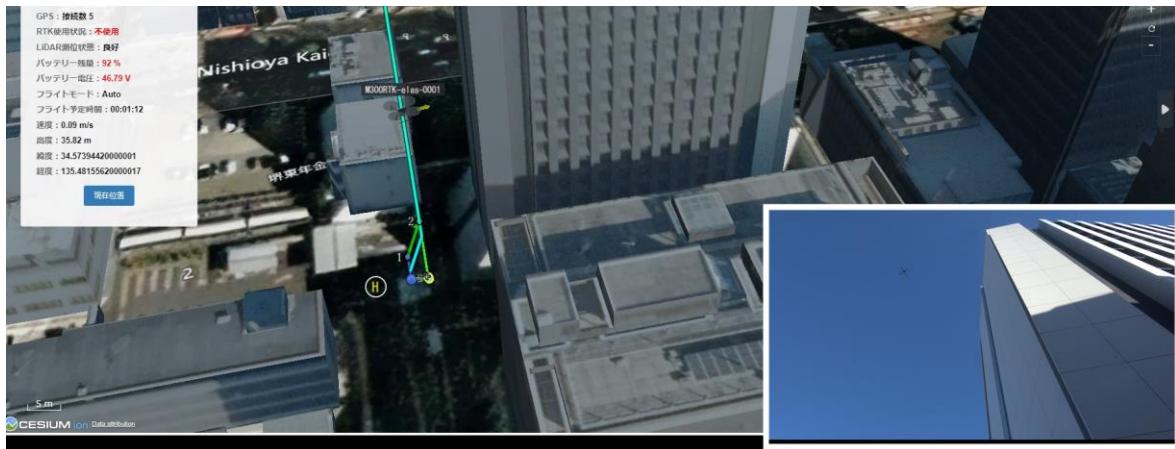


図 4-18 飛行中の様子（高度 35m 付近）

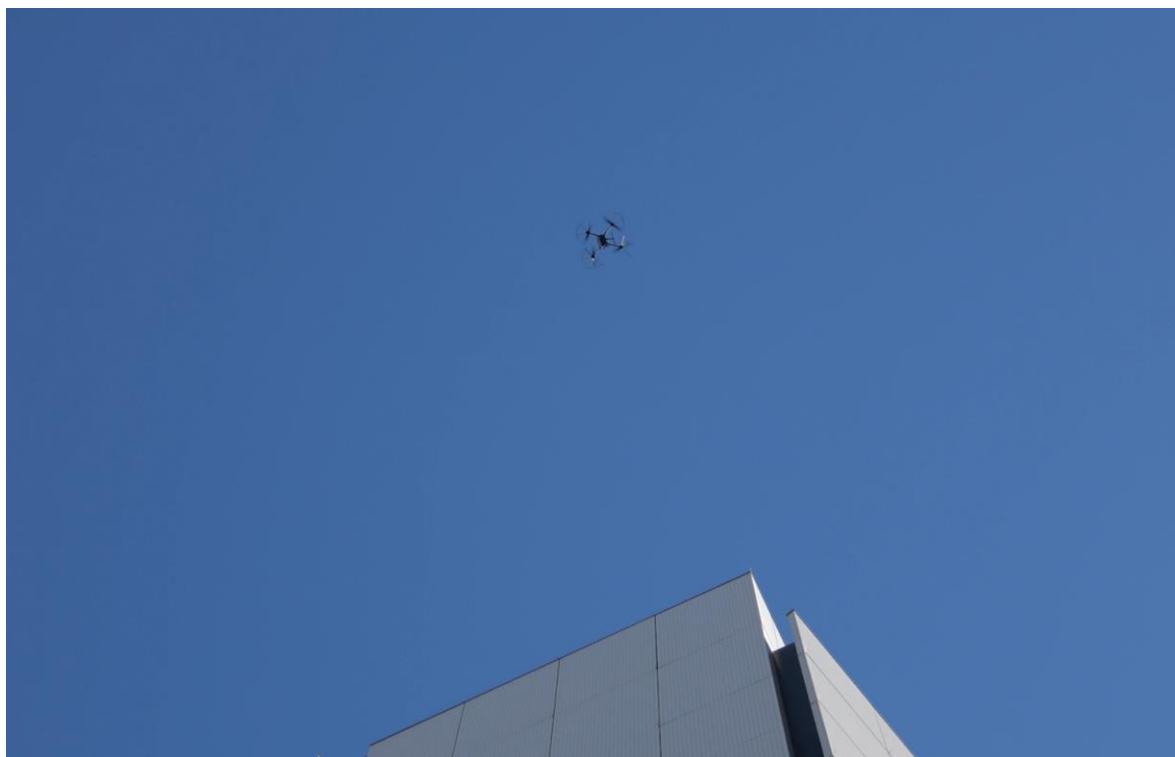


図 4-19 飛行中の様子（高度 70m 付近）

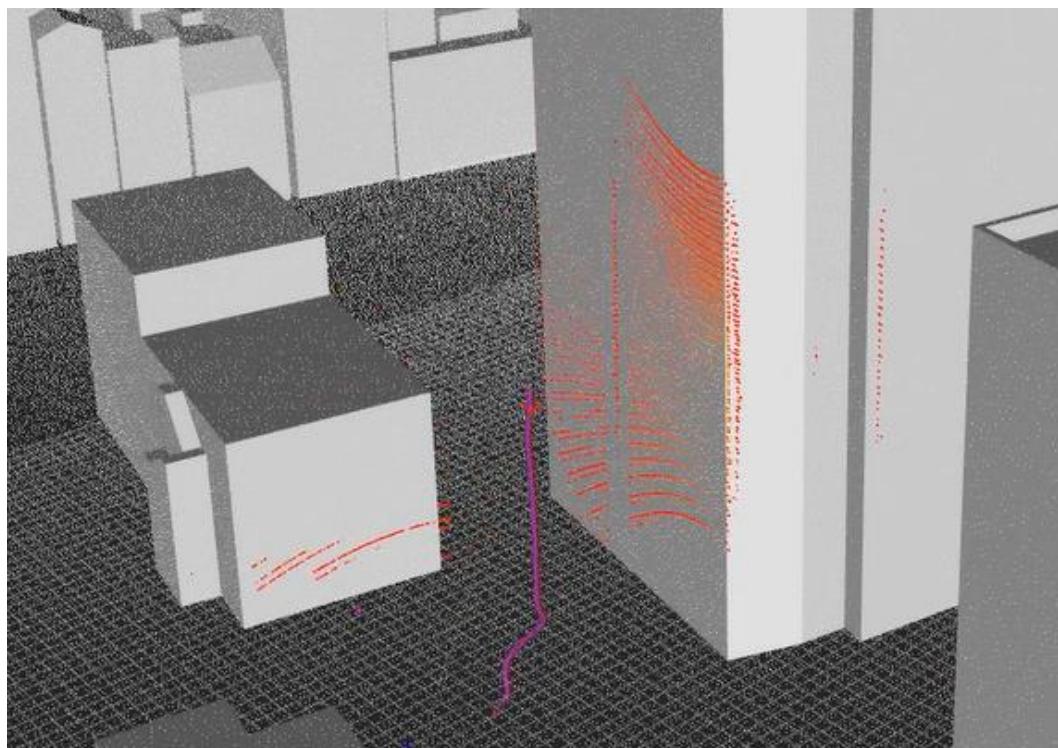


図 4-20 VIO を使用せず LiDAR のみで飛行した場合のシミュレーション結果

※この結果は記録したデータを元にしたシミュレーションにより得たもの。

中央の紫線がシミュレーション結果（ドローンの軌跡）で、上昇の途中で自己位置推定に失敗し自律飛行ルートから逸脱、意図しない下降を開始しているのが分かる。この後、ドローンは自律飛行を達成できなかった。

5. BtoB ビジネスでの有用性検証

5-1. 検証目的

実証仮説に基づき、以下の検証目的を設定する。

【検証仮説】

3D都市モデルを活用したドローンの自律運航システムについて、専門的な知識を持たないユーザーオペレーション可能なGUIを含むシステムをWebアプリケーション（株式会社センシンロボティクス製のGCSアプリ「SENSYN CORE Pilot」）上に追加開発することにより、誰でも手軽に利用することができる。

主に以下の2点について、BtoBビジネスに向けた有用性検証を行った。

- システムのユーザビリティ検証
 - 今年度追加開発した「SENSYN CORE Pilot」上での本システムにおけるUI/UXの分かりやすさ・使いやすさを確認する
- 既存システムとの利便性比較
 - 昨年度開発した既存の可視化システム（ドローン位置可視化ツール）「Rviz」と、本年度追加開発した「SENSYN CORE Pilot」の両方で同様のシミュレーションを行うことで、利便性や操作性の違いを確認する
 - シミュレーションには、本年度実証実験本番ルート（大阪府堺市）のデータを用いる

