



PLATEAU
by MLIT

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



3D 都市モデルを活用した高精度デジタルツイン構築 技術検証レポート

series
No. 69

Technical Report on the Development of High-Precision Digital Twins Utilizing
3D City Models

目次

目次	- 1 -
1. ユースケースの概要	- 1 -
1-1. 現状と課題	- 1 -
1-2. 課題解決のアプローチ	- 1 -
1-3. 創出価値	- 3 -
1-4. 想定事業機会	- 3 -
2. 実証実験の概要	- 4 -
2-1. 実証仮説	- 4 -
2-2. 実証フロー	- 4 -
2-3. 検証ポイント	- 5 -
2-4. 実施体制	- 5 -
2-5. 実証エリア	- 6 -
2-6. スケジュール	- 8 -
3. 実証システム	- 9 -
3-1. アーキテクチャ	- 9 -
3-1-1. システムアーキテクチャ	- 9 -
3-1-2. データアーキテクチャ	- 12 -
3-1-3. ハードウェアアーキテクチャ	- 13 -
3-2. システム機能	- 15 -
3-2-1. システム機能一覧	- 15 -
3-2-2. 利用したソフトウェア・ライブラリ	- 17 -
3-2-3. 開発機能の詳細要件	- 18 -
3-3. アルゴリズム	- 32 -
3-3-1. 利用したアルゴリズム	- 32 -
3-3-2. 開発したアルゴリズム	- 35 -
3-4. データインタフェース	- 38 -
3-4-1. ファイル入力インタフェース	- 38 -
3-4-2. ファイル出力インタフェース	- 39 -
3-4-3. 内部連携インタフェース	- 42 -
3-4-4. 外部連携インタフェース	- 44 -
3-5. 実証に用いたデータ	- 45 -
3-5-1. 活用したデータ一覧	- 45 -
3-5-2. 生成・変換したデータ	- 49 -
3-6. ユーザーインタフェース	- 50 -
3-6-1. 画面一覧	- 50 -
3-6-2. 画面遷移図	- 50 -

3-6-3. 各画面仕様詳細.....	- 51 -
3-7. 実証システムの利用手順.....	- 53 -
3-7-1. 実証システムの利用フロー.....	- 53 -
3-7-2. 各画面操作方法.....	- 54 -
4. 実証技術の検証.....	- 57 -
4-1. シミュレーション精度の検証.....	- 57 -
4-1-1. 検証目的.....	- 57 -
4-1-2. KPI.....	- 57 -
4-1-3. 検証方法と検証シナリオ.....	- 58 -
4-1-4. 検証結果.....	- 59 -
4-1-5. 渋谷のデジタルツインデータ生成.....	- 60 -
4-2. パフォーマンス評価の検証.....	- 65 -
4-2-1. 検証目的.....	- 65 -
4-2-2. KPI.....	- 65 -
4-2-3. 検証方法と検証シナリオ.....	- 65 -
4-2-4. 検証結果.....	- 66 -
4-3. デジタルツインデータの位置精度検証.....	- 71 -
4-3-1. 検証目的.....	- 71 -
4-3-2. KPI.....	- 71 -
4-3-3. 検証方法と検証シナリオ.....	- 71 -
4-3-4. 検証結果.....	- 72 -
5. 測量成果としての有用性検証.....	- 74 -
5-1. 検証の前提について.....	- 74 -
5-1-1. 検証データの概要.....	- 74 -
5-1-2. 3D モデルデータの形状.....	- 74 -
5-1-3. 検証に使用したデータ.....	- 75 -
5-1-4. 検証対象範囲.....	- 76 -
5-2. 完全性の検証.....	- 77 -
5-2-1. 検証目的.....	- 77 -
5-2-2. KPI.....	- 77 -
5-2-3. 検証方法と検証シナリオ.....	- 78 -
5-2-4. 検証結果.....	- 81 -
5-3. 位置正確度の検証.....	- 89 -
5-3-1. 検証目的.....	- 89 -
5-3-2. KPI.....	- 89 -
5-3-3. 検証方法と検証シナリオ.....	- 89 -
5-3-4. 活用技術で想定される誤差.....	- 91 -
5-3-5. 検証結果.....	- 95 -

6. BtoB ビジネスでの有用性検証.....	- 102 -
6-1. 検証目的	- 102 -
6-2. 検証方法	- 102 -
6-3. 被験者	- 103 -
6-4. ヒアリング・アンケートの詳細.....	- 104 -
6-4-1. 検証項目と評価方法	- 104 -
6-4-2. 実証実験の様子.....	- 105 -
6-5. 検証結果	- 112 -
7. 成果と課題	- 119 -
7-1. 本実証で得られた成果.....	- 119 -
7-1-1. 3D 都市モデルの技術面での優位性	- 119 -
7-1-2. 3D 都市モデルのビジネス面での優位性	- 119 -
7-2. 事業化に向けた課題と解決策の案.....	- 121 -
7-3. 今後の展望.....	- 123 -
8. 用語集.....	- 124 -

1. ユースケースの概要

1-1. 現状と課題

現実の都市を忠実に再現する 3D 都市モデルの登場により、防災やまちづくりといった分野のみならず、メタバース等のコンシューマ向けサービスへの活用の期待も高まっている。他方、現在提供されている 3D 都市モデルの多くは、テクスチャ解像度が低く、道路付帯設備や植栽等のデータも整備されていないため、そのままではコンシューマが没入できるクオリティには達していない。

3D 都市モデルを利用したコンシューマ向けのコンテンツ制作を促進するためには、ハイクオリティのテクスチャ付与や植栽、看板、信号機等の都市設備を追加した高品質デジタルツインデータが必要である。

1-2. 課題解決のアプローチ

今回の実証実験では、コンシューマ向けサービスにも活用できる高精度かつハイクオリティなデジタルツインデータの自動生成を実現するため、AI 等を活用した自動生成技術・手法の確立を図る。

具体的には、3D 都市モデルと衛星写真から都市情報をデータベース化し、プロシージャルモデリングを活用したデジタルツイン生成 AI を開発する。この AI に 3D 都市モデルから取得した対象エリア内の地物の位置情報及び 2D 図形情報を正解データとして学習させることで、衛星写真のみでは捉えられない建物の位置情報や高さを補完する。

これにより、建築物に対する高解像度テクスチャの付与や、都市設備を追加した高精度なデジタルツインデータの生成を可能とする。また、生成した高精度デジタルツインデータをゲームエンジンで利用できるようにローポリゴン化し、コンシューマ向けサービスコンテンツとして活用可能にする。

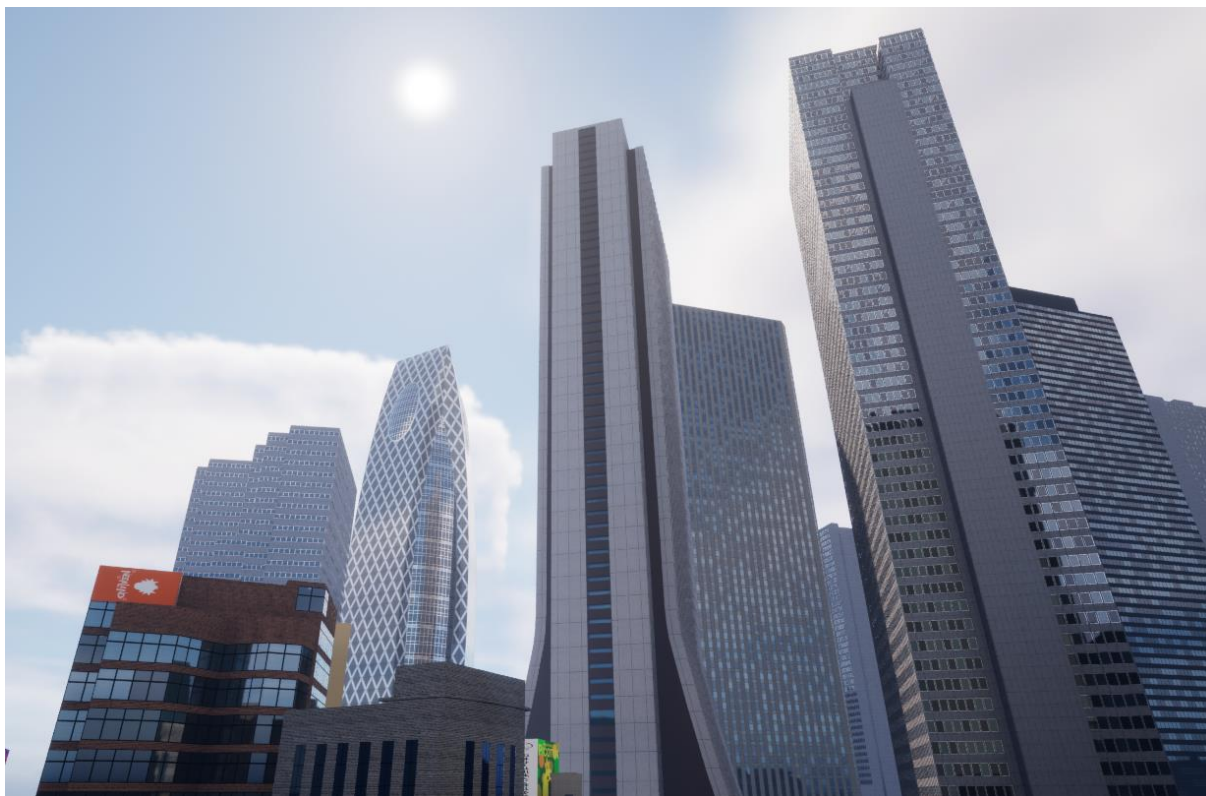


図 1-1.Unreal Engine に読み込んだ高精度デジタルツインデータ

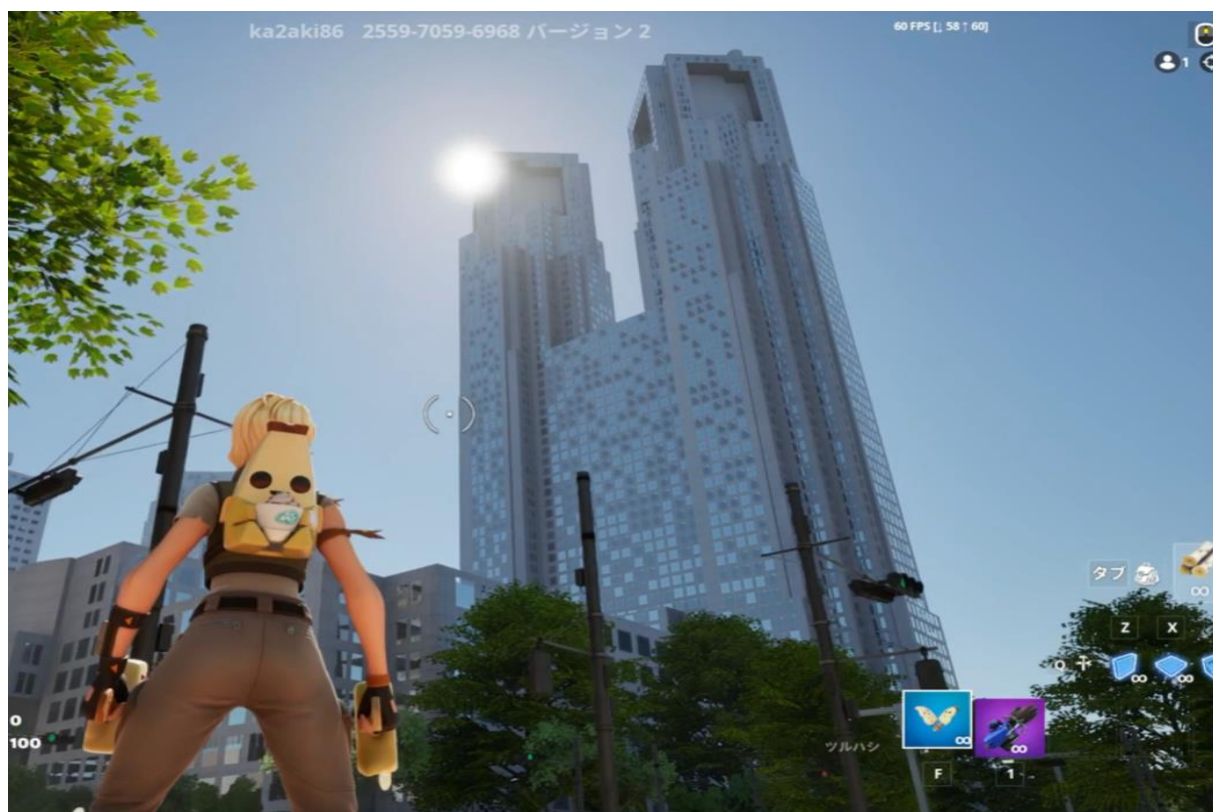


図 1-2.コンシューマ向けコンテンツ（西新宿を再現しゲーム内の空間として活用）

1-3. 創出価値

3D 都市モデルが様々な領域における価値創出をもたらしていくためには、地図としての利用のみならず、映像制作やゲーム開発、VR・ARなどの領域においても活用を拡大していく必要がある。他方、現在提供されている 3D 都市モデルの多くは、コンシューマ向けのサービスで利用できるクオリティではなく、活用には課題がある。

一方で、ハイクオリティな 3D モデルを構築するためには、人手でのモデリングが必要となることが一般的であり、高コストになりがちである。

このため、オープンデータである 3D 都市モデルを活用することで、汎用性とスケーラビリティが担保された高精度なデジタルツインデータの生成方法を確立し、コンシューマ向けサービスや都市開発など、3DCG 技術が必要とする様々なサービスへの展開によるデジタルツイン市場の拡大に寄与することを目指す

1-4. 想定事業機会

表 1-1 想定事業機会

項目	内容
利用者	<ul style="list-style-type: none"> ● ゲーム開発者 ● バーチャルリアリティ（AR・VR等）を活用したサービス提供会社 ● 映像制作会社 ● 都市開発事業者（ゼネコン・不動産事業者等） ● 運輸サービス事業者
サービス仮説	<ul style="list-style-type: none"> ● 利用者が利活用したい任意の場所のデジタルツインデータを地図上からダウンロード可能な Software as a service (SaaS)
提供価値	<ul style="list-style-type: none"> ● 3DCG 技術を必要とするサービスに対して、オープンデータである 3D 都市モデルと今回開発した本システムを活用し、位置精度や高精細なテクスチャ、都市設備付与した汎用性の高い高精度なデジタルツインデータの提供

2. 実証実験の概要

2-1. 実証仮説

- オープンデータである 3D 都市モデルを正解データとして生成 AI の機械学習を行い、高精細なテクスチャや都市設備を付与した高精度デジタルツインデータを生成する AI モデルを構築することによって、PLATEAU の利用用途が地図だけではなく、コンシューマ向けサービス利用に向けた活用余地を拡大することができるようになる。
 - ▶ 今回の実証実験では東京 23 区内のエリアを対象に高精度デジタルツインデータを作成し、ゲームエンジン向けのオープンデータ化や、Unreal Engine 5 のプロジェクトファイルとして配布し、ゲームクリエイターやユーザーのニーズ、ポテンシャルを検討する。

2-2. 実証フロー

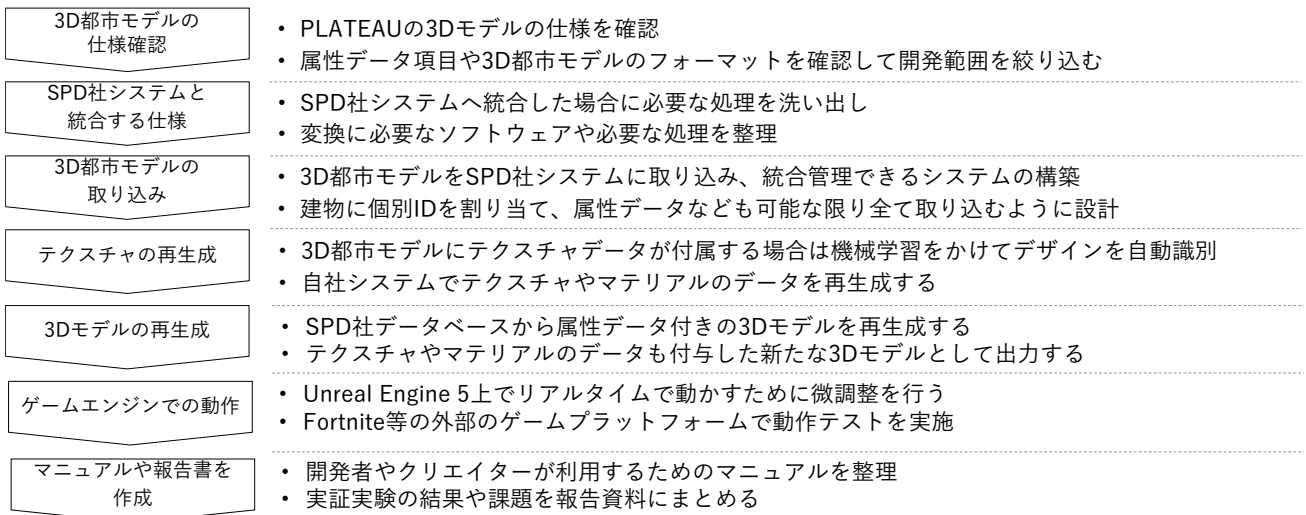


図 2-1 実証フロー

2-3. 検証ポイント

- 高精度デジタルツインデータ生成に必要な建物構造判定アルゴリズムの機能検証
 - 3D 都市モデルを活用し、より簡易且つ効率的に高精度デジタルツインデータを生成するため、建物判定アルゴリズムによって構築したデータの外見上の差異及びモデリングに必要な工数を測定することで有用性を確認する。
- AI を活用し位置精度が担保された高精度デジタルツインデータ生成システムの確立
 - 公共測量データを教師データとして追加し、AI による生成アルゴリズムを通して作成した 3D モデルの位置精度を検証することで、本システムの有用性を検証する。

上記 2 点の検証ポイントについては、【4 章：実証技術の検証】にて検証結果を記載

- 公共測量観点における高精度デジタルツインデータの位置精度検証
 - 作成した高精度デジタルツインデータを 3D 都市モデルや空中写真と比較し、地物の完全性や水平絶対位置正確度や相対位置正確度の検証を行うことで、公共測量観点での有用性を確認する。

上記 1 点の検証ポイントについては、【5 章：測量成果としての有用性検証】にて検証結果を記載

- 今回の実証実験で開発した高精度デジタルツインデータの有用性検証
 - オープンデータである 3D 都市モデルを正解データとして活用し、高精細なテクスチャや都市設備を付与した高精度デジタルツインデータを構築することによって、3D 都市モデルがコンシューマ向けサービス利用に適したデータとして利用者のニーズやポテンシャルを満たしたデータになっているかを確認する。

上記 1 点の検証ポイントについては、【6 章：BtoB ビジネスでの有用性検証】にて検証結果を記載

2-4. 実施体制

表 2-1 実施体制

役割	主体	詳細
全体管理	国土交通省 都市局	プロジェクト全体ディレクション
	アクセンチュア	プロジェクト全体マネジメント
実施事業者	スペースデータ	ユースケース実証における企画・開発・検証・運営

2-5. 実証エリア

表 2-2 実証エリア①

項目	内容
実証地	東京都新宿区
面積	4 km ²
マップ (対象エリアは赤枠内)	 <p data-bbox="359 1422 893 1456">赤枠：高精度デジタルツインデータのエリア</p> <p data-bbox="359 1467 1141 1500">青枠：Unreal Engine 5 での配布エリア（ゲームへの適用エリア）</p>

表 2-3 実証エリア②

項目	内容
実証地	東京都渋谷区
面積	4 km ²
マップ (対象エリアは赤枠内)	<p>赤枠：高精度デジタルツインデータのエリア 青枠：Unreal Engine 5 での配布エリア（ゲームへの適用エリア）</p>

2-6. スケジュール

表 2-4 スケジュール

実施事項	2023 年										2024 年		
	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月	3 月	
1.計画書の作成	↔												
2.フィジビリティスタディと仕様確定	↔	↔											
3.PLATEAU データの仕様調査		↔											
4.自社システムへの取り込み設計		↔											
5.PLATEAU モデルの統合開発			↔										
6.統合後システムのテスト			↔										
7.モデルテクスチャの生成処理				↔									
8.テクスチャの見栄えの調整				↔									
9.UE5 上での動作処理					↔								
10.開発ドキュメントの作成					↔								
11.修正・微調整						↔							
12.プロトタイプ検証・確認						↔							
13.モデルテクスチャの生成処理最終化							↔						
14.テクスチャの見栄えの最終化								↔					
15.UE5 上での動作処理の最終化									↔				
16.統合後システム最終化									↔				
17.実証調査と分析										↔	↔		
18.報告書作成											↔	↔	
19.渋谷のデータ作成											↔	↔	

3. 実証システム

3-1. アーキテクチャ

3-1-1. システムアーキテクチャ

本システムでは、AI 等を活用した自動生成技術を用い、3D 都市モデルと衛星写真から都市情報をデータベース化し、プロシージャルモデリングを活用したデジタルツイン生成 AI の開発を行なった。

スペースデータ社が保有する高精度なデジタルツインデータを自動生成する既存システムに、対象エリアの衛星画像や 3D 都市モデルを都市情報のデータベースとして追加することにより、公共測量成果である 3D 都市モデル位置正確度や都市構造物の精度を維持しつつ、プロシージャルモデリングを用いて高精細なテクスチャの付与や建物のディテールアップ、看板や信号機等の都市設備の追加等を自動付与が可能な高精度デジタルツインデータの自動生成 AI とする改修を行った。

今回追加開発したシステムにより、オープンデータである 3D 都市モデルをインプットデータとして地物の位置情報及び 2D 図形情報、高さを担保しつつ、AI を活用してテクスチャ・都市設備を再現することで、コンシューマ向けコンテンツに利用可能な高精度な都市のデジタルツインデータの自動生成を実現することを目的とする。

そのために、以下の追加開発を行なった。

まず、3D 都市モデルを建物ごとの教師データとして取り扱うことを目的として、CityGML 形式の 3D 都市モデルのデータをスペースデータ社で独自に開発した Python スクリプトを使用して個別の建物ごとに FBX 形式に変換し、3D 都市モデルが保有する地物の位置や高さなどの属性情報を地理情報データベースである「統合データベース基盤」に読み込んだ。「統合データベース基盤」は PostgreSQL で地理情報を扱うことが可能な拡張モジュールである PostGIS を利用し、建物 1 棟ごとに接地面であるフットプリント情報(建物の外周面を平面化した緯度経度のポリゴンデータ) の地理情報をデータベース化している。

次に、3D 都市モデルが持つテクスチャの画像データを「機械学習基盤」に読み込み、画像認識処理 AI を用いて、3D モデルの再生成時にプロシージャルモデリングで補完すべき建物の階層や窓の領域などを抽出した。読み込んだテクスチャの画像をスペースデータ社で独自に構築した教師データと比較することで側面構造物(窓、扉、壁面等)のデザイン・素材とその領域を予測し、出力された構造情報をレイヤー化して再度テクスチャ画像にデータとして追加している。

これらの処理によって、対象エリアの地図情報や建物の敷地データ、テクスチャ画像の構造などの情報が追加されたデータベースを、FBX 形式のテクスチャ画像や XML 形式の地図情報として出力し、3DCG ソフトウェアの Houdini 上の「3D モデル生成基盤」に読み込んだ。

「3D モデル生成基盤」ではまちの外観を高精度に再現するために、入力されたデータからロジックベースの機能を活用してテクスチャの再生成、都市設備の自動追加を行った上で 3D モデルの再生成を実現する機能を

有する。

また、建物の 3D モデル構造から入り口や屋上付近に任意の点を置き、自動販売機や広告看板の位置など予測することで、自動的に都市設備を追加・設置する処理を行っている。

そして、規則的な配置が可能な建物の窓や地上階部分の生成に対してプロシージャルモデリングを活用することで高精度な 3D デジタルツインデータとして新たに再生成し、このデジタルツインデータをゲームエンジンへマップとして入力するために 3D モデルアセット郡とアセット配置座標マッピング情報を自動生成する機能を開発した。

最後に、これらのデータをコンシューマ向けコンテンツへ適用できるようにするため、ゲームエンジン環境でローポリゴン化（データ軽量化）とゲームに適したテクスチャ・都市設備の再現度を高める処理を実施し、Unreal Engine 5 のプロジェクトファイルとして出力した。

本実証では、Houdini と Unreal Engine 5 のファイル連携をクラウド上で自動処理可能となるシステム構築した。

本システムのシステム・アーキテクチャは下図の通りである。

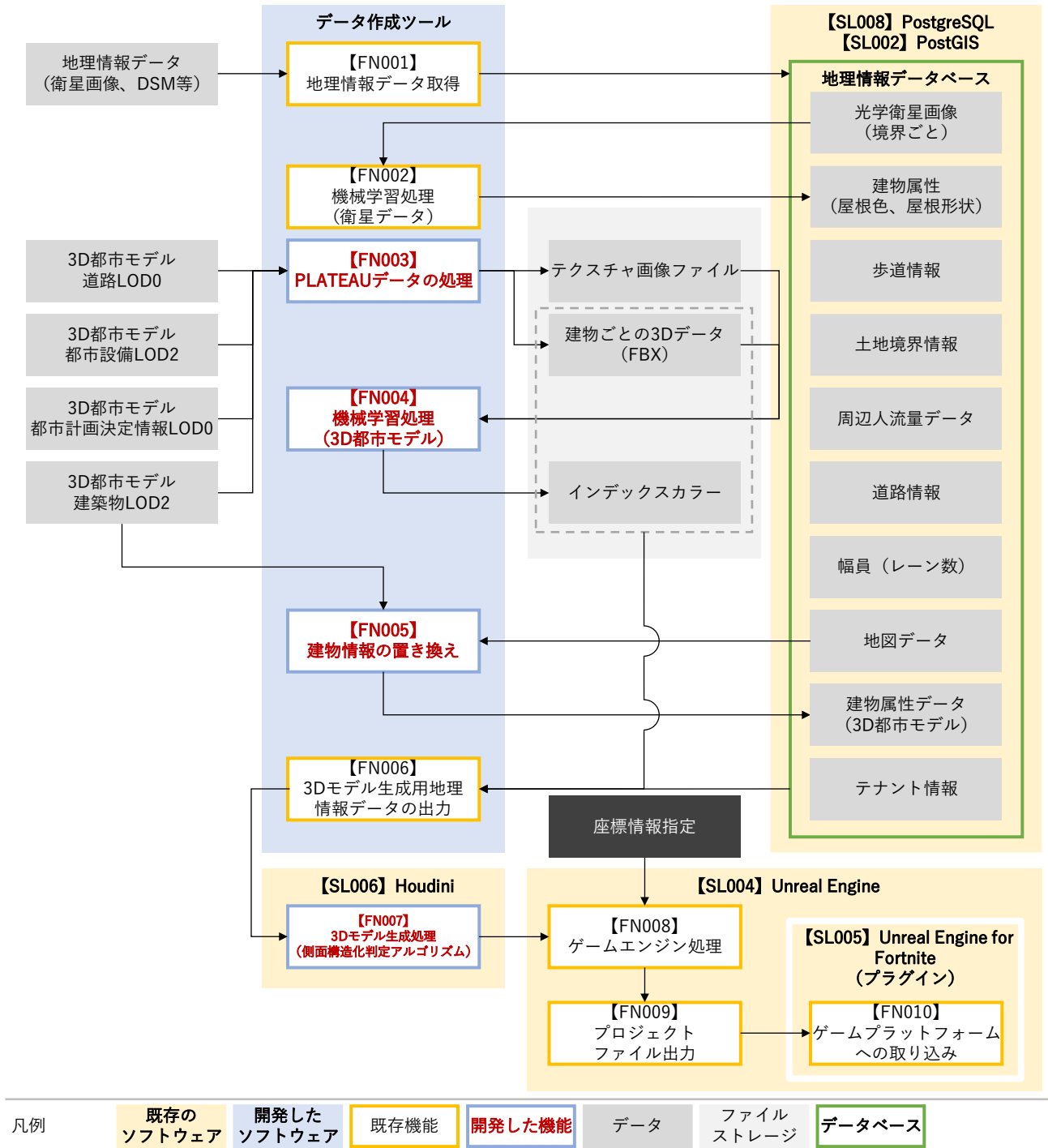


図 3-1 システムアーキテクチャ

3-1-2. データアーキテクチャ

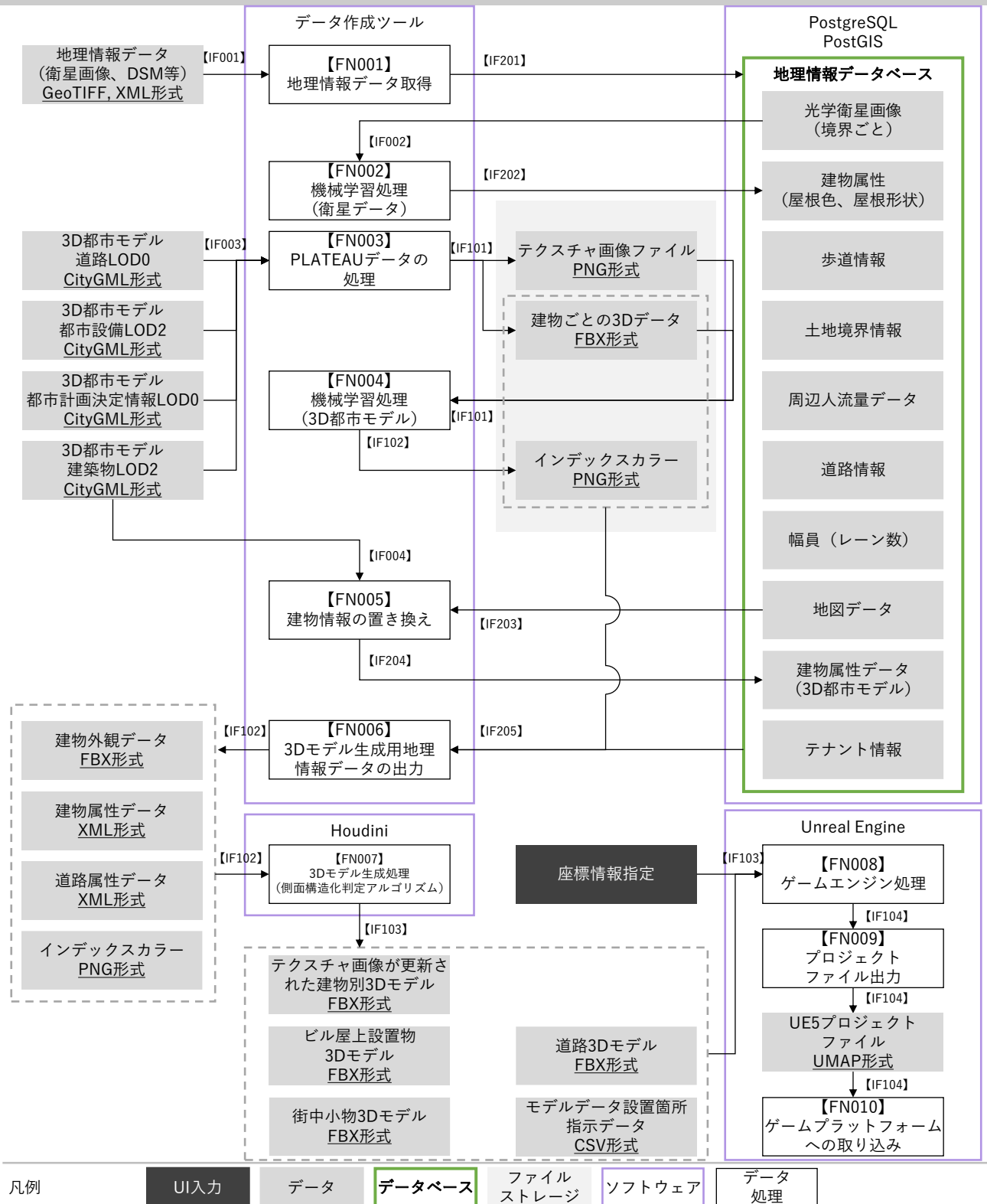


図 3-2 データアーキテクチャ

3-1-3. ハードウェアアーキテクチャ

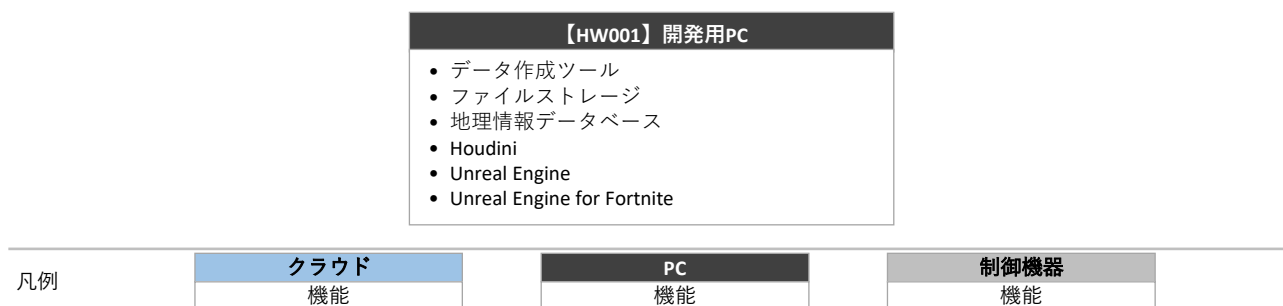


図 3-3 ハードウェアアーキテクチャ

3-1-3-a. 利用したハードウェア一覧

表 3-1 利用したハードウェア一覧

ID	種別	品番	用途
HW001	PC	HP ENVY Desktop TE02-1097jp	<ul style="list-style-type: none"> ● 機械学習 ● Houdini 実行 ● Unreal Engine、 Unreal Engine for Fortnite 実行

3-1-3-b. 利用したハードウェア詳細

1) 【HW001】 HP ENVY Desktop TE02-1097jp

- 選定理由
 - Houdini や Unreal Engine、機械学習処理に適したメモリ及び GPU を搭載している。
- 仕様・スペック
 - OS : Windows 11 Pro
 - CPU : インテル® Core™ i9-13900 プロセッサー
 - GPU : NVIDIA® GeForce RTX™ 3080 Ti Laptop GPU
 - メモリ : 64GB (16GB × 4) DDR5-4000MHz
 - ストレージ : 1TB Gen4 M.2 SSD (PCIe NVMe) + 1TB Gen4 M.2 SSD (PCIe NVMe)
- イメージ



