

3D 都市モデルへのテクスチャ自動付与技術の開発 技術検証レポート

series
No. 139

Technical Report on the Development of Automatic Texture Mapping Technology for 3D City Models

目次

1. ユースケースの概要	- 1 -
1-1. 現状と課題	- 1 -
1-1-1. 課題認識	- 1 -
1-2. 課題解決のアプローチ	- 2 -
1-3. 創出価値	- 3 -
1-4. 想定事業機会	- 4 -
2. 実証実験の概要	- 5 -
2-1. 実証仮説	- 5 -
2-2. 検証ポイント	- 6 -
2-3. 実証フロー	- 7 -
2-4. 実施体制	- 7 -
2-5. 実証エリア	- 8 -
2-6. スケジュール	- 10 -
3. 開発スコープ	- 11 -
3-1. 概要	- 11 -
3-2. 開発内容	- 12 -
4. 実証システム	- 13 -
4-1. アーキテクチャ	- 13 -
4-1-1. システムアーキテクチャ	- 13 -
4-1-1. データアーキテクチャ	- 14 -
4-1-2. ハードウェアアーキテクチャ	- 15 -
4-2. システム機能	- 19 -
4-2-1. システム機能一覧	- 19 -
4-2-2. 利用したソフトウェア・ライブラリ	- 21 -
4-2-3. 開発機能の詳細要件	- 23 -
4-3. アルゴリズム	- 36 -
4-3-1. 利用したアルゴリズム	- 36 -
4-3-2. 開発したアルゴリズム	- 37 -
4-4. データインタフェース	- 38 -
4-4-1. ファイル入力インタフェース	- 38 -
4-4-2. ファイル出力インタフェース	- 39 -
4-4-3. 内部連携インタフェース	- 40 -
4-4-4. 外部連携インタフェース	- 63 -
4-5. 実証に用いたデータ	- 64 -
4-5-1. 利用したデータ一覧	- 64 -
4-5-2. 生成・変換するデータ	- 68 -

4-6. ユーザーインターフェース	- 69 -
4-6-1. 画面一覧	- 69 -
4-6-2. 画面遷移図	- 70 -
4-7. 実証システムの利用手順	- 71 -
4-7-1. 実証システムの利用フロー	- 71 -
4-7-2. 各画面操作方法	- 72 -
5. システムの非機能要件	- 81 -
5-1. 社会実装に向けた非機能要件	- 81 -
6. 品質	- 83 -
6-1. 機能要件の品質担保	- 83 -
6-2. 非機能要件の品質担保	- 84 -
7. 実証技術の機能要件の検証	- 85 -
7-1. 貼付け位置補正機能の検証	- 85 -
7-1-1. 検証目的	- 85 -
7-1-2. KPI	- 85 -
7-1-3. 検証方法と検証シナリオ	- 86 -
7-1-4. 検証結果	- 87 -
7-2. 操作性の検証	- 95 -
7-2-1. 検証目的	- 95 -
7-2-2. KPI	- 95 -
7-2-3. 検証方法と検証シナリオ	- 95 -
7-2-4. 検証結果	- 96 -
7-3. データ整合性/CityGML との互換性の検証	- 97 -
7-3-1. 検証目的	- 97 -
7-3-2. KPI	- 97 -
7-3-3. 検証方法と検証シナリオ	- 97 -
7-3-4. 検証結果	- 98 -
8. 実証技術の非機能要件の検証	- 105 -
8-1. 検証目的	- 105 -
8-2. KPI	- 105 -
8-2-1. 検証方法と検証シナリオ	- 105 -
8-2-2. 検証結果	- 106 -
9. 公共政策面の有用性検証	- 108 -
9-1. 検証目的	- 108 -
9-2. 検証方法	- 110 -
9-3. 被験者	- 111 -
9-4. ヒアリング・アンケートの詳細	- 112 -
9-4-1. アジェンダ・タイムテーブル	- 112 -

9-4-2. アジェンダの詳細.....	- 112 -
9-4-3. 検証項目と評価方法.....	- 113 -
9-4-4. 実証実験の様子.....	- 114 -
9-5. 検証結果.....	- 117 -
10. 成果と課題.....	- 141 -
10-1. 本実証で得られた成果.....	- 141 -
10-1-1. 3D 都市モデルの技術面での優位性.....	- 141 -
10-1-2. 3D 都市モデルのビジネス面での優位性.....	- 141 -
10-1-3. 3D 都市モデルの公共政策面での優位性.....	- 142 -
10-2. 実証実験で得られた課題と対応策.....	- 143 -
10-3. 今後の展望.....	- 146 -
11. 用語集.....	- 147 -

1. ユースケースの概要

1-1. 現状と課題

1-1-1. 課題認識

現在、全国各地で整備が進む 3D 都市モデルは、都市政策やインフラ管理、防災計画、観光など様々な分野での活用が期待されている。しかし、従来の 3D 都市モデルは航空測量等を基に構築されており、そのテクスチャの画質は十分とは言えない。特に人の目線に近いアイレベルからの景観において、建築物の外壁や看板などのディテールが曖昧であり、実際の都市空間とかい離した印象を与えてしまう。このため、都市景観の可視化やシミュレーションにおける没入感の欠如又は都市再現の信頼性不足が課題となっている。さらに、こうしたディテール不足が原因で、住民との合意形成や観光プロモーション等の活用シーンで説得力を欠くケースも見られる。都市デジタルツインの社会実装を推進する上で、より現実的かつ視覚的に高精度な都市再現が求められており、その中でも建築物への高画質画像によるテクスチャ自動付与は、緊急性と汎用性の高い技術課題として位置付けられている。

上記課題の解決に向け、昨年度は、スマートフォンで建物を撮影し、その画像を基に 3D 都市モデルへテクスチャを付与するスマートフォンアプリケーション（以下、スマホアプリ）の開発と実証を行った。アプリケーションでは、位置情報付きの高画質画像を簡易的に取得・登録できる点が高く評価された一方、建築物の形状と画像の整合性を取る処理や、貼付け位置の微調整には手動操作が必要であり、ユーザーの負担が大きかった。これにより、スモールスケールでの利用は可能だが、都市全体にスケールする段階では自動化処理の精度とスループットに課題が残った。さらに、スマホアプリでは建築物の面指定や画像の選別、貼付け結果の確認といった一連の作業を一画面内で処理する設計が困難であり、操作性や視認性にも限界が見られた。これらの課題は、今年度のブラウザベースでのシステム開発において解決を図るべき重要な示唆となった。

1-2. 課題解決のアプローチ

昨年度の実証実験では、スマートフォンで撮影された高画質な画像を 3D 都市モデルにテクスチャとして貼り付ける際、いくつかの技術的課題が明らかになった。具体的には、(1) 画像と建物面との位置・角度の整合性の確保、(2) 画像の位置合わせ・切り抜き・貼付け作業における手動操作負荷といった点である。

本年度の開発では、これらの課題に対して、主にブラウザベースの 3D 都市モデル編集・可視化システムを通じた自動化と視覚的補助によって解決を図る。

まず、画像と建物面との位置・角度の整合性については、撮影画像に含まれるカメラの位置・姿勢情報を基に、対象となる建築物モデルのジオメトリと照合し、正射変換を行う機能（要件 A002）を開発する。これにより、建物面の斜め方向から撮影された画像でも、歪みを最小限に抑えつつ正確な投影が可能となる。

次に、画像の貼付け処理における操作負荷については、自動補正機能（機能要件 A003、機能要件 A008）と自動トリミング機能（機能要件 A005）を組み合わせ、ユーザーが指定した建物面に対して、画像の位置・サイズを自動調整した上で、必要な範囲だけを切り出して貼り付ける処理を実現している。本システムでは、すべての処理を一律に完全自動化するのではなく、全体の 6 割以上の建物で自動処理が成立することを目標に、自動処理による作業負荷の軽減を図りつつ、処理結果の確認および必要最小限の調整をユーザーが行える構成としている。自動処理で対応しきれない場合には、ユーザーが微調整可能な手動修正機能（機能要件 A004）を利用し、貼付け結果を補正できる構成としている。なお、本実証で想定する実務利用においては、専門的な 3D 編集スキルを有しない自治体職員等が操作可能な範囲での手動調整を前提としている。

また、画像の選定や建物面との対応関係について、ユーザーが視覚的に把握できるブラウザ UI を整備し、対象面の指定（機能要件 A001）と貼付けの確認・更新を一貫して行える構成とする。これにより、作業効率の向上と品質管理の平準化を同時に達成する。

さらに、貼付け処理の結果は、データベース更新機能（機能要件 A008）を通じて反映され、完成データは CityGML 形式での出力機能（機能要件 A009）を通じてオープンデータとしての再利用が可能となり、自治体や民間事業者の利活用を支援する。

このように、本年度は、昨年度の実証で確認された課題を個別に抽出・分析し、それぞれに対応した自動処理・補助機能をブラウザベースで設計・実装することで、都市デジタルツインに求められる精度・効率・拡張性をバランス良く実現する開発アプローチを採用した。

1-3. 創出価値

本技術により、都市デジタルツインのリアリティが飛躍的に向上する。市民がアイレベルで 3D 都市モデルを閲覧した際に、実際に目にする街並みに近い外観が再現されることで、住民説明会、観光プロモーション、景観計画、防災訓練など、多様な活用シーンにおいて合意形成や理解促進に大きな力を発揮する。また、自治体にとっては、比較的低コストで 3D 都市モデルの更新・高度化を実現できる利点がある。さらに、民間事業者がこの高精度 3D 都市モデルをベースに各種サービス（ナビゲーション、観光アプリ、広告など）を展開することで、新たなビジネスが創出される可能性も広がる。ひいては、PLATEAU エコシステムにおける価値循環の中心的技術として、この自動テクスチャ付与技術は位置付けられる。

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"> ● 自治体職員 ● 一般ユーザー
サービス仮説	<ul style="list-style-type: none"> ● 専門的な技術・デバイス不要で高精度なテクスチャを生成できることによる利便性の向上と整備費用の削減を実現する ● 一般ユーザー巻き込みによる整備スピードの早期化を図る
提供価値	<ul style="list-style-type: none"> ● 自治体職員自らが 3D 都市モデルを整備する機会を創出する <ul style="list-style-type: none"> ➢ 専門的な技術・デバイス不要で高精度なテクスチャを生成できることによる利便性の向上と整備費用の削減を図る ➢ MMS 車両での計測等を行うことなく、自治体所有のスマートフォンで建物壁面を撮影しテクスチャ割り当てを行う機能を整備することによる費用削減を可能とする ➢ また、テクスチャの不十分な建物のみ絞って撮影することによる整備費用削減や効率化なども可能となる ● 一般ユーザーによる 3D 都市モデルの整備の促進 <ul style="list-style-type: none"> ➢ 一般ユーザーによる地元エリアの建物の写真撮影・登録、というサイクルによって 3D 都市モデルの整備促進が見込まれる

2. 実証実験の概要

2-1. 実証仮説

【クラウドソーシング型基盤の応用による画像活用の大衆化】

- 本システムは、令和6年度に開発された PLATEAU SNAP のクラウドソーシング型 3D 都市モデル作成基盤を応用し、自治体職員や一般ユーザーが撮影した高画質な建物画像を建築物モデルのテクスチャとして活用する仕組みである。都市空間の実写画像を公共データとして活用することは従来、撮影条件や正確な位置情報の管理などに専門性を要し、限定的な運用にとどまっていた。しかし、PLATEAU SNAP で確立されたスマホアプリによる都市情報収集スキームをベースとすることで、自治体職員が一般ユーザーが収集した画像を都市モデル整備に直接反映させる仕組みを確立できると仮定する。これにより、高精度なテクスチャ画像の収集・運用が日常的かつ持続的に行えるようになる

【テクスチャ作成工程の自動化による作業負荷軽減】

- 令和6年度に開発した PLATEAU SNAP におけるテクスチャ作成は、画像の選定、整形、位置合わせ、切り抜き、貼付けなど多くの工程で手作業が必要であり、高度なスキルと多くの工数を要するプロセスであった。本開発では、これら一連の作業をシステム上で自動化し、ユーザーは最小限の操作で高品質なテクスチャ付与を行えることを目指す。仮説として、画像と建物面の幾何情報に基づいた正射変換、自動トリミング、位置補正等のアルゴリズムを統合することで、作業の専門性を下げ、効率を大幅に向上できると考える。結果として、想定ユーザーである自治体や民間事業者が、専門的な知識なしに高解像度の都市景観データを構築できるようになる

【自治体・民間事業者による実務利用の促進と関与意欲の向上】

- 本開発により、高画質なテクスチャを持つ 3D 都市モデルが簡易に、かつ現実感をもって生成可能となることで、これまで「リアリティに欠け、活用しづらい」とされてきた課題が解消され、実務への導入が促進されると仮定する。景観シミュレーション、防災訓練、観光プロモーション、不動産開発説明など、視覚的説得力が求められる多様な現場での活用が進むと期待される。また、自治体職員自らが画像提供やモデル作成工程に関与できることで、技術的な理解や関心が高まり、都市モデル整備に対する主体的な取り組みが促される。結果として、都市デジタルツインの社会実装が加速されるとの仮説を置く

2-2. 検証ポイント

自治体職員及び民間事業者に対するアンケート及びヒアリングを実施することによって、以下の観点について有用性の評価を行う。

なお、本技術検証レポートでは、検証内容を分かりやすく示す目的で、商業施設や店舗等を含む建築物の外観写真を掲載している。

掲載内容について要望がある場合には、下記の間合せフォームから連絡されたい。

<PLATEAU 間合せフォーム>

<https://www.mlit.go.jp/plateau/contact/>

- 精度
 - テクスチャ貼付け位置の自動補正は正確か
 - 出力した CityGML データのテクスチャ貼付け位置は正確か
- ユーザビリティ
 - 操作方法は分かりやすいか
 - 画面表示は分かりやすいか
 - データ出力までの作業時間は実用的な水準か
- ユーザー検証・活用シナリオ評価
 - 3D 都市モデルへのテクスチャ付与を効率化できているか
 - 高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるか
 - LOD3 建築物モデルと比較して、LOD2 建築物モデル+SNAP によるテクスチャ付与モデルは有用か

2-3. 実証フロー

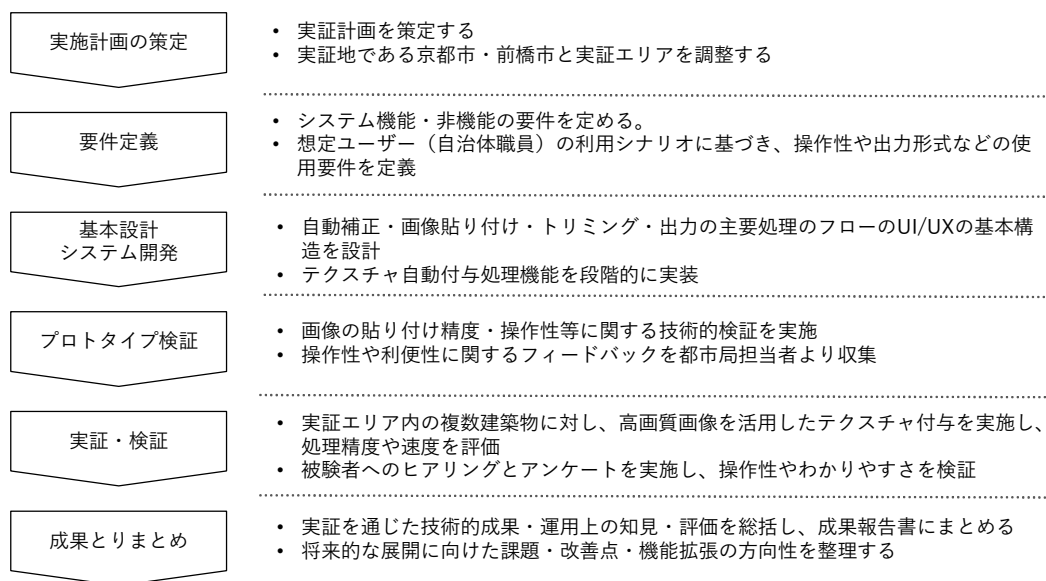


図 2-1 実証フロー

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	シナスタジア	● 計画立案 ● 開発 ● 実証運営
実施協力	京都市	● 実証実験への協力
	前橋市	● 実証実験への協力
	パスコ	● 実証実験への協力
	国際興業	● 実証実験への協力

2-5. 実証エリア

表 2-2 実証エリア(京都)

項目	内容
実証地	京都府京都市
面積	1.5 km ²
マップ (対象エリアは赤枠内)	 <p data-bbox="687 1330 1023 1361">実証エリア (京都府京都市)</p>

表 2-3 実証エリア（前橋）

項目	内容
実証地	群馬県前橋市
面積	0.5 km ²
マップ (対象エ リア は赤枠 内)	 <p data-bbox="687 1205 1023 1240">実証エリア（群馬県前橋市）</p>

2-6. スケジュール

表 2-4 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、要件定義、実証計画作成	←————→												
2. プロトタイプ開発			←————→										
3. プロトタイプ確認					↔		↔						
4. 実証実施								←→					
5. 成果取りまとめ									←————→				

3. 開発スコープ

3-1. 概要

本年度の開発では、スマートフォンで撮影された位置情報付きの高画質画像を、建築物モデルの面に対して自動的にテクスチャとして貼り付けることができるブラウザベースのシステムを構築する。このシステムは、都市デジタルツインにおいて視覚的リアリティを飛躍的に高める中核技術として位置付けられ、ユーザーが直感的に操作できるインタフェースを通じて、建物ごとの外壁に高精細な画像を効率良く適用できる仕組みを提供する。

システム上では、ユーザーが対象となる建物の面及びテクスチャとして使用する画像を選択し、画像の歪み補正や位置合わせ、不要部分の切り抜きなどが半自動で実行される。これにより、従来は手作業で行っていた煩雑な貼付け作業が大幅に省力化される。

屋根面についても、オルソ画像を活用し、半自動的にテクスチャを更新することで、側面と併せて建物全体の外観が高解像度化される。また、テクスチャ更新済みの建築物モデルは、3D 都市モデルの標準形式である CityGML 形式で出力できるよう設計され、他システムとの連携や将来的なデータ利活用を見据えた拡張性を備える。

こうした一連の開発は、都市デジタルツインの普及と利活用に向けて、より現実に近い都市景観の再現を可能とし、自治体や民間事業者による多様な分野での応用を後押しするものである。

3-2. 開発内容

「PLATEAU SNAP CMS」は、建物面の撮影画像から正射画像を生成し、建築物モデルのテクスチャとして付与するツールである。従来、建築物モデルへのテクスチャ付与は、専門的な 3D 編集ソフトを用いた手作業が中心であり、高度な専門知識と多くの工数を要していた。本システムは、これらの工程を簡略化し、実務におけるテクスチャ更新作業の効率化を図ることを目的として開発した。

本実証では、PLATEAU が提供する LOD2 建築物モデル（CityGML 形式）を使用し、自治体職員及び民間事業者がスマートフォンで撮影した建物面画像を基に、テクスチャを生成・付与した。建物種別（平屋、商業ビル、大型商業施設、寺社、教会、駅等）ごとに複数の画像を用いてテクスチャ付与を行い、テクスチャ貼付け位置の自動補正の正確性について評価を行った。また、テクスチャを付与した CityGML データを PLATEAU Builder や PLATEAU SDK 等のビューアツールに読み込み、問題なく可視化できるかを検証した。

さらに、自治体職員及び民間事業者を対象に、操作性や活用可能性に関するアンケート調査を実施し、LOD2 建築物モデルへのテクスチャ付与の有用性や、LOD3 建築物モデルとの比較における評価を収集した。これにより、技術的な実現可能性に加え、実務利用を想定した場合の有用性についても検証を行った。

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

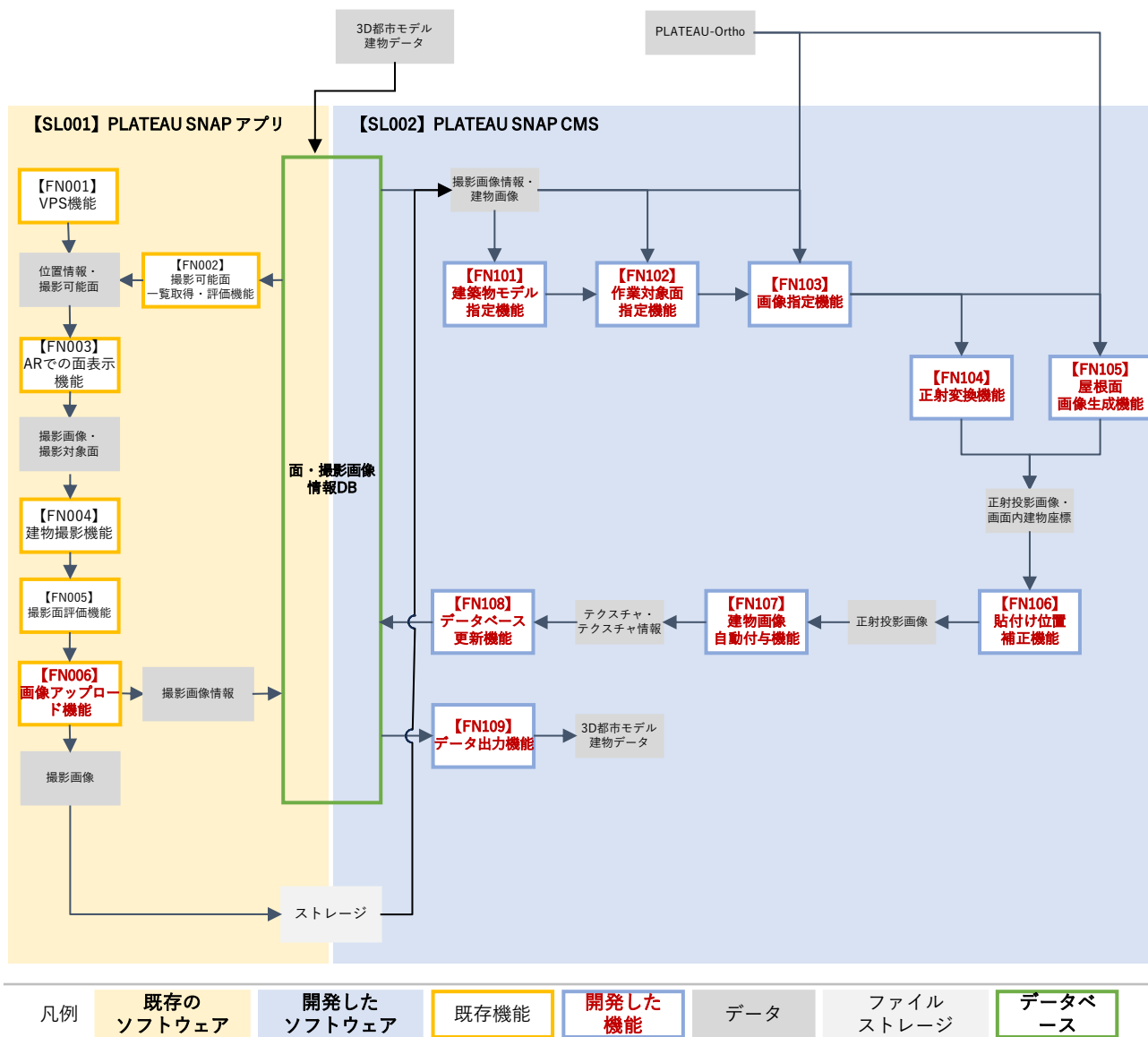
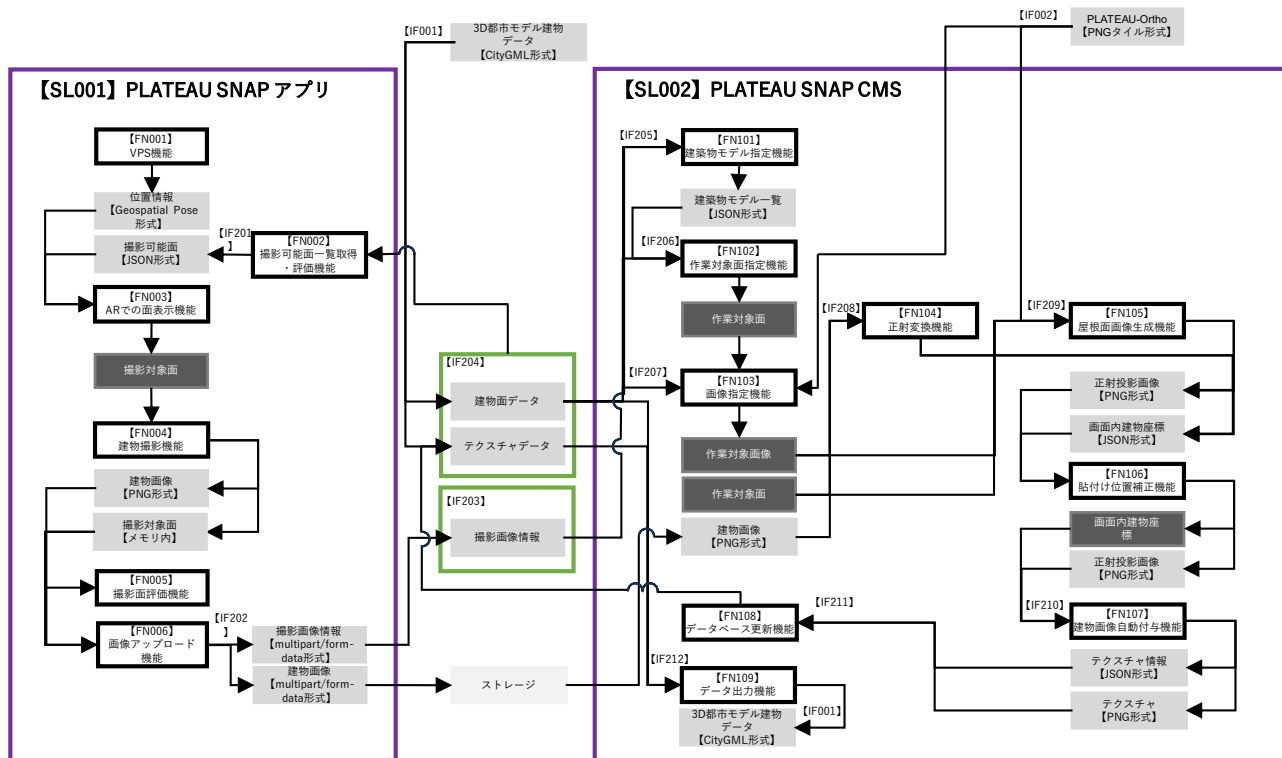


図 4-1 システムアーキテクチャ

4-1-1. データアーキテクチャ



凡例

ソフトウェア	【FNxxx】 データ処理	データ 【形式】	ファイル ストレージ	データ ベース	UI入力項目	【IFxxx】 インターフェース
--------	------------------	-------------	---------------	------------	--------	---------------------