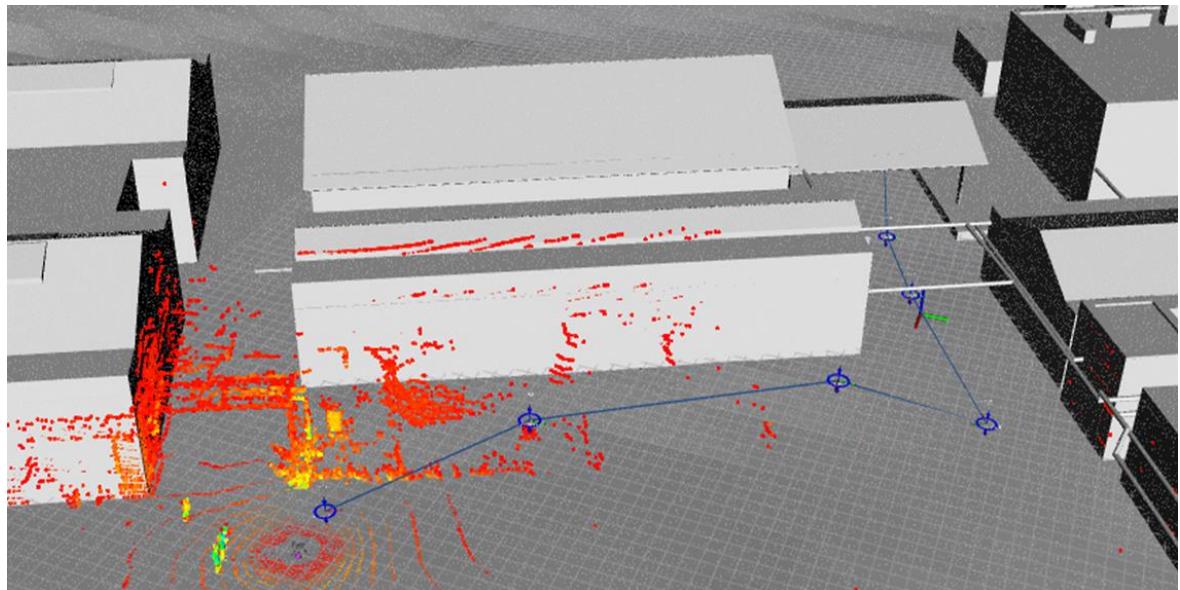


図 5-1 ドローン位置可視化システム「Rviz」

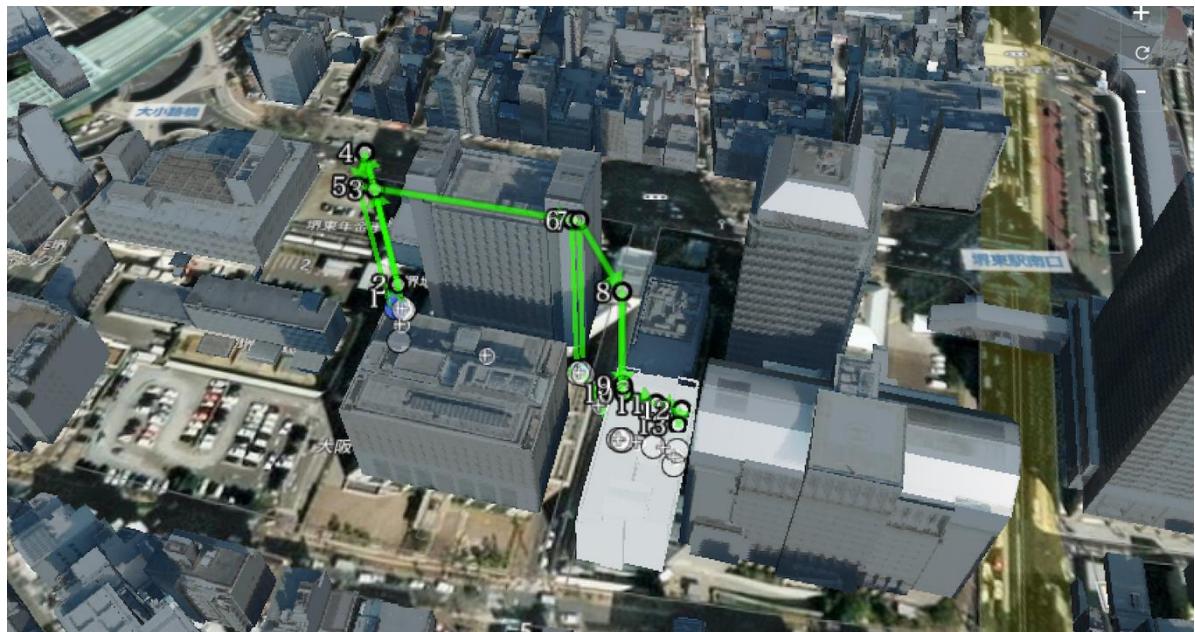


図 5-2 株式会社センシンロボティクス製の GCS アプリ「SENSYN CORE Pilot」

5-2. 検証方法

検証方法としては、被験者に対してデモンストレーションストレーションを取り入れたヒアリング・アンケートを実施した。

実施方法、環境については以下のとおりである。（ヒアリング・アンケートの項目については「7-4.ヒアリング・アンケートの詳細」で記載）

事業者向けヒアリングの実施方法（被験者の所属部署ごとに個別にヒアリング会を実施したが、内容は同様）

- 会場：被験者の自社会議室
- 機材：体験・デモンストレーション用に以下のスペックの社用PCを用意する。
 - CPU : 12th Gen Intel(R) Core(TM) i7 以上
 - メモリ : 16.0 GB 以上
 - OS : Windows 10 以降、Linux 18.04
 - 通信環境：各社で用意された社用Wi-Fi

5-3. 被験者

本ユースケースでは、既にドローンを自律飛行で使用している事業者のうち、飛行ルートの計画又は実際に飛行のオペレーションを行う部署の担当者をターゲットとしている。

本実証実験では、これらのユーザー該当する以下の方々にヒアリング・アンケートを行い、本システムの価値を検証する。

表 5-1 被験者リスト

分類	具体名称	部署	役職	担当業務	人数
事業者/ ユーザー	大手建機レンタル メーカー	道路機械事業部	課長	ドローン飛行オペレーション・ソフトウェアの運用	1名
		ICT 施工推進課			
	大手電機機器メー カー (ドローン製造)	産業機械事業部	営業主事	ドローン飛行オペレーション・ソフトウェアの運用	1名
		ロボティクス関連グ ループ事業戦略	役職なし	ドローン飛行計画・ソフ トウェアの運用	1名
		開発関連ソフト開発 担当	役職なし	ドローン飛行オペレーシ ョン・ソフトウェアの運 用	1名
株式会社 センシンロボティ クス	ソリューショングル ープ	プロジェクトマネ ージャー	プロジェクトマネ ージャー	ドローン飛行計画・ソフ トウェアの運用	1名
		フィールドサービス グループ	カスタマーサクセ ス	ドローン飛行計画・ソフ トウェアの運用	1名

5-4. ヒアリング・アンケートの詳細

5-4-1. アジェンダ・タイムテーブル

表 5-2 アジェンダ・タイムテーブル

No.	アジェンダ	所要時間
1	実験の目的を説明	5 分
2	作業手順のデモンストレーション、説明 (Rviz と SENSYN CORE Pilot)	5 分
3	操作の体験① (Rviz・昨年度ユースケースで開発した既存システム)	20 分
4	操作の体験② (SENSYN CORE Pilot・本年度ユースケースで改良したシステム)	20 分
5	操作感のヒアリング	5 分
6	アンケート回答 (Google フォームでの回答)	5 分

5-4-2. アジェンダの詳細

表 5-3 アジェンダの詳細

No	アジェンダ（再掲）	内容
1	実験の目的を説明	<ul style="list-style-type: none"> ● 本実証実験でアプローチする課題や背景の説明 ● 本実証実験の比較対象となる従来手法の説明 ● 本実証実験で用いるシステムの提供価値 ● システムの全体像の説明
2	作業手順のデモンストレーション、説明 (Rviz と SENSYN CORE Pilot)	<ul style="list-style-type: none"> ● Rviz (既存システム) <ul style="list-style-type: none"> ➢ Viewer PC 上で、昨年度ユースケースで開発したシステムを用いた CityGML を利用したドローン経路計画（ドローンルートの作成）機能のデモンストレーション、操作説明 ● SENSYN CORE Pilot (改良システム) <ul style="list-style-type: none"> ➢ SENSYN CORE Pilot 上で、本年度ユースケースで開発したシステムを用いた CityGML 及び BIM を利用したドローン経路計画（ドローンルートの作成）機能のデモンストレーション、操作説明
3	操作の体験① (Rviz)	<ul style="list-style-type: none"> ● 上記デモンストレーション内容を事業者が体験
4	操作の体験② (SENSYN CORE Pilot)	<ul style="list-style-type: none"> ● 同上
5	操作感のヒアリング	<ul style="list-style-type: none"> ● 質疑応答及びヒアリングを実施
6	アンケート回答	<ul style="list-style-type: none"> ● Google フォームでアンケート回答を依頼し、その場で回答（10分）、送信を依頼

5-4-3. 検証項目と評価方法

既存システムとの比較とユーザビリティ評価を検証項目とし、それぞれ定量・定性的に評価した。

表 5-4 検証項目と評価方法

検証観点	No	検証項目	定量評価	定性評価
1) 既存のシステム と利便性比較	1	計画したルートの保管が しやすいか	<ul style="list-style-type: none"> ● 対象ユーザーに既存の システム及び本年度開 発したシステムを体験 していただいた後、ア ンケートを実施 ● 選択肢は「とても不 満」を1、「とても満 足」を5とした5段階 で設定 	<ul style="list-style-type: none"> ● 操作の体験中及び体験 後にヒアリングを実施 ● アンケートの各設問に 自由記入欄を設定
	2	計画したルートの共有が しやすいか		
	3	ルート名の編集がしやす いか		
2) ユーザビリティ の評価	4	画面表示 (UI) が分かり やすいか	<ul style="list-style-type: none"> ● 回答を集計し、各選択 肢の選択率から評価 (各設問で、過半数の 「とても満足」、「満 足」の回答を得ること を目標とする) 	
	5	直感的な操作が可能か		