図 3-4 HP ENVY TE02-1097jp¹

¹ 公式 HP より抜粋 : https://jp.ext.hp.com/desktops/personal/envy_te02

3-2. システム機能

3-2-1. システム機能一覧

1) 機能一覧

表 3-1 機能一覧

※赤文字：既存改修・新規開発

大分類	小分類	ID	機能名	機能説明
基本機能	共通	FN001	地理情報データ取得	<ul style="list-style-type: none"> ● 地図情報や衛星データの情報を取得
		FN002	機械学習処理（衛星データ）	<ul style="list-style-type: none"> ● 衛星データから、画像認識で屋根の色を判別
		FN003	3D 都市モデルデータの処理	<ul style="list-style-type: none"> ● 3D 都市モデルの CityGML 形式データから、建物ごとに FBX 形式の 3D モデルデータ形式へ変換
		FN004	機械学習処理（3D 都市モデル）	<ul style="list-style-type: none"> ● 3D 都市モデルにおける個別建築物のデータから、画像認識で側面構造等を判別
		FN005	建物情報の置き換え	<ul style="list-style-type: none"> ● 3D 都市モデルの CityGML データから、建物の位置情報や高さ情報を補完
		FN006	3D モデル生成用地理情報データの出力	<ul style="list-style-type: none"> ● Houdini で処理を目的として、地理情報のデータを CSV、FBX、XML ファイルで出力
		FN007	3D モデル生成処理（側面構造化判定アルゴリズム）	<ul style="list-style-type: none"> ● Houdini で 3D モデルを生成 ● 3D 都市モデルにおける個別の建築物モデルに対して側面構造化データを反映し再生成 ● 建築物モデルごとの屋上広告部分を判定 ● 再生成された 3D モデルの配置マップの生成 ● 街中に置かれる草木や標識等のアセットファイルの配置マップの生成
		FN008	ゲームエンジン処理	<ul style="list-style-type: none"> ● Unreal Engine 等で生成された配置マップやテクスチャ情報に従って、3D 空間上にモデルを設置 ● 天候情報や光の反射の反映 ● カメラ視点での見え方に対するフィルター等の反映
		FN009	プロジェクトファイル出力	<ul style="list-style-type: none"> ● サービス利用者が自身のゲームエンジンで開発ができるようにプロジェクトファイルを出力 ● 出力されたデータは配布用 Web サイトで取

				得可能
		FN010	ゲームプラットフォームへの取り込み	<ul style="list-style-type: none">● Unreal Engine for Fortnite 等のゲームプラットフォーム開発環境へのインポート処理● テクスチャ情報等の軽量化処理

3-2-2. 利用したソフトウェア・ライブラリ

表 3-3 利用したソフトウェア・ライブラリ

ID	項目	内容
SL001	Ubuntu	● Linux Distribution
SL002	PostGIS	● GIS 情報をデータベース上で扱えるようにする拡張を導入した PostgressDB ● PosgreSQL 上で GIS 情報を扱えるようにする拡張
SL003	Poetry	● Python ライブラリの管理用ソフトウェア
SL004	Unreal Engine	● Epic Games から提供されているゲームエンジン
SL005	Unreal Editor for Fortnite	● ゲームプラットフォーム特化開発エンジン
SL006	Houdini	● SideFX Software から提供されている 3D コンピュータグラフィックスソフトウェア
SL007	GDAL	● 地理空間情報データ操作ライブラリ（衛星データの機械学習用データ抽出等に利用）
SL008	PostgreSQL	● リレーショナルデータベース
SL009	NeRF	● 生成 AI モデル、画像データから 3D モデルを生成する

3-2-3. 開発機能の詳細要件

1) 機能一覧

1. 【FN001】 地理情報データ取得

- 機能概要

- 地図情報や光学衛星データを取得してデータベースへ取り込む

- フローチャート

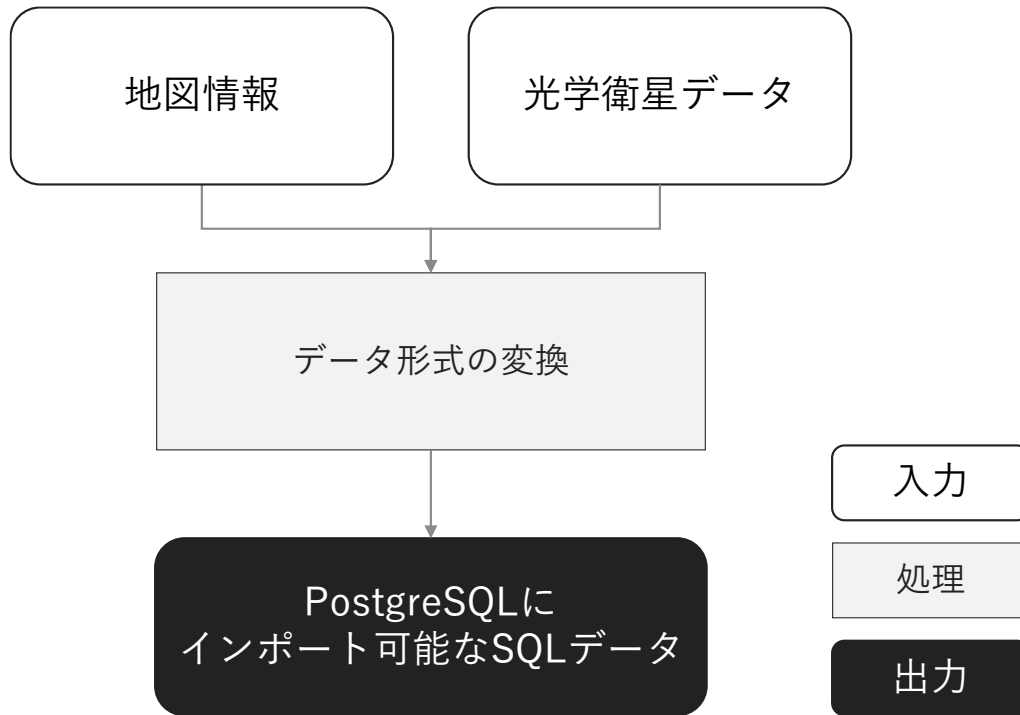


図 3-5 地理情報データ取得のフローチャート

- データ仕様

- 出力: 【IF201】

- ◇ 建物の情報

- 建物 ID

- 識別用のユニークな ID

- 側面情報

- 窓の分類

- ◇ ユニットのインデックス番号

- 色

- 屋上情報

- 屋根形状の分類

- ◇ インデックス番号

- 色

- 建物高さ

- 設置面(緯度経度ポリゴン)
 - 設置面で切り取った光学衛星データの画像
 - 広告の情報
 - 屋上広告の有無
 - 側面広告の有無
 - 入り口広告の有無
 - ◇ テナント情報
 - 建物1階部分に入っているコンビニ等テナント情報
 - テナントインデックス番号
 - テナント詳細インデックス番号
 - ◇ 土地境界(緯度経度ポリゴン)
 - ◇ 周辺人流データ
 - 日常的にどれだけの人があるかという統計データ
 - ◇ 道路の情報
 - 形状(緯度経度ポリゴン)
 - 幅員(レーン数)
 - 歩道の有無
 - ◇ 境界情報で切り取った光学衛星データの画像
 - 機能詳細
 - 利用するライブラリ
 - ◇ PostgreSQL
2. 【FN002】機械学習処理
- 機能概要
 - 光学衛星データの画像ファイルから、建物の屋根の構造や色を予測する
 - フローチャート

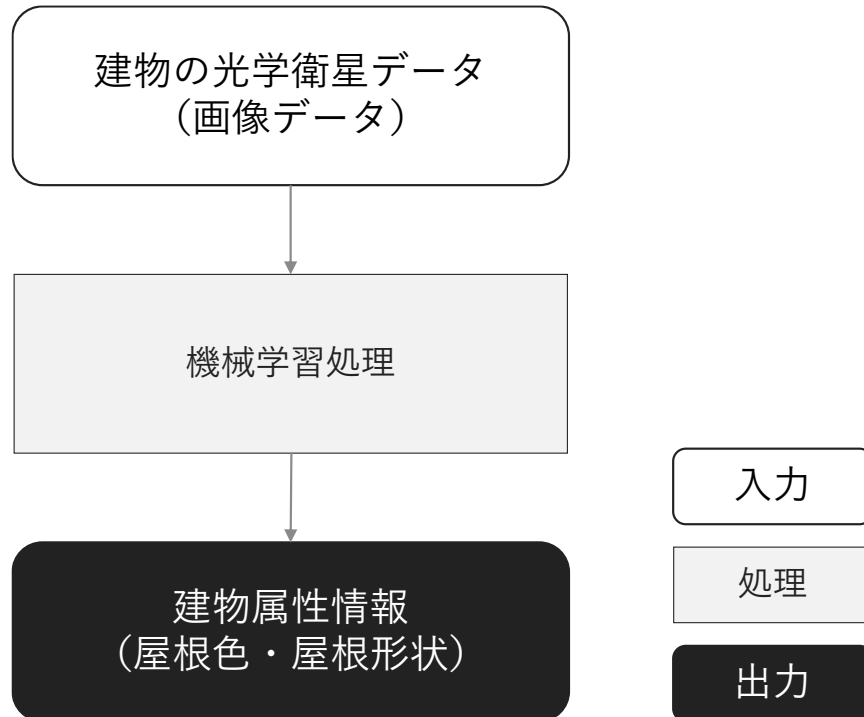


図 3-6 機械学習処理のフローチャート

- データ仕様
 - 入力
 - ◇ 建物の光学衛星データの画像ファイル
 - 出力
 - ◇ 建物属性
 - 屋根色
 - 機能詳細
 - 利用するライブラリ
 - ◇ OpenCV
 - ◇ Timm
 - 利用するアルゴリズム
 - ◇ Vision Transformer
3. 【FN003】 3D 都市モデルデータの処理
- 機能概要
 - 3D 都市モデルを建物ごとの教師データとして取り扱うことを目的として、CityGML 形式の 3D 都市モデルのデータをスペースデータ社で独自に開発した Python スクリプトを使用して個別の建物ごとに FBX 形式に変換する
 - フローチャート

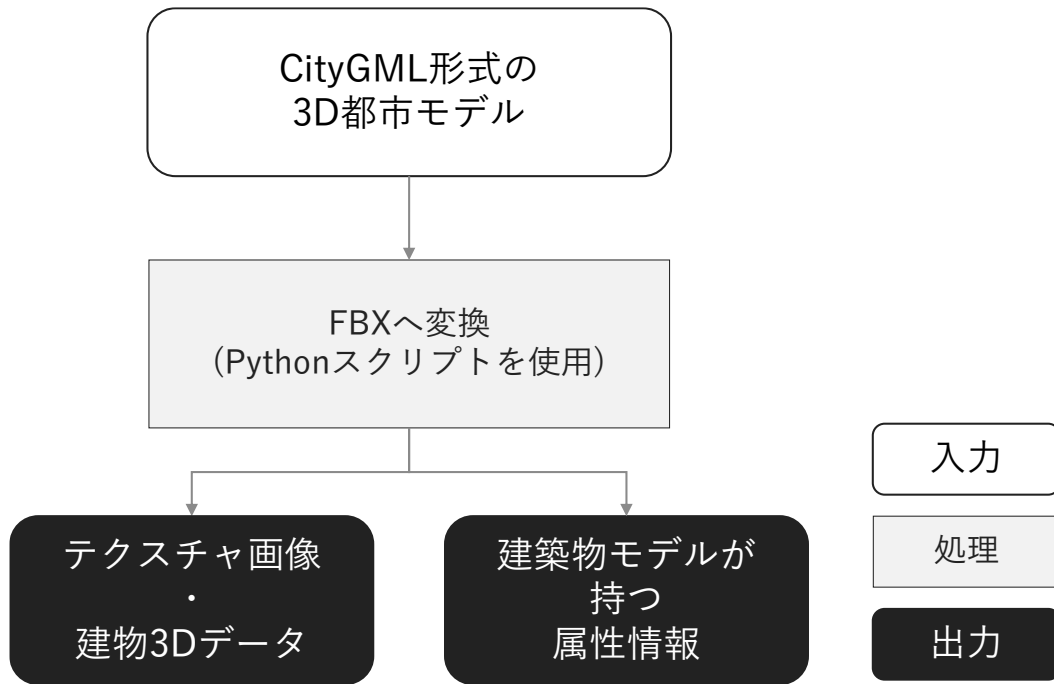


図 3-7 3D 都市モデルデータの処理フローチャート

- データ仕様

- 入力

- ◇ 3D 都市モデルの配布データ udx/bldg/フォルダに含まれる、CityGML (XML) の個別建物を持つ属性情報

- 汎用属性

- gen:stringAttribute

- ◇ name"13_区市町村コード_大字・町コード_町・丁目コード"

- 建物の情報

- bldg:class

- bldg:usage

- bldg:roofType

- bldg:measuredHeight

- bldg:storeysAboveGround

- bldg:storeyHeightsAboveGround

- bldg:Building, bldg:BuildingPart

- ◇ bldg:lod0FootPrint

- ◇ bldg:lod0RoofEdge

- ◇ bldg:lod1Solid

- ◇ bldg:lod2Solid

- ◇ bldg:lod3Solid

- bldg:outerBuildingInstallation

- ◇ bldg:BuildingInstallation
- bldg:boundedBy
 - ◇ bldg:RoofSurface
 - ◇ bldg:WallSurface
 - ◇ bldg:GroundSurface
 - ◇ bldg:OuterCeilingSurface
 - ◇ bldg:OuterFloorSurface
 - ◇ bldg:ClosureSurface
 - ◇ bldg:Door
 - ◇ bldg:Window
- bldg:address
- 建物の追加情報
 - uro:buildingIDAttribute
 - uro:buildingDetailAttribute
 - uro:largeCustomerFacilityAttribute
 - uro:keyValuePairAttribute
- 建物の表現情報
 - app:appearance
 - 建物のテクスチャ画像 (TIFF)
- 土地の情報
 - luse:class
 - luse:usage
 - luse:lod1MultiSurface
 - uro:landUseDetailAttribute
- 道路の情報
 - tran:class
 - tran:function
 - tran:usage
 - tran:trafficArea
 - tran:lod0Network
 - tran:lod1MultiSurface
- 都市設備の情報
 - frn:class
 - frn:function
 - frn:usage
 - frn:lod1Geometry
 - uro:cityFurnitureDetailAttribute
- 都市計画区域の情報

- urf:function
 - urf:usage
 - urf:areaClassification
 - urf:lod1MultiSurface
 - 出力：【IF101】
 - ◇ 建物ごとの 3D データ (FBX)
 - テクスチャ画像ファイル
 - 高さ、フットプリント等の建物に関連する情報を FBX 仕様のカスタムパラメータとして保持
 - 機能詳細
 - 使用するライブラリ
 - ◇ OpenCV
 - ◇ Lxml
 - 使用するアルゴリズム
 - ◇ なし
4. 【FN004】機械学習処理 (3D 都市モデルデータ)
- 機能概要
 - 3D 都市モデル (CityGML 形式) における建築物モデルごとテクスチャファイルから、窓の形状などの側面の構造化データを生成する
 - フローチャート

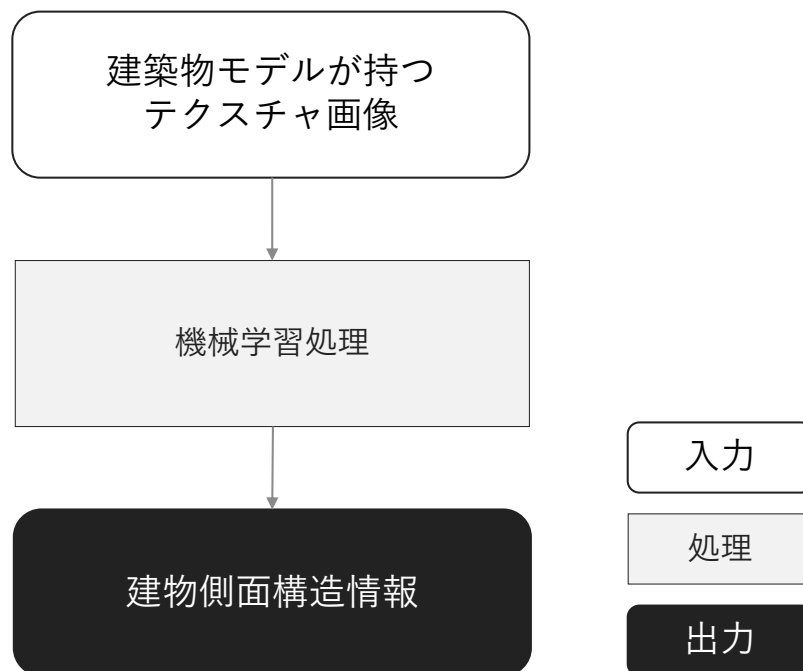


図 3-8 3D 機械学習処理 (3D 都市モデルデータ) フローチャート

- データ仕様
 - 入力
 - ◇ FBX ファイルの中にテキストで記述されている建築物モデルごとのテクスチャ情報
 - 出力：【IF102】
 - ◇ 側面構造を反映したテクスチャ画像（FBX）
 - 3D モデルに対して、どの面が広告、窓、壁なのかを PNG インデックスカラーとしてデータを保持する
 - 窓部分については、どのユニットを敷き詰めるかという情報を含む
- 機能詳細
 - 使用するライブラリ
 - ◇ OpenCV
 - ◇ Timm
 - 使用するアルゴリズム
 - ◇ Vision Transformer

5. 【FN005】 建物情報の置き換え

- 機能概要
 - 3D 都市モデルが保持する bldg:lod0RoofEdge（屋根面外周）のジオメトリデータとデータベース内の建物フットプリントのジオメトリデータカラムを ST_Intersects という 2 つの形状データを重ね合わせた時に交差の発生有無を確認する関数を使用して突合し、エリア内の各建築物同士を対応付けることで、既存のデータベースの建物の高さ情報等を 3D 都市モデルの属性情報で更新する
- フローチャート

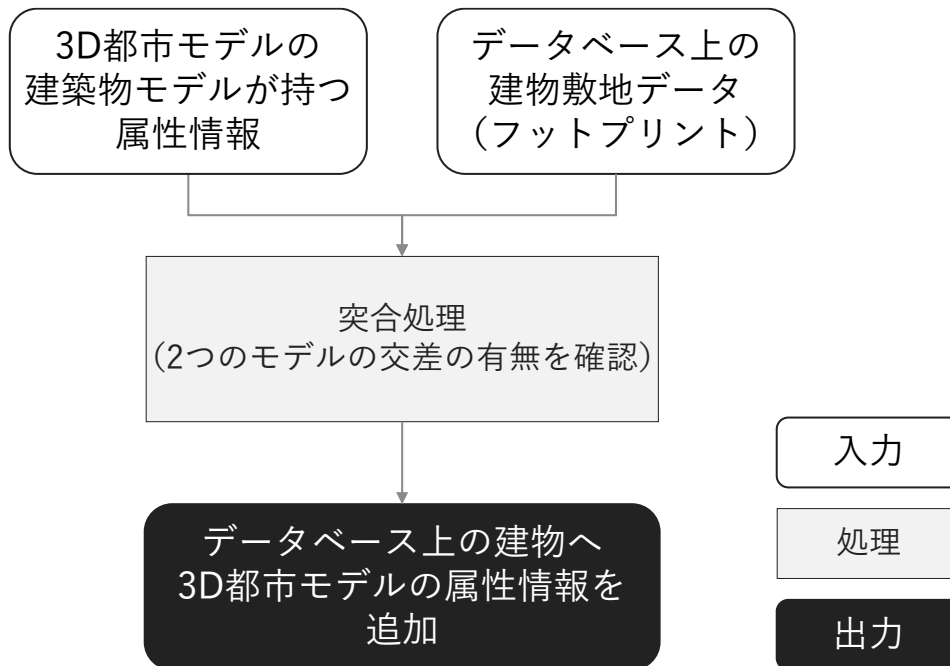


図 3-9 建物情報の置き換えフローチャート

- データ仕様
 - 入力：【IF101】
 - ◇ FBX
 - フットプリント緯度経度ポリゴン
 - 高さ
 - 建物名
 - 住所
 - 地域地区
 - 土地利用
 - ◇ データベースに取り込まれた地図データ
 - フットプリント緯度経度ポリゴン
 - 出力
 - ◇ 既存のデータベースと 3D 都市モデルのフットプリントが一致した建物について以下のデータ項目が更新される
 - 高さ
 - 建物名
 - 住所
 - 地域地区
 - 土地利用
 - 機能詳細
 - 利用するライブラリ
 - ◇ GeoAlchemy
 - 利用するアルゴリズム
 - ◇ Plane Sweep Algorithm
 - ◇ Bounding Box Check
6. 【FN006】 3D モデル生成用地理情報データの出力
- 機能概要
 - データベースに保存されている情報を Houdini で利用できる形にして出力する
 - フローチャート

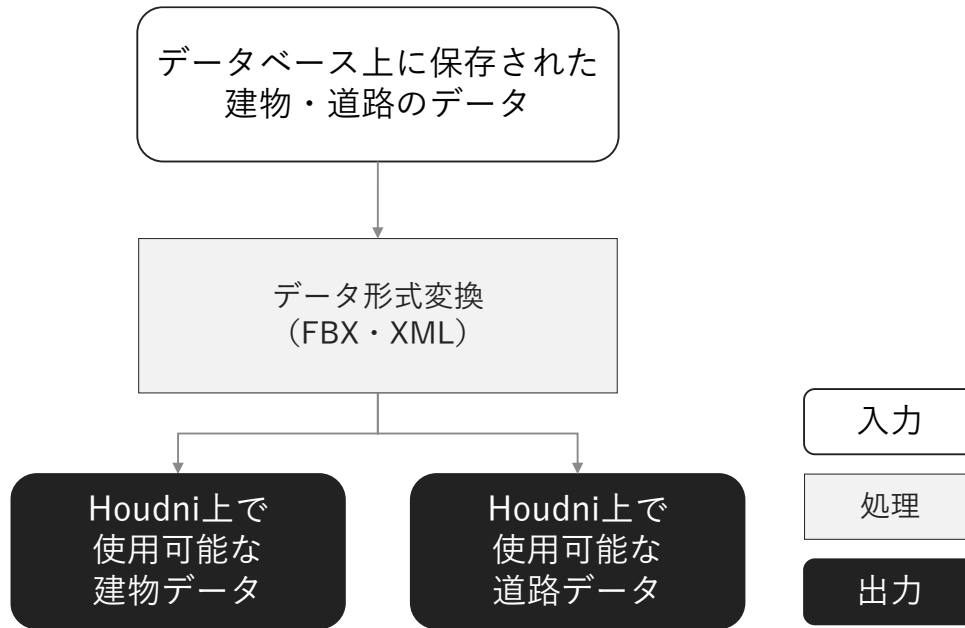


図 3-10 3D モデル生成用地理情報データの出力フローチャート

- データ仕様

- 入力：【IF205】

- 道路情報

- 緯度経度ポリゴン
 - 高架線
 - 道路タイプのインデックス番号(歩道が存在するか等)
 - 幅員

- 建物情報

- フットプリント緯度経度ポリゴン
 - FBX ID
 - 高さ
 - 屋根色
 - 屋根形状

- 側面色

- 側面形状

- ユニット ID

- ◇ FBX 側のテクスチャにユニット情報は載っているが、3D 都市モデル未整備の地区であればこのユニット ID を全体に利用する

- 周辺人流スコア

- テナント情報

- テナントインデックス番号

- 出力

- ◇ 建物データ (xml)
 - 建物 ID
 - フットプリント緯度経度ポリゴン
 - 側面色
 - 高さ
 - テナント情報
 - 側面構造
 - 周辺人流量スコア
 - 屋根色
 - 屋根形状
 - 看板情報
 - 建物名
 - 住所
 - 地域地区
 - 土地利用
 - 3D データ (FBX)
 - テクスチャ画像

- ◇ 道路データ (XML)
 - 緯度経度ポリゴン
 - 道路タイプ
 - 高架線
 - 幅員

- 機能詳細
 - 利用するライブラリ
 - ◇ Postgresql
 - ◇ lxml

7. 【FN007】 3D モデル生成処理

- 機能概要
 - 【IF102】 のテクスチャデータを参考に、3D 都市モデルの 3D モデルデータにユニット（壁を構成する最小単位 3D モデル）を敷き詰めることで建物の 3D モデルを再構成する
 - ゲームエンジン連携するため、建物側面構造化データを生成し、建物等の配置マップを生成する
 - 街中小物を設置するため、建物の入口や道路に面している部分に任意の点を設置し一定の確率で自転車や自動販売機が設置されるように配置マップを生成する
- フローチャート

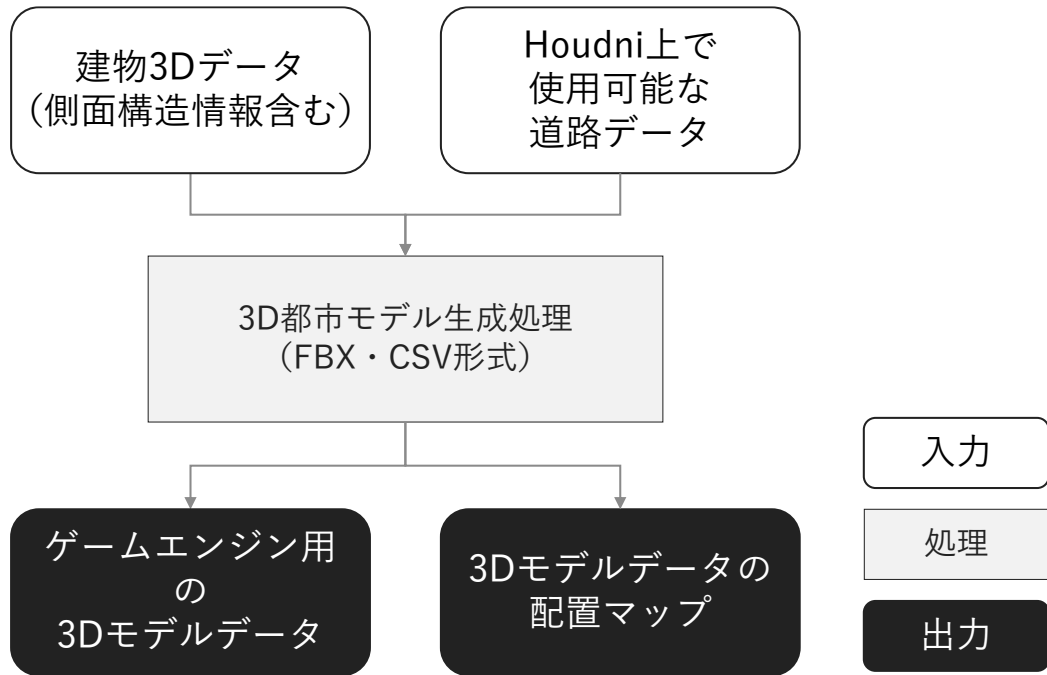


図 3-11 3D モデル生成処理フローチャート

- データ仕様
 - 入力
 - ◇ CityGML の udx/bldg/フォルダに含まれる、GML ファイルから生成された 3D モデル
 - ◇ 機械学習により生成された面構造化データ
 - テクスチャ画像の色を Index 番号で管理し、どの側面がどの側面のユニットなのかを指定する画像ファイル (PNG インデックスカラー)
 - 出力
 - ◇ テクスチャ画像が更新された建物別 3D モデルデータ (FBX)
 - ◇ ビル屋上設置物 3D モデルデータ (FBX)
 - ◇ 街中小物 3D モデルデータ (FBX)
 - ◇ 道路 3D モデルデータ (FBX)
 - ◇ モデルデータ設置箇所指示データ (CSV)
- 機能詳細
 - 利用するソフトウェア
 - ◇ Houdini
 - 利用するアルゴリズム
 - ◇ Wave Function Collapse
 - ◇ L-system
 - ◇ プロシージャルモデリング
 - ◇ プロシージャルテクスチャリング

8. 【FN008】 ゲームエンジン処理

- 機能概要
 - Houdini から生成された 3D データを配置し、空間表示に必要なシミュレーションモデルを設定する
- フローチャート

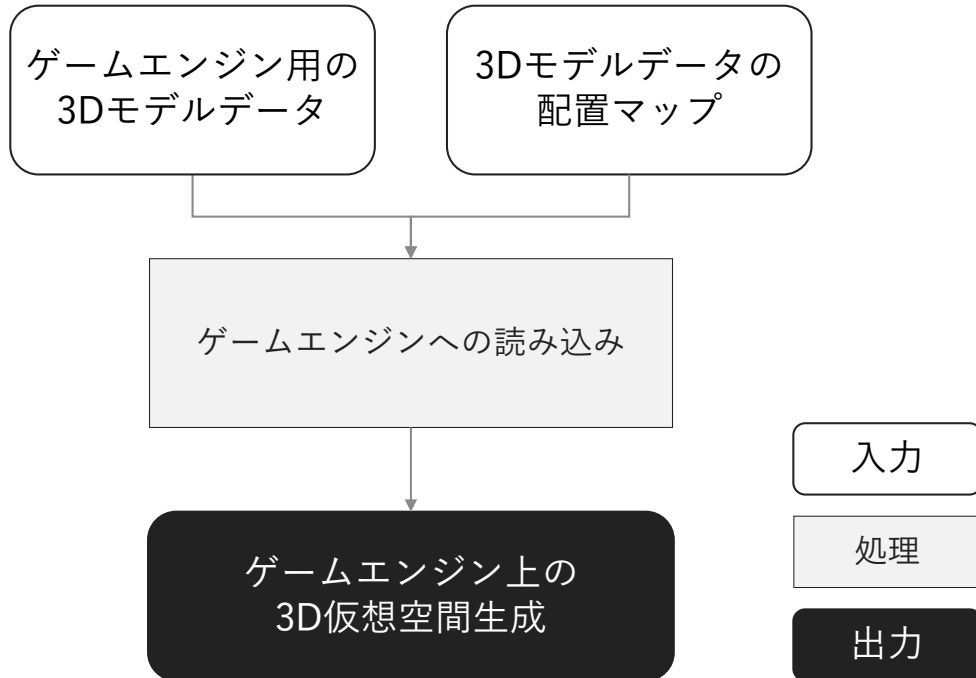


図 3-12 ゲームエンジン処理フローチャート

- データ仕様
 - 入力
 - ◇ Houdini で生成された 3D モデルデータ (FBX)
 - ◇ 3D モデルを配置するためのデータ (CSV)
 - 三次元座標
 - 設置モデル情報
 - 出力
 - ◇ 3D モデルが設置された 3D 仮想空間
 - ◇ 大気シミュレーションモデル
 - ◇ 光源情報
 - ◇ 霧シミュレーションモデル
 - ◇ 雲シミュレーションモデル
 - ◇ レイトレーシング (ガラスの反射情報)
 - ◇ 表示映像フィルター情報
- 機能詳細
 - 利用するライブラリ
 - ◇ Unreal Engine
 - 利用するアルゴリズム

◇ レイキャスティング

9. 【FN009】プロジェクトファイル出力

● 機能概要

- 作成した高精度デジタルツインデータのデータ配布を目的として、Unreal Engine のプロジェクトファイルを出力する

● フローチャート

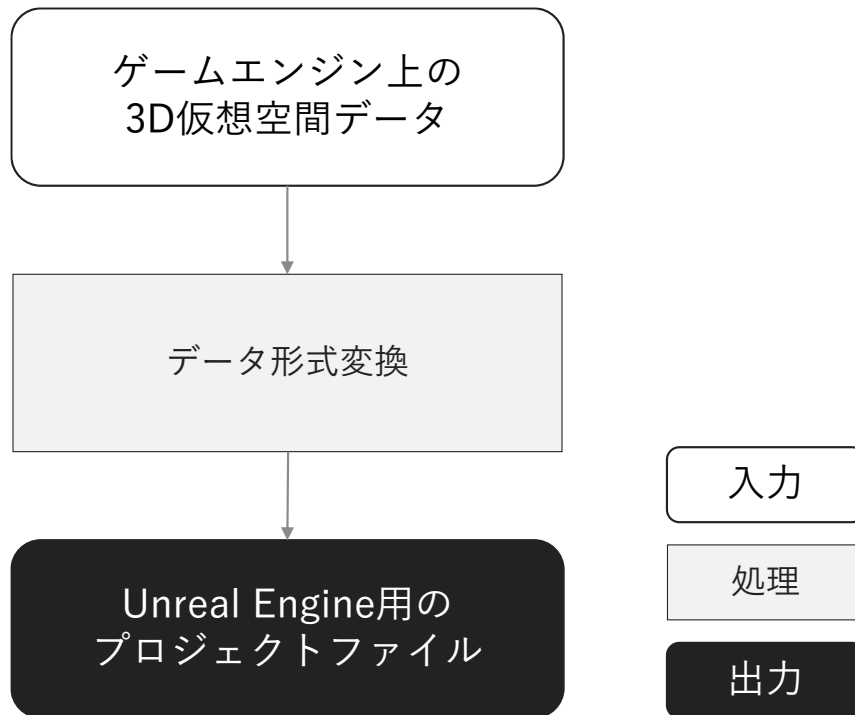


図 3-13 プロジェクトファイル出力フローチャート

● データ仕様

➢ 出力

- ◇ 3D モデル情報やシミュレーションの設定値を記録している、Unreal Engine のプロジェクト (.uproject) ファイル

● 機能詳細

➢ 利用するライブラリ

- ◇ Unreal Engine

10. 【FN010】ゲームプラットフォームへの取り込み

● 機能概要

- 出力したプロジェクトファイルを Unreal Engine for Fortnite のゲームプラットフォームにアップロードするため、フォーマットに変換する。

● データ仕様

➢ 入力

- ◇ プロジェクト.uproject ファイル
- 出力
 - ◇ Unreal Engine for Fortnite 用シミュレーション数値
 - Day Sequence Device (シミュレーションのテンプレートセット)
 - Environment Light Rig Device (光源情報)
 - Lumen Exposure Manager (レイトレーシング情報)
 - ◇ ローポリゴン化することによって軽量化された 3D モデル
- 機能詳細
 - 利用するライブラリ
 - ◇ Unreal Engine
 - ◇ Unreal Engine for Fortnite

