



PLATEAU  
by MLIT

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



3D 都市モデル生成システムの開発 技術検証レポート

series  
No. 138

# 目次

---

---

1. 3D 都市モデル生成シミュレータの概要 .....	- 1 -
1-1. 課題認識 .....	- 1 -
1-1-1. 現状と課題 .....	- 1 -
1-1-2. 過年度の手法とその認識 .....	- 1 -
1-1-3. 既存業務フロー .....	- 3 -
1-2. 課題解決のアプローチ .....	- 4 -
1-3. 創出価値 .....	- 7 -
1-4. 想定事業機会 .....	- 8 -
2. 検証の概要 .....	- 9 -
2-1. 仮説 .....	- 9 -
2-2. 検証ポイント .....	- 10 -
2-3. 検証フロー .....	- 11 -
2-4. 実施体制 .....	- 11 -
2-5. 検証エリア .....	- 12 -
2-6. スケジュール .....	- 13 -
3. 開発スコープ .....	- 14 -
3-1. 概要 .....	- 14 -
3-2. 開発内容 .....	- 14 -
4. 実証システム .....	- 15 -
4-1. アーキテクチャ .....	- 15 -
4-1-1. システムアーキテクチャ .....	- 15 -
4-1-2. データアーキテクチャ .....	- 16 -
4-1-3. ハードウェアアーキテクチャ .....	- 17 -
4-2. システム機能 .....	- 20 -
4-2-1. システム機能一覧 .....	- 20 -
4-2-2. 利用したソフトウェア・ライブラリ .....	- 22 -
4-2-3. 開発機能の詳細要件 .....	- 26 -
4-3. アルゴリズム .....	- 54 -
4-3-1. 利用したアルゴリズム .....	- 54 -
4-3-2. 開発したアルゴリズム .....	- 63 -
4-4. データインタフェース .....	- 77 -
4-4-1. ファイル入力インタフェース .....	- 77 -
4-4-2. ファイル出力インタフェース .....	- 84 -
4-4-3. 内部連携インタフェース .....	- 84 -

4-4-4. 外部連携インタフェース .....	- 92 -
4-5. 検証に用いたデータ .....	- 93 -
4-5-1. 検証したデータの一覧 .....	- 93 -
4-5-2. 生成・変換するデータ .....	- 98 -
4-6. ユーザーインタフェース .....	- 99 -
4-6-1. 画面一覧 .....	- 99 -
4-6-2. 画面遷移図 .....	- 100 -
4-6-3. 各画面仕様詳細 .....	- 101 -
4-7. 実証システムの利用手順 .....	- 108 -
4-7-1. 実証システムの利用フロー .....	- 108 -
4-7-2. 各画面操作方法 .....	- 109 -
5. システムの非機能要件 .....	- 112 -
5-1. 社会実装に向けた非機能要件 .....	- 112 -
6. 品質 .....	- 114 -
6-1. 機能要件の品質担保 .....	- 114 -
6-2. 非機能要件の品質担保 .....	- 115 -
7. 実証技術の機能要件の検証 .....	- 116 -
7-1. 3D 都市モデル生成シミュレータの検証 .....	- 116 -
7-1-1. 検証目的 .....	- 116 -
7-1-2. KPI .....	- 116 -
7-1-3. 検証方法と検証シナリオ .....	- 117 -
7-1-4. 検証結果 .....	- 120 -
8. 実証技術の非機能要件の検証 .....	- 131 -
8-1. 検証目的 .....	- 131 -
8-2. KPI .....	- 131 -
8-2-1. 検証方法と検証シナリオ .....	- 131 -
8-2-2. 検証結果 .....	- 131 -
9. 成果と課題 .....	- 133 -
9-1. 本実証で得られた成果 .....	- 133 -
9-1-1. 3D 都市モデルの技術面での優位性 .....	- 133 -
9-1-2. 3D 都市モデルのビジネス面での優位性 .....	- 134 -
9-2. 実証実験で得られた課題と対応策 .....	- 135 -
9-3. 今後の展望 .....	- 137 -
10. 用語集 .....	- 138 -

# 1. 3D 都市モデル生成シミュレータの概要

## 1-1. 課題認識

### 1-1-1. 現状と課題

令和 2 年度に Project PLATEAU が始動して以降、全国において官民が連携し、3D 都市モデルの整備、活用及びオープンデータ化が着実に進展している。現在では多くの地方公共団体において 3D 都市モデルが整備されており、さまざまな政策領域での活用が見られる。また、民間においても、オープンデータとして公開された 3D 都市モデルを活用した多様なプロダクトやサービスが創出され、イノベーションの促進に寄与している。Project PLATEAU においては、国際標準規格である CityGML 2.0 を基に、我が国初となる都市デジタルツイン用の標準データモデルを「3D 都市モデル」として定義し、標準製品仕様を策定の上全国的な整備、利活用及びオープン化を推進してきた。令和 5 年度末時点では、全国約 200 都市において 3D 都市モデルの整備が進められており、今後より一層の拡大が見込まれている。

しかしながら、3D 都市モデルのデータ作成コストは、航空測量などの従来のデータ収集方法や目視による確認などの手作業に依存しており非常に高い。この現状は、3D 都市モデルの整備、利用及び更新を困難にしている。更に詳細度 (LOD) を高めることは、リアリスティックな 3D 都市モデルを生み出す上で重要だが、データ種類/量の増加と処理の複雑さなどの技術的な課題が伴う。

特に建築物の詳細情報を含む LOD3 の整備は技術的にも難しく課題となっている。この課題を解決するには、低コストの高解像度衛星画像や街路画像の利用が有用と考えられるが、これらの画像データから 3D 都市モデル作成に必要な詳細な情報を抽出し、3D モデルに統合する手法の開発が不可欠である。一方で、近年は生成 AI 技術、特に画像生成技術である GAN や Stable Diffusion が大きく進展している。これらは学習データが得やすい身近なオブジェクトなどにとどまっており、3D 都市モデルの生成にはまだまだ不十分である。このため、衛星画像や街路景観画像を基に、高 LOD かつ精緻な 3D 都市モデルを生成する、新しい生成 AI モデルの開発が喫緊の課題である。

### 1-1-2. 過年度の手法とその認識

前述の課題解決に向けて、2023 年度の「3D 都市モデル生成シミュレータ」では、生成 AI を活用して 3D 都市モデルをフットプリントデータ、衛星画像の入力とパラメータ設定のみによって自動生成できるモデル及び Web システムを開発し、仮想都市、実際の都市上のいずれかで、建築物、道路、植生などの 3D 都市モデルの自動生成を可能とした。

パラメータ入力のみで生成可能な仮想都市モデル生成シミュレータについては開発が完了した。他方、実際の都市の 3D モデルについては精度、生成時間、モデルの詳細度の三つの課題が残った。

具体的には、生成された建築物の形状にノイズが見られるなど生成モデルの精度に課題があった。これは生成アルゴリズムの精度が影響している可能性がある。生成時間に関しては、計算コストの高さ（200m グリッドで 1600s）に課題が残った。そのためには計算の並列化やデータサイズ削減等が必要である。また、モデルの詳細度は、開口部情報取得にも課題があり、LOD3 の実現には至らなかった。これらの課題の解決に向けて、生成アルゴリズムの改良、最適化及び開口部情報や屋根種別など詳細な情報を抽出する技術の開発が必要であった。

2024 年度の「3D 都市モデル生成シミュレータ」では、既存の 3D 都市モデルを学習データとして活用し、生成 AI を用いた自動生成モデル及び Web システムの開発を行った。本システムでは、フットプリントデータ及び衛星画像並びにパラメータ設定のみを入力とし、3D 都市モデルの自動生成を可能とするものである。

具体的には、衛星画像からフットプリントを抽出し、さらに、屋根形状タイプを判別可能なモデルを開発した上で、それらの情報をパラメータとして用いて 3D 都市モデルを生成する生成 AI モデルを構築した。これにより、フットプリントデータが未整備の地域や再開発が進む地域においても、3D 都市モデルの整備を可能とする仕組みを実現した。結果として、実都市における建築物、道路、植生等を含む LOD1～LOD3 の 3D 都市モデルの自動生成が実現された。

しかしながら、本システムには主に建築物の生成精度において課題が残った。具体的には、生成された建築物モデルのメッシュにノイズや欠損が見られ、幾何形状の正確性に難があった（図 1-1 参照）。これらの課題は、生成アルゴリズムの精度に起因する可能性が高い。また、現時点ではメッシュの品質評価は目視によって行っているが、都市スケールでの利用を前提とした場合、客観的かつ定量的な自動精度評価指標の開発が不可欠である。

さらに、LOD3 建築モデルに必要な開口部情報（窓、ドア等）については、ドライブレコーダーで撮影された沿道画像を用いた抽出の精度が不十分であり、詳細な建築物表現の再現には課題が残った。

以上の課題に対して、今後は以下の技術開発が求められる。

- 生成アルゴリズムの改良による形状精度の向上
- ノイズや欠損を定量的に評価するためのメッシュ精度評価指標の開発
- LOD3 開口部情報の抽出精度を高めるため、MMS（Mobile Mapping System）による沿道画像と三次元点群の併用技術の導入

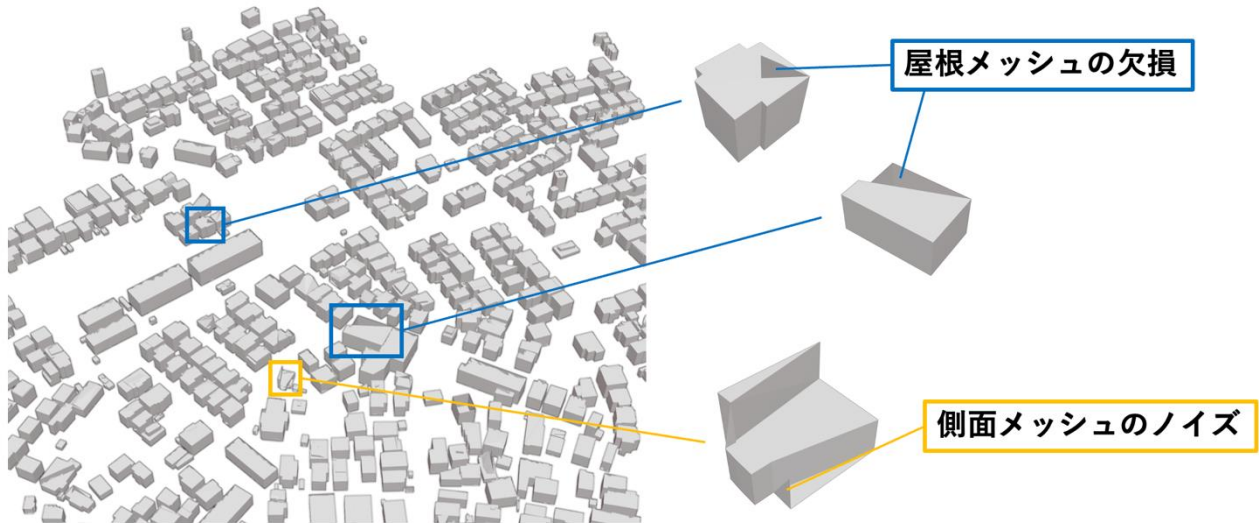


図 1-1 2024 年度の建築物の 3D 都市モデル生成におけるノイズ発生例

### 1-1-3. 既存業務フロー

一般的な 3D 都市モデル整備にかかわる業務フロー概要及びボトルネックとなる部分を以下に示す。

#### 従来の業務フロー

①データの収集	<ul style="list-style-type: none"> <li>対象地域の都市計画基礎調査等から得られる建物フットプリントデータと航空レーザ測量（DSM）を取得する</li> <li>データ取得コストが高い</li> <li>観測頻度が低い</li> </ul>
②データのマッチング	<ul style="list-style-type: none"> <li>建物フットプリントデータに高さ情報を付与するため、DSMと重ね合わせて両データを統合する必要がある</li> <li>高度な専門知識が必要</li> <li>開口部情報の取得に課題</li> </ul>
③3D都市モデルの生成	<ul style="list-style-type: none"> <li>建物フットプリントデータと高さ情報を基に3Dモデルを立ち上げる</li> </ul>
④出力・評価	<ul style="list-style-type: none"> <li>CityGML形式のファイルを出力し、可視化ツール上で定性的な品質評価のために可視化を行う必要がある。</li> </ul>

図 1-2 既存業務フロー

表 1-1 既存業務概要

実施項目	実施主体	業務概要
①データの収集	測量会社	このステップでは、3D 都市モデルを構築するための基本的なデータを取得する。通常、建築物のフットプリントデータや航空写真などが含まれる。収集されたデータは 3D 都市モデルの精度に影響を与えるため、高精度であり、モデルの基盤となる建築物の形状を正確に反映させる必要がある。航空写真は高精度な一方で、衛星画像と比べて観測頻度が低く、コストが高い課題がある
②データのマッチング	測量会社	収集した航空写真を建築物のフットプリントにマッチングする工程である。GIS（地理情報システム）技術を用いて、マッチングすることで各建築物に航空写真の高さ情報を付与する。また 3D 都市モデルの詳細化には、沿道画像など用いた開口部情報の取得などが課題である
③3D 都市モデルの整備	測量会社	マッチングされたデータを基に、実際に 3D 都市モデルを作成する。ここでは、3D モデリングソフトウェアなどを使用して、2D データから 3D 都市モデルを立ち上げる
④出力・評価	測量会社	CityGML 形式のファイルを出力し、可視化ツール上で定性的な品質評価のために可視化を行う必要がある

## 1-2. 課題解決のアプローチ

本年度は、2024 年度に開発した「3D 都市モデル生成シミュレータ」における課題を克服し、より高精度なモデル生成の実現を目指す。具体的には、従来達成していたジオメトリ精度（1/2500）に加え、メッシュにノイズがなく、目視においても不自然さのない建築モデルの生成、さらには、LOD3 レベルの詳細情報を再現可能な生成手法の確立に重点を置く。

モデル精度の改善に向けては、昨年度、PLATEAU の膨大な建築物モデルを生成 AI に直接学習させ、建築物のフットプリント、衛星画像等から AI により自動抽出した各種パラメータを入力とすることで、現実即した 3D モデル生成技術を開発した。しかし、その評価指標は主にジオメトリの精度指標である RMSE（Root Mean Square Error）に依拠しており、ノイズやメッシュ欠損など視覚的な品質は目視によって確認せざるを得ず、都市スケールでの客観的評価には限界があった。

この課題に対しては、生成結果に含まれるノイズの有無を定量的に評価する新たな指標として「Building Mesh Quality Index（BMQI）」を開発する。これにより、視覚的な品質を客観的かつ機械的に評価可能とし、ノイズが検出された場合には自動的に再生成を行う仕組みを導入することで、モデル精度の向上と自動検証の両立を図る。

さらに、本年度は、MMS（Mobile Mapping System）由来の沿道画像に加えて三次元点群データを活用し、

建築物の開口部情報（窓・ドア等）を AI により従来よりも高精度に抽出し、生成モデルへ統合することを旨とする。これにより、LOD2 レベルの都市モデル精度を向上させるとともに、LOD3 に必要な属性情報の抽出及び統合処理を実現し、現実に近い LOD3 モデルの生成を可能とする。

以上の技術的アプローチにより、フットプリントデータが整備されていない地域においても、衛星画像及び MMS データを入力とすることで、精度と汎用性を兼ね備えた 3D 都市モデルの生成が可能となる。

本システム導入後に期待される 3D 都市モデル整備の各工程における改善点を以下に示す。

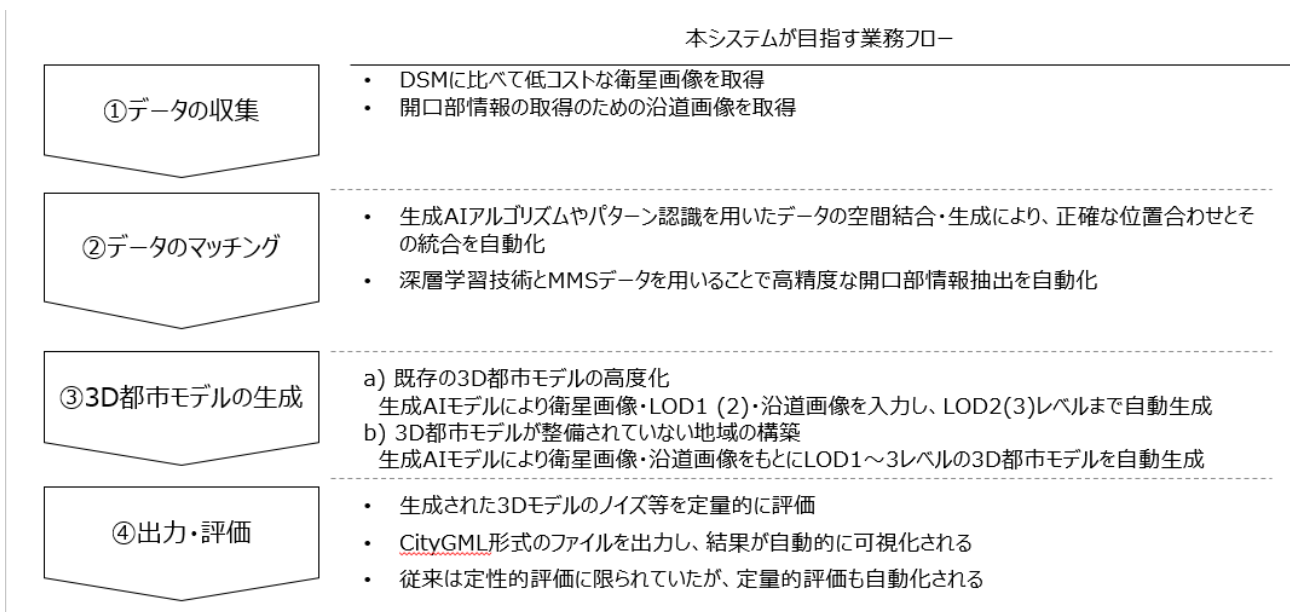


図 1-3 改善後の業務フロー

表 1-2 本システム導入による改善点

実施項目	実施主体	本システム導入による改善点
①データの収集	測量会社	本システムを導入することで、航空測量といった従来の高コストなデータ収集手法に依存する必要がなくなる。ユーザーは、低コストな衛星画像や公開されている建築物のフットプリントデータを利用して、効率的かつ経済的に必要なデータを収集できるようになる
②データのマッチング	測量会社	データのマッチングの工程では、プロセスを人工知能により自動化する。このシステムでは、機械学習アルゴリズムやパターン認識技術を活用して正確な位置合わせとデータの整合性を自動化する。これにより、データのマッチングにかかる時間と労力が大幅に削減される
③3D 都市モデルの整備	測量会社	3D 都市モデルの整備工程では、生成 AI を用いた本システムによって複雑なモデリング作業が自動化され、より迅速に 3D 都市モデルを生成できるようになる。ユーザーは、細かい手作業を行うことなく、データ入力とパラメータの調整だけで実際の都市の多様なシナリオの都市モデルを構築することができる
④出力・評価	測量会社	最終的に、PLATEAU が整備されていないエリアを含む 3D 都市モデルがオープンソースの形式で提供されることにより、従来と比べて低コストで、誰もがアクセスしやすい形で 3D 都市モデルを利用できるようになると期待される。これにより、日本全国の 3D 都市モデルの利用が促進される。また開発した評価指標により、3D モデルのメッシュ欠損などを目視に頼らず定量的に評価できるようになると共に従来よりも高品質なデータ提供が実現可能となる

### 1-3. 創出価値

3D 都市モデル生成ツールの主要な創出価値は、生成 AI を活用し、低コストで精緻かつ LOD0-3 のマルチ LOD に対応した 3D 都市モデリングを可能にすることである。

具体的には、第一に既存の 3D 都市モデルの高度化が挙げられる。ここでは、建築物フットプリントデータ、衛星画像、沿道画像、三次元点群を組み合わせることで、より精緻な都市モデルとなるよう LOD の向上を図り、可視化する。これにより、既存の 3D 都市モデルはより詳細で現実に即したモデルへと高度化され、多様な地物の 3D 都市モデル化や可視化のニーズに応えることができる。

第二に、二次元の建築物のフットプリントデータのみを持つ地域における 3D 都市モデル化がある。このニーズに対応するため、開発ツールでは衛星画像と沿道画像を基に、建築物や植生などの地物の 3D モデルを自動生成する。衛星画像や沿道画像を利用することで詳細な情報が取得できるだけでなく、都市の空間分布を反映させることが可能になる。これにより、リアルな都市の 3D モデルを低コストで構築し、実際の都市の 3D 都市モデルを整備できるだけでなく、シミュレーションにより開発予定の 3D 都市モデルの作成及び可視化ができる。特に 3D 都市モデルを整備する初期段階での検討において、低コストで整備可能なこのアプローチは極めて有効である。

## 1-4. 想定事業機会

表 1-3 想定事業機会

項目	内容
利用者	<ul style="list-style-type: none"> <li>● 測量会社</li> </ul>
サービス仮説	<ul style="list-style-type: none"> <li>● 3D 都市モデルのない地域での 3D 都市モデル整備</li> <li>● 建築物、道路、植生、都市設備、地形を網羅</li> <li>● 既存の 3D 都市モデルの高度化（LOD 向上）</li> <li>● データ取得、整備費用削減</li> <li>● さまざまなパラメータ設定によるシミュレーションを行うことで 3D 都市モデルシナリオ生成（再開発、復興等のまちづくり計画支援）</li> </ul>
提供価値	<ul style="list-style-type: none"> <li>● 3D 都市モデル整備、シミュレーション、可視化できるサービス（Web アプリや専用ソフトウェアの提供） <ul style="list-style-type: none"> <li>➢ 3D 都市モデルを生成、シミュレーション、可視化できる機能を使用できるパッケージの販売や定額サービス（サブスクリプション）を展開</li> </ul> </li> <li>● 3D 都市モデルなどの基盤情報の整備 <ul style="list-style-type: none"> <li>➢ 衛星画像と沿道画像を用いた低コストでの 3D 都市モデルの作成業務</li> </ul> </li> </ul>

## 2. 検証の概要

### 2-1. 仮説

#### 【生成 AI による 3D 都市モデル整備の自動化によるコスト、整備時間の削減】

- 生成 AI を用いることで、衛星画像や沿道画像といった低コストで取得可能な観測データを入力とする 3D 都市モデル生成シミュレータを開発し、高精度な LOD3 までの 3D 都市モデル整備の自動化を実現できる。これにより、従来に比べて低コストかつ高精度な 3D 都市モデルの整備が自動で可能となり、整備及び更新に係るコストの大幅な軽減が期待される

#### 【衛星画像による LOD1-2 生成の自動化】

- 従来の航空測量に依存しない衛星画像から高精度な LOD1-2 モデルを生成可能になることで、3D 都市モデル整備コストを削減する

#### 【MMS データ統合による LOD3 生成の自動化】

- LOD2 建築物モデルと MMS データの統合による、LOD3 の自動整備が可能な手法を確立することにより、現在未整備となっている地域、物に対する 3D 都市モデルの新規整備を促進するとともに、既存の 3D 都市モデルの LOD3 化にも資するものとなる

## 2-2. 検証ポイント

開発した 3D 都市モデル生成モデルとツールに対する検証を行うことによって、以下の観点について評価を行う。

- 3D 都市モデルの生成モデルの自動化による計算時間の削減  
検証エリアにおける建築物モデルを生成するのに必要な時間を算出し昨年度のシミュレータでの生成時間と比較する。これにより、本年度開発したアルゴリズムの計算時間における優位性を評価する
- 生成 AI を活用した 3D 都市モデル生成手法確立に向けたモデル精度  
生成 AI を活用した 3D 都市モデルの精度を、以下の 3 点から評価する。
  1. ジオメトリ精度：PLATEAU のデータを正解データとし、生成結果の形状を水平方向及び鉛直方向の誤差（RMSE）により評価する
  2. ノイズ評価：開発した BMQI（Building Mesh Quality Index）により、三角メッシュのズレや凹凸を定量的に評価する
  3. 開口部抽出精度：MMS と AI を用いた開口部抽出精度を評価するこれらの指標を通じて、航空測量を用いずとも、衛星画像及び沿道画像から高精度なモデルが生成可能であることを検証する
- 生成 AI を活用したデータ品質の担保  
出力された 3D 都市モデルのファイルが、CityGML（PLATEAU 製品仕様書第 4.0 版）仕様に準拠して正しく出力されているかを確認する。また、出力データのファイルサイズが既存の PLATEAU 製品と同等水準にあることを検証する。これにより、生成 AI を活用したモデル生成においてもデータ品質が担保されていることを確認する

上記については「7. 実証技術の機能要件の検証」で検証結果を記載

## 2-3. 検証フロー

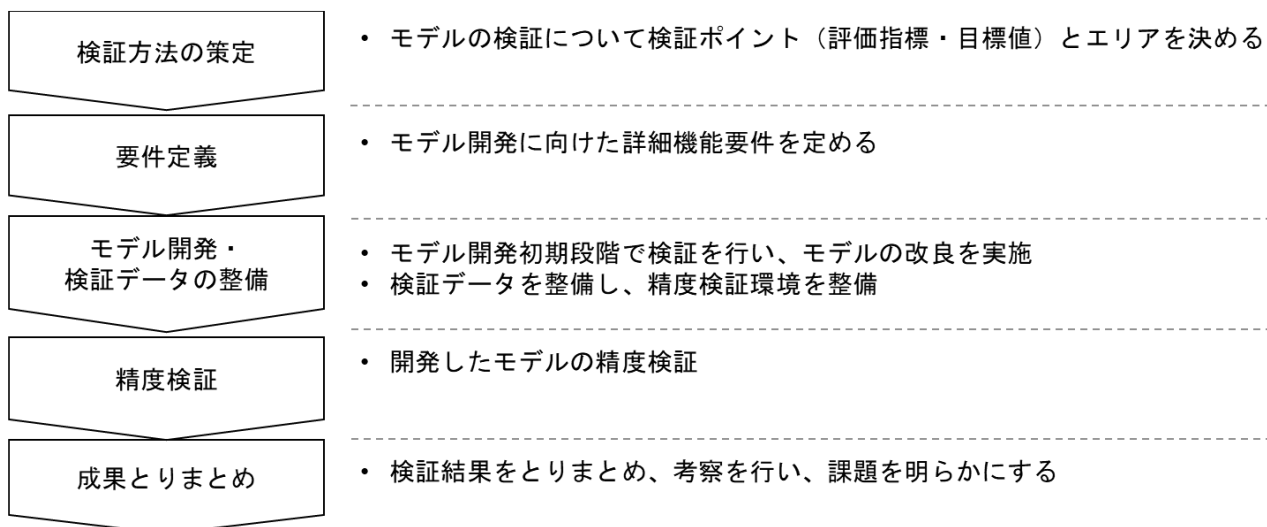


図 2-1 実証フロー

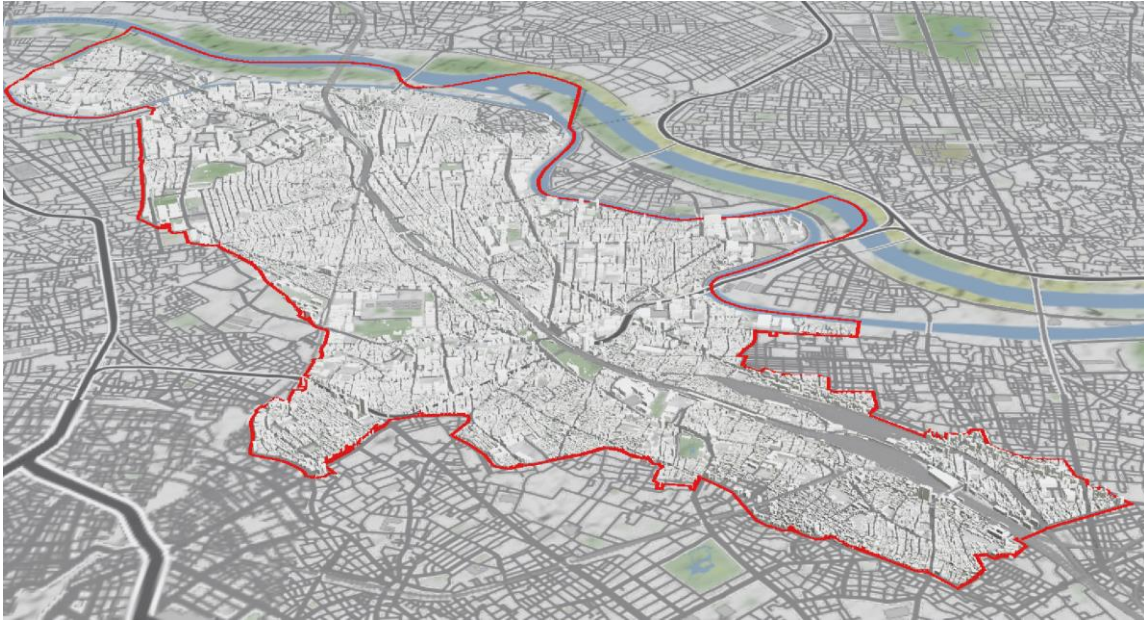
## 2-4. 実施体制

表 2-1 実施体制

役割	主体	詳細
全体管理	国交省 都市局	プロジェクト全体ディレクション
	社会基盤情報流通推進協議会	プロジェクトマネジメント
実施事業者	東京大学	技術開発
実施協力	北川 悦司	開口部抽出モデルの開発

## 2-5. 検証エリア

表 2-2 検証エリア

項目	内容
実証地	東京都北区
面積	10 km <sup>2</sup>
マップ (対象エリアは赤枠内)	

## 2-6. スケジュール

表 2-3 スケジュール

実施事項	2025 年									2026 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. 実施計画			↔									
2. 要件定義				↔								
3. モデル開発・検証データの整備				↔								
4. 精度検証									↔			
5. 成果とりまとめ										↔		

## 3. 開発スコープ

### 3-1. 概要

「3D 都市モデル生成ツール」は、衛星画像、沿道画像、建築物のフットプリントデータに基づき、3D 都市モデル（建築物、道路、植生、都市設備、地形）を自動生成するシミュレータであり、Web システムとして構築している。本ツールは、生成 AI を活用することで、既存の都市データを基にした 3D 都市モデルの整備に加え、詳細レベル（LOD）の向上機能を備えている。

具体的には、3D 都市モデルを学習させた生成 AI 技術を活用し、MMS データ、衛星画像、フットプリントデータや LOD1 レベルの 3D 都市モデルを入力することで、より詳細な 3D 都市モデルを自動的に出力可能とする。

また、生成された 3D 都市モデルは、PLATEAU 製品仕様書第 4.0 版に準拠した CityGML 形式で出力されるとともに、ツール上での視覚的な確認・表示も可能である。

### 3-2. 開発内容

「4. 実証システム」で記載

## 4. 実証システム

### 4-1. アーキテクチャ

#### 4-1-1. システムアーキテクチャ

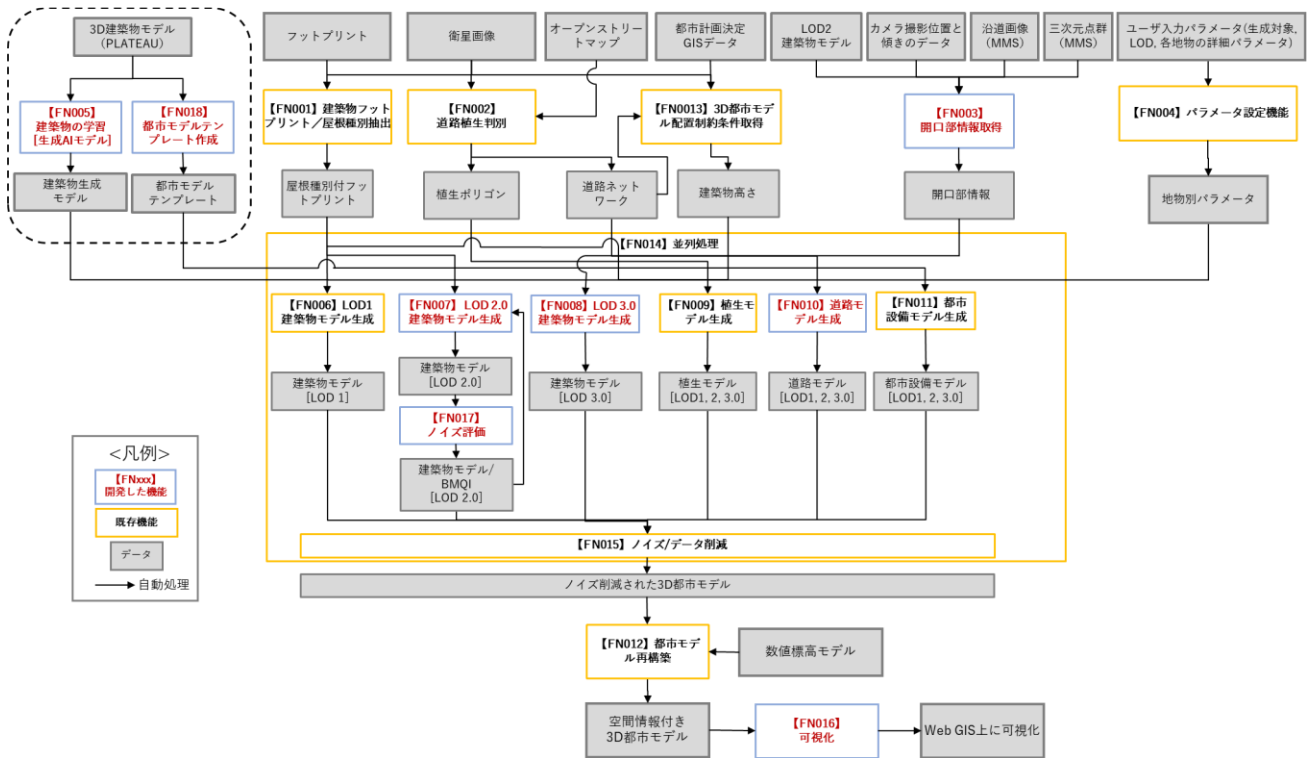


図 4-1 システムアーキテクチャ

4-1-2. データアーキテクチャ

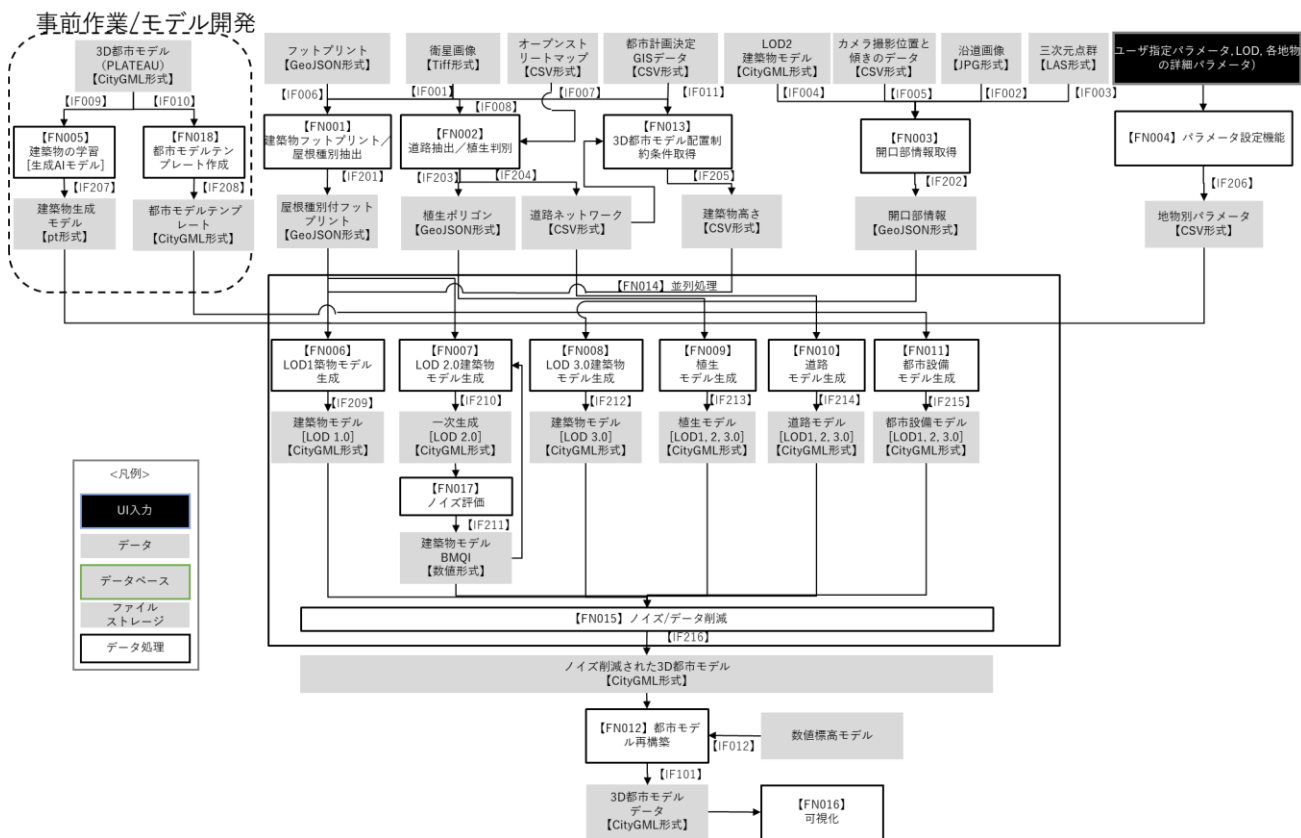


図 4-2 データアーキテクチャ

### 4-1-3. ハードウェアアーキテクチャ

#### 4-1-3-1. 利用するハードウェア一覧

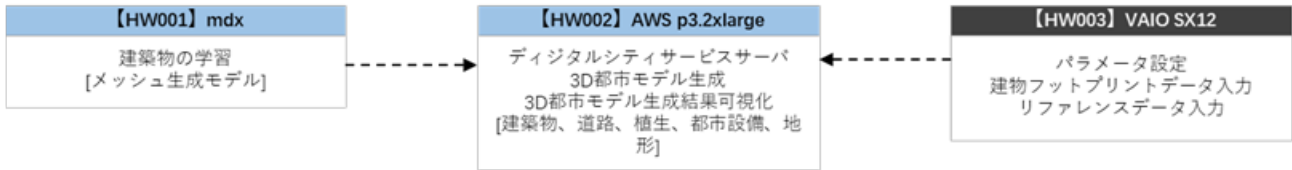


図 4-3 ハードウェアアーキテクチャ

表 4-1 利用するハードウェア一覧

ID	種別	品番	用途
HW001	HPC	mdx	● 生成モデルの学習に用いる高性能 GPU 計算機
HW002	AWS EC2	p3.2xlarge	● 学習済みモデル実装や 3D 都市モデルの生成と可視化
HW003	ノートパソコン	VAIO SX12	● 利用者がツールを利用する PC