図 4-2 データアーキテクチャ

4-1-2. ハードウェアアーキテクチャ

4-1-2-1. 利用するハードウェア一覧

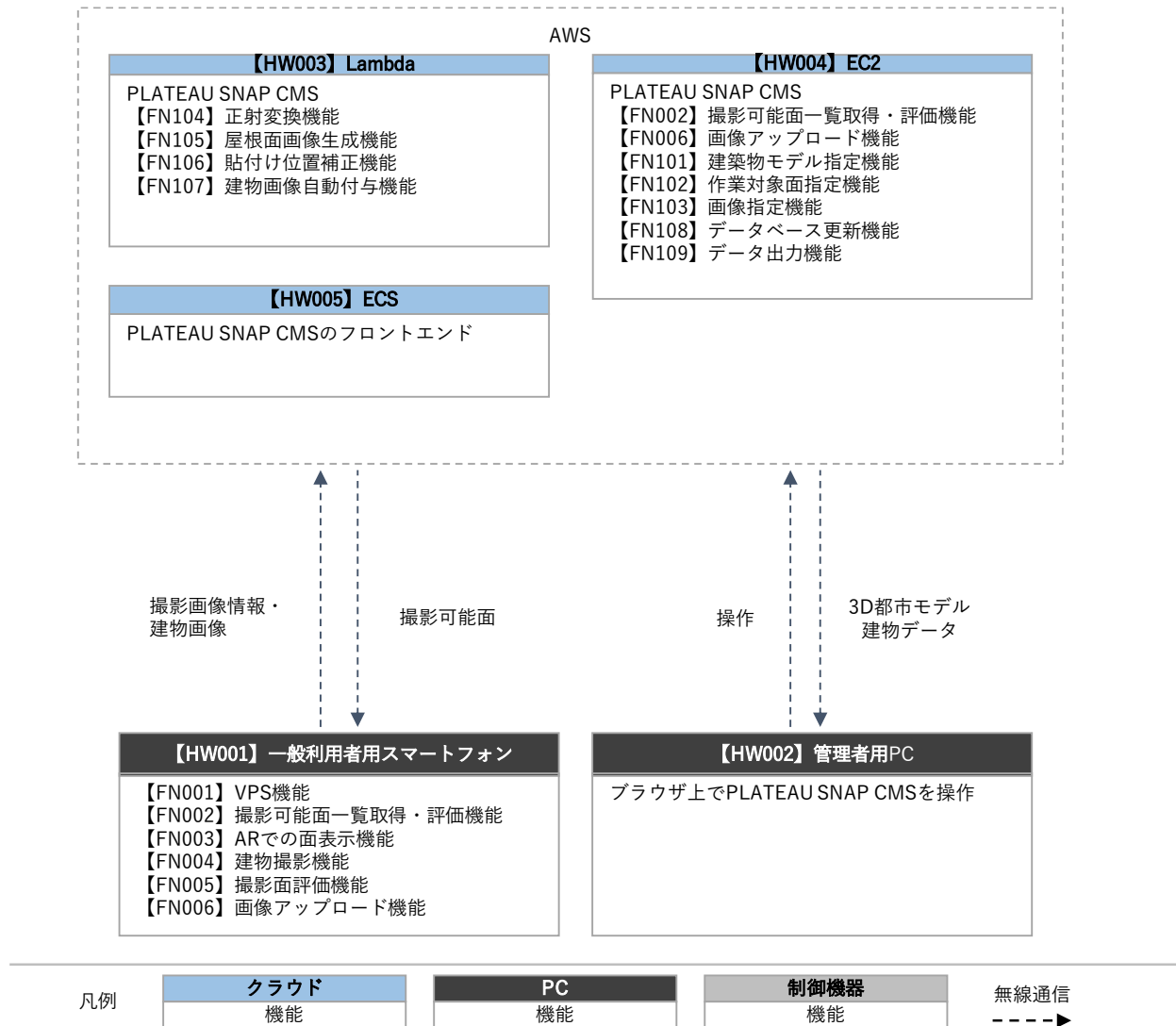


図 4-3 ハードウェアアーキテクチャ

表 4-1 利用するハードウェア一覧

ID	種別	品番	用途
HW001	一般利用者用スマートフォン	iPhone 13 Pro	● FN001～N006
HW002	管理者用 PC	ROG Zephyrus G14 GA401QC	● PLATEAU SNAP CMS を操作する
HW003	Lambda	AWS	● FN104～FN107
HW004	EC2	AWS	● FN002、FN006、FN101～FN103、FN109
HW005	ECS	AWS	● PLATEAU SNAP CMS のフロントエンドホスティング

4-1-2-1. 利用するハードウェア詳細

1) 【HW001】 Apple 社 iPhone 13 Pro

- 選定理由
 - 実証システムを利用できるのに十分なスペックである
 - 市販品として広く流通している
- 仕様・スペック
 - CPU 及び GPU : A15 Bionic
 - メモリ : 6GB
 - ストレージ : 256GB
 - OS : iOS 18.3.1
 - ディスプレイサイズ : 6.1 インチ
 - カメラ性能 : 12MP、 $f/1.5$ 絞り値 (デフォルトの広角モード)
 - 画像撮影フォーマット : JPEG
 - 対応している衛星測位システム (GNSS) : GPS、GLONASS、Galileo、QZSS、BeiDou
- イメージ



図 4-4 iPhone 13 Pro

2) 【HW002】 ASUS 社 ROG Zephyrus G14 GA401QC

- 選定理由
 - 実証システムを利用できるのに十分なスペックである
 - 市販品として広く流通している
- 仕様・スペック
 - CPU : AMD Ryzen 7 5800HS
 - GPU : NVIDIA GeForce RTX 3050
 - メモリ : 24GB
 - ストレージ (SSD) : 512GB
 - OS : Windows11 Home
- イメージ



図 4-5 ROG Zephyrus G14 GA401QC

3) ¹ 【HW003】 Lambda

- 選定理由
 - それぞれが独自のサービスとして構成されており、疎結合になっている
 - 柔軟なスケールが可能である
 - 運用コストが低い
- 仕様・スペック
 - コンテナイメージ (cv2 が標準の Lambda には入っていないため)

¹ 公式 HP より抜粋 : <https://ouster.com/jp/products/scanning-lidar/os0-sensor/>

4) 【HW004】 EC2

- 選定理由
 - 実証システムを利用できるのに十分なスペックであるため
 - 3DCityDB Importer/Exporter を利用することで工数を削減できる
 - PLATEAU SNAP アプリで使用しているものを流用できる
- 仕様・スペック
 - ami_id: ami-06c6f3fa7959e5fdd
 - instance_type: t3.large

5) 【HW005】 ECS

- 選定理由
 - 実証システムを利用できるのに十分なスペックであるため
 - Remix 製のアプリをホスティングできる
 - 柔軟なスケールが可能である
 - 運用コストが低い
- 仕様・スペック
 - CPU : 0.25 vCPU
 - メモリ : 0.5GB

4-2. システム機能

4-2-1. システム機能一覧

表 4-2 【HW001】スマートフォン用機能一覧

※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
【SW001】	FN001	VPS 機能	<ul style="list-style-type: none"> カメラ映像を利用して、周囲の建物やランドマークからカメラの位置や向きを高精度で推定する機能
	FN002	撮影可能面一覧取得・評価機能	<ul style="list-style-type: none"> 現在位置で撮影した際に、画像をテクスチャとして貼付け可能な面を計算し、画像の品質を評価する機能
	FN003	AR での面表示機能	<ul style="list-style-type: none"> クラウドから撮影可能な面と品質評価値を取得し、カメラ映像に AR 表示する機能
	FN004	建物撮影機能	<ul style="list-style-type: none"> 撮影対象の面を選択し、建物の面の画像を撮影する機能
	FN005	撮影面評価機能	<ul style="list-style-type: none"> 撮影した画像の品質を評価する機能
	FN006	画像アップロード機能	<ul style="list-style-type: none"> 建物面を撮影した画像をアップロードする機能

表 4-3 【HW002】 PC 用機能一覧

※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
【SW002】	FN101	建築物モデル指定機能	<ul style="list-style-type: none"> 対象のモデルを一覧で表示し、選択することにより、作業対象面選択へ遷移する
	FN102	作業対象面指定機能	<ul style="list-style-type: none"> 作業対象面を一覧で表示し、選択することにより、画像選択へ遷移する
	FN103	画像指定機能	<ul style="list-style-type: none"> 画像を一覧表示し、選択することにより、貼付け位置補正へ遷移する
	FN104	正射変換機能	<ul style="list-style-type: none"> PLATEAU SNAP アプリ撮影画像を正射投影された画像に変換する
	FN105	屋根面画像生成機能	<ul style="list-style-type: none"> PLATEAU-Ortho から画像をダウンロードし、対象の屋根面を内包する画像を生成する
	FN106	貼付け位置補正機能	<ul style="list-style-type: none"> 【FN005】又は【FN006】の画像と建物の枠を表示し、テクスチャの貼付け位置を補正する
	FN107	建物画像自動付与機能	<ul style="list-style-type: none"> テクスチャにする領域で画像をトリミングする 対象面を UV 展開し、テクスチャとして表示するためのデータを生成する
	FN108	データベース更新機能	<ul style="list-style-type: none"> 作成したテクスチャを基に、データベースを更新する
	FN109	データ出力機能	<ul style="list-style-type: none"> データベースの値から詳細情報を表示する

4-2-2. 利用したソフトウェア・ライブラリ

表 4-4 利用するソフトウェア一覧

※朱文字：新規開発・既存改修

ID	項目	バージョン	内容
SW001	PLATEAU SNAP App	-	<ul style="list-style-type: none"> 令和 6 年度に Project PLATEAU で開発したデジタルツインの実現に向けたクラウドソーシング型 3D 都市モデル作成システム (PLATEAU SNAP) のモバイルアプリケーション
SW002	PLATEAU SNAP CMS	-	<ul style="list-style-type: none"> 本年度開発する 3D 都市モデルへのテクスチャを付与するブラウザシステム 令和 6 年度に開発した PLATEAU SNAP のバックエンドサーバーを基に、本システムを開発した
SW003	C#	8.x	<ul style="list-style-type: none"> ゲーム開発や業務アプリケーション開発に広く使用されるプログラミング言語。特に Microsoft の .NET 環境での開発に適しており、Unity を用いたゲーム開発にも活用される
SW004	Unity	2022.3.44f1	<ul style="list-style-type: none"> 3D・2D ゲームやインタラクティブなシミュレーションを開発するためのゲームエンジン。リアルタイムレンダリングやマルチプラットフォーム対応に優れ、ゲーム以外にも建築、映像、VR/AR アプリケーション開発にも利用される
SW005	CityGML Importer/Exporter	5.5.0	<ul style="list-style-type: none"> CityGML データをデータベースに読み込むためのツール
SW006	PostgreSQL	16.6	<ul style="list-style-type: none"> オープンソースのデータベース管理システム
SW007	Docker	-	<ul style="list-style-type: none"> アプリケーションをコンテナ化し、どこでも動作させることができる軽量な仮想化ツール
SW008	Docker Compose	-	<ul style="list-style-type: none"> 複数の Docker コンテナを簡単に管理・起動できるツール
SW009	Python	3.12	<ul style="list-style-type: none"> 画像処理の自動化に適しているプログラミング言語で、物体検出・変換・解析など多様な処理が効率良く行える
SW010	Remix	2.16.7	<ul style="list-style-type: none"> Web アプリ開発用フレームワーク

表 4-5 利用するライブラリー一覧

※朱文字：新規開発・既存改修

ID	項目	バージョン	内容
LB001	PostGIS	3.1.4	● PostgreSQL で地理情報システム (GIS) を扱えるようにするためのオープンソースの拡張機能
LB002	NetTopologySuite	2.5.0	● オープンソースの.NET 用の GIS ライブラリ
LB003	Npgsql	8.0.6	● オープンソースの ADO.NET ドライバ
LB004	AWSSDK	3.7.400.86	● .NET 用の AWS にアクセスするための SDK
LB005	boto3	1.37.20	● Python の AWS 公式 SDK
LB006	opencv-python	4.9.0.80	● Python の画像処理ライブラリ
LB007	numpy	1.26.4	● Python の数値計算ライブラリ
LB008	mercantile	1.2.1	● Python のタイル座標変換ライブラリ
LB009	shapely	2.0.4	● Python の幾何学的オブジェクトの操作・解析ライブラリ

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) スマートフォン用機能一覧

1. 【FN006】画像アップロード機能

- 機能概要
 - 撮影画像と貼付け先の面情報をクラウドにアップロードする
- データ仕様
 - 入力
 - ◇ 撮影対象面情報
 - 内容
 - 撮影対象面の座標情報
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF202】を参照
 - 出力
 - ◇ 撮影画像データベース
 - 内容
 - 撮影画像情報に撮影位置情報及び撮影面情報をひも付けたデータベース
 - 形式
 - PostgreSQL
 - データ詳細
 - 内部連携インタフェース【IF204】を参照
 - 機能詳細
 - 建築物モデル取得
 - ◇ 処理内容

2) 管理者用 PC 機能一覧

2. 【FN101】建築物モデル指定機能

- 機能概要
 - テクスチャ更新対象の建築物モデルを指定する

- データ仕様
 - 入力
 - ◇ なし
 - 出力
 - ◇ 建築物モデル一覧
 - 内容
 - テクスチャを更新可能な建築物モデル一覧
 - 形式
 - JSON
 - データ詳細
 - 内部連携インターフェース【IF205】を参照
- 機能詳細
 - 建築物モデル取得
 - ◇ 処理内容
 - 撮影画像がひも付けられている建築物モデルを取得する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW008】 Docker Compose
 - ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB002】 NetTopologySuite
 - 【LB003】 Npgsql
 - ◇ 利用するアルゴリズム
 - なし

3. 【FN102】作業対象面指定機能

- 機能概要
 - テクスチャ更新対象の建物面を指定する
- データ仕様
 - 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式

- API の Route パラメータ
- データ詳細
 - 内部連携インタフェース【IF206】を参照
- 出力
 - ◇ 建築物モデル一覧
 - 内容
 - テクスチャを更新可能な建物面一覧
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF206】を参照
- 機能詳細
 - 建築物モデル取得
 - ◇ 処理内容
 - 指定された建築物モデルのうち、撮影画像がひも付けられている建物面を取得する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW008】 Docker Compose
 - ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB002】 NetTopologySuite
 - 【LB003】 Npgsql
 - ◇ 利用するアルゴリズム
 - なし

4. 【FN103】画像指定機能

- 機能概要
 - テクスチャとして使用する撮影画像を指定する
- データ仕様
 - 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - API の Route パラメータ

- データ詳細
 - 内部連携インタフェース【IF207】を参照
- ◇ 建物面 ID
 - 内容
 - テクスチャを更新する建物面の ID
 - 形式
 - API の Route パラメータ
 - データ詳細
 - 内部連携インタフェース【IF207】を参照
- 出力
 - ◇ 画像一覧
 - 内容
 - 利用可能な撮影画像一覧
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF207】を参照
- 機能詳細
 - 建築物モデル取得
 - ◇ 処理内容
 - 指定された建築物モデルと建物面にひも付く撮影画像を取得する
 - 画像の Presigned URL を取得する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW008】 Docker Compose
 - ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB002】 NetTopologySuite
 - 【LB003】 Npgsql
 - 【LB004】 AWSSDK
 - ◇ 利用するアルゴリズム
 - なし

5. 【FN104】正射変換機能

- 機能概要
 - 撮影面情報と建物面の位置等から、正射投影画像に変換し、画像内の建物座標を取得する

- データ仕様
 - 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - ◇ 建物面 ID
 - 内容
 - テクスチャを更新する建物面の ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - ◇ 撮影画像 ID
 - 内容
 - テクスチャ撮影画像の ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - 出力
 - ◇ 正射投影画像
 - 内容
 - 建物面に対して正射投影された画像
 - 形式
 - PNG 画像
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - ◇ 画像内建物座標
 - 内容
 - 正射投影画像内の建物座標
 - 形式
 - JSON
 - データ詳細

- 内部連携インターフェース【IF208】を参照

- 機能詳細

- 正射変換

- ◇ 処理内容

- 指定された面 ID・画像 ID をキーに、DB から建物面情報・撮影面情報を取得する
 - 指定された画像 ID をキーに、ストレージから建物画像を取得する
 - 画像と画像内建物座標を基に、射影変換行列を計算する（このとき、位置ずれを考慮して、少しバッファした領域で計算する）
 - 画像に射影変換を適用し、正射画像と変換後の画像内座標を出力する

- ◇ 利用するソフトウェア

- 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW009】 Python

- ◇ 利用するライブラリ

- 【LB001】 PostGIS
 - 【LB005】 boto3
 - 【LB006】 opencv-python
 - 【LB007】 numpy
 - 【LB009】 shapely

- ◇ 利用するアルゴリズム

- 【AL001】 正射変換(正面から見たように射影変換)

6. 【FN105】 屋根面画像生成機能

- 機能概要

- PLATEAU-Ortho から屋根面のテクスチャ画像を作成する

- データ仕様

- 入力

- ◇ 建築物モデル ID

- 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インターフェース【IF209】を参照

- ◇ 建物面 ID

- 内容

- テクスチャを更新する建物面の ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF209】を参照
- 出力
 - ◇ 屋根面画像
 - 内容
 - 建物面に対して正射投影された画像
 - 形式
 - PNG 画像
 - データ詳細
 - 内部連携インタフェース【IF209】を参照
 - ◇ 画像内建物座標
 - 内容
 - 屋根面画像内の建物座標
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF209】を参照
- 機能詳細
 - 屋根面画像生成
 - ◇ 処理内容
 - 指定された面 ID をキーに、DB からや屋根面情報を取得する
 - 屋根面を含むタイルの範囲を計算する（このとき、位置ずれを考慮して、少しバッファした領域で計算する）
 - PLATEAU-Ortho から屋根面を含むタイル画像を取得する
 - タイルを結合して一つの画像にする
 - 少しバッファをもって、結合した画像を屋根面の領域でトリミングする
 - トリミングした画像内での建物座標を計算する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW009】 Python
 - ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB005】 boto3

- 【LB006】 opencv-python
- 【LB007】 numpy
- 【LB008】 mercantile
- 【LB009】 shapely
- ◇ 利用するアルゴリズム
 - なし

7. 【FN106】 貼付け位置補正機能

- 機能概要
 - 建物面に対する建物面画像の貼付け位置を手動で補正する
- データ仕様
 - 入力
 - ◇ 建物面画像
 - 内容
 - 建物面に対して正射投影された画像/生成された屋根面画像
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF209】を参照
 - ◇ 画像内建物座標
 - 内容
 - 画像内の建物座標
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF209】を参照
 - 出力
 - ◇ 画像内建物座標
 - 内容
 - 画像内の建物座標
 - 形式
 - PLATEAU SNAP CMS アプリ内のメモリ
 - データ詳細
 - なし
- 機能詳細
 - 貼付け位置補正

- ◇ 処理内容
 - 建物面画像と画像内建物座標を重ねて描画する
 - UI で画像内建物座標をマウス操作により補正する
- ◇ 利用するライブラリ
 - 【SW010】 Remix
- ◇ 利用するアルゴリズム
 - なし

8. 【FN107】建物画像自動付与機能

- 機能概要
 - 建築物モデルを UV 展開した上で、【FN004】～【FN006】により補正等された高画質な画像について、【FN001】～【FN003】で指定した建築物モデルの側面にテクスチャとして置換して更新する
- データ仕様
 - 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF210】を参照
 - ◇ 建物面 ID
 - 内容
 - テクスチャを更新する建物面の ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF210】を参照
 - ◇ 画像内建物座標
 - 内容
 - 画像内の建物座標
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF210】を参照
 - 出力
 - ◇ テクスチャ画像

- 内容
 - 建物面でトリミングされた画像
- 形式
 - PNG 画像
- データ詳細
 - 内部連携インタフェース【IF210】を参照
- ◇ テクスチャ情報
 - 内容
 - PLATEAU SNAP CMS アプリでプレビューするためのテクスチャ情報
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF210】を参照
- 機能詳細
 - 建物画像自動付与機能
 - ◇ 処理内容
 - 指定された面 ID をキーに、DB から建物面情報を取得する
 - 画像内建物座標を内包する領域で建物画像をトリミングする
 - UV 展開してテクスチャ情報を作成する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker
 - 【SW009】 Python
 - ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB005】 boto3
 - 【LB006】 opencv-python
 - 【LB007】 numpy
 - 【LB009】 shapely
 - ◇ 利用するアルゴリズム
 - なし
- 9. 【FN108】 データベース更新機能
 - 機能概要
 - FN107 で付与した建物画像、テクスチャ情報をデータベースに反映する
 - データ仕様

- 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF211】を参照
 - ◇ 建物面 ID
 - 内容
 - テクスチャを更新する建物面の ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF211】を参照
 - ◇ 画像内建物座標
 - 内容
 - 画像内の建物座標
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース【IF211】を参照
- 出力
 - ◇ 3D City DB
 - 内容
 - 3D City DB 内の建築物モデル
 - 形式
 - PostgreSQL
 - データ詳細
 - 内部連携インタフェース【IF104】を参照
- 機能詳細
 - 建物画像自動付与機能
 - ◇ 処理内容
 - 指定された情報を基に、データベースを更新する
 - ◇ 利用するソフトウェア
 - 【SW006】 PostgreSQL
 - 【SW007】 Docker

- 【SW008】 Docker Compose
- ◇ 利用するライブラリ
 - 【LB001】 PostGIS
 - 【LB002】 NetTopologySuite
 - 【LB003】 Npgsql
 - 【LB004】 AWSSDK
 - 【LB005】 Swashbuckle.AspNetCore
- ◇ 利用するアルゴリズム
 - なし

10. 【FN109】 データ出力機能

- 機能概要
 - 高画質な画像を付与した建築物モデルを CityGML 形式で出力する
- データ仕様
 - 入力
 - ◇ 建築物モデル ID
 - 内容
 - テクスチャを更新する建築物モデルの ID
 - 形式
 - JSON
 - データ詳細
 - 内部連携インタフェース 【IF212】 , 【IF213】 , 【IF214】 , 【IF215】 を参照
 - 出力
 - ◇ CityGML
 - 内容
 - 高画質な画像を付与した建築物モデルの CityGML
 - 形式
 - CityGML
 - データ詳細
 - ファイル出力インタフェース 【IF101】 を参照
 - 機能詳細
 - データ出力
 - ◇ 処理内容
 - 3DCityDB Importer/Exporter で、gmlid を指定して、CityGML を出力する
 - ◇ 利用するソフトウェア
 - 【SW005】 CityGML Importer/Exporter

- 【SW007】 Docker
- 【SW008】 Docker Compose
- ◇ 利用するライブラリ
 - なし
- ◇ 利用するアルゴリズム
 - なし

4-3. アルゴリズム

4-3-1. 利用したアルゴリズム

表 4-6 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	FN104	正射変換(正面から見たように射影変換)	● 画像や座標を別の視点から見たように変形する	● PLATEAU SNAP Appで撮影した建物面の画像を、正面から撮影したように変換するのに適しているため

1) 【AL001】射影変換

● 計算

➤ 元の画像内の頂点座標を (x,y)、返還後の頂点座標を (x',y')、変換パラメータを a~h として

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

● イメージ

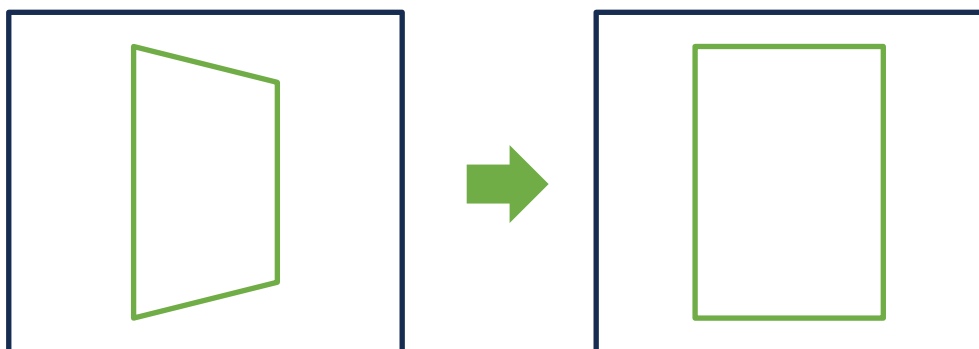


図 4-6 正射変換のイメージ

4-3-2. 開発したアルゴリズム

開発したアルゴリズムはなし。

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】 CityGML ファイル入力

- インタフェースの概要
 - 3D 都市モデルを記述・交換するための国際標準のデータフォーマット
- 本インタフェースを利用する機能
 - なし
- ファイル詳細
 - 建築物モデル (LOD2)
 - ファイルの構造やデータ型はこちら²を参照

2) 【IF002】 PLATEAU-Ortho ファイル入力

- インタフェースの概要
 - 航空写真測量によって作成したオルソ画像をタイル化したもの
- 本インタフェースを利用する機能
 - 【FN103】
 - 【FN105】
- ファイル詳細
 - WMTS 形式の PNG 画像
 - ファイルの構造やデータ型はこちら³を参照

² OGC (Open Geospatial Consortium) によって定義された CityGML のスキーマファイル：
<https://schemas.opengis.net/citygml/building/2.0/building.xsd>

³ Project PLATEAU で航空写真測量によって作成したオルソ画像をタイル化したもの：
<https://github.com/Project-PLATEAU/plateau-streaming-tutorial/blob/main/ortho/plateau-ortho-streaming.md>

4-4-2. ファイル出力インターフェース

1) 【IF101】 CityGML ファイル出力

- インタフェースの概要
 - 3D 都市モデルを記述・交換するための国際標準のデータフォーマット
- 本インタフェースを利用する機能
 - 【FN109】
- ファイル詳細
 - テクスチャの更新された建築物モデル (LOD2)
 - ファイルの構造やデータ型はこちら²を参照

4-4-3. 内部連携インターフェース

1) 【IF201】 撮影可能面一覧取得 API

- インタフェースの概要
 - 現在位置から撮影可能な建物面の一覧を取得する
- 本インターフェースを利用する機能
 - 【FN002】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /api/visible-surfaces
- リクエストパラメータ

```
{
  "from": {
    "longitude": number,
    "latitude": number,
    "altitude": number
  },
  "to": {
    "longitude": number,
    "latitude": number,
    "altitude": number
  },
  "field_of_view": number,
  "max_distance": number
}
```

- レスポンス

```
{
  "surfaces": [
    "gmlid": string,
    "coordinates": [
      [
        [number, number, number]
        [number, number, number]
        [number, number, number]
        ...
      ],
      ...
    ],
    ...
  ]
}
```

表 4-7 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
500	処理エラー

2) 【IF202】 画像アップロード API

- インタフェースの概要
 - 撮影した画像をアップロードする
- 本インタフェースを利用する機能
 - 【FN005】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /api/building-image
- リクエストパラメータ

multipart/form-data 形式で以下のパートを含む

表 4-8 リクエストパラメータ

パラメータ	説明	値	必須
image	ファイル	-	○
metadata	画像に関連するメタデータの JSON 文字列	-	○

```

{
  "gmlid": string,
  "from": {
    "longitude": number,
    "latitude": number,
    "altitude": number
  },
  "to": {
    "longitude": number,
    "latitude": number,
    "altitude": number
  },
  "role": number,
  "timestamp": DateTime,
  "coordinates": "POLYGON ((77 725, 77 1790, 1100 1790, 1100 725, 77 725))"
}
    
```

- レスポンス

表 4-9 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
500	処理エラー

3) 【IF203】撮影画像データベース

- インタフェースの概要
 - 撮影した画像を保持するデータベース
- 本インタフェースを利用する機能
 - 【FN006】
 - 【FN101】
 - 【FN102】
 - 【FN103】
 - 【FN104】

表 4-10 images テーブル

フィールド	データ型	説明
id	BIGSERIAL	画像のユニーク ID
uri	VARCHAR (256)	画像の URL
from_latitude	DOUBLE PRECISION	カメラの緯度
from_longitude	DOUBLE PRECISION	カメラの経度
from_altitude	DOUBLE PRECISION	カメラの高度
to_latitude	DOUBLE PRECISION	カメラが向く方向の緯度
to_longitude	DOUBLE PRECISION	カメラが向く方向の経度
to_altitude	DOUBLE PRECISION	カメラが向く方向の高度
roll	DOUBLE PRECISION	カメラの回転
timestamp	TIMESTAMP	撮影日時
coordinates	geometry (MultiPolygon)	画像内建物座標

表 4-11 image_surface_relations テーブル

フィールド	データ型	説明
id	BIGSERIAL	ユニーク ID
image_id	BIGINT	画像の ID
gmlid	VARCHAR (256)	撮影面の ID

4) 【IF204】 CityGML データベース

- インタフェースの概要
 - 3D 都市モデルをデータベース上で管理・分析するためのデータベーススキーマ
 - 3DCityDB⁴の仕様に準拠したデータベースを構築する
- 本インタフェースを利用する機能
 - 【FN002】
 - 【FN006】
 - 【FN101】
 - 【FN102】
 - 【FN103】
 - 【FN104】
 - 【FN105】
 - 【FN107】
 - 【FN108】
 - 【FN109】

⁴ CityGML 規格の 3D 都市モデルを効率よく管理・活用するためのデータベース基盤：
<https://github.com/3dcitydb>

5) 【IF205】 建築物モデル取得 API

- インタフェースの概要
 - テクスチャを編集可能な建築物モデルの一覧を取得する
- 本インタフェースを利用する機能
 - 【FN101】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /api/buildings
- リクエストパラメータ

表 4-12 リクエストパラメータ

パラメータ	説明	値	必須
sort_type	ソート順	id_asc id_desc	×
page_number	ページ番号	-	×
page_size	ページサイズ	-	×