3-3. アルゴリズム

3-3-1. 利用したアルゴリズム

表 3-4 利用したアルゴリズム一覧

ID	アルゴリズムを利用した機能	名称	説明	選定理由
AL001	FN002	ViT (Vision Transformer)	<ul style="list-style-type: none"> 画像認識タスクにおいて、自己注意メカニズムを使用して画像内の各パートが他のパートと関連しているかを捉え、特徴間の関連性を把握する 	<ul style="list-style-type: none"> すでに大規模なデータセットで訓練されており、トレーニングデータ数が限られている場合などに高い精度が出しやすくなるため
AL002	FN007	Catmull-Clark Subdivision	<ul style="list-style-type: none"> 設定された側面構造ユニットを敷き詰める 	<ul style="list-style-type: none"> 建物側面は建物ごとに大きさが異なるが、当機能は新規にユニットを、側面に穴が開かないように敷き詰めることができるため
AL003	FN002	NeRF (Neural Radiance Fields)	<ul style="list-style-type: none"> 画像から 3D シーンを再構築することで、屋上の構造物等を生成する 	<ul style="list-style-type: none"> 既存公開モデルの生成精度が向上しており、学習を必要とせず 3D 空間を構築できるため

1) 【AL001】 ViT (Vision Transformer)

● 計算量

- 画像のパッチ数を P 、パッチの次元を D 、Transformer のレイヤー数を L 、Attention のヘッド数を H 、Attention のヘッドの次元を K とすると
 - ◇ パッチのエンベディングの計算量は $O(PD)$
 - ◇ Attention の計算量は、各レイヤーにおいて $O(P^2HK)$ 。 L レイヤー分で $O(LP^2HK)$ 。
 - ◇ MLP の計算量は、各レイヤーにおいて $O(PD^2)$ 。 L レイヤー分で $O(LPD^2)$ 。
 - ◇ クラストークンの計算量は $O(D)$ 。
- 全体を通しての ViT の計算量は、 $O(PD+LP^2HK+LPD^2+D)=O(LP^2HK+LPD^2)$
- 通常 $P \gg D > H, K$ であることから、ViT の計算量は主に $O(LP^2HK)$

● イメージ

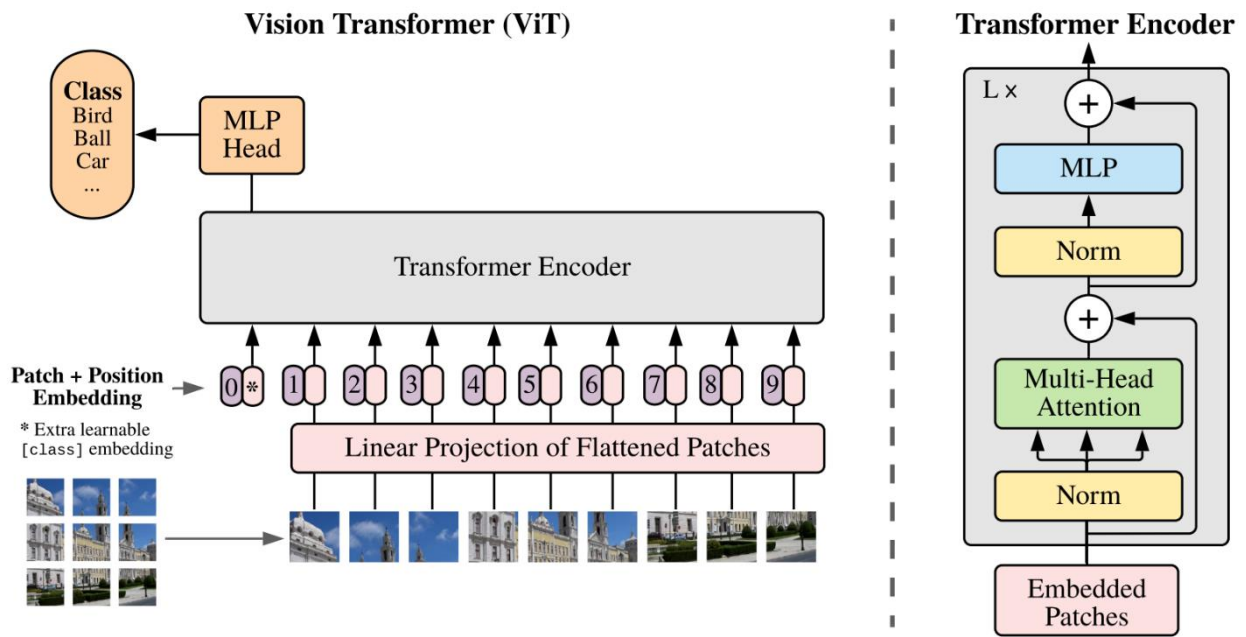


図 3-15 ViT のニューラルネットワークのイメージ²

2) 【AL002】 Catmull-Clark Subdivision

● 計算量

- Catmull-Clark Subdivision の計算量は、メッシュの面数、辺数、頂点数に依存する
- 面数を F、辺数を E、頂点数を V とすると、1 回の Subdivision における計算量は以下ようになる
 - ◇ 各面の中心点の計算量は $O(F)$
 - ◇ 各辺の中点の計算量は $O(E)$
 - ◇ 各頂点の平均点の計算量は $O(V)$
 - ◇ 新しい頂点の位置の計算量は $O(V)$
 - ◇ 新しい面の生成の計算量は $O(F+E)$
- 1 回の Subdivision の全体の計算量は、 $O(F+E+V)$
- Subdivision を繰り返し行う回数を n とすると、n 回の Subdivision の計算量は、Subdivision ごとにメッシュの面数、辺数、頂点数が 4 倍、2 倍、4 倍ずつ増加する
- $O(F+E+V)+O(4F+2E+4V)+O(16F+4E+16V)+\dots+O(4^{(n-1)}F+2^{(n-1)}E+4^{(n-1)}V)=O(4^nF+2^nE+4^nV)$
- Catmull-Clark Subdivision の全体の計算量は $O(4^nF+2^nE+4^nV)$

● イメージ

² 出典: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (<https://arxiv.org/abs/2010.11929>)

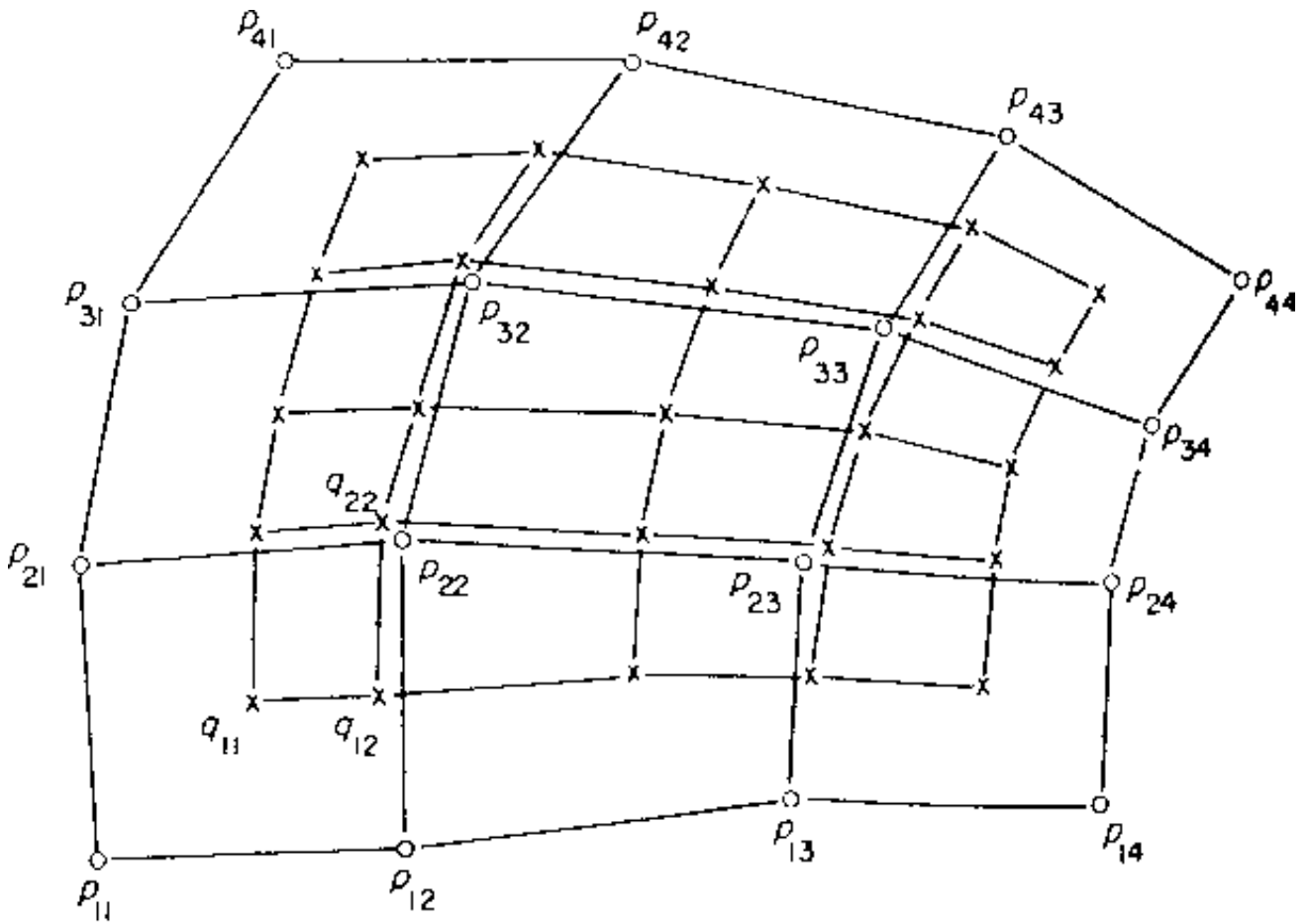


図 3-16 Catmull-Clark Subdivision の面を分割するイメージ³

3) 【AL003】 NeRF (Neural Radiance Fields)

● 計算量

- NeRF の計算量は、シーンを表現するために使用される点の数、ニューラルネットワークの構造、および画像の解像度に依存する
- シーンを表現するために使用される点の数を P 、ニューラルネットワークの層数を L 、各層のニューロン数を N 、画像の解像度を (H,W) とすると
- 各点における色と密度の計算量は $O(LN)$ 。 P 個の点について計算を行うので、全体で $O(PLN)$
- 各画素における色の計算量は、レイ上のサンプル点の数を S とすると $O(S)$ 。画像全体で (H,W) 個の画素があるので、全体で $O(HWS)$
- レイ上のサンプル点の色と密度の計算量は、上記の 2 つを組み合わせると $O(PLNHWS)$
- 勾配の計算と重みの更新の計算量は、ニューラルネットワークのパラメータ数を M とすると $O(M)$
- NeRF の全体の計算量は、 $O(PLNHWS+M)$

³ 出典: Recursively generated B-spline surfaces on arbitrary topological meshes
<https://dl.acm.org/doi/abs/10.1145/280811.280992>

- 通常 $P, L, N, H, W, S \gg M$ であるため、NeRF の計算量は主に $O(PLNHWS)$

- イメージ

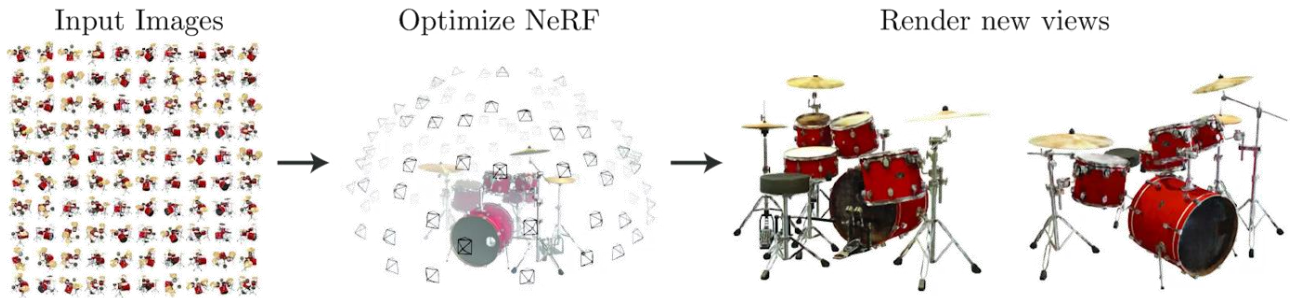


図 3-17 NeRF の写真から 3D モデルを生成するイメージ⁴

3-3-2. 開発したアルゴリズム

1) 【AL101】側面構造化判定アルゴリズム

- 本アルゴリズムを利用した機能
 - 【FN004】機械学習処理（3D 都市モデルデータ）
- アルゴリズムの詳細
 - セマンティックセグメンテーションタスクという画像内において同様の色や形状の繰り返しを一つの領域として予測するアルゴリズムをベースとしており、読み込んだテクスチャの画像をスペースデータ社で独自に構築した教師データと比較することで側面構造物のデザイン・素材とその領域を予測する
 - 上記のアルゴリズムの仕組みを活用し、3D 都市モデルのテクスチャ画像から、特徴量を抽出し、建物側面の構造化情報（窓など）を取得する
 - ◇ テクスチャ画像から屋上部分と側面部分を抽出
 - ◇ 側面部分から窓のユニット（壁・窓を構成する要素の最小構成単位）を分類予測
 - ユニットには四角い窓や全面窓等が複数用意されており、ユニットの色は建物の色情報を反映させる

⁴ 出典: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis (<https://arxiv.org/abs/2003.08934>)

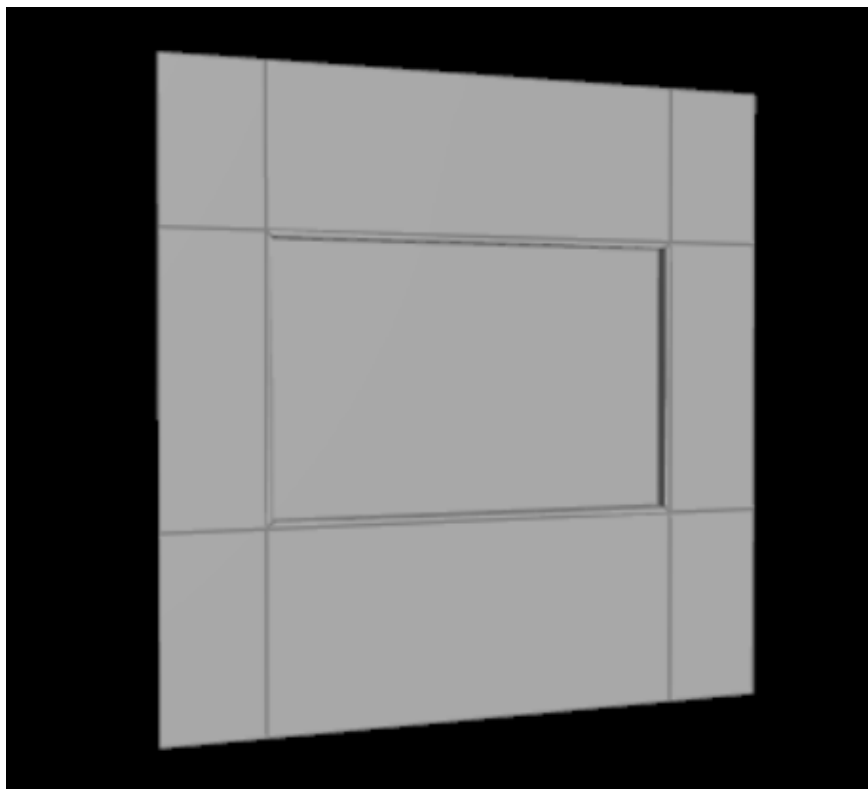


図 3-18 窓のユニット例(中央部分が窓)

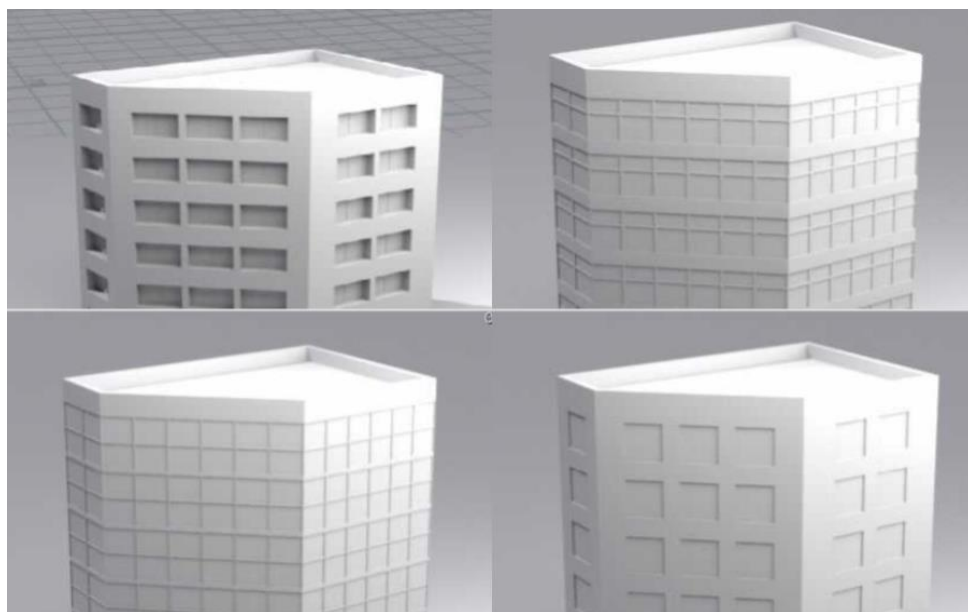
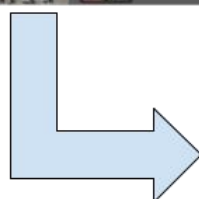
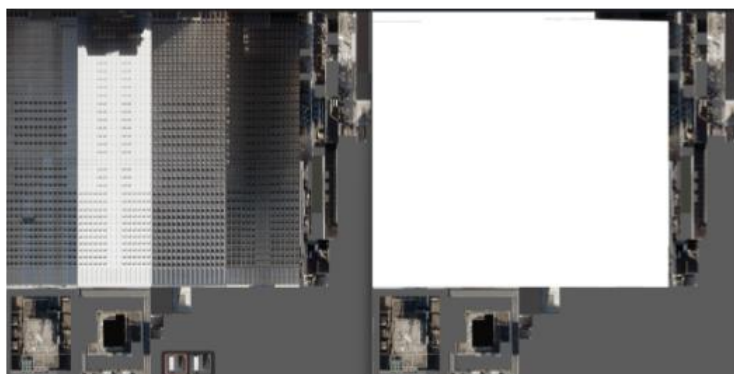


図 3-19 窓のユニットを敷き詰めた場合

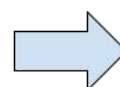
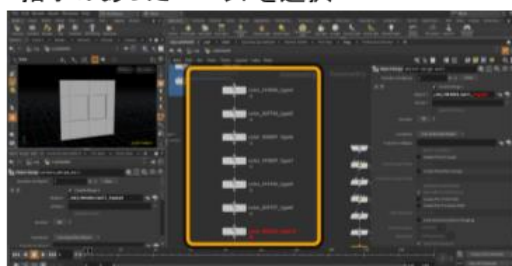
- Houdini 等の 3DCG ソフトウェアで処理が行えるように、位置情報と分類の予測結果を構造情報としてレイヤー化して、再度 3D 都市モデルのテクスチャ画像にデータを追加する
- ✧ Houdini では本アルゴリズムで出力した予測結果を読み取り、その情報に合わせて 3D モデル生成基盤上でテクスチャ画像や 3D モデルを生成する

側面部分判定(窓部分テクスチャ判定:白)



Houdiniへ

指示のあったユニットを選択



ユニットを敷き詰める



図 3-20 アルゴリズム詳細イメージ

3-4. データインタフェース

3-4-1. ファイル入力インタフェース

- 1) 【IF001】 地理情報の入力
 - 本インタフェースを利用した機能
 - 【FN001】
 - データ形式
 - 衛星画像データ
 - ◇ GeoTIFF
 - 地図データ
 - ◇ XML
 - 本インタフェースを利用する機能：【FN006】

- 2) 【IF002】 機械学習情報の入力
 - 本インタフェースを利用した機能
 - 【FN002】
 - データ形式
 - PNG 画像
 - 建物 ID

- 3) 【IF003】 3D 都市モデルデータの入力
 - 本インタフェースを利用した機能
 - 【FN003】
 - データ形式
 - CityGML
 - ◇ 道路 LOD0
 - ◇ 都市設備 LOD2
 - ◇ 都市計画決定情報 LOD0
 - ◇ 建築物 LOD2

- 4) 【IF004】 地理情報データベースへ 3D 都市モデルの建物属性を入力し反映させる
 - 本インタフェースを利用した機能
 - 【FN005】
 - データ形式
 - CityGML
 - ◇ 建築物 LOD0
 - bldg:lod0RoofEdge

◇ 建築物 LOD1

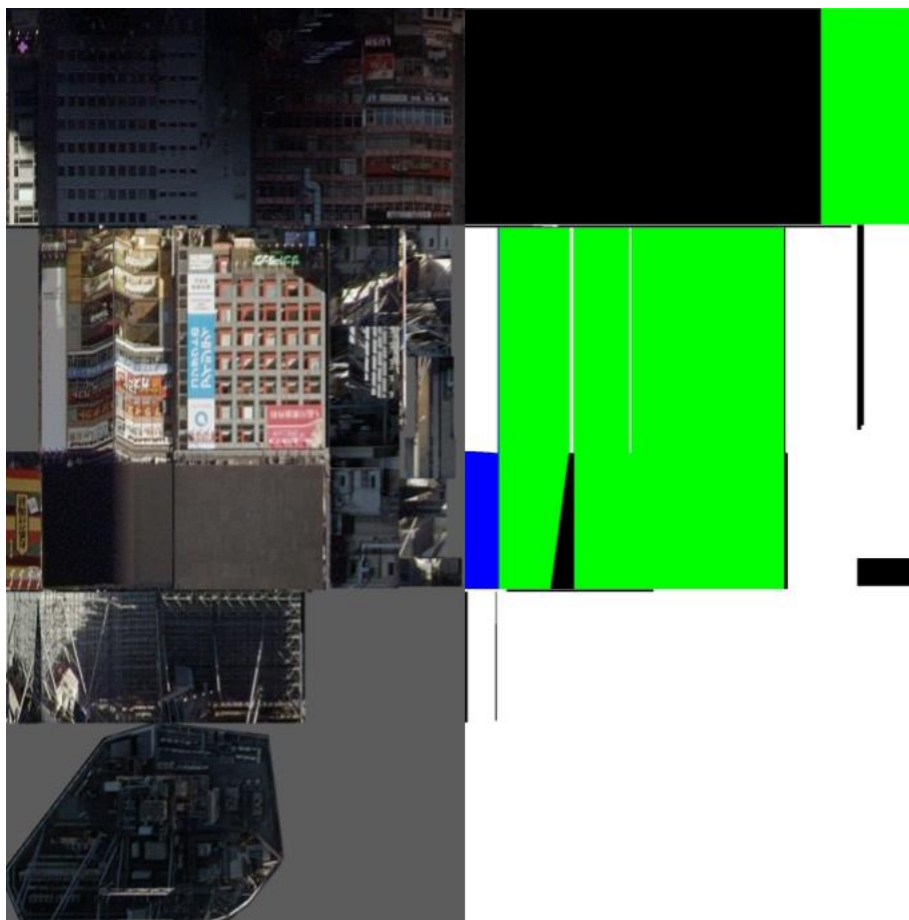
3-4-2. ファイル出力インターフェース

1) 【IF101】 PLATEAU データの処理ファイル

- 本インターフェースを利用した機能
 - 【FN003】
- データ形式
 - FBX
 - ◇ テクスチャ画像
 - ◇ 3D 都市モデルを参照するカスタムパラメータ
 - フットプリント緯度経度ポリゴン
 - 高さ
 - 建物名
 - 住所
 - 地域地区
 - 土地利用
 - ◇

2) 【IF102】 Houdini へのファイル

- 本インターフェースを利用した機能
 - 【FN006】
- ファイル形式
 - FBX
 - ◇ 側面構造を反映したテクスチャ画像
 - 3D モデルに対して、どの面の部分が、広告・窓・壁なのかの情報を PNG インデックスカラーで保持
 - ◇ 窓部分の場合には、どのユニットを敷き詰めるのかの情報も含む



● 図 3-21 左: PLATEAU のテクスチャ画像、右: テクスチャの側面構造データ例

➤ 建物 XML

表 3-5 Houdini へのデータ受け渡し建物 XML ファイルサンプル

ID	Polygon	FbxId	Unit Id	Height	RoofColor	RoofShape	BuildingColor	Tenant	Sign
136397979	POLYGON (((15551679.290075667 4253989.810761496,...)))	13113-bldg-10020	2	33.4	#A9A9A4	1	#530000	{convenience:1}	{}
155807205	POLYGON (((15551040.349594362 4253642.781569757,...)))	13113-bldg-10234	21	56.5	#BAB5B5	3	#ffedd3	{}	{entrance:true}

※Tenant : ビルの1階部分にあるコンビニ等のテナントの有無を示すデータ

※Sign : 建物に対してどこに広告が設置されているかということを示すデータ

➤ 道路 XML

表 3-6 Houdini へのデータ受け渡し道路 XML ファイルサンプル

ID	Type	Bridge	Polygon	Width
990655378	2	true	LINestring (15561178.850142002 4246585.538424026,...)	12
1123647488	5	false	LINestring (15559901.837471416 4247720.539426402,	20

3) 【IF103】 3D モデルデータ&配置マップ出力

- 本インターフェースを利用した機能
 - 【FN007】
- データ形式
 - 3D モデルデータ
 - ◇ FBX(テクスチャ画像を含む)
 - 配置マップ
 - ◇ CSV

表 3-7 配置マップ CSV 出力サンプル

Assets	Position	Scale	Color Data	Rotation
air_conditioner_unit	(792.8748, 0.0000, 1213.3579)	(1.0000, 1.0000, 1.0000)	(1.0000, 1.0000, 1.0000)	(0.0000, 183.9645, 0.0000)
air_conditioner_unit	(752.0370, 0.0000, 1287.4985)	(1.0000, 1.0000, 1.0000)	(1.0000, 1.0000, 1.0000)	(0.0000, 3.3889, 0.0000)

4) 【IF104】 Unreal Engine 5 プロジェクトファイル等バイナリデータ

- 本インターフェースを利用した機能
 - 【FN009】
- データ形式
 - Unreal Engine プロジェクトファイル

3-4-3. 内部連携インターフェース

1) 【IF201】地図情報のデータ処理後

- 本インターフェースを利用した機能
 - 【FN001】
- データ形式
 - ◇ 建物の情報
 - 建物 ID
 - 識別用のユニークな ID
 - 側面構造
 - 窓の分類
 - ◇ ユニットのインデックス番号
 - 建物色
 - 屋上情報
 - 屋根形式
 - ◇ インデックス番号
 - 色
 - 建物高さ
 - 設置面(緯度経度ポリゴン)
 - 設置面で切り取った光学衛星データの画像
 - 広告の情報
 - 屋上広告の有無
 - 側面広告の有無
 - 入り口広告の有無
 - ◇ テナント情報
 - 建物 1 階部分に入っているコンビニ等テナント情報
 - テナントインデックス番号
 - テナント詳細インデックス番号
 - ◇ 土地境界 (緯度経度ポリゴン)
 - ◇ 周辺人流スコア
 - 日常的にどれだけの人がある建物の周辺や建物内にいるかという統計データを
 - ◇ 道路の情報
 - 形状 (緯度経度ポリゴン)
 - 幅員 (レーン数)
 - 歩道の有無
 - ◇ 境界情報で切り取った光学衛星データの画像

表 3-8 地図情報のデータ処理後サンプル

データ項目	値
建物 ID	116596178
設置面	Polygon(...) 緯度経度
土地境界	Polygon(...) 緯度経度
建物色	#FDDDCD
高さ	30.2
建物敷地面積	930.0
テナント情報	None
側面構造	{window: 2}
周辺人流量スコア	3
屋根色	#AAAAAA
屋根形式 Index	6
看板入口	False
看板側面	True
看板屋上	False
建物名	新宿郵便局
住所	東京都新宿区西新宿一丁目
地域地区	高度利用地区
土地利用	商業用地
道路形状	Line(...) 緯度経度
道路幅員	16
道路の歩道有無	True
衛星画像	画像データ(PNG)のバイナリ形式

2) 【IF202】機械学習での衛星データから予測した建物属性

- 本インターフェースを利用した機能
 - 【FN002】
- データ形式
 - 建物属性値
 - ◇ 屋根色
 - ◇ 屋根形状

3) 【IF203】地理情報データベースと 3D 都市モデルの建築物モデルを一致させるための緯度経度情報

- 本インターフェースを利用した機能

- 【FN005】
 - データ形式
 - 建物位置情報
 - ◇ 緯度経度ポリゴン
- 4) 【IF204】 3D 都市モデルの建物属性データから抽出したデータ
- 本インターフェースを利用した機能
 - 【FN005】
 - データ形式
 - 建物属性値
 - ◇ 高さ
- 5) 【IF205】 3D モデル生成用地理データ情報
- 本インターフェースを利用した機能
 - 【FN006】
 - データ形式
 - FBX (テクスチャ画像を含む) : 【IF102】
 - データベース
 - ◇ 道路情報
 - 緯度経度ポリゴン
 - 高架線
 - 道路タイプのインデックス番号(歩道が存在するか等)
 - ◇ 建物情報
 - フットプリント緯度経度ポリゴン
 - FBX ID
 - 高さ
 - 屋根色
 - 屋根形状
 - 側面色
 - 側面形状
 - ユニット ID
 - ◇ FBX 側のテクスチャにユニット情報は載っているが、PLATAU 未対応の地区であればこのユニット ID を全体に利用する

3-4-4. 外部連携インターフェース

本プロジェクトにおいて、外部とのデータ連携は実施しない。

3-5. 実証に用いたデータ

3-5-1. 活用したデータ一覧

1) 利用した 3D 都市モデル

- 年度：2022 年度
- 都市名：東京 23 区
- ファイル名：53393585_bldg_6697_2_op.gml、53393586_bldg_6697_2_op.gml、53393595_bldg_6697_2_op.gml、53393596_bldg_6697_2_op.gml、53394525_bldg_6697_2_op.gml、53394526_bldg_6697_2_op.gml、53394536_bldg_6697_2_op.gml、53394536_bldg_6697_2_op.gml
- メッシュ番号：53393585、53393586、53393595、53393596、53394525、53394526、53394536、53394536

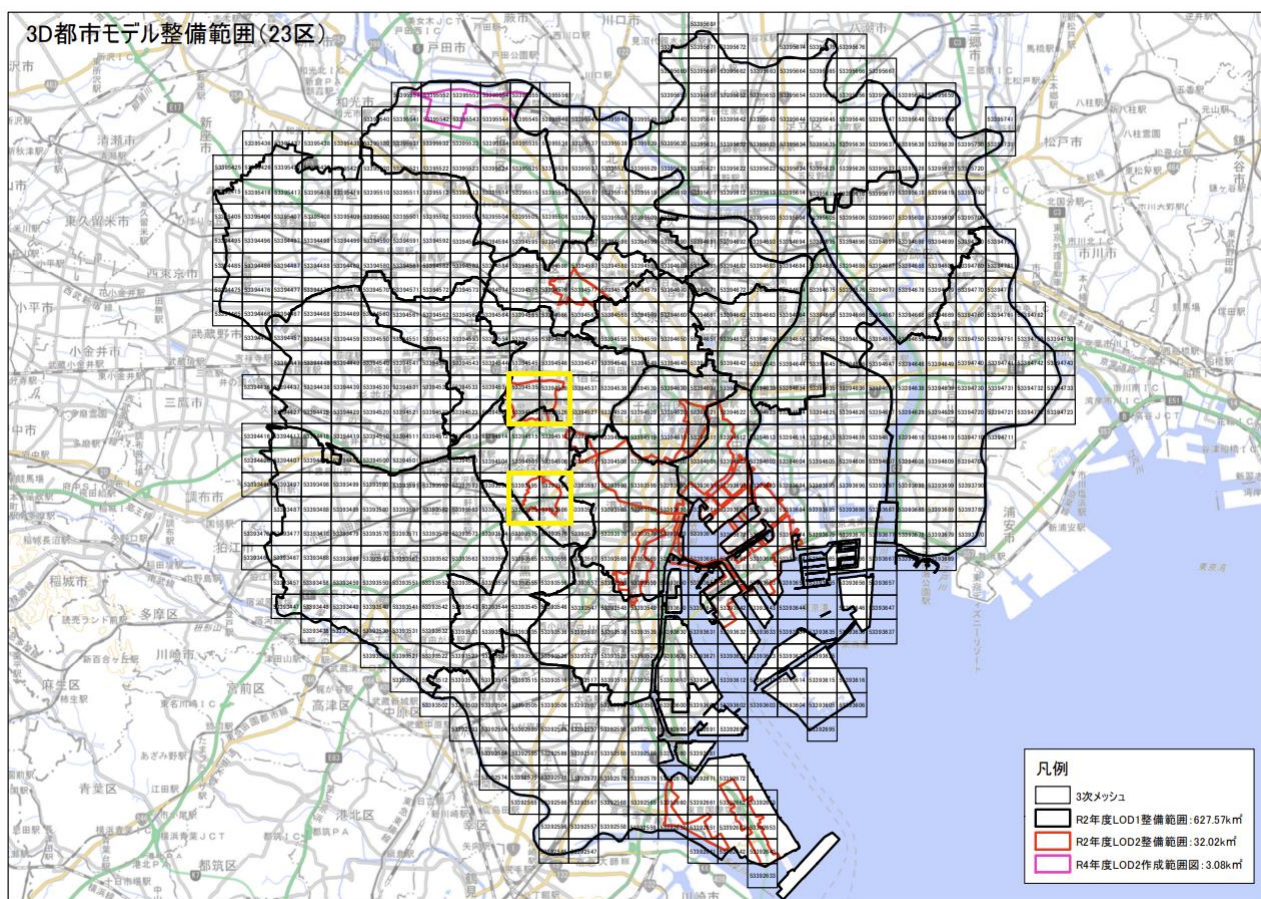


図 3-22 インデックスマップ（東京都）

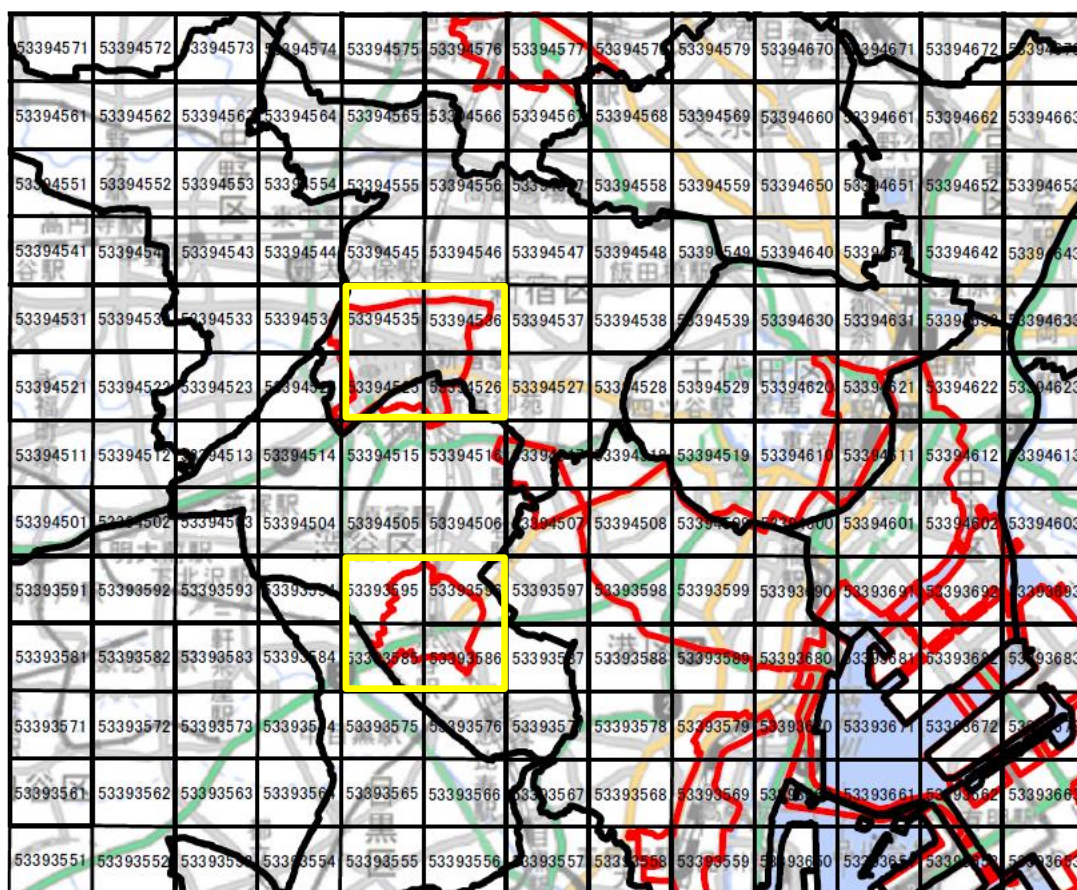


図 3-23 インデックスマップ (対象エリア拡大版)