#### 4-1-3-2. 利用するハードウェア詳細

1) 【HW001】 mdx<sup>1</sup>

- 選定理由
  - 大規模ストレージと高性能な GPU を多数枚搭載しているため、大規模な深層学習の計算に適しており、国内トップクラスの計算環境が利用可能
- 仕様・スペック
  - CPU（ノード当たりの個数）：Intel Xeon Platinum 8368 Processor（38core, 2.4GHz）× 2 個
  - メモリ：512 GiB
  - GPU（ノード当たりの個数）：NVIDIA A100（GPU メモリ 40GB）× 8 個
  - 総理論演算性能(倍精度演算)：6.4PFLOPS
  - 総理論演算性能(単精度演算)：6.7PFLOPS
  - 総理論演算性能(半精度演算)：100.7PFLOPS

- イメージ

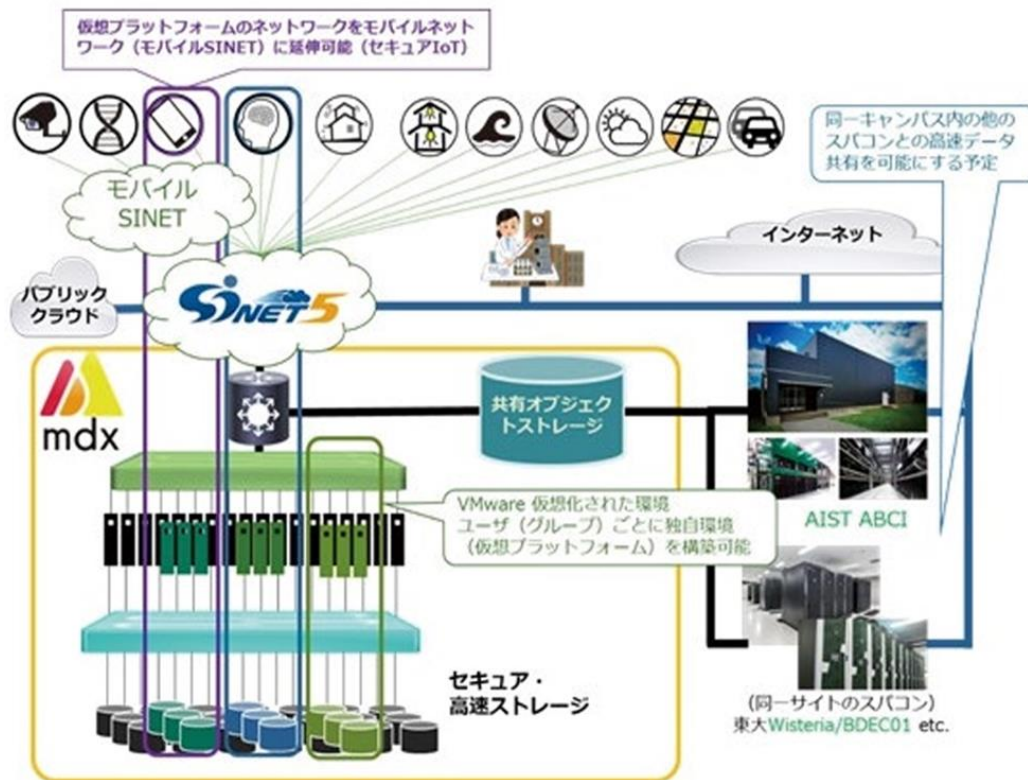


図 4-4 mdx<sup>1</sup>

2) 【HW002】 AWS EC2

- 選定理由
  - デジタルシティサービスサーバーとして、学習済みモデル実装や 3D 都市モデルの生成と可視化に使用されている高性能計算設備
- 仕様・スペック
  - CPU : 2.7 GHz (turbo) Intel® Xeon E5-2686 v4
  - GPU : NVIDIA Tesla V100
  - メモリ : 61GB
  - ストレージ : 1TB
  - OS : Linux 20.04

<sup>1</sup> 公式 HP : <https://mdx.jp/>

<sup>2</sup> 公式 HP より抜粋 : <https://aws.amazon.com/jp/ec2/>

- イメージ

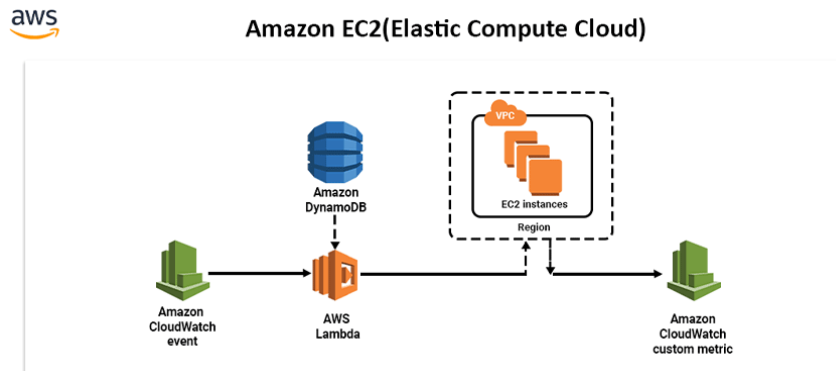


図 4-5 AWS EC2<sup>2</sup>

3) 【HW003】 ノートパソコン VAIO SX12

- 選定理由
  - アプリを利用するのに十分なスペックである
- 仕様・スペック
  - CPU : Intel® Core™ i7-1360P
  - メモリ : 32GB
  - ストレージ : 1TB
  - OS : Windows 11 Home 64 ビット
- イメージ



図 4-6 ノートパソコン VAIO SX12<sup>3</sup>

<sup>1</sup> 公式 HP : <https://mdx.jp/>

<sup>2</sup> 公式 HP より抜粋 : <https://aws.amazon.com/jp/ec2/>

<sup>3</sup> 公式 HP より抜粋 : <https://store.vaio.com/shop/pages/sx126g.aspx>

## 4-2. システム機能

## 4-2-1. システム機能一覧

表 4-2 機能一覧

※朱文字：新規開発・既存改修

ソフトウェア	ID	機能名	機能説明
衛星画像・沿道画像 リファレンス機能	FN001	建築物フットプリント／屋根種別抽出	衛星画像を入力し、建築物輪郭と屋根種別が抽出できる機能
	FN002	道路／植生判別	衛星画像を入力し、道路、植生を判別できる機能
	FN003	開口部情報取得	LOD1 - 2、沿道画像、MMS を入力し、画像処理と三次元点群処理を併用して開口部の情報を取得できる機能
パラメータ設定機能	FN004	パラメータ設定機能	3D 都市モデル生成時の汎用的なパラメータとして、精細度、乱数シード、高さ、屋根形状種別、開口部情報、建築物フットプリントデータを設定でき、建築物精細度の設定に対して異なるパラメータ種類を配置できる機能。また各地物のパラメータを設定できる機能
建築物モデル生成	FN005	建築物の学習	既存の 3D 都市モデルを MeshXL に基づく Transformer 系自己回帰生成モデルで直接学習する機能
	FN006	LOD1 建築物モデル生成	建築物フットプリントデータから建築物モデルを LOD や屋根タイプ別に LOD1 を直接構築するモデルを開発する。これにより実際の都市データ形状に基づいた基本的な建築物モデルの生成を可能にする CityGML(PLATEAU 製品仕様書第 4.0 版)形式で出力する
	FN007	LOD2.0 建築物モデル生成	建築物モデルを学習することで建築物フットプリントデータから建築物モデルを LOD や屋根タイプ別に LOD2 を生成するモデル[学習させられた MeshXL に基づく Transformer 系自己回帰生成モデル]を開発する。これにより実際の都市データ形状に基づいた多様な建築物モデルの生成を可能にする CityGML(PLATEAU 製品仕様書第 4.0 版)形式で出力する

建築物以外の 3D 都市モデル生成機能	FN008	LOD3.0 建築物モデル生成	LOD2 と三次元点群データ、あるいは街路画像を入力として LOD3 モデルを生成する
	FN009	植生モデル生成	衛星・沿道画像リファレンスの結果をパラメータとして、植生の 3D 都市モデルを入力パラメータ (LOD, 種別) に基づいて LOD0-3 を生成するモデルを開発する。CityGML(PLATEAU 製品仕様書第 4.0 版)形式で出力する
	FN010	道路モデル生成	OSM 道路データをリファレンスとして、道路中心線・種別 (幅員、レーン数など) を抽出し、3D 都市モデル生成の入力パラメータ (LOD、幅員) として活用する。 OpenStreetMap の属性に基づき、交差点の立体構造も考慮した LOD0-3 の道路モデルを自動生成する手法を開発する。CityGML (PLATEAU 製品仕様書第 4.0 版) 形式で出力する
	FN011	都市設備モデル生成	沿道画像リファレンスの結果をパラメータとして、都市設備の 3D 都市モデルを入力パラメータ (LOD、高さ、種別) に基づいて LOD0-3 を生成するモデルを開発する。 CityGML(PLATEAU 製品仕様書第 4.0 版)形式で出力する
3D 都市モデル再構築／配置	FN012	3D 都市モデル再構築	生成された建築物／道路／都市設備モデルを統合し、空間上に配置する
	FN013	3D 都市制約条件取得	都市計画決定情報モデルを用いて実際の都市計画を反映する機能
並列処理機能	FN014	並列処理機能	3D 都市モデル生成に並列処理の機能を実装することで計算コスト及び生成時間を削減する
ノイズ／ファイルサイズ削減機能	FN015	ノイズ／ファイルサイズ削減機能	繰り返し生成に基づく後処理でノイズを低減し、Blocked and Patchified Tokenization (BPT) による効率的な符号化手法でファイルサイズを削減する
可視化機能	FN016	可視化	生成結果を可視化する機能
ノイズ評価	FN017	ノイズ評価	生成された 3D モデルのノイズ評価を行う機能。もしノイズが大きい場合は再度生成を行う

## 4-2-2. 利用したソフトウェア・ライブラリ

表 4-3 利用するライブラリー一覧

ID	ライブラリ	説明
LB001	PyCharm Community	Python で開発する時に使うソフトウェア
LB002	Visual Studio Code	C++で開発する時に使うソフトウェア
LB003	Feature Manipulation Engine (FME)	建築物 GML ファイルを OBJ に変換する時に使うソフトウェア
LB004	Computational Geometry Algorithms Library (GAL)	メッシュを生成、最適化する時に使うライブラリ
LB005	OpenCV	二次元画像データに関する処理に効くライブラリ
LB006	Point Cloud Library (PCL)	三次元点群データに関する処理に効くライブラリ
LB008	PyTorch	生成的モデルの実行に必要となり、基本的な深層学習モジュールも提供されるライブラリ
LB009	torchvision	ツールと方法を提供し、画像の読み込み、前処理、モデル訓練、評価などのタスクを簡素化するライブラリ
LB010	cuda toolkit	GPU 高速計算をサポートし、CUDA プログラミングを行い、複数の GPU での並列アルゴリズムとデータ構造を実現するためのライブラリ
LB011	conda-forge	Conda 環境において、バージョン管理、環境隔離、クロスプラットフォーム互換性、統合開発環境を実現するライブラリ
LB012	yacs	コンピュータビジョンプロジェクトにおいて、設定ファイルを定義し管理するためのライブラリ
LB013	pyyaml	YAML 形式のデータを読み込み、修正し、書き込むために使用される
LB014	scipy	numpy に基づくオープンソースの科学計算ライブラリで、科学、技術、工学分野に多くの数学アルゴリズムを提供しているライブラリ。例えば、線形代数、積分と微分方程式、信号処理などが含まれる
LB015	numpy	大量の次元配列と行列演算をサポートするために使用される
LB016	timm	PyTorch に基づく深層学習画像モデルライブラリで、多数の現代の画像認識・分類モデルの実装が含まれており、最新の研究成果を含めて継続的に更新されている
LB017	gdal	多様な空間データ形式を読み書きするために使用されるライブラリ
LB018	ogr	GDAL のサブセットで、ベクターデータの処理に特化している
LB019	pandas	高性能で使いやすいデータ分析ツールを提供し、DataFrame や Seriesなどをサポートし、CSV、JSON、HTML などのデータの処理が可能

LB020	geopandas	pandas に基づく拡張で、地理空間データの処理に特化している
LB021	shapely	平面オブジェクト (Points、Lines、Polygon など) の処理に使用され、平面オブジェクトの作成、空間分析、トポロジー分析機能を含む
LB022	argparse	プログラム実行時のオプションとパラメータをユーザーが指定できるようにする Python の標準ライブラリ
LB023	albumentations	深層学習及び機械学習における画像の前処理とデータ拡張のために設計されたライブラリ
LB025	omegaconf	軽量な設定ライブラリで、YAML や JSON などの設定ファイルを扱い、強力なマージ、補間及び型チェック機能を提供し、設定管理を簡単かつ柔軟にします
LB026	test-tube	機械学習の実験結果を追跡、整理、分析するためのライブラリで、ハイパーパラメータの最適化と実験のバージョン管理をサポートする
LB027	einops	テンソル操作とリシェイプをより直感的で柔軟に行うための Python ライブラリで、多次元データの再配置、次元変換、スケーリングを処理するための簡潔な表現方法を提供する
LB028	transformers	オープンソースの自然言語処理 (NLP) ライブラリで、BERT、GPT、RoBERTa などの多数の事前訓練済みモデルの実装を提供し、テキスト分類、生成、翻訳などの多様な NLP タスクに使用され、迅速なデプロイと使いやすい API をサポートする
LB029	kornia	PyTorch に基づくオープンソースのコンピュータビジョンライブラリで、一連の微分可能な視覚変換関数を提供し、画像処理、強化、特徴抽出などの操作を実現し、GPU 加速と自動微分をサポートする
LB030	open_clip_torch	PyTorch に基づくオープンソースライブラリで、OpenAI CLIP に似た機能を提供し、自然言語のプロンプトを通じて画像を検索・理解できるようにすることを目的としており、多様な事前訓練モデルとカスタマイズ可能な訓練プロセスをサポートする
LB031	torchmetrics	PyTorch に基づく機械学習の指標ライブラリで、モデル評価用の標準指標 (正確性、精度、再現率など) を一連提供し、モデル性能評価プロセスを簡素化し統一することを目的としている
LB032	addict	Python の dict を拡張し、属性アクセスを通じて動的にネストされた dict を作成し、及びアクセスできる使いやすいインターフェースを提供し、dict 操作の複雑さを簡素化する
LB033	yapf	Python コードフォーマットツールで、PEP 8 スタイルガイドに準拠するように Python コードを自動的に再フォーマットし、コードの可読性と一貫性を向上させることを目的とする

LB034	trimesh	三次元メッシュの読み込み、処理、表示、分析を行うためのもので、多様なファイル形式に対応し、三次元幾何データを扱うためのシンプルなインタフェースを提供する
LB035	random	擬似乱数を生成するためのもので、整数、浮動小数点数、シーケンスからの要素の選択、ランダムな文字の生成など、さまざまな乱数生成器を提供する
LB036	lxml	XML 及び HTML ドキュメントを解析するために使用され、XML 及び HTML コンテンツに迅速にアクセス、修正、作成するためのシンプルで使いやすい API を提供する
LB037	earcut	軽量な多角形三角形分割ツールで、複雑な多角形を三角形メッシュに変換するために使用され、グラフィックレンダリングや地理情報システム (GIS) データ処理によく使用される
LB038	math	一連の数学関数と定数を提供し、浮動小数点数の数学演算を実行するために使用される。これには、三角関数、対数関数、指数関数などが含まれる
LB040	OpenGL	グラフィックスライブラリであり、クロスプラットフォームに設計されている。二次元及び三次元レンダリングに用いられる
LB041	accelerate	GPU を使用して Haskell プログラムの並列計算を高速化するためのツール
LB042	beartype	Python で静的型チェックを実行し、型アノテーションの正確性を検証するライブラリ
LB043	Huggingface-hub	機械学習モデルやデータセットの共有、バージョン管理及びコラボレーションをサポートするプラットフォーム
LB044	ema-pytorch	深層学習モデルのトレーニングにおいて、指数移動平均 (EMA) を使用してモデルのパラメータをスムーズに更新するための PyTorch ライブラリ
LB045	environs	環境変数を簡単に管理し、設定を読み込むための Python ライブラリ
LB046	Gateloop-transformers	Transformer モデルの効率的なトレーニングと推論を目的としたライブラリ
LB047	jaxtyping	JAX を用いた数値計算において、型アノテーションを活用して型の正確性を保証する Python ライブラリ
LB048	tqdm	Python でプログレスバーを簡単に追加できるライブラリ
LB049	x-transformers	柔軟でモジュール化されたトランスフォーマーモデルの実装を提供し、Transformer のさまざまなバリエーションを簡単に構築及びカスタマイズできる Python ライブラリ
LB050	torch_geometric	PyTorch 用のグラフニューラルネットワークライブラリ

LB051	mesh2sdf	3D メッシュから Signed Distance Field (SDF) を生成するライブラリ
LB052	einx	エンタープライズ向けの多目的数値計算ライブラリ
LB053	optimum	機械学習モデルの最適化とハイパーパラメータ調整を簡単に行うためのライブラリ
LB054	Plateaukit	PLATEAU の 3D 都市モデルを扱うための変換ツール及び Python ライブラリ
LB055	Detectron2	Facebook によって開発されたオープンソースの物体検出とインスタンスセグメンテーションライブラリで、深層学習による画像分割モデルの訓練と推論フレームワークの提供に特化している
LB056	Detrex	Transformer ベースの検出アルゴリズムを提供するオープンソースのツールボックス
LB057	OpenGL	グラフィックスライブラリであり、クロスプラットフォームに設計されている。二次元及び三次元レンダリングに用いられる
LB058	React (React.js)	Web サイトを構築するための JavaScript ライブラリ
LB059	Next.js	React を基に開発された Web アプリケーションフレームワーク
LB060	Three.js	三次元コンピュータグラフィックスを描画する JavaScript ライブラリ
LB061	FastAPI	Python ベースの RESTful API を構築するための Web フレームワーク
LB062	citygml-convert-tools	CityGML 変換ツール
LB063	xgboost	分類と回帰のための教師付き機械学習手法
LB064	scikit-learn	Python のオープンソース機械学習ライブラリ
LB065	You Only Look Once (YOLO)	AI を活用した画像認識及び動画認識で広く使われている代表的な物体検出ライブラリ

### 4-2-3. 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**朱文字**で示す。

#### 1. 【FN001】建築物フットプリント／屋根種別抽出

- 機能概要

- 3D 都市モデルを生成したい地域の衛星画像を入力し、建築物のインスタンスセグメンテーションと分類を行うことで建築物モデルの生成に必要な建築物の輪郭と屋根形状タイプを判別することが可能となる

- フローチャート

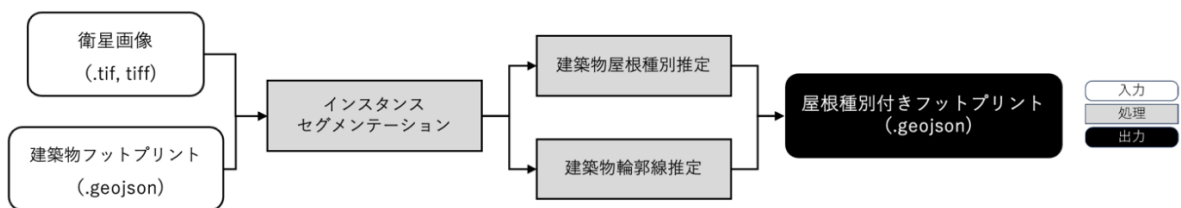


図 4-7 建築物フットプリント／屋根種別抽出機能のフローチャート

- データ仕様

- 入力

- ◇ 衛星画像

- 内容
  - 対象エリアの衛星画像
- 形式
  - tif、tiff 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

- ◇ フットプリント

- 内容
  - 対象エリアのフットプリント
- 形式
  - GeoJSON 形式
- データ詳細
  - ファイル入力インターフェース【IF006】を参照

- 出力

- ◇ 屋根種別付フットプリント

- 内容
  - 屋根種別付建築物ポリゴン
- 形式

- GeoJSON 形式
- データ詳細
  - 内部連携インタフェース【IF201】を参照
- 機能詳細
  - インスタンスセグメンテーションによる建築物輪郭抽出
    - ◇ 処理内容
      - アルゴリズムトレーニング向けの国内の衛星画像と建築物のデータセット【DT101】と手動によるアノテーション建築物輪郭抽出のデータを用意する
      - Mask DINO (Swin-L バックボーン) モデル (25 エポック) をトレーニングする
      - 学習済みモデルを入力 of 衛星画像に実装し、建築物インスタンスセグメンテーションを実施し、建築物輪郭を自動抽出する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【LB001】【LB002】【LB005】【LB008】【LB009】【LB010】【LB011】【LB017】【LB020】【LB021】【LB028】【LB031】【LB050】【LB055】を参照
    - ◇ 利用するアルゴリズム
      - 建築物抽出アルゴリズム：Mask-Dino (アルゴリズム【AL004】を参照)
      - 建築物ポリゴン正規化アルゴリズム (アルゴリズム【AL104】を参照)
  - 屋根種別の判別
    - ◇ 処理内容
      - PLATEAU のオープンデータに含まれる建築物輪郭を用い、建築物ごとに適切なバッファを設定した上で衛星画像を切り出す
      - 建築物のフットプリント形状で切り出した衛星画像に対して屋根種別のアノテーションを実施し、合計 56,955 棟の屋根種別判別データセットを作成する
      - Swin Transformer モデルを採用し、屋根分類アルゴリズムの学習を行う
      - 建築物輪郭の自動抽出及び入力 of 衛星画像の切り出しで得た画像に対し、学習済みモデルを適用して屋根分類を実装し、その結果を実地の地理座標に反映する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【LB001】【LB002】【LB005】【LB008】【LB009】【LB010】【LB011】【LB015】【LB017】【LB020】【LB021】【LB031】【LB048】を参照
    - ◇ 利用するアルゴリズム
      - 屋根種別分類アルゴリズム：Swin Transformer (アルゴリズム【AL005】を参照)
- 2. 【FN002】道路／植生判別
  - 機能概要
    - 生成したい地域の衛星画像を入力し、道路データを OSM から切り取る作業や、セマンティックセグメンテーションを行うことで建築物モデルを生成時に必要な植生エリアのポリゴンを推定することが可能な機能
  - フローチャート

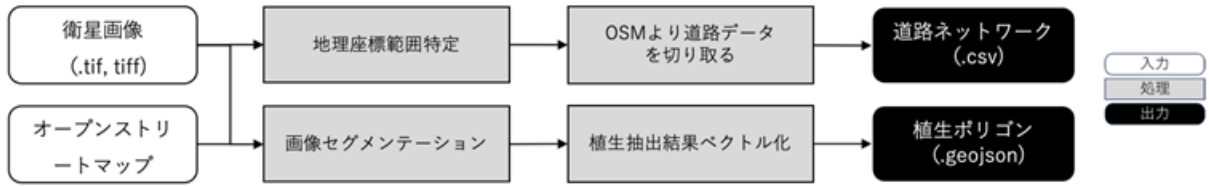


図 4-8 道路植生抽出のフローチャート

- データ仕様

- 入力

- ◇ 衛星画像

- 内容
        - 対象エリアの衛星画像
      - 形式
        - tif、tiff 形式
      - データ詳細
        - ファイル入力インターフェース【IF008】を参照

- ◇ オープンストリートマップ

- 内容
      - 対象エリアのオープンストリートマップ
    - 形式
      - CSV 形式
    - データ詳細
      - ファイル入力インターフェース【IF007】を参照

- 出力

- ◇ 植生ポリゴン

- 内容
        - 植生ポリゴン
      - 形式
        - Geojson 形式
      - データ詳細
        - 内部連携インターフェース【IF203】を参照

- ◇ 道路ネットワーク

- 内容
      - 道路ネットワーク
    - 形式
      - CSV 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

➤ 植生抽出

◇ 処理内容

- 植生セグメンテーションデータセット【DT105】に対して前処理を行い、都市緑地のためのセマンティックセグメンテーションアルゴリズムの学習用データを作成する
- InternImage-Base モデルを用いて、都市緑地のセマンティックセグメンテーションの学習を行う
- 学習済みモデルを入力の中の写真に適用し、都市緑地の自動抽出を実施する
- 自動抽出結果をベクトル化し、実際の地理座標系へ変換して反映する

◇ 利用するライブラリ

- ソフトウェア・ライブラリ【LB001】【LB002】【LB015】【LB017】【LB018】【LB019】【LB020】【LB035】【LB037】【LB038】【LB048】を参照

◇ 利用するアルゴリズム

- 【AL006】【AL104】を参照

➤ 道路抽出

◇ 処理内容

- 入力された衛星画像から座標系情報及び座標範囲を取得する
- 取得した座標系と範囲に基づき、OSM の API を用いて対象範囲の道路データを取得する

◇ 利用するライブラリ

- ソフトウェア・ライブラリ【LB001】【LB002】【LB015】【LB017】【LB019】【LB020】【LB021】【LB035】【LB038】を参照

◇ 利用するアルゴリズム

- 【AL105】を参照

3. 【FN003】開口部情報取得

● 機能概要

- 生成したい地域の沿道で取得された MMS データ（三次元点群データ、画像、カメラの撮影位置と傾き）と、PLATEAU の LOD1/2 の建築物データを入力し、画像処理と三次元点群処理を併用することで LOD3 作成に必要な建築物の開口部を三次元的に取得し、パラメータとして出力する機能

● フローチャート

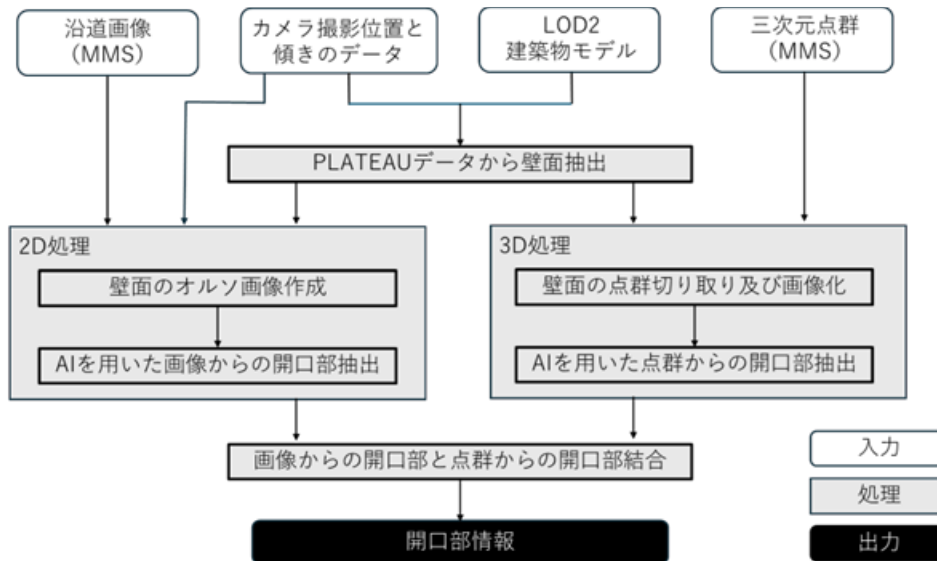


図 4-9 開口部情報取得のフローチャート

- データ仕様

- 入力

- ◇ MMS で撮影された沿道画像データ

- 内容
        - 対象エリアの沿道画像データセット
      - 形式
        - JPEG などの画像ファイル形式
      - データ詳細
        - ファイル入力インターフェース【IF002】を参照

- ◇ MMS で撮影された画像のカメラの撮影位置と傾きのデータ

- 内容
      - 走行軌跡と MMS の機種ごとのカメラの設置角度など
    - 形式
      - テキストファイル形式
      - NTT フォーマットにのみ対応
    - データ詳細
      - ファイル入力インターフェース【IF005】を参照

- ◇ MMS の三次元点群データ

- 内容
      - 対象エリアの MMS データセット
    - 形式
      - LAS 形式
    - データ詳細
      - ファイル入力インターフェース【IF003】を参照

- ◇ PLATEAU の建築物データ

- 内容
  - 対象エリアの PLATEAU の LOD1-2 の建築物データ
- 形式
  - CityGML 形式
- データ詳細
  - ファイル入力インタフェース【IF004】を参照
- 出力
  - ◇ 開口部の PLATEAU 更新ファイル
    - 内容
      - LOD3 作成に必要な建築物の開口部を三次元的に取得し、パラメータとして出力
    - 形式
      - GeoJSON 形式
    - データ詳細
      - 内部連携インタフェース【IF202】を参照
- 機能詳細
  - PLATEAU データから壁面抽出
    - ◇ 処理内容
      - 走行軌跡とカメラ角度を用いて処理対象壁面のみを抽出する
      - オクルージョン処理として、見えていない平面を削除した分割平面を作成する
      - 全ての壁面データに対して、画像を対応付ける
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【LB001】【LB002】を参照
    - ◇ 利用するアルゴリズム
      - 【AL103】を参照
  - 2D 処理
    - ◇ 処理内容
      - 撮影位置と角度から画像と PLATEAU データ（壁面）を数学的に関連付ける
      - 壁面のオルソ画像を作成する
      - AI を用いて画像からの開口部を抽出する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【LB001】【LB002】【LB005】【LB038】【LB065】を参照
    - ◇ 利用するアルゴリズム
      - 【AL015】を参照
  - 3D 処理
    - ◇ 処理内容
      - 三次元点群（MMS）と壁面（PLATEAU）の重ね合わせを行う
      - 壁面上の点群を切り取り、画像化する
      - AI を用いて点群からの開口部を抽出する

- ◇ 利用するライブラリ
  - ソフトウェア・ライブラリ 【LB001】 【LB002】 【LB005】 【LB006】 【LB038】 【LB065】を参照
- ◇ 利用するアルゴリズム
  - 【AL014】 【AL015】 を参照
- 画像からの開口部と点群からの開口部結合
  - ◇ 処理内容
    - 2D 処理の結果と 3D 処理の結果の重ね合わせを行う
    - 重複している開口部に対して、どれを採用するかの判定処理を行う
    - PLATEAU 更新ファイル作成する
  - ◇ 利用するライブラリ
    - ソフトウェア・ライブラリ 【LB001】 【LB002】 【LB005】 【LB006】 【LB038】 【LB065】を参照
  - ◇ 利用するアルゴリズム
    - 【AL014】 【AL015】 を参照

#### 4. 【FN004】パラメータ設定機能

- 機能概要
  - 3D 都市モデル生成時の汎用的なパラメータとして、対象地物、精細度、乱数シード、高さ、屋根形状種別、開口部情報、建築物フットプリントデータを設定できる。具体的には、乱数シード、スケール、対象地物を設定することで、生成結果の固定や、出力したい地物だけの生成が可能になる。また各地物のパラメータを設定できるようにする。地物別には、建築物：精細度、高さ、屋根形状、開口部情報。道路：精細度、幅員。設備：精細度、種別。そのために、これらのパラメータを手動で設定できる機能を実装する。なお、建築物には、異なる精細度の設定に対して別々にパラメータ種類の構成を配置できる
- フローチャート

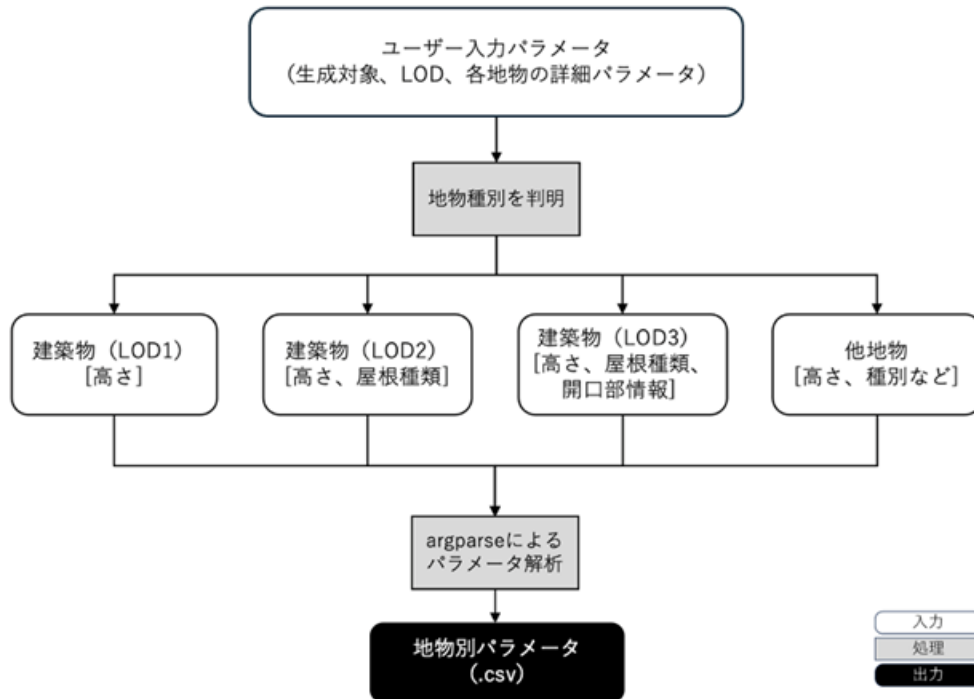


図 4-10 パラメータ設定機能のフローチャート

- データ仕様
  - 入力
    - ◇ ユーザ指定パラメータ
      - 内容
        - 3D 都市モデル生成時に設定する汎用パラメータ
      - 形式
        - 数値
      - データ詳細
        - なし
  - 出力
    - ◇ 地物別パラメータ
      - 内容
        - 地物別パラメータ
      - 形式
        - 数値
      - データ詳細
        - 内部連携インタフェース【IF206】を参照
- 機能詳細

表 4-4 設定可能なパラメータ一覧

地物種類	パラメータ	サブ	変数名	初期値	レンジ	備考
共通	乱数シード		random_seed	1024	random_seed ∈ [0, 65535] (int)	Random seed of generation
建築物	生成レベル選択		lod_building	2	lod_building ∈ {0, 1, 2} (int)	building lod
	階数レンジ	最小値 (1)	storey_low	1	storey_low ∈ [1, 50] (int)	low limit of building storey
		最大値 (50)	storey_high	50	storey_high ∈ [1, 50] (int)	high limit of building storey
	屋根形状割合 (%) : (lod_building=2 の場合)	フラット	prob_t1	0.2	prob_t1 ∈ [0., 1.] (float)	probability of type 1, flat
		段フラット	prob_t2	0.3	prob_t2 ∈ [0., 1.] (float)	probability of type 2, flat with different storey
		腰折	prob_t3	0.3	prob_t3 ∈ [0., 1.] (float)	probability of type 3, mixed
		寄棟	prob_t4	0.2	prob_t4 ∈ [0., 1.] (float)	probability of type 4, slope1
		切妻	prob_t5	0	prob_t5 ∈ [0., 1.] (float)	probability of type 5, slope2
		一様フラット	prob_t6	0	prob_t6 ∈ [0., 1.] (float)	probability of type 6, pure flat
開口部情報		aperture_info	1	aperture_info ∈ [0, 10]	Number of apertures	
道路	生成レベル選択		lod_road	1	lod_road (0,1,2) (int)	road lod
	幅員レンジ (m)	最小値	road_width_low	1	road_width_low ≥ 1.0 (float)	low limit, width of road
		最大値	road_width_high	20	road_width_high ≤ 20.0 (float)	high limit, width of road
	車道と歩道の幅員の割合	車道部	road_width_main	0.9	float	width ratio of main road
		歩道	road_width_sub	0.1	float	width ratio of sub road

設備	生成レベル 選択		lod_device	2	lod_device (0, 1, 2) (int)	device lod
	設備種別割 合)	信号	traffic_light_rat io	0.1	Float	traffic light ratio
		電柱	telegraph_pole _ratio	0.9	float	telegraph pole ratio