- レスポンス

表 4-13 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
500	処理エラー

```
{
  "total_count": 23,
  "page_size": 10,
  "current_page": 1,
  "total_pages": 3,
  "has_next": true,
  "values": [
    {
      "id": 1
      "gmlid": "poly-68487504-60be-438e-bf50-52648210cf3d",
      "thumbnail": "base64 エンコードされたサムネイル画像",
      "address": "東京都港区東新橋 2 丁目"
    }
  ]
}
```

6) 【IF206】 作業対象面取得 API

- インタフェースの概要
 - 建築物モデルの ID を指定してテクスチャを編集可能な建物面の一覧を取得する
- 本インタフェースを利用する機能
 - 【FN102】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /api/faces/{buildingId}
- リクエストパラメータ

表 4-14 リクエストパラメータ

パラメータ	説明	値	必須
building_id	建築物モデルの ID	-	○
sort_type	ソート順	id_asc id_desc	×
page_number	ページ番号	-	×
page_size	ページサイズ	-	×

- レスポンス

表 4-15 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{
  "total_count": 23,
  "page_size": 10,
  "current_page": 1,
  "total_pages": 3,
  "has_next": true,
  "values": [
    {
      "id": 1044,
      "gmlid": "poly-68487504-60be-438e-bf50-52648210cf3d",
      "thumbnail": "base64 エンコードされたサムネイル画像",
      "timestamp": "2025-11-29T05:21:08.234Z"
      "is_ortho": false
    }
  ]
}
```

7) 【IF207】 画像取得 API

- インタフェースの概要
 - 建築物モデルの ID と建物面の ID を指定してテクスチャとして編集できる画像の一覧を取得する
- 本インタフェースを利用する機能
 - 【FN103】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /api/images/{buildingId}/{faceId}
- リクエストパラメータ

表 4-16 リクエストパラメータ

パラメータ	説明	値	必須
building_id	建築物モデルの ID	-	○
face_id	建物面の ID	-	○
sort_type	ソート順	id_asc id_desc	×
page_number	ページ番号	-	×
page_size	ページサイズ	-	×

- レスポンス

表 4-17 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{
  "total_count": 23,
  "page_size": 10,
  "current_page": 1,
  "total_pages": 3,
  "has_next": true,
  "values": [
    {
      "id": 4,
      "gmlid": "poly-68487504-60be-438e-bf50-52648210cf3d",
      "thumbnail": "base64 エンコードされたサムネイル画像",
      "timestamp": "2025-05-11T06:08:54.875Z",
      "is_ortho": false
    }
  ]
}
```

8) 【IF208】 正射変換 API

- インタフェースの概要
 - 指定された画像を正射変換する
- 本インタフェースを利用する機能
 - 【FN104】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /api/transform
- リクエストパラメータ

```

{
  "building_id": 1,
  "face_id": 1044,
  "image_id": 4
}
```

- レスポンス

表 4-18 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```

{
  "path": "S3://xxxxx",
  "uri": "<Presigned URL>",
  "coordinates": "POLYGON ((77 725, 77 1790, 1100 1790, 1100 725, 77 725))"
}
```

9) 【IF209】屋根面画像生成 API

- インタフェースの概要
 - 指定された屋根面のテクスチャ画像を生成する
- 本インタフェースを利用する機能
 - 【FN105】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /api/roof-extraction
- リクエストパラメータ

```
{
  "building_id": 1,
  "face_id": 1044
}
```

- レスポンス

表 4-19 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{
  "path": "S3://xxxxx",
  "uri": "<Presigned URL>",
  "coordinates": "POLYGON ((77 725, 77 1790, 1100 1790, 1100 725, 77 725))"
}
```

10) 【IF210】 建物画像自動付与 API

- インタフェースの概要
 - 建物面にテクスチャを付与した建築物モデルを取得する
- 本インタフェースを利用する機能
 - 【FN107】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /api/textures/preview
- リクエストパラメータ

```

{
  "building_id": 1,
  "face_id": 1044,
  "uri": "<Presigned URL>",
  "coordinates": [
    [15.752775844330506, 56.51818473499491],
    [12.404103313648555, 1583.7171296636216],
    [452.5081102425978, 1591.3582710948883],
    [449.8219848674731, 46.579248886169665],
    [15.752775844330506, 56.51818473499491]
  ]
}
    
```

- レスポンス

表 4-20 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{  
  "gltf": "ewogICJhY2Nlc3NvcnMil" // glTF Embedded の base64 エンコード,  
}
```

11) 【IF211】 データベース更新 API

- インタフェースの概要
 - 生成したテクスチャをデータベースに反映する
- 本インタフェースを利用する機能
 - 【FN108】
- プロトコル
 - HTTPS
- メソッド
 - PUT
- パス
 - /api/textures
- リクエストパラメータ

```

{
  [
    "building_id": 1,
    "face_id", 1044,
    "path": "S3://xxxxx",
    "coordinates": [
      [15.752775844330506, 56.51818473499491],
      [12.404103313648555, 1583.7171296636216],
      [452.5081102425978, 1591.3582710948883],
      [449.8219848674731, 46.579248886169665],
      [15.752775844330506, 56.51818473499491]
    ],
    ...
  ]
}
    
```

- レスポンス

表 4-21 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

12) 【IF212】 データ出力 API

- インタフェースの概要
 - 建築物モデルを指定して CityGML として出力する
- 本インタフェースを利用する機能
 - 【FN109】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /export
- リクエストパラメータ

```

{
  "id": 1,
  "file_name" "1.zip",
}
```

- レスポンス

表 4-22 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

13) 【IF213】メッシュデータ出力 API

- インタフェースの概要
 - メッシュを指定して CityGML として出力する
- 本インタフェースを利用する機能
 - 【FN109】
- プロトコル
 - HTTPS
- メソッド
 - POST
- パス
 - /export-mesh-async
- リクエストパラメータ

```

{
  "mesh_code": 54394065,
  "file_name" "1.zip",
}
```

- レスポンス

表 4-23 レスポンス

ステータスコード	説明
202	受付に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{  
  "id": 1,  
  "type": "export_mesh",  
  "status": "pending",  
  "parameter": {  
    "type": "export_mesh"  
  },  
  "result_parameter": {  
    "type": "export_mesh"  
  },  
  "message": "string",  
  "created_at": "2025-11-29T05:33:53.928Z",  
  "updated_at": "2025-11-29T05:33:53.928Z"  
}
```

14) 【IF214】メッシュコード取得 API

- インタフェースの概要
 - 建築物モデルの ID を指定して、その建物が属する 3 次メッシュコードを取得する
- 本インタフェースを利用する機能
 - 【FN109】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /buildings/{building_id}/mesh-code
- リクエストパラメータ

表 4-24 リクエストパラメータ

パラメータ	説明	値	必須
building_id	建築物モデルの ID	-	○

- レスポンス

表 4-25 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```

{
  "mesh_code": 54394065
}
```

15) 【IF215】 Job 取得 API

- インタフェースの概要
 - 非同期で動作する API のステータスを取得する
- 本インタフェースを利用する機能
 - 【FN109】
- プロトコル
 - HTTPS
- メソッド
 - GET
- パス
 - /job/{job_id}
- リクエストパラメータ

表 4-26 リクエストパラメータ

パラメータ	説明	値	必須
job_id	Job の ID	-	○

- レスポンス

表 4-27 レスポンス

ステータスコード	説明
200	取得に成功
401	認証エラー
403	認可エラー
404	指定したリソースが存在しない
500	処理エラー