表 3-9 利用した 3D 都市モデル

地物	地物型	属性区分	ID	属性名	内容	データを利用した機能 (ID)
建築物	bldg:Buiding	主題属性	DT001	bldg:class	形態による区分	FN004
			DT002	bldg:usage	主な使い道	FN004
			DT003	bldg:roofType	屋根形状の種類	FN004
			DT004	bldg:measuredHeight	地上の最低点から最高点までの高さ	FN005
			DT005	bldg:storeysAboveGround	地上階の階数	FN005
			DT006	bldg:storeyHeightsAboveGround	各階の高さ	FN005
			DT007	bldg:lod0FootPrint	接地面での外形線により囲まれた面	FN005
			DT008	bldg:lod0RoofEdge	屋根の外形線に	FN005

					より囲まれた面	
			DT009	bldg:lod1Solid	形状を示す立体	FN005
			DT010	bldg:lod2Solid	主要構造の外形を示す立体	FN005
			DT011	bldg:lod3Solid	主要構造の外形を示す立体	FN004
			DT012	bldg:outerBuildingInstallation	付属する設備	FN004
			DT013	bldg:boundedBy	外壁等の境界面	FN005
			DT014	bldg:RoofSurface	上部を覆う構造物	FN004
			DT015	bldg:WallSurface	外周を構成する壁面	FN005
			DT016	bldg:GroundSurface	立体形状の底面	FN005
			DT017	bldg:OuterCeilingSurface	外側を覆う部分	FN005
			DT018	bldg:OuterFloorSurface	屋根面のうち、通行可能な面	FN004
			DT019	bldg:ClosureSurface	仮想的な面	FN005
			DT020	bldg:Door	ドア面	FN004
			DT021	bldg:Window	窓面	FN004
			DT022	bldg:address	住所	FN005
			DT023	uro:buildingIDAttribute	識別情報	FN005
			DT024	uro:buildingDetailAttribute	基礎的な情報	FN005
			DT025	uro:largeCustomerFacilityAttribute	大規模集客施設に関する情報	FN005
			DT026	app:appearance	建物のテクスチャ画像	FN004
土地利用	luse:LandUse	主題属性	DT027	luse:class	利用用途の区分	FN005
			DT028	luse:usage	利用用途	FN005
			DT029	luse:lod1MultiSurface	土地利用の範囲	FN005
			DT030	uro:landUseDetailAttribute	詳細情報	FN005
道路	tran:Road	主題属性	DT031	tran:class	分類	FN005
			DT032	tran:function	道路区分	FN005
			DT033	tran:usage	利用方法	FN005
			DT034	tran:trafficArea	通行可能な領域	FN005
			DT035	tran:lod0Network	連続性を表現する線	FN005
			DT036	tran:lod1MultiSurface	道路の範囲	FN005

DT23-04_技術検証レポート_3D 都市モデルを活用した高精度デジタルツインの構築

都市設備	frn:CityFurniture	主題属性	DT037	frn:class	区分	FN005
			DT038	frn:function	種類	FN005
			DT039	frn:usage	用途	FN005
			DT040	frn:lod1Geometry	専有範囲の立体	FN005
			DT041	uro:cityFurnitureDetailAttribute	詳細	FN005
都市計画 区域	urf:UrbanPlanningArea	主題属性	DT042	urf:function	種類	FN004
			DT043	urf:usage	用途	FN004
			DT044	urf:areaClassification	区域区分	FN004
			DT045	urf:lod1MultiSurface	準都市計画区域の範囲	FN004

3-5-2. 生成・変換したデータ

表 3-10 生成・変換したデータ

ID	システムに入力するデータ (データ形式)	用途	処理内容	データ処理ソフトウェア	活用データ (データ形式)	データを利用した機能 (ID)
DT201	3D 都市モデル (CityGML 形式)	個別の建物情報に切り分ける	3D 都市モデル (CityGML) から、FBX 形式とテクスチャ画像を生成	Python	建物 3D ファイル (OBJ) とテクスチャ画像 (JPEG)	FN003

3-6. ユーザーインターフェース

3-6-1. 画面一覧

1) 画面

表 3-11 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	-	ログイン画面	スペースデータが、ディベロッパー向けにデータを提供するサービスのログイン画面	FN009
SC002	-	ダウンロード画面	スペースデータが、ディベロッパー向けにデータを提供するサービスのダウンロード画面	FN009

3-6-2. 画面遷移図

1) 画面

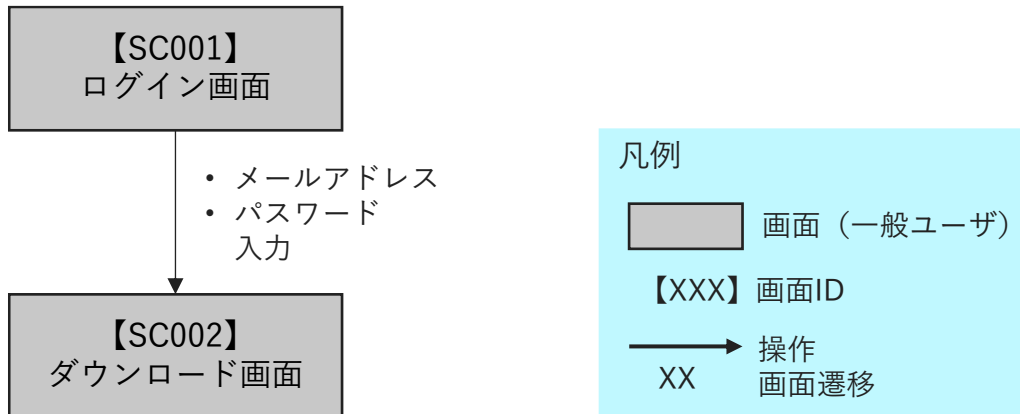


図 3-24 画面遷移図

- 本ユーザーインターフェースは、スペースデータがディベロッパーに提供するデータダウンロードの画面仕様を記載
- ディベロッパーが実際に使用する画面は Fortnite 側の画面となるため省略、下記リンクを参照
 - <https://dev.epicgames.com/documentation/ja-jp/uefn/user-interface-reference-for-unreal-editor-for-fortnite>

3-6-3. 各画面仕様詳細

1) 画面説明

1. 【SC001】ログイン画面

- 画面の目的・概要

- サービス利用者は登録された Email address と Password 情報を入力することで、ダウンロード画面へ遷移することができる

- メイン画面の UI 説明

表 3-12 メイン画面で表示するウィンドウ

項目名	説明
Email address	登録された Email address を入力するフォーム
Password	登録された Password を入力するフォーム

- 画面イメージ

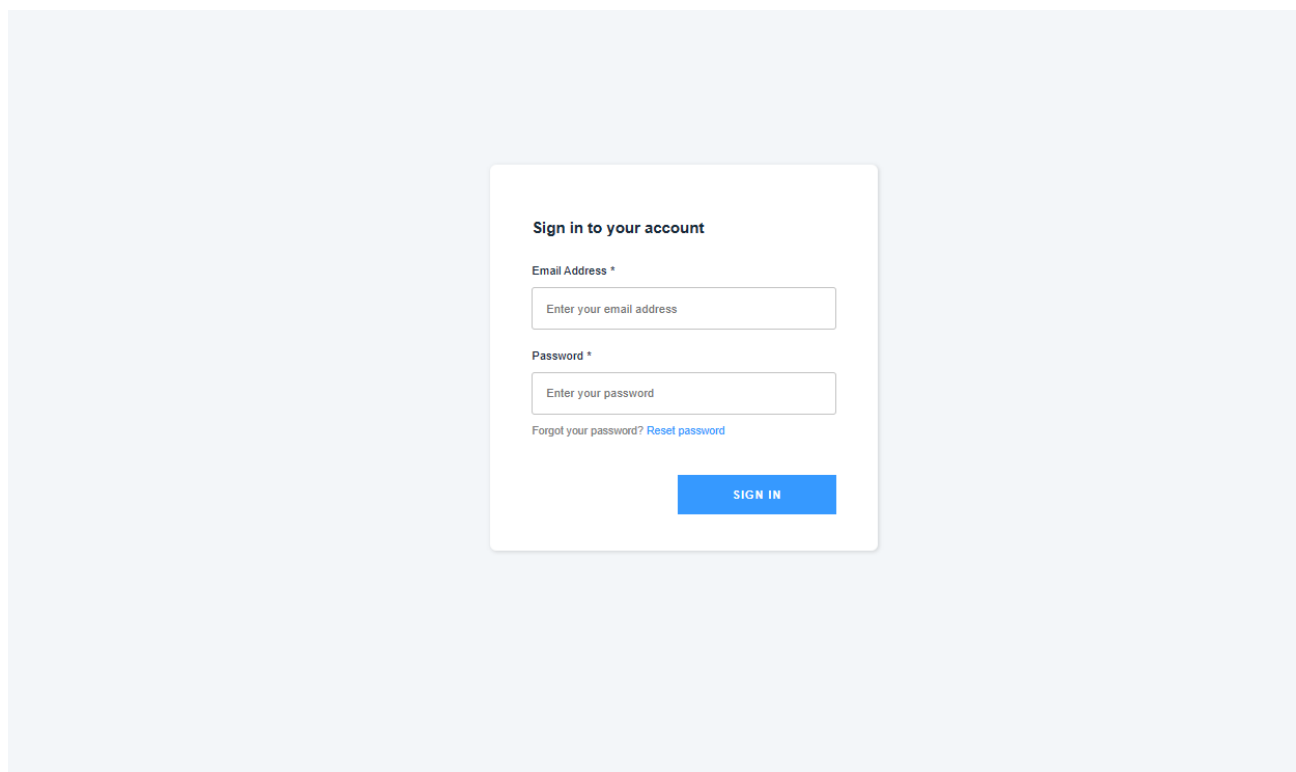


図 3-25 ログイン画面のイメージ

2. 【SC002】ダウンロード画面

- 画面の目的・概要
 - 地図上の範囲を選択することで、その範囲の Unreal Engine 又は Unreal Engine for Fortnite のプロジェクトファイルデータをダウンロードすることができる
 - Unreal Engine へのデータインポート手順のドキュメントサイトへ遷移することができる
- メイン画面の UI 説明

表 3-13 メイン画面で表示するウィンドウ

項目名	説明
Places	登録されたエリアを一覧することができる
Document	Unreal Engine へのデータインポート手順のドキュメントサイトへ遷移することができる
選択可能エリア (画面上の青で囲まれた部分)	選択可能エリアをクリックすることで、Unreal Engine 又は Unreal Engine for Fortnite のプロジェクトファイルデータをダウンロードすることができる

● 画面イメージ

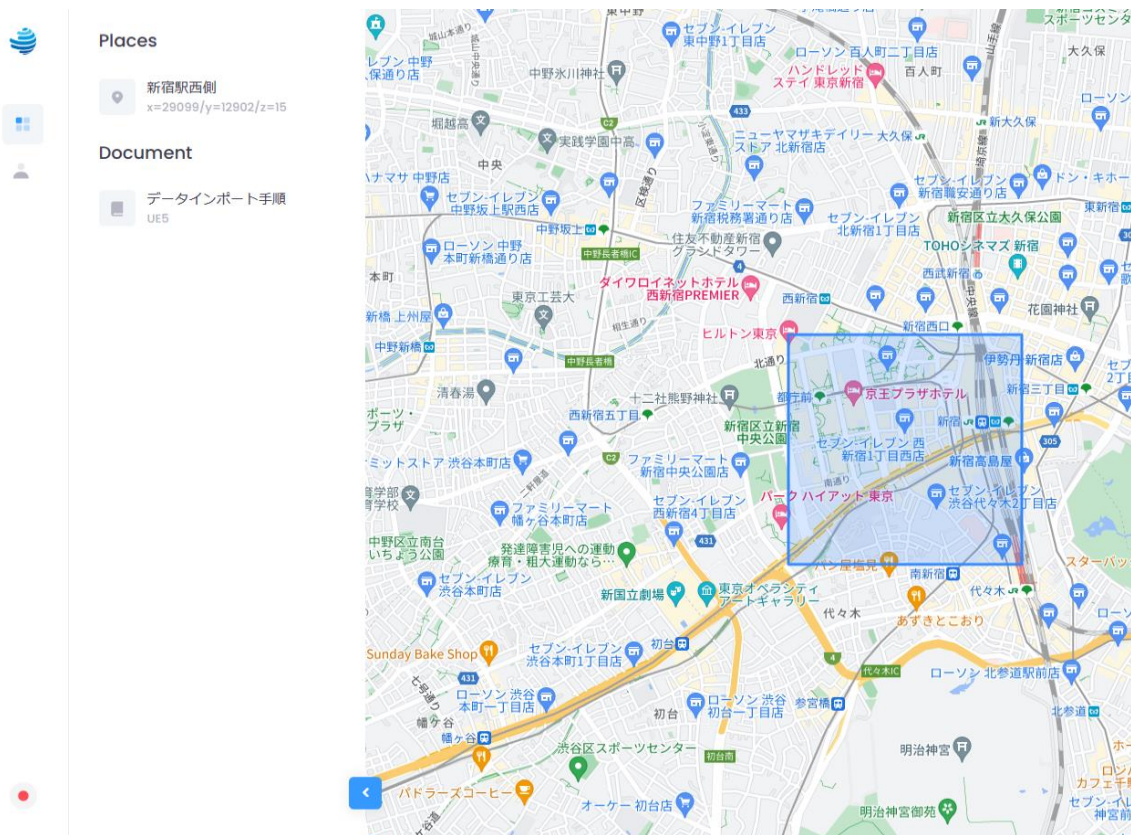


図 3-26 ダウンロード画面のイメージ

3-7. 実証システムの利用手順

3-7-1. 実証システムの利用フロー

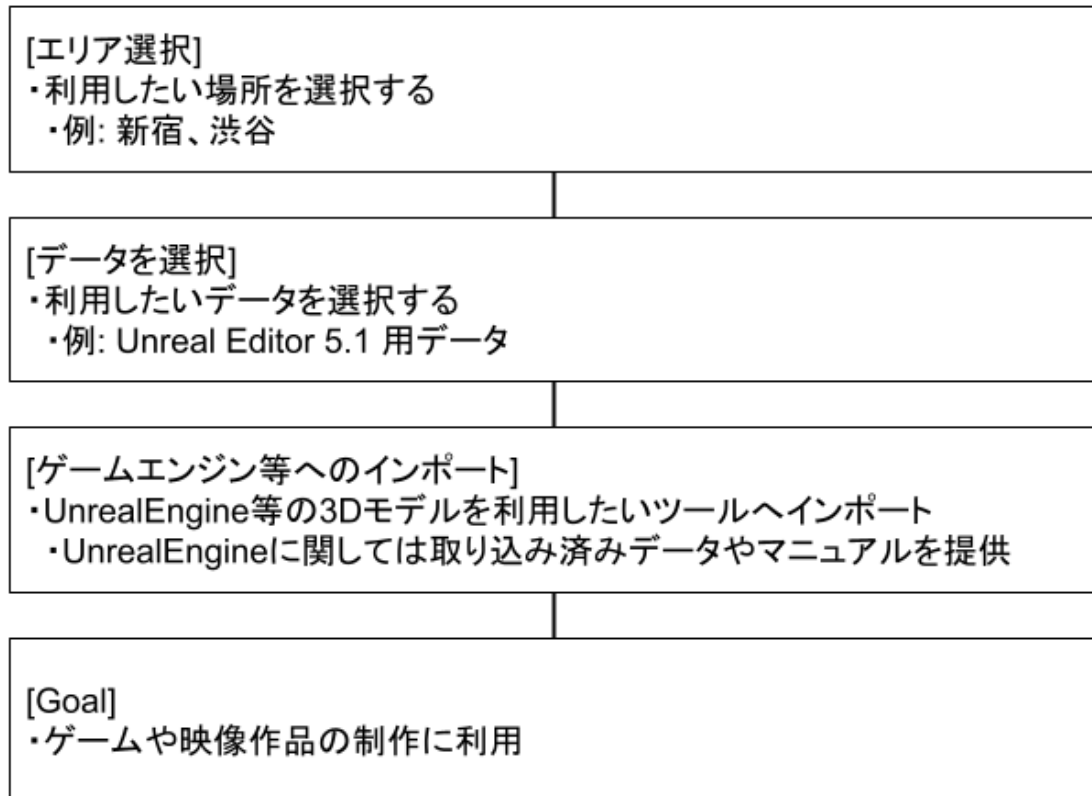


図 3-27 システムの利用フロー

- ダウンロードサイトで取得したいエリアとデータ形式を選択しデータを取得
- その後、ダウンロードしたデータを利用したいゲームエンジン等にインポートをし、作品に利用する。

3-7-2. 各画面操作方法

1) エリア選択

データが提供されているエリアを選択する。



図 3-28 エリア選択

2) データを選択

エリア選択後に必要なデータ形式を選択してダウンロードを行う。

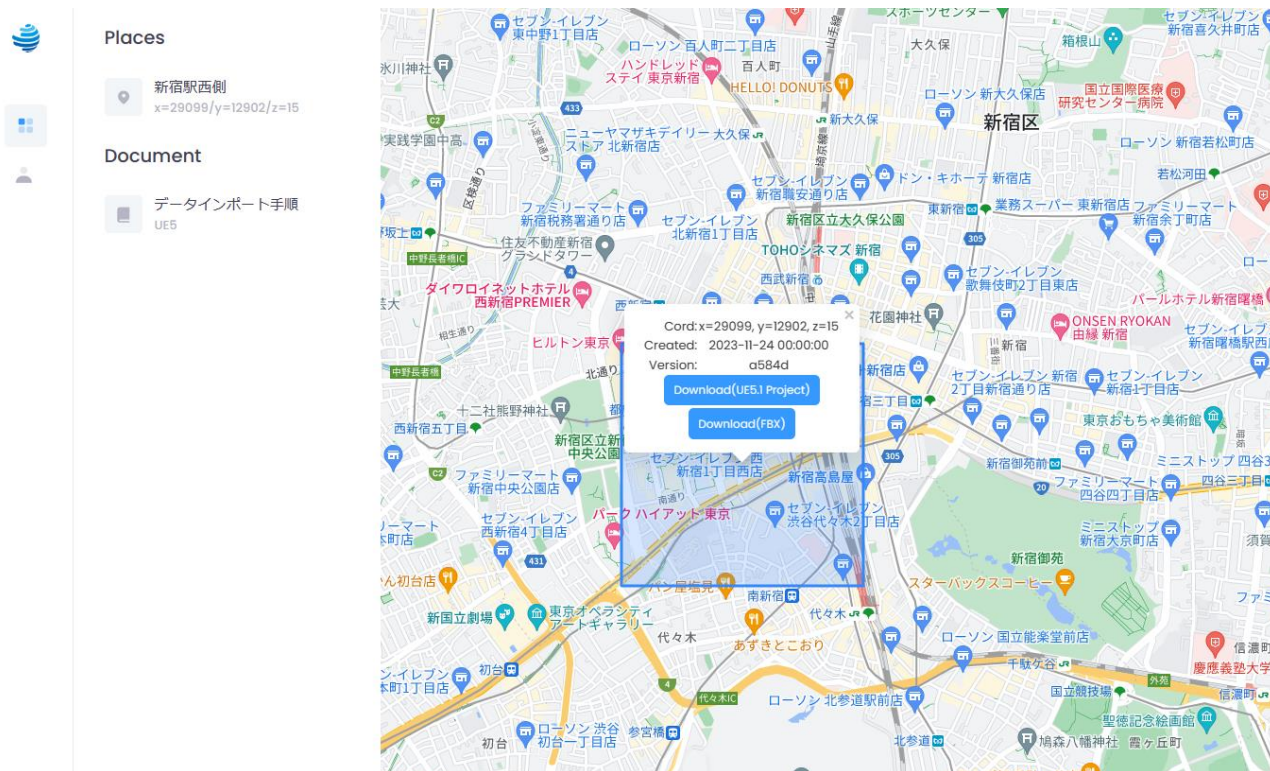


図 3-29 データ形式を選択

3) ゲームエンジン等へのインポート

ダウンロードしたデータを Unreal Engine 等のゲームエンジンや 3D ツールへインポートを行う。



図 3-30 ゲームエンジン等へのインポート

4. 実証技術の検証

4-1. シミュレーション精度の検証

3D 都市モデルを活用した高精度デジタルツインデータの再生成に必要な建物構造判定アルゴリズムの機能検証を行った。

4-1-1. 検証目的

- 3D 都市モデルを活用し、より簡易且つ効率的に高精度デジタルツインデータを生成するため、建物判定アルゴリズムによって構築したデータの外見上の差異と、モデリングに必要な工数を測定することで有用性を確認する。

4-1-2. KPI

表 4-1KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	正答率	80%	● マルチクラス分類における正解率で、データにノイズが含まれており、個別建物テクスチャの解像度も粗いため 8 割程度を想定するため	分類精度（建物側面構造、建物の色）の正答率比較
2	作業時間短縮率	90%	● 今まで手作業であった建物構造判断作業を自動化するため、既存作業時間の 10 分の 1 程度への短縮を見込むため	既存の建物構造情報追加作業時間との比較

4-1-3. 検証方法と検証シナリオ

1) 分類結果の精度

AI 判読結果において、既存手法（a-1）と比較し、2 つの観点で分類結果が正しいのかを正答率を用いて評価する。

- 建物側面構造
- 建物の色

表 4-2 検証シナリオ一覧（分類結果の精度）

No.	検証方法	エリア	検証シナリオ
a-1	既存手法での精度	新宿区	分類精度の正答率比較
a-2	閾値分類結果の精度	新宿区	<ul style="list-style-type: none"> ● 機械学習モデルの学習で利用していない新宿区の建物データの中からランダムに抽出した 100 件について手動でアノテーションを行った正解データに対して、どれだけ正しく予測できたかを正答率とする

2) 既存の建物構造情報追加作業時間との比較

高精度な都市デジタルツインデータを生成する際に、建物構造情報追加プロセスを手作業と比較してどれだけ作業時間を短縮できたかで判断する。

表 4-3 検証シナリオ一覧（既存の建物構造情報追加作業時間との比較）

No.	検証方法	エリア	検証シナリオ
b-1	作業時間の比較	新宿区	既存の建物構造情報追加作業時間との比較 <ul style="list-style-type: none"> ● 既存の手法で新規の建物 1 棟の建物構造情報を追加する際に要した時間と、3D 都市モデルの建築物モデルとテクスチャ画像を機械学習モデルの予測も用いて作業効率化した際の作業時間を比較する

4-1-4. 検証結果

3D 都市モデルを活用することで、教師データとなる側面構造や色判断に関わるデータ量が増加したため、既存の衛星データのための判定率よりも大幅に正答率が向上することが確認できた。

作業工数という観点では 90%の作業効率化という目標値の達成に一步及ばなかったが、既存手法による建物データ生成作業を 80%以上削減することが出来たため、3D 都市モデル活用の有用性を十分に証明できる結果となったと言える。

表 4-4 検証結果サマリ

黄セル：KPI 達成

青セル：KPI 未達

検証内容	評価指標・KPI	目標値	結果		示唆
			項目	評価値 (既存手法での評価値)	
正答率	正答率	80%以上	側面構造	82% (42%)	<ul style="list-style-type: none"> 側面構造については、衛星データでは情報が限られているため既存の手法に比べて顕著に精度が向上した。 側面の色分類については、もともと衛星データからも予測することはできていたが、より精度が向上した。
			側面色分類	94% (78%)	
作業時間短縮	作業平均時間	90%以上	建物データ生成作業	87%	<ul style="list-style-type: none"> 既存手法の作業平均が 15 分 10 秒程度で PLATEAU の 3D のテクスチャ画像による機械学習予測にすることで、作業時間が平均 2 分程度まで短縮を行うことができた。目標値に達することはできなかったが作業効率は大幅に短縮された。

4-1-5. 渋谷のデジタルツインデータ生成

4-1-5-a. 検証目的

- 過去のデジタルツイン生成の事例が少なく、象徴的なランドマークや看板及び構造が複雑な建築物が多い渋谷エリアにおいて、今回の実証実験で開発したアルゴリズム及びシステムを用いた高精度デジタルツインデータの生成を行うことで、本システムの有用性と今後に向けた課題を抽出する。

4-1-5-b. 検証の様子

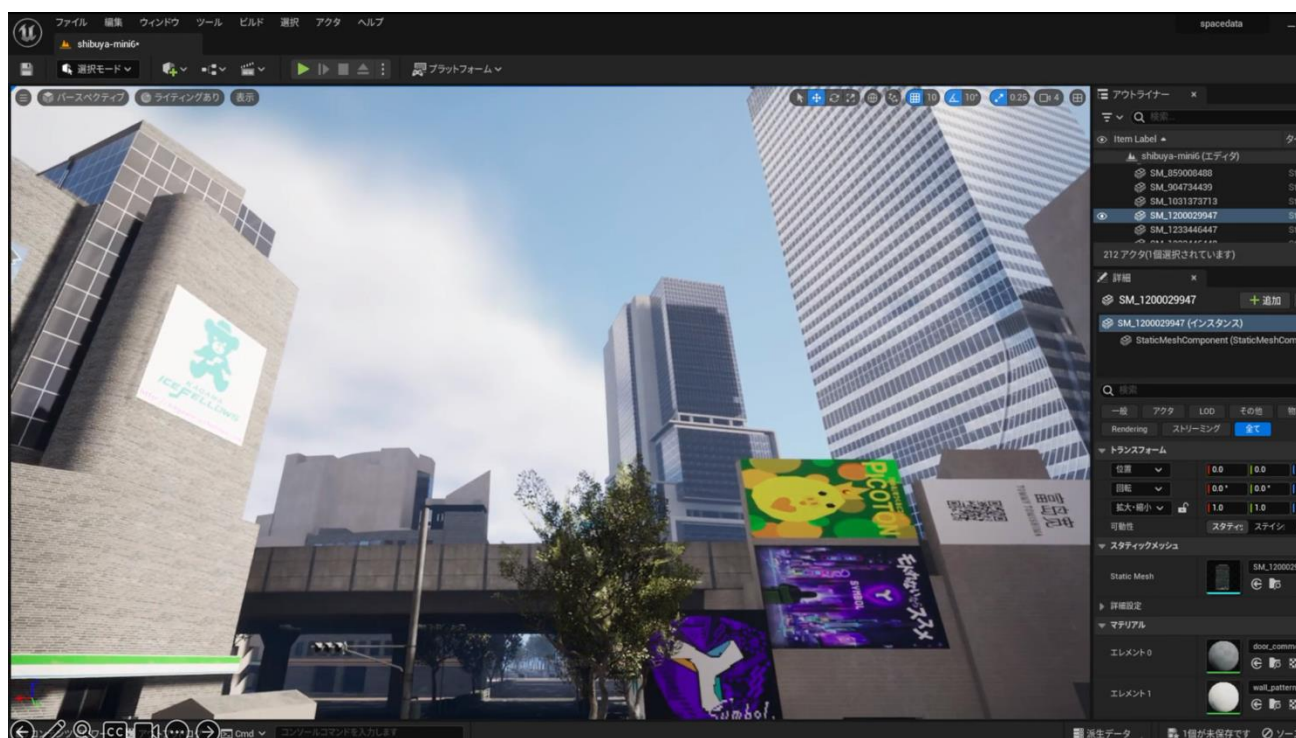


図 4-1 スクランプル交差点付近の渋谷

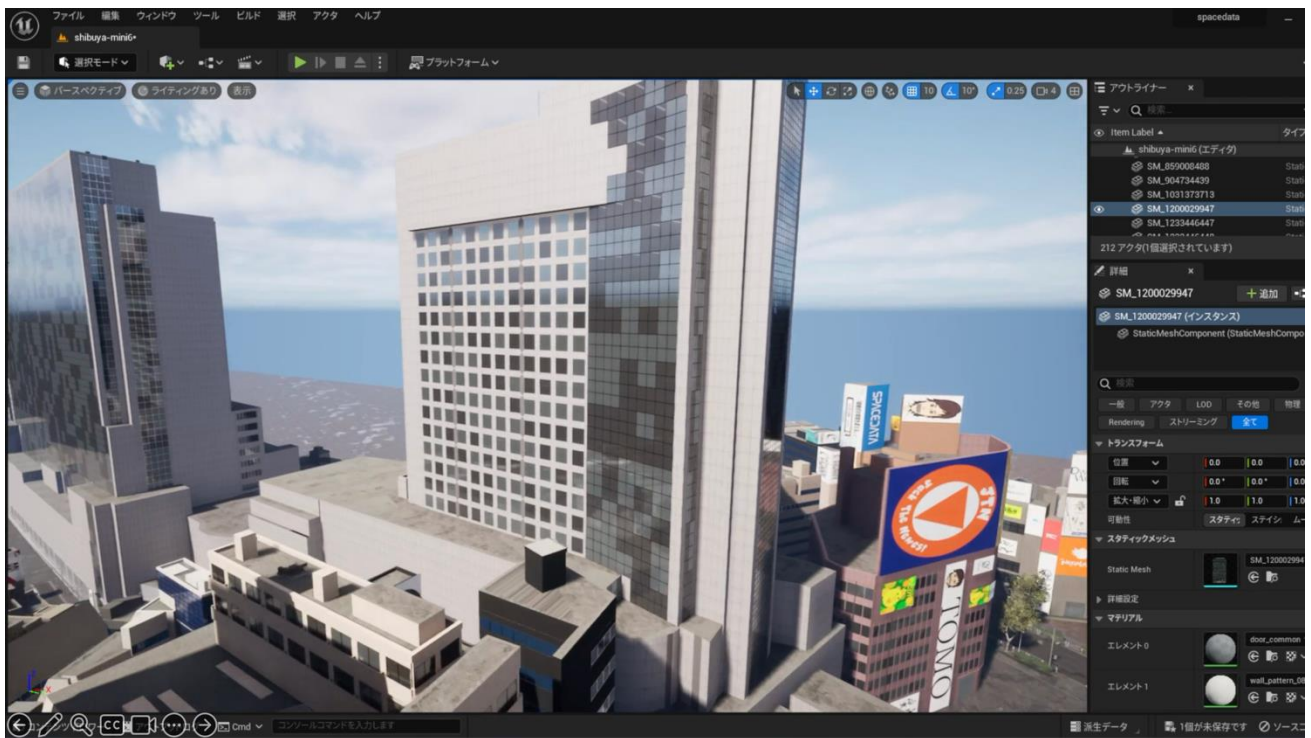


図 4-2 上空からの渋谷

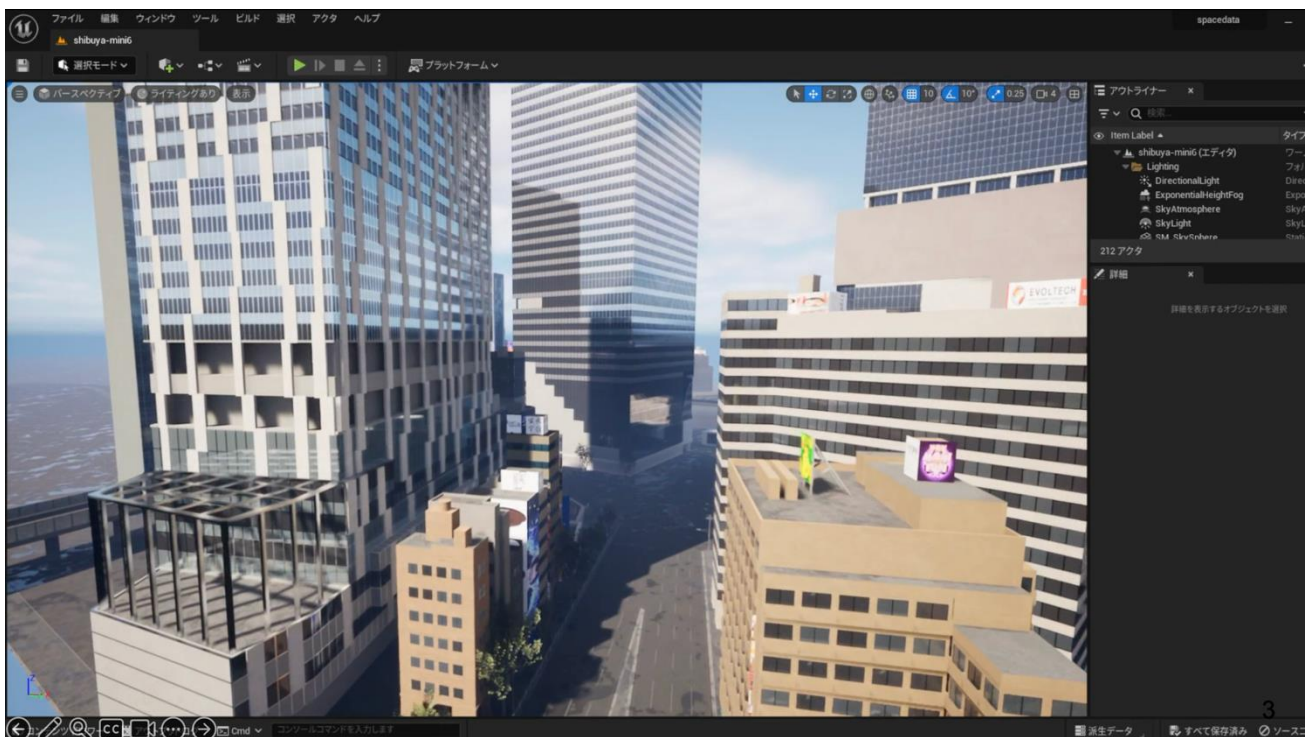


図 4-3 ストリームホール、ヒカリエ側の渋谷

4-1-5-c. 検証結果

渋谷エリアにおいても、新宿エリアと同様のランドマークやシンプルな構造の建築物については、高精度なデジタルツインの生成をすることができたが、複雑な構造である建物デザインの生成、象徴的な看板や、特殊な構造物の生成においては、課題と対応策の方向性を確認することができた。