- ◇ 利用するライブラリ
  - ソフトウェア・ライブラリ【SL022】を参照
- ◇ 利用するアルゴリズム
  - なし

5. 【FN005】建築物の学習

● 機能概要

- 膨大な PLATEAU の建築物モデルは MeshXL に基づく Transformer 系自己回帰生成モデルを用いて二次元への変換をせず、三次元形状を保ったまま学習することで、建築物フットプリントデータ及び他のパラメータを入力として現実に即した 3D モデルが生成可能なモデルを開発する。これにより実際の建築物フットプリントデータ形状に基づいた多様な建築物モデルの生成を可能にする

● フローチャート

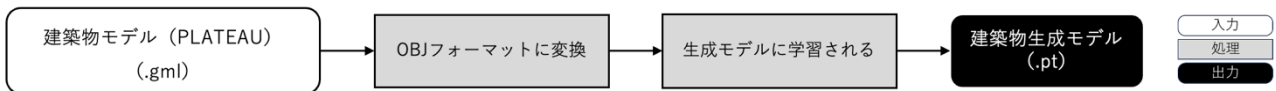


図 4-11 建築物の直接学習機能のフローチャート

● データ仕様

- 入力
  - ◇ PLATEAU の建築物モデル
    - 内容
      - PLATEAU の建築物モデル
    - 形式
      - CityGML 形式
    - データ詳細
      - ファイル入力インターフェース【IF009】を参照
- 出力
  - ◇ 多様な建築物モデルを生成するモデル
    - 内容
      - Transformer 系自己回帰 3D 生成モデル
    - 形式

- モデルの checkpoint ファイル (.pt) 形式
- データ詳細
  - 内部連携インタフェース【IF207】を参照
- 機能詳細
  - データ形式変換
    - ◇ 処理内容
      - CityGML である建築物モデルを OBJ フォーマットに変換する
    - ◇ 利用するライブラリ
      - 【SL003】
    - ◇ 利用するアルゴリズム
      - なし
  - モデル学習
    - ◇ 処理内容
      - 変換された建築物モデルをモデルに学習する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【SL001】【SL008】【SL009】【SL027】【SL028】を参照
    - ◇ 利用するアルゴリズム
      - 【AL001】【AL002】を参照

6. 【FN006】 LOD1 建築物モデル生成

- 機能概要
  - 建築物生成モデル (Transformer 系モデル) を用いて、実際の都市の建築物フットプリントデータ及び高さなどをパラメータ入力として LOD1 以下の建築物モデルを直接構築する機能
- フローチャート

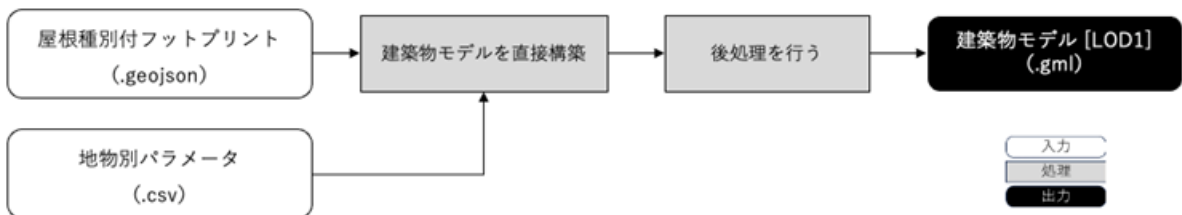


図 4-12 建築物モデル生成機能のフローチャート

- データ仕様
  - 入力
    - ◇ 地物別パラメータ
      - 内容
        - 建築物モデルを生成するためのパラメータ
      - 形式
        - CSV 形式
      - データ詳細

- 内部連携インタフェース【IF206】を参照
  - ◇ 屋根種別付フットプリント
    - 内容
      - 建築物モデルを生成するためのフットプリント
    - 形式
      - GeoJSON 形式
    - データ詳細
      - 内部連携インタフェース【IF201】を参照
  - 出力
    - ◇ LOD 1 の建築物モデル
      - 内容
        - 建築物モデル
      - 形式
        - 三角形又はポリゴンメッシュ（CityGML ファイルで出力）、LOD 1（PLATEAU 製品仕様書第 4.0 版に準拠）
      - データ詳細
        - 内部連携インタフェース【IF209】を参照
  - 機能詳細
    - 建築物モデル生成のパラメータ配置
      - ◇ 処理内容
        - 設定された LOD レベルにより、異なるパラメータの配置を行う
      - ◇ 利用するライブラリ
        - 【SL022】
      - ◇ 利用するアルゴリズム
        - なし
    - 建築物モデルの生成
      - ◇ 処理内容
        - 入力されたパラメータにより建築物モデルを直接構築する
      - ◇ 利用するライブラリ
        - ソフトウェア・ライブラリ【SL001】【SL008】【SL009】【SL027】【SL028】を参照
      - ◇ 利用するアルゴリズム
        - なし
7. 【FN007】 LOD2 建築物モデル生成
- 機能概要
    - 建築物生成モデル（学習された MeshXL に基づく Transformer 系モデル）を用いて、実際の都市の建築物フットプリントデータ及び精細度、屋根タイプ、高さなどをパラメータ入力として LOD2 の建築物モデルを生成する機能

● フローチャート

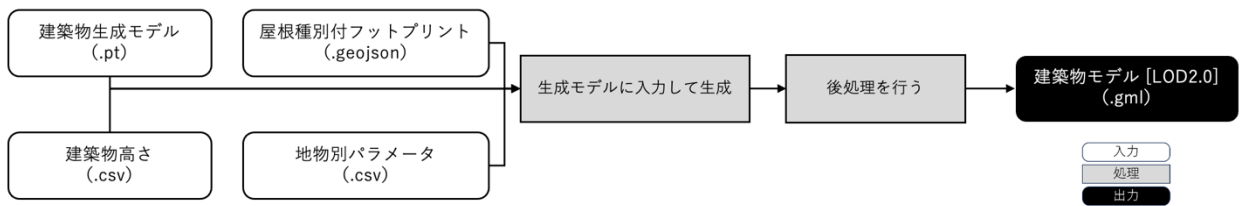


図 4-13 建築物モデル生成機能のフローチャート

● データ仕様

➤ 入力

◇ フットプリント

- 内容
  - 対象エリアのフットプリント
- 形式
  - GeoJSON 形式
- データ詳細
  - ファイル入力インターフェース【IF006】を参照

◇ 地物別パラメータ

- 内容
  - 建築物モデルを生成するためのパラメータ
- 形式
  - CSV 形式
- データ詳細
  - 内部連携インターフェース【IF206】を参照

◇ 建築物高さ

- 内容
  - 建築物高さ
- 形式
  - CSV 形式
- データ詳細
  - 内部連携インターフェース【IF205】を参照

◇ 生成モデル

- 内容
  - 建築物モデルを生成するためのモデル
- 形式
  - pt 形式
- データ詳細
  - 内部連携インターフェース【IF207】を参照

➤ 出力

◇ LOD2 の建築物モデル

- 内容
  - LOD2 の建築物モデル
- 形式
  - 三角形メッシュ (CityGML ファイルで出力)、LOD 2 (PLATEAU 製品仕様書第 4.0 版に準拠)
- データ詳細
  - 内部連携インタフェース【IF210】を参照

● 機能詳細

➢ 建築物モデル生成のパラメータ配置

◇ 処理内容

- 設定された LOD レベルにより、異なるパラメータの配置を行う

◇ 利用するライブラリ

- ソフトウェア・ライブラリ【SL022】を参照

◇ 利用するアルゴリズム

- なし

➢ 建築物モデルの生成

◇ 処理内容

- 入力されたパラメータにより建築物モデルを生成する

◇ 利用するライブラリ

- ソフトウェア・ライブラリ【SL001】【SL008】【SL009】【SL027】【SL028】を参照

◇ 利用するアルゴリズム

- 【AL001】を参照

➢ 建築物モデルの後処理

◇ 処理内容

- 生成された建築物モデルにあるノイズとローカルエラーなどに対処する

◇ 利用するライブラリ

- ソフトウェア・ライブラリ【SC001】【SL034】を参照

◇ 利用するアルゴリズム

- なし

8. 【FN008】 LOD3 建築物モデル生成

● 機能概要

- FN007 で生成させられた LOD2 の建築物モデルを用いて、沿道画像・MMS 点群から抽出された開口部情報をパラメータ入力として、LOD3 の建築物モデルを生成する機能

● フローチャート

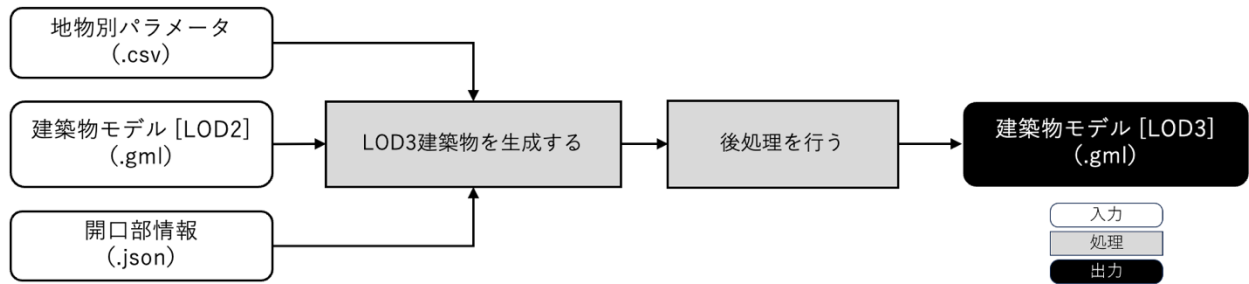


図 4-14 機能のフローチャート

- データ仕様

- 入力

- ◇ 建築物モデル (LOD2)

- 内容
        - LOD2 の建築物モデル
      - 形式
        - CityGML 形式
      - データ詳細
        - 内部連携インターフェース【IF004】を参照

- ◇ 開口部情報

- 内容
        - 沿道画像/MMS 点群から抽出された開口部情報
      - 形式
        - GeoJSON 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

- ◇ LOD3 の建築物モデル

- 内容
        - 建築物モデル (LOD3)
      - 形式
        - 三角形メッシュ (CityGML ファイルで出力)、LOD 3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
      - データ詳細
        - 内部連携インターフェース【IF212】を参照

- 機能詳細

- LOD3 モデルの生成

- ◇ 処理内容

- LOD2 建築物モデルと沿道画像・MMS 点群から抽出された開口部情報をパラメータ入力として、LOD3 の建築物モデルを生成する

- ◇ 利用するライブラリ
  - ソフトウェア・ライブラリ 【SL001】 【SL008】 【SL009】 【SL027】 【SL028】 を参照
- ◇ 利用するアルゴリズム
  - なし

9. 【FN009】 植生モデル生成

● 機能概要

- 植生の 3D 都市モデルを入力パラメータ (LOD, 幅員, 種別) に基づいて生成する機能を開発する。具体的には、衛星／沿道画像リファレンスの結果を取得しパラメータとして、入力すると相応しい植生モデルを生成できるモデルを開発する。生成された植生モデルの結果は、CityGML(PLATEAU 製品仕様書第 4.0 版に準拠)形式で出力する

● フローチャート



図 4-15 植生モデル生成機能のフローチャート

● データ仕様

➢ 入力

◇ 地物別パラメータ

- 内容
  - 3D 植生モデルを生成するためのパラメータ (LOD、幅員、種別)
- 形式
  - CSV 形式
- データ詳細
  - 内部連携インタフェース 【IF206】 を参照

➢ 出力

◇ 植生モデル

- 内容
  - 3D 植生モデル
- 形式
  - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD0／1／2／3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
- データ詳細
  - 内部連携インタフェース 【IF213】 を参照

● 機能詳細

- 植生モデルの生成

- ◇ 処理内容
  - 入力されたパラメータにより 3D 植生モデルを生成する
- ◇ 利用するライブラリ
  - 【SL001】，【SL008】，【SL009】，【SL027】，【SL028】
- ◇ 利用するアルゴリズム
  - なし
- 植生モデルの後処理
  - ◇ 処理内容
    - 生成された植生モデルにあるノイズとローカルエラーなどに対処する
  - ◇ 利用するライブラリ
    - 【SL001】，【SL034】
  - ◇ 利用するアルゴリズム
    - なし

## 10. 【FN010】 道路モデル生成

- 機能概要
  - 衛星画像から特定された OSM 道路データを基に、道路の 3D 都市モデルを入力パラメータ（LOD、幅員）に基づいて生成する機能を開発する。具体的には、OSM の属性情報と空間構造を解析し、幅員や道路種別、現実に即した道路モデルを生成するアルゴリズムを構築する。生成された道路モデルは、CityGML（PLATEAU 製品仕様書第 4.0 版 に準拠）形式で出力する
- フローチャート



図 4-16 道路モデル生成機能のフローチャート

- データ仕様
  - 入力
    - ◇ 道路ネットワーク
      - 内容
        - FN002 から抽出された OSM 道路データ
      - 形式
        - CSV 形式
      - データ詳細
        - 内部連携インターフェース 【IF204】 に参照
    - ◇ 地物別パラメータ
      - 内容

- 植生モデルを生成するためのパラメータ (LOD、幅員、種別)
  - 形式
    - CSV 形式
  - データ詳細
    - 内部連携インタフェース【IF206】を参照
- 出力
  - ◇ 3D 道路モデル
    - 内容
      - OSM より生成された道路モデル
    - 形式
      - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
    - データ詳細
      - ファイル入力インタフェース【IF214】に参照
- 機能詳細
  - 道路モデルの生成
    - ◇ 処理内容
      - 入力されたパラメータと OSM 道路データにより道路モデルを生成する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【SL001】【SL015】【SL019】【SL020】【SL021】【SL034】を参照
    - ◇ 利用するアルゴリズム
      - なし
  - 道路モデルの後処理
    - ◇ 処理内容
      - 生成された道路モデルにあるノイズとローカルエラーなどに対処する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【SC001】【SL034】を参照
    - ◇ 利用するアルゴリズム
      - なし

## 11. 【FN011】都市設備モデル生成

- 機能概要
  - 都市設備の 3D 都市モデルを入力パラメータ (LOD, 高さ, 種別) に基づいて生成する機能を開発する。具体的には、沿道画像リファレンスの結果を取得しパラメータとして、入力するとふさわしい設備モデルを生成できるモデルを開発する。生成された設備モデルの結果は、CityGML (PLATEAU 製品仕様書第 4.0 版) 形式で出力する
- フローチャート

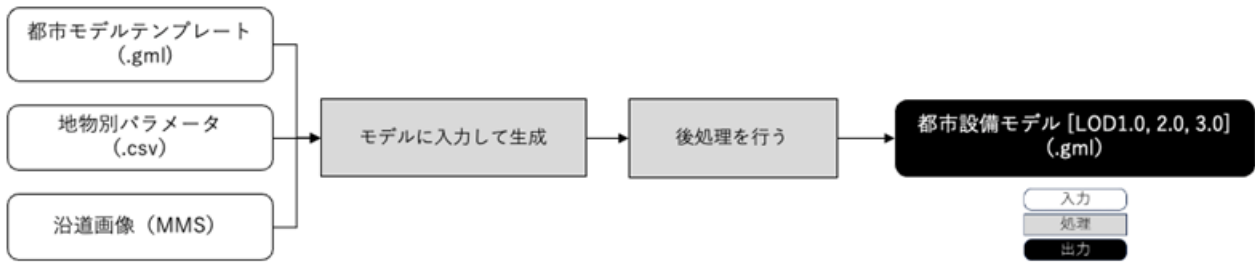


図 4-17 都市設備モデル生成機能のフローチャート

● データ仕様

➤ 入力

◇ 地物別パラメータ

- 内容
  - 3D 都市設備モデルを生成するためのパラメータ (LOD、種別)
- 形式
  - CSV 形式
- データ詳細
  - 内部連携インタフェース【IF206】を参照

◇ 都市モデルテンプレート

- 内容
  - 都市設備モデルのテンプレート
- 形式
  - CityGML 形式
- データ詳細
  - 内部連携インタフェース【IF208】を参照

◇ MMS で撮影された沿道画像データ

- 内容
  - 対象エリアの沿道画像データセット
- 形式
  - JPEG などの画像ファイル形式
- データ詳細
  - ファイル入力インタフェース【IF002】を参照

➤ 出力

◇ 都市設備モデル

- 内容
  - 3D 都市設備モデル
- 形式
  - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)

- データ詳細
  - 内部連携インターフェース【IF215】を参照
- 機能詳細
  - 都市設備モデルの生成
    - ◇ 処理内容
      - 入力されたパラメータにより 3D 都市設備モデルを生成する
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【SL001】【SL008】【SL009】【SL027】【SL028】を参照
    - ◇ 利用するアルゴリズム
      - なし
  - 都市設備モデルの後処理
    - ◇ 処理内容
      - 生成された都市設備モデルにあるノイズとローカルエラーなどに対処する
    - ◇ 利用するライブラリ
      - 【SL001】，【SL034】
    - ◇ 利用するアルゴリズム
      - なし

## 12. 【FN012】 3D 都市モデル再構築

- 機能概要
  - 生成された建築物／道路／都市設備モデルを統合し、空間上に配置する機能を開発する。具体的によ  
ると、入力されたフットプリントと画像リファレンスから取得した空間情報により生成されたメッシ  
ュモデルを配置する手法を開発する
- フローチャート

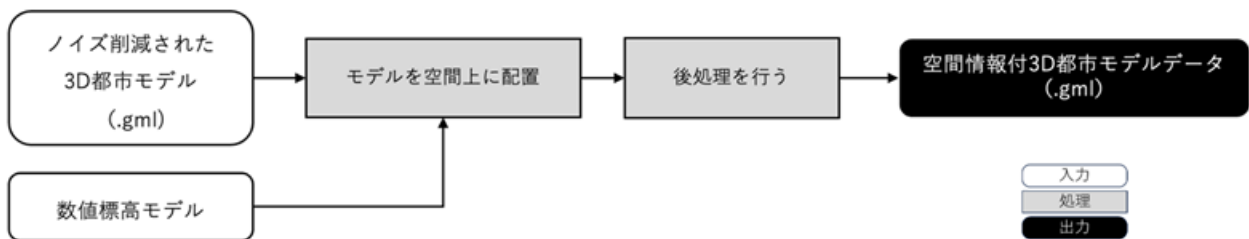


図 4-18 3D 都市モデル再構築機能のフローチャート

- データ仕様
  - 入力
    - ◇ ノイズ削減された 3D 都市モデル
      - 内容
        - 生成された建築物・道路・都市設備モデル及び入力されたりファレンスデータ
      - 形式

- 建築物・道路・都市設備モデルは三角形又はポリゴンメッシュ
- データ詳細
  - 内部連携インタフェース【IF216】を参照
- ◇ 数値標高モデル
  - 内容
    - 標高データ
  - 形式
    - CSV 形式
  - データ詳細
    - ファイル入力インタフェース【IF012】を参照
- 出力
  - ◇ 3D 都市モデルデータ
    - 内容
      - 統合された 3D 都市モデル
    - 形式
      - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
    - データ詳細
      - ファイル出力インタフェース【IF101】を参照

13. 【FN013】 3D 都市モデル配置制約条件取得

- 機能概要
  - 都市計画決定情報モデルを用いて実際の都市計画を反映した配置アルゴリズムを用いた 3D 都市モデルの配置処理を行う
- フローチャート

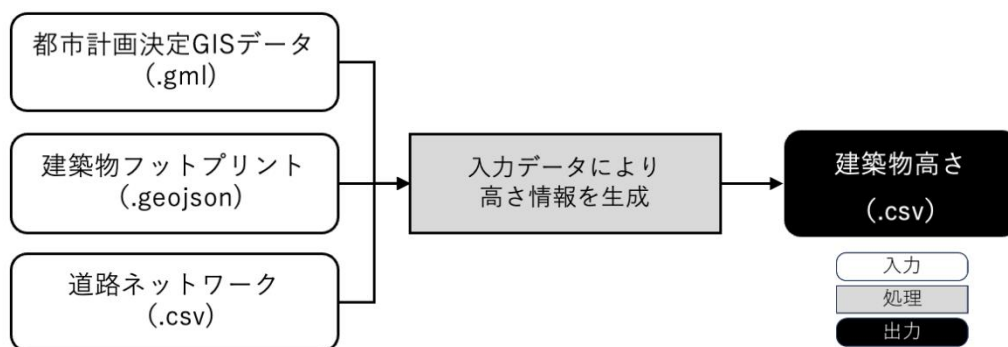


図 4-19 都市モデル配置機能のフロー

- データ仕様
  - 入力
    - ◇ 建築物フットプリント

- 内容
  - 建築物フットプリント
- 形式
  - GeoJson 形式
- データ詳細
  - ファイル入力インタフェース【IF006】を参照
- ◇ 都市計画決定情報データ
  - 内容
    - 都市計画決定情報データ
  - 形式
    - CSV 形式
  - データ詳細
    - ファイル入力インタフェース【IF011】を参照
- ◇ 道路ネットワークデータ
  - 内容
    - 道路ネットワークデータ
  - 形式
    - CSV 形式
  - データ詳細
    - ファイル入力インタフェース【IF204】を参照
- 出力
  - ◇ 建築物高さ
    - 内容
      - 推定された各建築物の高さデータ
    - 形式
      - CSV 形式
    - データ詳細
      - 内部連携インタフェース【IF205】を参照
- 機能詳細
  - 建築物高さ推定
    - ◇ 処理内容
      - 入力した GIS データから各建築物の高さを推定する機械学習モデルを開発し、各建築物の高さの推定を可能にする
    - ◇ 利用するライブラリ
      - ソフトウェア・ライブラリ【SL015】【SL021】【SL023】【SL024】【SL025】【SL062】【SL063】【SL064】を参照
    - ◇ 利用するアルゴリズム
      - 【AL112】【AL013】を参照

14. 【FN014】 並列処理機能

● 機能概要

- 3D 都市モデル生成アルゴリズムに、並列処理の機能を実装することで計算コスト及び生成時間を削減する。具体的には、プロセッサのマルチプロセッシング機能を利用し、3D 都市モデル生成アルゴリズムを複数のコアで動作させて生成時間を削減する

● フローチャート

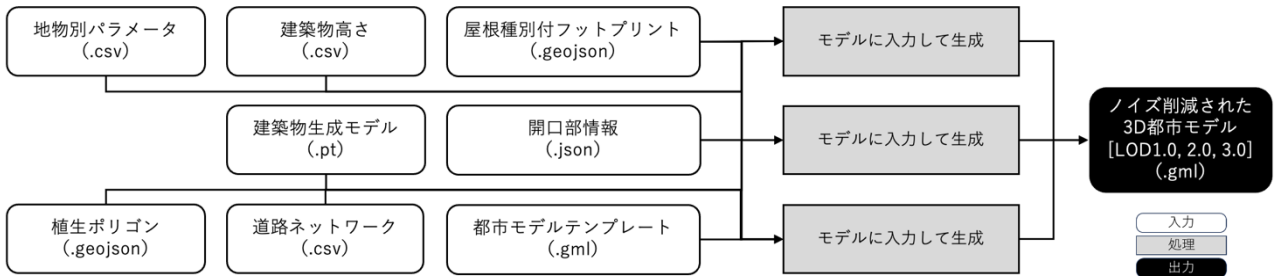


図 4-20 3D 都市モデル再構築機能のフローチャート

● データ仕様

➢ 入力

◇ 並列処理する入力データ

- 内容
  - 建築物、植生、道路及び都市設備モデルを生成するためのパラメータ
- 形式
  - 各モデルを参照
- データ詳細
  - 内部連携インターフェース【IF201】【IF202】【IF203】【IF204】【IF205】【IF206】【IF207】【IF208】を参照

➢ 出力

◇ 内容

- 建築物、植生、道路及び都市設備モデル

◇ 形式

- 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)

◇ データ詳細

- 内部連携インターフェース【IF216】を参照

● 機能詳細

- 3D 都市モデル生成に並列処理を行う

◇ 処理内容

- 3D 都市モデル生成アルゴリズムに並列処理の機能を実装することで計算コスト及び生成時間を削減する

◇ 利用するライブラリ

- ソフトウェア・ライブラリ 【SC001】 【SL008】 【SL009】 【SL027】 【SL028】 【SL034】 を参照
- ◇ 利用するアルゴリズム
  - なし

15. 【FN015】 ノイズ／ファイルサイズ削減機能

● 機能概要

- 繰り返し生成に基づく後処理を導入することで、モデル表面のノイズを効果的に低減する。さらに、Blocked and Patchified Tokenization (BPT) による効率的な符号化手法を適用し、複雑なメッシュ構造を抑えつつ、ファイルサイズの大幅な削減を実現する

● フローチャート

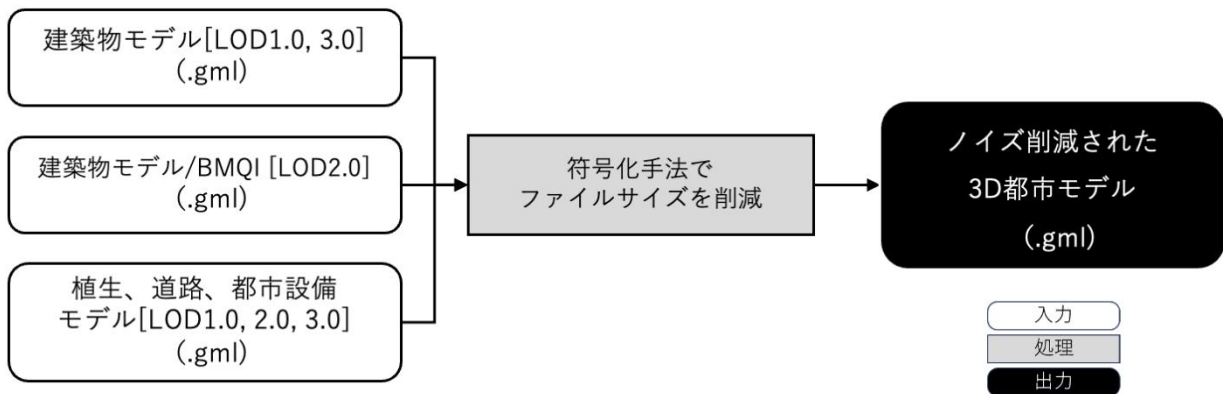


図 4-21 3D 都市モデル再構築機能のフローチャート

● データ仕様

➢ 入力

◇ LOD1 建築物モデル

- 内容
  - LOD1 の建築物モデル
- 形式
  - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
- データ詳細
  - 内部連携インターフェース 【IF209】 を参照

◇ LOD2.0 建築物モデル/BMQI

- 内容
  - LOD2.0 の建築物モデルに BMQI 評価結果が付与されたもの
- 形式
  - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
- データ詳細
  - 内部連携インターフェース 【IF211】 を参照

- ◇ LOD3.0 建築物モデル
  - 内容
    - LOD3.0 の建築物モデル
  - 形式
    - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
  - データ詳細
    - 内部連携インタフェース【IF212】を参照
- ◇ 植生モデル
  - 内容
    - LOD1 - 3.0 の植生モデル
  - 形式
    - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
  - データ詳細
    - 内部連携インタフェース【IF213】を参照
- ◇ 道路モデル
  - 内容
    - LOD1 - 3.0 の道路モデル
  - 形式
    - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
  - データ詳細
    - 内部連携インタフェース【IF214】を参照
- ◇ 都市設備モデル
  - 内容
    - LOD1 - 3.0 の都市設備モデル
  - 形式
    - CityGML (PLATEAU 製品仕様書第 4.0 版に準拠)
  - データ詳細
    - 内部連携インタフェース【IF215】を参照
- 出力
  - ◇ ノイズ削減された 3D 都市モデル
    - 内容
      - 最適化された建築物・植生・道路・都市設備モデル
    - 形式
      - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
    - データ詳細
      - 内部連携インタフェース【IF216】を参照

- 機能詳細

- 3D 都市モデルにノイズ削減機能を実装する

- ◇ 処理内容

- メッシュモデルにジオメトリ最適化などの手法を行い、生成した 3D 都市モデルであった複雑なメッシュを数百枚に削減して、ノイズとファイルサイズを削減する

- ◇ 利用するライブラリ

- 【SL001】，【SL008】，【SL009】，【SL027】，【SL028】，【SL034】

- ◇ 利用するアルゴリズム

- なし

16. 【FN016】 可視化

- 機能概要

- Web ブラウザ上でインタラクティブに 3D 都市モデルを表示し、ユーザーが回転、ズーム、パン操作を行える。生成した 3D 都市モデルのリアルタイムレンダリングを実現する

- フローチャート

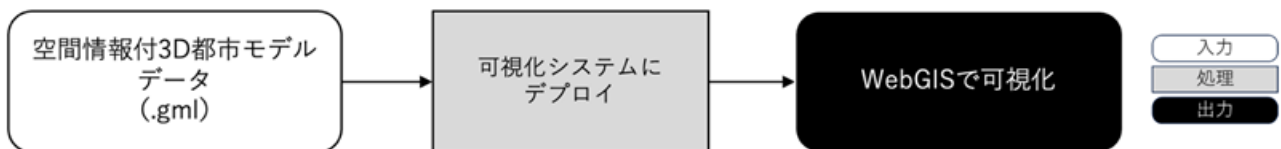


図 4-22 可視化機能のフローチャート

- データ仕様

- 入力

- ◇ 3D 都市モデルデータ

- 内容
        - 統合された 3D 都市モデル
      - 形式
        - 三角形又はポリゴンメッシュ (CityGML ファイルで出力)、LOD 0/1/2/3.0 (PLATEAU 製品仕様書第 4.0 版に準拠)
      - データ詳細
        - ファイル出力インターフェース 【IF101】 を参照

- 出力

- ◇ 内容

- Web ブラウザ上でインタラクティブな表示をした 3D 都市モデル

- ◇ 形式

- なし

- 機能詳細

- 3D 都市モデル可視化

- ◇ 処理内容

- 生成された 3D 都市モデルを可視化する
- ◇ 利用するライブラリ
  - 【LB058】 【LB059】 【LB060】
- ◇ 利用するアルゴリズム
  - なし

## 17. 【FN017】 ノイズ評価

- 機能概要
  - 屋根・側面・地面の部分に分けられた建築物モデルにおいて、フットプリントの範囲収まり、側面の鉛直、そして屋根メッシュの欠損の三要素で、生成された 3D 都市モデルのノイズ有無を定量的に評価する BMQI 指標を開発する
- フローチャート

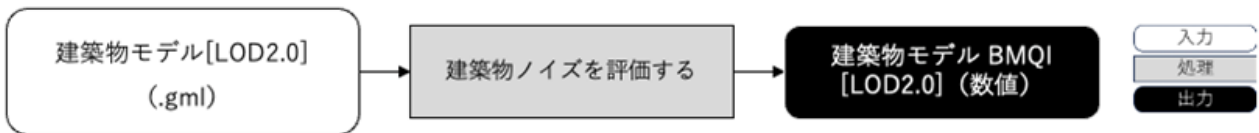


図 4-23 3D 都市モデルのノイズ評価フローチャート

- データ仕様
  - 入力
    - ◇ 建築物モデル[LOD2.0]
      - 内容
        - 生成された LOD2.0 の建築物モデル
      - 形式
        - CityGML 形式
      - データ詳細
        - 内部連携インタフェース 【IF210】 を参照
  - 出力
    - ◇ 建築物モデル BMQI
      - 内容
        - 生成された建築物モデルの地物単位のノイズ評価指標(BMQI)の結果
      - 形式
        - 数値 (0-1.0)
      - データ詳細
        - 内部連携インタフェース 【IF211】 を参照
- 機能詳細
  - 3D 都市モデルのノイズ評価を行う
    - ◇ 処理内容

- 建築物を屋根・側面・地面の三つの部分に分け、それぞれにおいてノイズ評価を行い、0-100で数値化する
- ◇ 利用するライブラリ
  - ソフトウェア・ライブラリ【LB015】【LB034】を参照
- ◇ 利用するアルゴリズム
  - 【AL118】を参照

## 4-3. アルゴリズム

## 4-3-1. 利用したアルゴリズム

表 4-5 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、FN006	MeshXL	<ul style="list-style-type: none"> <li>OPT モデルを活用し、三角形メッシュモデルをグラフ構造として自己回帰的に生成して再構築する手法</li> </ul>	<ul style="list-style-type: none"> <li>MeshXL は、学習された幾何学的ボキャブラリーからトークンを生成するように学習された手法であるため、選定した</li> <li>自己回帰的にサンプリングすることにより、三角メッシュを作成することが可能であるため、選定した</li> </ul>
AL002	FN001	Mask DINO	<ul style="list-style-type: none"> <li>ユーザー側から入力された衛星画像の該当範囲で、建築物輪郭を抽出するためのインスタンスセグメンテーションアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>Mask Dino は、インスタンスセグメンテーションタスクで SOTA 精度を得るために、シンプルで効率的かつスケラブルなものであるため、選定した</li> </ul>
AL003	FN001	Swin Transformer	<ul style="list-style-type: none"> <li>抽出された建築物フットプリントに基づいて屋根の形状タイプを判別するための画像分類アルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>Swin transformer は、より良い全体的及び局所的特徴融合性能を達成するための強力な自己注意メカニズムを持っているため、選定した</li> </ul>
AL004	FN002、FN003	InternImage	<ul style="list-style-type: none"> <li>ユーザー側から入力された衛星画像の該当範囲で、真実の植生と道路の分布を抽出のためのセマンティックセグメンテーションアルゴリズム沿道画像から道路や建築物の領域を抽出するためにも使用</li> </ul>	<ul style="list-style-type: none"> <li>InternImage は、大規模カーネルの畳み込みと変形可能な畳み込みを組み合わせた強力なバックボーンモデルであり、多様なスケールの特徴を効果的に捉えることができるため、選定した</li> </ul>

AL005	FN014	3D モデル表示	<ul style="list-style-type: none"> <li>WebGL だけで 3D 表現をするための JavaScript ライブラリである、Three.js を利用して表示</li> </ul>	<ul style="list-style-type: none"> <li>Three.js は、WebGL 技術を抽象化し、ブラウザ上で高品質な 3D グラフィックスを容易に実装できる業界標準のライブラリであるため、選定した</li> </ul>
AL006	FN014	RESTful API 通信	<ul style="list-style-type: none"> <li>REST (representational state transfer) アーキテクチャ・スタイルの設計原則に準拠する API を利用した通信</li> </ul>	<ul style="list-style-type: none"> <li>RESTful API は、HTTP プロトコルに基づいた標準的な通信方式であり、システム間の疎結合を実現できるため、選定した</li> </ul>
AL007	FN011	Random Forest	<ul style="list-style-type: none"> <li>多数の決定木を組み合わせてデータの分類や回帰を行うアンサンブル学習法</li> </ul>	<ul style="list-style-type: none"> <li>Random Forest は、過学習を抑制しながら高い汎化性能を持つアンサンブル学習手法であり、特徴量の重要度を評価でき、解釈性が高く、欠損値やノイズに対してロバストな予測が可能であるため、選定した</li> </ul>
AL008	FN011	XGBoost	<ul style="list-style-type: none"> <li>勾配ブースティングフレームワークを用いた高性能な機械学習アルゴリズムで、分類や回帰問題に対して精度の高い予測を行う</li> </ul>	<ul style="list-style-type: none"> <li>XGBoost は、勾配ブースティングの効率的な実装であり、正則化により過学習を防ぎながら高精度な予測を実現する。並列処理に対応し、大規模データセットでも高速に学習が可能であり、分類・回帰タスクで広く実績があるため、選定した</li> </ul>
AL009	FN003	RANSAC	<ul style="list-style-type: none"> <li>三次元点群に対して、直線や平面を抽出するアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>直線や平面抽出する際にノイズなどの外れ値に強い手法であるため、選定した</li> </ul>
AL010	FN003	射影変換	<ul style="list-style-type: none"> <li>画像を任意平面にオルソ化するアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>平面の部分をオルソ画像化する際の一般的な手法であるため、選定した</li> </ul>

1) 【AL001】 MeshXL

- 概要
  - 3D メッシュを離散座標トークンのシーケンスとして表現
  - 明示的な座標表現と暗黙的なニューラル埋め込みとの組み合わせ
  - 複雑な前処理なしで直接的な自己回帰生成を実現
- 計算量
  - 三角メッシュ頂点集合を  $V$  として  
 $O(|V|)$
- イメージ

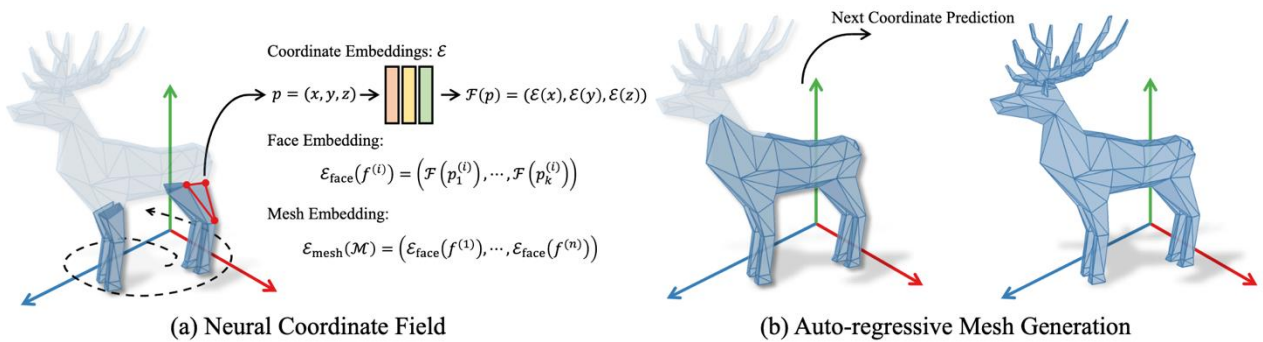


図 4-24 MeshXL のイメージ

2) 【AL002】 MaskDINO

- 概要
  - 画像中の物体領域をインスタンス単位で検出・セグメンテーションするために設計された統合型フレームワーク
  - DETR 系列モデルを基盤とし、クエリを用いたエンドツーエンド学習によって、NMS を必要としない
  - Mask2Former のアイデアを採り入れ、マスク予測・特徴融合・クエリ拡張を統合的に最適化
  - 特に衛星画像などの複雑背景において、高精度な建築物輪郭抽出が可能
- 計算量
  - 入力画像の特徴マップを  $F$  とし、クエリ数を  $Q$  とする場合：  
 $O(|F| + Q^2)$   
 $|F|$ : バックボーンによる特徴抽出 (画像サイズに線形依存)  
 $Q^2$ : クエリ間の自己注意機構による計算量
- イメージ