5-4-4. 実証実験の様子

- 会議室で本システムの操作手順を説明している様子



図 5-3 操作手順の説明

- 被験者がRvizでルート作成している様子

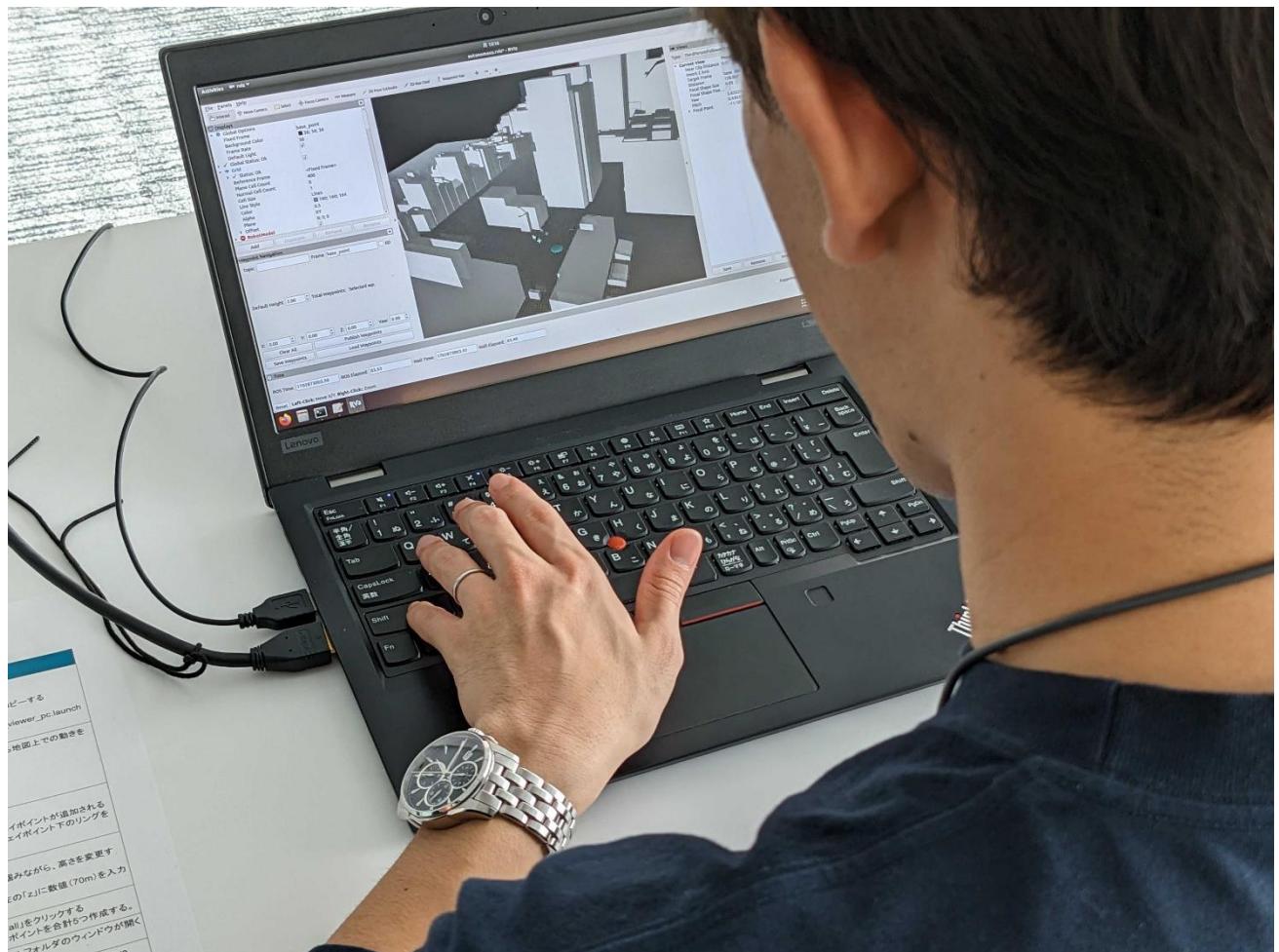


図 5-4 被験者による Rviz でのルート作成

● Rviz 画面

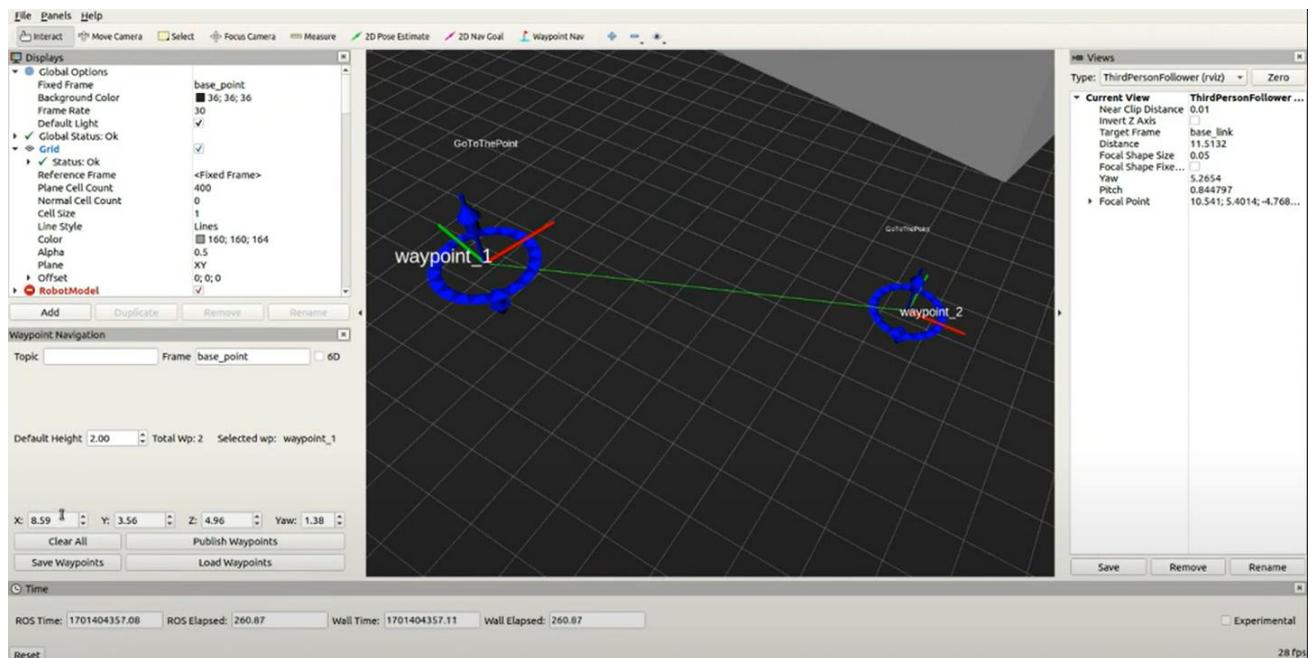


図 5-5 Rviz 画面

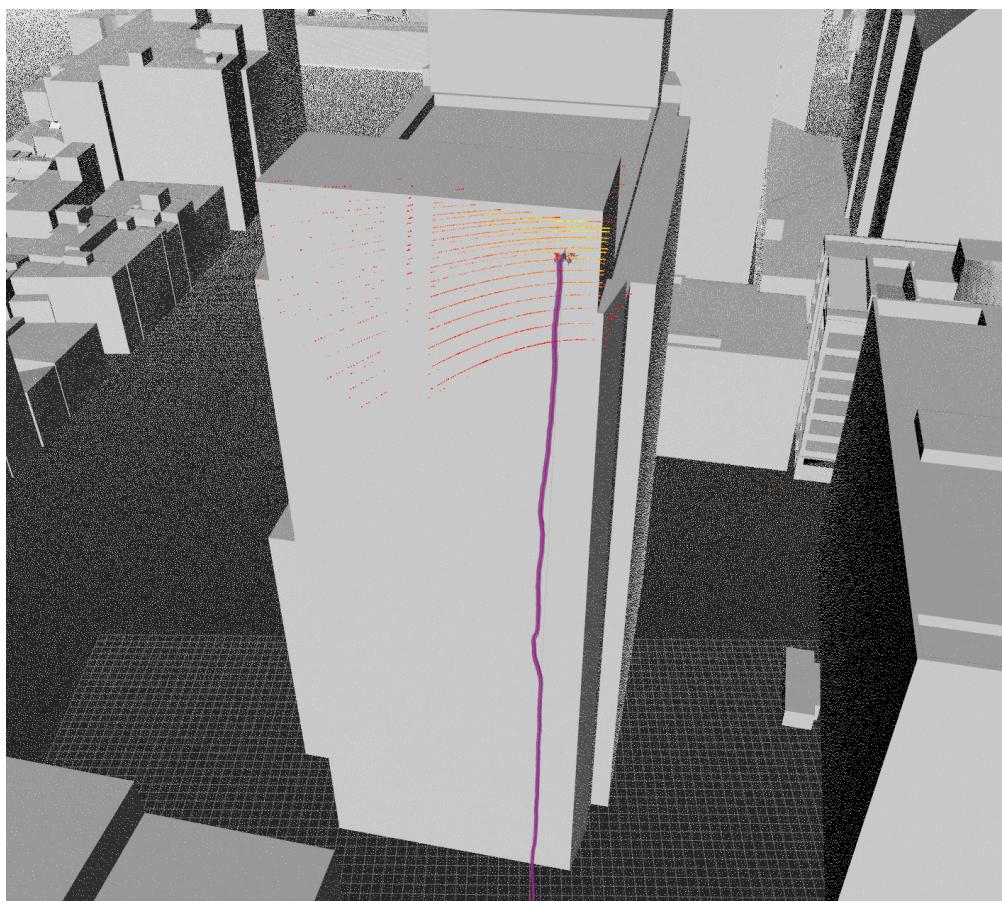


図 5-6 ドローン飛行中の Rviz 画面（ドローン後方から見た様子）

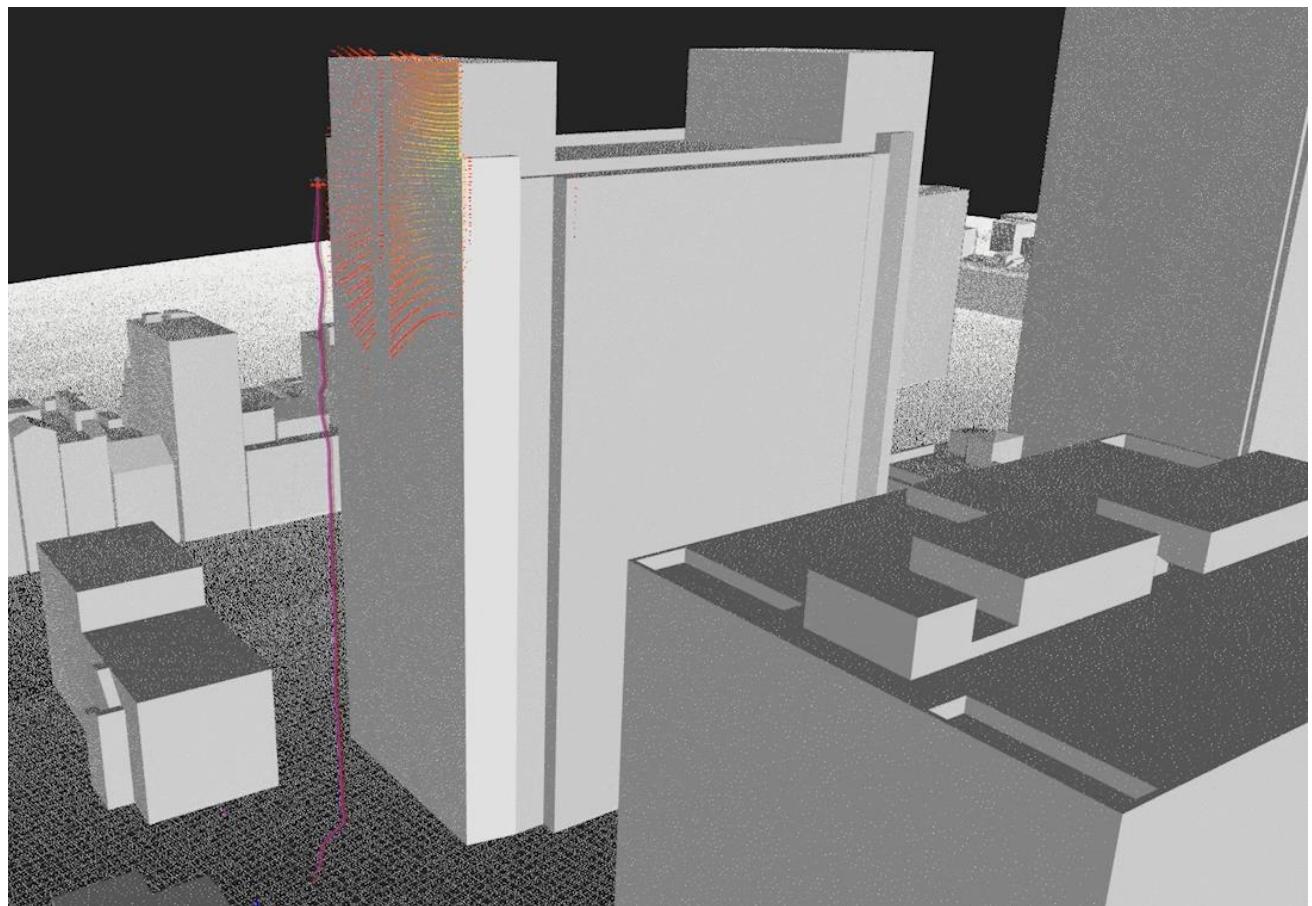


図 5-7 ドローン飛行中の Rviz 画面（ドローン側面方向から見た様子）

- 被験者がSENSYN CORE Pilotでルート作成している様子



図 5-8 被験者による SENSYN CORE Pilot でのルート作成

● SENSYN CORE Pilot 画面

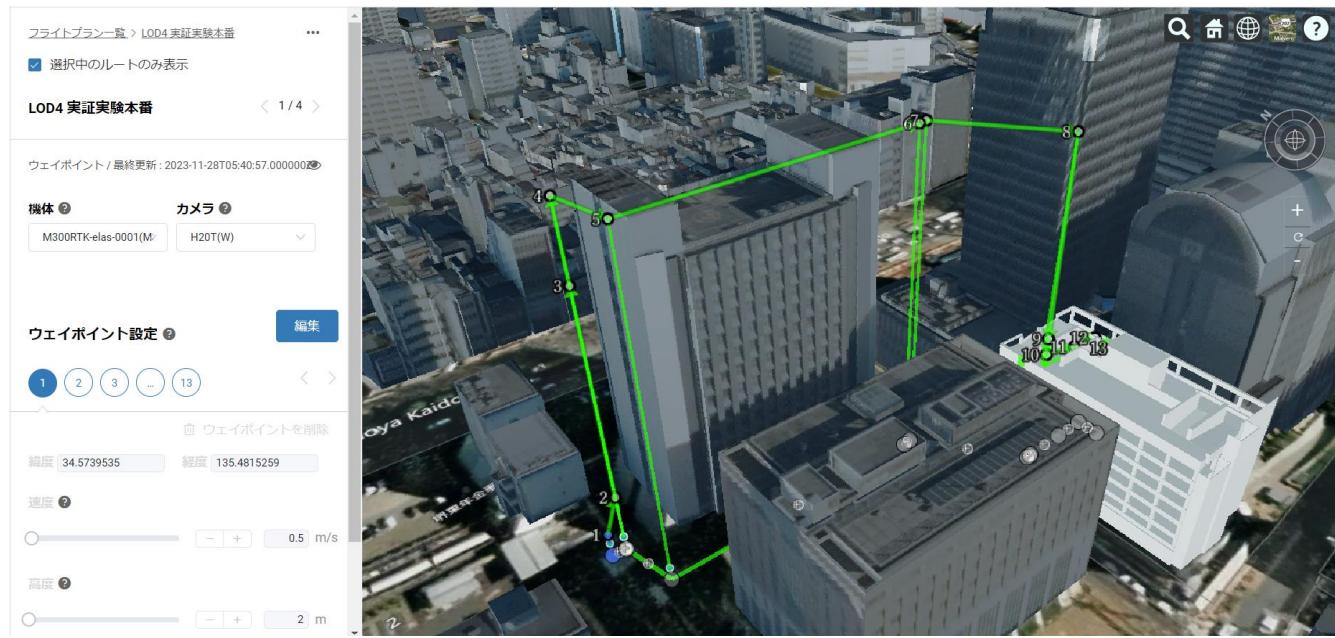


図 5-9 SENSYN CORE Pilot 画面