複雑な構造を持つ建物デザインの生成において、側面のコンクリートや窓などの箇所をシンプルに自動判定する処理だけでは、再現度を高めていくことが難しかった。しかし、複数の窓のデザインパターンがある建物が対象の場合においても、一つの建物が複数の構造を持つように分類数を増加させることで、複数の異なる窓のデザインが混在したような建物の壁面デザインに対しても対応できるように改善した。

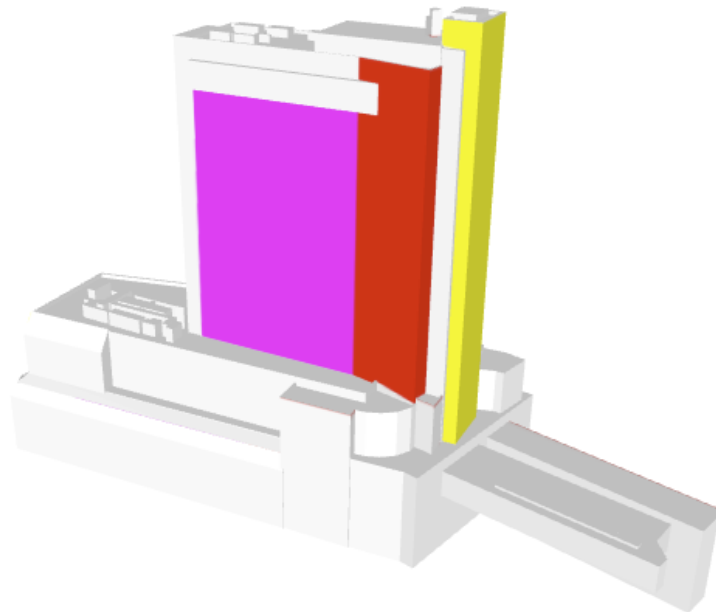


図 4-4 複数窓のデザインパターン対応

また、3D 都市モデルの LOD1 および LOD2 においてテクスチャデータが付属されていないエリアに関してはこのロジックが使えないため、自社独自でオープンデータをクロージングして同じ仕組みを実装する必要性が確認できた。

象徴的な看板や特殊な構造物においては、3D 都市モデルの地図精度が担保された正確な建物形状や付属のテクスチャデータからより正確に看板の位置とサイズを判定できることが確認できた。

また、アルゴリズム側で建物の 3D モデルの形状の中で看板の可能性の高い面を自動識別するエンジンを開発し、テクスチャデータからの識別と 3D モデルからの類推の両方のプロセスを実行することで精度の高い看板位置を推定することができた。

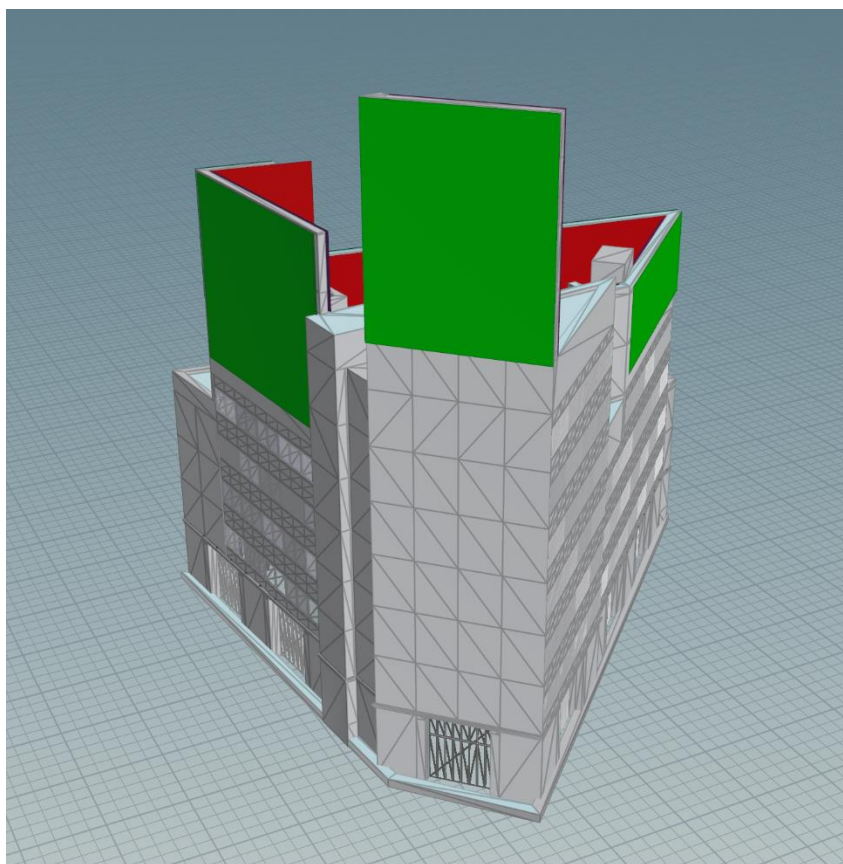


図 4-5 形状による看板判定

一方で、渋谷エリアにおいては、看板デザインそのものがランドマークになっている場合が多くあり、この場合は、看板に使用する画像データなどを、テクスチャデータや周囲の景観から予測し、生成 AI で近しい画像を自動生成していくなどの今後の対応策が検討できた。また、テクスチャデータが存在しない場合には自前のデータベースを拡充して判定を補う必要があることが確認できた。

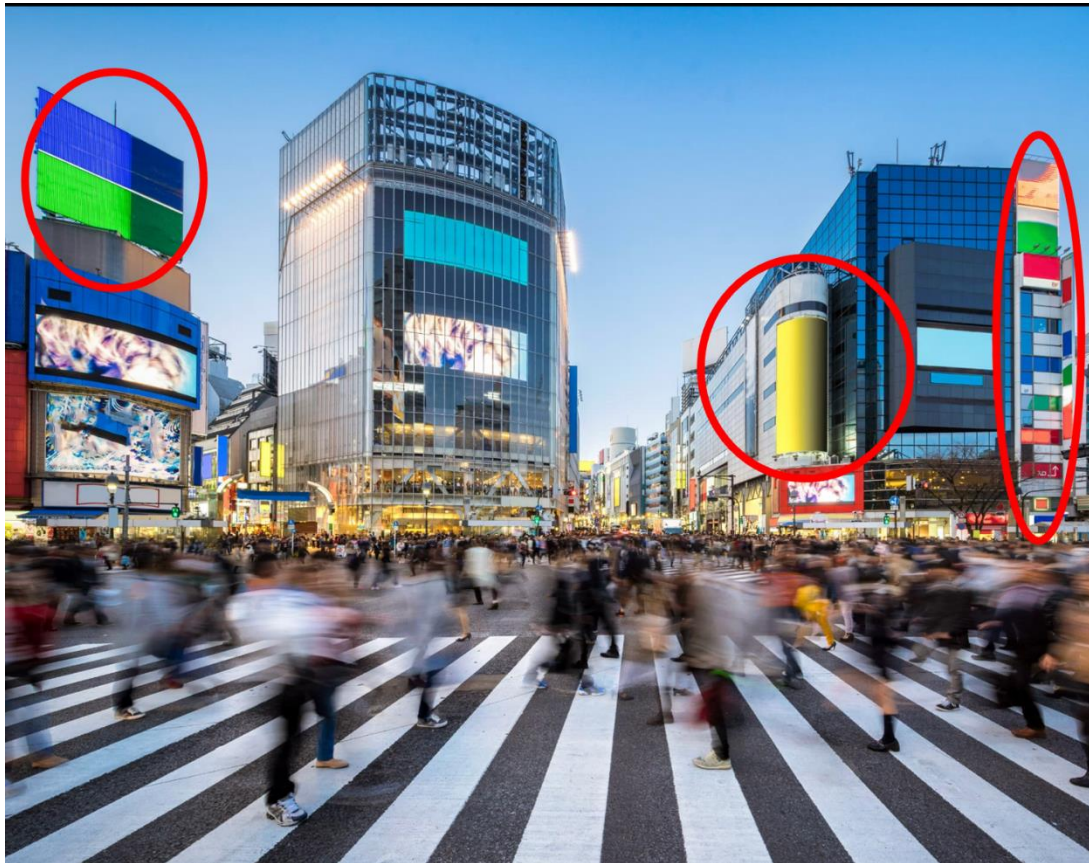


図 4-6 看板の色情報を抽出して当てはめた渋谷のイメージ

看板のデザインそのものが人々の記憶に残っているエリアに関しては、看板そのもののデザインを似せて再現する必要があることが分かった。そのためには、看板の色情報を識別してテクスチャの再生成を行うことや、新しいアルゴリズムを開発及び活用し、点群情報などの新しいレイヤーを用いて文字や模様を再現したデザイン追加することで、まちの再現や表現力を引き上げていくことが必要である。

4-2. パフォーマンス評価の検証

3D モデル生成時及びレンダリング時における 3D データ最適化アルゴリズムのパフォーマンス評価を検証した。

4-2-1. 検証目的

- 生成した高精度デジタルツインデータをコンシューマ向けコンテンツとして利用を想定した際に、画面表示時のパフォーマンス性能を検証することによって、本データの優位性を検証する。

4-2-2. KPI

表 4-5 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	表示速度 (FPS)	30FPS	● ゲームなどで遊べる最低限の速さを保証するための数値を目標値とした	● 標準的なゲーミング PC におけるゲームエンジンでの動作速度を測定

4-2-3. 検証方法と検証シナリオ

1) 表示速度の測定

標準的なゲーミング PC (Fortnite の最低動作保証条件を満たす PC) において、ゲームエンジンで描画をした際の FPS (※) 値を測定する

※frames per second : 1 秒間の動画が何枚の画像で構成されているかを示す単位

表 4-6 検証シナリオ一覧 (表示速度の測定)

No.	検証方法	エリア	検証シナリオ
1	表示速度の測定	新宿区	サービス利用者向けに配布した Fortnite マップを標準的なゲーミング PC を使用して 15 分間プレイし、1 分ごとの FPS 数値の最大値、最小値を集計する

4-2-4. 検証結果

検証には【HW001】HP ENVY Desktop TE02-1097jp の PC 用いて計測を行った。Fortnite 及び Unreal Engine 上では、KPI として設定した 30FPS 以上を達成しており、通常のゲームプレイや 3D コンテンツ作成作業を安定して行えることが確認できた。

1) 表示速度の測定

● Fortnite

標準的なゲーミング PC で、今回作成したデータを使用した Fortnite マップを 15 分間プレイした際の FPS の推移は以下のとおりとなった。

表 4-7 標準的なゲーミング PC で Fortnite をプレイした時の FPS

黄セル：KPI 達成

青セル：KPI 未達

プレイ時間	最大 FPS 値	最小 FPS 値
0:01～1:00	60	60
1:01～2:00	60	59
2:01～3:00	60	51
3:01～4:00	60	46
4:01～5:00	60	47
5:01～6:00	60	60
6:01～7:00	60	59
7:01～8:00	60	54
8:01～9:00	60	60
9:01～10:00	60	57
10:01～11:00	60	60
11:01～12:00	60	56
12:01～13:00	60	60
13:01～14:00	60	55
14:01～15:00	60	50

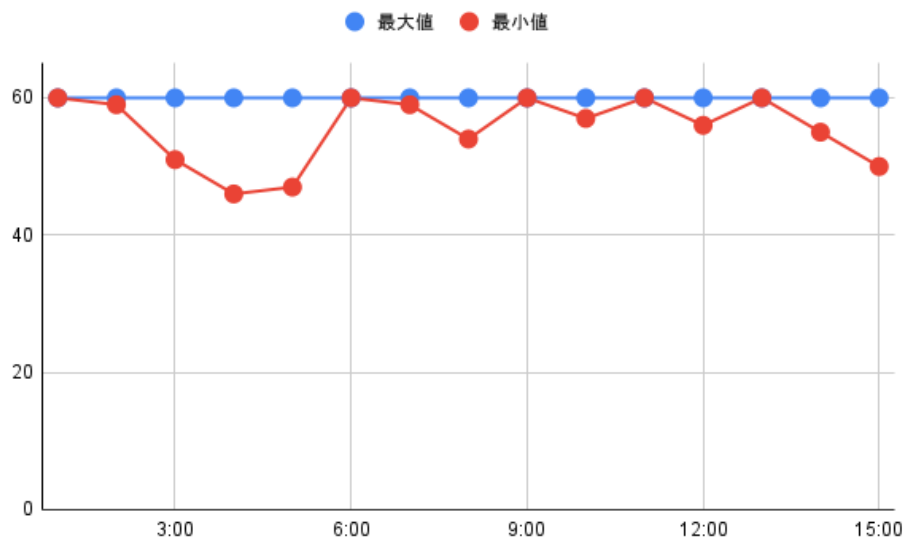


図 4-7 Fortnite プレイ時の FPS 推移

最大値は安定して 60FPS を計測しているが、プレイヤー自体の激しいアクション等が発生すると、45FPS 付近まで低下することもある。しかし動作終了後に 60FPS 付近まで回復しているため、パフォーマンスの大きな劣化にはつながらないと判断できる。

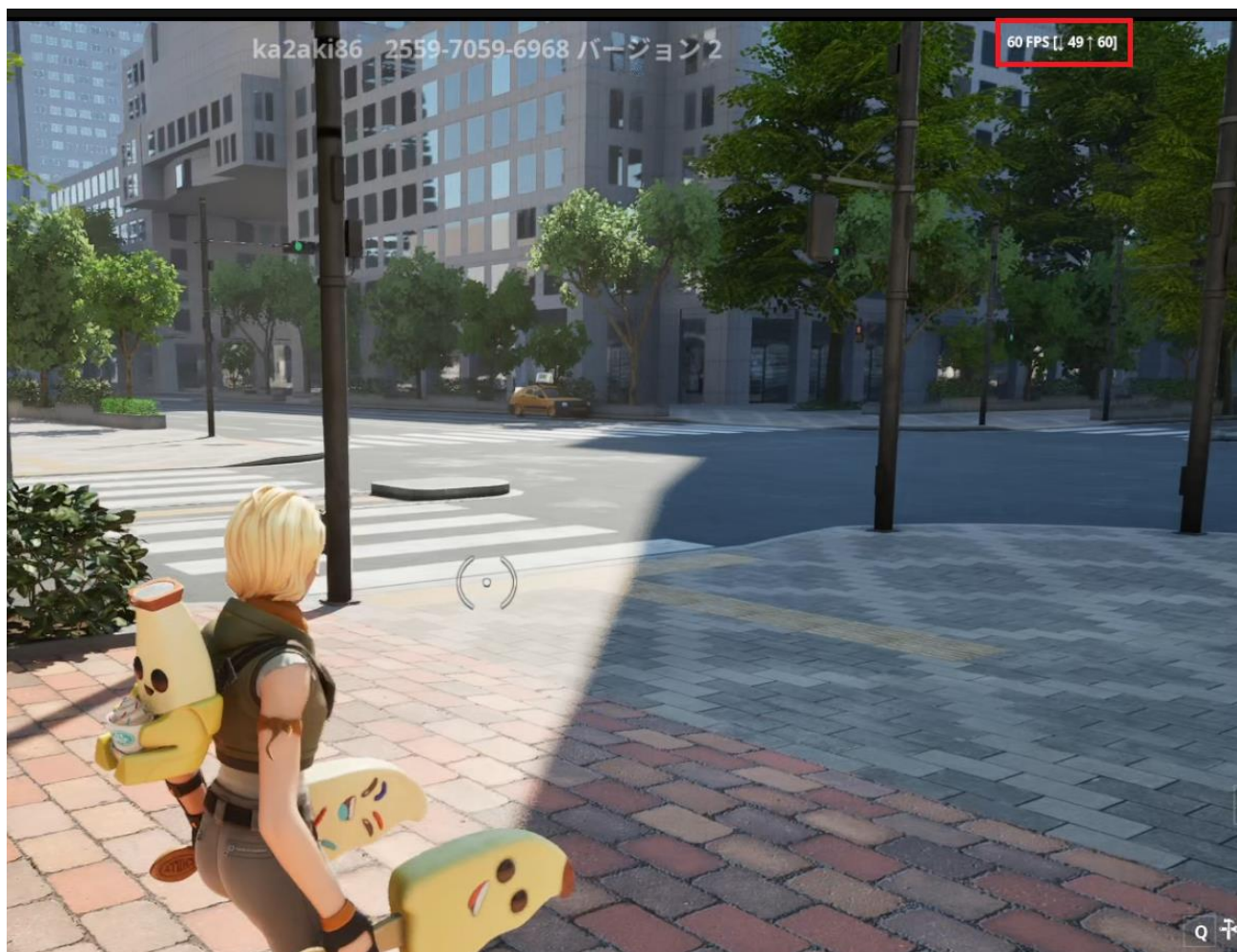


図 4-8 Fortnite を使用した FPS 計測イメージ

- Unreal Engine 5.1

Unreal Engine 5.1 で作成し、クリエイター向けに配布した新宿のマップ上を 5 分間動いて FPS 計測を行った。

表 4-8 Unreal Engine 5.1 を使用した際の FPS

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

プレイ時間	最大 FPS 値	最小 FPS 値
0:01~1:00	55	32
1:01~2:00	42	27
2:01~3:00	60	52
3:01~4:00	66	55
4:01~5:00	72	52

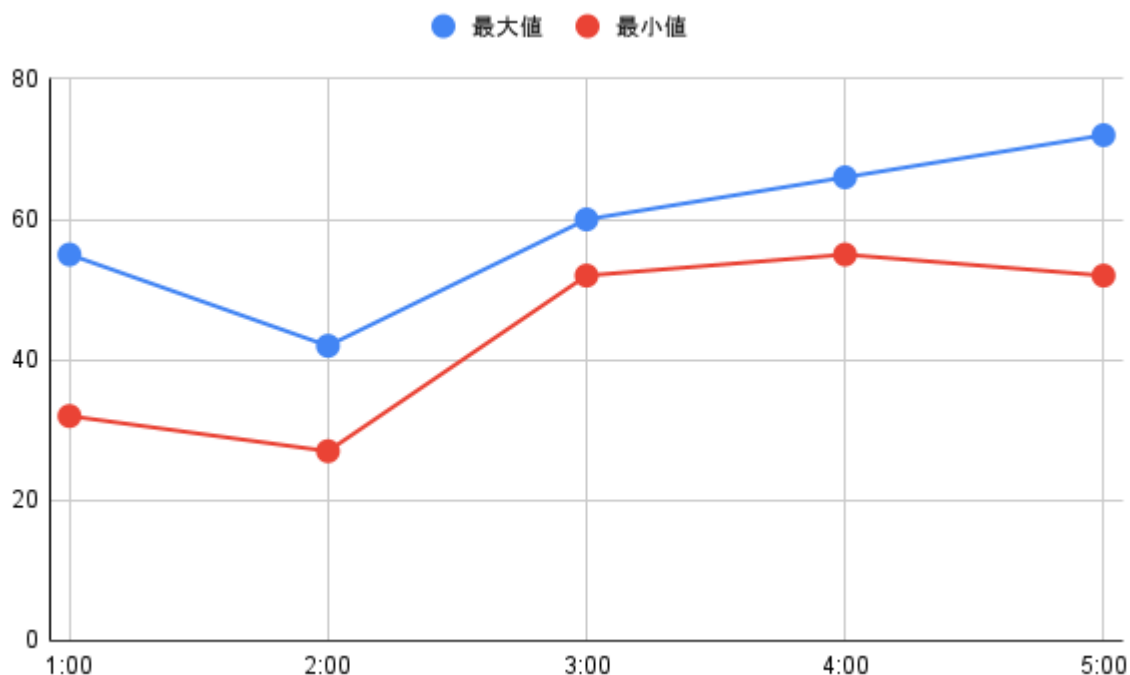


図 4-9 Unreal Engine 5.1 の FPS 推移

Unreal Engine 5.1 上でもマップの大部分で安定して 50~60FPS のパフォーマンスを出すことができた。しかし 2:00 前後のパフォーマンスが 30FPS 付近までの低下したタイミングがあり、その要因としてはビルの密集部分や木などのオブジェクトが多いところで画面表示や切替えに処理時間を要していることが確認できた。

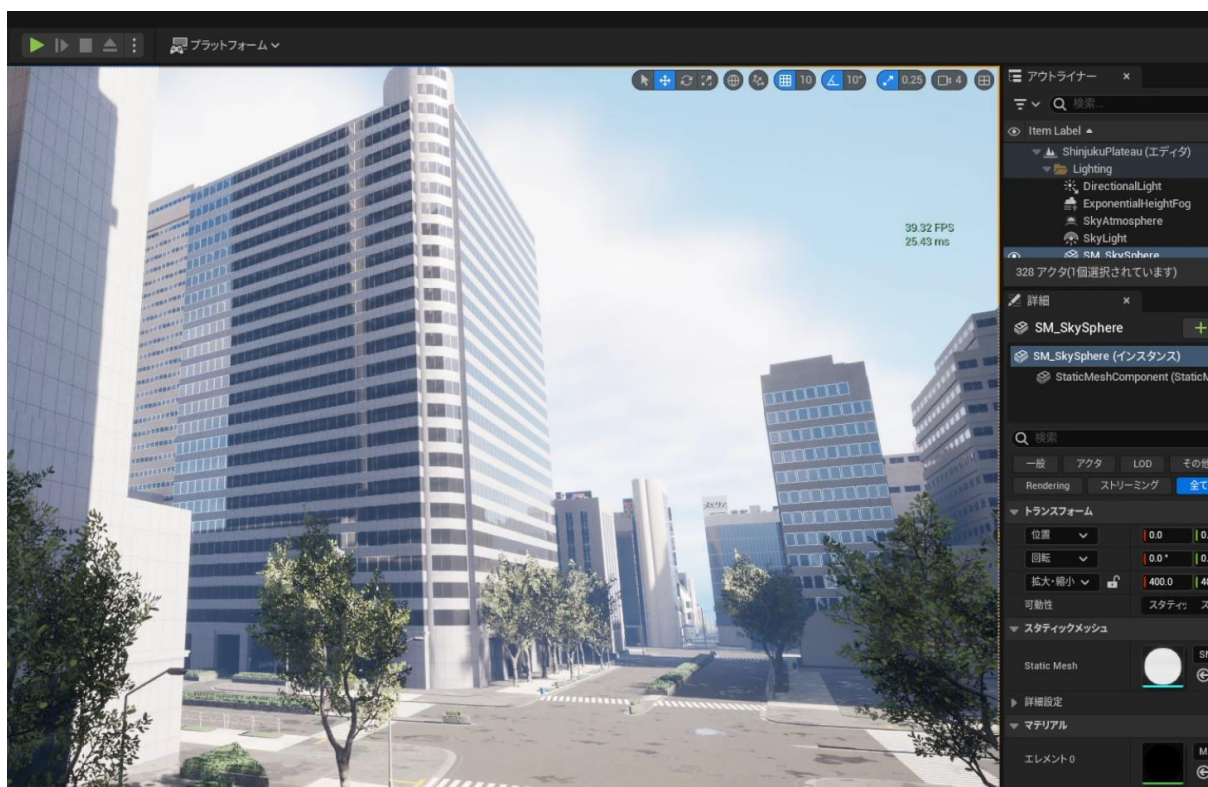


図 4-10 Unreal Engine 5.1 を使用した FPS 計測イメージ

4-3. デジタルツインデータの位置精度検証

空間配置アルゴリズムを使用して生成したデジタルツインデータにおける建物配置の位置精度を検証した。

4-3-1. 検証目的

- 位置精度が担保されたデジタルツインデータは様々な 3DCG 技術を必要とするサービスへの展開が想定される。今回の実証実験のように 3D 都市モデルの公共測量データを教師データとして追加し、AI による生成アルゴリズムを通して作成した 3D モデルの位置精度を検証することで、アルゴリズムの有用性を検証する。

4-3-2. KPI

表 4-9 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	IoU (Intersection over Union)	0.6	<ul style="list-style-type: none"> ● 地図データなども利用しているため、平面における一致精度は高いと想定する ● しかし、現在のアルゴリズムでは土地の高低差を考慮できておらず、実際の測量データとは誤差が出ると考えられるため低めの値を設定する 	位置精度の調査と検証

IoU：2 種類のポリゴンデータなどがどれだけ重なっているかを示す指標であり、一般的に画像処理や地理情報システム（GIS）の分野で利用される

4-3-3. 検証方法と検証シナリオ

1) 位置精度

平面図における建物敷地の一致度を評価する

- IoU (Intersection over Union)

3D 都市モデルの建築物モデルが持つ地面ポリゴンと生成したデータの地面ポリゴン情報から IoU を計算する。

分子：交差部分の面積（比較対象のポリゴンが重なり合う部分の面積）

分母：合併部分の面積（比較対象のポリゴンの重なり合う部分と重なり合わない部分を含む全体の面積）

$$\text{IoU} = \frac{\text{交差部分の面積}}{\text{合併部分の面積}}$$

表 4-10 検証シナリオ一覧（位置精度精度）

No.	検証方法	エリア	検証シナリオ
1	位置精度算出	新宿区	IoU を用いて 3D 都市モデルと生成したデジタルツインデータの交差部分の面積を比較する

4-3-4. 検証結果

IoU の位置精度は目標を達成した。3D 都市モデルを用いて、建築物の 2D 図面情報を活用することで AI による 3D モデルの再生成をしても一定の位置精度が担保されていることが確認できた。しかし、モデル生成のアルゴリズムの仕様上、テクスチャと形状の優先度や閾値により建物自体が削除されることや、大きく位置ズレを起こすことも確認されたため、3D 都市モデルも位置情報をそのまま取り扱う仕様へアルゴリズム自体を改変するなど、高度化に向けた課題点が明らかになった。

（測量観点での優位性についてはアジア航測による検証内容を 5 章で記載）

1) IoU

表 4-11 IoU による位置精度検証結果

黄セル：KPI 達成

青セル：KPI 未達

検証内容	評価指標・KPI	目標値	結果		示唆
			項目	評価値（既存手法での評価値）	
位置精度	IoU	0.6 以上	建物生成位置	0.87	<ul style="list-style-type: none"> ● 全体の面積ベースでは生成位置に対する目標値は達成した。 ● しかし、3D 都市モデル側に存在するが、生成ができていない建物が存在している。ランドマークのような大きいビルの生成の誤差はなく、小さな家屋や建築物の生成漏れが確認できた。

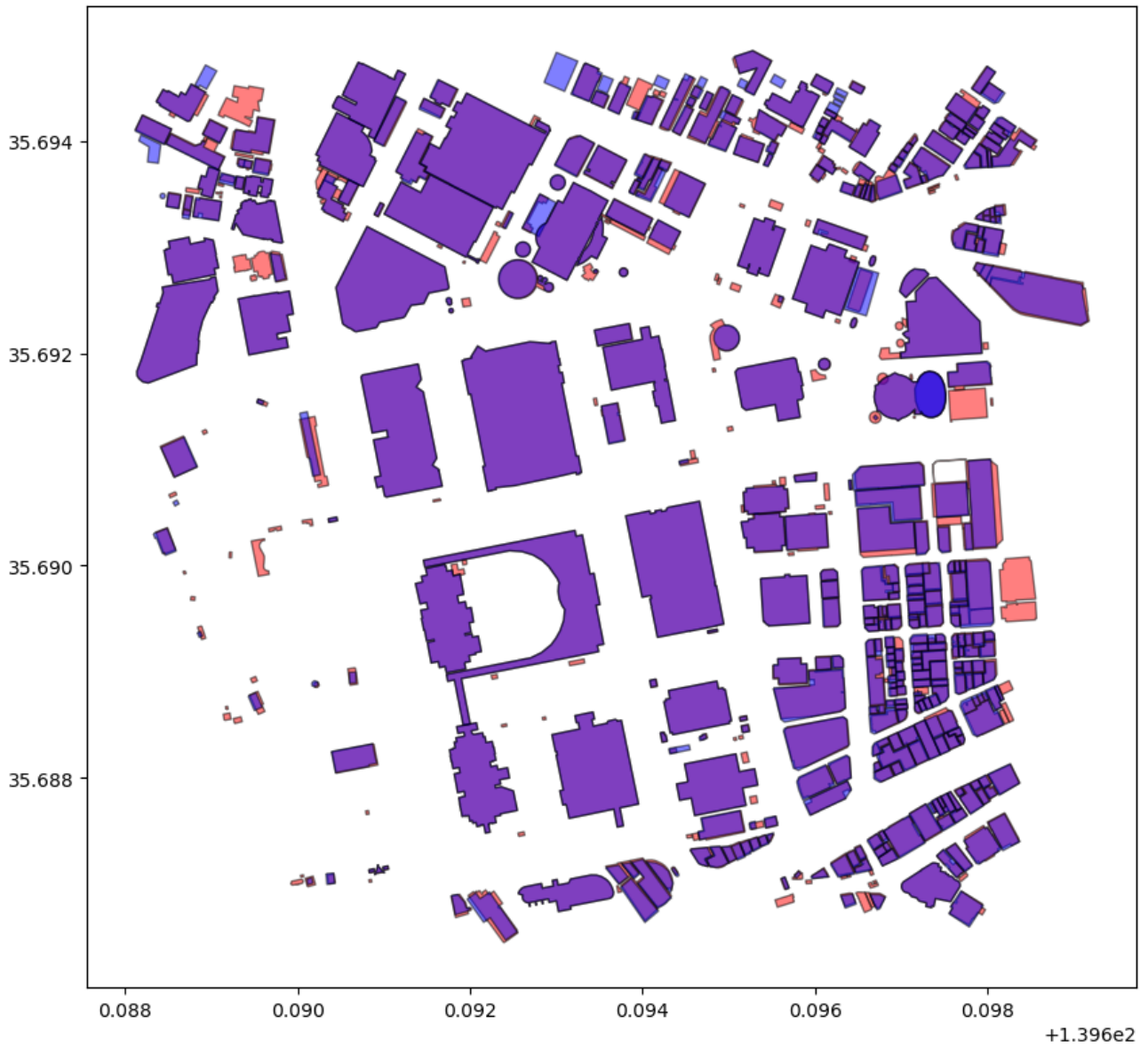


図 4-11 PLATEAU 上の建物ポリゴン（赤色）と生成したポリゴン（青色）と重なりポリゴン部分（紫色）

5. 測量成果としての有用性検証

5-1. 検証の前提について

5-1-1. 検証データの概要

本検証に使用するデータは①3D モデルデータと②原点地理座標値によって構成される。

① 3D モデルデータ

任意座標系の座標値で表されたジオメトリ情報により表現されている 3D モデルのデータであり、この任意座標系は真北方向を軸の一つとして持つ直交座標系である。

② 原点地理座標値

データ全体を地理座標系の位置に合わせるために設定されている任意座標系の原点 (0,0) の地理学的経緯度の座標である。

これらのデータを組み合わせて使用することで、任意座標系の座標値で表された 3D モデルデータを地理座標系のデータと比較するために水平移動（シフト）することができる。今回はこのデータのうち、建築物と道路を表すモデルデータを検証対象とした。

5-1-2. 3D モデルデータの形状

3D モデルデータの高さ方向に関して、本システムで作成されたデータの高さは全て Z=0m を基準面としている。そのため、建築物モデルは底面高を 0m としてそこから立ち上げが行われた形状を持つ（図 5-1 左）。また、道路に関しては全ての頂点が高さ 0m の水平面形状である（図 5-1 右）。