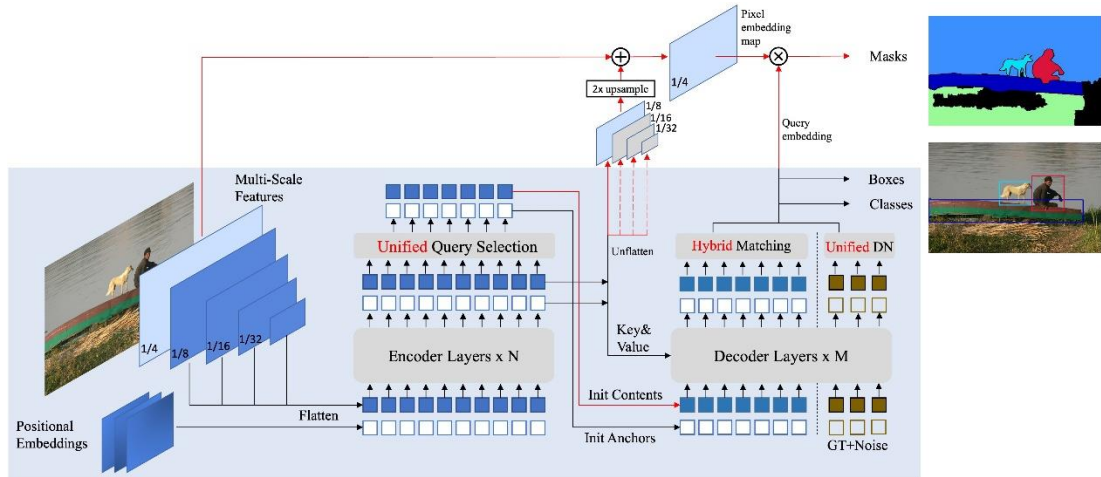


図 4-25 MaskDINO のイメージ

### 3) 【AL003】 Swin Transformer

#### ● 概要

- 階層型構造を持つ Vision Transformer の改良モデルで、画像処理タスク向けに最適化されたバックボーン
- 画像を小領域に分割し、Shifted Window を用いることで、グローバル情報とローカル情報を効率的に融合
- 計算量を局所ウィンドウ内に限定することで、高解像度画像に対してもスケーラブルで安定した処理が可能

#### ● 計算量

- 入力特徴マップを  $F$ 、ウィンドウサイズを  $M$  とすると：  
 $O(|F| \cdot M^2)$

#### ● イメージ

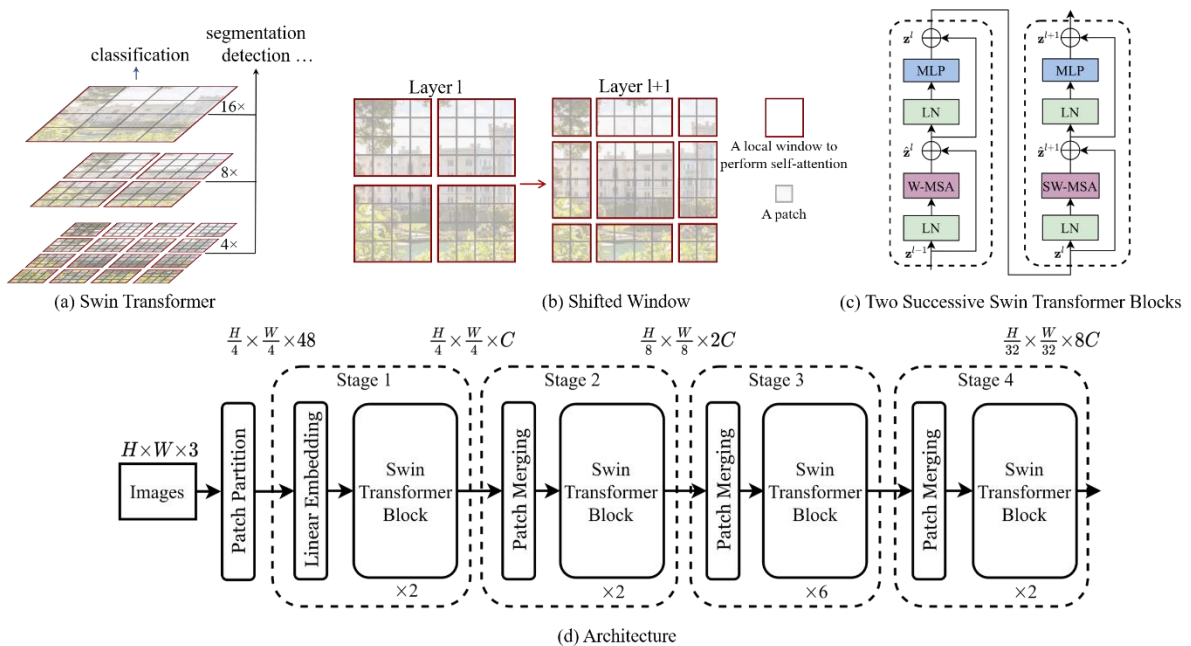


図 4-26 Swin Transformer のイメージ

4) 【AL004】 InternImage

● 概要

- 可変形畳み込み (DCN) を基にした次世代ビジョンバックボーンで、従来の CNN と Transformer の長所を統合した高性能アーキテクチャ
- ピクセルごとに最適化された Large Kernel Dynamic Convolution を用いて、グローバル情報と局所特徴を同時に学習
- Attention に依存せず、計算効率を維持しながら高解像度衛星画像に対して強い表現能力を発揮

● 計算量

- 入力特徴マップを  $F$ 、動的畳み込みのカーネルサイズを  $K$  とすると：  
 $O(|F| \cdot K^2)$

● イメージ

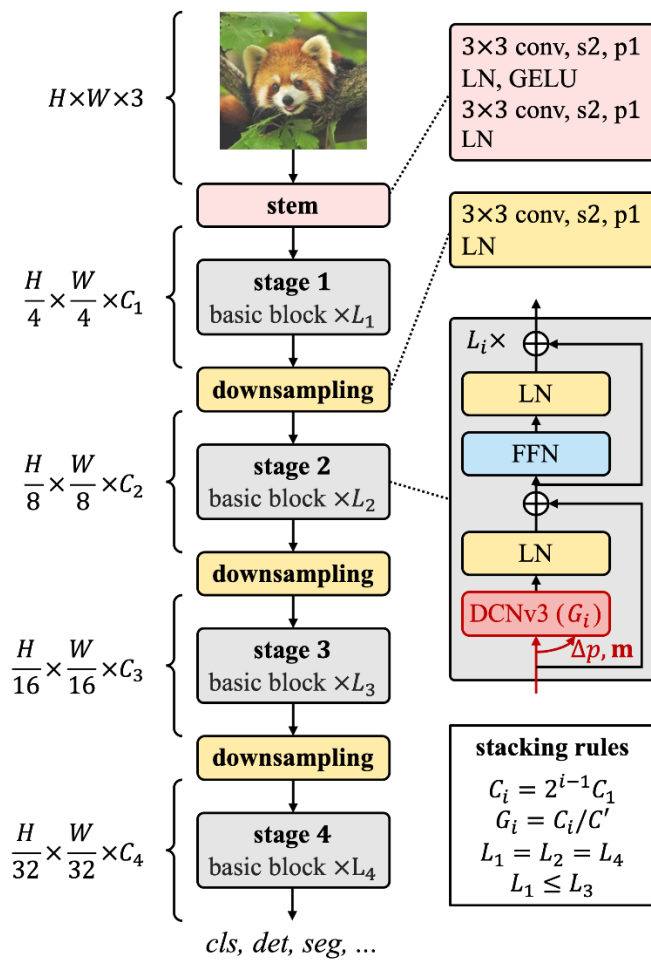


図 4-27 InternImage のイメージ

5) 【AL005】 3D 都市モデル表示

● 概要

- Web ブラウザ上で 3D 都市モデルをインタラクティブに可視化・操作するためのシステム。特別なソフトウェアのインストール不要でアクセス可能

● イメージ



図 4-28 3D 都市モデル可視化のイメージ

6) 【AL006】 RESTful API 通信

● 概要

- REST (Representational State Transfer) アーキテクチャスタイルの設計原則に準拠した Web API を利用してサーバーと通信を行う方法
- HTTP メソッド (GET/POST/PUT/DELETE 等) を用いて、資源 (Resource) を統一的な URI で扱うことで、拡張性・保守性が高いサービス構築が可能
- JSON を中心とした軽量データ交換形式を採用し、都市モデル生成、GIS データ取得、機械学習推論結果の受け渡しに適した通信方法

● 計算量

- リクエストデータサイズ：R
  - レスポンスデータサイズ：S
  - API 呼び出し回数：N
- $O(N \cdot (R + S))$

● イメージ

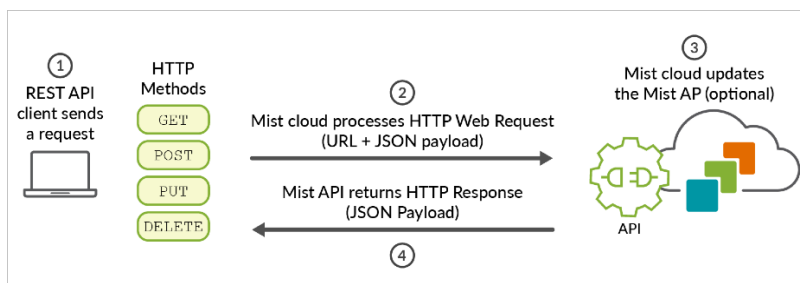


図 4-29 RESTful API 通信のイメージ

7) 【AL007】 Random Forest

● 概要

- 複数の決定木 (Decision Tree) を組み合わせて推論を行うアンサンブル学習モデル
- 各決定木は学習データからブートストラップサンプリングによってランダムに生成され、特徴量もランダムに選択して分岐を構築
- これにより、単一決定木に比べて 過学習の抑制、安定した分類性能、ノイズへの耐性を実現

● 計算量

- 決定木の本数を  $T$ 、各木の深さを  $D$ 、特徴量数を  $F$ 、学習サンプル数を  $N$  とすると：
  - 学習時： $O(T \cdot N \cdot F \cdot \log N)$
  - 推論時： $O(T \cdot D)$

● イメージ

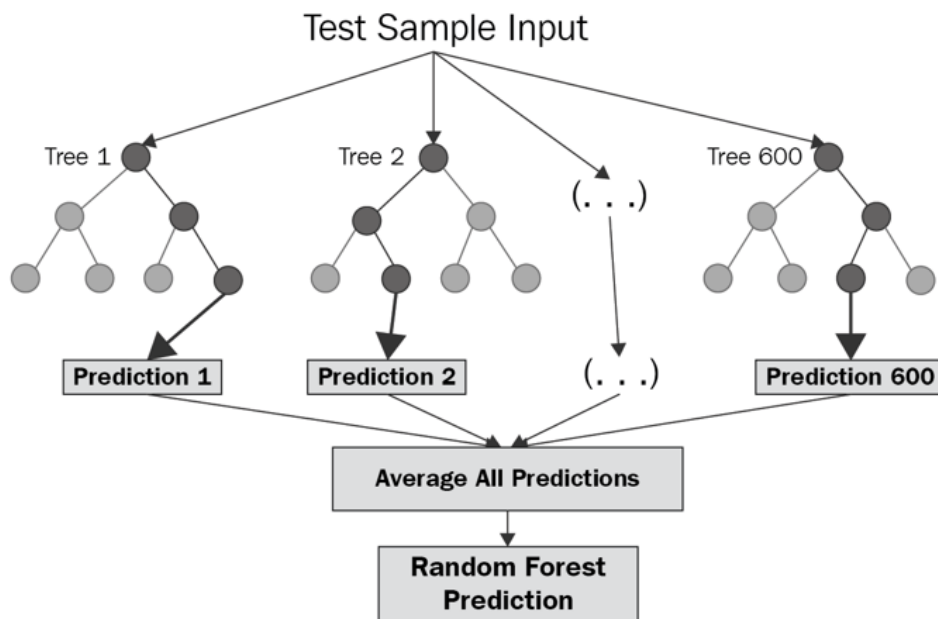


図 4-30 Random Forest のイメージ

8) 【AL008】 XGBoost

● 概要

- 勾配ブースティング (Gradient Boosting) を高速化・高性能化した決定木ベースのアンサンブル学習アルゴリズム
- Hessian を利用した近似的な最適化により、従来の GBDT より学習が高速かつ高精度
- 木構造の分岐計算を効率化する分散並列処理・ブロック構造化データフォーマットを採用
- 過学習を防ぐための正則化項 (L1・L2) を導入している点が大きな特徴

● 計算量

- 特徴量数を  $F$ 、学習サンプル数を  $N$ 、各ブースト木の深さを  $D$ 、木の本数を  $T$  とすると：
  - 学習時： $O(T \cdot N \cdot F \cdot \log N)$
  - 推論時： $O(T \cdot D)$

● イメージ

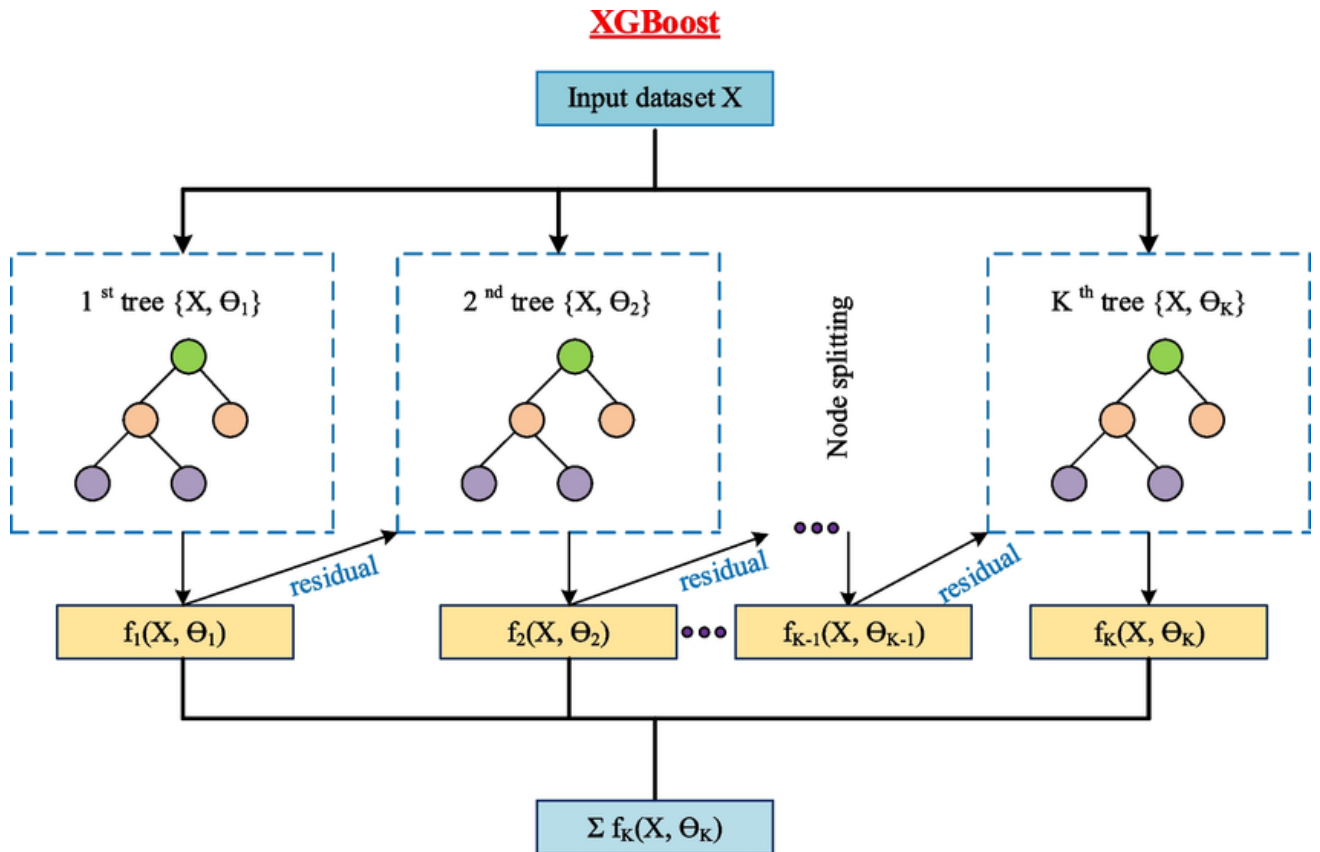


図 4-31 XGBoost のイメージ

9) 【AL009】 RANSAC

● 概要

- ノイズを多く含む観測データから、外れ値 (Outlier) に強いモデル推定を行うための反復的アルゴリズム
- データ集合から少数のサンプルをランダムに選び、仮のモデル (例: 直線、平面、ホモグラフィ、円など) を構築
- 残りのデータのうち、そのモデルに一致する点 (Inlier) の数を評価し、最も Inlier が多いモデルを採択

● 計算量

- 1 回の試行に必要なサンプル数を  $s$ 、総データ数を  $N$ 、必要な反復回数 (確率的に決まる) を  $k$  とすると:  
 $O(k \cdot s \cdot N)$

● イメージ

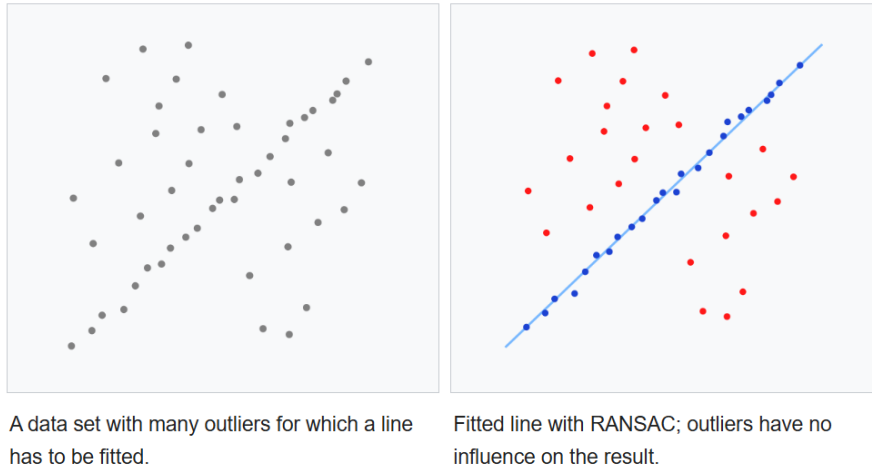


図 4-32 RANSAC のイメージ

10) 【AL010】射影変換

● 概要

- 射影変換とは、射影幾何学に基づき、画像の形状を変形させる変換手法の1つ
- 任意の形の四角形を別の形の四角形に変換するとき、各頂点の座標をそれぞれ対応付けして、変換係数を求める
- 三次元の射影変換の計算式は下記の式である

$$x' = \frac{a_1x + b_1y + c_1z + d_1}{a_0x + b_0y + c_0z + d_0}$$

$$y' = \frac{a_2x + b_2y + c_2z + d_2}{a_0x + b_0y + c_0z + d_0}$$

$$z' = \frac{a_3x + b_3y + c_3z + d_3}{a_0x + b_0y + c_0z + d_0}$$

● イメージ

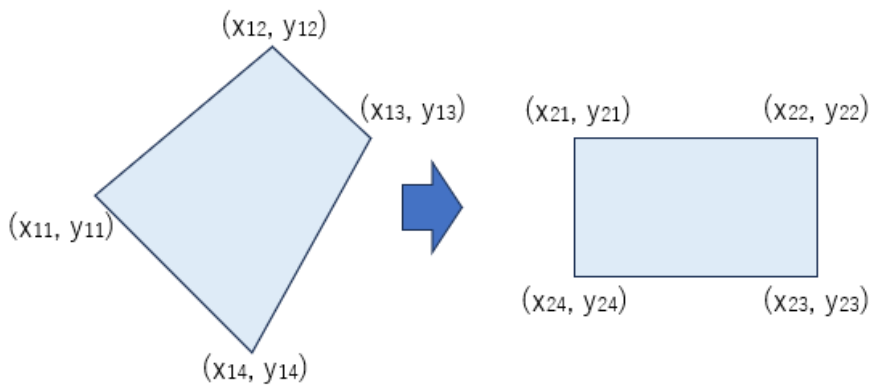


図 4-33 射影変換のイメージ (引用先：[https://qiita.com/ydclab\\_0003/items/1fc3d1468a39623bae60](https://qiita.com/ydclab_0003/items/1fc3d1468a39623bae60))

## 4-3-2. 開発したアルゴリズム

表 4-6 開発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
AL101	FN001	建築物フットプリントと汎用パラメータを用いた建築物モデルを生成するモデル	● 建築物フットプリントから建築物モデルを生成するための生成 AI モデル	● 深層学習に基づく生成モデルにより、多様な建築形状を学習データから獲得し、現実的な 3D モデルを効率的に生成することが可能であるため
AL102	FN007	ノイズ評価結果による建築物を繰り返しで生成するアルゴリズム	● 生成された建築物モデルのノイズ評価の結果により、設定された閾値を達するまで生成モデルを用いて数回の生成を行うアルゴリズム	● 閾値ベースの品質管理により、人手による品質チェックの負担を軽減し、システム全体の自動化を実現するため
AL103	FN001	衛星画像から屋根抽出機能：建築物ポリゴン正規化アルゴリズム	● インスタンスセグメンテーションアルゴリズムによって抽出された建築物輪郭はピクセルベースの不規則なマスクであり、3D モデル生成のための規則的なポリゴンデータに変換する	● ピクセルベースの不規則なセグメンテーション結果を、3D モデル生成に適した規則的なポリゴンデータに変換する前処理として不可欠である。ポリゴンの頂点数削減や直角化処理により、後の 3D モデル生成処理の精度と効率を向上させるため
AL104	FN002	ポリゴン化アルゴリズム	● セマンティックセグメンテーションアルゴリズムによって抽出された植生はピクセルレベルのラスターデータであり、3D モデル生成のためのベクトルデータに変換する	● ベクトルデータへの変換により、データ容量の削減と編集／加工の柔軟性を確保するため

AL105	FN002	OSM から道路データを抽出するアルゴリズム	<ul style="list-style-type: none"> <li>● 入力衛星画像から特定された実世界の座標範囲から道路データを抽出する</li> </ul>	<ul style="list-style-type: none"> <li>● 衛星画像の座標範囲に基づいて必要な道路データのみを効率的に抽出し、システム全体のデータ整合性を確保するため</li> </ul>
AL106	FN002	植生 3D モデル生成アルゴリズム	<ul style="list-style-type: none"> <li>● 植生ポリゴンから、CityGML への変換処理</li> </ul>	<ul style="list-style-type: none"> <li>● 植生の高さや樹冠形状を考慮した 3D 表現により、リアリティのある都市景観の再現を実現するため</li> </ul>
AL107	FN002 FN009	3D 道路モデル出力機能 (CityGML)	<ul style="list-style-type: none"> <li>● ユーザーが供与した道路幅や地表などのパラメータに基づいて、道路線データを分割したポリゴンに変換するアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>● ユーザー指定のパラメータ (道路幅員、地表高さ等) を反映した柔軟な道路モデル生成を可能にするため</li> </ul>
AL108	FN003	LOD2 建築物モデルからの壁面抽出アルゴリズム	<ul style="list-style-type: none"> <li>● LOD2 建築物モデルから、MMS の沿道画像に写っている壁面のみを抽出する処理</li> </ul>	<ul style="list-style-type: none"> <li>● LOD2 建築物モデルと MMS の沿道画像を関連付けるために必要</li> </ul>
AL109	FN003	画像からの開口部抽出アルゴリズム (2D 処理)	<ul style="list-style-type: none"> <li>● 沿道画像に対してオルソ画像化を行い、AI を用いて開口部を抽出する処理</li> </ul>	<ul style="list-style-type: none"> <li>● 沿道画像からの開口部抽出に必要</li> </ul>
AL110	FN003	三次元点群からの開口部抽出アルゴリズム (3D 処理)	<ul style="list-style-type: none"> <li>● 三次元点群に対して平面を抽出し、AI を用いて開口部を抽出する処理</li> </ul>	<ul style="list-style-type: none"> <li>● 三次元点群からの開口部抽出に必要</li> </ul>
AL111	FN003	画像からの開口部と点群からの開口部結合アルゴリズム	<ul style="list-style-type: none"> <li>● 2D 処理と 3D 処理それぞれの開口部抽出結果を結合する処理</li> </ul>	<ul style="list-style-type: none"> <li>● 2D 処理と 3D 処理を併用するために必要</li> </ul>
AL112	FN003	都市設備パラメータ推定アルゴリズム	<ul style="list-style-type: none"> <li>● 位置合わせされた点群とカメラ軌道データから、都市設備 (標識・信号) の位置・サイズのパラメータを推定する</li> </ul>	<ul style="list-style-type: none"> <li>● 点群データとカメラ軌道データを統合的に活用し、都市設備の三次元位置とサイズを高精度に推定することで、LOD3 レベルの都市モデル生成を効率化するため</li> </ul>

AL113	FN004	パラメータ設定機能： パラメータ解析アルゴリズム	<ul style="list-style-type: none"> <li>● ユーザーが提供したパラメータを都市 3D 生成コードに入力するために、ユーザー側のパラメータを解析する必要がある</li> </ul>	<ul style="list-style-type: none"> <li>● 入力パラメータの妥当性チェックとデフォルト値の補完により、システムの堅牢性とユーザビリティを向上させるため</li> </ul>
AL114	FN006, FN007, FN008	建築物モデル出力機能 (CityGML)	<ul style="list-style-type: none"> <li>● 建築物 (Trimesh) から、目標の CityGML への処理</li> </ul>	<ul style="list-style-type: none"> <li>● 生成された三角形メッシュを CityGML 形式に変換するため</li> </ul>
AL115	FN009	3D 道路モデル出力機能 (CityGML)	<ul style="list-style-type: none"> <li>● 入力した OSM の切り出しデータに対して、ユーザーが指定した道路幅員に基づき道路データを 3D 化し、各道路に用途属性を付与した上で、最終的に CityGML ファイルとして出力する</li> </ul>	<ul style="list-style-type: none"> <li>● 道路用途属性を付与し CityGML 形式で出力するため</li> </ul>
AL116	FN011	都市設備モデル生成： 都市設備配置アルゴリズム	<ul style="list-style-type: none"> <li>● PLATEAU の都市設備モデルの電柱と信号をテンプレートの対象として、入力パラメータに基づいて都市設備モデルを配置する</li> </ul>	<ul style="list-style-type: none"> <li>● テンプレートベースのアプローチにより、高品質な都市設備モデルを効率的に生成し、PLATEAU データとの整合性を確保するため</li> </ul>
AL117	FN012	3D 都市モデルの再構築:3D 都市モデル統合アルゴリズム	<ul style="list-style-type: none"> <li>● 道路データを基に各 3D 都市モデルを配置するアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>● 道路ネットワークを基準として各種 3D 都市モデル (建築物・植生・都市設備) を空間的に統合し、モデル間の位置整合性を確保するため</li> </ul>
AL118	FN013	3D 都市モデルの配置処理：建築物高さ生成アルゴリズム	<ul style="list-style-type: none"> <li>● 建築物フットプリントと都市計画決定情報に基づく 3D 都市モデルを配置するアルゴリズム</li> </ul>	<ul style="list-style-type: none"> <li>● 法規制情報 (容積率・建ぺい率等) を考慮することで、都市計画シナリオに沿った現実的な 3D 都市モデルの配置を実現するため</li> </ul>

AL119	FN017	3D 都市モデルのノイズ評価アルゴリズム	● 生成された建築物モデルに対するノイズ評価を行うアルゴリズム	● 面の向き不整合等の指標に基づき、生成モデルの幾何学的品質を定量的に評価し、品質改善処理への入力を提供するため
-------	-------	----------------------	---------------------------------	--

開発アルゴリズムの詳細を以下に示す。

1) 【AL101】 建築物フットプリントと汎用パラメータを用いた建築物モデルを生成するモデル

- 本アルゴリズムを利用する機能

➤ 【FN007】 【FN008】

- アルゴリズムの詳細

大規模言語モデル(LLM)を用いて、トークン化をした建築物フットプリントと指定された LOD レベル (LOD1、LOD2、LOD3) に応じて、それぞれ独立した詳細度のモデルを生成するアルゴリズムを開発した

1. 本システムは建築物生成に特化した大規模言語モデル (以下 LLM) を以下の手順で学習する：

- a) 入力処理：層状構造で表現された建築物モデルをシステムに入力する
- b) トークン離散化：建築物の幾何学的情報を表現する離散的なトークンに変換する
- c) マスキング：フットプリントマスクを適用し、特定の建築物の領域にモデルの注意を集中させる
- d) トークン変換：交差エントロピー損失関数 ( $\mathcal{L}_{ce}$ ) の制約の下、生成モデルを学習させる

2. 学習済みの LLM モデルは以下の手順で建築物モデルを生成する：

- a) フットプリント入力：建築物フットプリントを初期入力として使用する
- b) トークン生成：特化 LLM がフットプリントを処理し、建築物トークンを生成する
- c) 座標変換：生成されたトークンを三次元座標に逆変換する
- d) メッシュ構築：最終的に座標情報から建築物モデルのメッシュを構築する

3. パラメータ設定機能よりのパラメータ入力及び配置：

- a) 建築物生成の精細度設定により、設定可能なパラメータ種別を表示する (LOD1 には高さのみ、LOD2 には高さと屋根種類)
- b) LOD3 以下の場合には、ユーザーは表示されたパラメータ種別に値を入力し、建築物結果を生成させる
- c) LOD3 の場合には、生成された LOD2 建築物モデルと共に、FN003 より獲得された開口部情報と組み合わせて、開口部付きの LOD3 建築物結果を生成させる

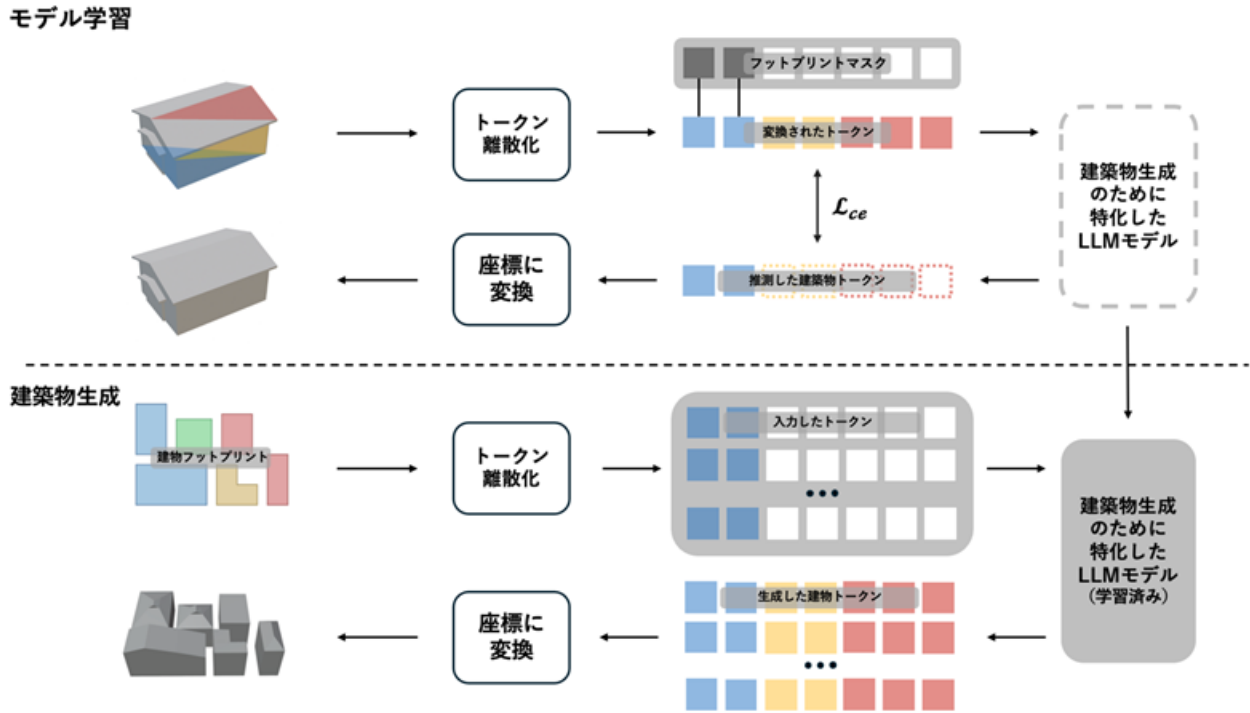


図 4-34 建築物生成モデルのフローチャット

2) 【AL102】 ノイズ評価結果による建築物を繰り返しで生成するアルゴリズム

- 本アルゴリズムを利用する機能
  - 【FN007】
- アルゴリズムの詳細

AL101 で生成された建築物モデルのノイズ評価の結果により、設定された閾値を達するまで生成モデルを用いて数回の生成を行って、外観にノイズを含まない建築物モデルを生成するアルゴリズムを開発した。

1. LLM による初期生成: 入力した建築物のフットプリントをトークン化し、指定された LOD レベル (LOD1、LOD2、LOD3) に応じて、特化 LLM で建築物モデルを生成する
2. 品質評価と判定: 生成されたモデルの屋根・側面・地面それぞれに対してノイズ評価を行い、設定された閾値と比較する
3. 反復精緻化と出力: 閾値に達していない場合は学習済み生成モデルで再生成を繰り返し、閾値を満たした建築物モデルを最終出力する

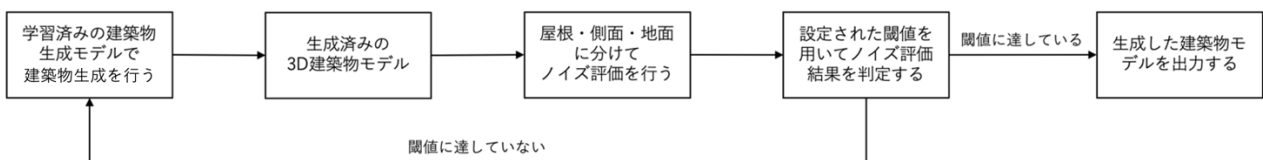


図 4-35 建築物モデルの繰り返し生成アルゴリズム

3) 【AL103】 衛星画像から屋根抽出機能：建築物のポリゴン正規化アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN001】

- アルゴリズムの詳細

抽出された建築物のフットプリント（ノイズのあるマスク）を入力し、マスクの外側の最小矩形を生成し、その内側にメッシュを生成する。 選択された全域から建築物外郭点を抽出し、冗長点を除去して建築物輪郭の最小点集合を形成する。 最後に建築物の輪郭は、建築物の方位軸を基準に再構築し、ベクトルポリゴンとして保存する。

1. ターニングポイント抽出: 抽出された建築物マスクを入力とし、建築物輪郭の方向変化が大きい特徴点（ターニングポイント）を検出する
2. 建築物輪郭再構築: 抽出されたターニングポイントを接続し、建築物の外形輪郭を再構築する
3. ポリゴン正規化と出力: 再構築された輪郭を幾何学的に正規化（直線化、直角補正など）し、正規化されたポリゴンとして出力する

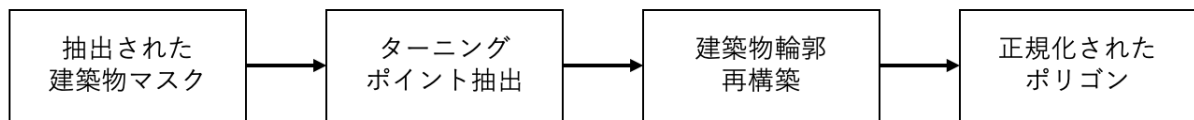


図 4-36 建築物ポリゴン正規化アルゴリズム

4) 【AL104】 ポリゴン化アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN002】

- アルゴリズムの詳細

空間データ処理ライブラリを用いて、ラスターデータをメモリ内ベクターデータソースに変換し、非ゼロピクセル領域をポリゴンに変換して出力するアルゴリズムを開発した。

1. 道路・植生のセグメンテーション結果ラスターを入力して、GDAL ライブラリの GetDriverByName を使用してメモリデータドライバーを作成し、このドライバーを通じてラスターデータセットを作成する
2. 次に、ラスターデータをこのデータセットに書き込む。OGR ライブラリの GetDriverByName('Memory') を使用して、メモリ内のベクターデータソースを作成し、ポリゴンオブジェクトを格納する
3. その後、gdal.Polygonize 関数を利用して、ラスターデータ内の非ゼロピクセル領域をポリゴンに変換し、それを保存及び出力する