```
{
  "id": 1,
  "type": "export_building",
  "status": "pending",
  "parameter": {
    "type": "export_building"
  },
  "result_parameter": {
    "type": "export_building"
  },
  "message": "string",
  "created_at": "2025-11-29T05:33:53.928Z",
  "updated_at": "2025-11-29T05:33:53.928Z"
}
```

4-4-4. 外部連携インターフェース

1) 【IF301】 PLATEAU-Ortho

- インタフェースの概要
 - 航空写真測量によって作成したオルソ画像をタイル化したもの
 - 高画質な屋根面のテクスチャを作成するための元データとして使用する
- 本インターフェースを利用する機能
 - 【FN001】
 - 【FN002】
 - 【FN003】
 - 【FN005】
- プロトコル
 - HTTPS
- API パス
 - <https://api.plateauview.mlit.go.jp/tiles/plateau-ortho-2023/{z}/{x}/{y}.png>
- メソッド
 - GET
- パラメータ
 - route パラメータ
 - ◇ x: タイルの X 座標
 - ◇ y: タイルの Y 座標
 - ◇ z: ズームレベル

4-5. 実証に用いたデータ

4-5-1. 利用したデータ一覧

1) 利用した 3D 都市モデル

- 年度：2024 年度（西暦）
- 都市名：京都市
- ファイル名：26100_kyoto-shi_city_2024_citygml_1_op
- メッシュ番号：52354601-52354602、52354611-52354612（インデックスマップで黄色囲いの箇所）

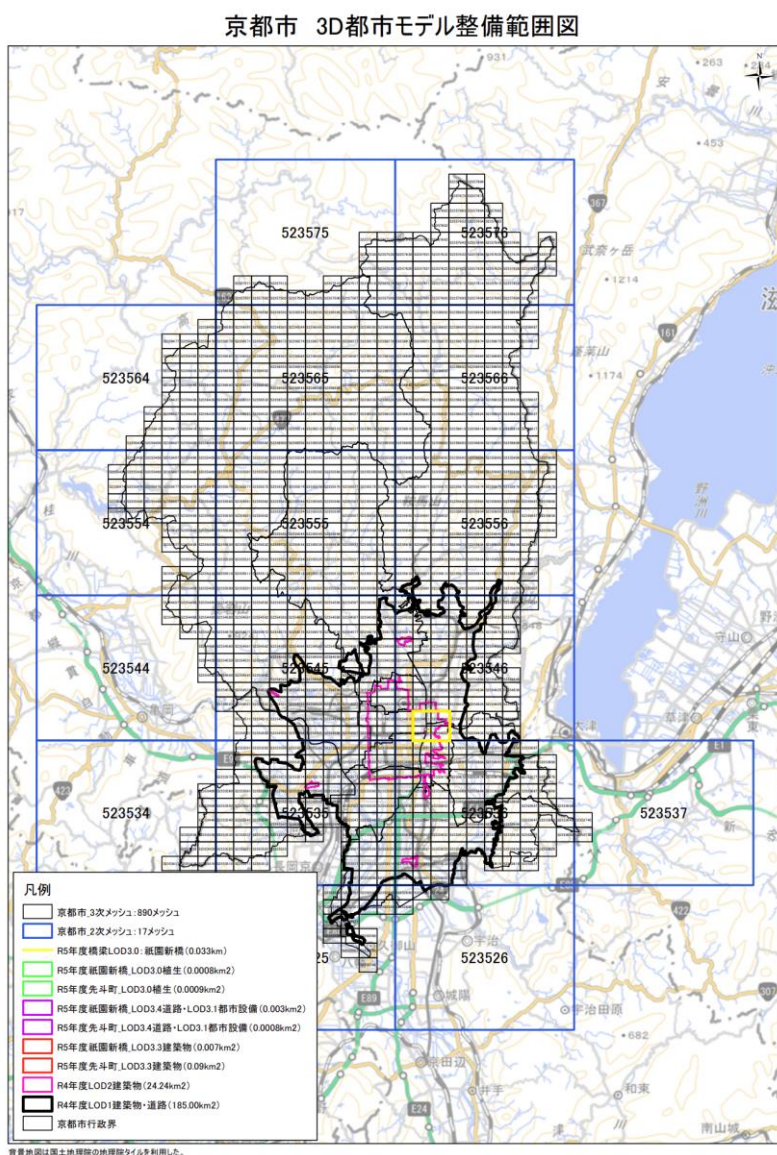


図 4-7 インデックスマップ（京都市）

- 年度：2023年度（西暦）
- 都市名：前橋市
- ファイル名：10201_maebashi-shi_city_2023_citygml_2_op
- メッシュ番号：54394064-54394065、54394074-54394075（インデックスマップでピンク色囲いの箇所）

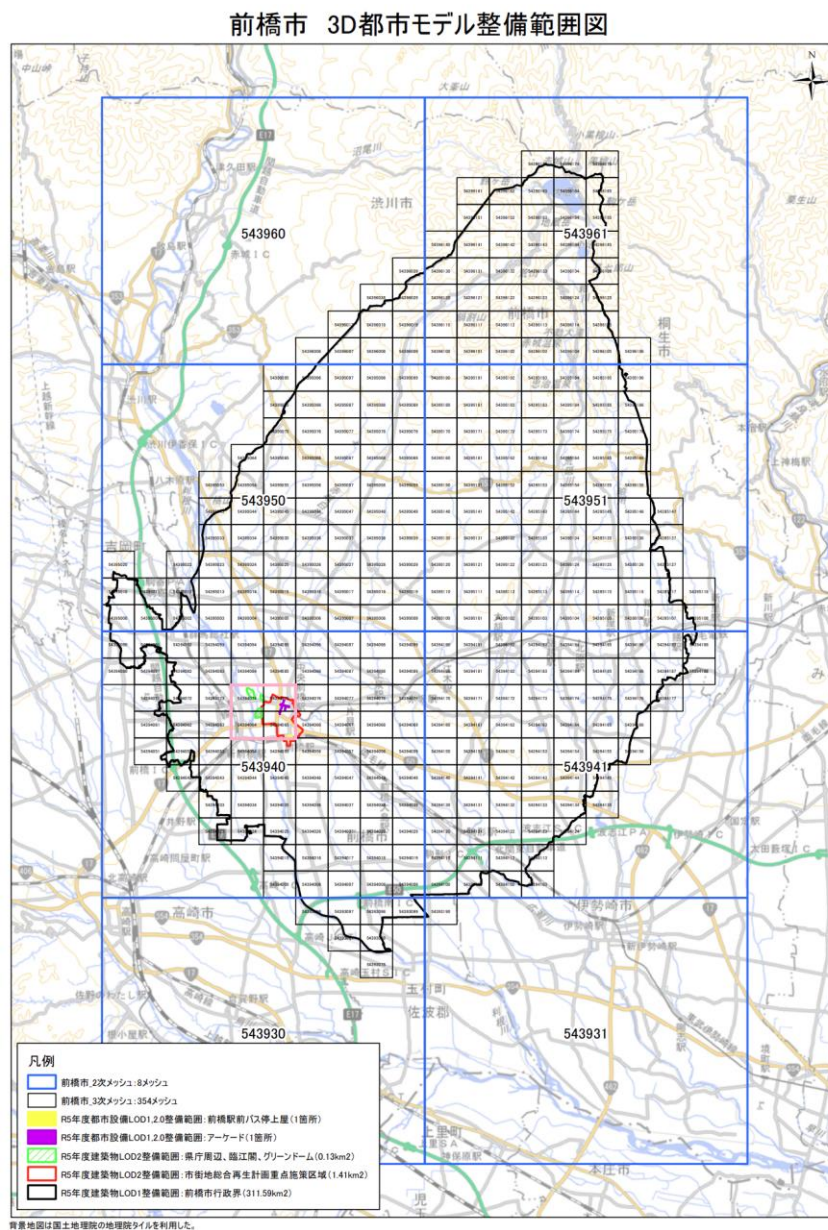


図 4-8 インデックスマップ（前橋市）

表 4-28 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 LOD2	bldg:Building	空間属性	bldg:lod2Solid	外形	FN002、FN003、FN005、FN104、 FN105、FN106、FN107、FN108、 FN109

2) 利用したデータ


1. データ一覧

表 4-29 利用するその他データ (一覧)

ID	データ名称	内容	データ形式	更新情報	出所	データを利用した機能 (ID)
DT101	PLATEAU-Ortho	航空写真測量によって作成・タイル化したオルソ画像	PNG 画像	前橋市 (2023) 京都市 (2022)	PLATEAU 配信サービス (plateau-streaming-tutorial)	FN105

2. データサンプル (イメージ)

表 4-30 利用するその他データ (サンプル)

ID	活用データ	サンプル・イメージ
DT101	PLATEAU-Ortho	 <p data-bbox="742 1160 1177 1189">PLATEAU-Ortho サンプル (京都市)</p>

4-5-2. 生成・変換するデータ

生成・変換したデータはなし。

4-6. ユーザーインターフェース

4-6-1. 画面一覧

今年度開発を行う 3D 都市モデル編集・可視化システムは、【HW002】管理者用 PC 上のブラウザの画面を操作して扱う。なお、いずれの画面においても自由に前工程の画面に戻れるようにする。

表 4-31 【HW002】管理者 PC 用画面一覧

ID	連携 (ID)	画面名	説明	機能 (ID)
SC001	SC002	タイトル画面	<ul style="list-style-type: none"> ● 本システムの URL にアクセス後、最初に表示する ● ボタンを押すことで、システムの利用ができる 	-
SC002	SC003	建物選択モーダル	<ul style="list-style-type: none"> ● 作業が可能な建物の一覧を表示する ● 作業対象の建物を選択すると、選択した建物に基づく面選択モーダル画面に進む 	FN101
SC003	SC004	面選択モーダル	<ul style="list-style-type: none"> ● 面の一覧を表示する ● 作業対象の面を選択すると、選択した面に基づくメイン画面に進む 	FN102
SC004	SC005 SC006	メイン画面	<ul style="list-style-type: none"> ● 撮影画像選択エリアとテクスチャ編集エリアで構成される画面 	-
SC005	SC004	撮影画像選択エリア	<ul style="list-style-type: none"> ● DB に保存されている撮影画像を表示する ● 画像を選択すると、サーバーにテクスチャ変換処理を依頼し、その結果をテクスチャ編集エリアに反映する 	FN103 FN104 FN105
SC006	SC004 SC007	テクスチャ編集エリア	<ul style="list-style-type: none"> ● テクスチャの面に対する位置調整や拡大縮小などを行う ● 確定ボタンを押すことで、DB に内容を反映し、次処理選択モーダル画面を表示する 	FN106 FN107 FN108
SC007	SC002 SC003 SC006	次処理選択モーダル	<ul style="list-style-type: none"> ● データベースの更新結果を表示する ● 「ダウンロード (建物/3 次メッシュ)」/「面選択から再処理」/「建物選択から再処理」/「終了」ボタンのいずれかを選択し、次処理を決める 	FN109

4-6-2. 画面遷移図

本システムにおける画面の遷移図を示す。

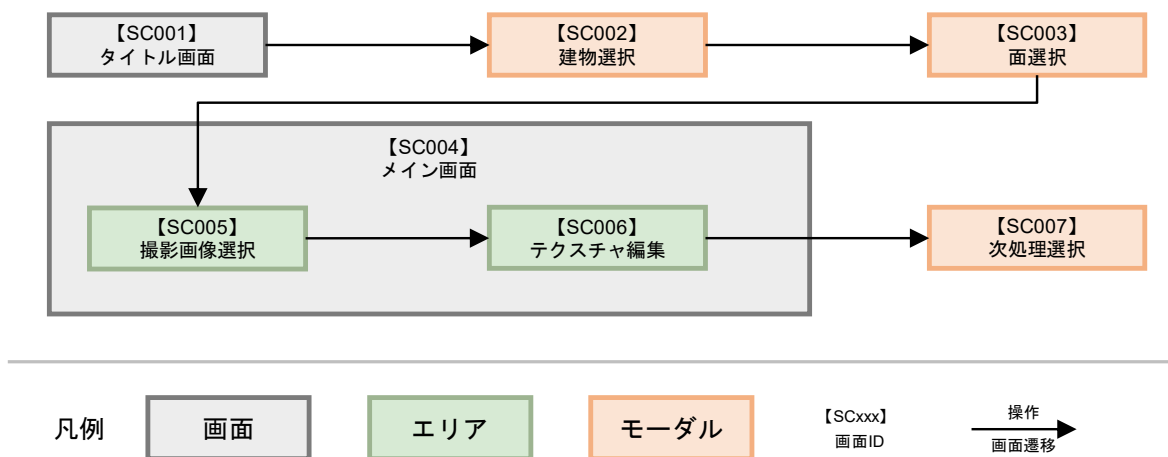


図 4-9 【HW002】 管理者用 PC の画面遷移図

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

本システムの利用の流れを図 4-10 に示す。

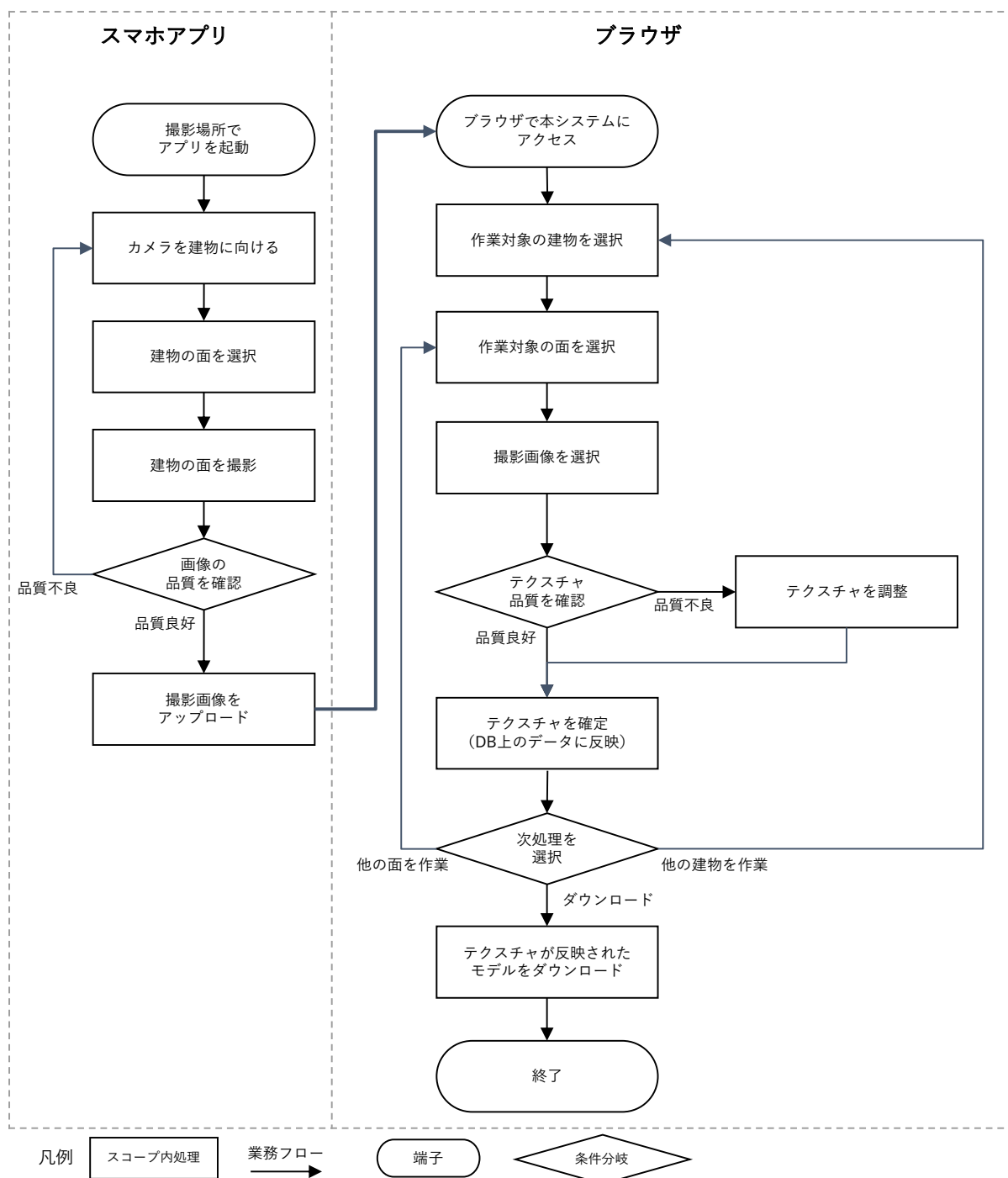


図 4-10 実証システムの利用フロー

4-7-2. 各画面操作方法

1. 撮影場所でアプリを起動

- メイン画面で「アプリ起動」ボタン（下図赤枠）を押す



図 4-11 メイン画面

2. カメラを建物に向ける

- 建物検出画面で撮影したい建物の面が画角に収まるようにカメラを向ける



図 4-12 建物検出画面（検出前）

3. 建物の面を選択

- 撮影対象の面（下図青面）をタップする



図 4-13 建物検出画面（検出後）

4. 建物の面を撮影

- 撮影対象の面がフォーカスされていることを確認し、「撮影」ボタン（下図赤枠）を押す



図 4-14 建物撮影画面

5. 画像の品質を確認

- 撮影した画像の品質を確認し、品質が不十分な場合や撮り直したい場合は、「再撮影」ボタン（下図青枠）を押す



図 4-15 撮影判定画面（再撮影）

6. 撮影画像をアップロード

- 撮影した画像の品質を確認し、品質が十分な場合は「登録」ボタン（下図赤枠）を押す



図 4-16 撮影判定画面（登録）

7. ブラウザで本システムにアクセス

- アクセス後に表示される画面の下部にある「起動する」ボタン（下図赤枠）を押す

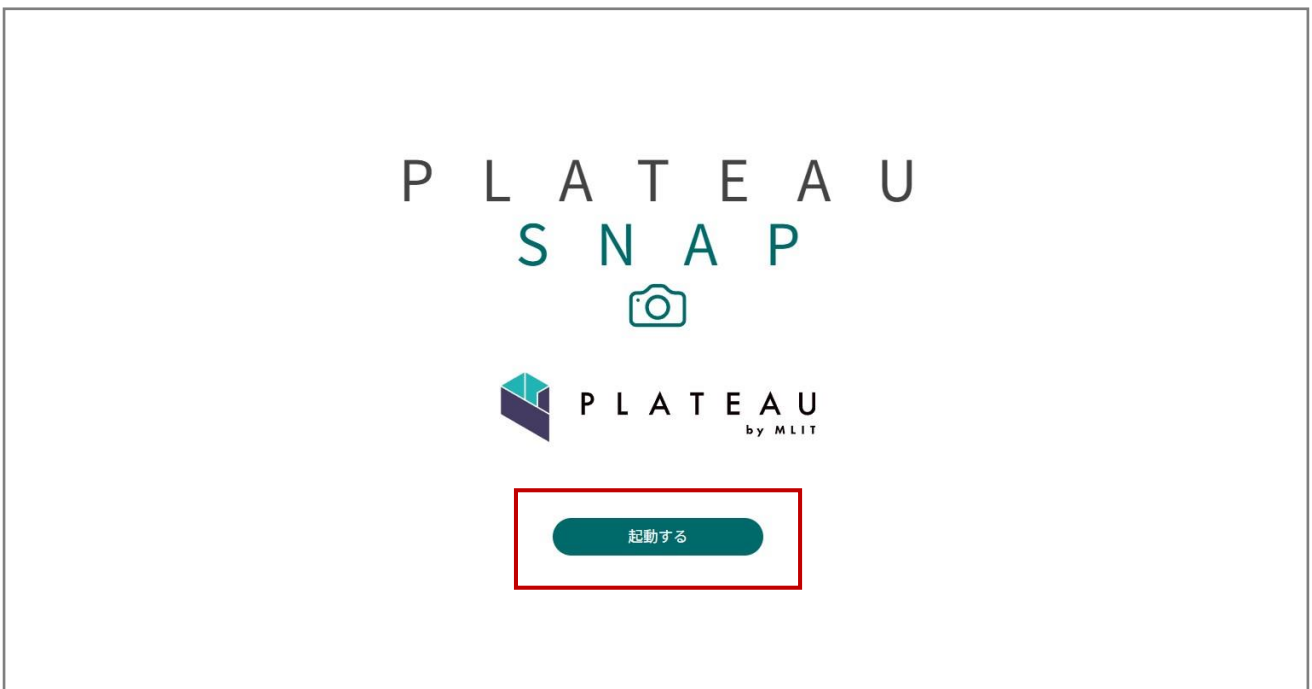


図 4-17 タイトル画面

8. 作業対象の建物を選択

- 建物の一覧から、作業したい建物（下図赤枠）を選択する



図 4-18 建物選択モーダル

9. 作業対象の面を選択

- 面の一覧から、作業したい面（下図赤枠）を選択する



図 4-19 面選択モーダル

10. 撮影画像を選択

- アプリで撮影した画像一覧から、テクスチャとして使用したい画像（下図赤枠）を選択する



図 4-20 撮影画像選択エリア

11. テクスチャ品質を確認

- テクスチャ編集エリア（下図赤枠）を確認し、品質を確認する

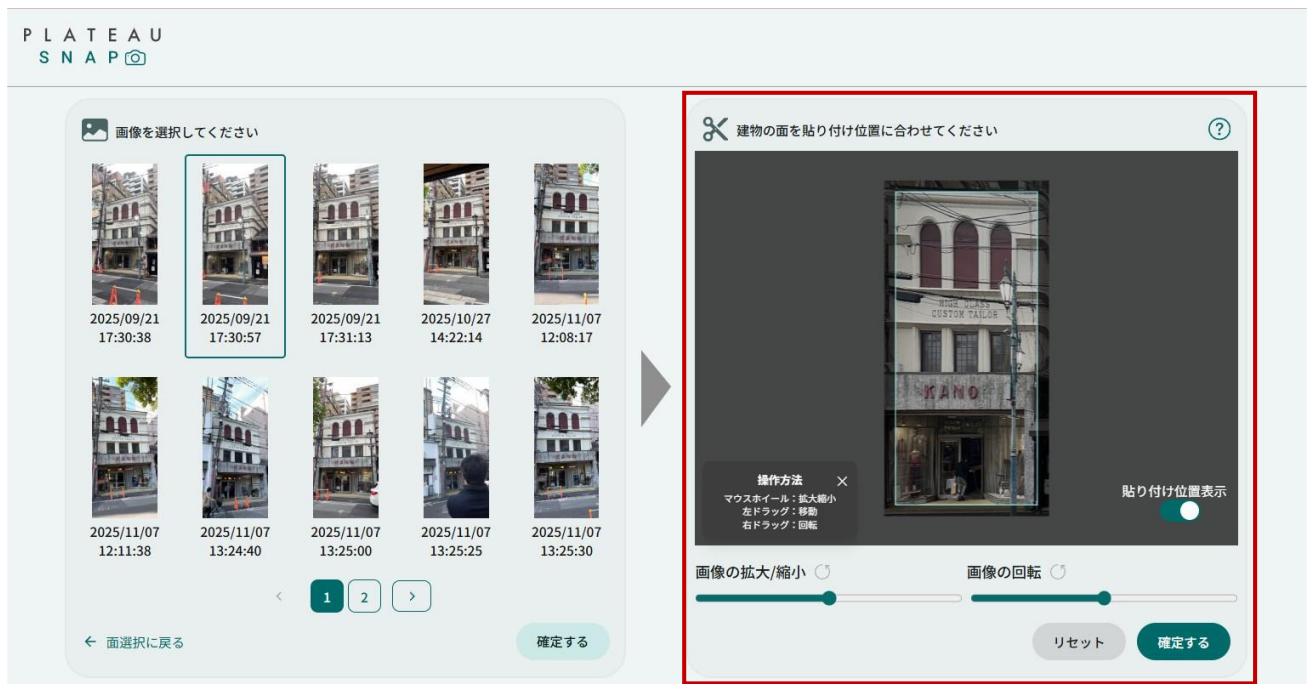


図 4-21 テクスチャ編集エリア（品質を確認）

12. テクスチャを調整（品質不良の場合）

- テクスチャ編集エリアで、画像を直接（下図赤枠）又は、バー（下図青枠）を用いて、位置や角度、サイズを調整する

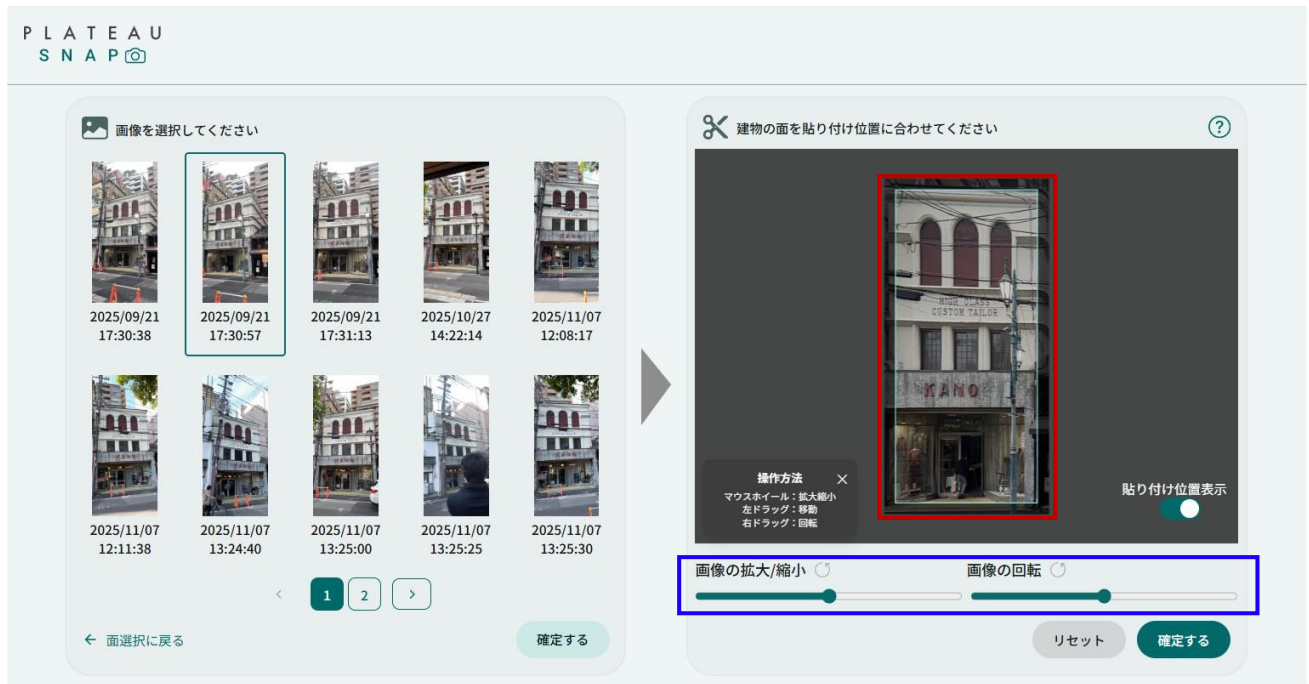


図 4-22 テクスチャ編集エリア（調整）

13. テクスチャを確定（DB 上のデータに反映）

- 「確定する」ボタン（下図赤枠）を押す

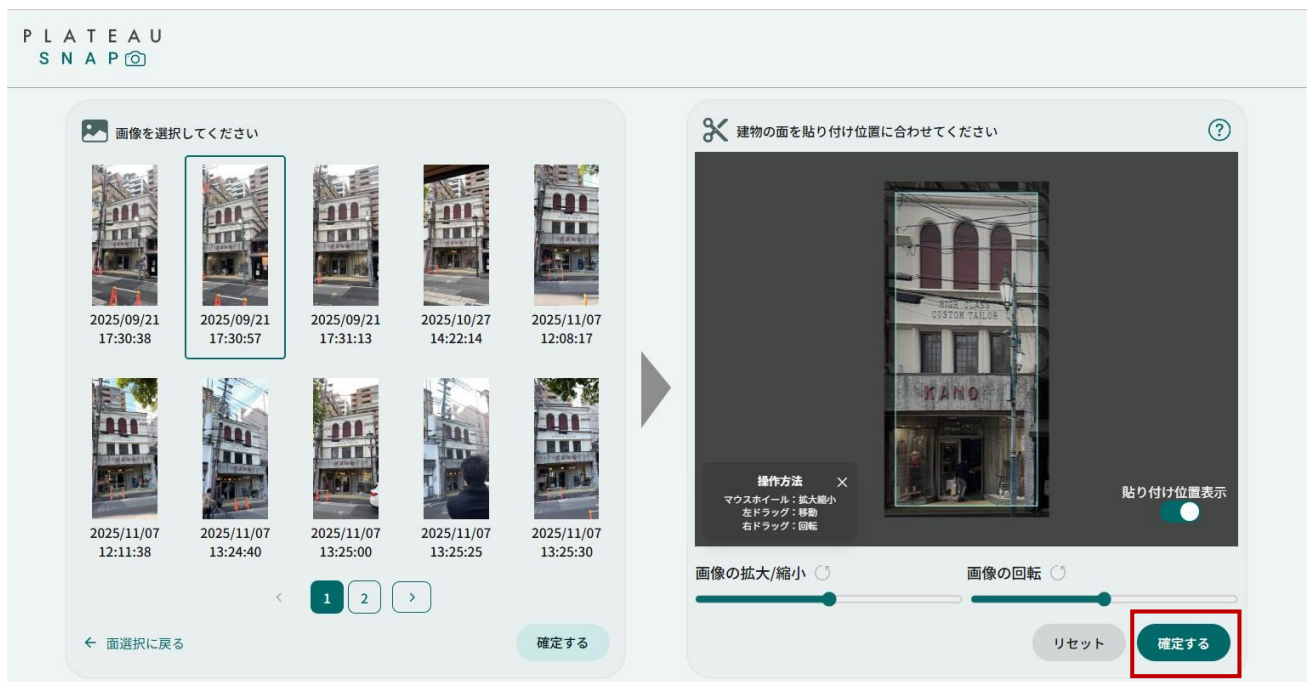


図 4-23 テクスチャ編集エリア（確定）

14. 次処理を選択

- モーダル上から次に行いたい処理（下図赤枠）を選ぶ

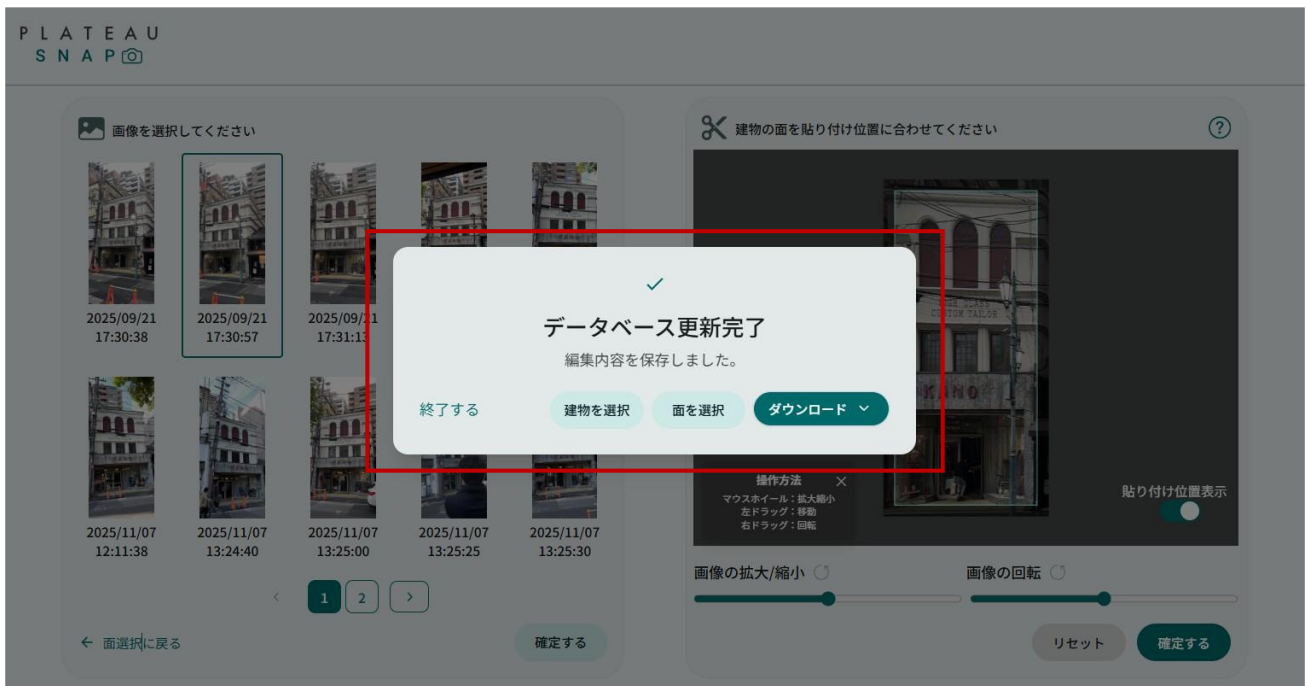


図 4-24 次処理選択モーダル

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	非機能項目	要件詳細
可用性	NR001	安定動作時間	<ul style="list-style-type: none"> 30 分以上の安定動作時間を確保すること
性能・拡張性	NR002	システムの処理実行速度	<ul style="list-style-type: none"> 各画面の表示を 15 秒以内で実施すること
有用性	NR003	ユーザー満足度	<ul style="list-style-type: none"> 想定ユーザーの 60%以上が本システムを「有用」と評価すること

1) 【NR001】安定動作時間

- 要件詳細
 - 30 分以上の安定動作時間を確保すること
- 本非機能要件の対象となるソフトウェア
 - 【SW002】 PLATEAU SNAP CMS
- 設定理由
 - 短時間の実証の間のみ安定稼働を要求されるため
 - 従来の業務フロー上、3D 都市モデルへのテクスチャ付与については、手作業では 30 分程度の工数を要している。今回の実証実験で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、複数の面や建物を対象とした作業が見込まれるため、30 分以上の安定動作時間を確保する
- 評価方法
 - 30 分のシステム稼働を 2 回行い、システムダウンやフリーズが発生しないことを確認する

2) 【NR002】システムの処理実行速度

- 要件詳細
 - 各画面の表示を 15 秒以内で実施すること
- 本非機能要件の対象となるソフトウェア
 - 【SW002】 PLATEAU SNAP CMS
- 設定理由
 - ユーザーの操作テンポや作業効率に支障を来さない範囲か確認するため
- 評価方法
 - 30 分のシステム稼働を通して、各画面の表示までに掛かる時間を確認する

3) 【NR003】 ユーザー満足度

- 要件詳細
 - 想定ユーザーの 60%以上が本システムを「有用」と評価すること
- 本非機能要件の対象となるソフトウェア
 - 【SW002】 PLATEAU SNAP CMS
- 設定理由
 - ニーズの存在と導入可能性を示す実用的な指標として、「有用」と感じるユーザーが多数派であるかを確認するため
- 評価方法
 - 実証実験においてアンケートやヒアリングを行い、3D 都市モデルへのテクスチャ付与が効率化できているかを確認する

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

対象システム	試験項目	確認内容	試験期間	アクティビティ
【FN106】 貼付け位置補正機能	画像の貼付け精度（位置ずれの有無）	60%以上の面に対して自動で貼付けが可能か	2025年 10～12 月	目視による検証
【SW002】 PLATEAU SNAP CMS	操作性	想定ユーザーの60%以上が「使いやすい」と評価するか	2025年 11～12 月	実証実験による検証
【FN108】 データベース更新機能、 【FN109】 データ出力機能	データ整合性 / CityGML との互換性	出力データがビューアで問題なく可視化が可能か	2025年 10～12 月	PLATEAU SDK などのビューアツールに出力データを取り込み、目視確認及びログ確認

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
【SW002】 PLATEAU SNAP CMS	安定動作時間	30 分以上の安定動作時間を確保するか	2025 年 10～12 月	テストによる検証
	システムの処理実行速度	各画面の表示を 15 秒以内で実施するか	2025 年 10～12 月	テストによる検証
	有用性（ユーザー満足度）	想定ユーザーの 60%以上が「有用」と評価するか	2025 年 11～12 月	実証実験による検証

7. 実証技術の機能要件の検証

7-1. 貼付け位置補正機能の検証

7-1-1. 検証目的

- 正射画像内の面の位置を自動で割り出し、貼り付ける際の位置ずれの精度を評価することにより、【FN106】貼付け位置補正機能の実用性を確認すること

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	画像の貼付け精度 (位置ずれの有無)	60%以上の面に対して 自動で貼り付け可能であること	<ul style="list-style-type: none"> ● 過半数の貼付けを自動化している場合、手動調整の負担を大幅に軽減できると判断したため 	<ul style="list-style-type: none"> ● 目視による検証

7-1-3. 検証方法と検証シナリオ

1. 画像の貼付け精度（位置ずれの有無）

- 検証に使用する全データに対して、手動補正が不要なずれのないデータの割合を、画像の貼り付け精度と定義する
- 検証シナリオに従い、様々なパターンで撮影した画像を精度検証に用いる
- 分母：検証に使用する全データ数
- 分子：手動補正が不要なずれのないデータ数

$$\text{画像の貼付け精度[\%]} = \frac{\text{ずれのないデータ数}}{\text{検証データ数}} \times 100$$

表 7-2 検証シナリオ一覧（画像の貼付け精度）

No.	検証方法	パターン	撮影方法
a-1	画像の貼付け精度	平屋	重要な視点場を中心として、異なる距離や角度から画像を撮影する
a-2		商業ビル	
a-3		大型商業施設	
a-4		タワー	
a-5		寺社	
a-6		教会	
a-7		駅	

7-1-4. 検証結果

貼付け位置補正機能（FN106）について、建物タイプ別に検証を実施した結果、検証データ全 12 件のうち、手動補正を行わずに貼付け可能であったデータは 8 件となり、貼付け精度は 66.7%となった。これは、KPI として設定した「60%以上の面に対して自動貼付けが可能」という目標値を下回る結果となり、本検証においては KPI 未達と判断された。