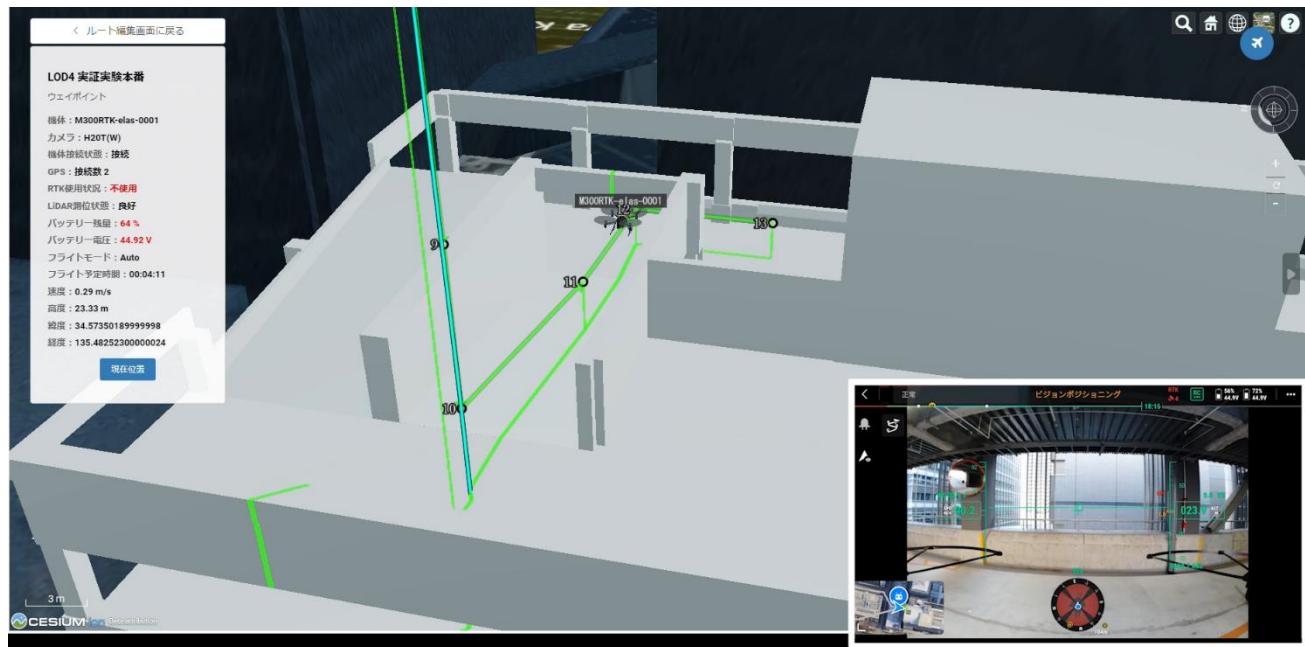


図 5-10 ドローン飛行中の SENSYN CORE Pilot 画面

- 被験者がアンケートに回答している様子

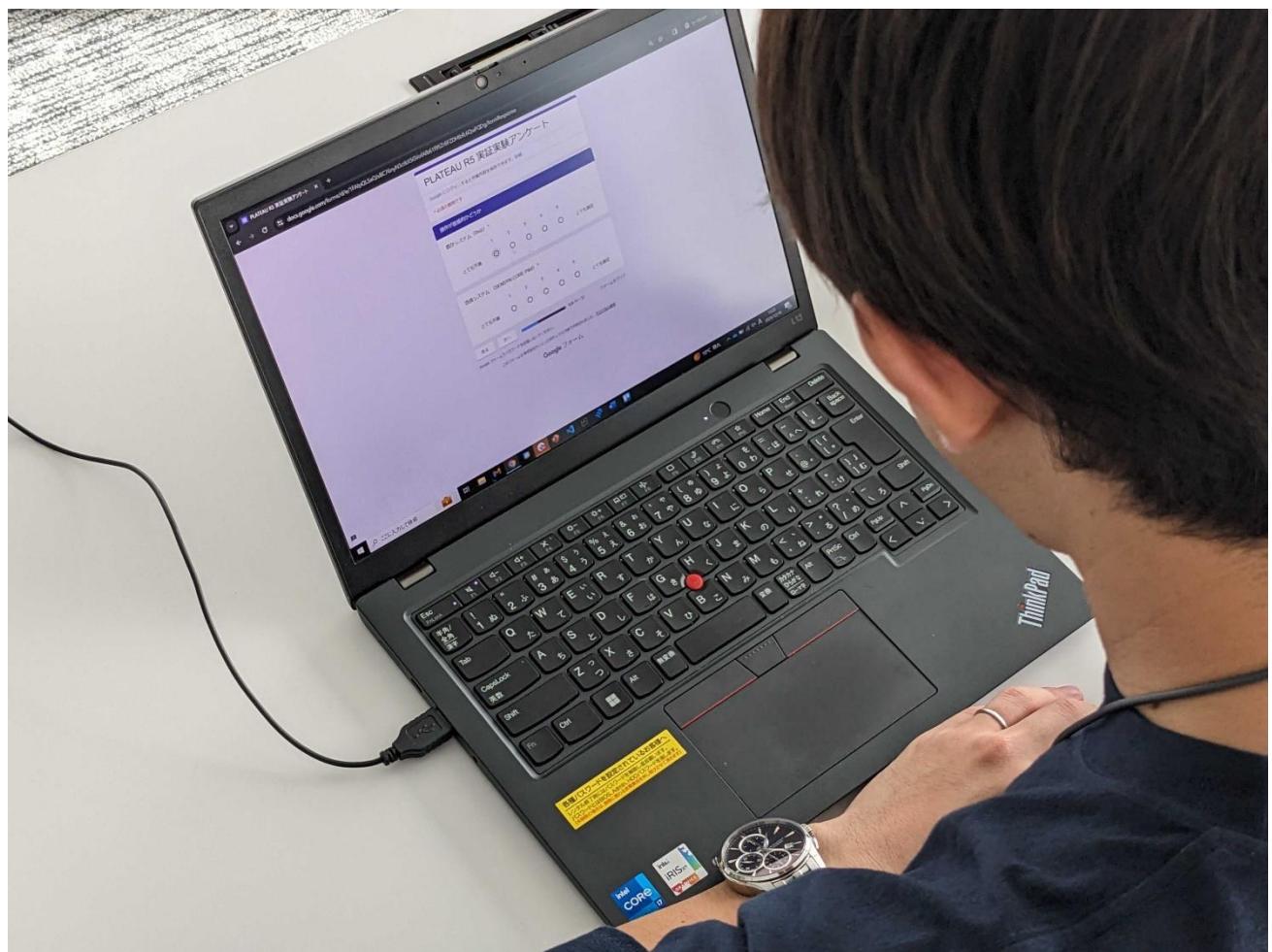


図 5-11 アンケート回答

5-5. 検証結果

ドローンの飛行ルートの計画及び、実際の飛行シミュレーションを実施する事業者に向けて、実際の業務を想定し必要十分な機能及びインターフェースを兼ね備えたシステムを提供し、全ての検証項目においてKPIで設定した目標以上の定量評価を得ることができた。また定性評価においても、一部課題はあるものの好意的なコメントを得た。

今回開発したインターフェースは「ルート作成」・「飛行ルートのアップロード」・「飛行開始指示」・「ドローンの位置表示」の4画面で制御を行う仕様となっている。その中デモンストレーション特に「ルート作成」については、3D都市モデルを活用し計画したルートを保存し、かつ他のオペレーターとの共有やルート名の編集などで意思疎通を図ることにも応用可能であることから、実業務上でオペレーターが課題と感じていた「システムを通じて共同で業務を行う」ことについても価値を提供することができた。

ユーザビリティに関しては、昨年度はLinux上のViewerを使用したシステムとなっていたため、システムへの理解及び知見がないと操作が困難な仕様になっていた。今年度は3D都市モデルを活用したドローンの経路計画について、ドローン関連の技術者ではなくてもオペレーション可能なGUIを含むシステムを構築し、UI/UX面でユーザビリティの高さは被験者より「直感的に操作できる」、「非常に使いやすい」など利便性の高さを示すフィードバックを得るなど、一定の評価を得ることができた。

しかし、ドローンの経路計画を作成するに当たり、都市部のビルが立ち並ぶ空間では、上空の意図した場所にウェイポイントを追加する際に少し慣れが必要だったため、より直感的な操作を実現するなどの課題点が明確になった。

- 1) 既存のシステムとの利便性比較

Q1 計画したルートが保存しやすいか

SENSYN CORE Pilotについて、ヒアリング対象者全員（6/6名）が「とても満足」か「満足」と回答した。一方Rvizでは、「とても満足」か「満足」と回答したヒアリング対象者はおらず、回答者の半数（3/6名）が「不満」又は「とても不満」と回答しており、SENSYN CORE Pilot上でシステム構築することによるルート保存の利便性の向上が明らかになった。