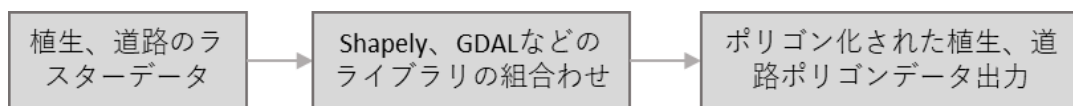


図 4-36 ポリゴン化アルゴリズム

5) 【AL105】 OSM から道路データを抽出するアルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN002】

- アルゴリズムの詳細
1. 入力された衛星画像からメタデータ又はジオリファレンス情報を解析し、画像がカバーする実世界の座標系（例：EPSG:4326／EPSG:3857 等）及び緯度・経度範囲を特定する
  2. この抽出された座標範囲を用いて、OpenStreetMap（OSM）の API へ問い合わせを行い、該当領域に含まれる道路リンクを取得する
  3. 取得されるデータは道路の中心線ジオメトリ、道路種別、付帯属性を含み、後続の 3D 化処理や CityGML モデル生成の基礎データとして利用される

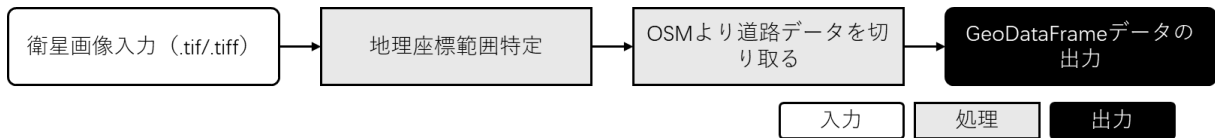


図 4-37 OSM から道路データを抽出するアルゴリズム

6) 【AL106】 植生 3D モデル生成アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN002】

- アルゴリズムの詳細

確定ポリゴンの範囲内でランダムな樹木ポジションをサンプリングし、PLATEAU 標準テンプレートから低木と高木を出力する形にするアルゴリズムを開発した。

1. 植生ポリゴンの範囲内、random ライブラリで樹木のポジションをサンプリング、PLATEAU の植生テンプレートから、ユーザー側から提供された植生種類割合によって、低木と高木を抽出し、サンプリングされた樹木のポジションで配置する
2. lxml ライブラリで PLATEAU 製品仕様書第 4.0 版に準拠して CityGML を出力



図 4-38 植生 3D モデル生成アルゴリズム

7) 【AL107】 3D 道路モデル出力機能（CityGML）

- 本アルゴリズムを利用する機能

➤ 【FN002】 【FN009】

- アルゴリズムの詳細

ユーザーが供与した道路幅や地表などのパラメータに基づいて、道路線データを分割したポリゴンに変換し、3D モデルの出力を整形するアルゴリズムを開発した。

1. 入力した OSM の切り出しデータに対して、ユーザーが指定した道路幅員に基づき道路中心線を左右方向へバッファリングし、道路面の形状を 3D メッシュとして生成する。生成された各道路セグメントには、

OSM の highway タグなどから抽出した 用途属性を付与し、道路種別に応じた構造的差異を保持する

2. さらに、生成した道路ジオメトリを指定の座標系(例:EPSG:4326 等)に変換した上で、CityGML の Road モジュールに準拠したデータ構造へマッピングし、LOD1-3 の 3D 道路モデルとして CityGML ファイルに出力する

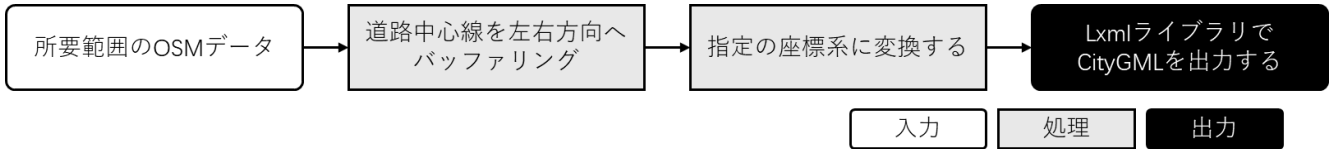


図 4-39 3D モデル出力機能 (CityGML)

8) 【AL108】 【AL108】 LOD2 建築物モデルからの壁面抽出アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN003】

- アルゴリズムの詳細

LOD2 建築物モデルから、MMS の沿道画像に写っている壁面のみを抽出するアルゴリズムである。そして、抽出した全ての壁面に対して、カメラの撮影位置や傾きを用いて MMS の画像の中から最適な画像を対応付けする処理も有する。

1. MMS の走行履歴から一定距離以内の壁面のみ抽出する
2. 1.で抽出した壁面に対し、カメラの撮影位置や傾きを用いて、カメラに写っている壁面のみを残す。その際、オクルージョン処理として、カメラに写っている部分と写っていない部分が混在する壁面は、分割処理を行う
3. 2.で抽出した壁面全てに対して、カメラの撮影位置や傾きを用いて、適切と思われる画像を 2 枚 (近い距離と遠い距離) を対応付ける

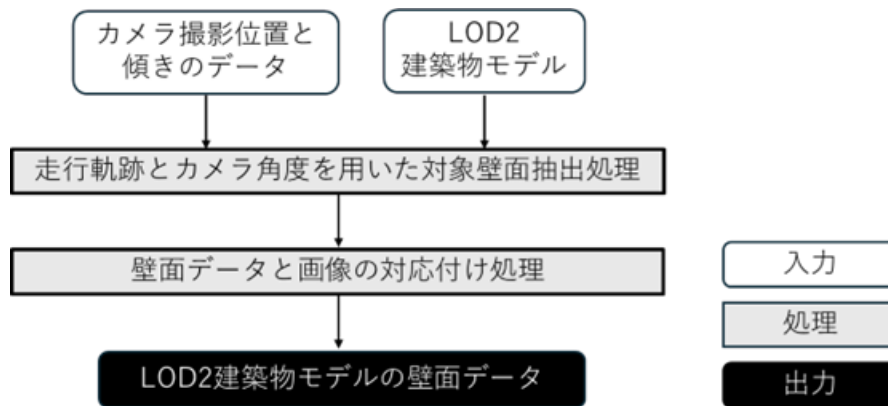


図 4-40 LOD2 建築物モデルからの壁面抽出アルゴリズム

9) 【AL109】 画像からの開口部抽出アルゴリズム (2D 処理)

- 本アルゴリズムを利用する機能

➤ 【FN003】

- アルゴリズムの詳細

【AL108】で対応付けた LOD2 建築物モデルの壁面と 2 枚の画像を用いて、建築物の壁面オルソ画像を作成し、深層学習を用いてオルソ画像内の開口部を抽出する。壁面オルソに対して深層学習を適用している点が、本アルゴリズムの特徴であり、精度的にも非常に重要なポイントである。

1. MMS データに付随しているカメラの撮影位置と傾きのデータを用いて、LOD2 建築物モデルの壁面と 2 枚の画像との数学的な関係を計算する
2. 1.で取得した関係式を用いて、【AL015】の射影変換を行い壁面のオルソ画像を作成する
3. 2.で作成したオルソ画像に対して、【LB065】の YOLO を用いた深層学習で、開口部を抽出するアルゴリズムを作成する

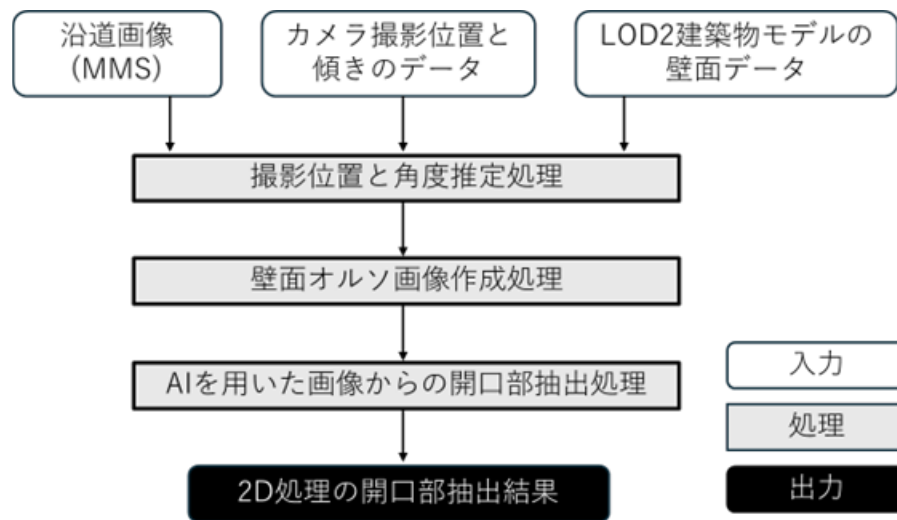


図 4-41 画像からの開口部抽出アルゴリズム (2D 処理)

10) 【AL110】三次元点群からの開口部抽出アルゴリズム (3D 処理)

- 本アルゴリズムを利用する機能
  - 【FN003】
- アルゴリズムの詳細

【AL108】で抽出した LOD2 建築物モデルの壁面と MMS の三次元点群に対して、深層学習を用いて開口部を抽出する。三次元点群に切り取り処理などを行うことで壁面の画像を作成し、画像処理用の深層学習を用いて実装している点が特徴である。

1. 三次元点群データ (MMS) と壁面 (LOD2 建築物モデル) の重ね合わせを行う。具体的には、平面の平行移動や回転といった補正処理を実装することで、高精度な重ね合わせを実現している
2. 1.で LOD2 建築物モデルと重なった三次元点群に対して、【AL014】の RANSAC を用いて平面の点群を切り取る。切り取り幅に関しては、平面の奥側は小さく切り取り、手前側は大きく切り取ることで高精度化を図っている
3. 2.で作成したオルソ画像に対して、【LB065】の YOLO を用いた深層学習で、開口部を抽出するアルゴリズムを作成する

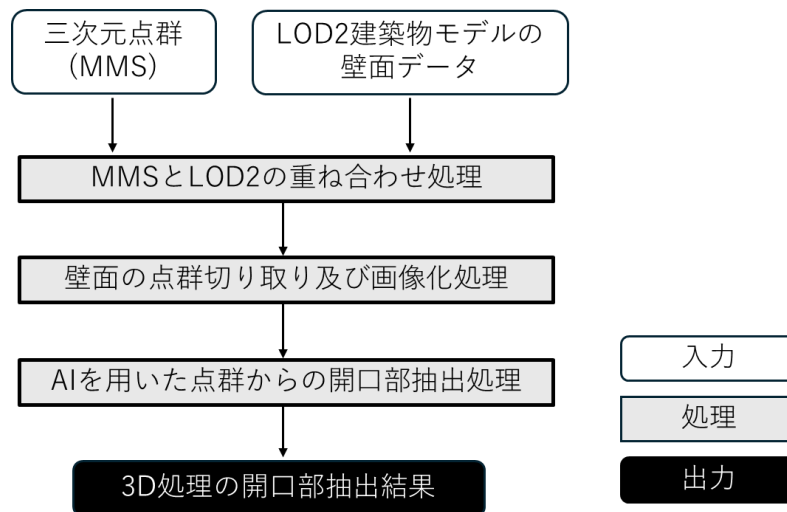


図 4-42 三次元点群からの開口部抽出アルゴリズム (3D 処理)

11) 【AL111】 画像からの開口部と点群からの開口部結合アルゴリズム

- 本アルゴリズムを利用する機能
  - 【FN003】
- アルゴリズムの詳細

【AL109】の2D処理の開口部抽出結果と【AL110】の3D処理の開口部抽出結果を結合して、LOD3を作成するための「開口部情報 (GeoJSON形式)」を作成するアルゴリズムである。

1. 2D処理の開口部抽出結果と3D処理の開口部抽出結果は、両方画像上の座標である。そして、それぞれの画像の縦横比は、LOD2建築物モデルの壁面のサイズを用いて決定している。そのため、スケール変換するだけで、2Dと3Dの結果を重ね合わせることができる
2. 重ね合わせた後、抽出した結果が重複している箇所の判定処理を行う。形状に関しては3D処理の結果を優先し、ラベルに関しては2D処理の結果を優先する
3. 2の結果をLOD2建築物モデルの座標系に変換し、LOD3を作成するための「開口部情報 (GeoJSON形式)」を作成する

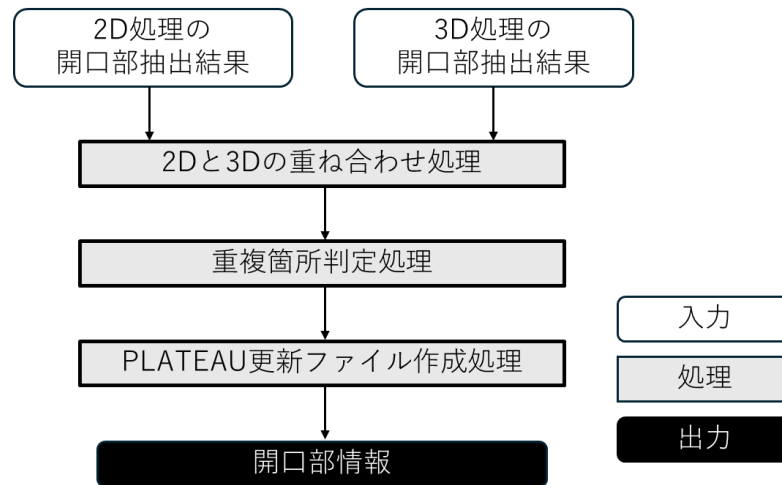


図 4-43 画像からの開口部と点群からの開口部結合アルゴリズム

12) 【AL112】 都市設備パラメータ推定アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN003】

- アルゴリズムの詳細

【AL107】で得られたセグメンテーション画像の標識・信号領域に対して連結成分を抽出し、標識・信号それぞれの矩形を求める。

1. 【AL110】で得られた各画像内の矩形を隣接画像間でマッチングを行い、矩形のトラッキングを行う。トラッキングでは、画像ごとに三次元点群を参照し矩形の大まかな三次元位置を推定する。位置を推定した矩形を隣接するカメラに投影してその画像内の矩形と比較することで、画像間の矩形のマッチングを行う。隣接する全ての画像ペアについて矩形のマッチングを行い、対応関係を集約することで矩形のトラッキング結果が得られる
2. トラッキングした各矩形について、位置とサイズを推定する。まず矩形を観測したカメラ列の中から、距離が近く観測した点群数が多いものを代表カメラとして選択する。点群と代表カメラの距離を推定し、代表カメラのカメラ姿勢を基に矩形の四隅の点を推定した距離の位置に設定する。この4点から矩形の三次元空間での位置とサイズを推定する

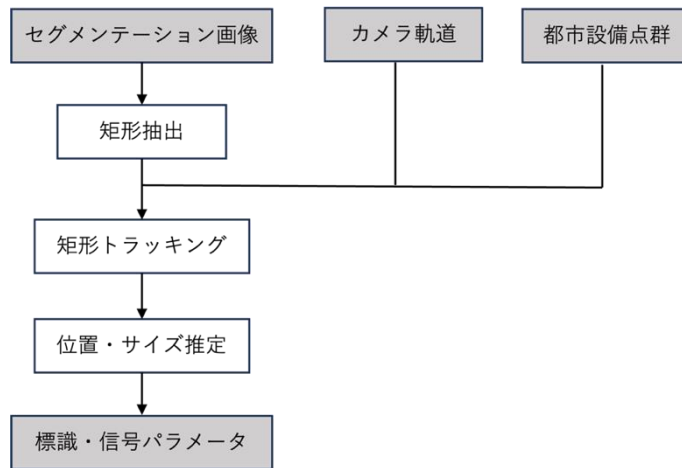


図 4-44 信号・標識パラメータ推定

13) 【AL113】パラメータ設定機能：パラメータ解析アルゴリズム

- 本アルゴリズムを利用する機能
  - 【FN004】
- アルゴリズムの詳細

OpenCV と GDAL ライブラリを使用して入出力画像を正規化してネットワークに入力、上記の SL025-037 のライブラリを使用して Stable Diffusion と ControlNet ネットワークを定義し、上記の図に示されたパラメータに基づいて、pytorch-lightning を使用して学習プロセスを制御。学習済みのモデルは、LOD0（建築物のフットプリントポリゴン）を入力として受け取り、建築物の 2D 画像を出力する処理を行う。

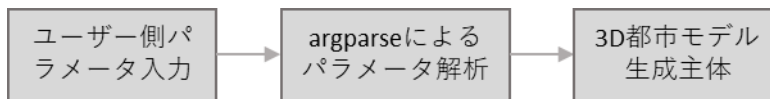


図 4-45 パラメータ解析アルゴリズム

14) 【AL114】建築物モデル出力機能（CityGML）

- 本アルゴリズムを利用する機能
  - 【FN006】【FN007】【FN008】
- アルゴリズムの詳細

Trimesh から numpy ライブラリで Trimesh 内のポイントを分類し、ユーザー側から提供されたパラメータによって、lxml ライブラリで PLATEAU 製品仕様書第 4.0 版に準拠して CityGML を出力。

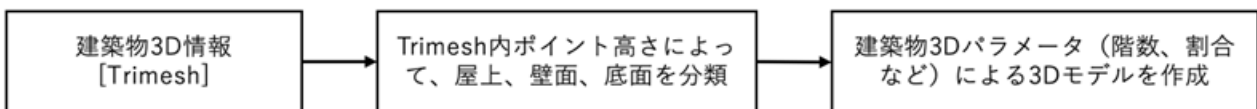


図 4-46 3D モデル出力機能（CityGML）

15) 【AL115】道路モデル生成（仮想都市）：仮想都市道路生成アルゴリズム

- 本アルゴリズムを利用する機能

➤ 【FN010】

● アルゴリズムの詳細

Shapely ライブラリを利用して、PLATEAU の 3D 道路モデル (LOD0) を LineString タイプに変換し、幅広い PLATEAU の 3D 道路モデルを仮想都市道路生成のためのソースデータベースとして利用する。Random ライブラリを使用してユーザー入力のランダムシードを設定し、このランダムシードに基づいて道路ソースデータベース内の切り取り範囲を決定する。該当する範囲の道路データを切り取った後、Shapely ライブラリを使って、ユーザーが提供するパラメータに基づいて LineString 道路を Polygon タイプに変換する。最後に、Ixml ライブラリで PLATEAU 製品仕様書第 4.0 版に準拠して CityGML を出力。

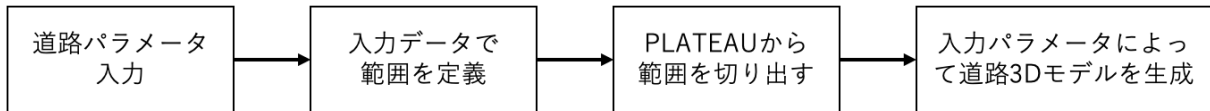


図 4-47 仮想都市道路生成アルゴリズム

16) 【AL116】 都市設備モデル生成：都市設備配置アルゴリズム

● 本アルゴリズムを利用する機能

➤ 【FN011】

● アルゴリズムの詳細

PLATEAU の都市設備モデルの電柱と信号をテンプレートの対象として、入力パラメータに基づいて都市設備モデルを配置する。道路 LineString 上にバッファを作成し、Shapely ライブラリを使用して一定の距離ごとに装置の位置をサンプリングする。ユーザーが提供した都市設備の種類や割合などのパラメータに基づいて PLATEAU の都市設備モデルのテンプレートから対応する 3D モデルを抽出し、numpy と trimesh ライブラリを使用して上述のサンプリングされた設備位置にモデルを配置する。最後に、Ixml ライブラリで PLATEAU 製品仕様書第 4.0 版に準拠して CityGML を出力。

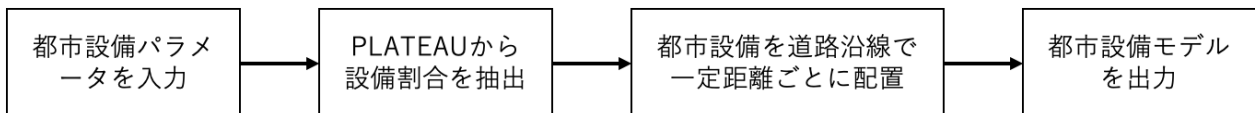


図 4-48 都市設備配置アルゴリズム

17) 【AL117】 3D 都市モデルの再構築：3D 都市モデル統合アルゴリズム

● 本アルゴリズムを利用する機能

➤ 【FN012】

● アルゴリズムの詳細

ユーザーが設定したランダムシードに基づいて決定された空間範囲で、PLATEAU テンプレートから切り出された地形、道路、建築物、設備、植生の Trimesh 情報を、順番に目標の.obj に追加し、追加の過程で shapely ライブラリを使用して空間関係を判断し、これらの地物のトポロジー関係が矛盾しないように保証し、最終的に.obj ファイルを出力。

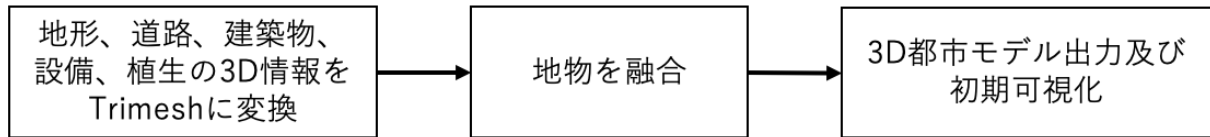


図 4-49 3D 都市モデル統合アルゴリズム

18) 【AL118】 3D 都市モデルの配置処理：建築物高さ生成アルゴリズム

- 本アルゴリズムを利用する機能

- 【FN013】

- アルゴリズムの詳細

建築物フットプリントと都市計画決定情報に基づく 3D 都市モデルを配置するアルゴリズムである。実際の都市に対して、生成する空間の範囲を決定した後、建築物のフットプリント、道路データ及び実際の都市の都市計画決定情報を取得し、個々の建築物の高さを自動生成する。

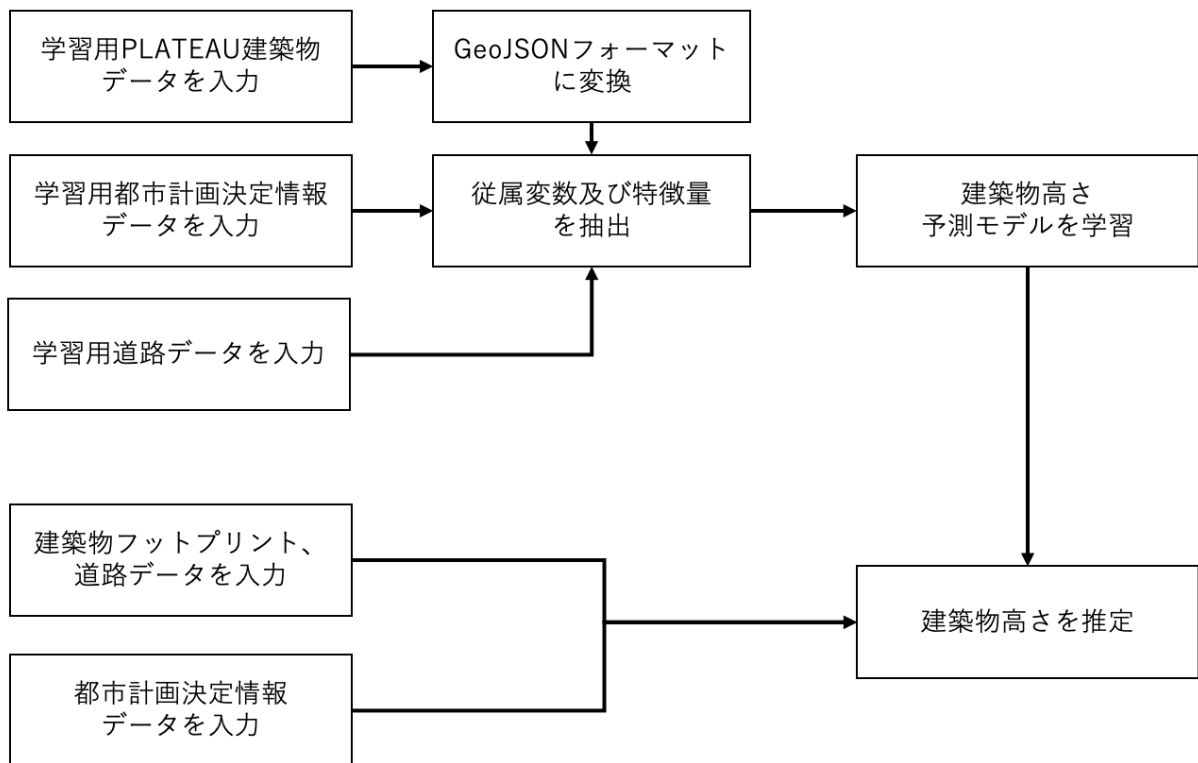


図 4-50 建築物高さ生成アルゴリズム

## 4-4. データインタフェース

### 4-4-1. ファイル入力インタフェース

#### 1) 【IF001】衛星画像ファイル入力

3D 都市モデル生成に関わるパラメータ推定に必要な衛星画像は tif 画像又は tiff 画像とする。本インタフェースでは、オルソ補正済み衛星画像を入力データとして受け付けるものとする。入力画像は、PLATEAU 標準製品仕様書に準拠した 3D 都市モデル生成を想定し、空間的整合性及び解析精度を確保できる品質を満たす必要がある。衛星画像の空間解像度は地上画素寸法 0.3m を用いる。画像は RGB バンドを必須とし、幾何補正が施されたオルソ画像である。座標参照系は建物フットプリントデータと一致している必要がある。入力画像に雲や極端な影、露出過多領域が広範囲に存在する場合、建築物モデル生成の精度に影響を及ぼす可能性がある。また、フットプリントデータと取得時期が大きく異なる場合には、現況との差異により生成結果に不整合が生じる場合がある。さらに、位置補正が不十分な場合には、生成された 3D 都市モデルに位置ズレが発生する可能性がある。

- 本インタフェースを利用する機能

- 【FN001】【FN002】



図 4-51 衛星画像の例

#### 2) 【IF002】沿道画像ファイル入力

3D 都市モデル生成の開口部に関わるパラメータ推定に必要な MMS で撮影した沿道画像は JPG 形式とする。本インタフェースでは、建築物の立面情報及び開口部形状の推定に利用する沿道画像を入力データとする。入力する沿道画像は、建築物の外壁面が十分に視認可能な品質を有している必要がある。沿道画像は走行しながら撮影された連続画像であり、各画像は撮影位置およびカメラ姿勢データと対応付けることにより、三次元点群データと統合的に利用する。

- 本インタフェースを利用する機能

- 【FN003】



図 4-52 沿道画像の例

3) 【IF003】 三次元点群ファイル入力

3D 都市モデル生成の開口部に関わるパラメータ推定に必要な MMS で取得した三次元点群データは、LAS 形式とする。入力する三次元点群データは、建築物立面を十分に観測できる密度及び精度を有している必要がある。具体的には、対象建築物の外壁形状や開口部の位置及び形状が識別可能となる点群密度を確保することが望ましく、建築物立面に対して概ね数百点/m<sup>2</sup>程度以上の密度を有することが推奨される。また、三次元点群の位置精度は、建築物形状の再現性を確保する観点から、水平精度及び高さ精度ともに概ね 10cm 程度以内であることが望ましい。点群データの座標参照系は、他の入力データ（衛星画像及び建築物フットプリントデータ）と一致させる必要がある。また、車両や植栽等による遮蔽が多い場合や、建築物立面が十分に観測されていない場合には、開口部推定精度が低下する可能性があるため留意する必要がある。

● 本インターフェースを利用する機能

➤ 【FN003】

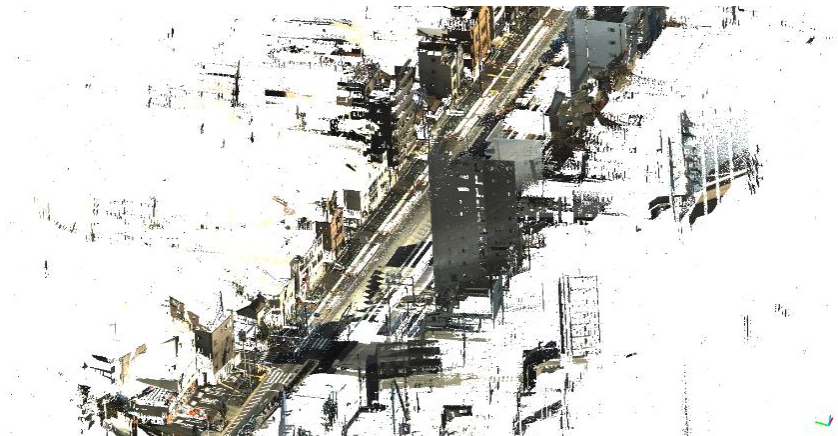


図 4-53 MMS 点群の例

4) 【IF004】 LOD2 建築物モデルファイル入力

3D 都市モデル生成の開口部に関わるパラメータ推定に必要な LOD2 の建築物モデルは、CityGML 形式とする。

- 本インターフェースを利用する機能
  - 【FN003】

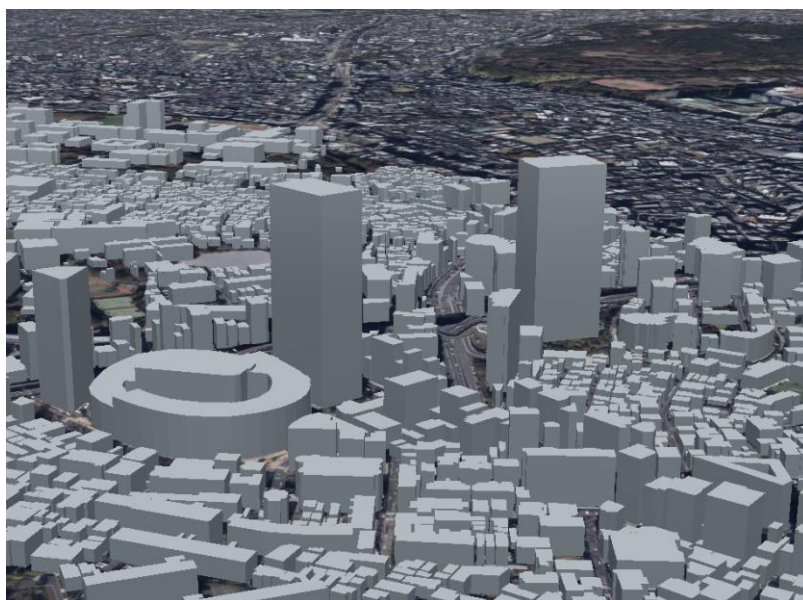


図 4-54 LOD2 建築物モデルファイルの例

5) 【IF005】カメラ撮影位置と傾きのデータ (CSV ファイル入力)

カメラの撮影位置の CSV ファイルと、カメラの傾きの CSV ファイルの 2 つが存在する。撮影位置 CSV ファイルには、各画像に対応する撮影時刻および三次元位置情報が記録される。また、画像ファイル名および GPS 時刻は、対応する画像データおよび姿勢データとの紐付けを行うために使用される。姿勢 CSV ファイルには、各撮影画像に対応するカメラの姿勢情報を記録する。姿勢情報はロール (Roll)、ピッチ (Pitch)、ヘディング (Heading) の 3 軸回転角として記録し、単位は度 (deg) とする。これらの姿勢情報は、三次元空間上でのカメラ視線方向を推定するために利用され、画像と三次元点群の対応付けおよび開口部位置推定に使用される。撮影位置データおよび姿勢データは、同一の画像ファイル名および GPS 時刻により対応付けられる必要がある。また、これらのデータの座標参照系および時刻系は、入力する三次元点群データおよび画像データと整合している必要がある。

- 本インターフェースを利用する機能
  - 【FN003】 開口部情報取得

表 4-7 カメラ撮影位置 CSV ファイルの例

行番号	画像ファイル名	GPS 時刻	Y 座標(m)	X 座標(m)	標高(m)
0	01_2023100610314938	437513.4	-10242.508	-24042.376	5.557
1	01_2023100610315060	437514.6	-10241.144	-24040.905	5.549
...	...	...	...	...	...

表 4-8 カメラ傾き CSV ファイルの例

行番号	画像ファイル名	GPS 時刻	ロール(deg)	ピッチ(deg)	ヘディング(deg)
0	01_2023100610314938	437513.4	-0.724970920	-0.568303540	42.974443040
1	01_2023100610315060	437514.6	-0.682293890	-0.611150840	42.865533280
...	...	...	...	...	...