建物種別ごとの結果を見ると、平屋、商業ビル、大型商業施設など、外形が比較的単純で壁面構成が明確な建築物では、位置ずれのない貼付けが実現できた一方、タワーや寺社、駅といった建物では位置ずれが発生するケースが確認された。特に、傾斜屋根や多面構成を有する建築物、軒が深い建築物、円すい形に近い外形を持つ建築物では、正射変換を前提とした貼付け処理の制約により、貼付け位置の精度が安定しない傾向が見られた。

これらの結果から、貼付け位置補正機能は、建物条件が適合する場合には一定の実用性を有するものの、全ての建築物に対して一律に適用できる段階には至っていないことが確認された。本機能を実務で活用するためには、建物形状や構成面の複雑さに応じた適用条件の整理や補助的な手動調整を前提とした運用設計、及びアルゴリズムの改善が今後の課題として整理される。

なお、本検証では、地域ごとの建物種別の分布差により、建物種別ごとのデータ数に偏りが生じている。本検証結果を解釈する際には、これらのデータ取得上の制約を考慮する必要がある。

表 7-3 位置補正機能の検証結果サマリー

No	パターン	地域	建物名	画像 ID	位置ずれ
a-1	平屋	京都	セブンイレブン京都市役所西店	図 7-1	有り
		前橋	みのやタバコ店	図 7-2	有り
a-2	商業ビル	京都	ミーナ京都	図 7-3	有り
		前橋	アーツ前橋	図 7-4	無し
a-3	大型商業施設	京都	京都高島屋 S.C.	図 7-5	有り
a-4	タワー	京都	京都タワー	図 7-6	有り
a-5	寺社	京都	東本願寺御影堂門	図 7-7	有り
		前橋	虎淵山宝池院大蓮寺	図 7-8	有り
a-6	教会	京都	ハリストス教会	図 7-9	無し
		前橋	前橋カトリック教会	図 7-10	有り
a-7	駅	京都	三条駅	図 7-11	無し
		前橋	中央前橋駅	図 7-12	無し

表 7-4 KPI 検証結果

項目	件数	割合
検証データ総数	12 件	100%
位置ずれ無し	4 件	33.3%
位置ずれ有り	8 件	66.7%

表 7-5 検証結果のまとめ

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

項目	達成/未達
KPI 達成状況（60%以上）	未達

● 画像の貼付け精度



図 7-1 a-1 平屋（京都）位置ずれ有り



図 7-2 a-1 平屋（前橋）位置ずれ有り



図 7-3 a-2 商業ビル（京都）位置ずれ有り



図 7-4 a-2 商業ビル（前橋）位置ずれ無し



図 7-5 a-3 大型商業施設（京都）位置ずれ有り

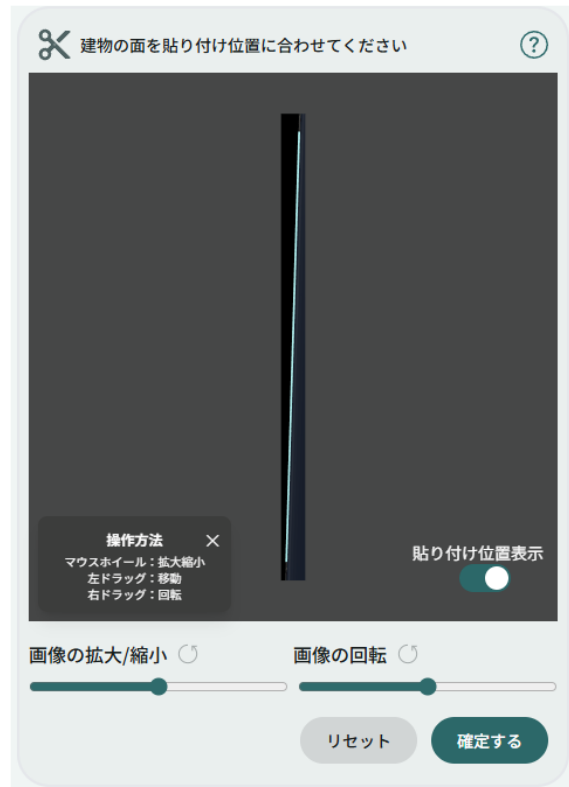


図 7-6 a-4 タワー（京都）位置ずれ有り



図 7-7 a-5 寺社（京都）位置ずれ有り



図 7-8 a-5 寺社（前橋）位置ずれ有り



図 7-9 a-6 教会（京都）位置ずれ無し



図 7-10 a-6 教会（前橋）位置ずれ有り



図 7-11 a-7 駅（京都）位置ずれ無し



図 7-12 a-7 駅（前橋）位置ずれ無し

7-2. 操作性の検証

7-2-1. 検証目的

- ユーザーが直感的に操作でき、効率的かつ満足度の高い体験を提供する UI/UX を実現し、システムの実用性を確認すること

7-2-2. KPI

表 7-6 ユーザビリティ評価の KPI

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	ユーザビリティ評価	想定ユーザーの 60%以上が「使いやすい」と評価すること	● 過半数のユーザーが「使いやすい」と評価する場合、簡素かつ明快な UI/UX となっていることを判断できるため	● 実証実験で被験者にアンケートを行い検証する

7-2-3. 検証方法と検証シナリオ

1) 検証方法と検証シナリオ

a. ユーザビリティ評価

- ユーザビリティに関する各設問の合計回答数に対して、使いやすいと評価された回答数の割合を、ユーザビリティ評価と定義する

分母：各設問の合計回答数

分子：各設問で使いやすいと回答した数

$$(\text{各設問の})\text{ユーザビリティ評価}[\%] = \frac{\text{使いやすいと回答した数}}{\text{合計回答数}} \times 100$$

7-2-4. 検証結果

ブラウザシステムの操作性について、被験者アンケートによる評価結果を基に検証を行った。その結果、「とても使いやすかった」「使いやすかった」と回答した割合は合計で 76.5%となり、KPI として設定した「想定ユーザーの 60%以上が『使いやすい』と評価すること」という目標値を上回る結果となった。このことから、本システムの操作性については、品質要件として求められる水準を満たしていると判断される。

一方で、「普通」との回答も 23.5%見られ、全ての利用者が直感的に操作できたわけではないことも確認された。操作に一定の慣れを要する場面が存在することが示唆されており、操作の分かりやすさや作業効率の更なる向上に向けた改善余地が残されている。

以上から、ブラウザシステムの操作性は、実務利用を想定した場合に一定の有用性を有していると評価できる一方、より幅広い利用者層にとっての使いやすさを確保するためには、UI/UX の改善を継続的に行うことが望ましい。

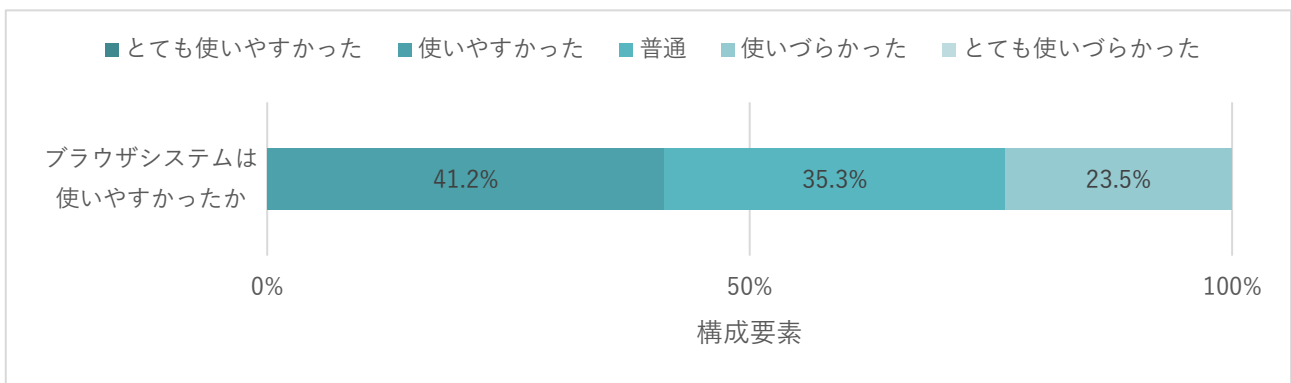


図 7-13 被験者による PLATEAU SNAP ブラウザシステムの使いやすさに関するアンケート結果 (n=17)

7-3. データ整合性/CityGML との互換性の検証

7-3-1. 検証目的

- 出力データの整合性を評価することで、テクスチャ更新の実用性を確認すること

7-3-2. KPI

表 7-7 データ整合性/CityGML との互換性検証の KPI

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	データ整合性 (CityGML との互換性)	出力データの 80%以上が可 視化可能であ ること	● 80%以上の出力データが 問題なく可視化できる場 合、整合性があると判断 できるため	● PLATEAU SDK などのビュー アツールに出力データを取 り込み、目視確認及びログ 確認を行う

7-3-3. 検証方法と検証シナリオ

b. データ整合性

- 検証に用いた全てのデータ数に対して、ビューアで問題なく可視化できたデータ数の割合を、データ整合性と定義する

分母：検証の全データ数

分子：可視化可能なデータ数

$$\text{データ整合性[\%]} = \frac{\text{可視化可能なデータ数}}{\text{検証の全データ数}} \times 100$$

7-3-4. 検証結果

データ整合性の検証では、テクスチャ更新後に出力された CityGML データについて、PLATEAU Builder を用いた可視化確認を行い、問題なく表示・操作が可能であるかを評価した。検証対象は全 12 件とし、このうちタワー建築 1 件については、PLATEAU SNAP でのテクスチャ生成及びデータダウンロードができなかったため、可視化を行うことができず、可視化不可（×）として判定した。

その結果、可視化が可能であったデータは 11 件となり、可視化可能なデータの割合は約 91.7%となった。この値は、KPI として設定した「80%以上のデータが問題なく可視化可能であること」を満たしており、データ整合性に関する品質要件は達成されたと判断される。

また、PLATEAU Builder による可視化確認に加え、PLATEAU SDK においても同データの読み込みを行い、問題なく処理できることを確認した。これにより、本システムで生成・更新されたデータは、各種ツールにおいて互換性を保って取り扱えることが確認された。

以上より、タワー建築 1 件を除き、テクスチャ自動付与後の CityGML データは、可視化及び後続処理の観点から実務利用に耐える整合性を有していることが確認された。一方で、タワー建築については、データ生成及び可視化が行えなかったことから、今後の運用及び検証に向けた課題として整理される。

なお、本検証では、地域ごとの建物種別の分布差により、建物種別ごとのデータ数に偏りが生じている。本検証結果を解釈する際には、これらのデータ取得上の制約を考慮する必要がある。

表 7-8 貼付け位置の正確性の検証結果サマリー

No	パターン	地域	建物名	画像 ID	可視化
a-1	平屋	京都	セブンイレブン京都市役所西店	図 7-14	○
		前橋	みのやタバコ店	図 7-15	○
a-2	商業ビル	京都	ミーナ京都	図 7-16	○
		前橋	アーツ前橋	図 7-17	○
a-3	大型商業施設	京都	京都高島屋 S.C.	図 7-18	○
a-4	タワー	京都	京都タワー	-	×
a-5	寺社	京都	東本願寺御影堂門	図 7-19	○
		前橋	虎淵山宝池院大蓮寺	図 7-20	○
a-6	教会	京都	ハリストス教会	図 7-21	○
		前橋	前橋カトリック教会	図 7-22	○
a-7	駅	京都	三条駅	図 7-23	○
		前橋	中央前橋駅	図 7-24	○

表 7-9 KPI 検証結果

項目	件数	割合
検証データ総数	12 件	100%
問題なく可視化できた件数	11 件	91.7%
可視化できなかった件数	1 件	8.3%

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

表 7-10 検証結果のまとめ

項目	達成/未達
KPI 達成状況 (60%以上)	達成

- データ整合性/CityGML との互換性

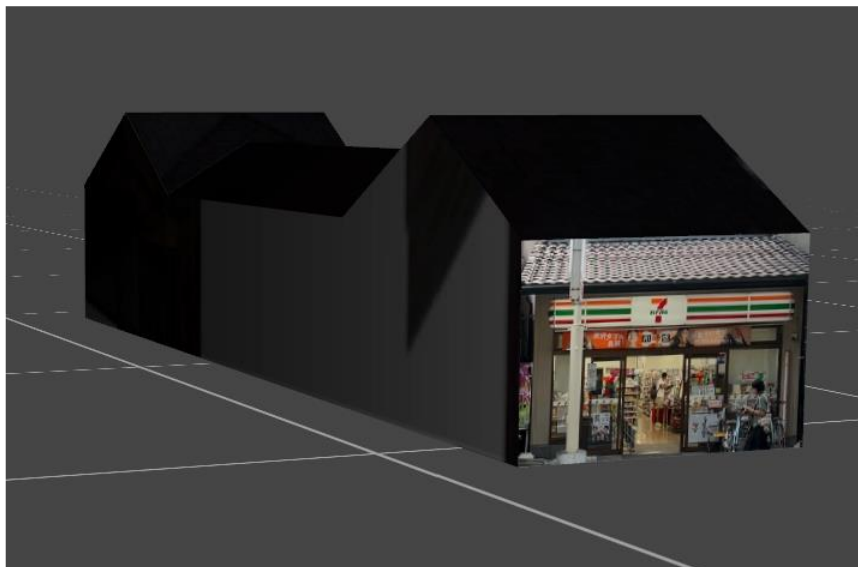


図 7-14 a-1 平屋 (京都) PLATEAU Builder で可視化

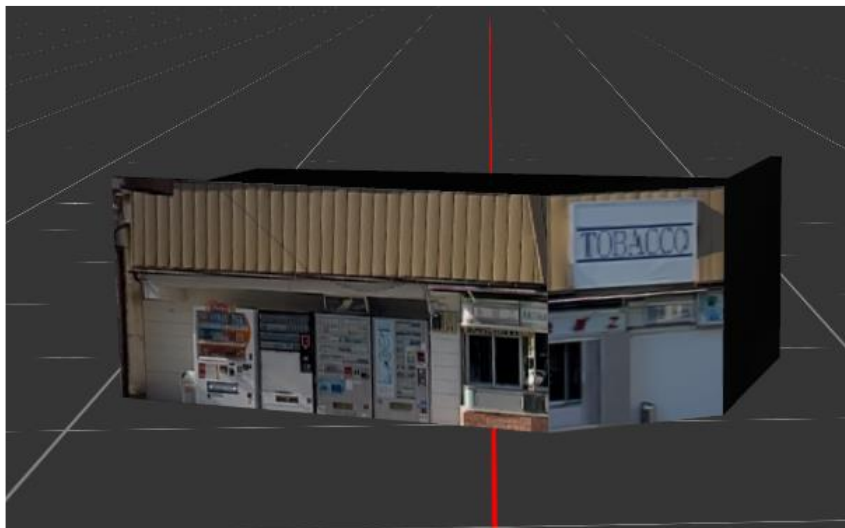


図 7-15 a-1 平屋（前橋） PLATEAU Builder で可視化



図 7-16 a-2 商業ビル（京都） PLATEAU Builder で可視化



図 7-17 a-2 商業ビル（前橋）PLATEAU Builder で可視化



図 7-18 a-3 大型商業施設（京都）PLATEAU Builder で可視化

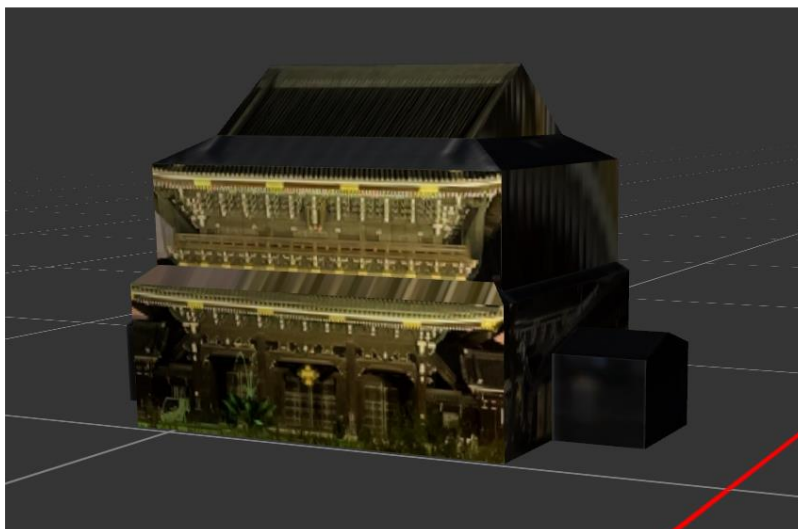


図 7-19 a-5 寺社（京都）PLATEAU Builder で可視化



図 7-20 a-5 寺社（前橋）PLATEAU Builder で可視化

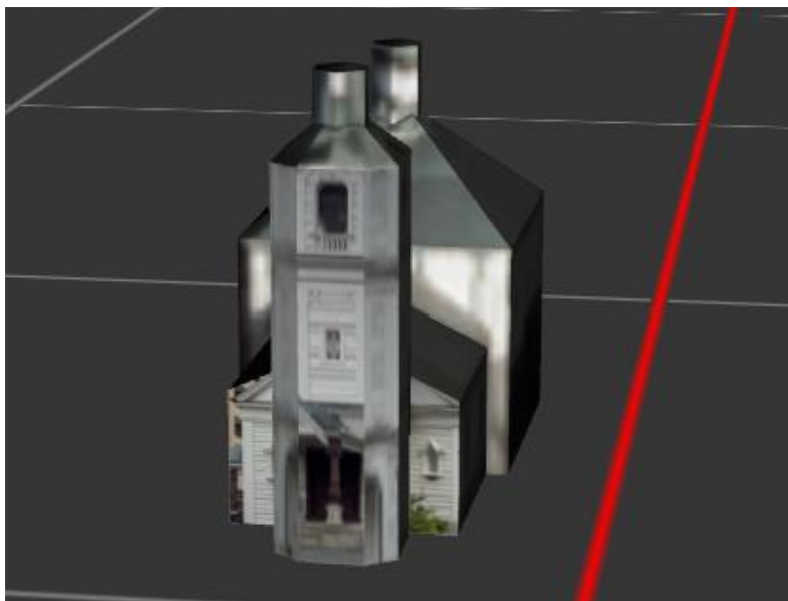


図 7-21 a-6 教会（京都）PLATEAU Builder で可視化

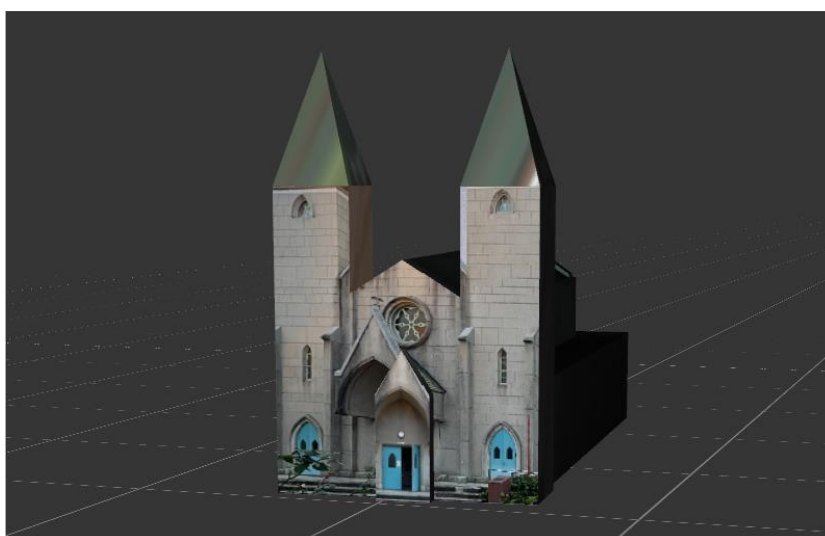


図 7-22 a-6 教会（前橋）PLATEAU Builder で可視化



図 7-23 a-7 駅（京都）PLATEAU Builder で可視化

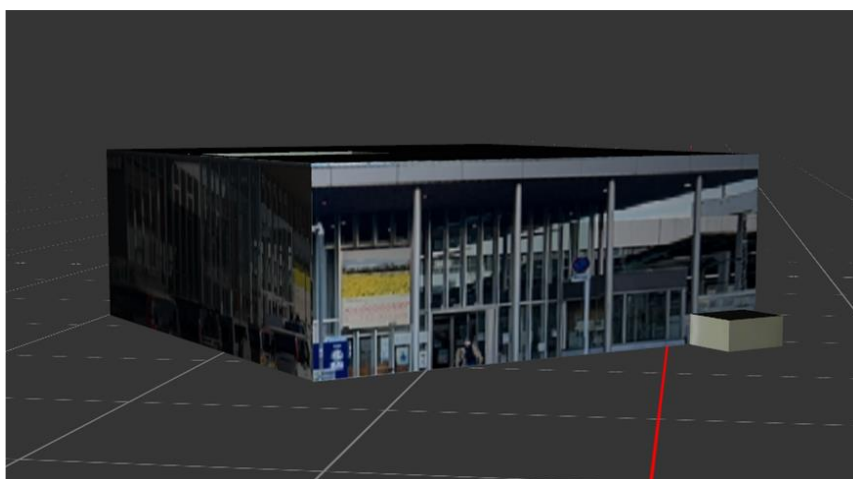


図 7-24 a-7 駅（前橋）PLATEAU Builder で可視化

8. 実証技術の非機能要件の検証

8-1. 検証目的

- 本検証では、非機能要件に基づき、性能・拡張性の観点から本システムの有用性を評価する。具体的には、標準的な動作環境への対応を検証し、実運用における適用可能性と拡張性を確認する

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	安定動作時間	● 30 分以上の安定動作時間を確保すること
性能・拡張性	NR002	システムの処理実行速度	● 各画面の表示を 15 秒以内で実施すること
有用性	NR003	ユーザー満足度	● 想定ユーザーの 60%以上が本システムを「有用」と評価すること

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
可用性	安定動作時間	● 30 分以上の安定動作時間を確保できること	2025 年 10～12 月	● テストによる検証
性能・拡張性	システムの処理実行速度	● 各画面の表示を 15 秒以内で実施できること	2025 年 10～12 月	● テストによる検証
ユーザー検証・活用シナリオ評価	有用性（ユーザー満足度）	● 想定ユーザーの 60%以上が「有用」と評価すること	2025 年 11～12 月	● 実証実験による検証

8-2-2. 検証結果

非機能要件の検証の結果、可用性及び処理実行速度については、いずれも想定どおり安定した動作が確認され、業務利用に支障のない水準であることが確認された。また、ユーザー検証においても本システムの有用性は高く評価されており、非機能要件全体として、実務利用を見据えた基本的な品質は確保されていると判断される。

表 8-3 検証結果サマリー

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

検証内容	評価指標・KPI	目標値	結果	示唆
可用性	安定動作時間	30 分以上	30 分	実務利用を想定した継続的な運用においても、一定の安定性を確保できると考えられる
性能・拡張性	システムの処理実行速度	15 秒以内	最大 10 秒程度 (データ出力)	ストレスなく利用できる水準の処理性能を有していると考えられる
ユーザー検証・活用シナリオ評価	有用性(ユーザー満足度)	60%以上	88.2%	LOD2 建築物モデルへのテクスチャ付与システムに一定の有用性が確認された

1) 可用性

動作環境については、30分の連続稼働テストを実施し、安定した動作を確認した。

2) 性能・拡張性

各機能の処理が想定時間内に完了した。

3) ユーザー検証・活用シナリオ評価

ユーザー検証の結果、本システムを用いた LOD2 建築物モデルへのテクスチャ付与について、「非常に有用である」「有用である」と評価した割合は合計で 88.2% となり、KPI として設定した「60%以上が有用と評価する」という目標値を大きく上回る結果となった。これにより、ユーザー視点において本システムの有用性は十分に確認されたと判断される。

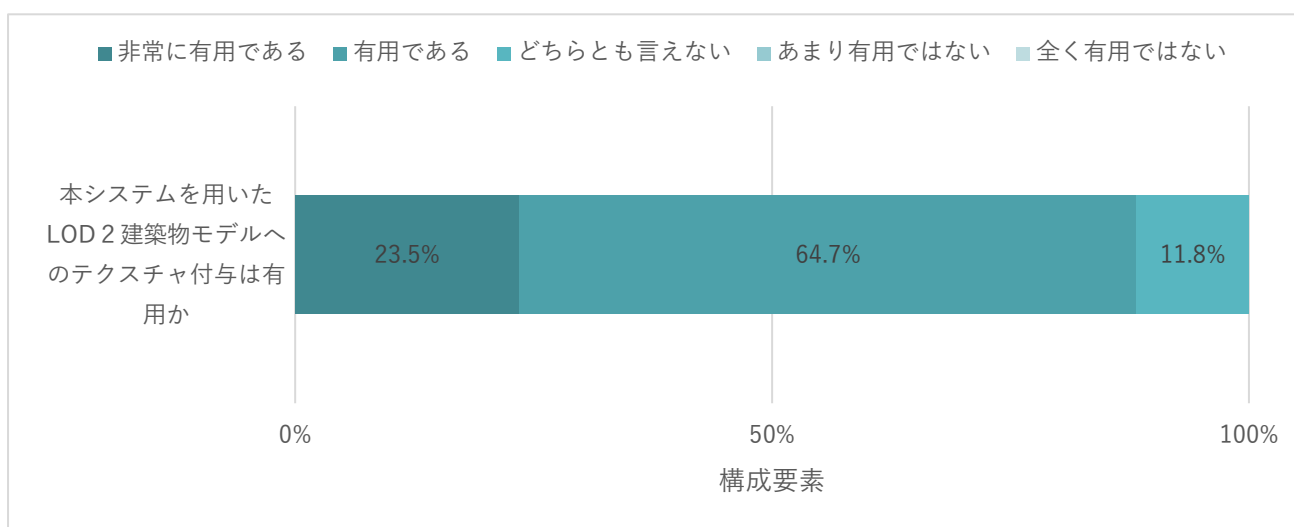


図 8-1 ユーザー検証・活用シナリオ評価のアンケート結果 (n=17)

9. 公共政策面の有用性検証

9-1. 検証目的

実証仮説に基づき、以下の検証目的を設定する。

【クラウドソーシング型基盤の応用による画像活用の大衆化】

本システムは、令和6年度に開発された PLATEAU SNAP のクラウドソーシング型 3D 都市モデル作成基盤を応用し、自治体職員や一般ユーザーが撮影した高画質な建物画像を建築物モデルのテクスチャとして活用する仕組みである。都市空間の実写画像を公共データとして活用することは従来、撮影条件や正確な位置情報の管理などに専門性を要し、限定的な運用にとどまっていた。しかし、PLATEAU SNAP で確立されたスマホアプリによる都市情報収集スキームを基とすることで、自治体職員や一般ユーザーが収集した画像を都市モデル整備に直接反映させる仕組みを確立できると仮定する。これにより、高精度なテクスチャ画像の収集・運用が日常かつ持続的に行えるようになる。

【テクスチャ作成工程の自動化による作業負荷軽減】

令和6年度に開発した PLATEAU SNAP におけるテクスチャ作成は、画像の選定、整形、位置合わせ、切り抜き、貼付けなど多くの工程で手作業が必要であり、高度なスキルと多くの工数を要するプロセスであった。本開発では、これら一連の作業をシステム上で自動化し、ユーザーは最小限の操作で高品質なテクスチャ付与を行えることを目指す。仮説として、画像と建物面の幾何情報に基づいた正射変換、自動トリミング、位置補正等のアルゴリズムを統合することで、作業の専門性を下げ、効率を大幅に向上できると考える。結果として、想定ユーザーである自治体や民間事業者が、専門的な知識なしに高解像度の都市景観データを構築できるようになる。