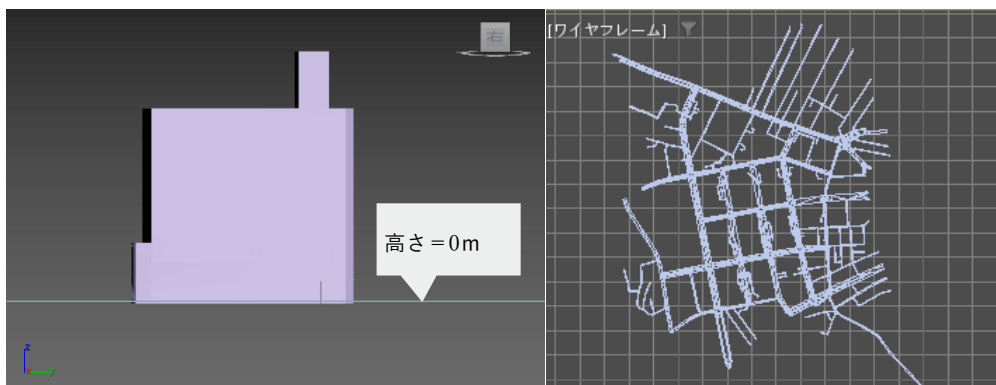


図 5-1 本システムで作成されたデータの建築物モデル側面図（左）と道路モデル上面図（右）

3D モデルの位相形状に関して、本システムで作成されたデータの建築物モデル形状は 3D 都市モデルのようなソリッドモデル形状ではなくサーフェスモデル形状である。図 5-2 は本システムで作成されたデータの建築物モデルをワイヤフレーム表示し、建築物モデルの内部に存在している一部の面を赤色で表示したものである。

このように本システムで作成されたデータの建築物モデルはソリッドを構成しない面の集まりで作成されている。道路モデルの形状は 3D 都市モデル（LOD1）同様平面のサーフェスモデル形状である。

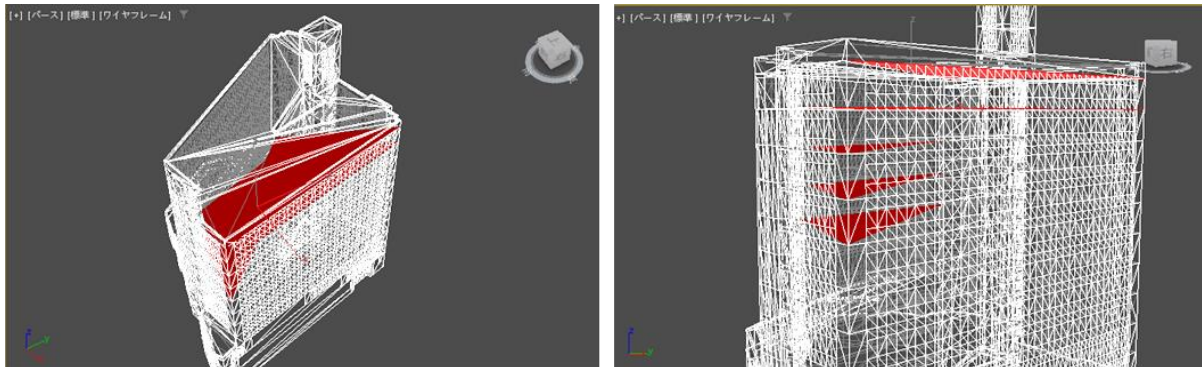


図 5-2 本システムで作成されたデータ建築物（ワイヤフレーム表示）

5-1-3. 検証に使用したデータ

本検証では真値となるデータとして①3D 都市モデルと②空中写真を用いた。

① 3D 都市モデル

- G 空間情報センター (<https://www.geospatial.jp/ckan/dataset/plateau>) で公開されている PLATEAU オープンデータ（東京都 23 区 2022 年）をダウンロードし、建築物モデルと道路モデルのデータを平面直角座標に変換して使用した。今回の対象範囲において、建築物は LOD2（一部 LOD1）、道路は LOD1 の詳細度のモデルである。

② 空中写真

- 本検証で作成したデータと 3D 都市モデルが異なる箇所の確認や検証用図化を行うために空中写真を使用した。本検証では、2023 年 3 月に撮影された地図情報レベル 2500 の空中写真を使用した（承認番号：令 4 関公第 983 号）。

5-1-4. 検証対象範囲

検証の対象範囲は本システムで作成されたデータの建築物モデルが存在する領域とした（図 5-3）。その領域の一部に、モデルデータは存在するものの、検証用の空中写真上では工事中であり、建築物が存在しない箇所があった（図 5-4）。本検証では本システムで作成されたデータと 3D 都市モデルを比較し、差異があった場合には空中写真による確認を行うため、この箇所はあらかじめ検証対象範囲から除外した。

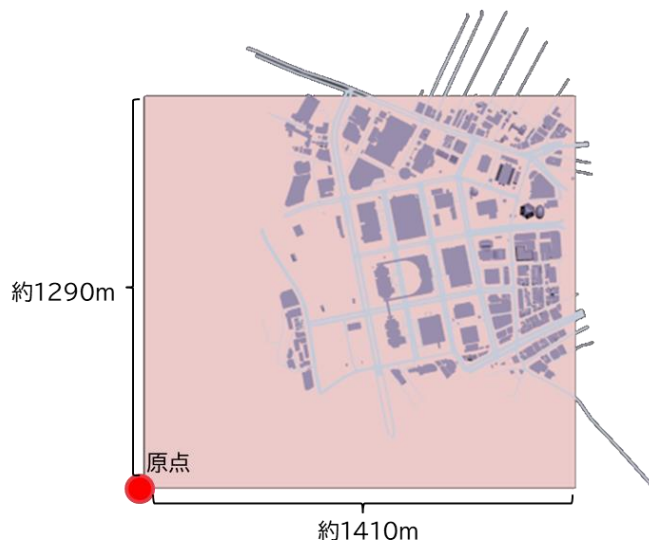


図 5-3 本システムで作成されたデータセットの範囲

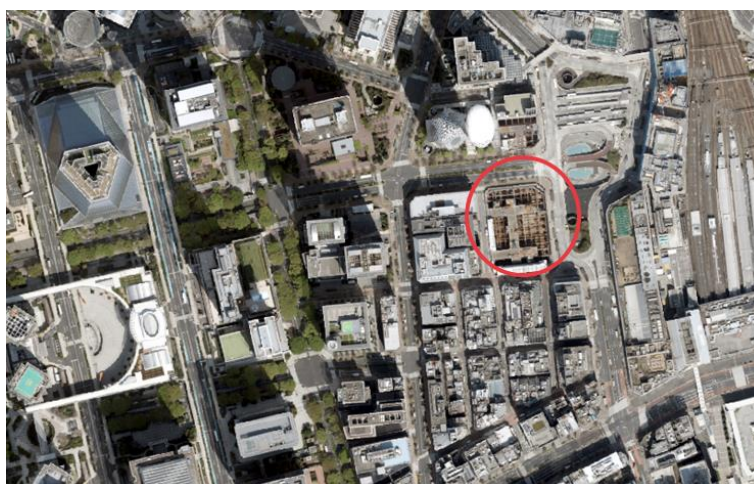


図 5-4 対象範囲内の工事箇所（赤丸内）

5-2. 完全性の検証

今回の実証実験で作成した高精度デジタルツインデータに対して、地物の存在及び欠落に関する評価である完全性（過剰・漏れ）の検証を行った。

5-2-1. 検証目的

- 本システムで作成されたデータと 3D 都市モデルの地物の有無に関する相違点を確認する

5-2-2. KPI

表 5-1 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
1	3D 都市モデル に対する建築物 単位での完全性	過剰・漏れ・ その他異常の 数量=0	● 3D 都市モデルをイン プットデータとして 作成したデジタルツ インデータのため、 生成前の各データは 引き継いだ状態でア ップデートされてい る状態が望ましいた め。	● 3D 都市モデルと建築物 単位で比較し、差異があ る箇所については空中写 真を用いた確認・評価を する
2	3D 都市モデル に対する建築物 部材単位での完 全性	過剰・漏れの 数量=0		● 3D 都市モデルと建築物 の部材単位で比較し、差 異がある箇所については 空中写真を用いた確認・ 評価をする
3	3D 都市モデル に対する道路の 完全性	過剰・漏れの 数量=0		● 3D 都市モデルと道路単 位で比較し、差異がある 箇所については空中写真 を用いた確認・評価をす る

※都市開発等による現実世界のアップデートを反映しているものについては、3D 都市モデルとの完全性を逸脱しているものではないとする

5-2-3. 検証方法と検証シナリオ

本システムで作成されたデータと 3D 都市モデルの比較に加え、差異が認められた部分については空中写真とも比較することで実在の確認を行った。これにより、本システムで作成されたデータ作成時に新設や滅失等の更新ができていない可能性の検証、その評価を行った。建築物モデルの完全性検証では、①建築物単位での完全性検証と②建築物部材単位での完全性検証を行った。

① 建築物単位での完全性検証

- 建築物単位での存在の有無に関する検証を実施した。

② 建築物部材単位での完全性検証

- 建築物の部材（屋根の上に存在する地物）の有無という観点での検証を実施した 3D 都市モデルとの比較を行うため、LOD2 で表現される屋根の形状について検証した。また、存在の有無に加えて形状の高精細化・劣化という観点でも評価を実施した。

5-2-3-a. 建築物単位での完全性検証

建築物単位での完全性検証では、3D 都市モデルと本システムで作成されたデータそれぞれの建築物モデルの存在の比較を行った。その結果、違いがあったものに関しては空中写真を用いて、実在する/実在しないを判定した。本検証は本システムで作成されたデータの範囲に存在する建築物モデル全数に対して実施した。図 5-5 に建築物単位での完全性検証フローを示す。

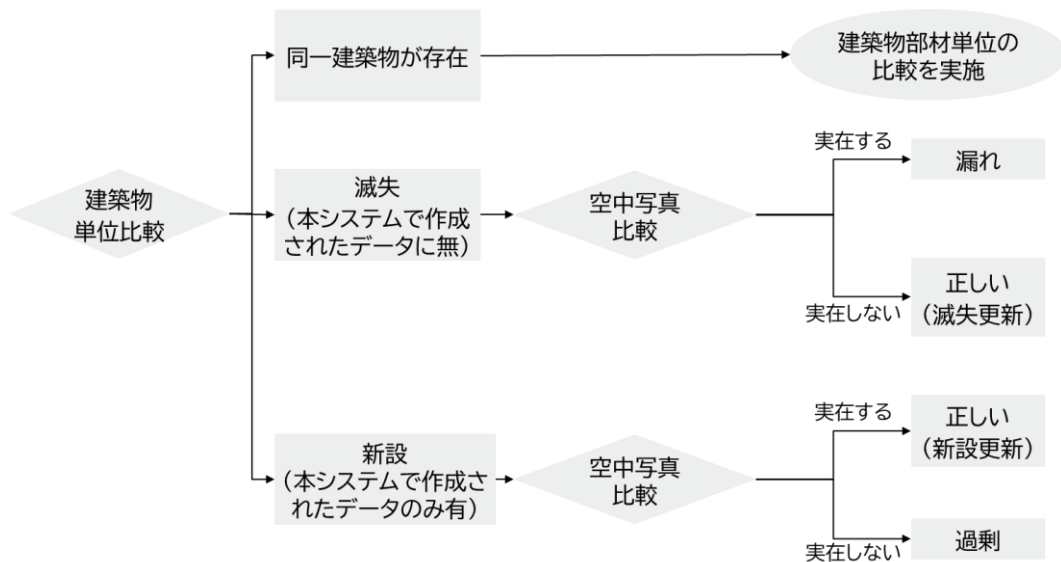


図 5-5 建築物単位での完全性検証フロー

5-2-3-b. 建築物部材単位での完全性検証

建築物部材単位での完全性検証は、建築物単位での完全性検証の結果、3D 都市モデルと本システムで作成されたデータ間で同一の建築物モデルが存在するものに対して抜き取りで実施した。検証結果は建築物モデルが同一形状の場合はその棟数、同一形状以外の場合には部材の箇所数で評価した。図 5-6 に建築物部材単位の完全性検証のフローを示す。

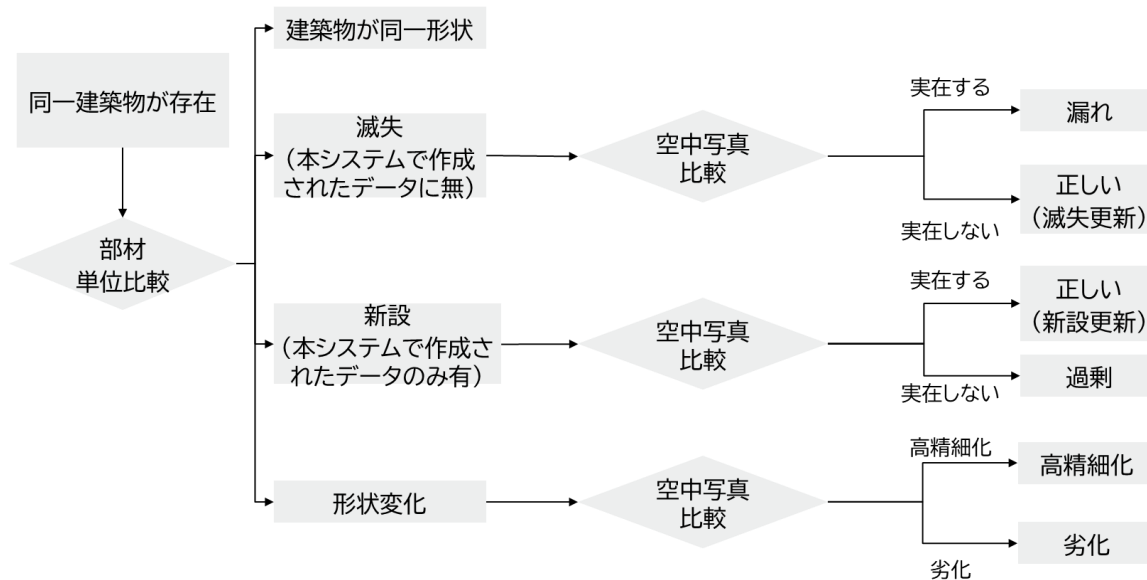


図 5-6 建築部材単位の完全性検証フロー

建築物単位及び建築物部材単位での完全性検証の結果パターンを表 5-2 に示す。

表 5-2 建築物モデル完全性検証パターン

パターン	本システムで作成されたデータと 3D 都市モデルの比較		写真による確認	本システムで作成されたデータに関する説明
	建築物単位	部材単位		
1	同一	同一	-	3D 都市モデルと同じ建築物、同じ形状を表現
2		減失	実在する	部材形状の作成漏れ
3			実在しない	部材の減失を更新
4			新設	実在する
5		実在しない		部材の過剰
6		形状変化	高精細化	3D 都市モデルよりも形状が詳細化
7			劣化	3D 都市モデルよりも形状が劣化
8		減失	対象外	実在する
9	対象外		実在しない	建築物の減失を更新
10	新設	対象外	実在する	建築物の新設を更新
11		対象外	実在しない	建築物の過剰
12	形状異常	-	-	上記に当てはまらない形状の異常

5-2-3-c. 道路の完全性検証

道路モデルの完全性検証では、本システムで作成されたデータと 3D 都市モデルを比較し両者の表現しようとしている道路の有無に関する違いを抽出し、空中写真によってその地点の実態を確認した。その確認結果から本システムで作成されたデータが下記に示す 5 つのうちのどれに該当するかを評価した。

- ① 過剰：実在しない道路を表現している場合
- ② 漏れ：実在する道路を表現していない場合
- ③ 新設：3D 都市モデルに存在しなかった道路を新たに作成している場合
- ④ 減失：3D 都市モデルに存在するが本システムで作成されたデータには存在せず、写真上は存在しない場合
- ⑤ 図化基準の違い：民地内等の道路（駐車場等）を作成している場合

本システムで作成されたデータの道路モデルの形状は車道と推定される部分の形状を有しており、道路縁（車道部だけでなく、歩道部も含めた道路の端）の形状を持つ 3D 都市モデルとは表現されている形状が大きく異なる。そのため、道路モデルの完全性形状に関しては道路形状の違いに関する評価は実施しなかった。

5-2-4. 検証結果

地物の完全性に関しては、本システムで作成されたデータと 3D 都市モデルの図化基準の違いによるものが存在する可能性はあるが、3D 都市モデルには表現されていない地物を新たに生成・高精細化することができていた。これはスペースデータ社のデジタルツインデータ作成技術を用いることで経年変化により新設された地物などを新たに生成ないし形状再現度を向上することができる可能性を示している。一方、3D 都市モデルに存在しており、空中写真の時点（2023 年 3 月撮影）にも存在している地物を作成されていない箇所があり、原典データとした 3D 都市モデルから形状的な情報量が低下している可能性を示しているといえる。

表 5-3 検証結果サマリ

黄セル：KPI 達成 青セル：KPI 未達

検証内容	評価指標・KPI	目標値	結果				示唆
			項目	評価	評価値	割合 (%)	
建築物単位での完全性（範囲内全数）	完全性	過剰・漏れ・その他異常の数量=0	3D 都市モデルと同じものが存在	同一	344 棟	69	<ul style="list-style-type: none"> ● 3D 都市モデルに存在するが、本システムで作成されたデータに存在しない建築物が多数見られ、変換時に情報が抜け落ちている可能性がある。 ● 5 棟の形状異常は明らかに建築物形状を成しておらず、変換過程でのエラーの可能性はある。
			3D 都市モデルにあるものが存在しない。	漏れ	96 棟	19	
				滅失更新	35 棟	7	
			3D 都市モデルにないものが存在する。	新設更新	8 棟	2	
				過剰	9 棟	2	
			その他	形状異常	5 棟	1	
建築物部材単位での完全性（69 棟抜き取り）	完全性	過剰・漏れの数量=0	3D 都市モデルと同じものが存在	同一	40 棟		<ul style="list-style-type: none"> ● 3D 都市モデルに存在するが、本システムで作成されたデータに存在しない建築物部材や形状の劣化が生
			3D 都市モデルにあるものが存在しない。	部材漏れ	29 か所		
				滅失更新	2 か所		
			3D 都市モデルにないものが存在する。	新設更新	2 か所		
				部材過剰	12 か所		

			形状変化	高精細化	6 箇所		じている箇所が見られ、変換時に情報が抜け落ちている可能性がある。
				劣化	15 箇所		
道路の完全性 (範囲内全域)	完全性	過剰・漏れの数量 =0	3D 都市モデルにあるものが存在しない。	過剰	0 箇所		● 3D 都市モデルと本システムで作成されたデータの図化基準が異なることによるものが多いが、漏れの2箇所については、現地に存在する車道が作成できていないものであった。
				滅失	0 箇所		
			3D 都市モデルにないものが存在する。	漏れ	2 箇所		
				新設	0 箇所		
			その他	図化基準の違い	21 箇所		

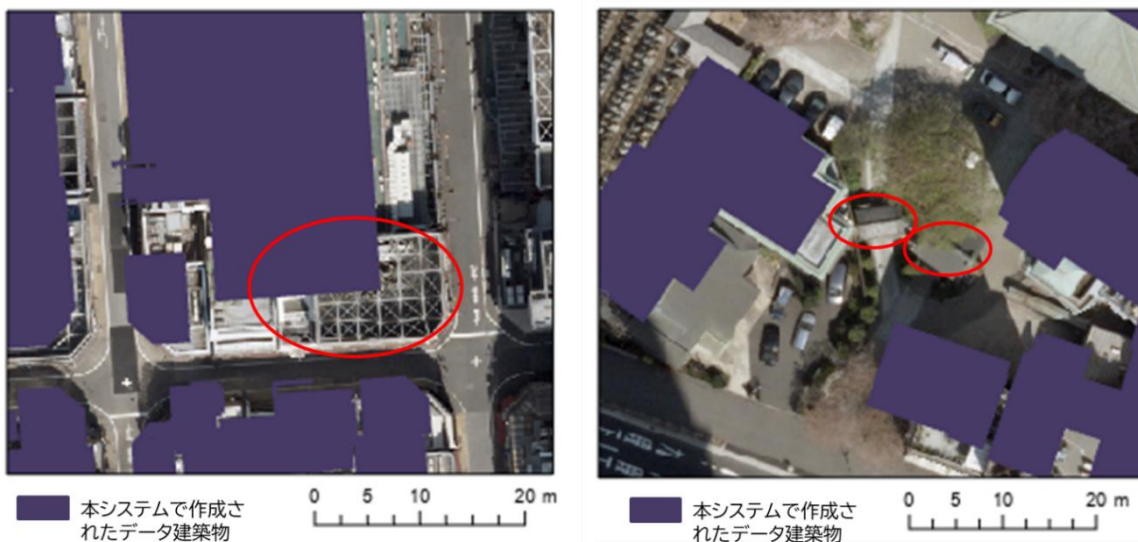
5-2-4-a. 建築物単位での完全性検証

検証対象範囲に存在する本システムで作成されたデータの建築物モデル 366 棟と、同じ範囲に存在する 3D 都市モデルの建築物モデルを全数比較した結果を表 5-4 に示す。

表 5-4 建築物単位の完全性検証結果

パターン	本システムで作成されたデータに関する説明	棟数	割合 (%)
1~7	3D 都市モデルと同じ建築物が存在	344	69
8	建築物の作成漏れ	96	19
9	建築物の滅失を更新	35	7
10	建築物の新設を更新	8	2
11	実在しない過剰な建築物が存在	9	2
12	同一と思われる建物が存在するが、形状異常	5	1

パターン 8 の作成漏れに関して、3D 都市モデルの建築物モデルの中でも比較的小さなものに本システムで作成されたデータの漏れが多数あった。今回検証に使用した 3D 都市モデルの建築物モデルの平均面積が約 544 m²であるのに対して、作成漏れに該当した建築物モデルの平均面積は 77 m²であった。また、作成漏れの 96 棟のうち 100 m²以上の建築物モデルは 13 棟であった。図 5-7 に作成漏れと判定された建築物の例を示す。

図 5-7 建築物モデルの作成漏れの例（左：100 m²以上の建築物 右：100 m²未満の建築物）

パターン 10 の新設された本システムで作成されたデータの建築物モデルの中には、明らかに現地の建築物と異なる形状の建築物モデルもあった。図 5-8 に新設された建築物モデルの例を示す。オルソ画像に示されるように実際は地下への入り口の建屋であり、屋根形状としては平屋根のような形状であるが、本システムで作成されたデータ上では一般家屋のような形状表現がされていた。このような場合も建築物単位としては存在して

いるため、新設として扱った。

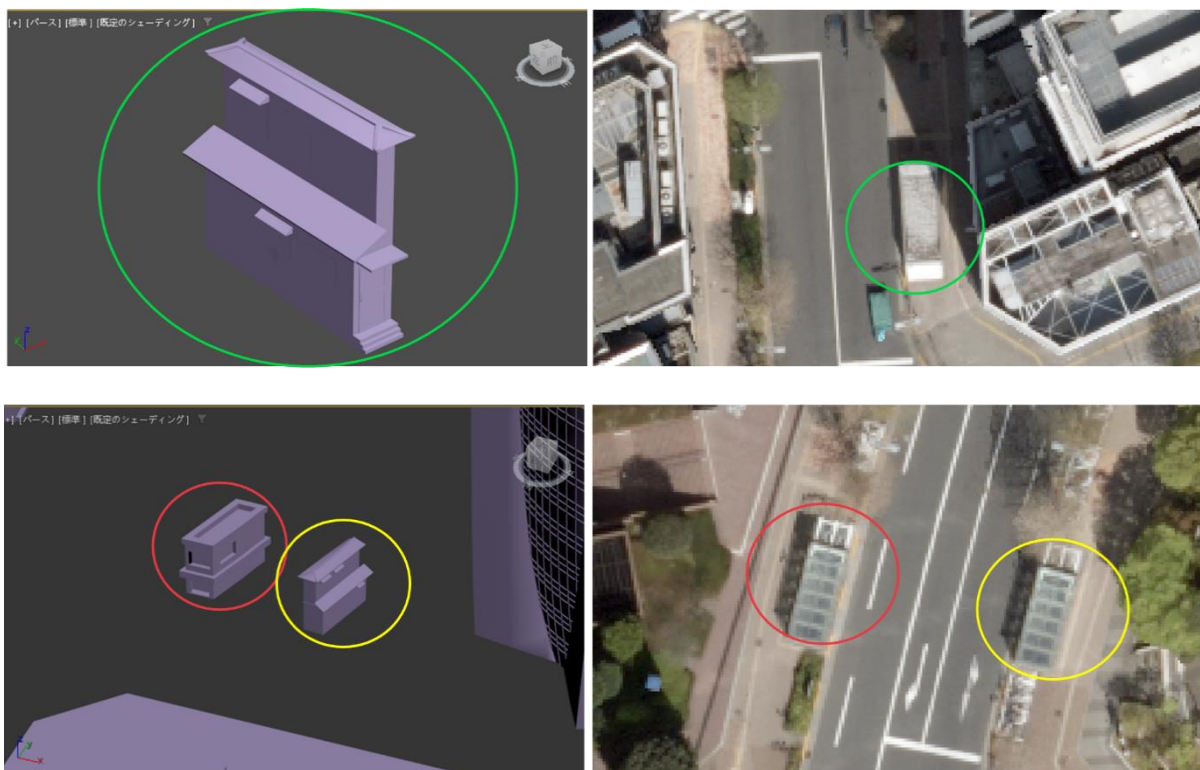


図 5-8 新設された建築物モデルの例

(左：本システムで作成されたデータ建築物モデル 右：オルソ画像)

本システムで作成されたデータの建築物モデルの中に図 5-9 に示すような形状異常の地物が存在した。本システムで作成されたデータの建築物モデルのうち、形状異常のものに関してはパターン 12 の形状異常の項目を追加して集計した。

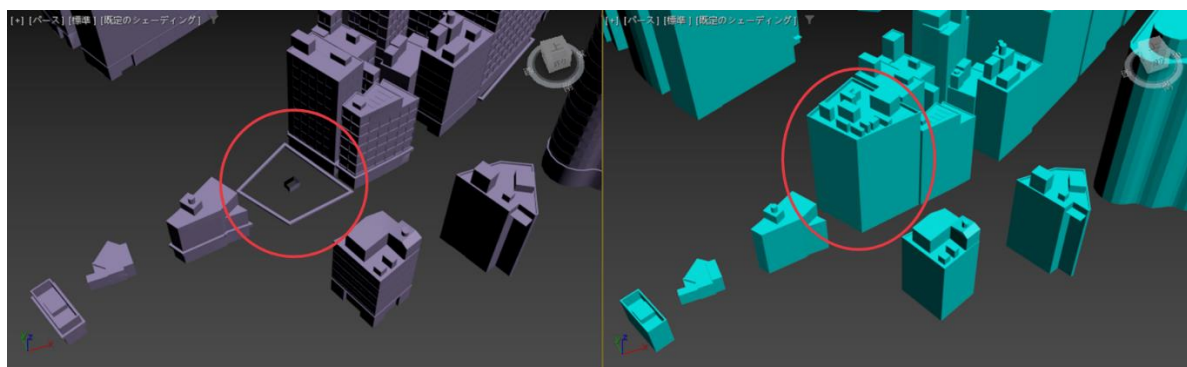


図 5-9 形状異常の例

(左：本システムで作成されたデータ建築物モデル 右：3D 都市モデル建築物モデル)

5-2-4-b. 建築物部材単位での検証

建築物単位での完全性検証の結果、3D 都市モデルと本システムで作成されたデータに同一建築物が存在する地物の中から約 20%に当たる 69 棟を無作為に抜き取り、部材単位での完全性検証を行った。パターン 2～7 に関しては、パターン 1 に属さなかった 29 棟の中に存在する建築物モデルの部材に対して 2～7 に当てはまる地物数を集計した。検証結果を表 5-5 に示す。

表 5-5 建築物部材単位の完全性検証結果

パターン	本システムで作成されたデータに関する説明	数量
1	3D 都市モデルと同じ建築物、同じ形状を表現 (※棟数で集計)	40 棟
2	部材形状の作成漏れ	29 箇所
3	部材の滅失を更新	2 箇所
4	部材の新設を更新	2 箇所
5	過剰な部材が存在	12 箇所
6	3D 都市モデルよりも形状が高精細化	6 箇所
7	地物は存在するが 3D 都市モデルよりも形状再現性が劣化 又は、誤った形状に変化	15 箇所

図 5-10 から図 5-12 に部材比較結果の例を示す。図 5-10 では、屋根の形状に枠部分が追加され、3D 都市モデルの形状よりも高精細化していた。図 5-11 では、実際の形状が切妻屋根形状であり、3D 都市モデルでは正しく屋根形状が表現されているのに対し、本システムで作成されたデータは陸屋根のような形状で表現されており、誤った形状への変化が見られた。図 5-12 は、部材の表現がそもそもされておらず、部材の作成漏れが生じている箇所と形状の形状再現性が劣化している箇所が 1 棟の建築物モデルの中に複合している事例である。

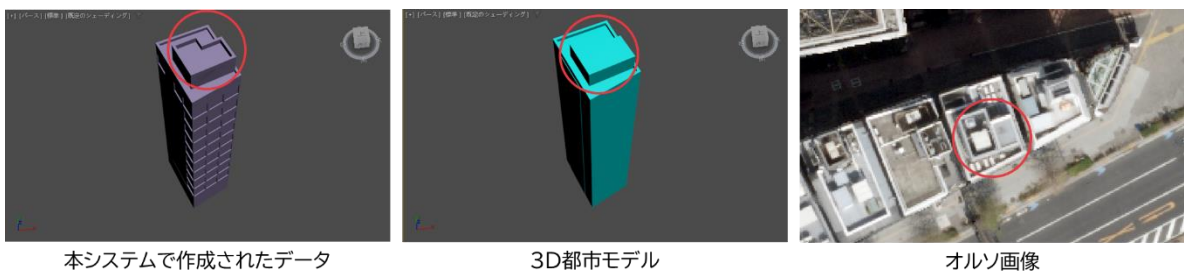


図 5-10 部材比較例 (赤丸：高精細化)

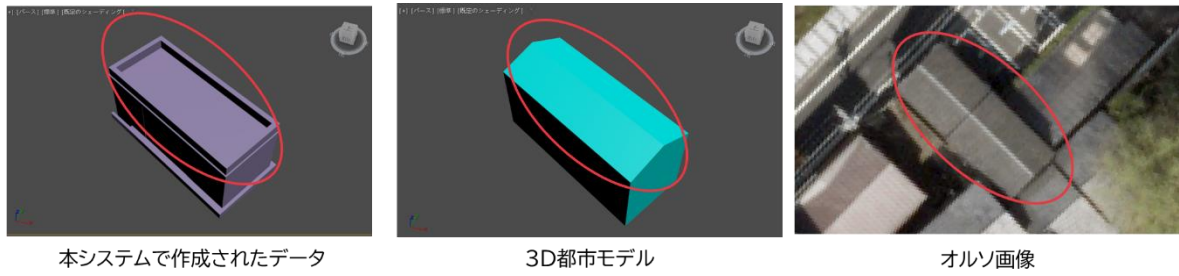


図 5-11 部材比較例（赤丸：劣化）

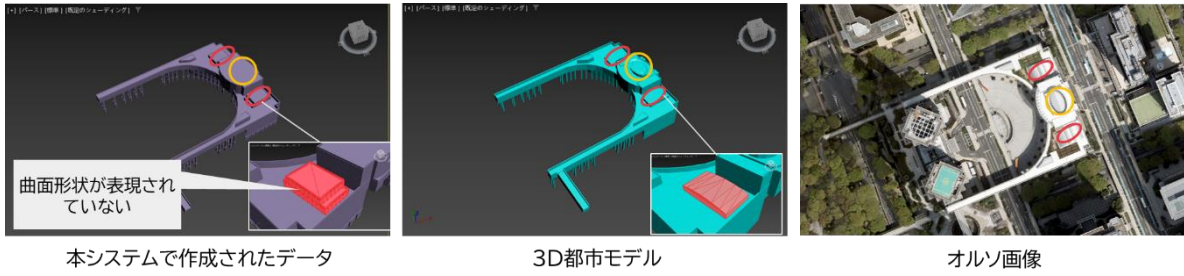


図 5-12 部材比較例（黄丸：作成漏れ 赤丸：劣化）

5-2-4-c. 道路の完全性検証

道路モデルの完全性検証の結果を表 5-6 に示す。本システムで作成されたデータと 3D 都市モデル間での差異については、民地内等の敷地にある道路（駐車場等）を作成している箇所によく見られた。図 5-13 に道路の比較結果例を示す。

表 5-6 道路完全性検証結果

状況	本システムで作成されたデータに関する説明	数量（箇所）
過剰	実在しない道路を表現	0
滅失	道路の滅失を更新	0
漏れ	実在する道路の作成漏れ	2
新設	3D 都市モデルに存在しなかった道路を新規作成	0
図化基準の違い	民地内等の道路（駐車場等）を作成	21

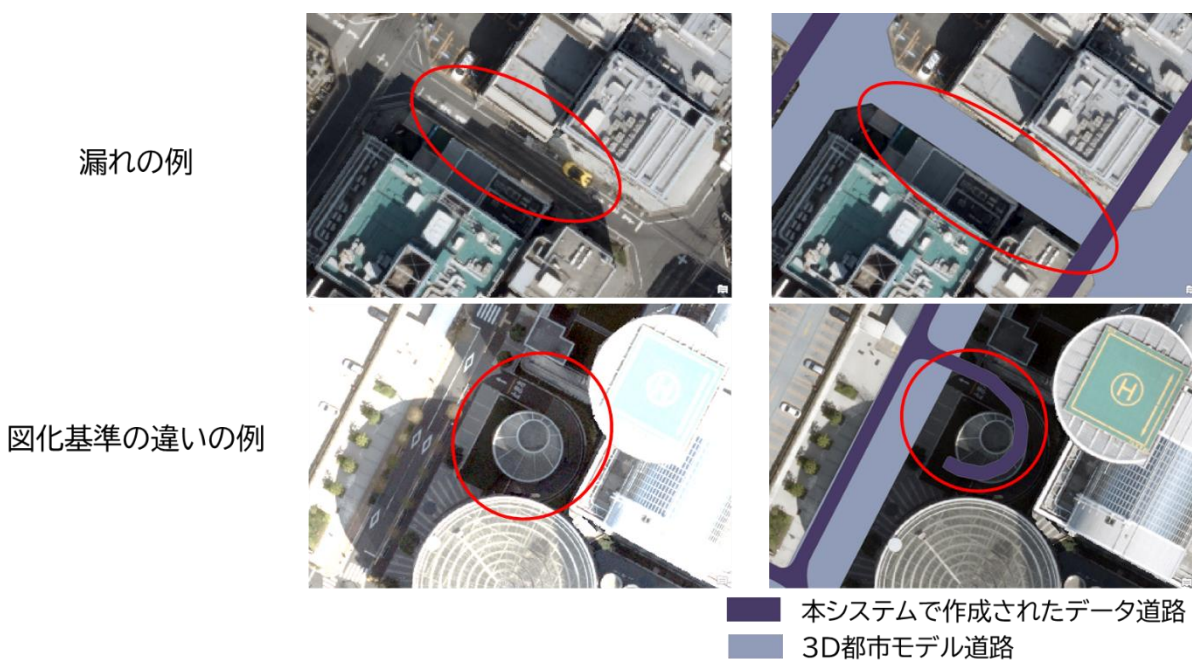


図 5-13 道路比較例

5-3. 位置正確度の検証

建築物モデルの水平絶対位置正確度の検証と相対位置正確度の検証を行った。検証は 3D 都市モデルを真値として実施したが、一部 3D 都市モデルと本システムで作成されたデータの形状が異なる箇所があったため、その部分は空中写真図化の結果を真値として検証を行った。