6) 【IF006】 建築物フットプリントデータ入力

建築物フットプリントデータは、3D 都市モデルにおける建築物モデル生成機能に必要な入力データであり、建築物の平面形状を定義する基礎データとして利用される。本インタフェースでは、LOD0 の建築物フットプリントデータを GeoJSON 形式のファイルとして入力するものとする。入力する建築物フットプリントデータは、各建築物の外形を表すポリゴンデータとして記録され、建築物の平面形状を表す閉合した座標列により構成されている必要がある。GeoJSON ファイルには、建築物の識別情報や属性情報を含めることができるが、ユーザーが入力するデータによって属性の種類や項目は異なる。

- 本インタフェースを利用する機能

➤ 【FN006】

表 4-9 建築物属性の例

建築物番号	面積	---	階数
3207000291641	10	---	1



図 4-55 建築物フットプリント（ピンク色）の例

7) 【IF007】 オープンストリートマップデータ

最終的な道路の CityGML ファイルを生成するためには、【FN002】 から取得した OSM 道路データを入力する必要があります。

- 本インタフェースを利用する機能

➤ 【FN009】



図 4-56 オープンストリートマップデータの例

- 8) 【IF008】 衛星画像 (OSM から範囲内の道路を抽出・植生判別)  
衛星画像範囲内の道路データを抽出し、植生範囲を判別してテンプレートを付ける。データ仕様は【IF001】に準拠する。
- 本インターフェースを利用する機能
    - 【FN002】

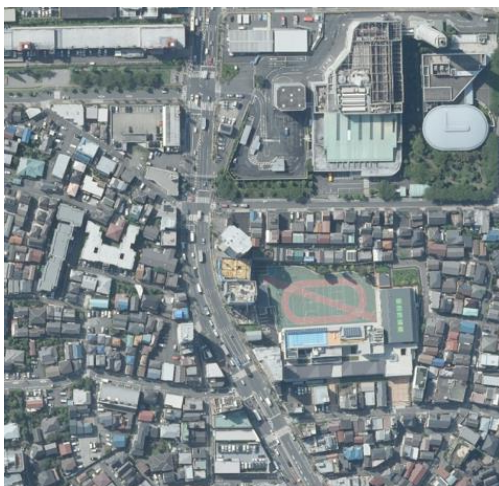


図 4-57 衛星画像データの例

- 9) 【IF009】 3D 都市モデル (建築物の学習)  
三次元都市モデルを生成するモデルの学習のための PLATEAU の建築物モデルである。
- 本インターフェースを利用する機能
    - 【FN005】



図 4-58 建築物モデルデータの例

10) 【IF010】 3D 都市モデル (3D 都市モデルテンプレート作成)

3D 都市モデルテンプレート作成は、植生および都市設備を含む都市空間要素を 3D 都市モデル内に再現するために必要な入力データとして利用される。テンプレートデータには、都市空間における植生分布状況や、道路周辺に設置される都市設備（電柱、街灯、交通信号機等）を再現するための 3 次元形状情報が含まれる。これらのデータは、PLATEAU の CityGML データから該当要素を抽出し、3D 都市モデル生成処理において利用しやすい形式として整理されたものである。本システムでは、テンプレートデータを 3D 形状モデルとして OBJ 形式で保存し、各テンプレートに対して実際の高さ等の属性情報を付与したうえで、プロジェクト内のモデルファイルとして管理する。生成処理では、これらのテンプレートデータを参照し、対象地域の都市環境に応じた植生および都市設備の 3D モデルを配置することで、3D 都市モデルの表現精度を向上させる。

- 本インターフェースを利用する機能

➤ 【FN018】

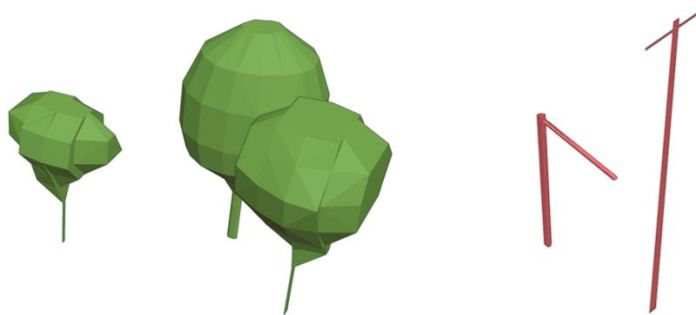


図 4-59 3D 都市モデルテンプレートデータの例

11) 【IF011】 都市計画決定 GIS データ

都市計画決定 GIS データは、建築物の高さ推定処理に必要な入力データとして利用される。本インターフェースでは、都市計画に基づく用途地域や高度地区等の情報を含む GIS データを入力データとする。都市計画決定 GIS データには、都市計画区域内における用途地域、高度地区、容積率、建ぺい率等の規制情報が含まれており、これらの情報は建築物高さの推定処理において参考情報として利用する。

- 本インターフェースを利用する機能

➤ 【FN008】

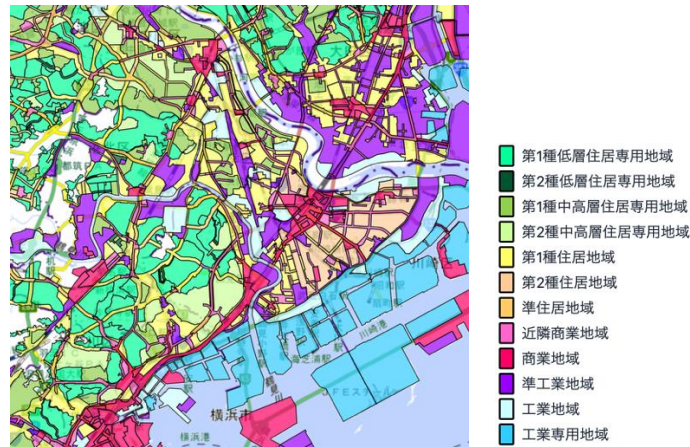


図 4-60 都市計画決定 GIS データの例 (用途地域)

12) 【IF012】 数値標高モデル

数値標高モデルは、3D 都市モデル再構築において地形モデルを生成するために必要な入力データとして利用する。本インターフェースでは、国土地理院が提供する DEM データを入力データとして用い、対象地域の地表標高情報を取得する。入力する数値標高モデルは空間分解能 1m の DEM データとし、GeoTIFF (.tif) 形式のファイルとして保存されたデータを利用する。本システムでは当該 DEM データをプロジェクトのモデルファイル内に保存して管理し、3D 都市モデル生成処理の際に参照する。生成処理では DEM データを三角化することで地形の三次元形状を生成し、3D 都市モデルの基盤となる地形モデルとして利用する。

- 本インターフェースを利用する機能

➤ 【FN012】



図 4-61 数値標高モデルデータの例

## 4-4-2. ファイル出力インターフェース

- 1) 【IF101】 3D 都市モデルデータ
  - 本インターフェースを利用した機能
    - 【FN012】

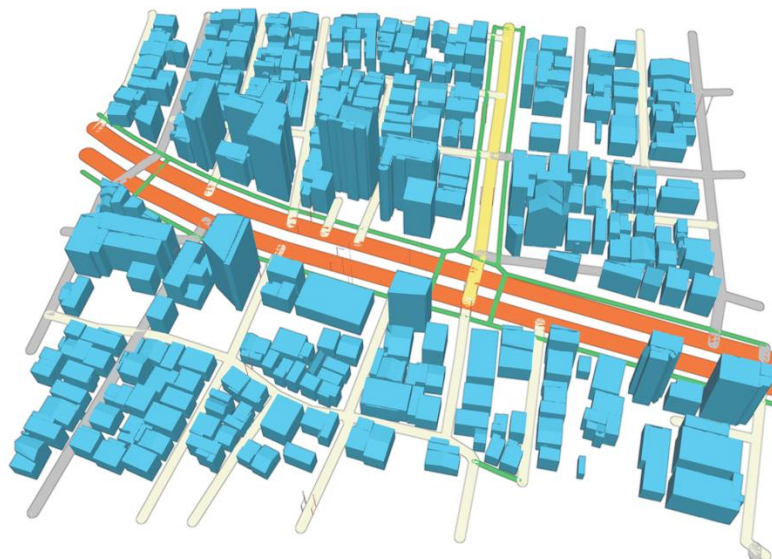


図 4-62 生成された 3D 都市モデルデータの例

表 4-10 出力ファイル一覧 CityGML ファイル出力

地物名	LOD
建築物	0, 1, 2, 3
道路	0, 1, 2, 3
植生	0, 1, 2, 3
都市設備	0, 1, 2, 3

## 4-4-3. 内部連携インターフェース

- 1) 【IF201】 建築物フットプリント／屋根種別抽出
  - 本インターフェースを利用した機能
    - 【FN001】



図 4-63 抽出された建築物フットプリントデータの例

表 4-11 屋根種別抽出結果の例

建築物 ID	屋根タイプ
Bldg_01	1
Bldg_02	4
...	...
Bldg_0x	2

- 2) 【IF202】 開口部情報取得
  - 本インターフェースを利用した機能
    - 【FN003】

```

{
  "id": "bldg_b42da902-c1b5-48e9-bc29-
  fdffb87bfa76",
  "height": 28.357,
  "footprint": [
    [
      -9716.299999973431,
      -24092.32000018203
    ],
    [
      -9733.599999975173,
      -24088.650000181377
    ],
    [
      -9732.649999972165,
      -24084.200000180484
    ],
    [
      -9726.789999975406,
      -24080.48000018153
    ],
    [
      -9714.339999974674,
      -24083.120000180537
    ],
    [
      -9716.299999973431,
      -24092.32000018203
    ]
  ],
  "faces": [
    {
      "eid": 0,
      "elems": [
        {
          "type": "window",
          "x,y": [
            0.780332234930894,
            0.7795958669816976
          ],
          "w,h": [
            0.024031971003306517,
            0.03879112741122122
          ]
        },
        {
          "type": "balcony",
          "x,y": [
            0.47922577471299466,
            0.4428183517297316
          ],
          "w,h": [
            0.3661341464621405,
            0.06788447296963712
          ]
        }
      ]
    }
  ]
}

```

図 4-64 開口部情報データの例

3) 【IF203】衛星画像からの植生判別

- 本インターフェースを利用した機能
  - 【FN002】



図 4-65 植生判別データの例 (マスク)

4) 【IF204】OSM からの道路抽出

- 本インターフェースを利用した機能
  - 【FN003】



図 4-66 道路抽出データの例

5) 【IF205】 建築物高さ

- 本インターフェースを利用した機能

➤ 【FN013】

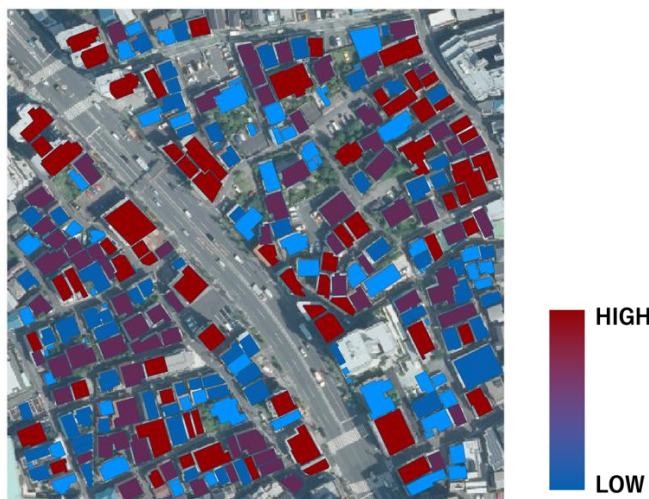


図 4-67 建築物高さデータの例

6) 【IF206】 パラメータ設定機能

- 本インターフェースを利用した機能

➤ 【FN004】



図 4-68 パラメータ設定画面の例

7) 【IF207】 建築物生成モデル

- 本インターフェースを利用した機能
  - 【FN005】

8) 【IF208】 都市モデルテンプレート

- 本インターフェースを利用した機能
  - 【FN018】



図 4-69 3D 都市モデルテンプレートの例 (植生・設備)

9) 【IF209】 LOD1 建築物モデル

- 本インターフェースを利用した機能
  - 【FN006】

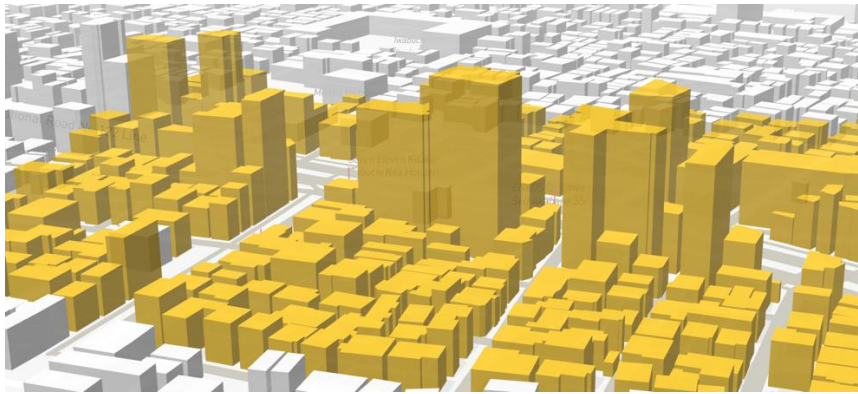


図 4-70 LOD1 建築物モデルの例

10) 【IF210】 LOD2 建築物モデル

- 本インターフェースを利用した機能

➤ 【FN007】

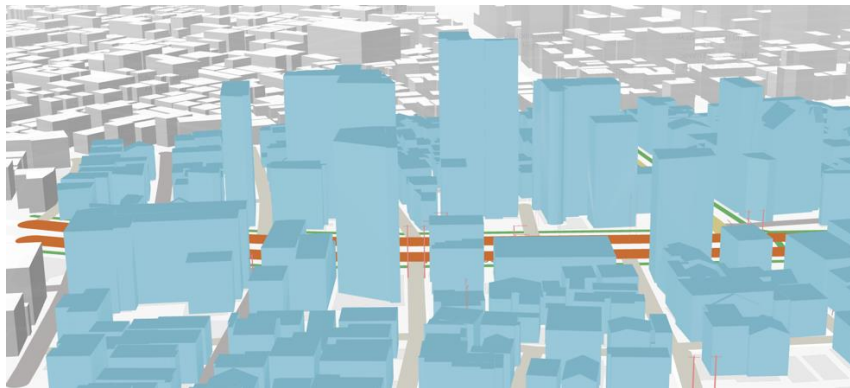


図 4-71 LOD2 建築物モデルの例

11) 【IF211】 建築物モデル BMQI

- 本インターフェースを利用した機能

➤ 【FN017】

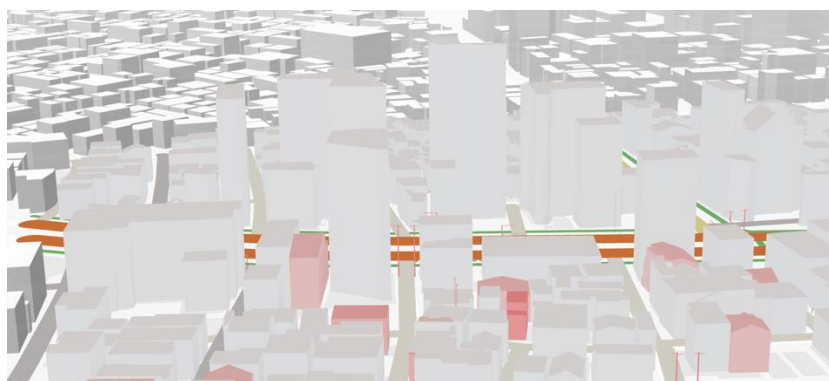


図 4-72 建築物モデル BMQI の例

12) 【IF212】 LOD3 建築物モデル

- 本インターフェースを利用した機能
  - 【FN008】

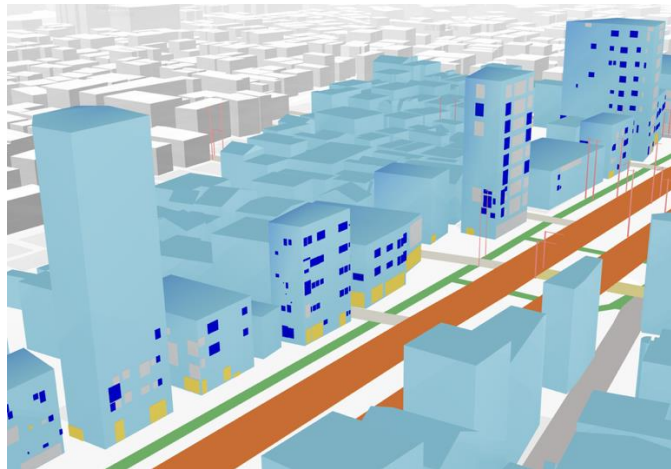


図 4-73 LOD3 建築物モデルの例

13) 【IF213】 植生モデル

- 本インターフェースを利用した機能
  - 【FN009】



図 4-74 植生モデルの例

14) 【IF214】 道路モデル

- 本インターフェースを利用した機能
  - 【FN010】

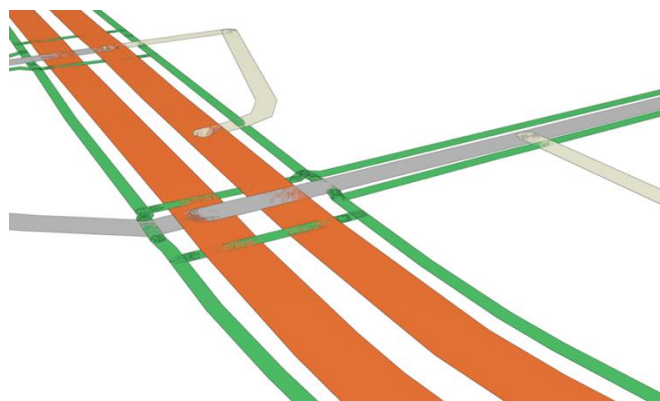


図 4-75 道路モデルの例

15) 【IF215】 都市設備モデル I

- 本インターフェースを利用した機能
  - 【FN011】

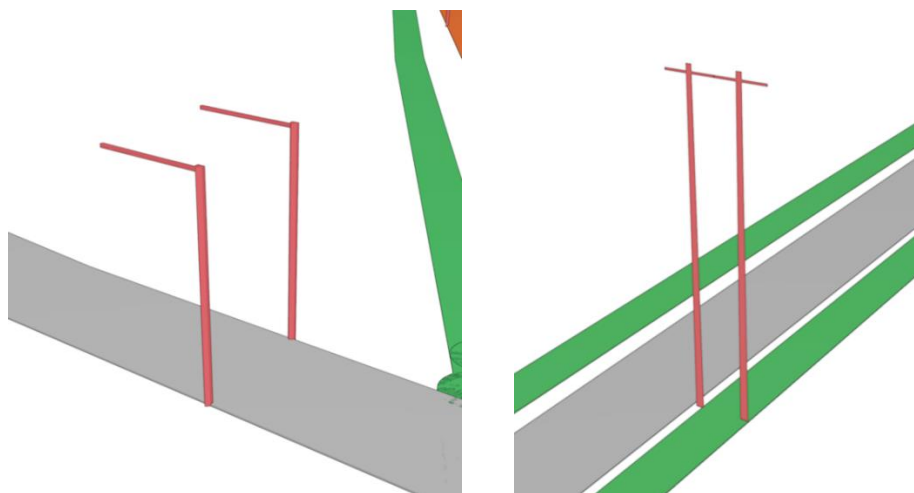


図 4-76 都市設備モデルの例

16) 【IF216】 ノイズ削減データ

- 本インターフェースを利用した機能
  - 【FN015】

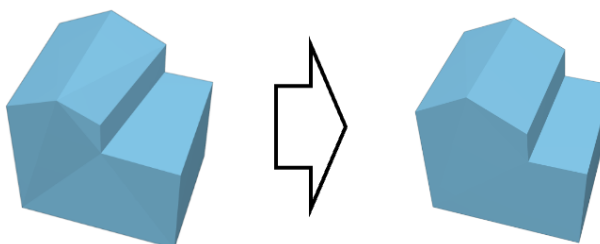


図 4-77 ノイズ削減されたデータの例

#### 4-4-4. 外部連携インターフェース

なし

### 4-5. 検証に用いたデータ

#### 4-5-1. 検証したデータの一覧

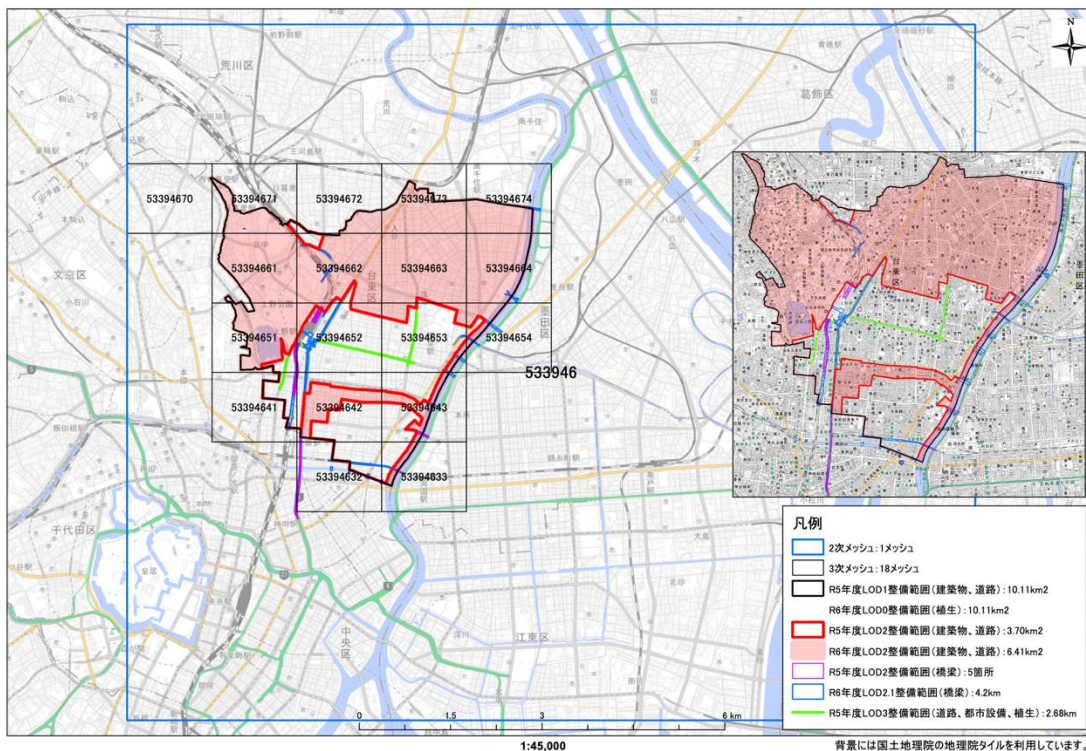


図 4-78 インデックスマップ (東京都台東区)

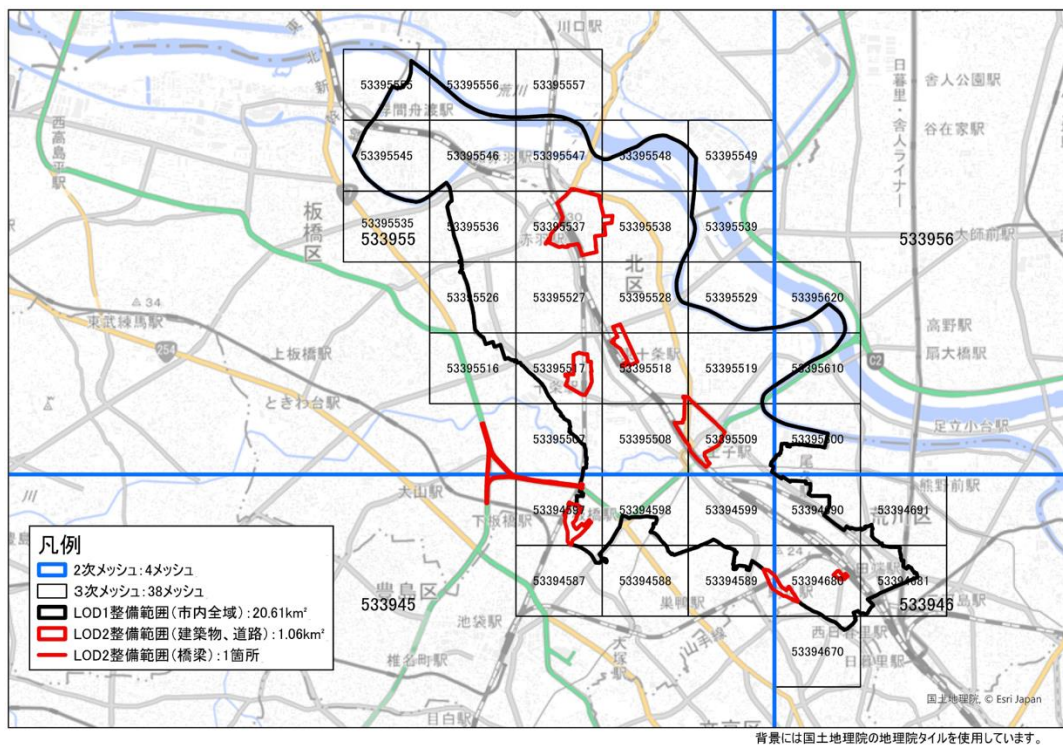


図 4-4 インデックスマップ (東京都北区)

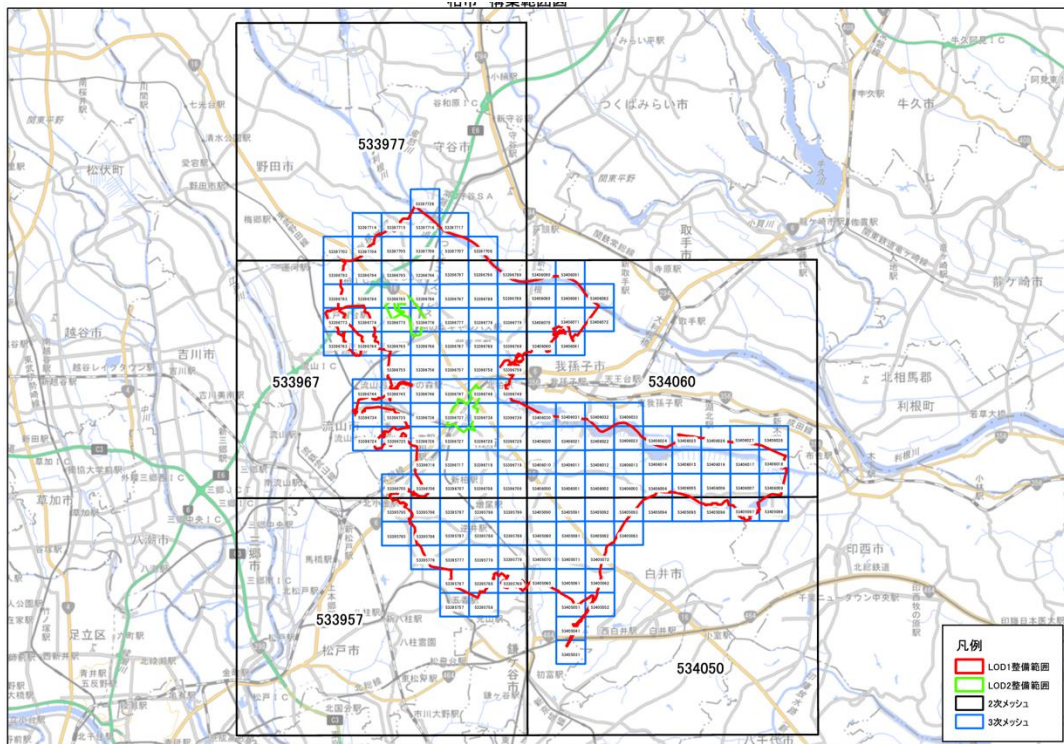


図 4-80 インデックスマップ (千葉県柏市)

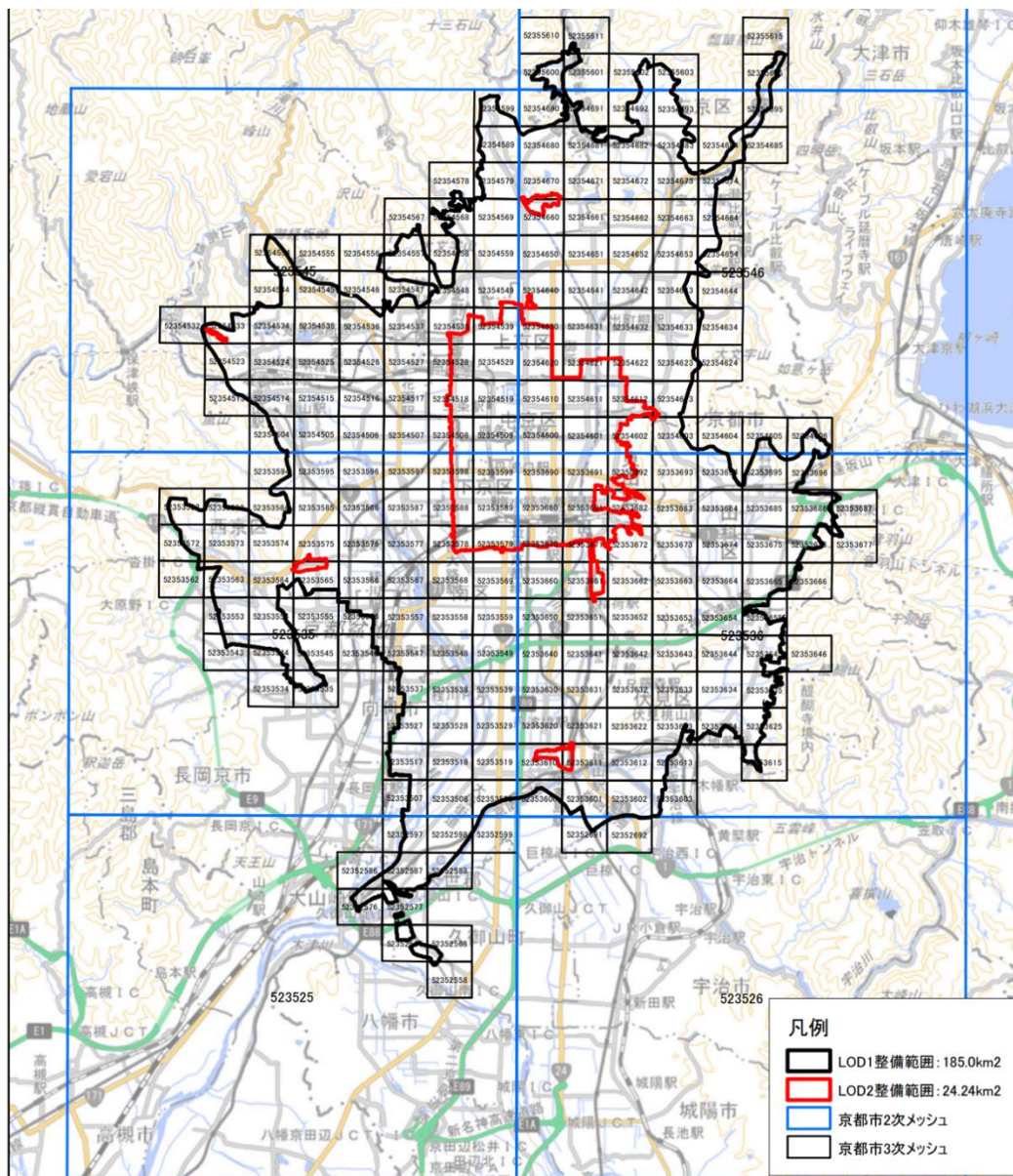


図 4-81 インデックスマップ (京都府京都市)

- 1) 利用した 3D 都市モデル
- 年度：2023 年度 (西暦)
  - 都市名：全国
  - ファイル名：(オープンデータ化後に記載)

表 4-12 利用する 3D 都市モデル

ID	地物	地物型	属性区分	属性名	内容	データを利用した機能 ID
DT001	建築物 (LOD0,	Building	空間属性	bldg:lod0RoofEdge	形状	FN005
DT002				bldg:lod1Solid	形状	FN005

DT003	LOD1,			bldg:lod2Solid	形状	FN001, FN005
DT004	LOD2, LOD3)			bldg:lod3Solid	形状	FN005
DT005	道 路 (LOD0)	tran	空間属性	tran:lod0MultiSurface	形状	FN009
DT006	植 生 (LOD3)	veg	空間属性	veg:lod3MultiPoint	形状	FN008
DT007	都 市 設 備 (LOD3 )	frn	空間属性	frn:lod3Solid	形状	FN010

## 2) 利用したデータ

## 1. データ一覧

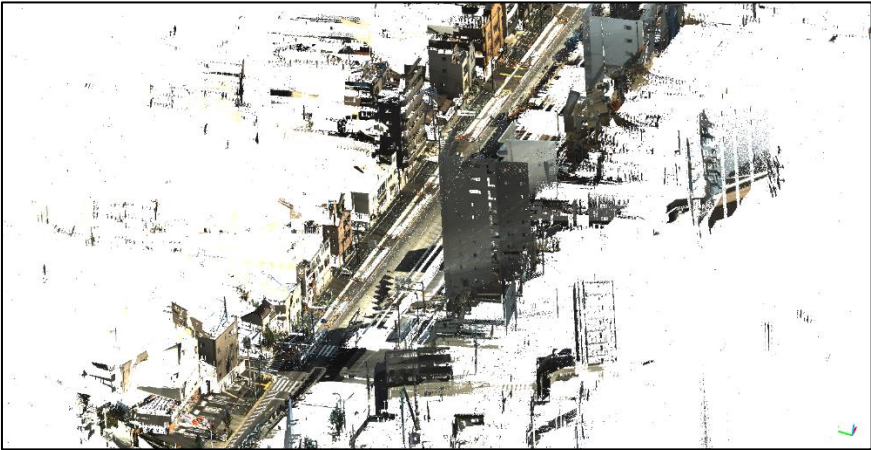
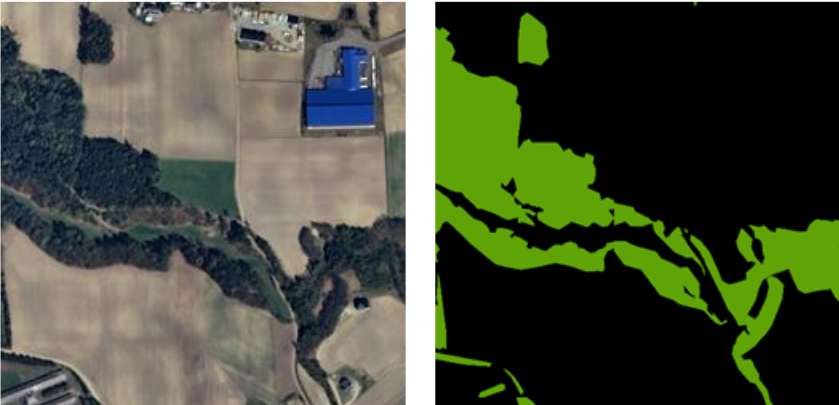
表 4-13 利用するその他データ (一覧)

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	東京都北 区、江東 区、千葉県 柏市、京都 府京都市	衛星画像	生成したい地域 の衛星画像 (RGB)	tif、tiff 形 式	MAPBOX	FN001 FN002
DT102	東京都北 区、江東 区、千葉県 柏市、京都 府京都市	フットプリ ント	建築物フットプ リント	GeoJson 形式	PLATEAU	FN006 FN007 FN008
DT103	東京都北区	沿道画像	MMS で撮影し た沿道画像	JPG 形式	NTT	FN003
DT104	東京都北区	三次元点群	MMS で計測し た三次元点群	LAS 形式	NTT	FN003
DT105	千葉県柏市	植生セグメ ンテーショ ンデータセ ット	植生自動抽出モ デル学習のため の衛星画像と植 生ラベリング	Tiff、png 形式	URUR dataset	FN002

2. データサンプル (イメージ)

表 4-14 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	衛星画像	
DT102	フットプリント	
DT103	街路画像	

DT104	三次元点群	
DT105	植生セグメンテーションデータセット	

4-5-2. 生成・変換するデータ

表 4-15 生成・変換するデータ

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	Obj	可視化のため	各地物の 3D 都市モデルを統合する処理	Python	生成した 3D 都市モデル (CityGML)	FN016
DT202	CityGML	PLATEAU 製品仕様書第 4.0 版の CityGML に出力するため	PLATEAU 製品仕様書第 4.0 版の CityGML に準拠したデータに変換する処理	Python	生成した 3D 都市モデル (CityGML)	FN006 FN007 FN008 FN009 FN010 FN011

## 4-6. ユーザーインターフェース

## 4-6-1. 画面一覧

## 1) 【HW001】画面

表 4-16 【HW001】PC用画面一覧

ID	連携 (ID)	画面名	画面説明	画面を表示した機能 (ID)
SC001	-	建築物モデル生成機能トップページ	デジタルシティサービス ( <a href="https://www.digitalsmartcity.jp/">https://www.digitalsmartcity.jp/</a> ) にアクセスし、建築物モデル生成機能を起動する	-
SC002	-	建築物 LOD 選択画面	生成する予定の建築物 LOD レベルを選定する	FN004
SC003	-	データ入力設定	選定された LOD レベルに応じて、必要なデータをそれぞれ入力する。 <ul style="list-style-type: none"> <li>● LOD0、1 の場合はフットプリントのみ</li> <li>● LOD2 の場合はそれ以上に衛星画像を追加</li> <li>● LOD3 の場合は両方以上に MMS 点群データを追加</li> </ul> 実行を押して 3D 都市モデル生成を実行する	FN001 FN003
SC004	-	生成過程画面	<ul style="list-style-type: none"> <li>● 生成過程が表示される</li> </ul>	-
SC005	-	データ出力/可視化画面	<ul style="list-style-type: none"> <li>● データが出力され可視化される</li> </ul>	FN014

4-6-2. 画面遷移図

1) 【HW001】 PC 用画面

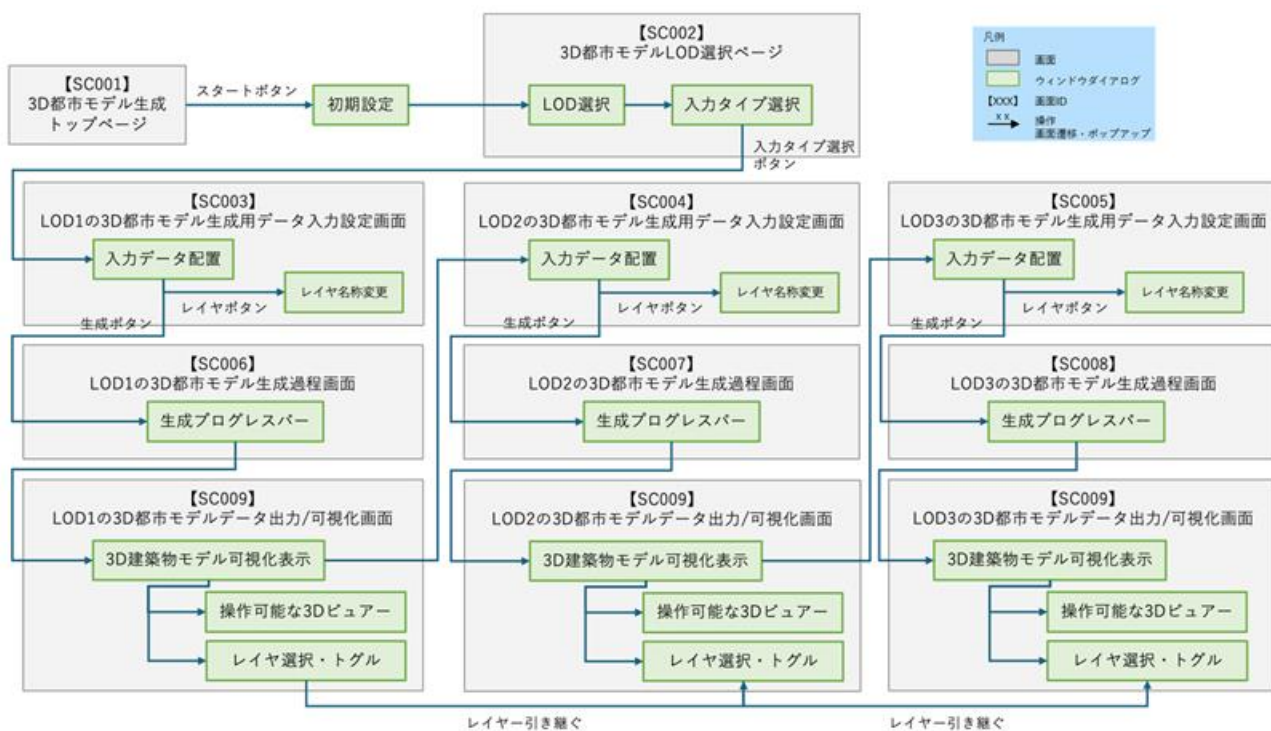


図 4-82 3D 都市モデル生成 PC 画面遷移図

### 4-6-3. 各画面仕様詳細

1) 【HW001】 PC 用画面

1. 【SC001】 3D 都市モデル生成機能トップページ



図 4-83 3D 都市モデル生成機能選択画面のイメージ

- 本画面の目的
  - 3D 都市モデル生成を起動する最初の画面
- 本画面から利用できる機能
  - 【FN004】パラメータ設定機能

2. 【SC002】 LOD 選択画面



図 4-84 LOD 設定画面のイメージ

- 本画面の目的
  - 生成する LOD を選定する画面
- 本画面から利用できる機能
  - 【FN004】パラメータ設定機能

3. 【SC003】LOD1 生成用データ入力設定画面



図 4-85 LOD1 生成用データ入力設定画面のイメージ

- 本画面の目的
  - LOD1 生成に必要なデータ（フットプリント）を入力する
  - 「生成」を押して次の画面に進める
- 本画面から利用できる機能
  - 【FN006】LOD1 以下の建築物モデル生成