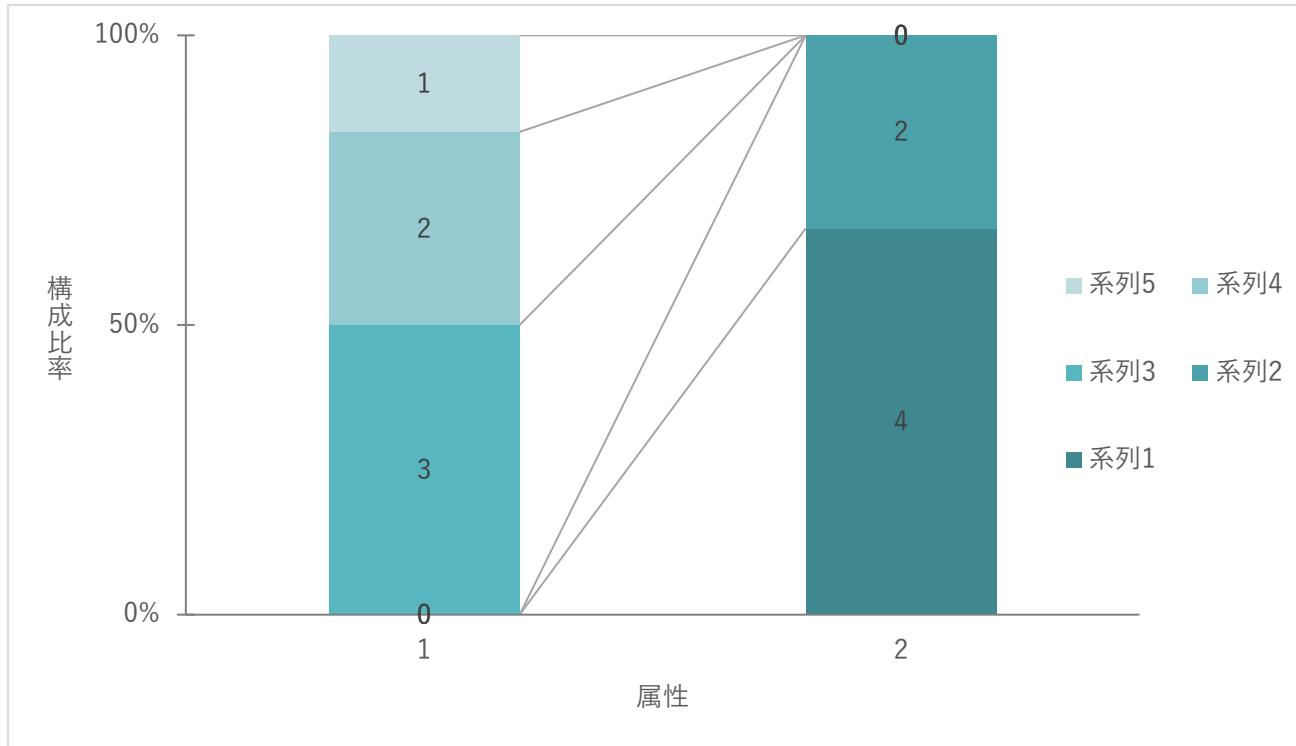


図 5-12 計画したルートが保存しやすいか

Q2 計画したルートが共有しやすいか

SENSYN CORE Pilotについて、ヒアリング対象者全員（6/6名）が「とても満足」か「満足」と回答。一方 Rvizでは、「とても満足」か「満足」と回答したヒアリング対象者はおらず、66%の回答者（4/6名）が「不満」又は「とても不満」と回答しており、SENSYN CORE Pilot上でシステム構築することによるルート共有の利便性の向上が明らかになった。

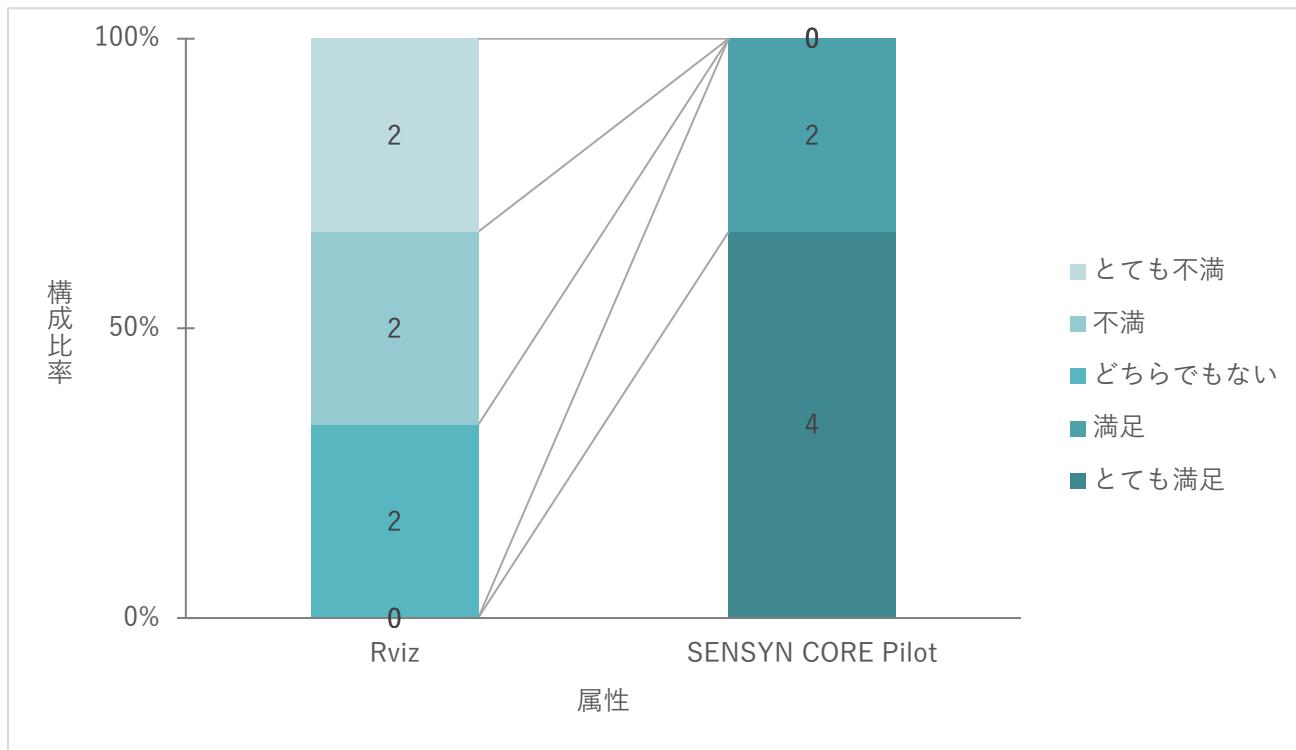


図 5-13 計画したルートが共有しやすいか

Q3 ルート名の編集がしやすいか

SENSYN CORE Pilotについて、全ヒアリング対象者のうち、66%の回答者（4/6名）が「とても満足」、「満足」と回答。一方 Rvizでは、「とても満足」、「満足」と回答した回答者は33%（2/6名）にとどまり、SENSYN CORE Pilot上でシステム構築することによるルート名編集の利便性の向上が明らかになった。

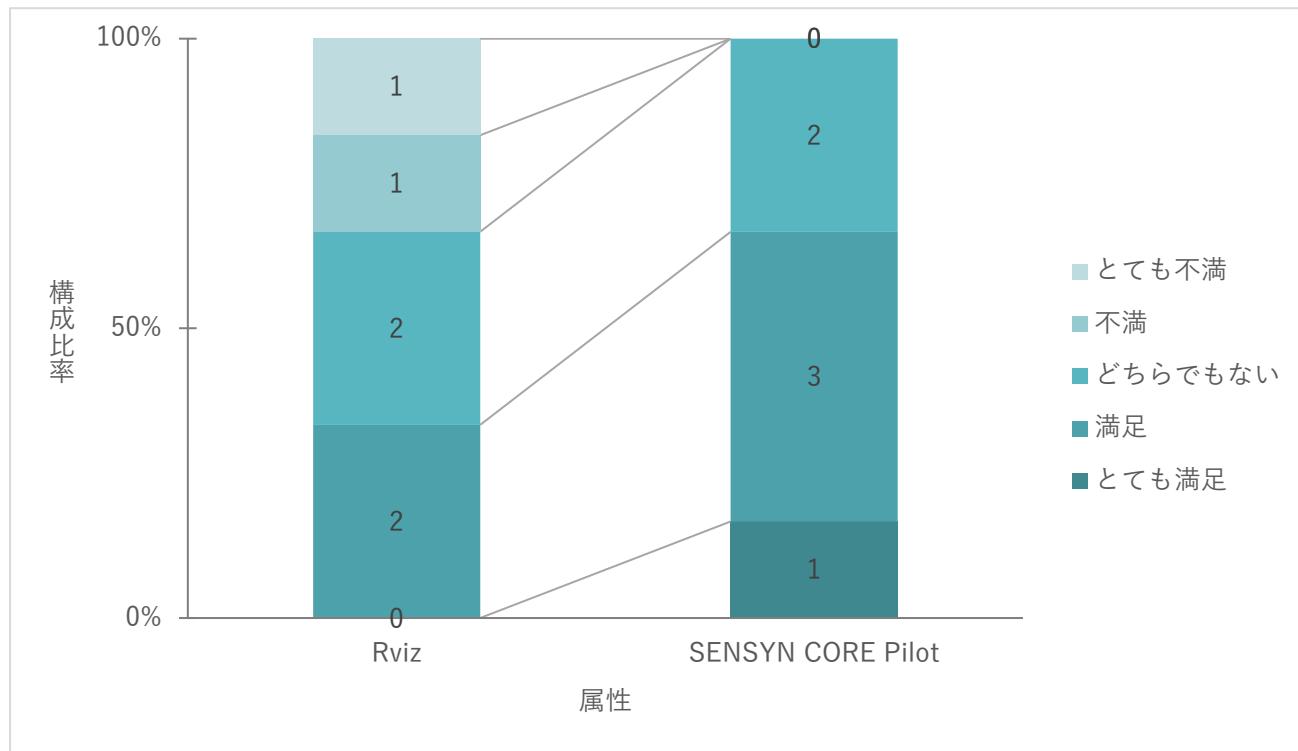


図 5-14 ルート名の編集がしやすいか

- 2) ユーザビリティ評価

Q4 画面表示 (UI) が分かり易いか

SENSYN CORE Pilotについて、ヒアリング対象者全員（6/6名）が「とても満足」、「満足」と回答。一方 Rvizでは、「とても満足」、「満足」と回答したヒアリング対象者はおらず、66%の回答者（4/6名）が「不満」、「とても不満」と回答しており、本年度追加改修したSENSYN CORE Pilotがユーザーフレンドリーな仕様であることがわかる。