5-3-1. 検証目的

- 任意座標を平面直角座標にシフトした後、座標としての正しさである絶対位置正確度を評価
- 地物の相対的な距離の正しさである相対位置正確度を評価

5-3-2. KPI

表 5-7KPI 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
1	絶対位置正確度 誤差の標準偏差	1.75m以内 (地図情報 レベル 2500 程度を想 定)	● 3D 都市モデルをインプット データとして作成したデジ タルツインデータのため、生成 前の各データは引き継いだ状 態でアップデートされている 状態が望ましいため。	● 本システムで作成された データの建築物モデルの 角などの形状変化点等の 明瞭な箇所の座標を真値 との座標比較を実施す る。
2	相対位置正確度 誤差の絶対値の最 大値	0m	● 3D 都市モデルをインプット データとして作成したデジ タルツインデータのため、生成 前の各データは引き継いだ状 態でアップデートされている 状態が望ましいため。	● 明瞭な箇所間の距離を計 測し、その距離を真値と して実施する。

5-3-3. 検証方法と検証シナリオ

本システムで作成されたデータの位置正確度の検証を行った。本検証に先立ち、平面直角座標ではなく本システムで作成されたデータのような任意座標系でモデル作成を行った際に生じると想定される誤差についての検討を行った。

本検証では、原典データである 3D 都市モデルとの比較による絶対位置正確度（水平）と相対位置正確度の検証を行い、3D 都市モデルと本システムで作成されたデータの形状表現が異なる箇所については空中写真図化結果を用いて検証を行った。

5-3-3-a. 水平絶対位置正確度の検証

本システムで作成されたデータは高さ方向に関する絶対位置座標を持たないため、絶対位置正確度の検証は水平方向のみで実施した。検証箇所は検証範囲全体から本システムで作成されたデータの建築物モデルの角や道路端等の明瞭な地点を抜き取り、真値との座標比較をすることで行った。図 5-14 は水平位置正確度の検証をした箇所の例である。本システムで作成されたデータと 3D 都市モデル間に対応する点が明瞭な場合は 3D 都市モデルを真値として扱った。図に示す空中写真（オルソ画像）は取得した地点を示した参考画像である。

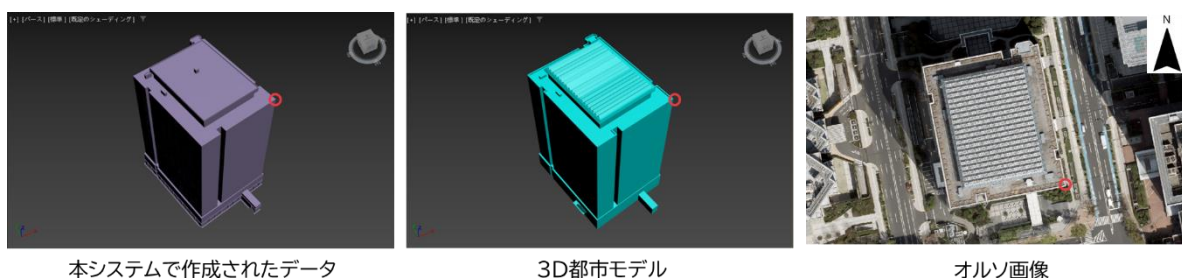


図 5-14 水平絶対位置正確度検証箇所の例（赤丸位置の座標を計測）

5-3-3-b. 相対位置正確度の検証

相対位置正確度は本システムで作成されたデータ及び真値それぞれで明瞭な地点間の距離を計測し、その距離の差を求めることで検証した。図 5-15 に相対位置正確度の検証箇所の例を示す。



図 5-15 相対位置正確度の検証箇所の例（建築物モデル）（赤線部の距離をそれぞれ計測）

5-3-4. 活用技術で想定される誤差

本システムで作成されたデータは平面直角座標系のデータではなく任意座標系のデータであるため、原点地理座標値を用いてシフトを行い、3D 都市モデルとの比較を行う場合に、①真北方向と図面北（平面直角座標系の北軸方向）が異なることによる誤差と②緯度経度を平面投影した際に生じる誤差（縮尺係数による誤差）の2つの誤差が生じる可能性がある。これらは水平方向に影響する誤差要因であり、高さ方向には影響しない。

本検証で用いる本システムで作成されたデータの範囲において生じ得る誤差の量を机上計算によって推定した。その際に設定したデータの範囲は図 5-3 を参照。本システムで作成されたデータは図郭のような明確な矩形の範囲で作成されたものではないため、データの範囲として任意座標原点から本システムで作成されたデータの建築物モデル全てを網羅する最大範囲を設定した。道路モデルは建築物モデルの存在領域よりも外側まで存在するが、その部分では道路形状の作成が部分的であり、抜けている道路形状が多いため、本検証の範囲外とした。

5-3-4-a. 真北方向と図面北方向が異なることによる誤差

任意座標系の軸である真北方向と平面直角座標系の図面北方向には角度差がある。そのため、任意座標をシフトして平面直角座標へ合わせた際に、移動させた地物が持つ座標値に誤差が生じる。その概念図を図 5-16 に示す。今回の実証範囲の新宿付近では、図面北方向に対して時計回り方向に真北がずれる。その結果、平面直角座標上で B'地点の座標が B 地点にずれるような誤差が生じる（今回の範囲では全体的に東・南方向への誤差が生じる）。

真北方向と図面北方向のなす角度（ θ ）は地点により異なるが、本システムで作成されたデータの既知点は原点のみであるため、今回の誤差の大きさの推定においては、原点位置における軸方向の違いによって最大でどの程度の誤差が生じ得るかを計算した。



図 5-16 軸方向の違いによる誤差概念図

原点座標を中心として真北方向を図面北方向に合わせた際に、移動量が最大になる B 地点における誤差を計算した。原点 (0,0) を中心に指定の角度を回転させた際に新たに得られる座標値は以下の数式 1 で計算できる。

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{ 数式 1}$$

XY：回転後の座標値 θ ：回転角度 xy：回転前の座標値

任意座標原点の座標値は緯度経度 (35.6834222,139.6830587) が与えられており、この座標値から国土地理院のホームページ「平面直角座標への換算 (<https://vldb.gsi.go.jp/sokuchi/surveycalc/surveycalc/bl2xyf.html>)」を用いて換算した平面直角座標 (9 系) 値 (数学座標系) は (X,Y) = (-13601.9055, -35112.3112) である。また、同ホームページにより任意座標の原点位置における真北方向角 (真北方向と図面北方向のなす角) を得

た。今回の計算における回転角は真北方向角の負数となる。

【計算に使用した値】

- 回転角=-0.087655556 度
- 原点を (0,0) とした際の B 地点 (回転前) の座標値 (x,y) : (1410.000, 1290.000)

この値を用いて数式 1 により B' の座標を計算すると、B' 地点 (回転後) の座標 = (1411.972, 1287.841) が得られる。

回転前 (B) と回転後 (B') の座標から誤差量を算出すると、誤差=2.924m が得られた。この検討結果から、本検証範囲においては北軸方向の違いにより、最大で 2.924m の絶対座標誤差が東・南方向に生じると推定された。

5-3-4-b. 平面投影の際に生じる誤差

緯度経度のように準拋楕円体上に表現される座標を任意座標平面に投影する際は、原点からの距離に応じて楕円体上と平面上との間に誤差（縮尺係数による誤差）が生じる。この誤差を補正するための補正量が縮尺係数である（図 5-17）。任意座標で縮尺係数の補正をしていない場合、平面直角座標系で表されたデータと比較する際に誤差が生じる。今回の実証範囲において縮尺係数補正の有無により生じ得る最大の誤差量を計算した。縮尺係数による誤差量は距離に比例するため、原点と最も遠い地点（B 地点）の距離で計算を行った。縮尺係数の補正をした距離は楕円体上の距離（任意座標上の距離）に 2 地点の縮尺係数の平均をかけることで算出できる。縮尺係数は国土地理院のホームページ「平面直角座標への換算」から得た。

【計算に使用した値】

- 原点位置の縮尺係数：0.99990228
- B 地点の縮尺係数：0.99990183
- 任意座標上での原点と B 地点の距離：1911.073m

この値を用いて縮尺係数補正後の距離を求めると

$$1911.073 \times ((0.99990228 + 0.99990183) \div 2) = 1910.886$$

が得られた。補正前と補正後の距離の差から、B 地点における縮尺係数の補正の有無による誤差量は 0.187m であった。この縮尺係数による誤差の方向は 2 点を結んだ直線方向に生じる。

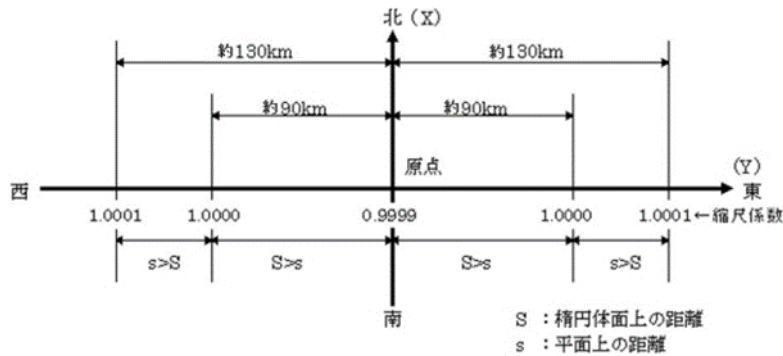


図 5-17 平面直角座標での縮尺係数

(引用：国土地理院ホームページ <https://www.gsi.go.jp/sokuchikijun/datum-main.html>)

5-3-4-c. 想定される誤差が相対距離に与える影響

2つの誤差の要因それぞれが相対距離に与え得る影響を検討した。仮に50mの距離を設定した場合、真北方向と図面北が異なることによる誤差は回転によるものであるため、両地点が同様に座標移動することを考慮すると、50m程度の相対距離に影響は生じない。縮尺係数の誤差に関しては、55-3-4-b同様に計算を行った結果、50mの距離において約5mmの誤差が生じるという結果が得られた。

5-3-5. 検証結果

水平位置正確度に関しては事前に検討していた誤差（北軸による誤差及び縮尺係数による誤差）よりも大きい誤差が得られた。また、誤差の方向に関しても事前検討による誤差の想定と異なる方向に誤差が生じていた。一部を除き、全体的に一様な方向の誤差が生じていたため、原点位置の座標自体に誤差がある可能性も想定されるが、それを解消した際に更に誤差が大きくなる地物も存在する。特に、絶対位置正確度に関しては全体的に一様な誤差量ではなく地物ごとに誤差量が大きく異なっていたためデータセット内部での建築物モデル同士の重なりのような不整合が生じていた。

表 5-8 検証結果サマリ

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

検証内容	評価指標・KPI	目標値	結果		示唆
			項目	評価値(m)	
水平位置正確度 (建築物：69地点)	誤差の標準偏差	1.75m 以内 (地図情報レベル 2500 程度を想定)	最大誤差	34.998	● 誤差量にばらつきが大きく、誤差方向も一定では無かった。そのため、一部の建築物の重なりが生じる箇所が存在した。
			最小誤差	2.734	
			平均	5.531	
			標準偏差	3.922	
相対位置正確度 (建築物：123測線)	誤差の絶対値の最大値	0m	最大	4.293	● 建築物においては水平方向よりも高さ方向における誤差が大きかった。
			最小	-5.385	

			平均	-0.022	
			標準偏差	0.921	

検証を平面直角座標（9系）で行うため、本システムで作成されたデータの原点座標として与えられた（緯度，経度） = （35.6834222,139.6830587）を国土地理院ホームページ「平面直角座標への換算」を用いて平面直角座標（9系）の座標値（X,Y） = （-35112.3112, -13601.9055）に変換した（図 5-18）。この測量座標系である値をCADやGISで扱う数学座標系にするため、X,Yの座標値を置き換え、（X,Y）=（-13601.9055, -35112.3112）とした。この値を、任意座標原点の座標値として使用した。



図 5-18 平面直角座標への換算（引用：国土地理院ホームページ）

任意座標原点位置を算出した座標値に合わせるために、本システムで作成されたデータ全体を X 方向に-13601.9055m、Y方向に-35112.3112m シフトした。シフト後の本システムで作成されたデータと空中写真から作成したオルソ画像を図 5-19 に示す。この際、55-3-4 で検討した誤差の補正は行わず、水平シフトのみを実施した。

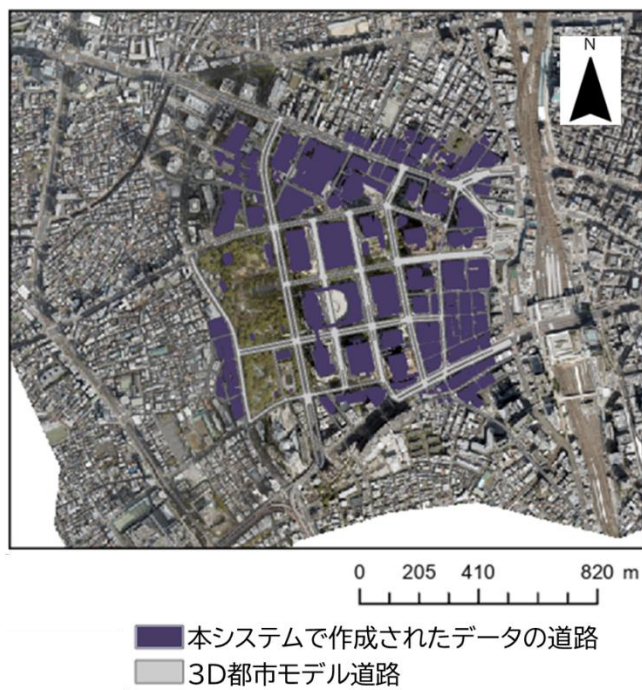


図 5-19 シフト後の本システムで作成されたデータ

5-3-5-a. 水平絶対位置正確度の検証

本システムで作成されたデータの建築物モデルの屋根の角のような形状に特徴のある箇所を 69 地点抜き取り、3D 都市モデル又は空中写真図化結果を真値として誤差量を求めた結果を表 5-9 建築物モデル水平絶対位置正確度の検証結果に示す。検証結果に誤差 34.998m と非常に大きい値が有り、統計量の計算における影響を鑑みて、その誤差を除いた場合についても同様に統計量を計算した。また、本システムで作成されたデータの原点からの距離と誤差の量の関係を図 5-20 に示す。原点から離れるほど誤差が大きくなる相関が多少見られた。

表 5-9 建築物モデル水平絶対位置正確度の検証結果

指標	値 (m)	最大誤差を除いた値 (m)
最大誤差	34.998	12.593
最小誤差	2.734	2.734
平均	5.531	5.085
標準偏差	3.922	1.503

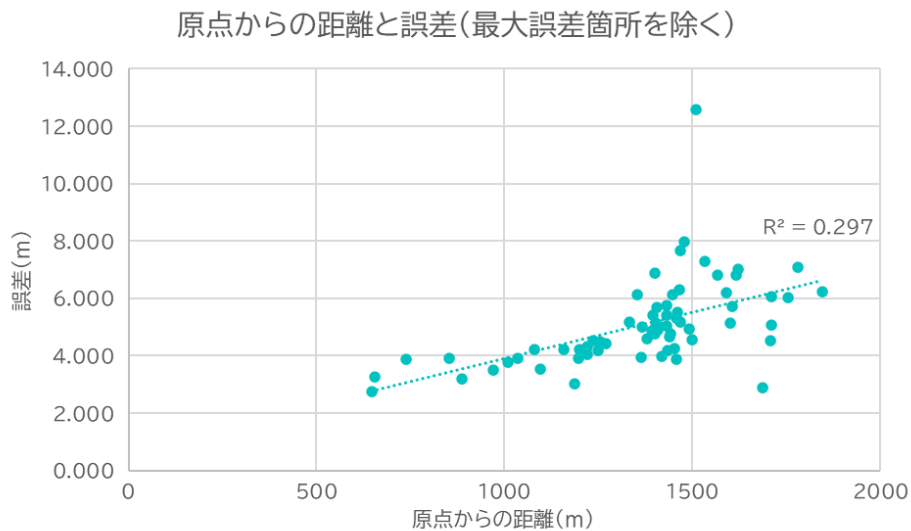
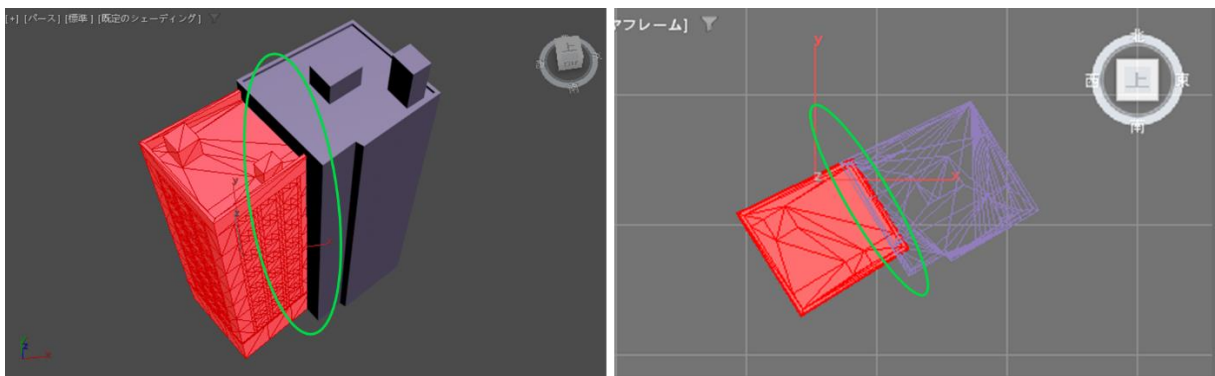


図 5-20 原点からの距離と水平絶対位置正確度の誤差の関係

水平絶対位置正確度の誤差方向と大きさをオルソ画像上に表示したものを図 5-21 に示す。全体的に北西方向への誤差傾向があった。しかし、図中の赤丸の箇所のように周囲の誤差傾向とは異なる箇所が存在するため、全体として同様のずれが生じていたわけではない。また、誤差量は一定ではなかつきが大きかった。その影響により、本システムで作成されたデータには図 5-22 に示すように建築物モデル同士が接触している箇所が生じていた。



図 5-21 水平位置の誤差方向



5-3-5-b. 相対位置正確度の検証

本システムで作成されたデータの建築物モデルの屋根部の高さや屋根形状の長さなどの特徴のある箇所を123 か所抜き取って距離を測定し、3D 都市モデル又は空中写真図化結果を真値として相対位置の誤差量を求めた結果をエラー! 参照元が見つかりません。に示す。そして図 5-23 に実際の距離（真値）と誤差の量の関係を示す。建築物モデルの相対位置について、大きな誤差は高さ方向に生じることが多かった。相対位置の誤差については真値との誤差が 0m であるものから約 5m の誤差を持つものまで様々であり、傾向があまり見られなかった。

表 5-10 建築物相対位置正確度の検証結果

指標	SPD-真値で求めた誤差量 (m)		
	3D 距離	2D 距離	Z 距離
最大値	4.293	1.081	4.293
最小値	-5.385	-1.028	-5.385
平均値	-0.022	0.028	-0.068
標準偏差	0.921	0.225	0.952

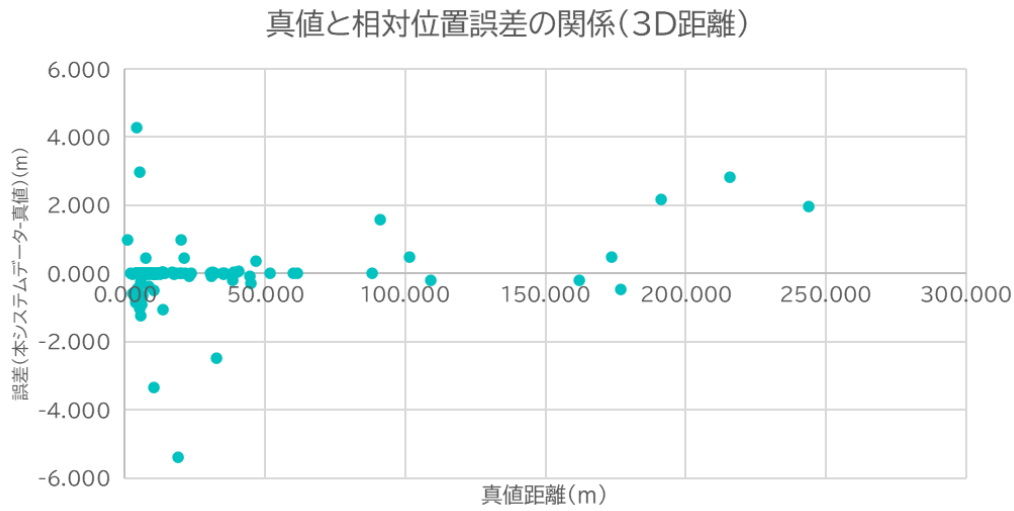


図 5-23 真値と相対位置誤差量の関係（建築物）

- 目標値⇒「100%」
 - 目標値の設定理由⇒3D 都市モデルをインプットデータとして作成したデジタルツインデータのため、生成前の各データは引き継いだ状態でアップデートされている状態が望ましいため。
- 目標値⇒「0m」
 - 目標値の設定理由⇒3D 都市モデルをインプットデータとして作成したデジタルツインデータのため、生成前の各データは引き継いだ状態でアップデートされている状態が望ましいため。

6. BtoB ビジネスでの有用性検証

6-1. 検証目的

検証仮説に基づき、以下の検証目的を設定する。

【検証仮説】

3D 都市モデルを利用したコンシューマ向けのコンテンツ制作を促進するためには、ハイクオリティのテクスチャ付与や「小物類」（道路マーキングや植栽、信号機等の設備等）を追加した高品質デジタルツインデータが必要である。

- システムのユーザビリティ検証
 - 本システムにおける UI/UX の使い勝手を確認する

6-2. 検証方法

クリエイター・サービス利用者双方に対して定量・定性評価を実施し、検証する。

定量評価では、ダウンロード数、閲覧数、アクセス数、会員登録者数を取得し、定性評価ではアンケートを実施した。

（ヒアリング・アンケートの項目については「6-4.検証結果」で記載）

6-3. 被験者

本ユースケースでは、生成した都市 3D データを一般に公開し、映像クリエイター等の 3D データを扱えるエンジニアやクリエイターをターゲットとし、アンケートを行い本システムの有用性を検証する。

また、Fortnite で公開し、ゲームを遊んだユーザー人数なども利用し有用性の検証を行う。

表 6-1 被験者リスト

年齢	性別	種別	特徴	人数（目標値）
全年齢	全性別	クリエイター	3D データを触ることのできるエンジニア・クリエイター	50 名 ※今までのサービスローンチ時の利用者から最低限確保可能な人数を想定
全年齢	全性別	ゲームプレイヤー	Fortnite 等でゲームを遊ぶプレイヤー	5,000 名

6-4. ヒアリング・アンケートの詳細

6-4-1. 検証項目と評価方法

クリエイターとゲームプレイヤーをそれぞれ被験者とし、定量・定性的に評価した。

表 6-2 検証項目と評価方法

No.	被験者	KPI	定量評価	定性評価
1	クリエイター	配布データの取得方法	<ul style="list-style-type: none"> ● アンケートを実施 ● 選択肢は「とても満足」「満足」「どちらでもない」「不満」「とても不満」 ● 回答を集計し、各選択肢の選択率から評価（各設問で、過半数の「満足」以上回答を目標とする） 	● アンケートに自由記入欄を設定
2		配布データの精度		
3		配布データの拡張性		
4		配布データの容量		
5	クリエイター	無償配布後1か月のダウンロード数	ダウンロード数	なし
6		無償配布後1か月の閲覧数・アクセス数・会員登録者数	閲覧数・アクセス数・会員登録者数	なし
7	ゲームプレイヤー	ゲームとしてのクオリティ（没入度・熱中度）	Fortnite等の外部PFでデモ公開後1か月のプレイ人数とプレイ時間を測定	なし

6-4-2. 実証実験の様子

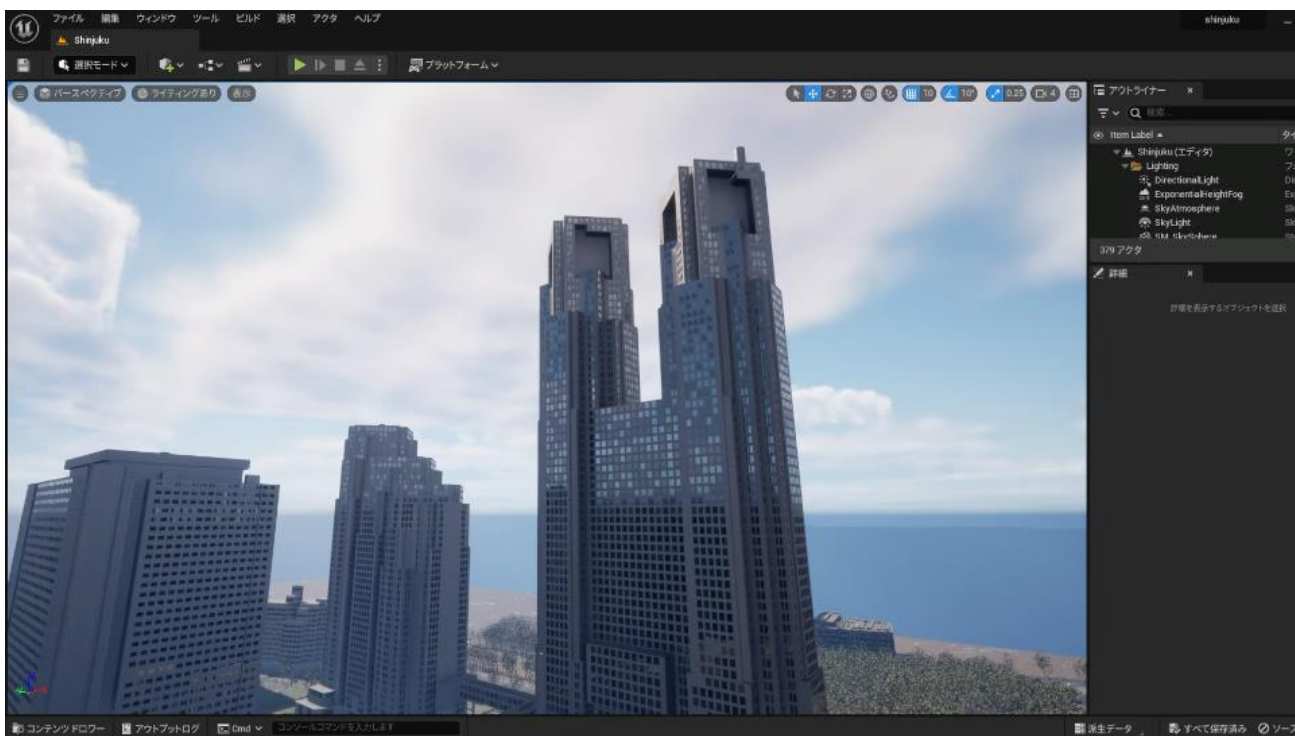


図 6-1 ゲーム開発に利用できるように、仕様に合わせてテクスチャ情報の軽量化等を行う

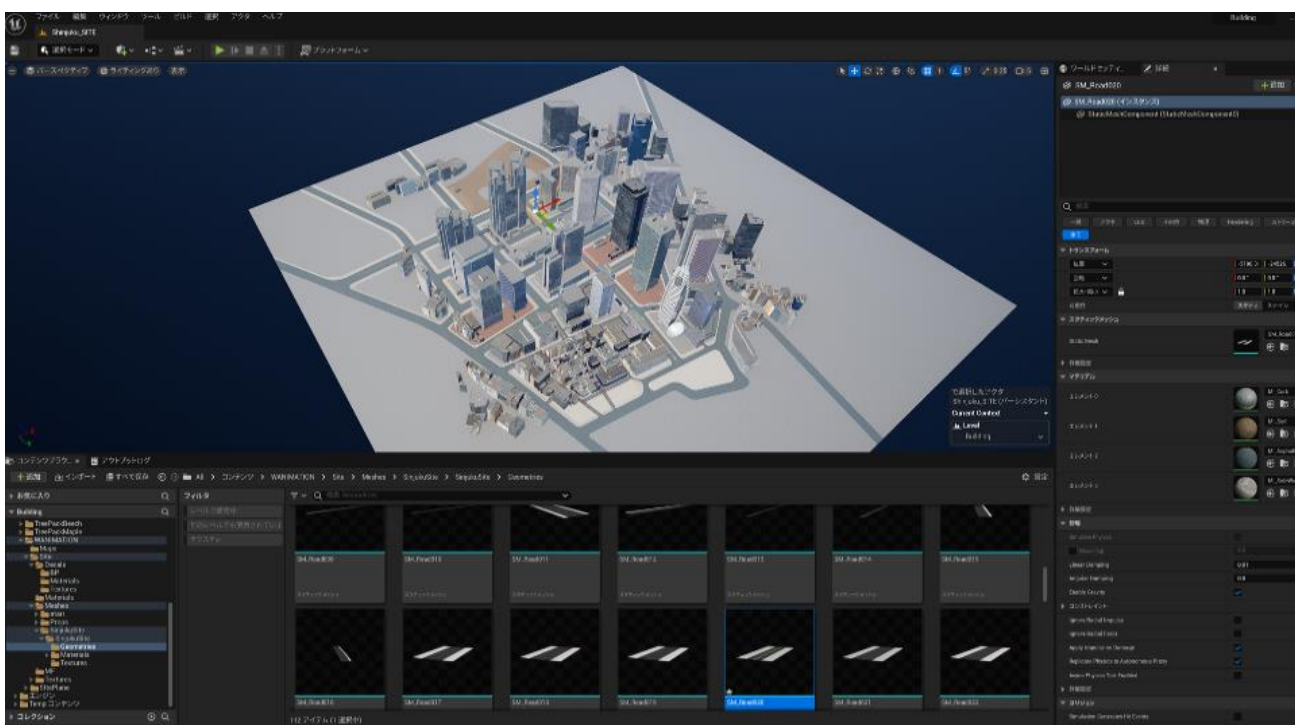


図 6-2 ゲームプラットフォーム向けに読み込み

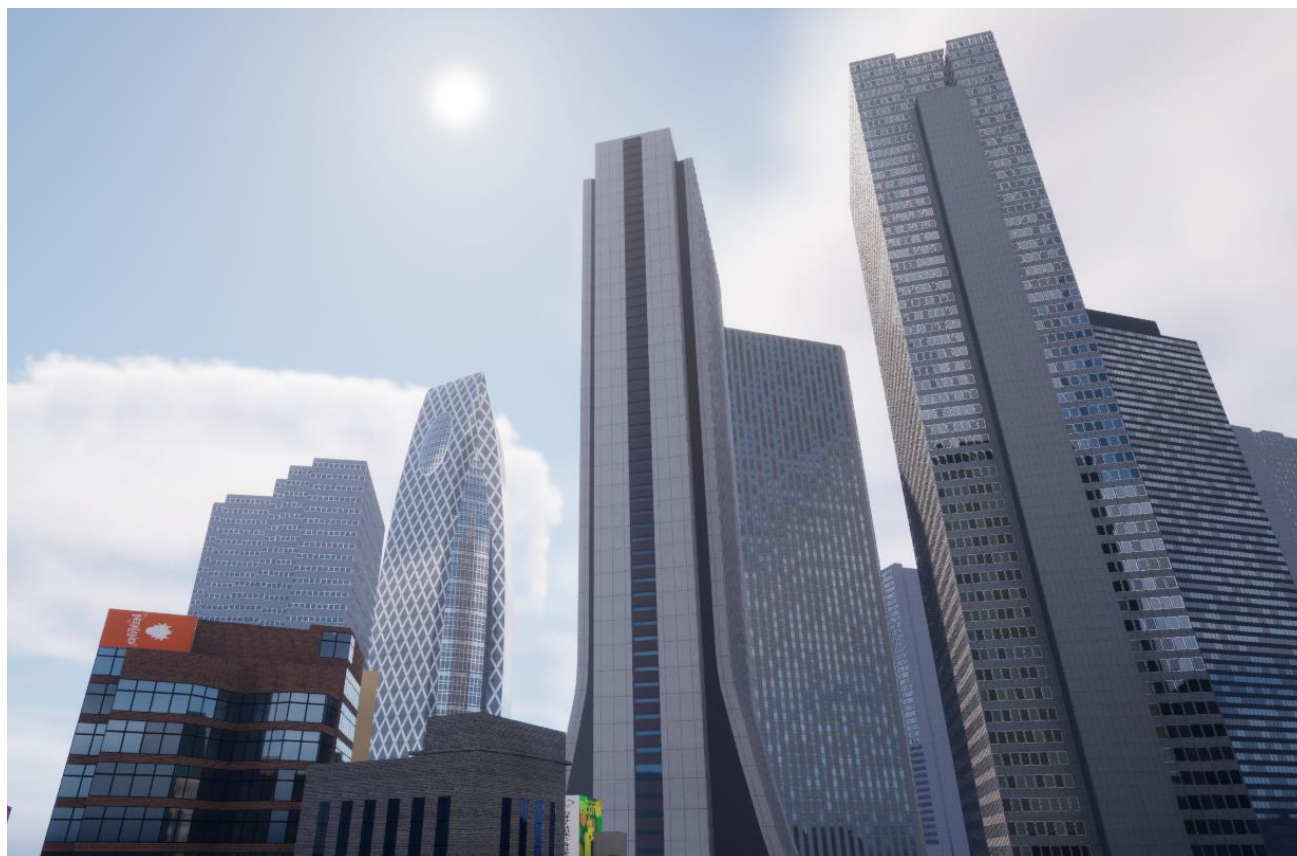


図 6-3 Unreal Engine に読み込んだ高精度デジタルツインデータ①

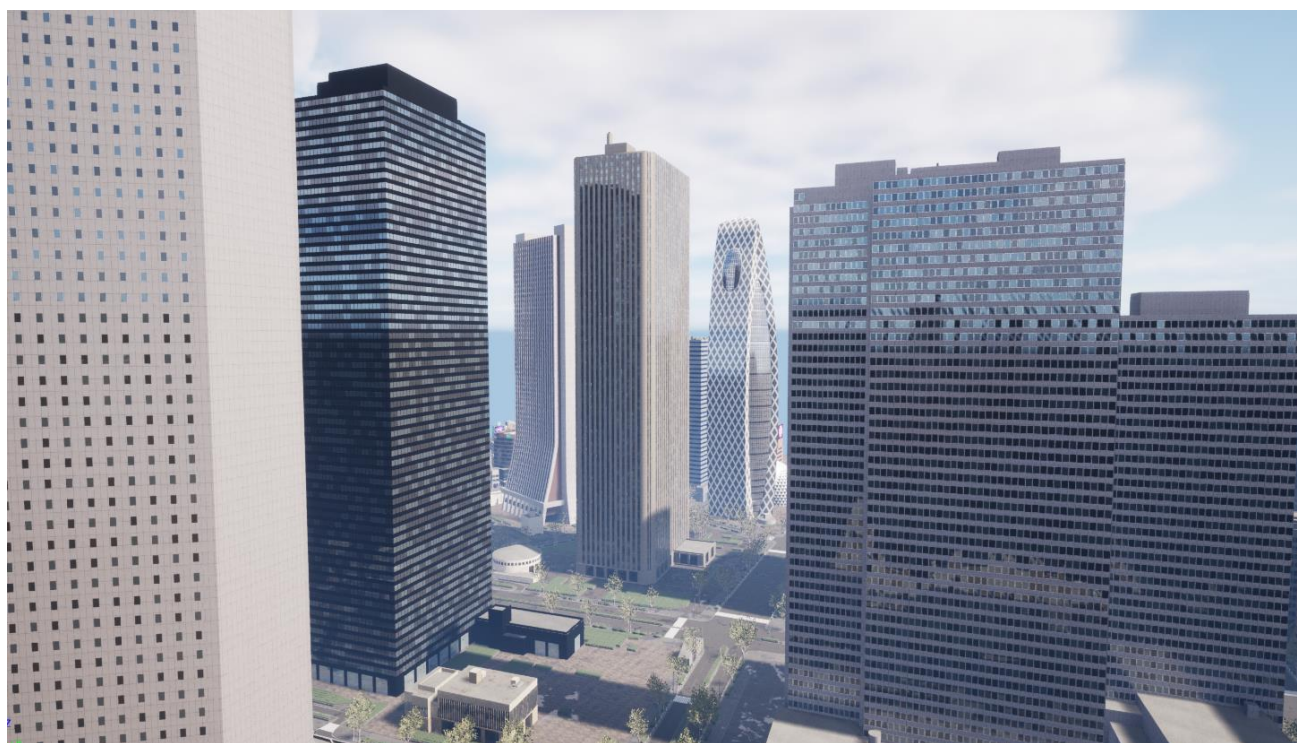


図 6-4 Unreal Engine に読み込んだ高精度デジタルツインデータ②



図 6-5 Unreal Engine に読み込んだ高精度デジタルツインデータ③



図 6-6 Unreal Engine に読み込んだ高精度デジタルツインデータ④



図 6-7 Unreal Engine に読み込んだ高精度デジタルツインデータ⑤



図 6-8 スペースデータ社が制作したゲームコンテンツ



図 6-9 配布データよりクリエイターが制作したゲーム例①



図 6-10 配布データよりクリエイターが制作したゲーム例②



図 6-11 配布データよりクリエイターが制作したゲーム③



図 6-12 配布データよりクリエイターが制作したゲーム④

※クリエイターが制作したゲームについて

- ① TOKYO GRAVITY PARKOUR (<https://www.fortnite.com/@perry-1115/3991-8064-8184>)
- ② ミラージュ・ハント in 新宿 (<https://www.fortnite.com/@yakimaru/7132-1928-6318>)
- ③ SHADOW FRIENDS (<https://www.fortnite.com/creative/island-codes/4173-0827-7507>)
- ④ MiniMiniGames (<https://www.fortnite.com/@irobuki/0682-5774-8867>)