4. 【SC004】LOD2 生成用データ入力設定画面

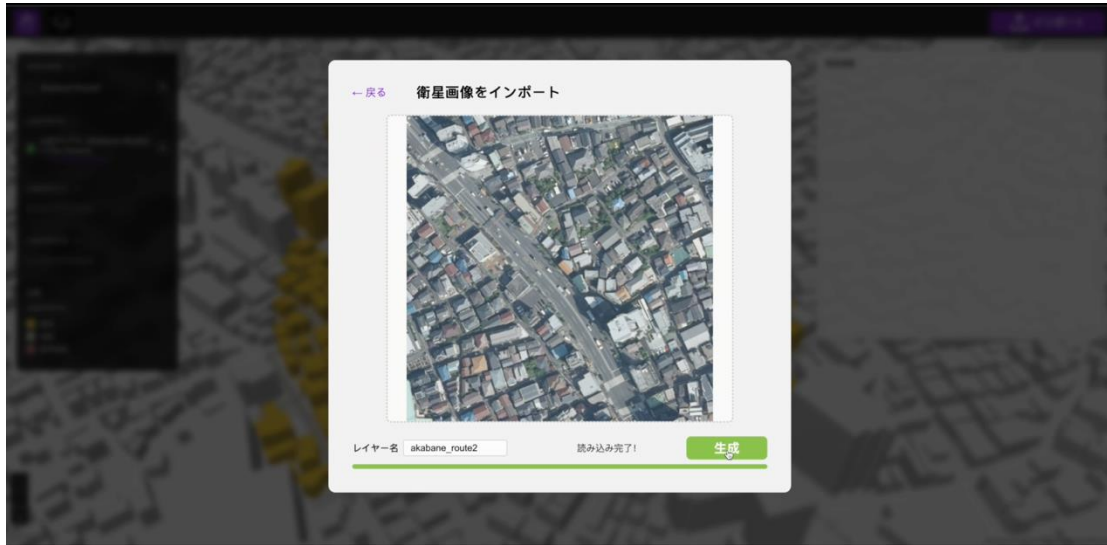


図 4-86 LOD2 生成用データ入力設定画面のイメージ

- 本画面の目的
  - LOD2 生成に必要なデータ（衛星画像）を入力する
  - 「生成」を押して次の画面に進める
- 本画面から利用できる機能
  - 【FN007】 LOD2 の 3D 都市モデル生成

5. 【SC005】 LOD3 生成のデータ入力設定画面

- 画面の目的・概要

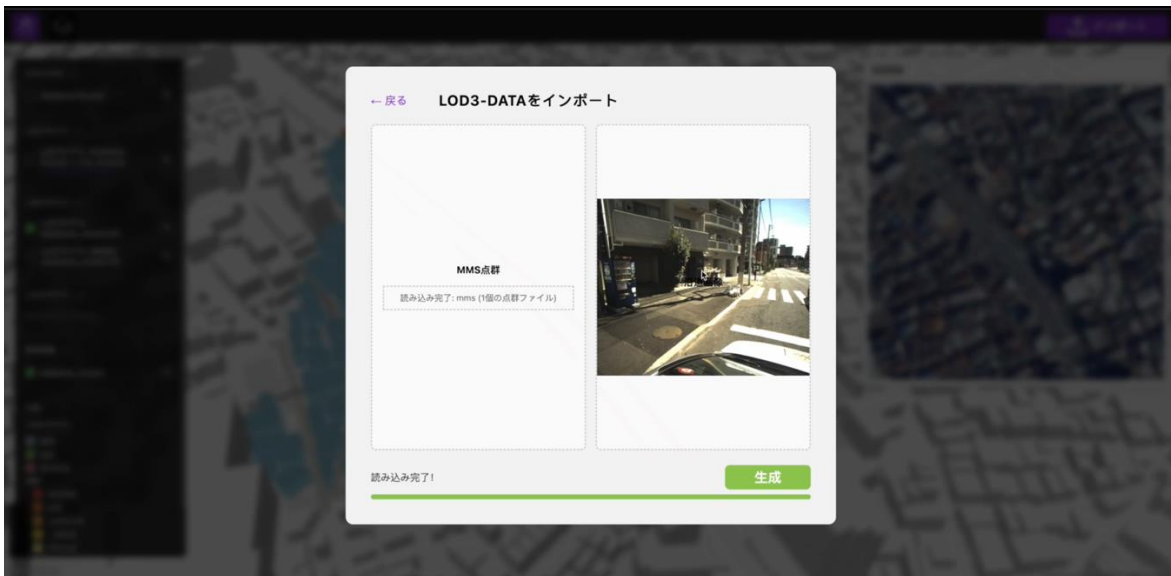


図 4-87 LOD3 生成のデータ入力設定画面のイメージ

- 本画面の目的
  - LOD3 の 3D 都市生成に必要なデータ（MMS 点群及び沿道画像）を追加入力する
  - 「生成」を押して LOD3 モデル生成を実行する

- 本画面から利用できる機能
  - 【FN008】 LOD3 の 3D 都市モデル生成
- 6. 【SC006】 LOD1 の 3D 都市モデル生成過程画面
- 画面の目的・概要



図 4-88 生成過程画面 (LOD1 モデル) のイメージ

- 本画面の目的
  - 生成過程が表示される画面
- 本画面から利用できる機能
  - なし
- 7. 【SC007】 LOD2 の 3D 都市モデル生成過程画面
- 画面の目的・概要

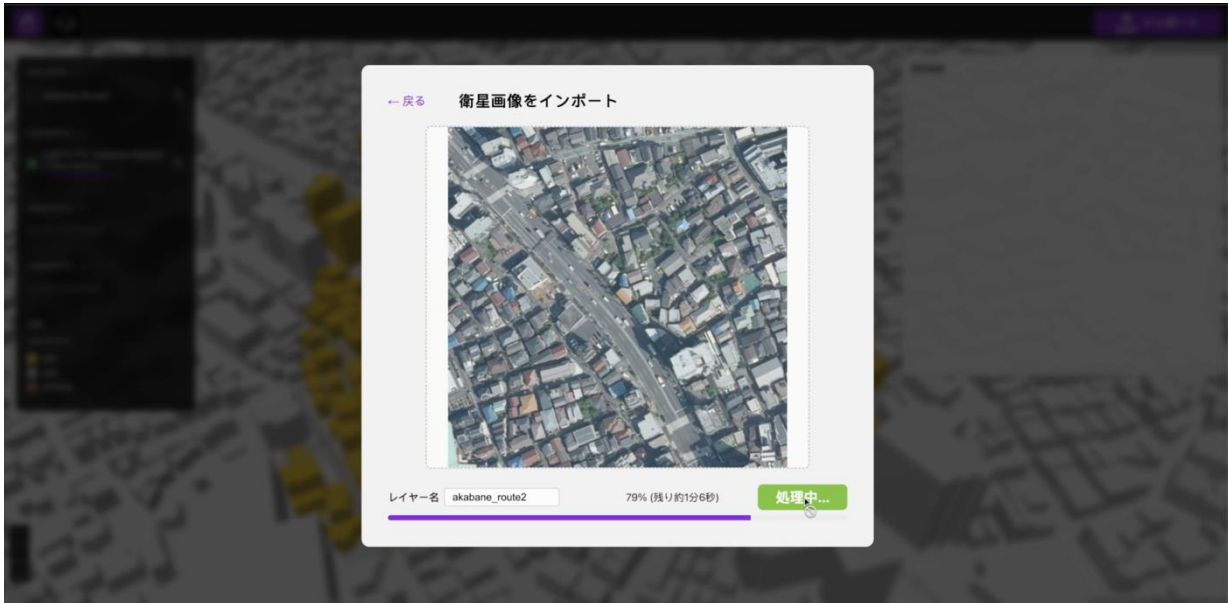


図 4-89 生成過程画面 (LOD2 モデル) のイメージ

- 本画面の目的
  - 生成過程が表示される画面
- 本画面から利用できる機能
  - なし

#### 8. 【SC008】 LOD3 の 3D 都市モデル生成過程画面

- 画面の目的・概要

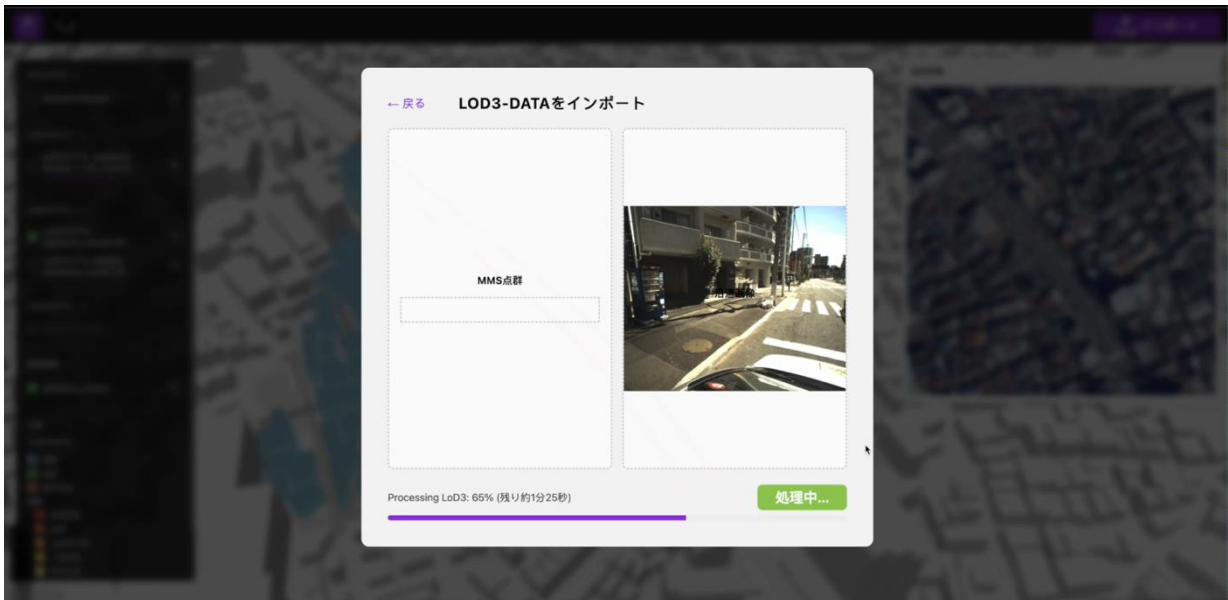


図 4-90 生成過程画面 (LOD3 モデル) のイメージ

- 本画面の目的

- 生成過程が表示される画面
- 本画面から利用できる機能
  - なし

9. 【SC009】 LOD1 都市モデルデータ出力／可視化画面

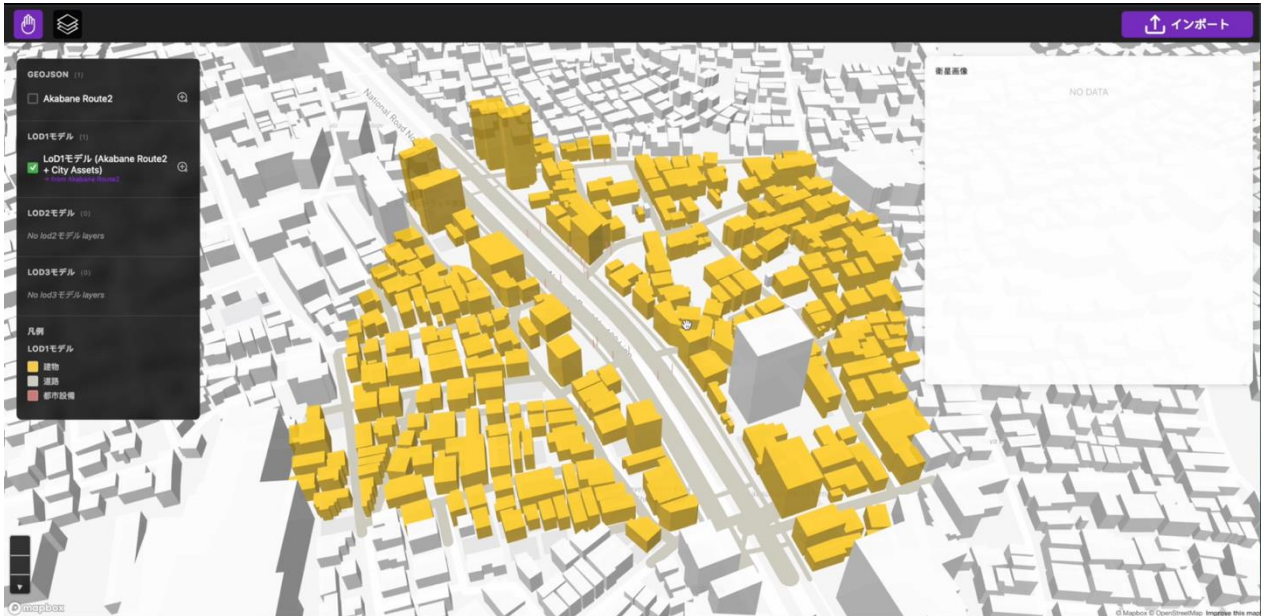


図 4-91 3D 都市モデル（LOD1）データ出力／可視化画面のイメージ

- 本画面の目的
  - 生成された 3D 都市モデル（LOD1）が出力され可視化される画面
  - 同一画面上で設定情報を確認可能
- 本画面から利用できる機能
  - 【FN016】可視化機能

10. 【SC010】 LOD2 の 3D 都市モデルデータ出力／可視化画面

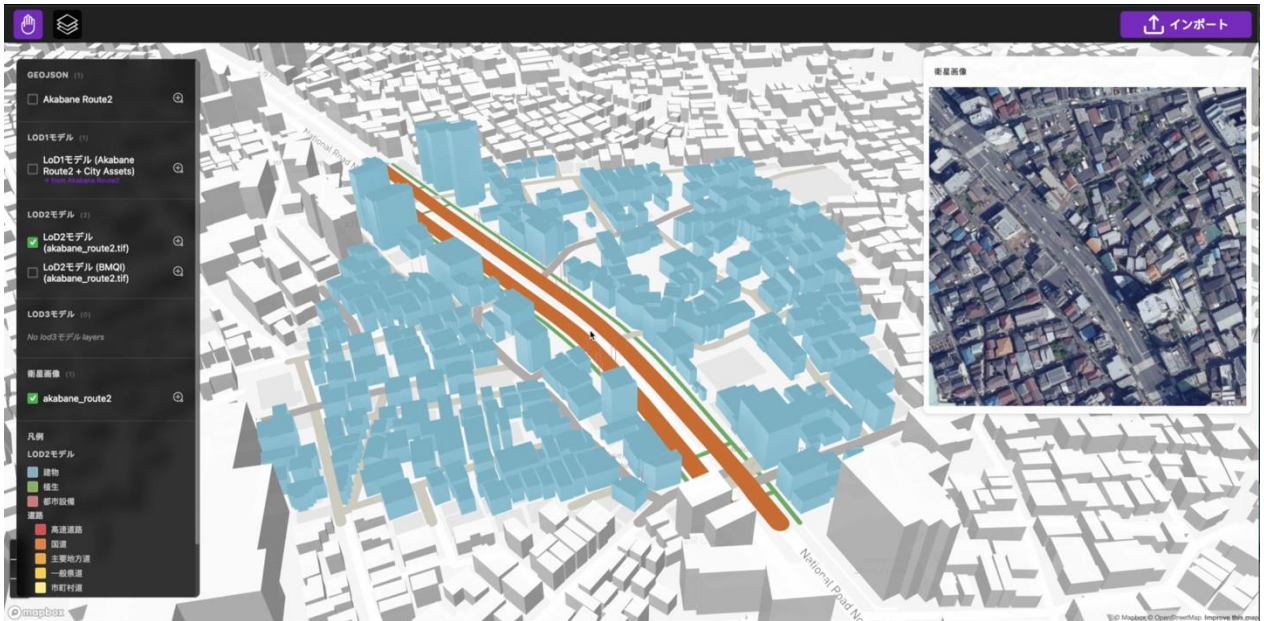


図 4-92 3D 都市モデル (LOD2) データ出力/可視化画面のイメージ

- 本画面の目的
  - 生成された 3D 都市モデル (LOD2) が出力され可視化される画面
  - 同一画面上で設定情報を確認可能
- 本画面から利用できる機能
  - 【FN016】可視化機能

11. 【SC011】 LOD3 の 3D 都市モデルデータ出力/可視化画面

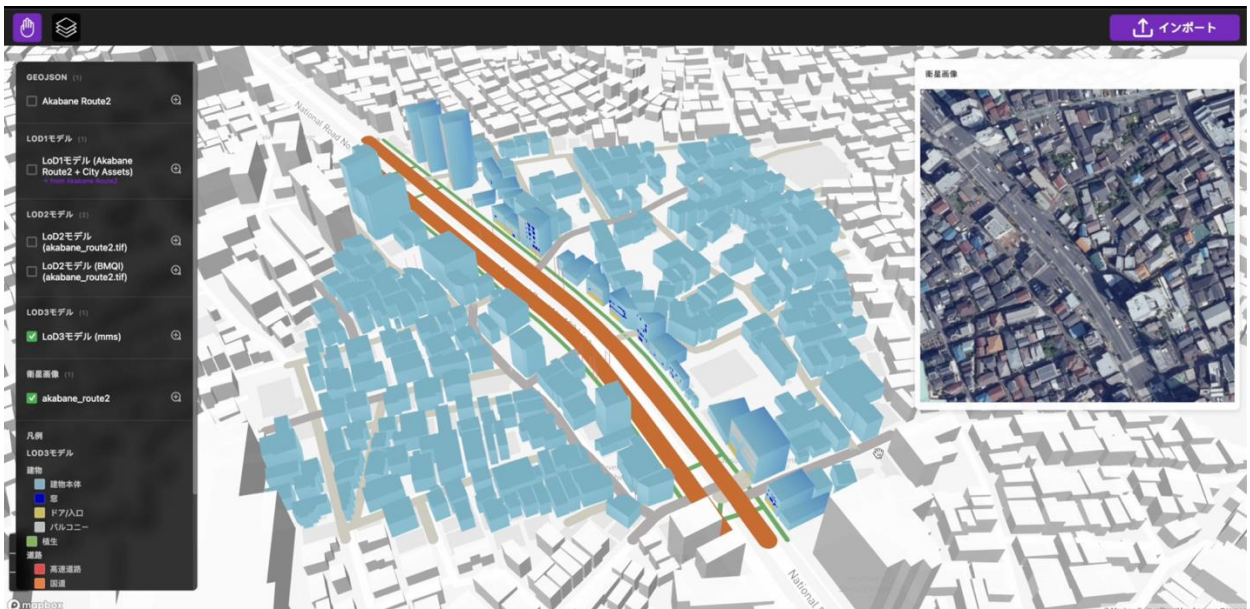


図 4-93 3D 都市モデル (LOD3) データ出力/可視化画面のイメージ

- 本画面の目的
  - 生成された 3D 都市モデル (LOD3) が出力され可視化される画面

- 同一画面上で設定情報を確認可能
- 本画面から利用できる機能
  - 【FN016】 可視化機能

## 4-7. 実証システムの利用手順

### 4-7-1. 実証システムの利用フロー

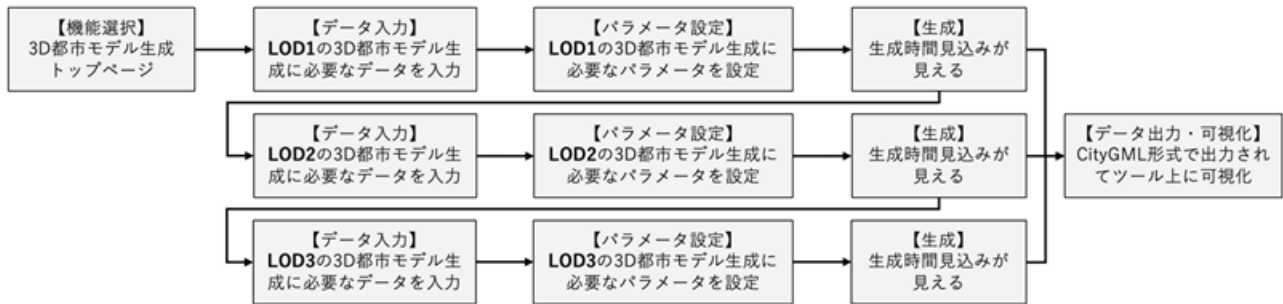


図 4-94 システムの利用フロー

ユーザーは衛星画像を入力して LOD 等のパラメータを設定することで、システムが入力データに基づき 3D 都市モデルを生成する。この機能により、ユーザーは具体的な地理的範囲を衛星画像に基づき指定でき、システムが自動的に衛星画像を AI が分析することで 3D 都市モデルを生成／可視化する。

## 4-7-2. 各画面操作方法

3D 都市モデル生成システム画面の詳細仕様を図に示す。

### 1) 機能選択

- デジタルシティサービスにアクセスし、自動的に建築物モデル生成機能を起動する



図 4-95 3D 都市モデル生成機能選択画面

### 2) 機能種別選択

- 生成する LOD を選択する



図 4-96 LOD を設定する画面

3) データ入力設定

- 設定した LOD に合わせた入力データをアップロードする



図 4-97 データ入力画面

- LOD3 の生成に必要な入力データを追加アップロードする
- 「生成」を押して 3D 都市モデル生成を実行する



図 4-98 LOD3 データの追加データ入力画面

4) モデル生成

- モデル生成プロセスを開始する。入力データとパラメータに基づいて自動的に 3D 都市モデルが生成される

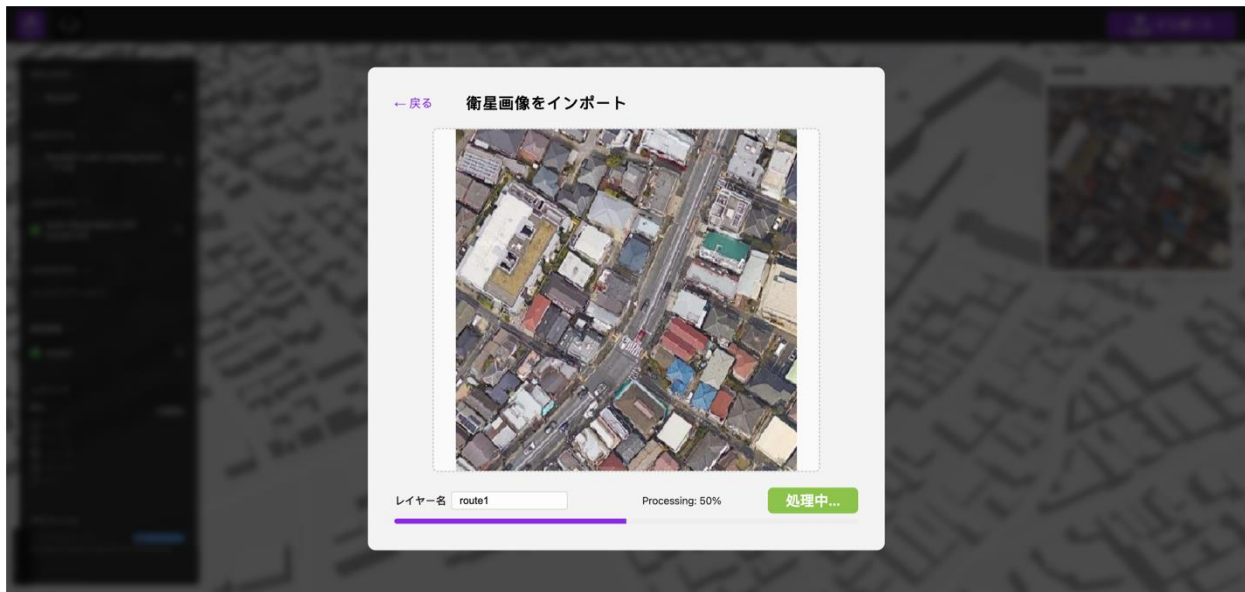


図 4-99 生成中の画面

5) データ出力／可視化

- 設定内容を確認できる
- 可視化: ツール内で直接生成された 3D 都市モデルを閲覧（ズーム、回転）できる
- データ出力: モデルが CityGML (PLATEAU v4) 形式で出力される

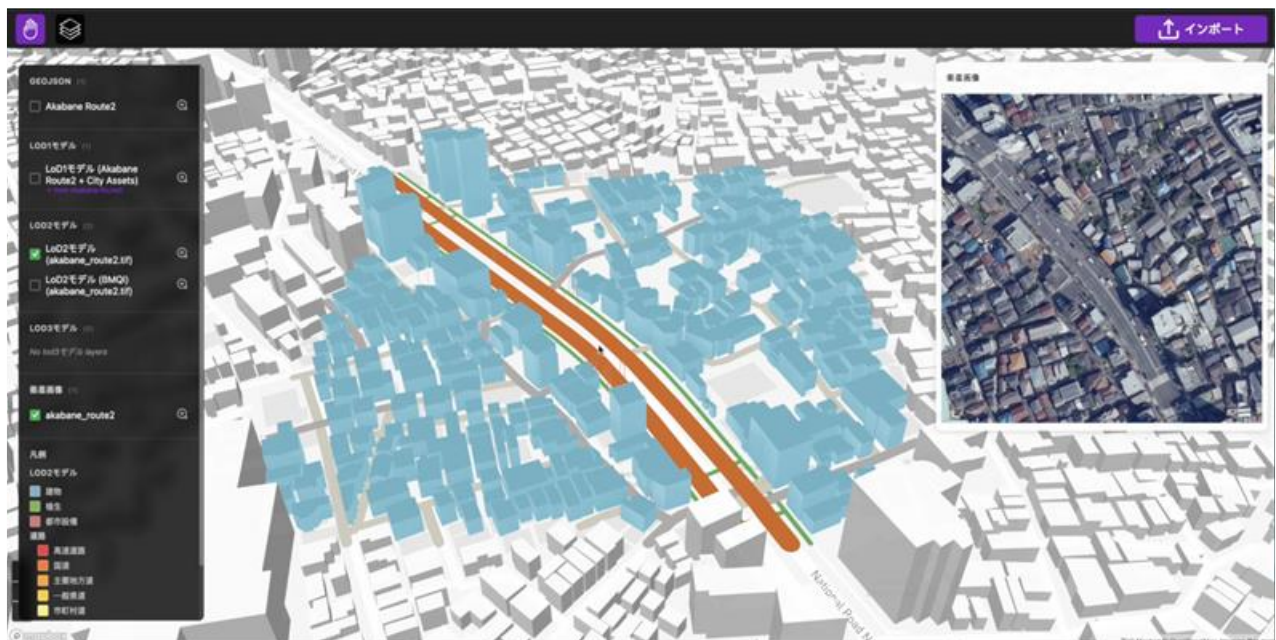


図 4-100 生成結果の可視化と出力画面

## 5. システムの非機能要件

### 5-1. 社会実装に向けた非機能要件

表 5-1 非機能要件一覧

カテゴリ	ID	項目	詳細
可用性	NR001	システムの連続稼働時間	● 生成シミュレーションを実施するため 5 時間程度稼働すること
	NR002	安定動作時間	● 1 度の生成に数十分かかるため、複数の生成を想定し、1 時間以上の安定動作時間を確保すること
性能・拡張性	NR003	データの読み込み速度	● システム利用者が不快なくシステム運用を行うことを想定し、インプットデータの読み込みを 30 秒以内で行うこと

#### 1) 【NR001】システムの連続稼働時間

- 本非機能要件を適用するシステム
  - 3D都市モデル生成
- 目標値
  - 5 時間
- 設定理由
  - 今回開発するシステムを用いて業務フロー自体の効率化が見込まれるが、シミュレーション範囲の拡大や、他エリアを対象とした連続稼働が見込まれるため、2 時間の連続稼働時間を確保する。
- 評価方法
  - 5 時間のシステム連続稼働を 5 回行い、システムダウンや、HW001 の PC のフリーズが発生しないことを確認する

#### 2) 【NR002】安定動作時間

- 本非機能要件を適用するシステム
  - 3D 都市モデル生成
- 目標値
  - 1 時間
- 設定理由
  - 一度の生成には数十分程度要している。今回のシステムを用いてシミュレーション範囲の拡大や、他エリアを対象とした連続稼働が見込まれるため、1 時間の連続稼働時間を確保する
- 評価方法
  - 1 時間にわたる生成を 10 回行い、システムダウンが発生しないことを確認する

#### 3) 【NR003】データ読み込み速度

## DT25-03\_技術検証レポート\_3D 都市モデル生成シミュレータの開発

- 本非機能要件を適用するシステム
  - 3D 都市モデル生成
- 目標値
  - 30 秒
- 設定理由
  - 衛星画像と沿道画像を入力として生成が行われるがシステム利用者が不快なくシステム運用を行うことを想定し、インプットデータの読み込みを 30 秒以内で行えることを確保する
- 評価方法
  - 衛星画像と沿道画像を入力とした生成を 10 回行い、30 秒以内に読み込みが行われていることを確認する

## 6. 品質

## 6-1. 機能要件の品質担保

表 6-1 機能要件の品質担保方針

対象プロセス/ サブシステム	品質評価項目	目標値	期間の単位	アクティビティ
3D 都市モデル 生成	モデル精度： 生成結果のジオメトリ誤差	鉛直方向 RMSE： ±1.75m 水平方向 RMSE： ±0.66m を満たす建築物の割合：100%	2025 年 12 月	ジオメトリは PLATEAU を正解データとして、生成結果の形状を水平方向及び鉛直方向の誤差（RMSE）で評価する
	モデル精度： 生成結果のノイズ検証	ノイズ評価指標である BMQI が検証エリアで平均 90 点以上（目視で分からないレベル）	2025 年 12 月	ノイズは BMQI を用いて、三角メッシュのズレや凹凸を評価する。対象は、生成された 3D モデルそのものである
	生成時間	500m 四方を 15 分で生成	2025 年 12 月	単体テストによる検証
	データ品質	ファイルの中身全体の検証作業を通じて、3D 都市モデル標準製品仕様書第 4.0 版の CityGML 形式の要件を参照し、出力フォーマットが規格に準拠しているかを厳密に検証する	2025 年 12 月	3D 都市モデル標準製品仕様書第 4.0 版を参照し、検証を行う

## 6-2. 非機能要件の品質担保

表 6-2 非機能要件の品質担保方針

対象システム	試験項目	確認内容	試験期間	アクティビティ
3D 都市モデル生成システム	システム稼働時間	5 時間以上の稼働に耐え得るか	2026 年 1 月～2 月	運用テストによる検証
	リファレンスデータの読み込み速度	30 秒以内にデータの読み込みが完了するか	2026 年 1 月～2 月	運用テストによる検証
	安定動作時間	1 時間以上安定動作するか	2026 年 1 月～2 月	運用テストによる検証
	画面描画フレームレート	シミュレーション中の画面描画フレームレートが 30fps 以上か	2026 年 1 月～2 月	運用テストによる検証

## 7. 実証技術の機能要件の検証

### 7-1. 3D 都市モデル生成シミュレータの検証

#### 7-1-1. 検証目的

- 開発した 3D 都市生成モデルの精度検証を通じて、提案モデルの優位性を検証する
- ファイルが CityGML 形式で正しく出力される割合を確認し、標準に準拠したデータの生成能力を検証する
- 生成に要する時間を測定し、モデル生成プロセスの効率性を評価する

#### 7-1-2. KPI

表 7-1 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	植生抽出の正解率と IOU	正解率:80% IOU: 60%	● 既往研究を参考に設定	● 検証データに対して、クラスごとの正解率 (Accuracy) と IOU(Intersection Over Union)を算出
2	建築物モデル精度：生成結果のジオメトリ誤差	1/2500 誤差以内 (鉛直方向 RMSE：±1.75m、 水平方向 RMSE：±0.66m)	● 建築物について、PLATEAU 同等水準の 1/2500 を満たすことが理想であるため	● ジオメトリは PLATEAU の 3D 都市モデルを正解データとして、生成結果の形状を水平方向及び鉛直方向の誤差 (RMSE) で評価
3	建築物モデル精度：生成結果のノイズ検証	ノイズ評価指標である BMQI が検証エリアで平均 90 点以上 (目視分からないレベル)	● 建築物について、ノイズが無い建築物を満たす BMQI が 90 点以上であるため、目標値を 90 点と設定	● ノイズは BMQI を用いて、三角メッシュのズレや凹凸を評価
4	開口部抽出精度	再現率と適合率 0.8 以上	● 既往研究を参考に設定	● 検証データに対して、開口部抽出の精度 (再現率・適合率) を算出

5	データ品質	3D 都市モデル標準製品仕様書第 4.0 版の CityGML フォーマットに準拠しているファイル割合が 100%	● 出力形式が正しくないと可視化等に影響が出るため、目標値を 100%と設定	● 出力ファイルの中身全体の検証作業を通じて、3D 都市モデル標準製品仕様書第 4.0 版の CityGML 形式の要件を参照し、出力フォーマットが規格に準拠しているかを厳密に検証する
6	LOD2 生成時間	500m 四方を 15 分で生成	● 参考にするため、目標値は設定しない	● 生成処理時間を計測
7	LOD3 (MMS) 生成時間	500m 四方を 15 分で生成	● 参考にするため、目標値は設定しない	● 生成処理時間を計測

### 7-1-3. 検証方法と検証シナリオ

#### 1) 植生抽出の正解率と IOU

開発された衛星画像からの植生の検出モデルの性能は、正解率、交差オーバーユニオン (IOU) の指標を用いて包括的に評価される。これらの指標は、モデルが植生と道路をどれだけ精密に識別できるかを定量的に示すために使用される。正解率はモデルの予測結果に対する正しい予測の割合、IOU は予測された領域と実際の領域の重なりを評価する。

- Accuracy (正解率) : モデルの予測結果において正しいピクセルの数を全ピクセル数で割った指標

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100(\%)$$

TP は真陽性、TN は真陰性、FP は偽陽性、FN は偽陰性

- IoU (交差オーバーユニオン) : 予測されたクラス領域と実際のクラス領域の重なりを評価する指標

$$\text{IoU}(\%) = \frac{\text{重なり領域}}{\text{結合領域}} \times 100(\%)$$

表 7-2 検証シナリオ一覧 (セグメンテーション精度 (植生))

No.	検証方法	エリア	対象データ
1-1	テストデータによる検証	千葉県柏市	植生の検証用データ*

#### 2) 建築物モデル精度 : 生成結果のジオメトリ誤差

検証では、生成された 3D 都市モデルのジオメトリ精度 (RMSE) とノイズ (BMQI) の観点から精度検証する。これにより、水平方向・鉛直方向及び外観の品質と観点から統合的に評価が可能となる。

- RMSE: 生成された各建築物の水平方向と鉛直方向の二乗平均平方根誤差 :

$$RMSE = \sqrt{\frac{\sum_1^N (y_i - \hat{y}_i)^2}{N}}$$

各建築物屋根の計測点数を N とし、真値を  $y_i$ 、推定値を  $\hat{y}_i$  とする

表 7-3 検証シナリオ一覧（屋根タイプ別建築物生成モデルの精度）

No.	検証方法	建築物屋根タイプ	対象データ
2-1	建築物の RMSE	柏市	建築物の生成データ
2-2		江東区	建築物の生成データ
2-3		京都市	建築物の生成データ

3) 築物モデル精度：生成結果のノイズ検証

- BMQI: 鉛直側面率、屋根接続側面率、フットプリント収まり率の 3 項目から構成される外見のノイズを定量的に評価するための統合評価指標：

$$BMQI = W_{roof}(r_{adj} * \sigma(R_{area})) + W_{facade}(r_{vertical} * \sigma(F_{area})) + W_{floor}r_{enclose}$$

ただし、

$r_{adj}$ (roof adjacency): 全方向隣接ある屋根三角形の割合

$r_{vertical}$ (façade verticality): 鉛直向きの側面三角形の割合

$r_{enclose}$ (floor enclosing): フットプリント範囲に収まる建築物全体三角形の割合

- $W_{roof}$ ,  $W_{facade}$ ,  $W_{floor}$ : 各部の重み
- $R_{area}$ : 屋根の平面投影面積と底面面積の比率
- $F_{area}$ : 側面が囲む平面面積と底面面積の比率
- $\sigma$ : 区分写像によるペナルティ項

$$\sigma(x) = \begin{cases} e^{-\lambda(l-x)}, & x < l \\ 1, & l \leq x \leq h \\ e^{-\lambda(x-h)}, & x > h \end{cases}$$

表 7-4 検証シナリオ一覧（建築物生成モデルの精度）

No.	検証方法	建築物屋根タイプ	対象データ
2-1	建築物の BMQI	柏市	建築物の生成データ
2-2		江東区	建築物の生成データ
2-3		北区赤羽	建築物の生成データ
2-4		京都市	建築物の生成データ

4) 出力ファイルが CityGML で正しく出力される割合 (%)

生成された 3D 都市モデルの地物出力ファイルが、3D 都市モデル標準製品仕様書第 4.0 版の CityGML 形式に準拠しているか否かを確認し、準拠しているファイルの正確な割合を算出する作業は、規格に基づい

た検証プロセスを要求する。このプロセスでは、初めに 3D 都市モデル標準製品仕様書第 4.0 版を参照し、出力されるフォーマットが規格に適合しているかを検証する。その上で、CityGML 形式で適切に出力されたファイルの割合を計算する。

- 正確な出力割合：CityGML（3D 都市モデル標準製品仕様書第 4.0 版）仕様に基づいて正しくファイルが出力されるファイルの割合

$$\text{正確な出力割合 (\%)} = \frac{\text{仕様に準拠したファイル数}}{\text{全ファイル数}} \times 100(\%)$$

表 7-5 検証シナリオ一覧（ファイルが CityGML で正しく出力される割合）

No.	検証方法	エリア	対象データ
4-1	正確な出力割合	柏市	生成された 3D 都市モデルデータ
4-2		江東区	生成された 3D 都市モデルデータ
4-3		北区赤羽	生成された 3D 都市モデルデータ
4-4		京都市	生成された 3D 都市モデルデータ