【自治体・民間事業者による実務利用の促進と関与意欲の向上】

本開発により、高画質なテクスチャを持つ 3D 都市モデルが簡易に、かつ現実感をもって生成可能となることで、これまで「リアリティに欠け、活用しづらい」とされてきた課題が解消され、実務への導入が促進されると仮定する。景観シミュレーション、防災訓練、観光プロモーション、不動産開発説明など、視覚的説得力が求められる多様な現場での活用が進むと期待される。また、自治体職員自らが画像提供やモデル作成工程に関与できることで、技術的な理解や関心が高まり、都市モデル整備に対する主体的な取り組みが促される。結果として、都市デジタルツインの社会実装が加速されるとの仮説を置く。

上記の実証仮説に基づき、以下の観点について有用性の評価を行う。

- 精度
 1. テクスチャ貼付け位置の自動補正は正確か
 2. 出力した CityGML データのテクスチャ貼付け位置は正確か
- ユーザビリティ
 3. 操作方法は分かりやすいか
 4. 画面表示は分かりやすいか
 5. データ出力までの作業時間は実用的な水準か
- ユーザー検証・活用シナリオ評価
 6. 3D 都市モデルへのテクスチャ付与を効率化できているか
 7. 高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるか
 8. LOD3 建築物モデルと比較して、LOD2+SNAP によるテクスチャ付与モデルは有用か

9-2. 検証方法

被験者に本システムを体験してもらい、本システムの使いやすさや分かりやすさへの満足度等に関するアンケートを実施する。

アンケートの実施方法

- 会場：自治体の会議室
- 機材：体験用に以下のスペックのスマートフォンと PC を用意する。
 - スマートフォン
 - ◇ 品番：Apple 社 iPhone 13 Pro
 - ◇ CPU 及び GPU：A15 Bionic
 - ◇ メモリ：6GB
 - ◇ OS：iOS 18.3.1
 - ◇ 通信環境：自社で用意する Wi-Fi ルーター
 - PC
 - ◇ 品番：ASUS 社 FA401UM-R7R5060
 - ◇ CPU：AMD Ryzen 7 260
 - ◇ GPU：NVIDIA GeForce RTX 5060
 - ◇ メモリ：32GB
 - ◇ ストレージ (SSD)：1TB
 - ◇ OS：Windows 11 Home
 - ◇ 通信環境：自社で用意する Wi-Fi ルーター

9-3. 被験者

表 9-1 被験者リスト

分類	具体名称	部署	役職	担当業務	人数
自治体	京都府京都市	都市計画局建築指導部建築指導課	企画基準係長	● 建築の許可	1名
		都市計画局都市景観部風致保全課	-	● 課の DX 担当	1名
		都市計画局まち再生・創造推進室	-	● まちづくり/建築、都市計画	3名
		都市計画局公共建築部公共建築企画課	-	● 公共施設管理や計画検討	1名
	群馬県前橋市	都市計画部都市計画課	課長 副主幹 主事 係長	● 都市計画関連業務 ● 景観審査業務	4名
		都市計画部市街地整備課	主事	● 官民連携まちづくりの推進	2名
民間事業者	パスコ	中央事業部空間情報コンサルタント室 DX 推進課	課長	● 3D 都市モデル整備 他	1名
		技術戦略本部空間情報処理センター空間情報二部信越国土情報課	-	● GIS モデリング業務	1名
	国際航業	事業統括本部地理空間基盤技術部 3D 都市モデルプロジェクト推進 G	主任技師 技師	● 3D 都市モデル整備 他	2名
		事業統括本部東日本支社群馬営業所	所長	● 3D 都市モデル企画・提案 他	1名

9-4. ヒアリング・アンケートの詳細

9-4-1. アジェンダ・タイムテーブル

表 9-2 アジェンダ・タイムテーブル

No.	アジェンダ	所要時間
1	実験の目的を説明	10分
2	スマホアプリの作業手順のデモ、説明	10分
3	移動	10分
4	操作の体験①（スマホアプリでの撮影、AR表示の確認）	30分
5	移動	10分
6	ブラウザシステムの作業手順のデモ、説明	10分
7	操作の体験②（ブラウザシステムでの操作の確認）	30分
8	品質の確認	10分
9	アンケート回答	10分

9-4-2. アジェンダの詳細

表 9-3 アジェンダの詳細

No	アジェンダ（再掲）	内容
1	実験の目的を説明	<ul style="list-style-type: none"> ● 本実証実験でアプローチする課題や背景の説明をする ● 本実証実験の比較対象となる従来の手法（専門業者による整備方法の例）の説明をする ● 本実証実験で用いるシステムの提供価値の説明をする ● システムの全体像の説明をする
2	スマホ作業手順のデモ、説明	<ul style="list-style-type: none"> ● スマホアプリの機能説明、デモ、操作説明をする ● テクスチャ取得対象の建築物についての説明をする
3	移動	<ul style="list-style-type: none"> ● 実証エリアに移動する
4	操作の体験①	<ul style="list-style-type: none"> ● スマホアプリで建築物を撮影、AR表示、操作性を確認してもらう
5	移動	<ul style="list-style-type: none"> ● 会議室に移動する
6	ブラウザシステムの作業手順のデモ、説明	<ul style="list-style-type: none"> ● ブラウザシステムの機能説明、デモ、操作説明する
7	操作の体験②	<ul style="list-style-type: none"> ● ブラウザシステムでテクスチャ生成の操作性を確認してもらう
8	品質の確認	<ul style="list-style-type: none"> ● 既存の3D都市モデル（LOD2/LOD3）の建築物テクスチャと本システムで生成したテクスチャを比較し、品質を確認してもらう
9	アンケート回答	<ul style="list-style-type: none"> ● Microsoft forms アンケートでアンケートの回答を依頼する ● その場で回答してもらい、送信を依頼する

9-4-3. 検証項目と評価方法

精度、ユーザビリティ評価、及びユーザー検証・活用シナリオ評価を検証項目とし、それぞれ定量・定性的に評価する。詳細把握のために、以下の観点を検証可能なアンケートを別途作成した。

表 9-4 検証項目と評価方法

検証ポイント	No	検証項目	定量評価	定性評価
1) 精度	1	テクスチャ貼付け位置の自動補正は正確か	<ul style="list-style-type: none"> ● 被験者に本年度開発したシステムを体験していただいた後、アンケートを実施する ● 選択肢は「とても不満」を1、「とても満足」を5とした5段階で設定する ● 回答を集計し、各選択肢の選択率から評価する（各設問で、60%以上が4以上の回答を目標とする） 	● アンケートの各設問に自由記入欄を設定する
	2	出力した CityGML データのテクスチャ貼付け位置は正確か		
2) ユーザビリティ	3	操作方法は分かりやすいか		
	4	画面表示は分かりやすいか		
	5	データ出力までの作業時間は実用的な水準か		
3) ユーザー検証・活用シナリオ評価	6	3D 都市モデルへのテクスチャ付与を効率化できているか		
	7	高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるか		
	8	LOD3 建築物モデルと比較して、LOD2 建築物モデル+SNAP によるテクスチャ付与モデルは有用か		

9-4-4. 実証実験の様子

スマホアプリで建物を撮影している様子。



図 9-1 スマホアプリで撮影している様子（京都）

スマホアプリで建物を撮影している様子。



図 9-2 スマホアプリで撮影している様子（京都）

スマホアプリで建物を撮影している様子。



図 9-3 スマホアプリで建物を撮影している様子（京都）

スマホアプリで建物を撮影している様子。



図 9-4 スマホアプリで建物を撮影している様子（前橋）

スマホアプリで建物を撮影している様子。



図 9-5 スマホアプリで建物を撮影している様子（前橋）

会議室でブラウザシステムの操作手順を説明している様子。



図 9-6 会議室でブラウザシステムの操作手順を説明している様子（前橋）

9-5. 検証結果

本章における検証結果から、本プロジェクトで開発した 3D 都市モデルへのテクスチャ自動付与技術は、建物条件が適合する場合において、都市モデル整備の効率化及び利活用促進に資する一定の有用性を有していることが確認された。特に、外形が比較的単純で平坦な壁面を有する建築物では、自動補正及びテクスチャ貼付け位置の正確性が安定しており、専門的な知識を有しない自治体職員や関係者であっても、実務で活用可能な水準に達していると評価できる。

一方で、軒の深い建築物や傾斜屋根を含む建物、円すい形や多面構成を有する大規模建築物など、形状や構成面が複雑な建築物に対しては、自動補正や貼付け位置の精度にばらつきが生じるケースが確認された。これらは正射変換を前提とした処理手法に起因する制約によるものであり、建物形状や規模に応じた適用範囲を整理した上での運用が、社会実装に向けた前提条件となる。

また、ユーザビリティの観点では、基本的な操作フローについては一定の理解が得られたものの、面選択や位置合わせにおける視認性、処理時間に関する改善要望も確認された。これらの点を改善することで、業務現場における利用ハードルをさらに下げ、日常的な都市モデル更新や利活用へとつなげていくことが可能になると考えられる。

さらに、ユーザー検証を通じて、高画質なテクスチャを付与した 3D 都市モデルは、景観シミュレーション、中心市街地の再整備検討、住民説明会やイベント時の事前検討など、視覚的な理解が求められる場面において有効であるとの評価が得られた。LOD3 モデルまでの整備が難しい場合においても、見た目を重視したユースケースにおいて本技術を補完的に活用することで、都市デジタルツインの社会実装を段階的に進めることが可能である。

以上より、本技術は全ての建築物に対して一律に適用するものではなく、建物条件や利用目的に応じて適切に使い分けることで、自治体や民間事業者による 3D 都市モデルの整備・活用を現実的に前進させる技術であると位置づけられる。

1) 精度

本システムの精度検証として、自動補正の正確性およびテクスチャ貼り付け位置の正確性について検証を行った。その結果、全建物を対象とした集計では、「非常に正確」「おおむね正確」と評価された割合がいずれも過半を占めており、本システムによる自動補正およびテクスチャ貼り付け処理は、全体として一定の妥当性を有していることが確認された。一方で、「どちらとも言えない」や「ずれている」との評価も一定数存在しており、すべての建物に対して一様に高い精度が得られているわけではないことも示された。

これまでの建物種別ごとの検証結果を踏まえると、自動補正及びテクスチャ貼付け位置の精度は、建物の形状や規模、構成面の複雑さによって大きく影響を受ける傾向が見られた。具体的には、外形が比較的単純で平坦な壁面を有する建物では安定した結果が得られやすい一方、軒が深い建築物、円すい形や多面構成の建物、傾斜屋根を含む建築物などでは、正射変換を前提とした処理の制約により、補正精度や貼付け位置の評価が低下するケースが確認された。

これらの結果から、本システムは建物条件によって効果に差が生じるものの、適用条件が合致する建築物においては、実務での活用に耐えうる精度を有していると評価できる。一方で、形状が複雑な建物への適用については、今後のアルゴリズム改善や補助的な編集機能の拡充が課題として整理される。

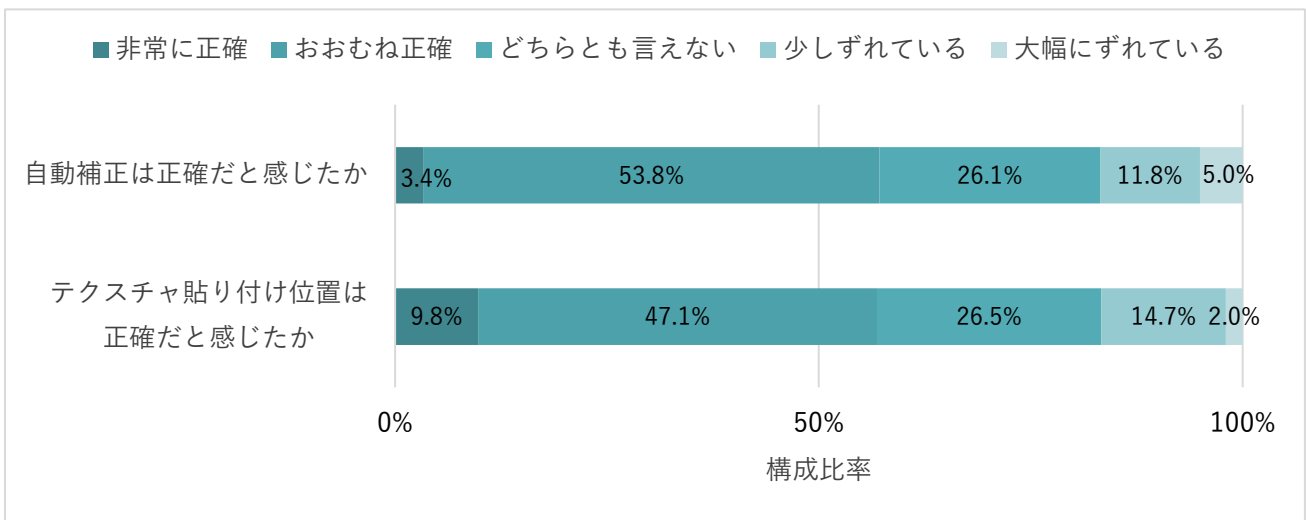


図 9-7 被験者による精度及び正確性に関するアンケート結果 (n=17)

1. 自動補正は正確だったか

● 平屋

平屋を対象とした自動補正では、「おおむね正確」と評価された割合が両エリアで最も高く、建物形状が単純なケースにおいて自動補正手法が有効に機能していることが示された。



図 9-8 平屋（京都）の自動補正のキャプチャ



図 9-9 平屋（前橋）の自動補正のキャプチャ

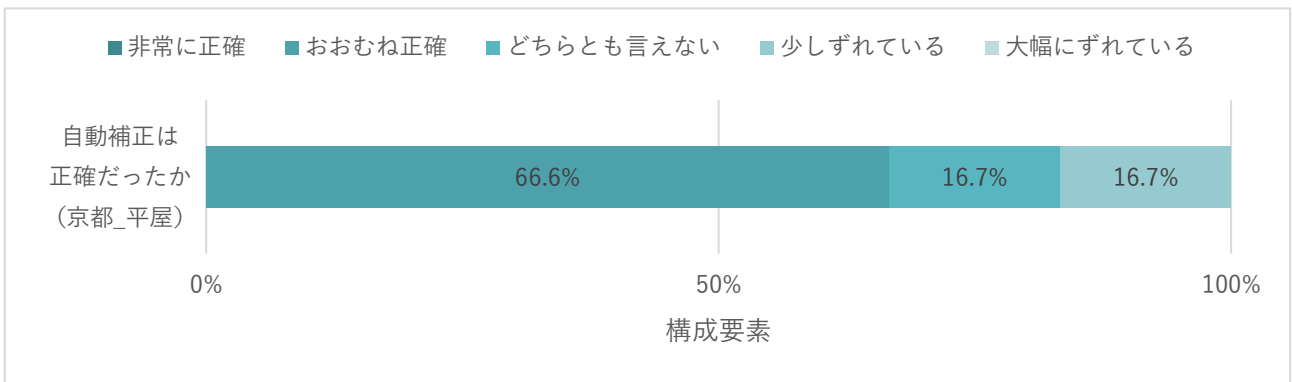


図 9-10 平屋の自動補正の正確性についてのアンケート結果_京都(n=6)

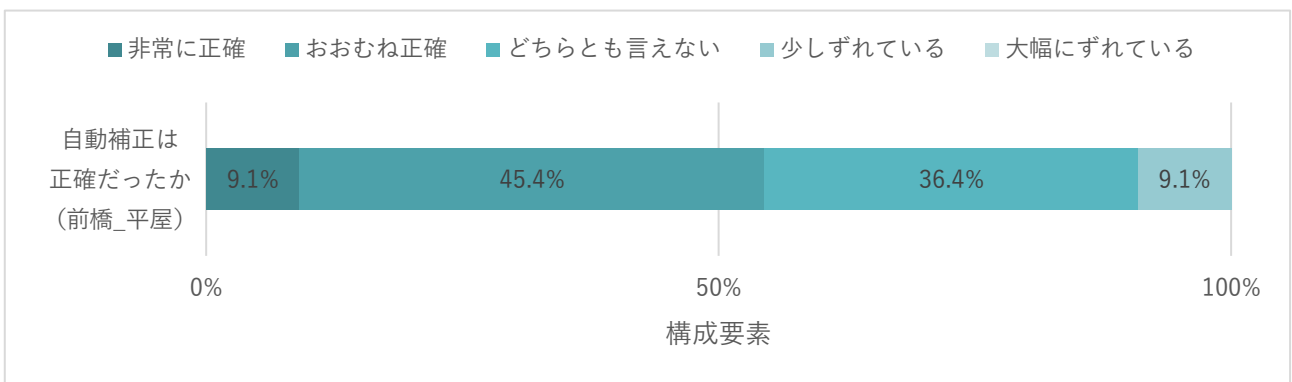


図 9-11 平屋の自動補正の正確性についてのアンケート結果_前橋 (n=11)

● 商業ビル

商業ビルにおける自動補正については、「おおむね正確」との評価が一定程度得られたものの、建物外観の複雑さに起因すると考えられる評価のばらつきも確認された。



図 9-12 商業ビル（京都）の自動補正のキャプチャ

図 9-13 商業ビル（前橋）の自動補正のキャプチャ

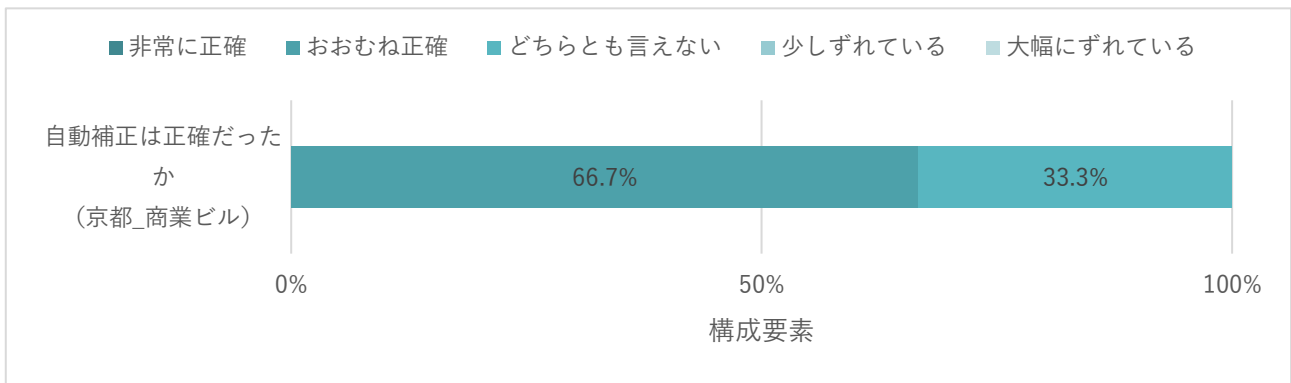


図 9-14 商業ビルの自動補正の正確性についてのアンケート結果_京都 (n=6)

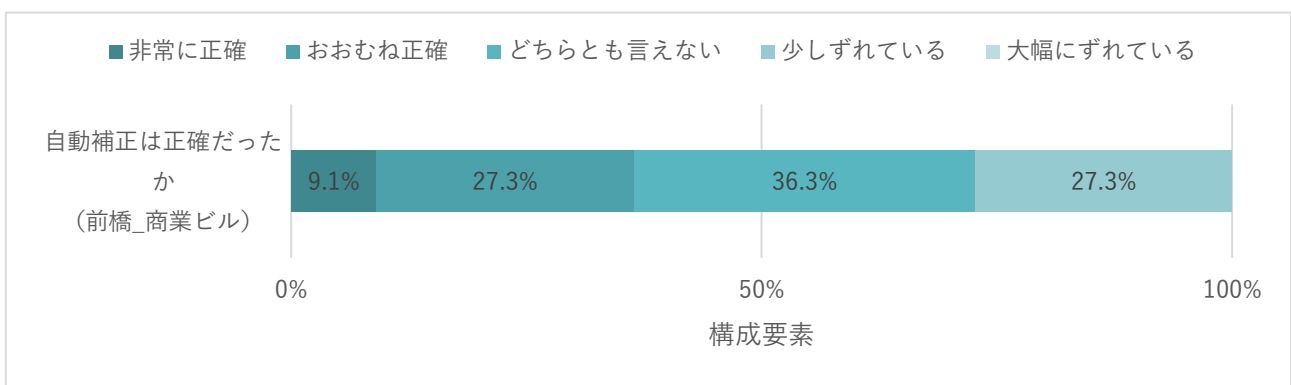


図 9-15 商業ビルの自動補正の正確性についてのアンケート結果_前橋 (n=11)

● 大型商業施設

大型商業施設については対象が 1 棟に限られるものの、「おおむね正確」との評価が大半を占めており、大規模で平坦な外壁面を有する建築物においては自動補正の有効性が確認された。なお、当該建築物については、京都の大型商業施設に対する自動補正結果を前橋の被験者にも提示し、評価を行っている。



図 9-16 大型商業施設（京都）の自動補正のキャプチャ

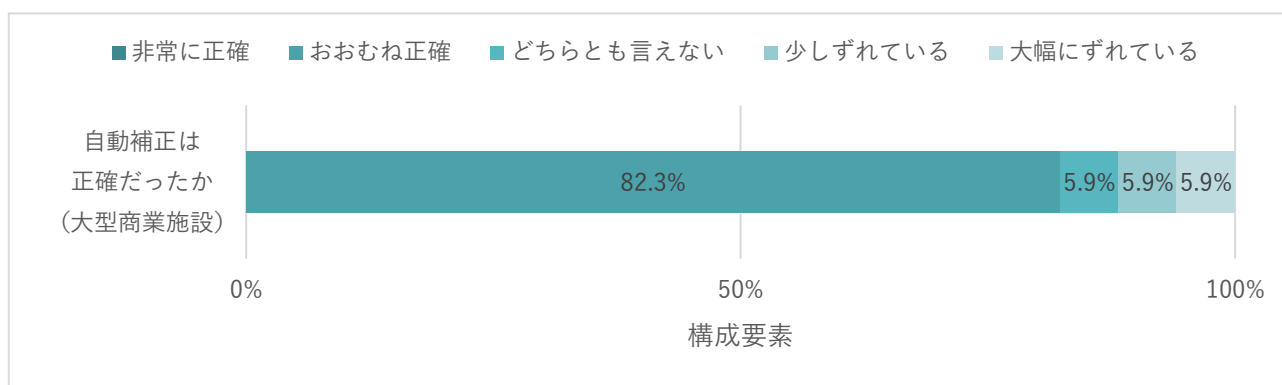


図 9-17 大型商業施設の自動補正の正確性についてのアンケート結果_京都・前橋合計 (n=17)

● タワー

タワー形状の建築物を対象とした自動補正では、「どちらとも言えない」及び「ずれている」との評価が大半を占めており、建物規模が大きく円すい形に近い複雑な外形を持つ建築物に対しては、本システムによる自動補正の適用が困難であることが確認された。なお、当該建築物については、京都のタワーに対する自動補正結果を前橋の被験者にも提示し、評価を行っている。

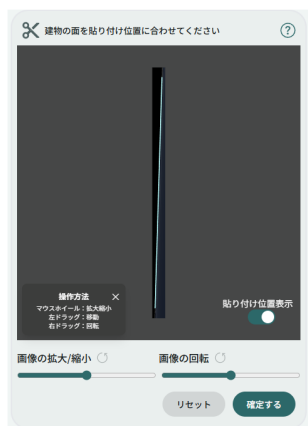


図 9-18 タワー（京都）の自動補正のキャプチャ

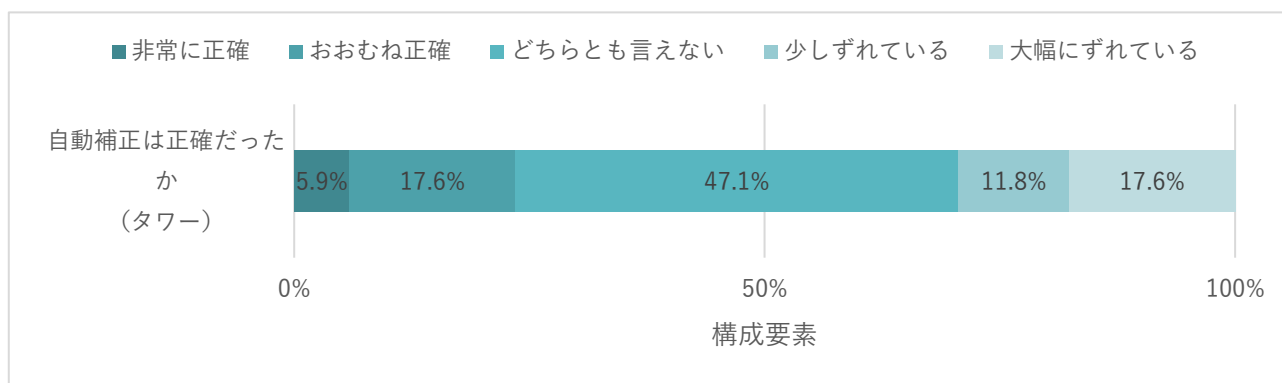


図 9-19 タワーの自動補正の正確性についてのアンケート結果_京都・前橋合計 (n=17)

● 寺社

寺社を対象とした自動補正では、「おおむね正確」との評価が一定程度得られたものの、屋根勾配が大きい構造や構成面が多い建物形状の影響により、「どちらとも言えない」「ずれている」との評価も見られ、正射変換を前提とした自動補正が難しいケースが確認された。



図 9-20 寺社（京都）の自動補正のキャプチャ



図 9-21 寺社（前橋）の自動補正のキャプチャ

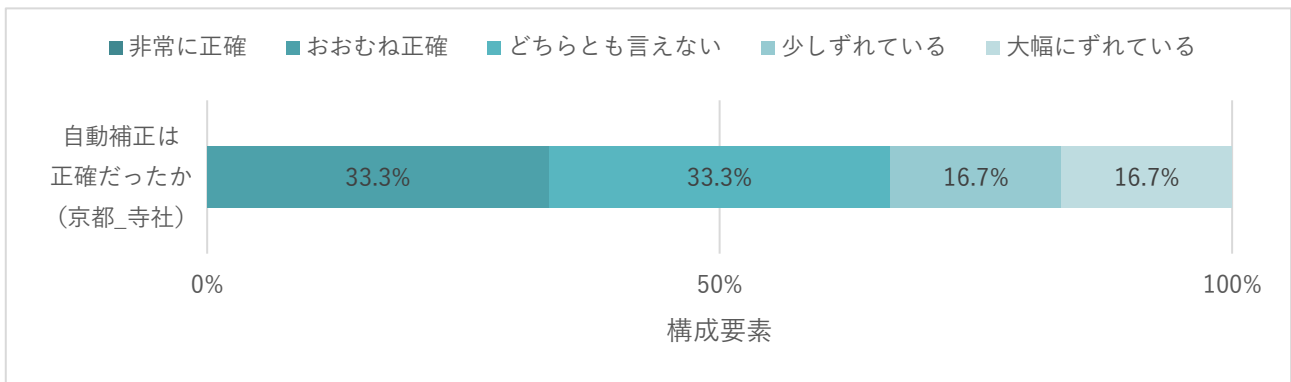


図 9-22 寺社の自動補正の正確性についてのアンケート結果_京都 (n=6)

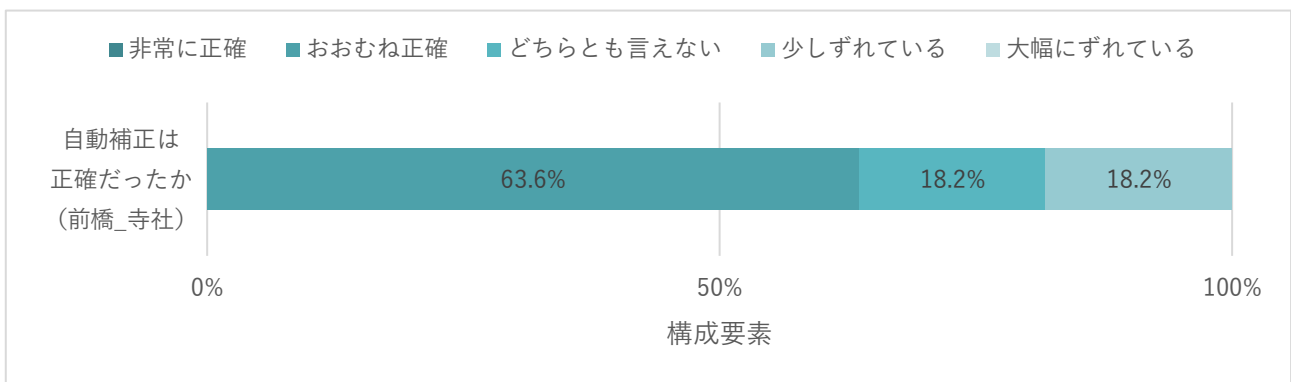


図 9-23 寺社の自動補正の正確性についてのアンケート結果_前橋 (n=11)

● 教会

教会建築を対象とした自動補正では、「おおむね正確」と評価された割合が高く、尖塔や装飾を含む外観であっても、壁面構成が比較的明確な建物においては自動補正が有効に機能することが示された。