6-5. 検証結果

データの利用状況については、公開後約 1 か月間の目標利用人数を過去の同様のサービスを参考にクリエイターが 50 人、サービス利用者を 5,000 人と定めていたところ、クリエイター向けのデータは約 4,000 人（1 人で複数回の利用があり、回数としては約 6,000 回）のダウンロード、サービス利用者は約 65,000 人であった。このような活発なデータやゲームの利用状況から、フォトリアルな都市デジタルツインデータに対する期待と興味が高まっていることが改めて明らかになった。

他方で、アンケート結果から作成したデータの精細度や容量など、今後の開発方針に対する課題と対策が明らかになった。まず、データの精細度については、現状の 3D 都市モデルや他の 3D 都市データと比較して、建築物のテクスチャや都市設備の付加の面では高い評価を得た一方で、ローポリゴン化の一環で軽量化することによる景観劣化やテクスチャの歪みに関して改善を求める意見があった。この劣化や歪みは、データベースに取り込んだ教師データにおいて、影などの影響により現状の画像処理のアルゴリズムでは色や構造を認識することが難しい建築物が存在したことによるものであり、今後は教師データを増やすとともに、事前の機械学習処理によって画像の歪みや影の削除などを実施することで予測精度を向上させる必要がある。

次に、データ容量という観点では、今回の実証実験の対象エリアである西新宿エリアの約 2 km 四方を 1 つのデータとして配布したことで、ローポリゴン化の処理を行ったものの、容量の大きいデータとなった。その結果、本格的な映像制作やゲーム開発を行うクリエイターが持つハイエンドな PC であれば問題なく動作するが、一般的なゲーミング PC では動作の遅延が見られるというコメントがあった。この点については、精度の劣化につながるデータ自体の軽量化ではなく、ダウンロードの際に対象エリアを任意に分割可能な UX/UI とすることや、同一のデータ内でもジオメトリを分離して建築物を設置することが必要である。前者については、クリエイターによる利用範囲の指定やダウンロード容量の表示、プレビュー等の画面を整備することで、ダウンロードの利便性を向上させる方法である。後者は、現在のプロジェクトファイルでは建築物などの地物が 1 つのオブジェクトとして固定されているのに対し、生成時にオブジェクトを分離する処理を追加し、特定のオブジェクトの削除や地物、都市設備を指定したダウンロードが可能な仕様にするすることで、より柔軟なデータ利活用を可能にする方法である。

上記の検証結果に加えて、今後の拡張性としては今回の実証実験の対象エリアにおける今後の都市開発による変化への対応や、新しいエリアにおけるデジタルツインの構築に必要な教師データを整備する工数が課題である。3D 都市モデルの更新に対してシステムが追随するような仕様への変更や、FBX 形式ではなく CityGML 形式のデータをそのまま扱えるようにするなど、データ取得や更新部分をプログラム制御によって自動化することで対応範囲を広げていくことが求められる。

1) クリエイター向けアンケート結果 (N=49)

(ア) 配布データの取得方法についての評価

全参加者のうち、50%以上 (28/49 名) が「とても満足」か「満足」と回答。

今回のデータ配布・アンケート対象としては 3D コンテンツを扱うクリエイター向けのため、全体的にネットリテラシーが高い人が多い。そういった人達でさえ、半数近くが満足できていなかったため、今後は取得に関する一連の流れや、実際にデータを展開するためのドキュメントなどを充実させていく必要があると考えられる結果となった。

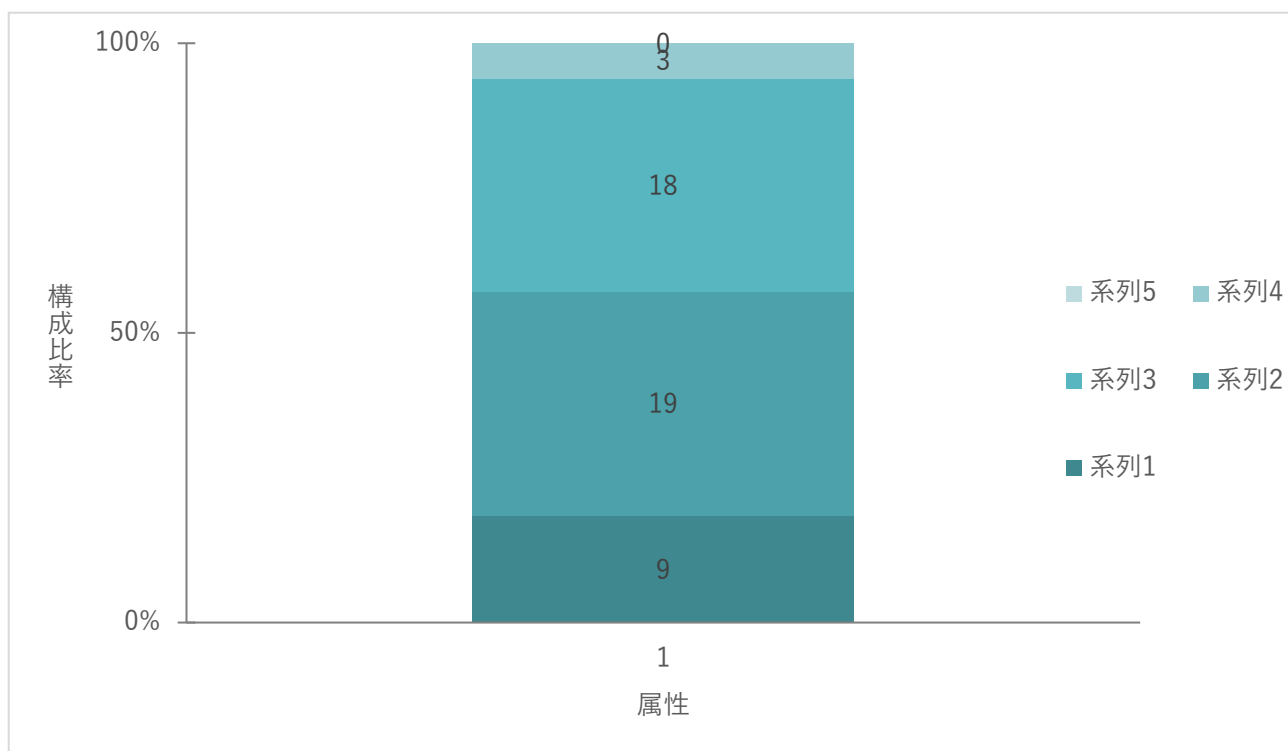


図 6-13 配布データの取得方法についての評価

定性評価では、以下のように一定評価があるものの、次年度に向けた課題となるフィードバックが寄せられた。

表 6-3 関連する定性コメント

No.	定量調査の結果・示唆	関連する定性コメント (例)
1	とても満足	● シンプルにダウンロードして、すぐ使える形式でデータが配布されており、ゲームエンジンの使用経験が浅くても問題なく確認できた
2	満足	● セキュリティも考慮されたインターフェースとなっている。ダウンロードまでのインターフェースも使いやすい。
3	どちらでもない	● 全体の取り込みにかかなりの時間掛かるため満足な速度とは言えないが、内容を考えると仕方ないと思う

4	不満	<ul style="list-style-type: none"> ● なんとか UE に取り込めたが、インポートの方法が初めて4か月ほどの私には理解が難しかった。Unreal Engine for Fortnite の移行もできたはできたが、容量が重くまだ軽くする術を知らないため Map 公開できていない
---	----	--

(イ) 配布データの精度についての評価

全参加者のうち、70%以上（35/49名）が「とても満足」か「満足」と回答。現状の3D都市モデルや他の3D都市データと比較して、建築物のテクスチャや都市設備の付加の面では高い評価を得た一方で、ローポリゴン化の一環で軽量化することによる景観劣化やテクスチャの歪みに関して改善を求める意見があった。この劣化や歪みは、データベースに取り込んだ教師データにおいて、影などの影響により現状の画像処理のアルゴリズムでは色や構造を認識することが難しい建築物が存在したことによるものであり、今後は教師データを増やすとともに、事前の機械学習処理によって画像の歪みや影の削除などを実施することで予測精度を向上させる必要がある。

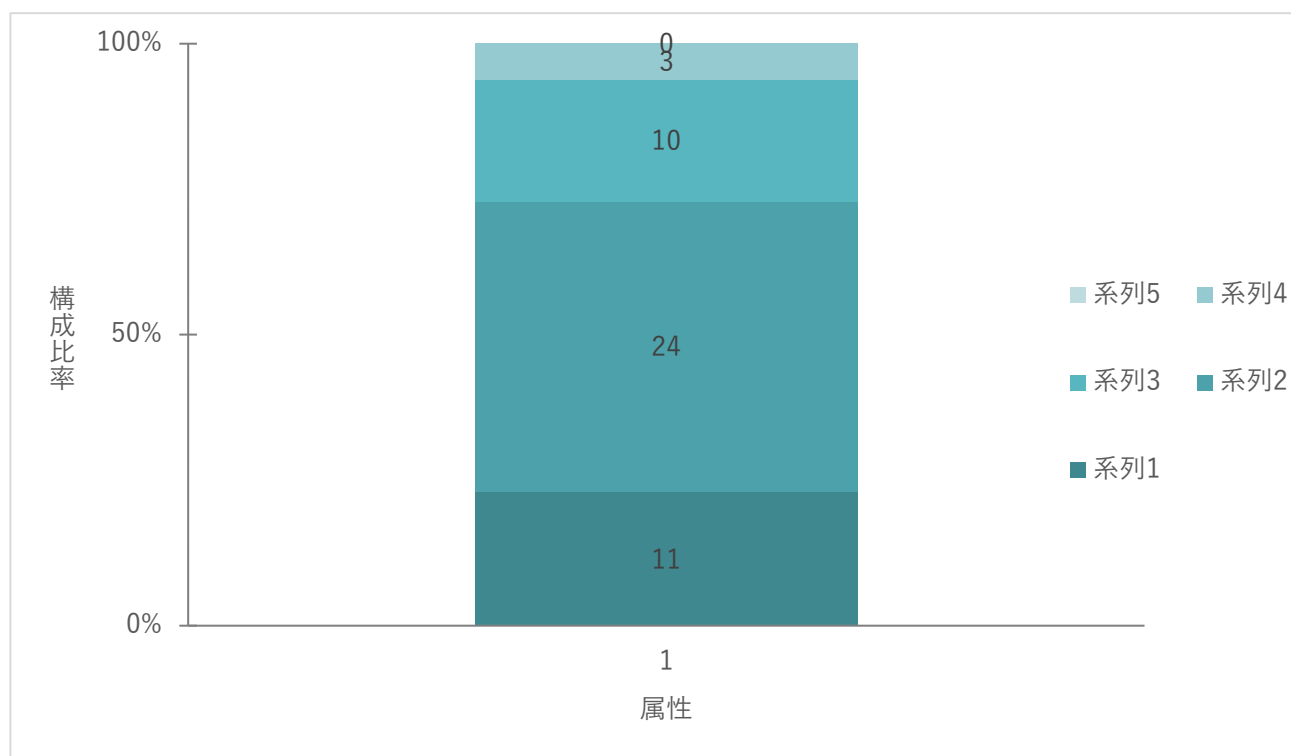


図 6-14 配布データの精度についての評価

表 6-4 関連する定性コメント

No.	定量調査の結果・示唆	関連する定性コメント（例）
1	とても満足	<ul style="list-style-type: none"> データの精度は PLATEAU や Google Photorealistic 3D Tiles で課題に感じていたデータ精度を解決しているので、とても良いと思った
2	満足	<ul style="list-style-type: none"> 配布されたデータではところどころに景観劣化が見られる。しかし、従来の PLATEAU や Google Earth、Cesium よりも景観劣化が少なく、誰が見てもその場所であることがわかるため非常に助かっている
3	どちらでもない	<ul style="list-style-type: none"> どうしてもテクスチャが歪むので難しいとは思いますが、精度がもっと上がっていくのを期待している。AI で歪み補正と一緒に修正するようになると思っている
4	不満	<ul style="list-style-type: none"> 遠くから見る分には綺麗ですが、よく見ると壁の UV が乱れていたり、看板が他のオブジェクトにめり込んでいたり、異常に小さい家屋が出現していたりするなど、商用ゲームに使用できるレベル・人間のモデラーがモデリングしたレベルにはなっていないと思う

(ウ) 配布データの拡張性についての評価

全参加者のうち、約 55% (27/49 名) が「どちらでもない」か「不満」と回答。今回の実証実験の対象エリアにおける今後の都市開発による変化への対応や新しいエリアにおけるデジタルツインの構築に必要な教師データを整備する工数が課題である。3D 都市モデルの更新に対してシステムが追随するような仕様への変更や、FBX 形式ではなく CityGML 形式のデータをそのまま扱えるようにするなど、データ取得や更新部分をプログラム制御によって自動化することで対応範囲を広げていくことが求められる。

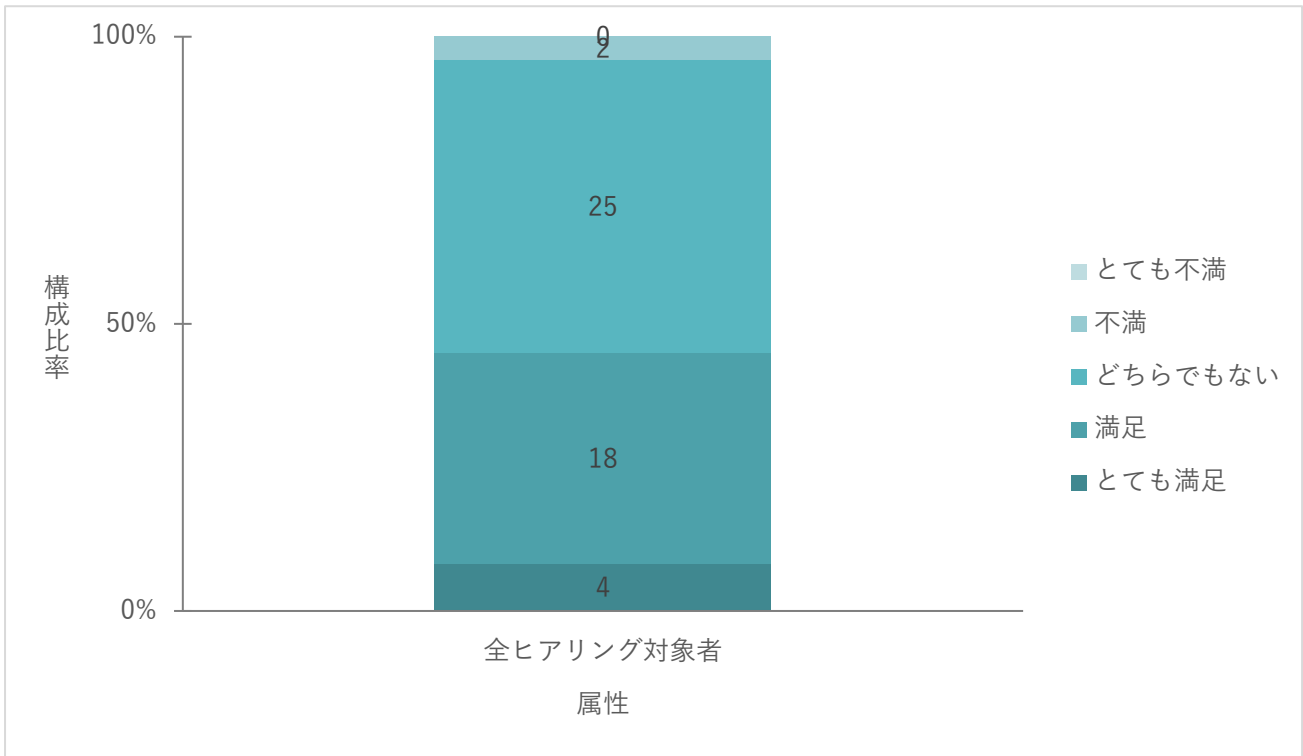


図 6-15 配布データの拡張性についての評価

表 6-5 関連する定性コメント

No.	定量調査の結果・示唆	関連する定性コメント (例)
1	とても満足	● Fortnite 上におけるメタバース空間の広がりを実感できた
2	満足	● モデル一つ一つの規模がとても巨大なので、LOD でしっかり分割していけば拡張性はあると思う
3	どちらでもない	● 配布データ形式が UE のプロジェクトだったので、まだ今の段階では何とも言えない
4	不満	● 道路などが全て繋がって配置されているので一部だけを利用したい場合には編集が難しかった

(エ) 配布データの容量についての評価

全参加者のうち、約 50% (25/49 名) が「とても満足」か「満足」と回答。本格的な映像制作やゲーム開発を行うクリエイターが持つハイエンドな PC であれば問題なく動作するが、通常の PC では動作の遅延が見られるというコメントがあった。この点については、精度の劣化につながるデータ自体の軽量化ではなく、ダウンロードの際に対象エリアを任意に分割可能な UX/UI とすることや、同一のデータ内でもジオメトリを分離して建築物を設置することが必要である。

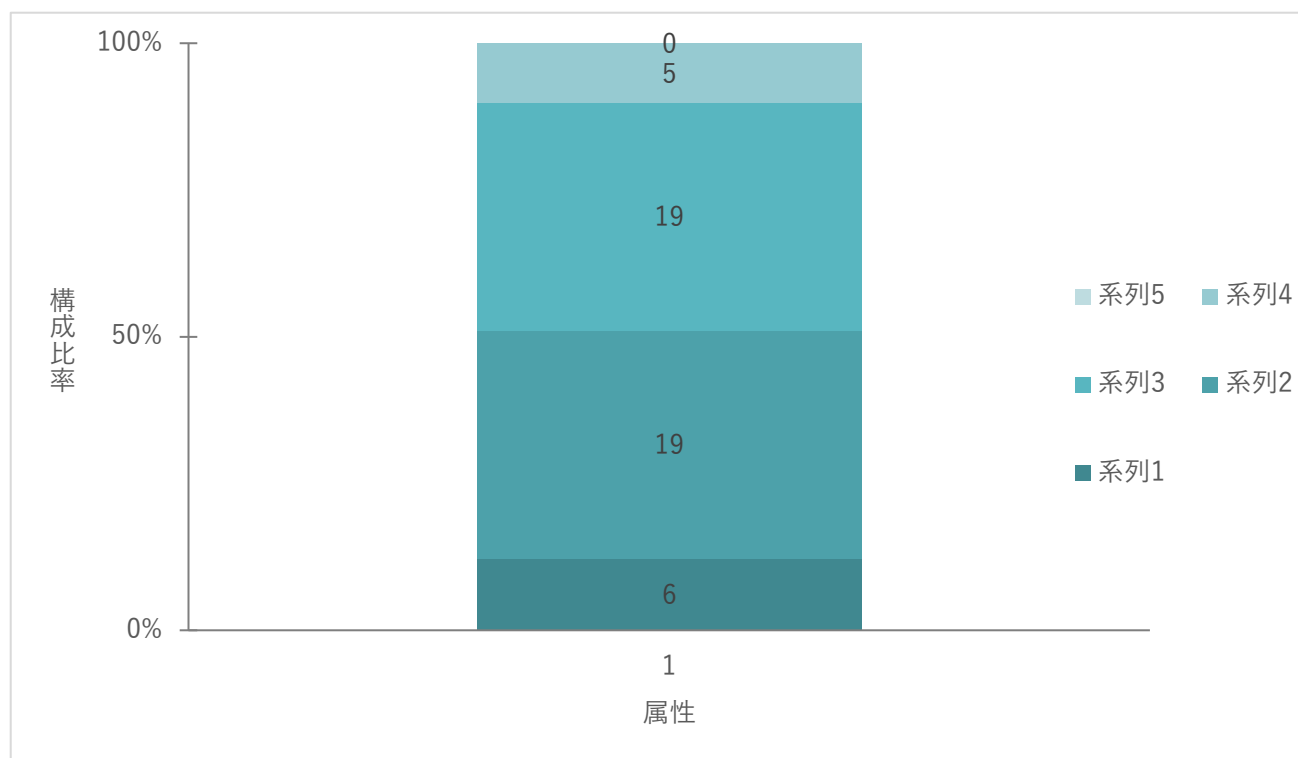


表 6-16 配布データの容量についての評価

表 6-6 関連する定性コメント

No.	定量調査の結果・示唆	関連する定性コメント (例)
1	とても満足	● 思っていたよりもかなり少ない容量に驚いている
2	満足	● データのクオリティに対して容量が少ないため、使いやすい仕様になっている。高精度版、通常版等で分けた仕様でも利用者のニーズに合わせられると思う
3	どちらでもない	● もっと細かく分割して読み込める方が用途に適したプロジェクトサイズにできる
4	不満	● 一般的な PC では動作が重い

2) サービス利用人数

- ゲームプレイヤー（Fortnite）のプレイ人数とプレイ時間（モデル公開後 1 か月）
 - プレイ人：61,023 人
 - プレイ時間：16,272 時間（プレイ時間平均 15～16 分/人）
- UE データ配布サイトの登録人数とダウンロード数（公開後 1 か月）
 - 登録人数：3,608 人
 - ダウンロード数：5231 回

7. 成果と課題

7-1. 本実証で得られた成果

7-1-1. 3D 都市モデルの技術面での優位性

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 7-1 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	テクスチャ画像による構造把握	● 航空写真や衛星データでは確認できない、建物側面の情報を 3D モデルのテクスチャ画像から情報を取得できるようになり、精度を上げることが可能
	特殊形状のランドマーク建築物の表現	● 都庁やコクーンタワーといった、形状が特殊で単純な立方体の集合では表現しにくい建築物についても、3D モデルが公開されているため、よりリアルな 3D モデルを生成することが可能
	建物情報	● CityGML で建物の高さや利用用途等が把握できるため、地図情報よりも機械学習における建物分類において精度を上げることが可能

7-1-2. 3D 都市モデルのビジネス面での優位性

表 7-2 3D 都市モデルのビジネス面での優位性

大項目	小項目	3D 都市モデルのビジネス面での優位性
サービス提供価値の向上	精度の高度化	● 3D 都市モデル自体のテクチャの高解像度化や屋上構造物及び都市設備等のデータがどの都市でも詳細に整備され、そのデータを活用した高精度デジタルツインデータを生成することが出来るようになれば、より位置精度及び外観が現実世界のまちに近似した 3D コンテンツとして作成することが出来るようになり、3DCG を必要とするサービスへの提供価値を向上することが可能
サービス開発期間・コストの削減	開発工数の削減	● 3D 都市モデルを利用することで、モデリングの工程を簡素化することが可能になり、スタートアップのような小規模な企業でも高品質な 3D モデルを作成することが可能
	オープンデータによる開発・運用コスト	● 3D 都市モデルは公的なオープンデータとして整備されているため、利用に関する法的な権利やライセンスが明確に

	削減	なっており、法的な不確実性を抱えることなく安心してデータを活用することが可能
--	----	--

7-2. 事業化に向けた課題と解決策の案

表 7-3 実証実験で得られた課題

大項目	小項目	実証実験で得られた課題	課題に対する対応策
システム (機能)	拡張性	<ul style="list-style-type: none"> ● 現在のプロジェクトファイルでは設置物が一つのオブジェクトとして固定されているため、特定のオブジェクトだけ取り除くことが困難 	<ul style="list-style-type: none"> ● 生成時にオブジェクトの分離をすることで、より柔軟に街の一部のみ利用することができるように仕様を変更する
	更新	<ul style="list-style-type: none"> ● 3D 都市モデルの更新に対して手動でのデータ取り込みが必要なため、工数負荷が大きい 	<ul style="list-style-type: none"> ● 3D 都市モデルの更新があったタイミングで適時システムを連携可能な API を準備し、自社が持つ地図データの更新を自動化できるようにする
	利用可能ソフトウェア	<ul style="list-style-type: none"> ● 現在は Unreal Engine 5.1 のみでしか利用ができないため、最新の Unreal Engine 環境や Unity 等の別のツールへの対応が必要 	<ul style="list-style-type: none"> ● より色々なソフトウェアで利用可能にするため、ゲームエンジンのプロジェクト形式での配布だけではなく、アセットとして配布するなど、それぞれが利用したい環境にインポートできるようにデータを配布する
システム (UI・UX)	ダウンロード方法	<ul style="list-style-type: none"> ● ダウンロードの画面が単純なマップ上で対象エリアを選択する仕様なため、対応エリアが増えた場合にユーザーが希望のエリアを選択することが困難 	<ul style="list-style-type: none"> ● 配布サイトでの検索機能や一覧機能、また、ダウンロード前のプレビュー画面等でユーザーが希望するデータをダウンロードしやすい UI・UX に変更する
アルゴリズム	予測精度向上	<ul style="list-style-type: none"> ● 建物によってテクスチャを用いても構造や色を把握することが困難 	<ul style="list-style-type: none"> ● 機械学習のデータの増加や、事前処理としてブレや歪みを修正や、影を取り除く等の機械学習処理を施して精度を向上する
	自動化	<ul style="list-style-type: none"> ● 手動でデータを更新している部分も多く、新しいエリアに対応するためには工数負荷が大きい 	<ul style="list-style-type: none"> ● データベースを用いて 3D 都市モデルの CityGML データを扱えるようにするなど、データの取得や更新部分をプログラムで制御し、自動更新可能な範囲を拡大する

サービス運用	ビジネスモデルの改善	<ul style="list-style-type: none">● 現在は Fortnite のみの対応であるため、収益機会が少ない	<ul style="list-style-type: none">● コミュニティを作るなど、利用者やプロジェクトの支援者を増やす施策を講じ、新しいマネタイズ方法を模索していく
	利用可能範囲の拡大	<ul style="list-style-type: none">● 現在新宿エリアのみのため、利用範囲が限られている	<ul style="list-style-type: none">● アルゴリズムやシステムの改変によって効率的に作成可能な仕様を整え、3D 都市モデルデータが充実しているエリアから順に対応エリアを拡充する
	ダウンロード	<ul style="list-style-type: none">● 3D データで容量が膨大になりダウンロードのためのデータ保持やダウンロード帯域にもコスト負荷が大きい	<ul style="list-style-type: none">● データの保管場所の検討やコンテンツデリバリネットワークを用いることでダウンロードの効率化とコスト低減を目指す

7-3. 今後の展望

今回の実証実験では、3D 都市モデルを教師データとして機械学習させた「高精度デジタルツインデータ自動生成 AI」を開発した。また、本システムにより構築した西新宿エリアのデジタルツインデータをコンシューマ向けのコンテンツとして配布すると多くの反響があり、この分野に対する期待と興味の高まりを改めて認識する機会となった。さらに、アンケートを通じたフィードバックは、今後の課題を明確にする上で貴重な情報源となった。

本システムの実用化を更に加速させるためには、アルゴリズムの改修によるデジタルツインデータの生成精度の向上はもとより、提供エリアの拡大とデータ活用時の利便性の向上が必要である。特に、提供エリアの拡大を実現するためには、今回活用したプラットフォームに限らず、ゲームエンジンやデータ整備事業者との技術パートナーシップを強化し、最新の 3D・AI 技術を活用した迅速なデータの更新と拡張が不可欠である。これにより、開発者やクリエイターが、より安価かつ効率的に品質の高い 3D コンテンツを用いてゲームや映像制作に限らず様々なイノベーションを創出することが可能となり、利用者に対して新たなユーザーエクスペリエンスを提供する地理的範囲の拡大が期待できる。

将来的には、3D 都市モデルの整備エリアが今後拡大することに合わせて、自動生成 AI を活用した高精度なデジタルツインデータを構築し、継続的に公開することを目指す。防災やまちづくり、モビリティなど 3DCG 技術を必要とする様々なサービスや研究においてデジタルツインデータの活用機会を増やし、デジタルツイン市場におけるプラットフォームとして PLATEAU の社会的価値を向上させることを志向する。

8. 用語集

A) アルファベット順

表 8-1 用語集（アルファベット順）

No.	用語	説明
1	CityGML	City Geography Markup Language の略。都市の 3 次元モデルデータの作成や利用のために開発された XML ベースのフォーマット。
2	FBX	Filmbox の略。3D アニメーションソフトウェア間でデータを交換するためのファイルフォーマット。
3	Fortnite	Epic Games が提供するオンラインゲーム。ユーザーが作成したマップを公開することができる。
4	Houdini	SideFX Software が提供する 3D アニメーション・VFX ソフトウェア。
5	LOD	Level of Detail の略。3D モデルの詳細度のレベルを表す。
6	NeRF	Neural Radiance Fields の略。ニューラルネットワークを使用して、複数の 2D 画像から 3D 空間を再構成する手法。
7	Unreal Engine	Epic Games が提供するゲームエンジン。
8	Unreal Editor for Fortnite (UEFN)	Fortnite 内で利用できるマップ作成向けに最適化された Unreal Engine を元にしたゲームエンジン。
9	ViT	Vision Transformer の略。自己注意メカニズムを利用した画像認識モデル。
10	Wave Function Collapse	観測済みの画像から類似パターンを生成するアルゴリズム。

B) 五十音順

表 8-2 用語集（五十音順）

No.	用語	説明
1	オルソ画像	傾きが補正され、地図と同じ正射影に変換された空中写真や衛星画像。
2	真値	測定や推定の正しさを評価する際の基準となる値。
3	テクスチャ	3D モデルの表面に貼り付ける画像。
4	プロシージャルモデリング	3D モデルを手動でモデリングするのではなく、アルゴリズムやルールに基づいて自動的に生成する手法。
5	ローポリゴン	比較的少ない数のポリゴンで構成された 3D モデル。
6	ワイヤフレーム	3D モデルを線分の集まりで表示したもの。

以上

3D 都市モデルを活用した高精度デジタルツインの構築
技術検証レポート

2024 年 3 月 発行

委託者：国土交通省 都市局

受託者：株式会社スペースデータ