#### 5) 開口部抽出精度

建築物開口部抽出の精度では、4 つの実験において、適合率と再現率で評価する。

- 適合率（Precision）：抽出した建築物の開口部に対してどれぐらい正しく抽出できたかを測る指標であり、数値が低いと開口部の誤抽出が多いことになる。

$$\text{適合率} = \frac{\text{正しく抽出できた建築物の開口部数}}{\text{抽出された建築物の開口部総数}}$$

- 再現率（Recall）：実際の建築物の開口部に対してどれだけ取りこぼしなく抽出できたかを測る指標であり、数値が低いと開口部の見逃しが多いことになる。

$$\text{再現率} = \frac{\text{抽出できた建築物の開口部数}}{\text{実際の建築物の開口部の総数}}$$

表 7-6 検証シナリオ一覧（建築物テクスチャの貼り付けの精度）

No.	検証方法	エリア	対象データ
5-1	下記 4 つの実験での開口部抽出精度(適合率、再現率)の確認 1) 2D 処理の AI の評価実験 正しく目視で確認できる開口部のみで実験（ピンぼけ、オクルージョンで見えない、などは評価対象外） 2) 3D 処理の AI の評価実験 3D 処理で作成した画像を目視で確認して穴が開いている開口部のみで実験（カーテンやシャッターなどで埋まっているや、オクルージョンで穴が開きすぎているなどは評価対象外）	東京都北区	開口部情報

	3) MMS での開口部すべてにおける 3D 処理の精度評価実験 MMS の点群が存在する開口部全てで実験 4) 2D 処理と 3D 処理の結合の評価実験 2D/3D ハイブリッド手法の有用性の確認のために、3)での取りこぼしを 2D 処理の結果でどの程度補えているかの確認		
--	--	--	--

6) LOD2 生成時間

3D 都市モデルの成処理開始から終了までの処理時間を計測する。

表 7-7 検証シナリオ一覧 (生成時間)

No.	検証方法	エリア	対象データ
6-1	生成時間	千葉県柏市	生成された 3D 都市モデルデータ
6-2		東京都江東区	生成された 3D 都市モデルデータ
6-3		東京都北区	生成された 3D 都市モデルデータ
6-4		京都府京都市	生成された 3D 都市モデルデータ

7) LOD3 (MMS) 生成時間

MMS データから、開口部情報抽出を実行するのに要した時間を計測する。走行ルートごとに走行距離が異なるので、100m 当たり時間に換算したものを評価値として用いる。

$$100\text{m あたり生成時間} = \frac{\text{処理時間の総和}}{\text{ルート長の総和}} \times 100$$

表 7-8 検証シナリオ一覧 (開口部抽機能の処理時間)

No.	検証方法	エリア	対象データ
7-1	処理時間(100m 換算)	東京都北区	赤羽の走行データ

7-1-4. 検証結果

1) セグメンテーション正解率と IOU (植生)

● 検証結果まとめ

- 衛星画像からの植生の検出に関する 3D 都市モデルの検証結果では、植生のセグメンテーション正解率は 96.97%であった。IOU は 90.94%であり、モデルが衛星画像の植生特徴を正確に識別し、3D 都市モデルの整備に有用な情報を推定する能力があることが示された。
- 目標値である正解率 80%と IOU60%は達成された。

表 7-9 検証結果サマリー(セグメンテーション精度 (植生))

		赤セル：KPI 達成		青セル：KPI 未達			
No.	検証内容	評価指標・KPI	目標値	結果			示唆
				項目	Accuracy (%)	IoU (%)	
1-1	衛星画像セグメンテーションモデル正解率と IOU	植生のセグメンテーション精度	正解率:80% IOU: 60%	植生	96.97	90.94	● 開発したモデルは衛星画像から植生を高精度に推定できることを示している

## 2) 建築物モデル精度：生成結果のジオメトリ誤差

## ● 検証結果まとめ

- 衛星画像からの植生の検出に関する 3D 都市モデルの検証結果では、植生のセグメンテーション正解率は 96.97%であった。IOU は 90.94%であり、モデルが衛星画像の植生特徴を正確に識別し、3D 都市モデルの整備に有用な情報を推定する能力があることが示された。
- 目標値である正解率 80%と IOU60%は達成された。

表 7-10 検証結果サマリー (RMSE)

		赤セル：KPI 達成		青セル：KPI 未達			
No.	検証内容	目標値	RMSE		示唆		
			項目	RMSE			
2-1	建築物の RMSE	1/2500 誤差以内 (鉛直方向 RMSE : ±1.75m,、水平方向 RMSE : ±0.66m 以内)	千葉県柏市	0.19	● 全 3 地域において RMSE が 0.3m 以下を記録し、目標値である水平方向 ±0.66m を大幅に上回る高精度な結果となった。特に柏市では 0.19m と最も小さい誤差を達成しており、生成 AI による建築物モデル生成が測量基準に準拠した高精度なジオメトリ再		
2-2			東京都江東区	0.31			

2-3			京都府京都市	0.24	現を実現できることが確認された ● 地域特性や建築物形状の違いに関わらず安定した精度を維持しており、本システムの信頼性と実用性が実証された
-----	--	--	--------	------	--

No 2-1

柏市の生成結果例と RMSE



図 7-1 千葉県柏市の生成結果の例（左図）と RMSE（右図）

No 2-2

東京都台東区の生成結果例と RMSE

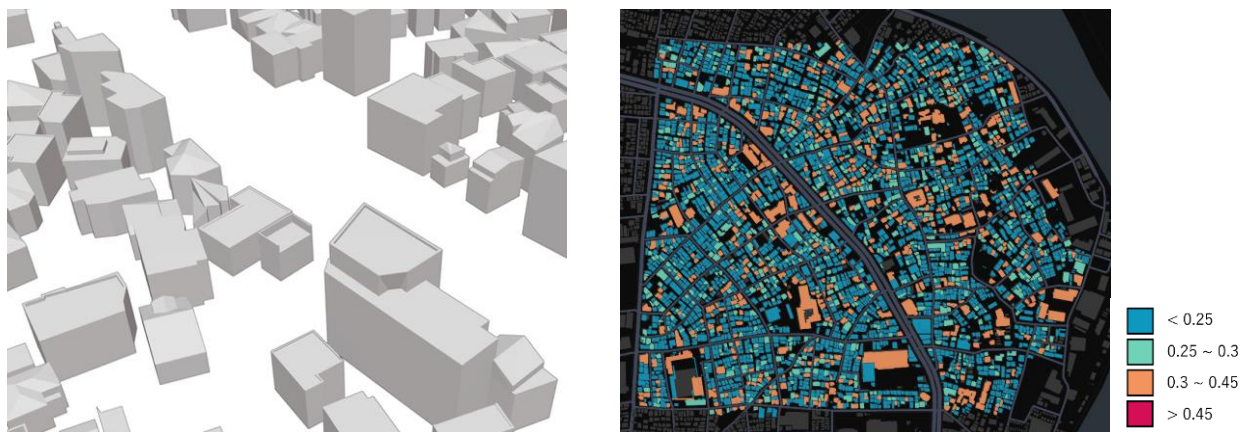


図 7-2 東京都北区の生成結果の例（左図）と RMSE（右図）

No 2-3

京都府京都市の生成結果例と RMSE

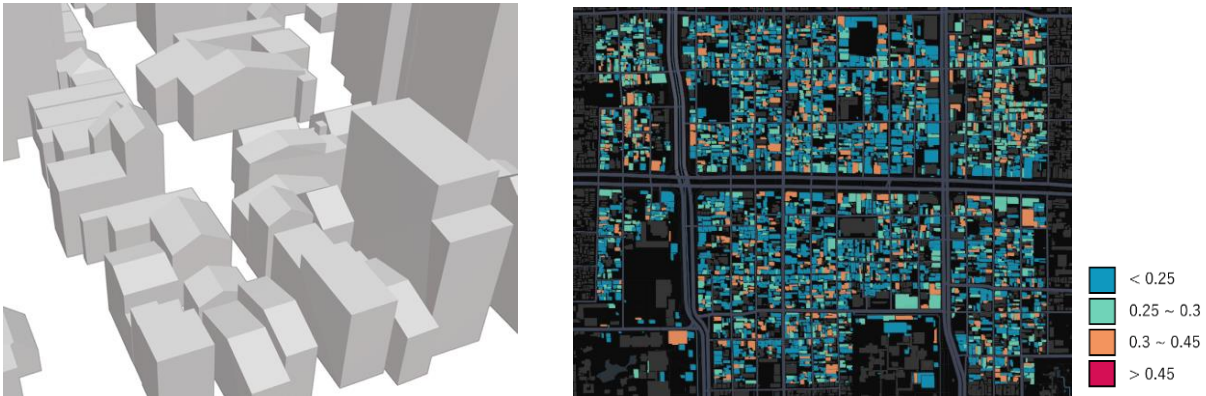


図 7-3 京都府京都市の生成結果の例（左図）と RMSE（右図）

3) 建築物モデル精度：生成結果のノイズ検証

● 検証結果まとめ

- 建築物モデル精度に関する 3D 都市モデルの検証結果では、4 地域（千葉県柏市、東京都台東区、東京都北区、京都府京都市）においてノイズ評価指標である BMQI（Building Mesh Quality Index）を計測した。BMQI は全地域において目標値である 90 点を達成した
- 本システムが生成する建築物モデルは目視でノイズがわからないレベルの高品質なメッシュ形状を有し、実用的な 3D 都市モデルとして十分な精度を持つことが示された

表 7-11 検証結果サマリー（BMQI）

		赤セル：KPI 達成		青セル：KPI 未達	
No.	検証内容	目標値	BMQI		示唆
3-1	建築物の BMQI	ノイズ評価指標である BMQI が検証エリアで 90 点以上（目視でわからないレベル）の割合	千葉県柏市	98.6	<ul style="list-style-type: none"> <li>● 全 4 地域において BMQI が 93 点以上を記録し、目標値である 90 点を安定して上回る結果となった</li> <li>● 地域間で若干の差異はあるものの、いずれも目視でノイズが判別できないレベルを維持しており、本システムの汎用性と安定性が実証された</li> </ul>
3-2			東京都江東区	96.1	
3-3			東京都北区	93.2	
3-4			京都府京都市	95.7	

No 3-1

千葉県柏市の BMQI



図 7-4 千葉県柏市の各建築物の BMQI の可視化

No 3-2

東京都台東区の BMQI

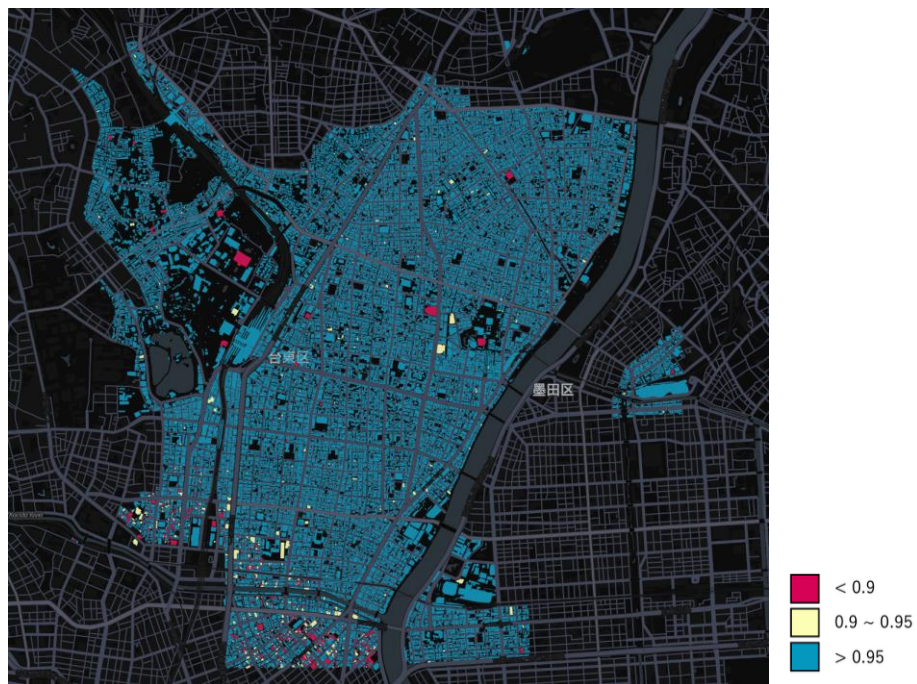


図 7-5 東京都台東区の各建築物の BMQI の可視化

No 3-3

東京都北区の BMQI



図 7-6 東京都北区の各建築物の BMQI の可視化

No 3-4

京都府京都市の BMQI



図 7-7 京都府京都市の各建築物の BMQI の可視化

4) ファイルが CityGML で正しく出力される割合 (%)

● 検証結果まとめ

- 柏市におけるデータ品質に関する 3D 都市モデルの検証結果では、建築物、道路、植生、都市設備の

全 4 カテゴリにおいて CityGML フォーマット準拠率を検証した。本システムが生成する 3D 都市モデルが 3D 都市モデル標準製品仕様書第 4.0 版に完全に準拠していることが確認された。これにより、PLATEAU データとの相互運用性が確保され、他システムやアプリケーションとの連携において技術的障壁のない実用的な 3D 都市モデルとして活用可能であることが示された

表 7-12 検証結果サマリー（柏市におけるデータ品質）

		赤セル：KPI 達成		青セル：KPI 未達	
No.	検証内容	目標値	CityGML として正しく出力された割合		示唆
4-1	ファイルの中身が 3D 都市モデル標準製品仕様書第 4.0 版の CityGML フォーマットに準拠しているファイル割合	CityGML フォーマットに準拠しているファイル割合が 100%	建築物	100	<ul style="list-style-type: none"> <li>建築物、道路、植生、都市設備の全地物カテゴリにおいて CityGML 準拠率 100%を達成した</li> <li>本システムが生成する全ての 3D 都市モデルが 3D 都市モデル標準製品仕様書第 4.0 版に準拠しており、PLATEAU データとの相互運用性が確保されていることが確認された</li> </ul>
4-2			道路	100	
4-3			植生	100	
4-4			都市設備	100	

5) 開口部抽出精度

● 検証結果まとめ

- 2D 処理の AI の評価実験では、目視で形状が確認できる開口部全てにおいて実験を行った。その結果、再現率はすべて 90%以上であり、正しく開口部を抽出できていることがわかった。ただし、窓以外の適合率が約 70%であり、誤抽出が若干多くなった。これは、窓に比べ様々な形状が存在するにもかかわらず、学習データ数が少ないことが原因だと考えられる。
- 3D 処理の AI の評価実験では、三次元点群において目視で形状が確認できる開口部全てにおいて実験を行った。再現率・適合率共に高い数値で開口部を取得できた。入口のみ適合率が 0.79 と誤抽出が若干多く目標値を達成できなかったが、学習データ数を増やすことで、0.8 以上は実現できると考えられる。また、画像より、三次元点群を用いた方が高精度に抽出できることがわかった。
- MMS での開口部すべてにおける 3D 処理の精度評価実験では、全体の再現率は 80.53%となり目標の数値は達成できた。そして、取得できなかった約 20%について詳細な検証を行った。その結果、ガラスの材質やカーテンなどが原因で穴が埋まってしまったこと（埋まり）と、障害物などのオクル

ーションが原因で点群が正しい形状で生成されていないこと（オクルージョン）の 2 つが大半の原因であることがわかった。そのため、作成した AI の精度は十分に運用に耐えうる精度があることがわかったが、現状の MMS の三次元点群だけでは限界があることもわかった。

- 2D 処理と 3D 処理の結合の評価実験では、2D 処理と 3D 処理の結果を結合することで、全体の再現率が 88.6%と高精度に開口部を抽出できることが示された。具体的には、3D 処理だけでは約 80%が限界であったが、2D/3D ハイブリッド処理を用いることで三次元点群では取得できなかった開口部が取得できるようになった。このことより、2D/3D ハイブリッド処理の有用性が示された。また、点群は存在するが画像に写っていない場所も多く、画像の撮影角度などを変更すれば精度がさらに向上すると考えられる。

表 7-13 検証結果サマリー（2D 処理の AI の評価実験）

				赤セル：KPI 達成							青セル：KPI 未達	
No.	検証内容	目標値	2D 処理結果								示唆	
			ラベル	TP	TFP	FP	FN	適合率	再現率	学習データ数		
5-1	2D 処理の開口部の適合率・再現率	適合率・再現率 0.8 以上	window (窓)	2,168	77	274	156	0.89	0.93	4481	すべてのラベルで再現率が高い。しかし、窓以外のラベルに関しては、適合率から少し誤抽出が多いことがわかる。これは、学習データ数が少ないためと考えられる。	
			door (扉)	147	54	55	10	0.73	0.94	54		
			shutter (シャッター)	58	4	23	0	0.72	1.0	36		
			garage (ガレージ)	108	22	68	4	0.61	0.96	82		
			balcony (バルコニー)	396	29	122	29	0.76	0.93	267		

※TFP は、ラベルが間違っているが、正しく開口部の場所は抽出できているため TP として扱っている。

表 7-14 検証結果サマリー (3D 処理の AI の評価実験)

				赤セル：KPI 達成							青セル：KPI 未達	
No.	検証内容	目標値	3D 処理結果								示唆	
			ラベル	TP	TFP	FP	FN	適合率	再現率	学習データ数		
5-2	3D 処理の開口部の適合率・再現率	適合率・再現率 0.8 以上	window (窓)	683	26	78	14	0.89	0.97	1,677	入口のみ誤抽出が若干多くなっているが、全体的にすべてのラベルに対して、適合率/再現率共に高く、正しく開口部を抽出できていることがわかる	
			entrance (入口)	140	36	36	4	0.79	0.97	284		
			fence (柵付き窓)	19	18	3	3	0.86	0.86	110		
			balcony (バルコニー)	447	25	42	15	0.91	0.96	574		

※TFP は、ラベルが間違っているが、正しく開口部の場所は抽出できているため TP として扱っている。

表 7-15 検証結果サマリー (MMS での開口部すべてにおける 3D 処理の精度評価実験)

				赤セル：KPI 達成			青セル：KPI 未達			
No.	検証内容	目標値	3D で抽出可能な開口部				抽出漏れの内訳			示唆
			ラベル	実開口部数	TP+TFP	FN (内訳は右の水色枠)	3D 処理 AI の見逃し	埋まり	オクルージョン	
5-2	開口部の適合率・再現率	適合率・再現率 0.8 以上	window (窓)	959	709	250	14	219	17	全体の再現率は目標値を達成できなかった。埋まりやオクルージョンが原因で MMS の三次元点群だけでは限界がある
			entrance (入口)	189	176	13	4	0	9	
			fence (柵付き窓)	60	37	23	3	20	0	
			balcony (バルコニー)	523	472	51	15	0	31	
全体の再現率：80.53%										

表 7-16 検証結果サマリー (2D 処理と 3D 処理の結合の評価実験)

													赤セル：KPI 達成		青セル：KPI 未達	
No	検証内容	目標値	ラベル	2D/3D 結合後の結果		ハイブリッドの有用性		2D で補えなかった数とその原因				3D 処理対象外で、2D 処理で取得できた数 (扉、シャッターなど)	示唆			
				実開口部数	TP+TFP	3D の抽出漏れ	2D で補えた数	2D の AI の見落とし	画像に写っていない	画質が悪い	オクルージョン					
5-2	開口部の適合率・再現率 0.8 以上	適合率・再現率 0.8 以上	window (窓)	959	812	250	103	20	121	2	4	9	3D だけなら再現率約 80%だが、2D/3D にすると、88.6%になった。画像に写っていない部分も多いため、撮影角度などを変更すればさらに精度が向上すると考えられる			
			entrance (入口)	189	182	13	6	0	3	0	4	32				
			fence (柵付き窓)	60	52	23	15	1	1	0	6	26				
			balcony (バルコニー)	523	488	51	16	0	22	0	13	24				
				全体の再現率：88.6%												

6) LOD2 生成時間

● 検証結果まとめ

- LOD2 生成時間に関する 3D 都市モデルの検証結果では、4 地域 (柏市、台東区、北区赤羽、京都市) において 500m 四方のモデル生成時間を計測した。生成時間は柏市 613 秒、台東区 672 秒、北区赤

羽 565 秒、京都市 540 秒であり、全地域において目標値である 15 分（900 秒）以内での生成を達成した。これにより、本システムが多様な都市環境において安定した高速生成能力を有し、3D 都市モデルの効率的な整備に十分な性能を持つことが示された

表 7-17 検証結果サマリー（LOD2 生成時間）

					赤セル：KPI 達成	青セル：KPI 未達
No.	検証内容	目標値	生成時間		示唆	
6-1	LOD2 生成 時間	500m 四方を 15 分で生成	柏市	613 秒	<ul style="list-style-type: none"> <li>• 全体的に目標に達し、十分高速で生成することができた</li> </ul>	
6-2			江東区	672 秒		
6-3			北区赤羽	565 秒		
6-4			京都市	540 秒		

7) LOD3 (MMS) 生成時間

- 検証結果まとめ
  - LOD3 生成時間に関する 3D 都市モデルの検証結果では、北区赤羽において 100m 換算での処理時間を計測した。目標値である 15 分（900 秒）を大幅に下回る結果となった。これにより、本システムが LOD3 レベルの詳細な建築物モデル生成においても十分な高速処理能力を有することが示された。

表 7-18 検証結果サマリー(LOD3 生成時間)

					赤セル：KPI 達成	青セル：KPI 未達
No.	検証内容	目標値	生成時間		示唆	
7-1	処理時間 (100m 換算)	15 分	北区赤羽	291 秒	目標に達し、十分高速で処理することができた	

## 8. 実証技術の非機能要件の検証

### 8-1. 検証目的

### 8-2. KPI

表 8-1 非機能要件の KPI 一覧

カテゴリ	ID	項目	詳細
可用性	NR001	システムの稼働時間	● 3D 都市モデルを生成するため 5 時間程度稼働すること
性能・拡張性	NR002	リファレンスデータの読み込み速度	● システム利用者が不快なくシステム運用を行うことを想定し、リファレンスデータの読み込みを 30 秒以内で行うこと
	NR003	画面描画速度	● 描画フレームレートが 30fps 以上であること
ユーザビリティ	NR004	操作性	● ユーザーが操作できる UI が明確に提示されていること
	NR005	デザイン	● デザイン・デザインルールの標準化がされており、分かりやすい UI となっていること
	NR006	デザイン	● ユーザーの操作に対する反応が適切にあること
	NR007	アクセシビリティ	● 色覚異常に対して見やすいカラースキームとなっていること
その他	NR008	OSS	● システムを OSS として公開すること

### 8-2-1. 検証方法と検証シナリオ

表 8-2 非機能要件の検証方法

対象項目	品質評価項目	目標値	期間の単位	アクティビティ
3D 都市モデル生成システム	システム稼働時間	● 5 時間	2026 年 1 月～2 月	● 運用テストによる検証
	リファレンスデータ読み込み速度	● 30 秒	2026 年 1 月～2 月	● 運用テストによる検証
	画面描画速度	● 30fps	2026 年 1 月～2 月	● 運用テストによる検証

### 8-2-2. 検証結果

実証実験を実施するに当たり、必要となる稼働時間、データ読み込み時間については目標値を達成できた。

表 8-3 検証結果サマリー

赤セル：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
システム稼働時間	稼働時間	5 時間	8 時間	● 安定して目標値を超える時間稼働した
リファレンスデータ読み込み速度	読み込み時間	30 秒	10 秒	● 十分高速にデータを読み込むことができた
画面描画速度	フレームレート	30fps	59fps	● 目標に達し、十分高速で使うことができた

## 9. 成果と課題

### 9-1. 本実証で得られた成果

#### 9-1-1. 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 9-1 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
アルゴリズム	3D 都市モデル生成	<ul style="list-style-type: none"> <li>● 従来の 3D 都市モデル作成では、航空測量データの取得や専門技術者による手動モデリングに依存しており、広域の都市モデル整備に多大な時間とコストを要するという課題があった</li> <li>● 本システムでは衛星画像と建築物フットプリントデータを入力として生成 AI により建築物モデルを自動生成するため、従来数ヶ月を要していたモデリング作業を約 1 時間に短縮することが可能であり、LOD2 レベルの 3D 都市モデルを迅速かつ低コストで整備できる点で技術的優位性を持つ</li> </ul>
	開口部情報抽出	<ul style="list-style-type: none"> <li>● LOD3 に必要な開口部情報を高精度に抽出するには手動に頼らざる得ない課題があった</li> <li>● 本システムでは、MMS の沿道画像と三次元点群をハイブリッド的に利用することで、両者のメリットデメリットを補完しあった開口部抽出手法を開発し、高い精度で開口部の自動抽出が可能になった。三次元点群だけでも約 80%程度の再現率（シャッター、扉は除く）となり、2D/3D のハイブリッド式では、約 89%（シャッター、扉も含む）となった。実験データも実際の道路数 km を対象としており、既存の研究に比べ実験の汎用性や信頼性も高い</li> </ul>
入力データ	衛星画像	<ul style="list-style-type: none"> <li>● LP 航空測量は高精度である一方で高コストであるという課題があった</li> <li>● 本システムは、LP 航空測量データの代わりにオープンな衛星画像を入力して用いるために低コストで 3D 都市モデルを生成することが可能であり、全国の PLATEAU データを学習データに用いることで PLATEAU 品質同等の 3D 都市モデルを生成可能である</li> </ul>
	MSS	<ul style="list-style-type: none"> <li>● LOD3 に必要な開口部情報を高精度に抽出するには手動に頼らざる得ない課題があり、コスト的にも労力的にも課題があった</li> </ul>

		<ul style="list-style-type: none"> <li>● 本システムでは、他業務で取得された既存の MMS データ（沿道画像と三次元点群）を利用した開口部の自動抽出手法を開発した。既存資産での自動抽出を可能としているため、入力データの的にも有用性や汎用性が高い</li> </ul>
--	--	---

## 9-1-2. 3D 都市モデルのビジネス面での優位性

表 9-2 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
低コスト化	自動生成による低コストでの広域整備	<ul style="list-style-type: none"> <li>● LOD2 の整備範囲は従来手法の場合整備コストが高く整備範囲が限られており、データ整備されていないエリアでは、太陽光発電量の推計等のユースケースを未整備エリアへの展開が困難であった。またデータ更新においては建築物と建て替わりは限られるため、更新コストの低コスト化が課題となっている</li> <li>● 本システムでは広域で入手可能な衛星画像を入力データとしており、なおかつ生成 AI を用いて自動生成が可能であるため、PLATEAU 未整備エリアでも容易にデータ整備が可能である</li> <li>● これにより低コストで広域の 3D 都市モデルを生成することが可能になり、PLATEAU 未整備エリアにおけるユースケースのスケールアップ（広域展開）が可能となる。また衛星画像のみによる自動生成が可能になったことによってデータ更新の低コスト化に寄与する</li> </ul>
	品質検証の自動化	<ul style="list-style-type: none"> <li>● PLATEAU の標準仕様の検査項目では、ジオメトリの精度と CityGML の検査項目はあるが外観の定量的評価指標は定義されていないため、目視による検査が行われておりコストが高い</li> <li>● 本システムでは外観のノイズ等の品質を定量的に評価する BMQI 指標により、自動で各建築物のノイズの有無を数値化することが可能である</li> <li>● これにより目視等の検査をせずとも自動で外観のノイズを評価できるため、担当者や自治体間の違いによる品質の差異を最小化できるために全国均一品質のデータ整備が可能になる</li> </ul>

## 9-2. 実証実験で得られた課題と対応策

表 9-3 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	計算資源	<ul style="list-style-type: none"> <li>● 生成 AI は GPU の計算コストが高いため、3D 都市モデルの生成に時間を要する</li> </ul>	<ul style="list-style-type: none"> <li>● 計算コストの削減と GPU の増強を行うことで生成過程を効率化する</li> </ul>
	可視化	<ul style="list-style-type: none"> <li>● 生成結果において、3D 空間上で各地物の属性の確認ができないため、別途、ファイルを開いて属性を確認するのが困難である</li> </ul>	<ul style="list-style-type: none"> <li>● 可視化の際に各地物の属性情報のテーブルを確認できるようにすることで、容易に各地物の属性を確認できるように修正する</li> </ul>
	開口部情報抽出精度の向上 (LOD3)	<ul style="list-style-type: none"> <li>● 画質や撮影角度の影響で正しく 2D 処理の開口部抽出ができない課題があった</li> <li>● 開口部の場所は正しく抽出できているが、形状の精度が良くない場合がある</li> <li>● 開口部のモデル化を平面矩形で描画しているため、見栄えが良くなかった（特にバルコニー）</li> <li>● 障害物などのオクルージョンの影響で AI の精度が悪くなる課題があった</li> <li>● 2D 処理の AI の誤抽出が少し多かった（窓以外）</li> <li>● ラベルが間違えることが多かった</li> <li>● カメラ撮影位置と傾きの記載フォーマットが、MMS のメーカーによって異なる。そのため、現状は 2 社のフォーマットのみに対応</li> <li>● MMS データは建築物の上の方が写らない</li> <li>● Windows では高速であるが、Linux では速度が遅くなる</li> </ul>	<ul style="list-style-type: none"> <li>● LOD3 作成に適したカメラ配置などの検討</li> <li>● 開口部抽出後に再度三次元点群を利用して正確な形状を取得する</li> <li>● LOD3 の描画方法を検討し、突起物なども正しく描画できる方法を実装する</li> <li>● 連続画像から障害物を除去するアルゴリズムの実装</li> <li>● 扉などの学習データを増やすことで対応</li> <li>● 学習データや方法などの再検討</li> <li>● 点群と画像を統一的に管理するフォーマットの標準化を検討</li> <li>● UAV のデータなどとの連携を検討</li> <li>● 点群読み込みライブラリなどの検討及び原因追及</li> </ul>

システム	建築物モデル生成精度の向上 (LOD2)	<ul style="list-style-type: none"> <li>● 生成 AI モデルの学習データが特定の建築様式に偏っており、多様な屋根形状（寄棟、入母屋、複合屋根等）に対する生成精度が不均一である</li> <li>● 建築物フットプリントの形状が複雑な場合(L字型、コの字型等)、生成されるメッシュモデルに自己交差や非多様体構造が発生する</li> <li>● ノイズ評価の閾値設定が固定的であり、建築物の用途や規模によって最適な品質基準が異なる</li> </ul>	<ul style="list-style-type: none"> <li>● 学習データセットの拡充と多様化を行い、日本の建築様式を網羅的にカバーする学習データの増強を実施する</li> <li>● メッシュ後処理アルゴリズムを強化し、自己交差検出・修復機能とトポロジー最適化処理を追加する</li> <li>● 建築物の分類に応じた動的な閾値設定機能を実装し、適応的な品質管理を実現する</li> </ul>
サービス運用	ユースケースの応用	<ul style="list-style-type: none"> <li>● 衛星画像と生成 AI によって生成された 3D 都市モデルをユースケースに適用する際には、PLATEAU とのデータ特性の違いを確認する必要がある</li> </ul>	<ul style="list-style-type: none"> <li>● ユースケースへの適用を念頭に PLATEAU との誤差やデータ特性の違いをまとめたマニュアルを作成することで、生成 AI による 3D 都市モデルの精度保証をする</li> </ul>

### 9-3. 今後の展望

本システムは、PLATEAU に代表される 3D 都市モデル整備において課題となっている整備コストの高さや高頻度な更新に対応するため、衛星画像および AI を活用した低コストかつ広域適用可能な 3D 都市モデリング手法の確立を目的として開発を行ったものである。具体的には、衛星画像、MMS データ等の既存データと生成 AI を組み合わせることで、建築物、道路、植生、都市設備を対象に、LOD0 から LOD3 までの 3D 都市モデルを自動生成するフレームワークを構築した。

本研究では、衛星画像を用いたリファレンス機能の導入により、広域かつ均質なデータ整備を可能とするとともに、MMS による沿道画像および三次元点群データを活用した開口部情報抽出手法を開発した。これにより、従来は人手による作業や高コストな測量に依存していた建築物形状や外観情報の取得について、自動化／工数削減の可能性を示した。これらの成果は、PLATEAU を補完するデータ整備手法として、整備カバレッジの拡大、更新頻度の向上、段階的な高度化を実現する基盤技術となることが期待される。

一方で、実運用を見据えた場合、いくつかの技術的課題が残されている。特に GPU を用いた生成 AI 処理に伴う計算コストの高さは、全国規模での継続的なデータ整備を行う上での制約要因となっている。このため、アルゴリズムの最適化や処理フローの改善等により、計算効率の向上を図る必要がある。これにより、計算時間および運用コストを抑制しつつ、複雑な形状を有する 3D 都市モデルの生成を可能とすることが期待される。また、開口部情報抽出については一定の精度を達成しているものの、都市景観の再現性向上に向けて、オクルージョン対応等を含むさらなる高精度化が求められる。

今後は、これらの技術的課題に対応しつつ、実証実験を通じて本手法の実用性および有効性を検証するフェーズへ移行することが重要である。特に、都市域レベルでの広域適用を想定し、PLATEAU との違いや補完関係を明確にした上で、都市計画、景観シミュレーション、防災・インフラ管理等の具体的なユースケースに基づく評価を行う必要がある。

これらの検証を通じて、生成 AI を活用した 3D 都市モデル整備手法の社会実装可能性を高め、持続的な都市データ基盤構築への貢献を目指す。

## 10. 用語集

### A) アルファベット順

表 10-1 用語集（アルファベット順）

No.	用語	説明
1	API（Application Programming Interface）	異なるソフトウェア同士が機能やデータをやり取りするためのインタフェース。本システムでは REST 形式の API 通信を利用する
2	BMQI（Building Mesh Quality Index）	生成された建築物メッシュの品質を評価するための指標。屋根接続性、側面の鉛直性、フットプリント整合性などを統合的に評価する
3	CityGML	都市空間の三次元地理情報を表現するための国際標準フォーマット。PLATEAU の 3D 都市モデルもこの規格に基づいている
4	DEM（Digital Elevation Model）	地表面の標高を格子状に記録した地形データ。本システムでは地形モデル生成に利用する
5	GAN（Generative Adversarial Network）	画像生成などに用いられる生成 AI モデルの一種。生成器と識別器の競合学習によって高品質なデータ生成を行
6	GeoJSON	地理空間データを JSON 形式で表現するフォーマット。建築物フットプリントなどのデータ交換に利用される
7	GPU（Graphics Processing Unit）	並列計算に優れたプロセッサ。本研究では生成 AI モデルの学習や推論を高速化するために利用する
8	HPC（High Performance Computing）	大規模計算を高速に処理する計算基盤。本研究では GPU クラスタを利用して生成モデルを学習する
9	IOU（Intersection over Union）	画像セグメンテーションの精度評価指標。予測領域と正解領域の重なり割合を示す
10	MMS（Mobile Mapping System）	車両に搭載したセンサで道路周辺の画像や点群を取得する測量システム
11	PyTorch	深層学習モデルの開発・実行に用いられるオープンソースライブラリ。
12	RANSAC	ノイズを含むデータから平面なモデルを推定するアルゴリズム
13	RESTful API	REST アーキテクチャに基づく Web API 通信方式
14	RMSE（Root Mean Square Error）	推定値と真値の誤差を評価する統計指標
15	Transformer	自己注意機構を用いた深層学習モデル。3D メッシュ生成モデルとして利用する
16	YOLO（You Only Look Once）	リアルタイム物体検出アルゴリズム。画像から窓や扉などの開口部を検出する際に利用する

## B) 五十音順

表 10-2 用語集（五十音順）

No.	用語	説明
1	インスタンスセグメンテーション	画像内の個々の物体を識別し、それぞれを独立したオブジェクトとして抽出する画像解析手法。建築物輪郭抽出などに利用される
2	オクルージョン	建物や樹木などの遮蔽物により対象物が隠れて見えなくなる現象。点群解析や画像解析の精度に影響する要因となる
3	開口部	建築物の壁面に存在する窓、ドアなどの開口構造。LOD3 建築物モデル生成に必要な情報である
4	生成 AI	画像・文章・3D モデルなど新しいデータを自動生成する人工知能技術の総称
5	三角メッシュ	三角形ポリゴンを組み合わせて三次元形状を表現する方法。3D 都市モデルの基本的な構造
6	三次元点群	レーザースキャナなどによって取得された三次元座標の集合。建築物形状の解析やモデル生成に利用される
7	自己回帰生成	過去の出力を入力として次のデータを順次生成するモデル方式
8	ジオメトリ誤差	生成された 3D モデルと実際の形状との差異を示す誤差指標
9	深層学習	多層のニューラルネットワークを用いてデータの特徴を学習する機械学習手法
10	セマンティックセグメンテーション	画像内の各ピクセルを道路、建築物、植生などの意味カテゴリごとに分類する画像解析手法
11	ノイズ	3D モデルにおける不要な凹凸や形状の乱れ
12	ポリゴン	複数の頂点で構成される多角形データ。GIS で地物形状を表す際に使用される
13	ポリゴン化	画像などのラスターデータをポリゴンなどのベクトルデータへ変換する処理
14	メッシュ	ポリゴンの集合によって三次元形状を表現する構造
15	ラスターデータ	格子状の画素で構成される画像データ。衛星画像などが該当する
16	ベクトルデータ	点・線・ポリゴンなどの幾何形状で表現された地理情報データ
17	並列処理	複数の計算を同時に実行して処理時間を短縮する計算方式
18	地理情報システム	地理空間データの管理、分析、可視化を行うシステム

以上

3D 都市モデル生成シミュレータの開発

技術検証レポート

2026 年 3 月 発行

委託者：国土交通省 都市局

受託者：国立大学法人東京大学