図 9-24 教会（京都）の自動補正のキャプチャ



図 9-25 教会（前橋）の自動補正のキャプチャ

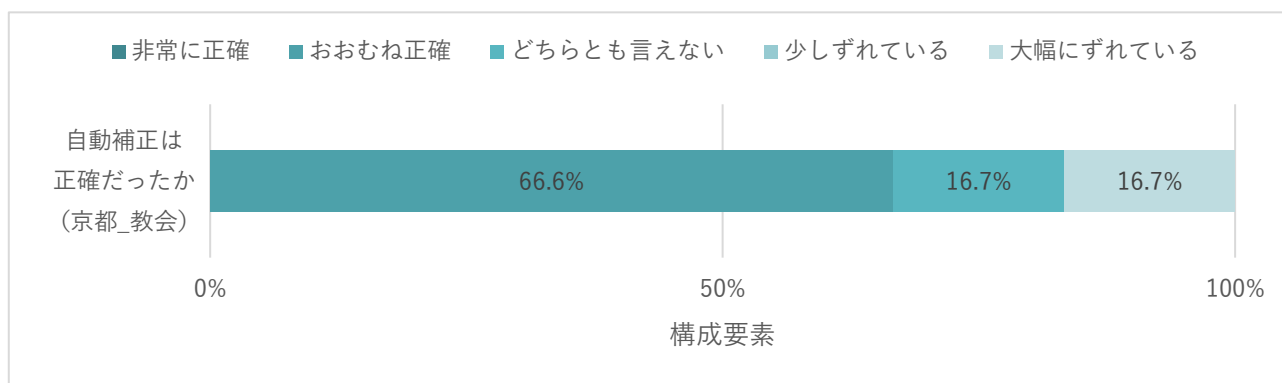


図 9-26 教会の自動補正の正確性についてのアンケート結果_京都 (n=6)

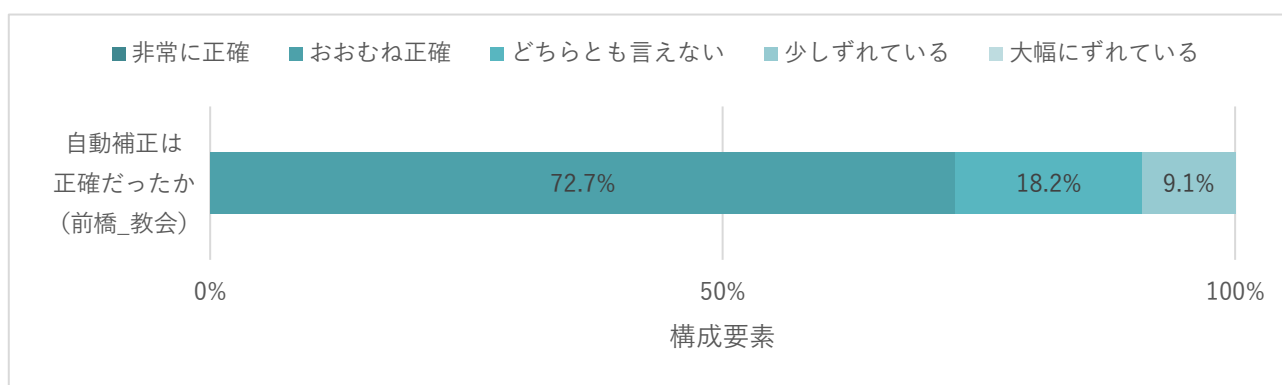


図 9-27 教会の自動補正の正確性についてのアンケート結果_前橋 (n=11)

● 駅

駅施設を対象とした自動補正では、構成面が単純で長方形に近い京都駅においては「おおむね正確」との評価が高かった一方、前橋駅では軒が深い構造により軒部分が十分に表現されず、補正自体は正確であるものの、表現範囲の制約が評価に影響した結果となった。



図 9-28 駅（京都）の自動補正のキャプチャ



図 9-29 駅（京都）の自動補正のキャプチャ

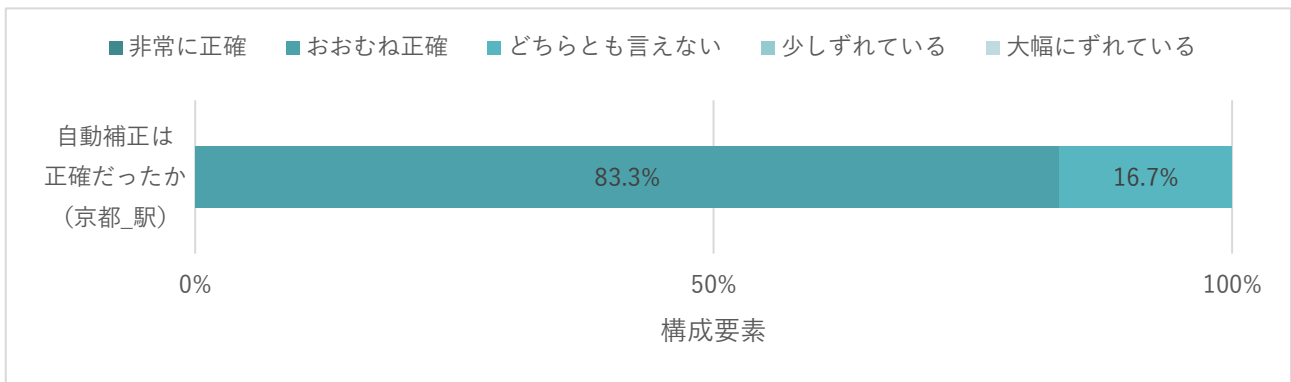


図 9-30 駅の自動補正の正確性についてのアンケート結果_京都 (n=6)

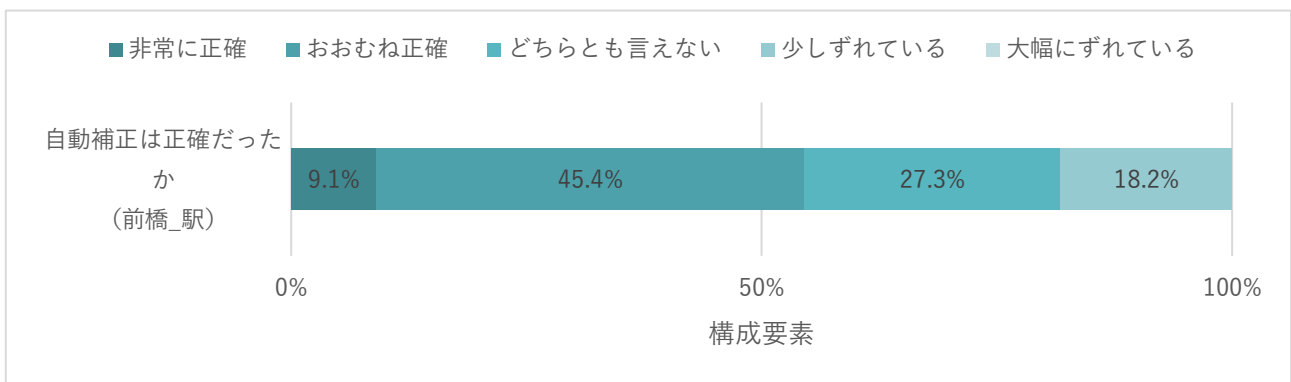


図 9-31 駅の自動補正の正確性についてのアンケート結果_前橋 (n=11)

2. テクスチャ貼付け位置は正確だったか

● 平屋

平屋におけるテクスチャ貼付け位置については、「おおむね正確」及び「どちらとも言えない」との評価が多くを占めたが、京都の事例では正面に屋根形状が入り込む構成となっており、屋根部分の表現が評価に影響したことで貼付け位置に関する評価が分かれる結果となった。



図 9-32 テクスチャ貼付け後の平屋（京都）



図 9-33 テクスチャ貼付け後の平屋（前橋）

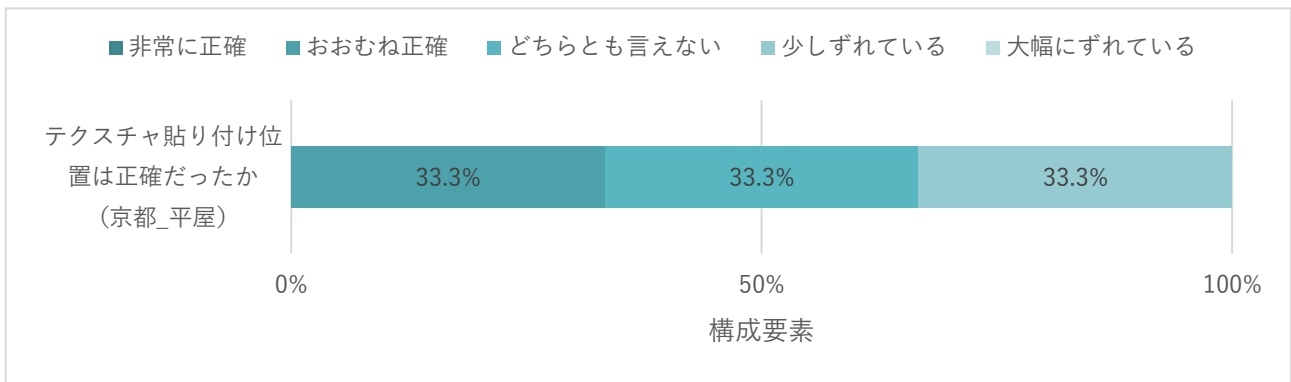


図 9-34 平屋のテクスチャ貼付け位置の正確性についてのアンケート結果_京都 (n=6)

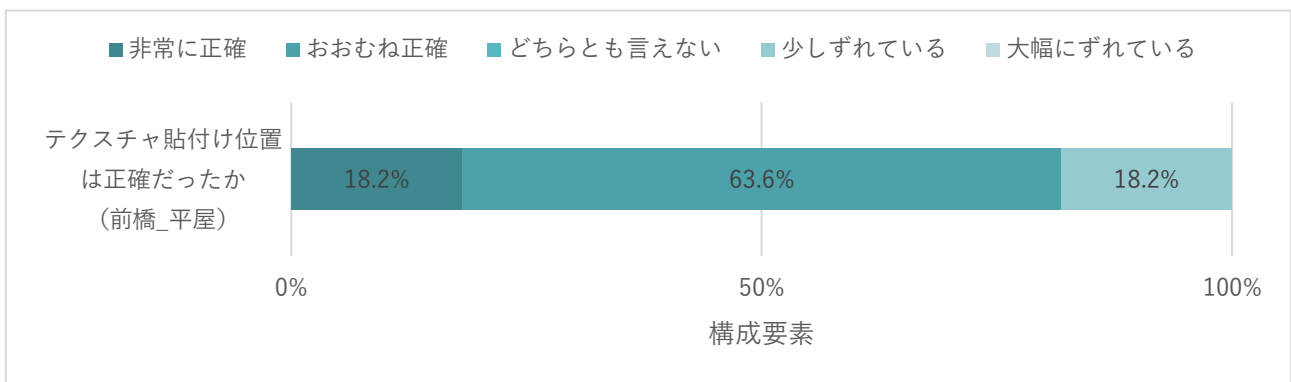


図 9-35 平屋のテクスチャ貼付け位置の正確性についてのアンケート結果_前橋 (n=11)

● 商業ビル

商業ビルにおけるテクスチャ貼付け位置については、京都では「おおむね正確」との評価が高かった一方、前橋では流線形で横に長い建物形状により一部の面が撮影対象とならず、その結果、貼付け位置の妥当性に関する評価が分かれる傾向が確認された

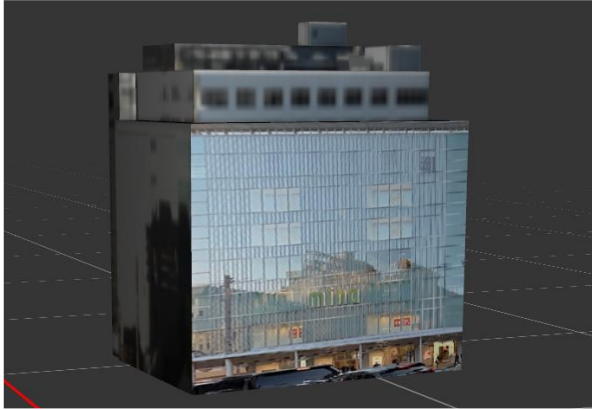


図 9-36 テクスチャ貼付け後の商業ビル（京都）

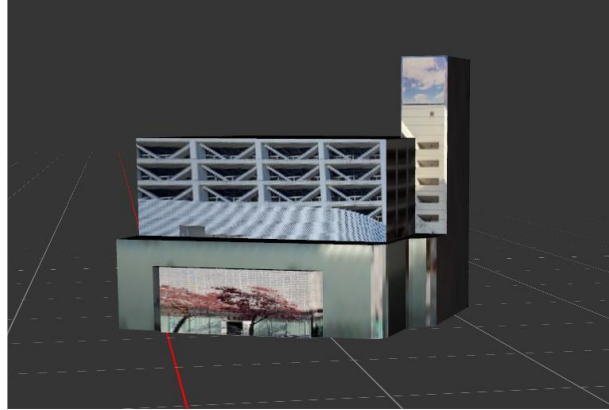


図 9-37 テクスチャ貼付け後の商業ビル（前橋）

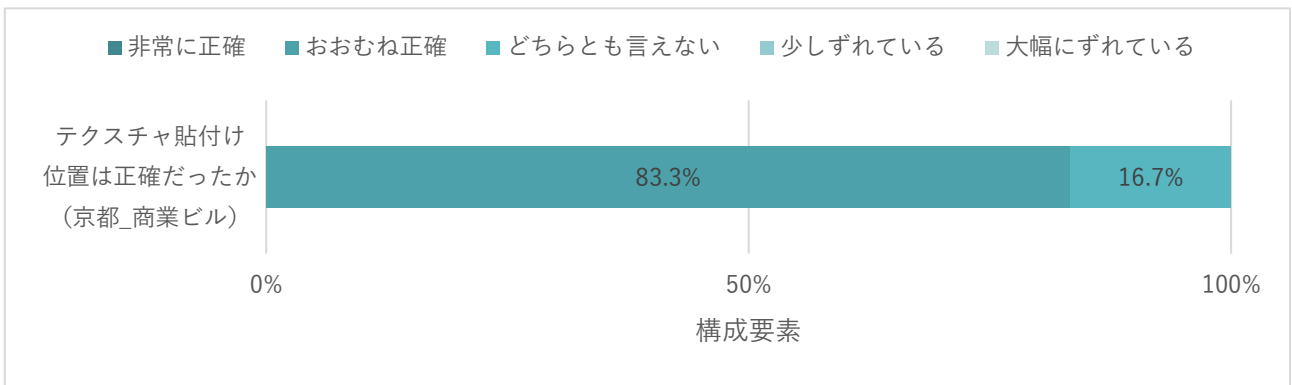


図 9-38 商業ビルのテクスチャ貼付け位置の正確性についてのアンケート結果_京都 (n=6)

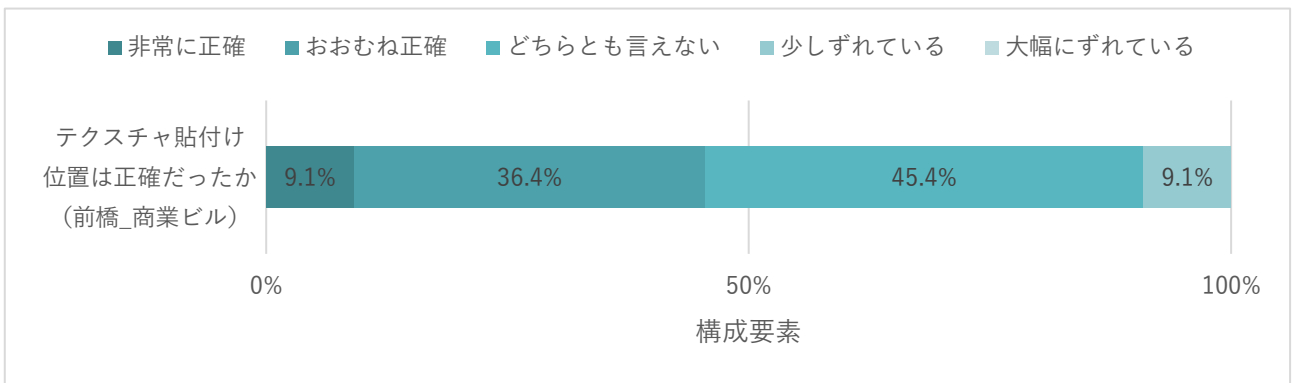


図 9-39 商業ビルのテクスチャ貼付け位置の正確性についてのアンケート結果_前橋 (n=11)

● 大型商業施設

大型商業施設におけるテクスチャ貼付け位置については評価が分かれ、「どちらとも言えない」との回答が最も多くなったが、これは前面の街路樹や構造物等の障害物が映り込むことで、貼付け位置の妥当性を判断しづらいケースが生じたことが評価に影響したと考えられる。なお、当該建築物については、京都の大型商業施設に対するテクスチャ貼付け結果を前橋の被験者にも提示し、評価を行っている

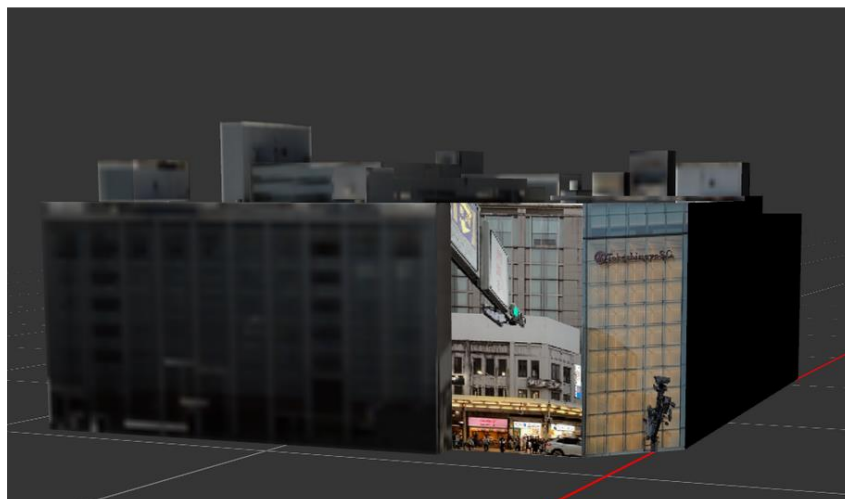


図 9-40 テクスチャ貼付け後の大型商業施設（京都）

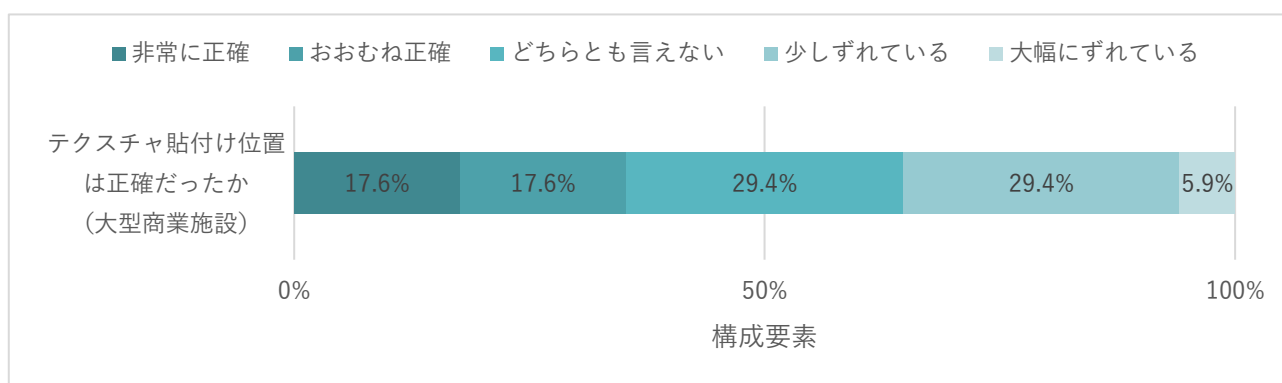


図 9-41 大型商業施設のテクスチャ貼付け位置の正確性についてのアンケート結果_京都・前橋 (n=17)

● 寺社

寺社では、前橋の事例において複数の面に同様の画像が貼り付けられる状況が見られ、撮影条件や編集による調整では十分に解消できず、貼り付け位置の正確性に関する評価が低下する要因となった。

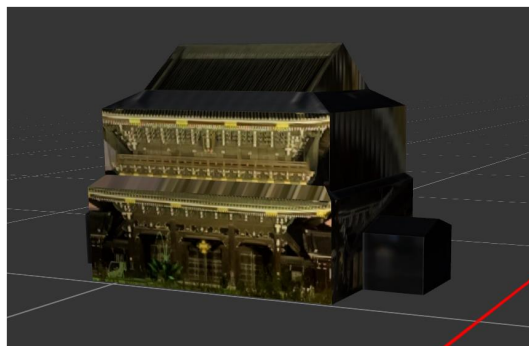


図 9-42 テクスチャ貼付け後の寺社（京都）

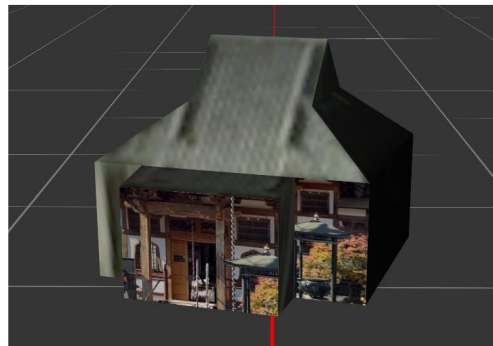


図 9-43 テクスチャ貼付け後の寺社（前橋）

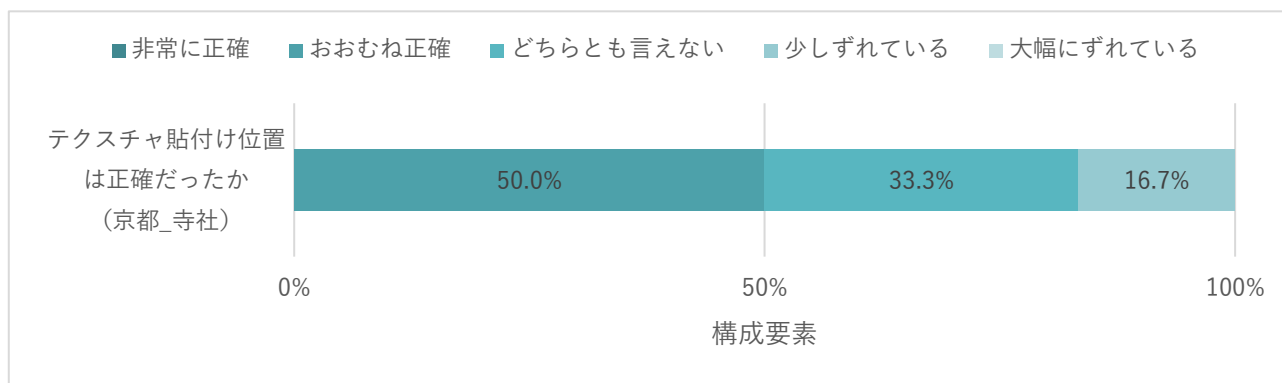


図 9-44 寺社のテクスチャ貼付け位置の正確性についてのアンケート結果_京都 (n=6)

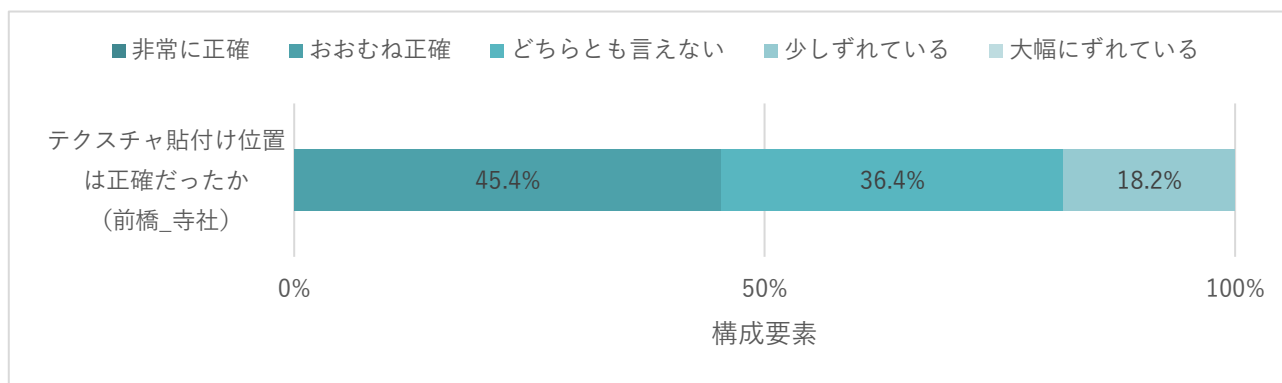


図 9-45 寺社のテクスチャ貼付け位置の正確性についてのアンケート結果_前橋 (n=11)

● 教会

教会におけるテクスチャ貼付け位置については、京都・前橋の両事例において「おおむね正確」との評価が多く、尖塔や装飾を含む外観であるものの、正面壁面の構成が明確な建物では貼付け位置の妥当性が比較的安定して確保されていることが確認された。

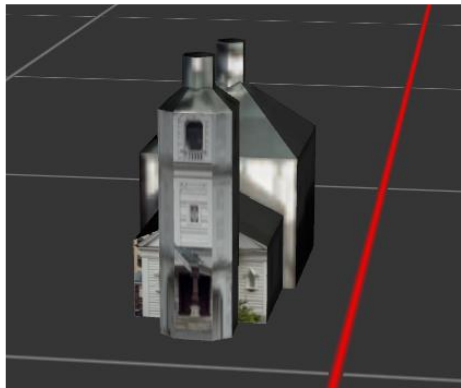


図 9-46 テクスチャ貼付け後の教会（京都）

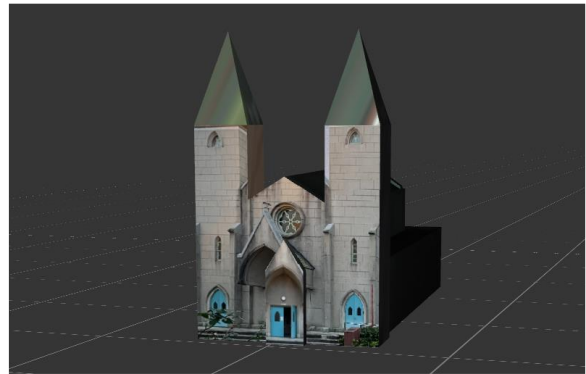


図 9-47 テクスチャ貼付け後の教会（前橋）

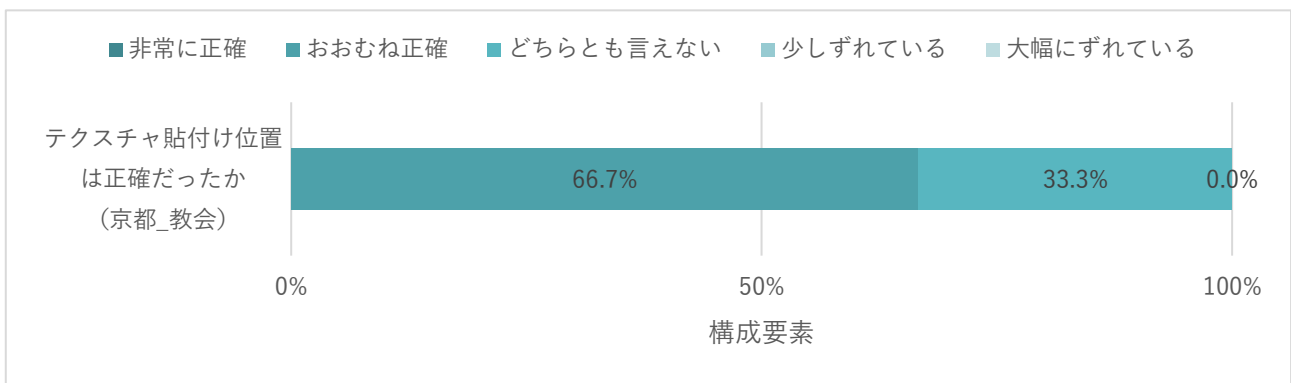


図 9-48 教会のテクスチャ貼付け位置の正確性についてのアンケート結果_京都 (n=6)

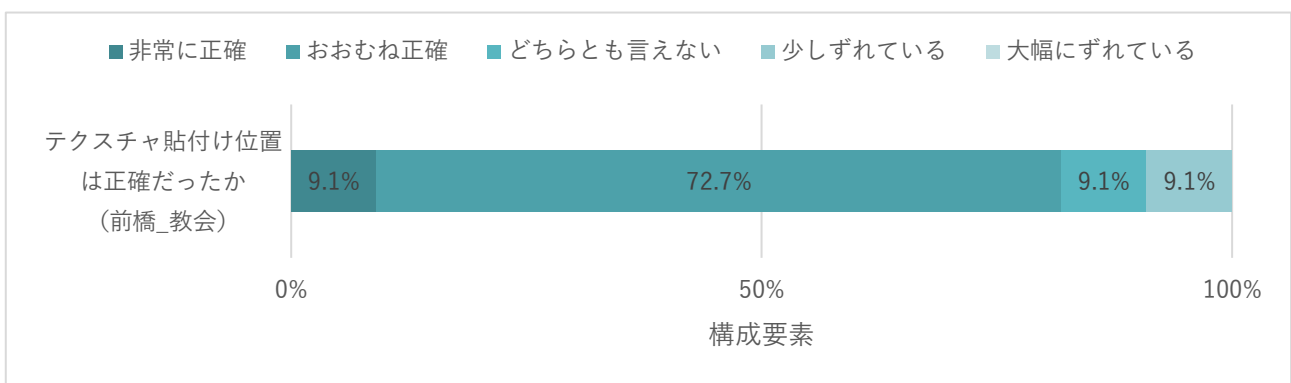


図 9-49 教会のテクスチャ貼付け位置の正確性についてのアンケート結果_前橋 (n=11)

● 駅

駅施設におけるテクスチャ貼付け位置については、京都駅では「おおむね正確」との評価が多く得られた一方、前橋駅では深い軒構造が十分に表現されておらず、貼付け位置自体は大きくずれていないものの、軒部分の再現性が評価に影響する結果となった。



図 9-50 テクスチャ貼付け後の駅（京都）

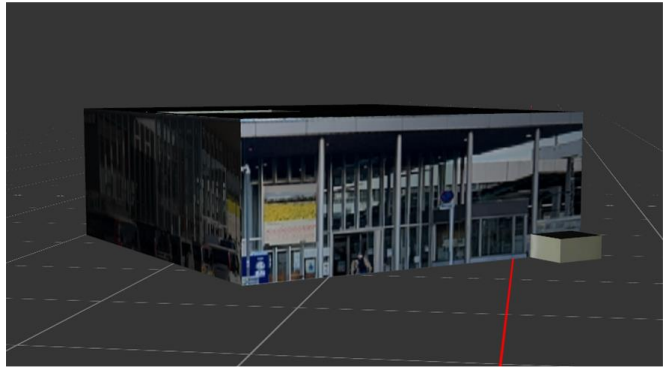


図 9-51 テクスチャ貼付け後の平屋（前橋）

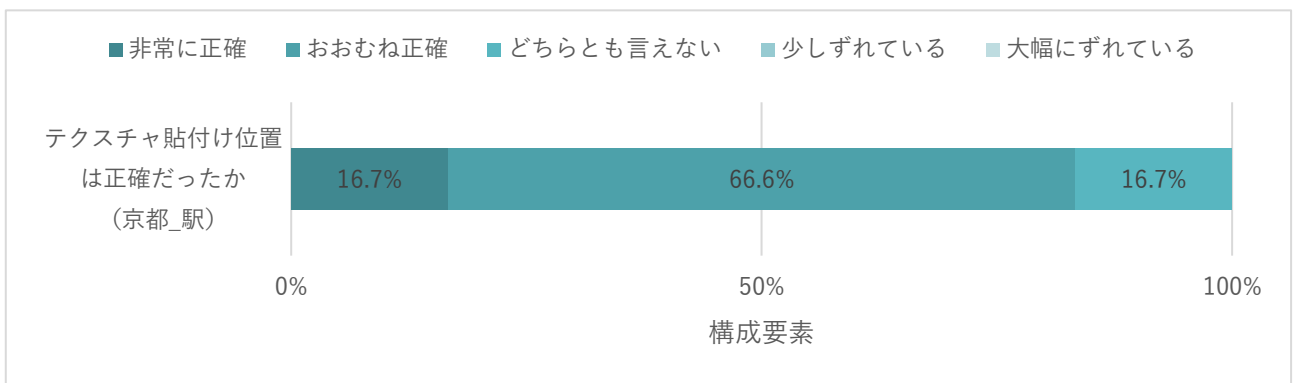


図 9-52 駅のテクスチャ貼付け位置の正確性についてのアンケート結果_京都 (n=6)

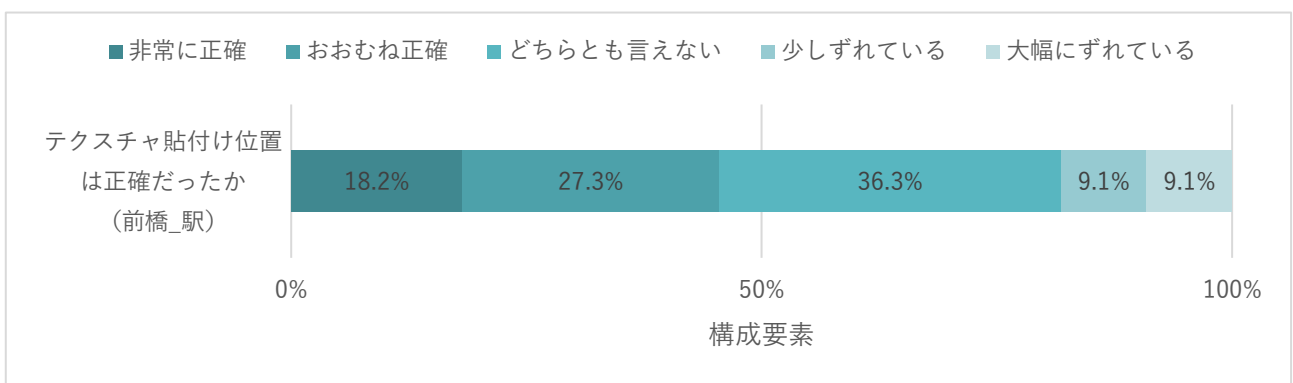


図 9-53 駅のテクスチャ貼付け位置の正確性についてのアンケート結果_前橋 (n=11)

表 9-5 精度に関連する定性コメント

No	検証項目	関連する定性コメント
1	テクスチャ貼付け位置の自動補正は正確か	<ul style="list-style-type: none"> ● チャンピオンデータとしては良くできているので、将来期待できる技術だと感じた ● モデルの形状が単純な場合はおおむね正確に配置されていると感じたが、LOD3 建築物モデルやタワーなど形状が複雑な建物では、面が細くなるため正確に貼り付けることが難しいと感じた ● 現地で撮影した画像をそのまま使用するよりも、加工・編集した画像を取り込むことで、テクスチャとしてより高品質に利用できると感じた ● 被写体による差は大きく感じられなかったが、複数枚撮影した画像のうち、使用可能な画像は限られる印象があった。そのため、一つの面に対して複数枚の撮影を行い、選択できるようにすることで精度が向上すると感じた
2	出力した CityGML データのテクスチャ貼り付け位置は正確か	<ul style="list-style-type: none"> ● 撮影面に対して補正処理は非常に良いと感じた ● 平面感が気になった ● 一部の建物では、形状と写真がずれている箇所が見られた ● 寺社や無壁舎の建物では、LOD2 建築物モデルの仕様上、写真どおりの形状が表現されにくく、その結果として LOD3 建築物モデルの方が適していると感じた建物もあった ● 画像の影や照り返しが気になる場面があった

2) ユーザビリティ

本システムのユーザビリティについては、操作方法の分かりやすさや画面表示に関して一定の評価が得られており、全体としては実務利用を想定した基本的な操作性が確保されていることが確認された。一方で、位置合わせやサイズ調整、建物・面の選択といった操作においては分かりにくさを感じるという回答も一定数見られ、利用者の理解度や操作経験によって評価が分かれる傾向が確認された。

また、データ出力までの作業時間については、多くの回答者が実用的な水準であると評価しているものの、3次メッシュダウンロードなど一部の処理においては待ち時間が長いと感じる回答も見られた。これらの結果から、本システムは基本的なユーザビリティを備えている一方、操作の直感性や処理状況の可視化といった点については、今後の改善余地が残されていることが示された。

● 操作方法是分かりやすいか

操作方法の分かりやすさに関して「分かりやすい」「とても分かりやすい」と回答した割合が 52.9%を占めており、基本的な操作フローは一定程度直感的に理解できるものと評価された。一方で、「やや分かりづらい」「分かりづらい」と感じた回答も 35.3%見られ、特に位置合わせやサイズ調整の操作において難しさを感じるケースがあることが定性コメントから確認された。

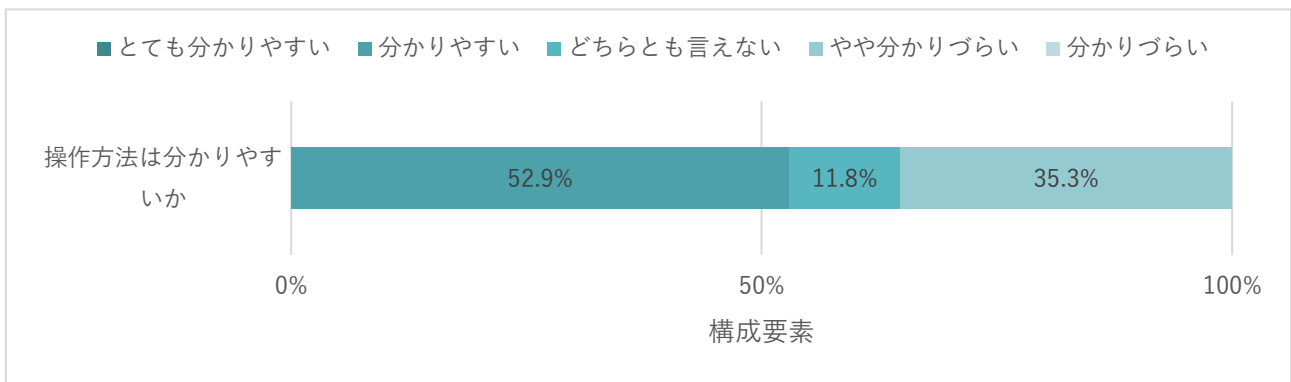


図 9-54 操作方法の分かりやすさについてのアンケート結果 (n=17)

● 画面表示は分かりやすいか

画面表示については、「分かりやすい」の評価が 41.2%であった一方、「普通」との回答が 29.4%、「やや分かりづらい」「分かりづらい」との回答が約 30%を占めており、面選択時に対象が分かりにくい点や、複数の面から選択する際の視認性に課題があることが示唆された。

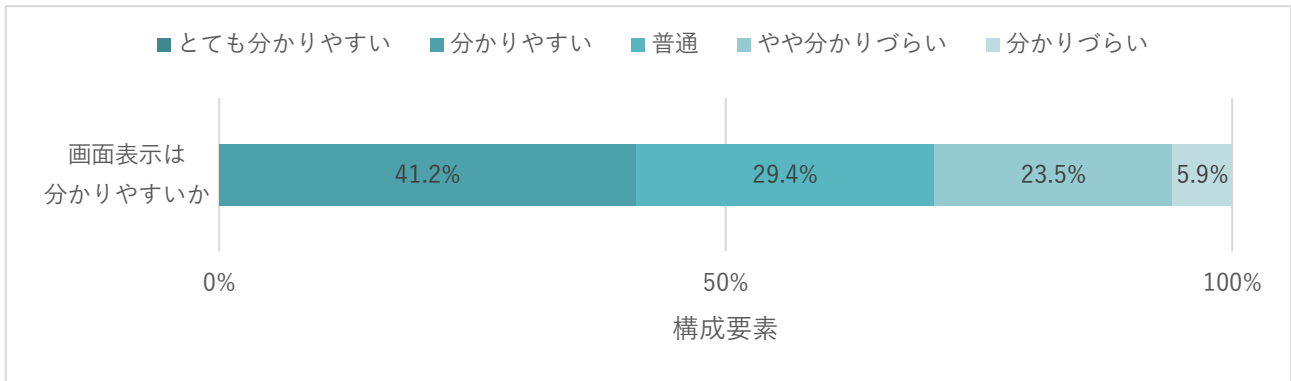


図 9-55 画面表示の分かりやすさについてのアンケート結果 (n=17)

● データ出力までの作業時間は実用的な水準か

また、データ出力までの作業時間については、「全くストレスを感じなかった」「特に問題はなかった」と回答した割合が建物データダウンロードでは 88.2%に達し、全体としては実用的な水準と評価された。一方、3次メッシュダウンロードの処理については、「やや不満を感じた」との回答が 27.3%見られ、処理時間の短縮や進捗状況を可視化する UI 改善が今後の課題として整理された。

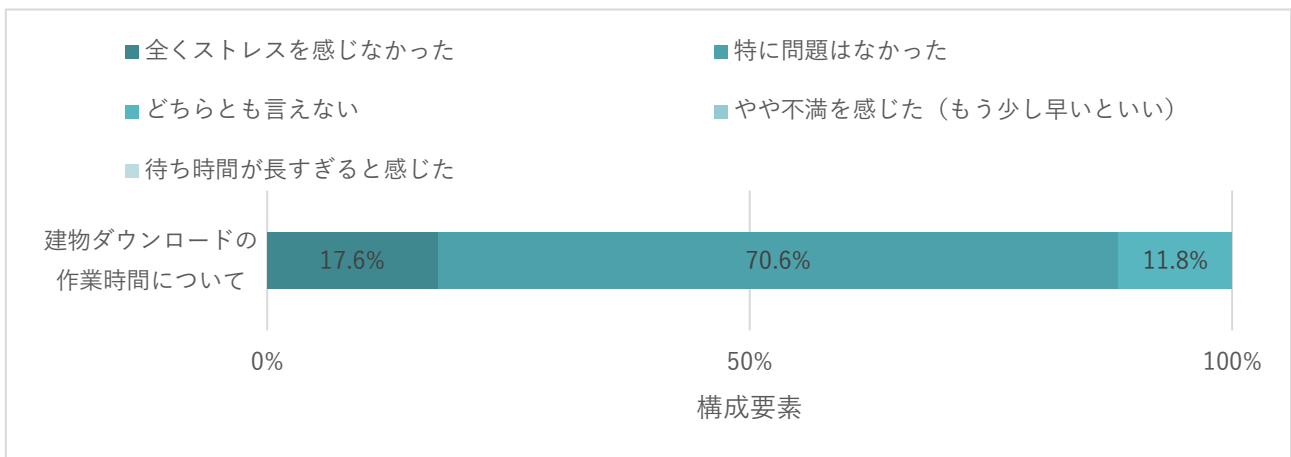


図 9-56 建物ダウンロードの作業時間についてのアンケート結果 (n=17)

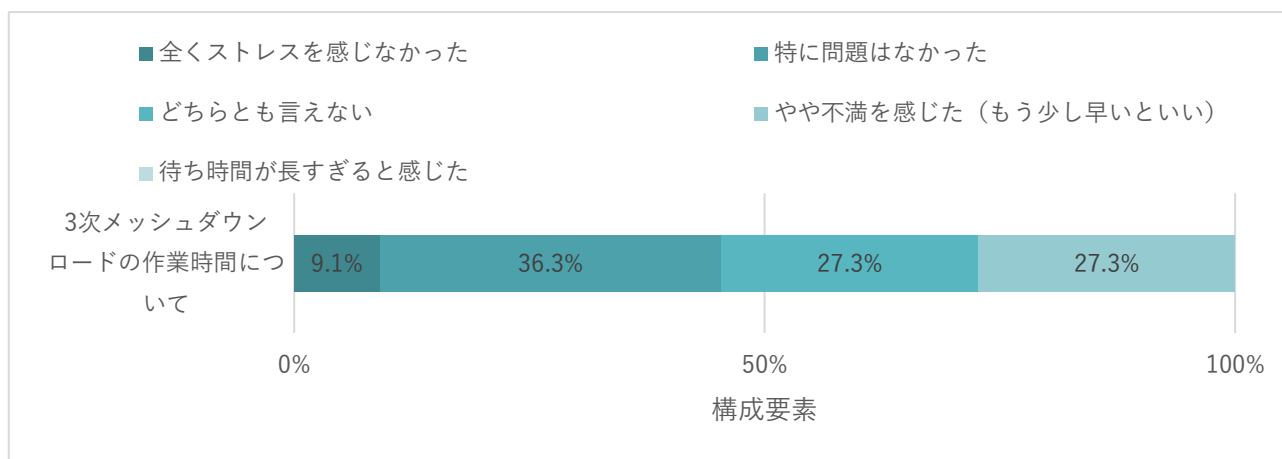


図 9-57 3次メッシュダウンロードの作業時間についてのアンケート結果 (n=17)

表 9-6 ユーザビリティに関連する定性コメント

No	検証項目	関連する定性コメント
1	操作方法は分かりやすいか	<ul style="list-style-type: none"> ● 位置合わせやサイズ合わせの操作が難しく、思ったとおりに調整するのが大変だった ● 細かい調整を行う際に、どの操作がどの結果に影響しているのか分かりにくかった
2	画面表示は分かりやすいか	<ul style="list-style-type: none"> ● 面選択や建物選択の際に撮影画像だけが表示されると、現在どの建物・どの面を操作しているのか分からなくなった ● 「面を選んでください」と表示される場面で複数の候補が表示されるが、どこを指しているのか分かりにくかった ● LOD2 建築物モデルなどを表示して、どの部分を修正しているのかが分かると、より直感的に操作できると感じた
3	データ出力までの作業時間は実用的な水準か	<ul style="list-style-type: none"> ● 建物データのダウンロードは問題なかったが、3次メッシュのダウンロードは少し遅く感じた

3) ユーザー検証・活用シナリオ評価

本システムのユーザー検証・活用シナリオ評価では、3D 都市モデルへのテクスチャ付与作業の効率化や、高画質な外観表現を活用したシミュレーション等への適用可能性について検証を行った。その結果、作業効率の向上や説明用途・検討用途における有用性については一定の評価が得られており、実務への導入を検討する余地があることが確認された。

一方で、活用の具体性や期待する表現水準によって評価が分かれる項目も見られ、特に LOD3 建築物モデルとの比較においては、LOD2 建築物モデル+SNAP によるテクスチャ付与モデルをどの用途で用いるかによって有用性の判断が異なる傾向が確認された。これらの結果から、本システムは全ての用途に一律に適用するものではなく、利用目的や前提条件を整理した上で活用することが重要であることが示された。

● 3D 都市モデルへのテクスチャ付与を効率化できているか

3D 都市モデルへのテクスチャ付与作業の効率化については、「やや上がる」と回答した割合が 60%を占めており、従来手法と比較して作業効率の向上が期待できるとの評価が得られた。一方で、「変わらない」「やや下がる」との回答も 40%見られた。定性コメントからは、面選択時に UUID での判別が分かりにくい点や選択中の面がモデル上で直感的に把握しづらい点が、効率向上を実感しにくい要因として挙げられた。これらの結果から、本システムは効率化のポテンシャルを有するものの、精度及び操作性の改善が実務利用に向けた前提条件となることが示唆された。

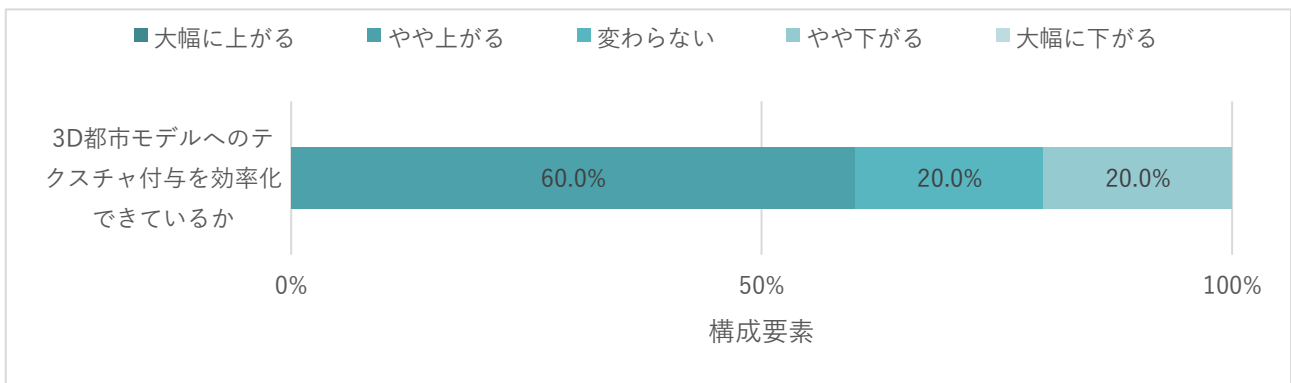


図 9-58 3D 都市モデルへのテクスチャ付与を効率化できているかについてのアンケート結果 (n=5、航測会社 5 名が回答)

● 高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるか

高画質な画像を付与した建築物モデルのシミュレーション等への活用可能性については、「大いに活用できる」「ある程度活用できる」との回答が 58.9%を占め、一定の活用可能性が確認された。定性コメントでは、景観シミュレーション、中心市街地の再整備検討、行政による土地利用調整、住民説明会や住民参加型プロジェクトなど、視覚的説得力が求められる場面での活用が期待されていることが示された。一方で、「分からない」との回答も 41.2%見られ、LOD3 モデルまでの整備が難しいケースにおいて、見た目を重視した用途に限定して活用することが現実的であるとの意見も確認された。

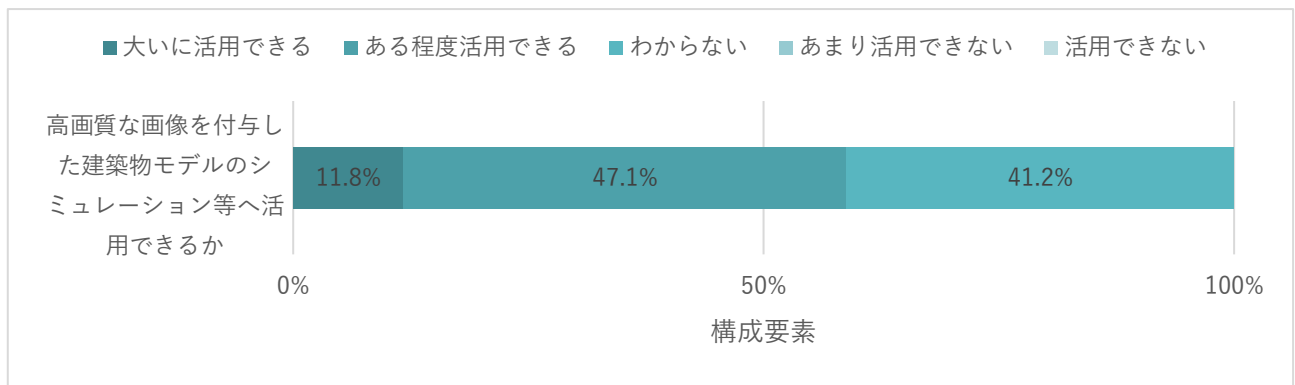


図 9-59 高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるかについてのアンケート結果(n=17)

● LOD3 建築物モデルと比較して、LOD2 建築物+SNAP によるテクスチャ付与モデルは有用か
 LOD3 建築物モデルと比較した場合の LOD2 建築物+SNAP によるテクスチャ付与モデルの有用性については、「非常に有用である」「有用である」と評価した回答が合計で 47.0%となり、一定の肯定的評価が確認された。一方で、「どちらとも言えない」との回答が 23.6%、「有用とは言えない」との回答が 29.4%を占めており、評価が分かれる結果となった。肯定的な評価では、LOD3 建築物モデル以上の品質と感じられるケースがあることや、構成部材ごとに丁寧に画像を貼り付けることが可能であれば有用であるとの意見が挙げられた。以上より、LOD2 建築物+SNAP によるテクスチャ付与モデルは、LOD3 建築物モデルと比較した場合において一定の有用性が認識される一方、用途や期待する表現水準によって評価が分かれることが確認された。