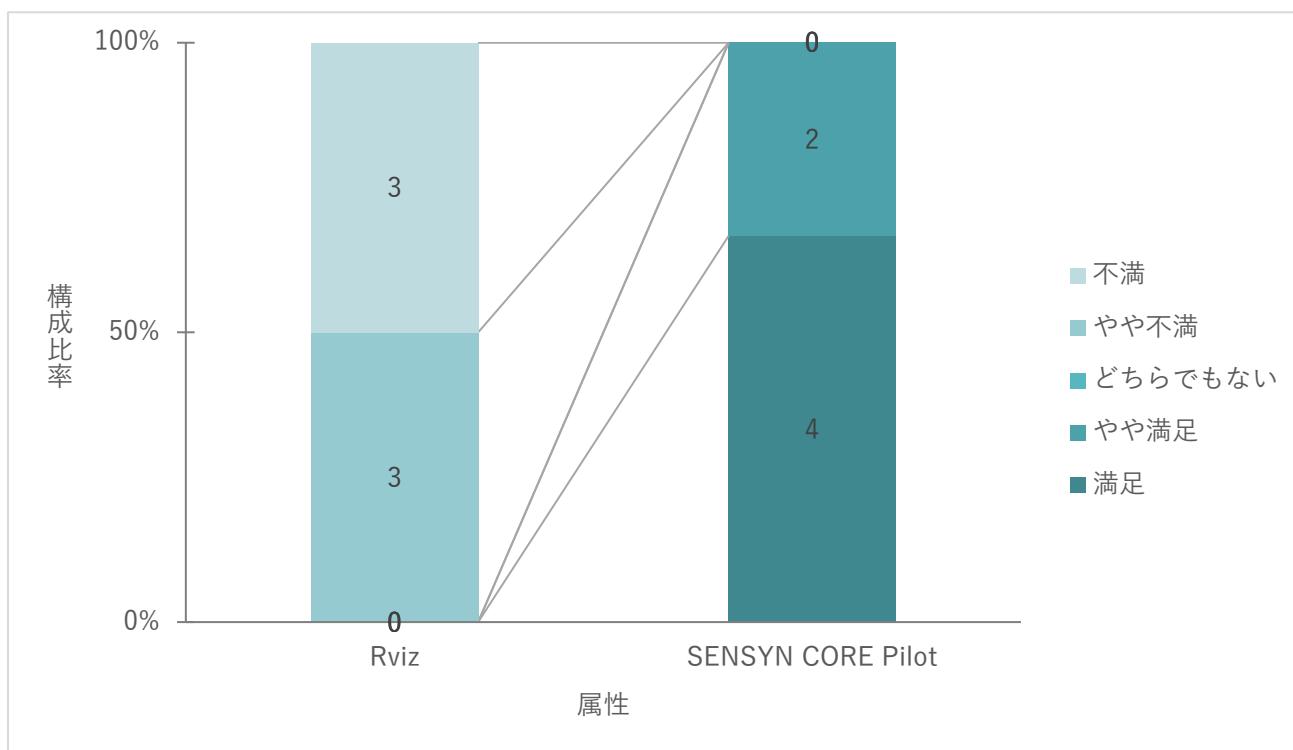


図 5-15 画面表示 (UI) が分かりやすいか

定性評価デモンストレーション、UIに対する好意的なフィードバックが寄せられた一方で、ウェイポイントの追加などの操作に慣れていないと設定が難しい項目があり、より簡便な操作を実現するための課題点が明確になった。

表 5-5 関連する定性コメント

No	定量調査の結果・示唆	関連する定性コメント
1	ルート作成 UI が分かりやすい	<ul style="list-style-type: none"> ● 地図上でルート作成できるため、位置関係の把握を含め使いやすい
2	ドローン向きの UI イメージの検討	<ul style="list-style-type: none"> ● より 3D 都市モデルが活用できるよう、ルート作成時にドローンの向きが UI 上に表示されるとわかりやすい

Q5 直感的な操作が可能か

SENSYN CORE Pilotについて、ヒアリング対象者のうち、83%の回答者（5/6名）が「とても満足」か「満足」と回答。一方 Rvizでは、「とても満足」、「満足」と回答したヒアリング対象者はおらず、83%の回答者（5/6名）が「不満」、「とても不満」と回答しており、本年度追加改修したSENSYN CORE Pilotの操作が直感的であることがわかる。

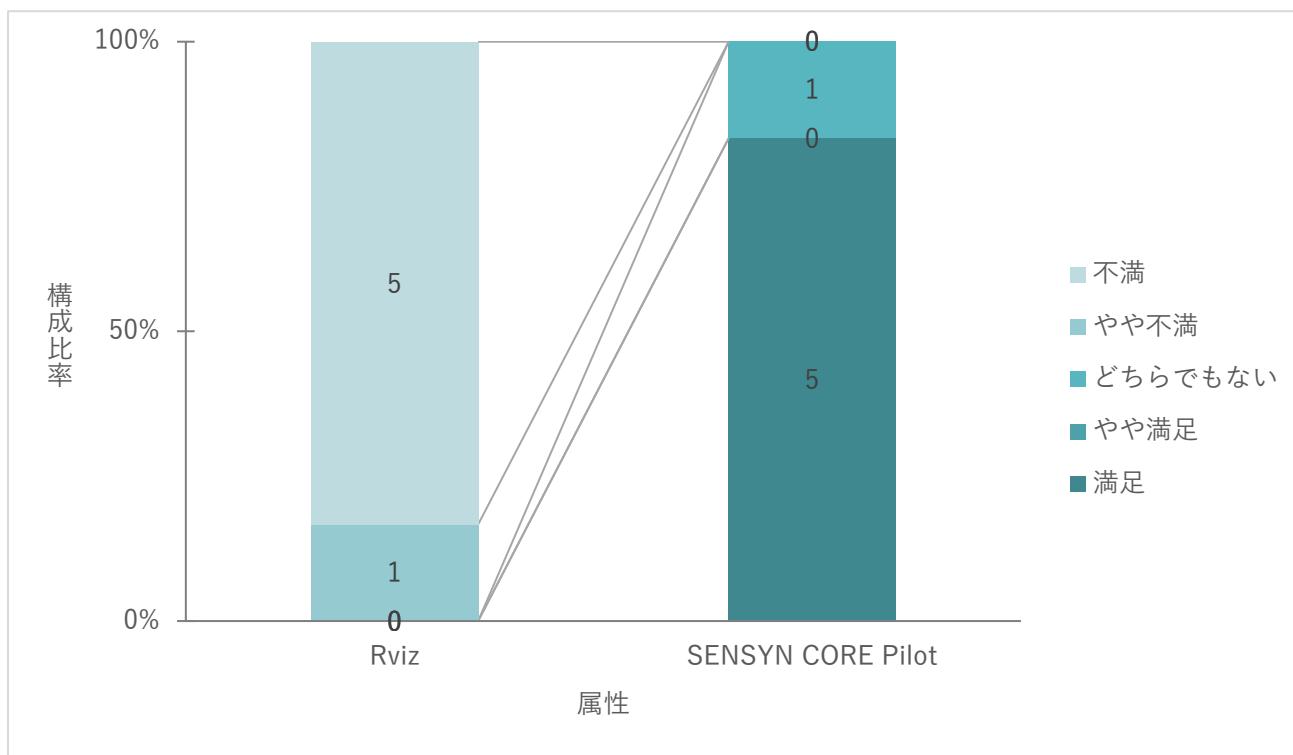


図 5-16 直感的な操作が可能か

定性評価デモンストレーション、「効率的な操作性」など操作性に対する好意的なフィードバックが寄せられた一方で、ウェイポイントの追加などの操作に慣れていないと設定が難しい項目があり、より簡便な操作を実現するための課題点が明確になった。

表 5-6 関連する定性コメント

No	定量調査の結果・示唆	関連する定性コメント
1	ウェイポイントが直感的に操作できる	<ul style="list-style-type: none"> ● ウェイポイントの設定が直感的かつ容易に操作することができる
2	都市部（建築物が密集するエリア）でのウェイポイント設定に改良の余地がある	<ul style="list-style-type: none"> ● ビルが立ち並ぶ空間では、上空の意図した場所にウェイポイントを追加する際に少し慣れが必要だった

6. 成果と課題

6-1. 本実証で得られた成果

6-1-1. 3D都市モデルの技術面での優位性

実証実験を通じて、以下のような3D都市モデルの技術面での優位性が示された。

表 6-1 3D都市モデルの技術面での優位性

大項目	小項目	3D都市モデルの技術面での優位性
システム・機能	複数視点のシームレスな切替え	<ul style="list-style-type: none"> 3D都市モデルにより俯瞰視点と1人称視点のシームレスな切替えが可能 Cesiumにもともと表示されている2D地図データを使ったドローンルート作成に加え、3D都市モデルを活用して構造物を考慮したルートの作成が可能
	LOD4利用による3Dオブジェクトの位置合わせ	<ul style="list-style-type: none"> BIMをLOD4に変換した3D都市モデルを用いることで、従来のBIMモデルを地図上に表示する前に行っていた位置合わせが不要となり、地図上での表示が簡素化
その他	ドローン用点群マップ作成の簡便化	<ul style="list-style-type: none"> 各事業体でドローン用点群マップを作成する場合、SLAMによる地図作成技術のコストの高さや精度の不足などの課題がある。一方で、3D都市モデルを利用することで事前のデータ取得や取得データをもとにした地図作成の必要がなく、低コスト・高精度な点群マップを簡便に作成可能