図 9-60 従来の CityGML (LOD3 建築物モデル)



図 9-61 LOD2 建築物モデルに本システムで生成したテクスチャを付与した CityGML

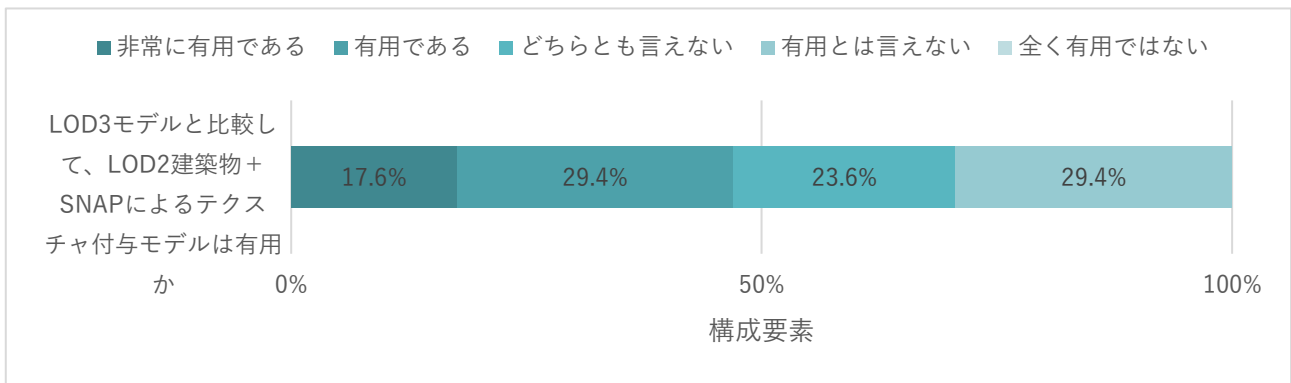


図 9-62 LOD3 建築物モデルと比較して、LOD2 建築物+本システムで生成したテクスチャ付与モデルの有用かについてのアンケート結果 (n=17)

表 9-7 ユーザー検証・活用シナリオ評価に関連する定性コメント

No	検証項目	関連する定性コメント
1	3D 都市モデルへのテクスチャ付与を効率化できているか	<p>【やや上がる】</p> <ul style="list-style-type: none"> ● 撮影後に、その場で写真を確認できる機能があると安心して作業できると感じた ● 既存の写真を机上操作で取り込める機能があると便利だと感じた ● 生成 AI と連携し、電柱などの遮蔽物を非表示にする機能や、影の影響を軽減できる機能があると良いと感じた ● Wi-Fi 接続や GNSS 受信が不安定な環境を考慮し、撮影画像を一時的にスタックできる機能が必要だと感じた ● 広角撮影には限界があるため、パノラマ撮影機能やモザイク撮影機能があるとよいと感じた ● 被写体の面について、正対（法線方向）から 60 度までを撮影可能角度とするデフォルト設定があるが、この角度を調整できるようにしたいと感じた ● 将来的には、ウェアラブルデバイスへの対応も検討できるとおもしろいと感じた ● 効率性については、従来のスナップ撮影と比較して効果は十分に期待できる一方で、精度及び可用性の改善が不可欠だと感じた <p>【やや下がる】</p> <ul style="list-style-type: none"> ● モデルのワイヤーフレームが画面に表示され、「この面を修正します」と分かるようになると使いやすいと感じた
2	高画質な画像を付与した建築物モデルのシミュレーション等へ活用できるか	<p>【大いに活用できる】</p> <ul style="list-style-type: none"> ● 再整備された街なかを広く活用してもらうためのシミュレーションに使いたい ● 都市景観のシミュレーションや再整備の検討に使いそうだと感じた <p>【ある程度活用できる】</p> <ul style="list-style-type: none"> ● 立体的な建物が表現されているため、今後の街づくりやイベントの説明に活用できそうだと感じた ● 中心市街地の再開発を含む再整備の検討に使いそうと思った ● 地元説明会での説明資料として利用できると感じた ● 行政で隣接する土地や建物との調和を検討する際に活用できそうと思った ● LOD3 モデルまで整備できない場合でも、見た目を良くしたい用途や、見た目を重視したユースケースでは活用できそうだと感じた <p>【分からない】</p> <ul style="list-style-type: none"> ● 住民体験型のプロジェクトに向いていると感じた ● 具体的な業務での活用場面は思い当たらなかった

3	LOD3 建築物モデルと比較して、LOD2 建築物+SNAPによるテクスチャ付与モデルは有用か	【有用】 ● LOD3 以上の品質だと思う 【どちらとも言えない】 ● LOD3 は構成部材ごとに丁寧に画像を貼り付けることが可能なら有用
---	---	--

10. 成果と課題

10-1. 本実証で得られた成果

10-1-1. 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 10-1 3D 都市モデルの技術面での優位性

大項目	小項目	3D 都市モデルの技術面での優位性
システム (機能)	正射変換機能	● 建物形状が比較的単純な場合には、建物面を斜めから撮影した画像を正射画像へと高精度に変換することができた
	屋根面画像生成機能	● PLATEAU-Ortho からオルソ画像を取得し、対象の屋根面を内包するようテクスチャ画像を生成することができた
	貼付け位置補正機能	● 建物形状が比較的単純な場合には、テクスチャ貼付け位置の自動補正が安定して機能し、手動補正を行わずにテクスチャ付与が可能であることを確認できた
	データ出力機能	● 出力データが CityGML としての仕様及び品質要件を満たしていることを確認できた
	テクスチャ品質	● LOD2 建築物モデルに対して撮影画像から生成したテクスチャを付与することで、建物外観の視認性や品質を向上させることができた

10-1-2. 3D 都市モデルのビジネス面での優位性

実証実験を通じて、以下のような 3D 都市モデルのビジネス面での優位性が示された。

表 10-2 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
システム (UI/UX)	データ整備コストの削減	● 専門的な撮影機材や 3D 編集ツールを用いないことで、3D 都市モデルのテクスチャ付与作業にかかる機材費や人件費の削減が可能であることを確認できた

10-1-3. 3D 都市モデルの公共政策面での優位性

表 10-3 3D 都市モデルの公共政策面での優位性

大項目	小項目	3D 都市モデルの公共政策面での優位性
システム (UI/UX)	テクスチャ付与作業の簡略化	<ul style="list-style-type: none"> ● 専門的な 3D 編集ツールを用いずに、テクスチャ付与作業を実施できる操作環境を提供できた
	テクスチャ品質	<ul style="list-style-type: none"> ● LOD2 建築物モデルに対して撮影画像から生成したテクスチャを付与することで、建物外観の視認性や品質を向上させることができた
	実務利用に向けた有用性	<ul style="list-style-type: none"> ● ユーザー検証の結果、LOD2 建築物モデルへのテクスチャ付与について高い有用性評価が得られ、説明用途や景観検討等への活用可能性を確認できた ● LOD3 建築物モデルと比較した場合でも、用途を限定すれば、SNAP で生成したテクスチャを付与した LOD2 建築物モデルが有用であると評価されるケースがあることを確認できた

10-2. 実証実験で得られた課題と対応策

表 10-4 実証実験で得られた課題と対応策

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)	自動補正機能の 適用範囲	<ul style="list-style-type: none"> ● 本実証では、自動補正による貼付け位置にズレがない建物面の割合について、目標を 60% に設定した ● 実証の結果、上記の割合は 33.3% にとどまった。建物面の形状や規模によって補正精度にばらつきが生じるため、一律の適用が困難なケースがあることが確認された 	<ul style="list-style-type: none"> ● 撮影画像内の建物面の頂点を手動で選択するオプションを追加し、貼付け位置補正および正射変換の精度を向上させる
	LOD3 建築物モデルとの位置づけ	<ul style="list-style-type: none"> ● 本実証では、SNAP によってテクスチャを付与した LOD2 建築物モデルを、LOD3 建築物モデルの代替として活用可能か検証し、その位置付けを整理することを目的とした ● 実証の結果、SNAP によってテクスチャを付与した LOD2 建築物モデルの評価は用途によって分かれ、LOD3 建築物モデルの代替として一律に扱うことは難しいとの指摘があった 	<ul style="list-style-type: none"> ● LOD3 建築物モデルの代替ではなく、LOD3 建築物のモデル整備が困難な場合の補完的手法として位置付け、用途や目的に応じた使い分け指針を整理する
システム (UI/UX)	建物選択の分かりやすさ	<ul style="list-style-type: none"> ● 本実証における UI/UX の実装はあくまで技術実証の範囲に留め、将来的な実務利用において課題となる操作上の論点を抽出することを目的とした ● 実証の結果、建物の選択時に撮影画像と町丁目レベルの住所が表示される状態では、どの建物を対象として操作しているのか把握しづらいとの指摘があった 	<ul style="list-style-type: none"> ● 建物選択時に 2D 又は 3D の地図を表示し、対象の建物を視覚的に確認できる UI を実装する

面選択の分かりやすさ		<ul style="list-style-type: none"> ● 本実証では、面の選択時における判別性の課題を抽出することを目的とした ● 実証の結果、面の選択時に撮影画像と UUID が表示される状態では、どの面を対象として操作しているのか把握しづらいとの指摘があった 	<ul style="list-style-type: none"> ● 面の選択時に撮影画像に紐づいている面情報を撮影画像上にハイライトで表示し、対象の面を視覚的に確認できる UI を実装する
テクスチャ貼付け漏れの把握		<ul style="list-style-type: none"> ● 本実証では、現状の UI でテクスチャ貼付け漏れを防止できるかを検証し、十分にカバーできない場合の課題を整理することを目的とした ● 実証の結果、テクスチャ貼付け済み・未貼付けの面が分かりにくく、漏れが発生する可能性があるとの意見が確認された 	<ul style="list-style-type: none"> ● テクスチャ貼付け済み・未貼付けの面数を表示し、テクスチャ貼付け状況を可視化する機能を追加する
撮影環境への依存		<ul style="list-style-type: none"> ● 本実証では、オフライン環境での撮影及びアップロード機能の実装は対象外とし、通信環境に起因する課題の整理を目的とした ● 実証の結果、インターネット接続が不安定な環境では、スマートフォンでの AR 表示や画像アップロードが行えず不便との指摘があった 	<ul style="list-style-type: none"> ● 今後は、AR 表示用のデータや撮影画像をローカルに保持する等、通信が不安定な環境でも運用可能な仕組みを検討する
操作の柔軟性		<ul style="list-style-type: none"> ● 本実証では、超広角・望遠・パノラマ等の撮影モード拡張の実装までは対象とせず、撮影条件に応じた課題整理を目的とした ● 広角カメラ限定での実証の結果、建物の形状・サイズ・立地によっては撮影が難しいとの意見があった 	<ul style="list-style-type: none"> ● 超広角カメラや望遠カメラ、パノラマ撮影など、複数の撮影モードを提供し、撮影条件に応じた選択肢を増やす

	<p>対応端末の制約</p>	<ul style="list-style-type: none"> ● 本実証では、スマートフォンの対応機種 of 拡張は対象とせず、iPhone 13 Pro における課題整理を目的とした ● iPhone 13 Pro 以外の iPhone シリーズ、Android 端末、タブレットへの対応を求める声が見られた 	<ul style="list-style-type: none"> ● 対応 OS やデバイスの拡張を段階的に検討し、利用環境の幅を広げる
--	----------------	---	---

10-3. 今後の展望

本実証を通じて、LOD2 建築物モデルに対してスマートフォンで撮影した画像を用いたテクスチャ付与の有用性が確認された。一方で、建物形状や構成面の複雑さに起因して、正射変換及びテクスチャ貼付け位置の補正精度にばらつきが生じることや、UI/UX 面に改善の余地があることも明らかとなった。今後は、データ整備の実務利用を見据え、技術面及び運用面の双方から段階的な改善を進めることが重要である。

技術面においては、1 枚の撮影画像に複数の建物面を対応させ、同時に複数面へテクスチャを付与する仕組みを導入することで、実務に適した作業効率の向上を図る。また、ブラウザシステム上でテクスチャ内の個人情報や写り込みなどの障害物を除去可能とすることで出力データの利活用範囲を拡大し、オープンデータとしての活用促進に繋げる。

UI/UX の観点では、建物選択や面選択の分かりやすさの向上、撮影済み・未撮影箇所の可視化、通信環境に依存しない作業フローの整備など、利用者の操作負荷を低減するための改善が必要である。これにより、専門的な知識を持たない利用者であっても、日常的にテクスチャ更新作業を行える環境の実現が期待される。

さらに、出力した CityGML データの整合性や互換性が確認されたことから、今後は自治体や民間事業者による実運用を想定し、都市景観の把握、説明用途、検討資料作成など、具体的なユースケースに即した導入検証を進めることが有効であると考えられる。段階的な導入と改善を重ねることで、3D 都市モデルの持続的な更新と利活用を支える基盤技術としての発展が期待される。

11. 用語集

A) アルファベット順

表 11-1 用語集（アルファベット順）

No.	用語	説明
1	AR	現実の映像にデジタル情報を重ねて表示する技術。本システムでは、建物モデルや撮影対象面を実空間上に重ねて表示するために用いられる
2	AWS	Amazon が提供するクラウドコンピューティングサービスの総称。サーバー、ストレージ、データベースなどの機能を提供し、本システムではデータ管理や処理基盤として利用される
3	CityGML 形式	3D 都市モデルを標準的に記述するための XML ベースのデータ形式。建物・道路・地形などの情報を階層的に定義可能
4	GNSS	Global Navigation Satellite System の略。GPS などの衛星測位システムの総称。撮影位置や端末位置の取得に用いられる
5	UUID	情報を一意に識別するための識別子。本システムでは、建築物や建物面などの識別に用いられる
6	UV 展開	3D モデルの表面を平面に展開する技術。画像（テクスチャ）を 3D モデルに貼り付けるために使う
7	LOD	3D 都市モデルにおける詳細度の区分。PLATEAU では、LOD1 は簡易的な建物形状、LOD2 は屋根形状を含む建物モデル、LOD3 は窓や開口部などの詳細な外観を含むモデルとして定義されている

B) 五十音順

表 11-2 用語集（五十音順）

No.	用語	説明
1	オルソ画像	航空写真やドローン画像などを地図のように「真上から見た形」に補正した画像
2	建築物モデル	建物の形状・構造を 3D データ化したもの。壁、屋根などの構成要素を持つ
3	3次メッシュ	日本の標準地域メッシュコードにおける区分の一つで、約 10km 四方の 2 次メッシュをさらに分割した約 1km×1km の地域単位。PLATEAU では、データ管理や配信、処理単位として用いられており、建物モデルや都市モデルデータを分割して扱う際の基準となる

4	正射変換	撮影位置や角度の情報を基に、写真の歪み（遠近感）を取り除き、正面から見た形に変換する処理
5	正射投影画像	正射変換を施した画像
6	テクスチャ	3D モデルの表面に貼り付ける画像。質感やディテールを再現するために使われる
7	ワイヤーフレーム	3D モデルの形状を、面ではなく線（エッジ）で表現した表示方式。モデルの構造や面の位置関係を確認するために用いられる

以上

まちづくり DX の推進に向けた
3D 都市モデルへのテクスチャ自動付与技術の開発業務
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：株式会社シナスタジア