6-1-2. 3D都市モデルのビジネス面での優位性

表 6-2 3D都市モデルのビジネス面での優位性

大項目	小項目	3D都市モデルのビジネス面での優位性
サービスの提供価値向上	LiDAR用点群マップ作成の低コスト化及び汎用性の担保	<ul style="list-style-type: none"> 各事業体でドローン用点群マップを作成する場合、SLAMによる地図作成技術のコストの高さや精度の不足などの課題がある。一方で、3D都市モデルを利用することで事前のデータ取得や取得データを元にした地図作成の必要がなく、低コスト・高精度な点群マップを簡便に作成可能 また、3D都市モデルの利用で汎用性を担保できることで、より幅広いユーザーの獲得を期待できる
	シミュレーションの高度化	<ul style="list-style-type: none"> 3D都市モデルを活用することで飛行ルートシミュレーションが高度化し、従来の手法では困難であった建物が密集する都市部や高層ビル間の正確な飛行ルートシミュレーション

		<p>ヨンを実現</p> <ul style="list-style-type: none"> これにより、飛行ルートシミュレーション及びドローン自律飛行を含むサービスの提供価値向上を見込むことが可能
	ユーザー体験価値の向上	<ul style="list-style-type: none"> 3D都市モデルによる立体的でリアルな都市空間の再現が可能 10-1-1 「3D都市モデルの技術面での優位性」記載のとおり、2D地図データを使ったドローンルート作成に加え、3D都市モデルを活用して構造物を考慮したルートの作成が可能 これにより、ユーザーによるGCSアプリの体験価値向上を見込むことが可能
サービス開発期間・コストの削減	ドローン用点群マップ作成工数の短縮	<ul style="list-style-type: none"> 3D都市モデルが整備される以前は、ドローンでの屋内外をシームレスに移動する自律飛行のためには事前に手動飛行による点群マップの作成が必要不可欠であり、飛行のハードルは高かった 3D都市モデルの導入により、事前に点群マップ作成のためのデータ取得を行わずに自律飛行用の点群マップが作成できるようになり、従来よりも工数を削減可能
	オープンデータによる開発・運用コスト削減	<ul style="list-style-type: none"> 3D都市モデルは公的なオープンデータとして整備されていることから、ドローン事業者としては追加コストや許可手続きを経ることなくスピード感をもって開発を進めることができ可能
	整備範囲の広さによるビジネスの拡張性	<ul style="list-style-type: none"> ビジネスとしてのスケールを考えた場合、3D都市モデルの全国的な整備が進むことで特定の地域に限らない活用が可能 日本各地で飛行が求められるドローンの役務との親和性が高い

6-2. 実証実験で得られた課題と対応策

表 6-3 実証実験で得られた課題

大項目	小項目	実証実験で得られた課題	課題に対する対応策
システム (機能)	障害物・構造物回避性能向上	<ul style="list-style-type: none"> ルート作成時、障害物や構造物に景教されないルート作成には微妙な調整が必要だったため、ラフにルートを作っても構造物をうまく回避できる仕組みの強化により対応可能になる可能性が示唆された 	<ul style="list-style-type: none"> 障害物回避アルゴリズムの実装
システム (UI・UX)	建物内部の表示	<ul style="list-style-type: none"> 今回の実証では、屋外から屋内へ侵入した後の飛行ルート（屋内部分）作成のため、屋内が見えやすいようにBIM側で屋内飛行を実施するフロアの天井より上の部分を外したデータを変換し3D都市モデル（LOD4）を作成した。しかし、ユーザーインターフェース側で任意のフロアを選択し、そのフロアの天井より上の部分を非表示にするステップのシステム化は未対応 実装に向けて、運用に慣れたユーザーがより容易に操作できるようなUI・UX設計が必要 	<ul style="list-style-type: none"> 実装に当たって、これから今回の実証と同様のサービスが増えることを考えると、具体的なユースケースの調査を通じて適切なアプローチでの開発が必要 今後の事業化等を見据え、以下の改善が必要 <ul style="list-style-type: none"> ➤ 屋外/屋内モードを追加 ➤ 屋内モード画面において、3D都市モデル（LOD4）の任意のフロアを表示/非表示できるよう修正
		<ul style="list-style-type: none"> 本実証のヒアリング・アンケートにおいて、参加ユーザーから以下のフィードバックを得た <ul style="list-style-type: none"> ➤ (3D都市モデルの)ビルが立ち並ぶ空間では、上空の意図した場所にウェイポイントを追加する際に慣れが必要 	<ul style="list-style-type: none"> 今後の事業化等を見据え、以下の改善が必要 <ul style="list-style-type: none"> ➤ ウェイポイント設定画面において、3D都市モデルが立ち並ぶ地図上でより直感的にウェイポイントが編集できるよう修正

6-3. 今後の展望

今回の実証実験では、昨年度開発したLiDAR自己位置推定とGNSS測位に加えて、ビジュアルマイナーシャルオドメトリを利用した測位情報を加えることで、それぞれ不得意な領域を補い合う統合的な自己位置測位システムを構築した。

事前に設定したKPIを達成し、本システムは昨年度の実証で開発したシステムよりもロバスト性（頑強性）が向上したことが示され、都市部での3D都市モデルを用いたドローン利活用がより正確で安全に行えるようになったと言える。

また、本年度のユースケース開発では、ドローン飛行ルートの設定、飛行指示及び機体位置を可視化するシステムである「SENSYN CORE Pilot」により、上記の統合的な自己位置測位システムを搭載したドローンを飛行させることで、既存のユーザーインターフェースと比較して直感的に操作できる改修を行った。その結果、試行利用した各社の事業担当者からは総じて好意的なフィードバックが寄せられ、また各設問で、過半数の「とても満足」、「満足」の回答というKPIを達成した。これにより昨年度のユーザビリティ面での課題である「自律運行システムの操作に専門知識が必要であること」を解決したことが示され、今後の展開に向けて好感を得ることができた。一方で、UI/UXの面において、ビルが立ち並ぶ都市部エリアのルート設計では、ビル外壁からの距離と地上からの高さを3Dマップ上で意図した位置に指定することへの難しさが判明した。そのため、3Dマップ上での設定だけではなく、飛行する位置や高さなどは手入力可能なパラメーターを併用し、飛行する場所応じて簡易にルート設計が出来る仕様に改修するなど、更なる利便性向上に向けた改善を続けたいと考えている。

更なる実用化に向けては、GUIの更なる利便性向上はもとより、都市部における運用を想定し、開始地点から終着地点まで航空輸送と陸上輸送を併用した自律運搬モビリティサービスが想定され、自動運転車両等との連携も必要になる。それにより、例えば都市部における建設工事の資材運搬等による交通渋滞が発生する課題や人手不足などの問題に対して自律運航可能なドローンや無人搬送車両の活用による解決が期待できる。

将来的には、3D都市モデルとBIMを統合したマップを用いた陸空モビリティのオペレーションシステムを開発することで、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運行や、有人地帯でのドローンの正確で安全な飛行を目指す。

7. 用語集

A) アルファベット順

表 7-1 用語集（アルファベット順）

No.	用語	説明
1	BeiDou	<ul style="list-style-type: none"> ● 中国の衛星測位システム「北斗」 ● 英語名称は「COMPASS」
2	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ライトコントローラーに接続可能
3	Galileo	<ul style="list-style-type: none"> ● EUによる全地球航法衛星システム
4	GCS	<ul style="list-style-type: none"> ● Ground Control System の略 ● 無人航空機を人間が制御するためのシステム
5	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
6	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
7	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 軸方向の並進運動及び回転運動）を検出する
8	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
9	MAVLink	<ul style="list-style-type: none"> ● ドローンと GCS の間で機体の状態やルート情報などのデータを送受信するプロトコル。メッセージフォーマットを XML で定義することで C++ や Python のヘッダファイルを自動的に生成する。
10	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
11	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
12	ROS	<ul style="list-style-type: none"> ● 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

		<ul style="list-style-type: none"> 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、及びそれを包括するロボット開発プラットフォーム全体のこと
13	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
14	SENSYN CORE Pilot	<ul style="list-style-type: none"> 株式会社センシンロボティクス製の GCS アプリ
15	SLAM	<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
16	State machine	<ul style="list-style-type: none"> 複雑な命令を与え、制御するための手法 いくつかの「状態 (State)」と、ある状態から別の状態への「遷移 (Transition)」で構成されており、遷移が起こる条件を満たした際に状態が変わる仕組みになっている
17	Ubuntu	<ul style="list-style-type: none"> 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、全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

B) 五十音順

表 7-2 用語集（五十音順）

No.	用語	説明
1	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つなぐこととなる
2	ウェイポイントアクション	<ul style="list-style-type: none"> ウェイポイントとして設定したポイント上で、ドローンの方向転換などを行うアクション
3	基準座標系	<ul style="list-style-type: none"> ELAS 独自の三次元直交座標系
4	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 の超音波センサを使用する手法などが存在する
5	自己位置測位	<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 の超音波センサを使用する手法などが存在する
6	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
7	ビジュアルイナーシャルオドメトリ (VIO)	<ul style="list-style-type: none"> 連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU 情報を加味して移動量を推定する技術
8	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されていた用語で「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

以上

3D都市モデルとBIMを活用したモビリティ自律運航システム
(ドローン) v2.0
技術検証レポート

2024年3月 発行

委託者：国土交通省 都市局

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