



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 技術検証レポート

Technical Report on PLATEAU SNAP

series
No. **119**

目次

1. プロジェクトの概要	- 1 -
1-1. 現状と課題	- 1 -
1-2. 課題解決のアプローチ	- 2 -
1-3. 創出価値	- 3 -
1-4. 想定事業機会	- 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	- 4 -
2-4. 実施体制	- 5 -
2-5. 実証エリア	- 5 -
2-6. スケジュール	- 6 -
3. 開発スコープ	- 7 -
3-1. 概要	- 7 -
3-2. 開発内容	- 7 -
4. 開発するシステム（機能要件）	- 8 -
4-1. アーキテクチャ	- 8 -
4-1-1. システム及びデータアーキテクチャ	- 8 -
4-1-2. ハードウェアアーキテクチャ	- 9 -
4-2. システム機能	- 12 -
4-2-1. システム機能一覧	- 12 -
4-2-2. 利用するソフトウェア・ライブラリ	- 12 -
4-2-3. 開発機能の詳細要件	- 14 -
4-3. アルゴリズム	- 23 -
4-3-1. 利用したアルゴリズム	- 23 -
4-3-2. 開発したアルゴリズム	- 23 -
4-4. データインタフェース	- 24 -
4-4-1. ファイル入力インタフェース	- 24 -
4-4-2. ファイル出力インタフェース	- 24 -
4-4-3. 内部連携インタフェース	- 24 -
4-4-4. 外部連携インタフェース	- 27 -
4-5. 実証に用いたデータ	- 28 -
4-5-1. 活用したデータ一覧	- 28 -
4-5-2. 生成・変換したデータ	- 28 -
4-6. ユーザーインタフェース	- 29 -
4-6-1. 画面一覧	- 29 -

4-6-2. 画面遷移図.....	29 -
4-6-3. 各画面仕様詳細	30 -
4-7. 実証システムの利用手順.....	35 -
4-7-1. 実証システムの利用フロー	35 -
4-7-2. 各画面操作方法	36 -
5. 開発するシステム（非機能要件）	42 -
5-1. 社会実装に向けた非機能要件	42 -
6. 品質.....	43 -
6-1. 機能要件の品質担保	43 -
6-2. 非機能要件の品質担保	43 -
7. 開発するシステムの機能要件の検証	44 -
7-1. 撮影可能面一覧取得・評価機能および AR での面表示機能.....	44 -
7-1-1. 検証目的.....	44 -
7-1-2. KPI	44 -
7-1-3. 検証方法.....	44 -
7-1-4. 検証結果.....	46 -
8. 開発するシステムの非機能要件の検証.....	53 -
8-1. 検証目的	53 -
8-2. KPI.....	53 -
8-2-1. 検証方法と検証シナリオ	53 -
8-2-2. 検証結果.....	53 -
9. 有用性検証	54 -
9-1. 検証目的	54 -
9-2. 検証方法	54 -
9-3. 被験者	55 -
9-4. ヒアリング・アンケートの詳細	55 -
9-4-1. アジェンダ・タイムテーブル	55 -
9-4-2. アジェンダの詳細	55 -
9-4-3. 検証項目と評価方法.....	56 -
9-4-4. 実証実験の様子	57 -
9-5. 検証結果	59 -
10. 成果と課題	68 -
10-1. 本実証で得られた成果	68 -
10-2. 実証実験で得られた課題と対応策	68 -
10-3. 今後の展望	69 -
11. 用語集.....	70 -

1. プロジェクトの概要

1-1. 現状と課題

近年、都市のデジタルツイン実現に向けた取り組みが進んでおり、3D 都市モデルの整備手法も徐々に確立されつつある。これにより、さまざまな政策領域や民間事業領域での活用可能性が示されているものの、特に精度の高いテクスチャ生成に関しては依然として課題が多い。例えば、航空測量成果を活用する 3D 都市モデルでは、撮影高度の影響で画質が不十分であること、また建物の側面が映らないことが多いことが挙げられ、アイレベルでの景観再現に際しては、追加の加工が不可欠である。そのため、現在の都市空間における表示や評価では、リアルな視点からの品質やリアリティが求められる一方で、それに十分対応できていない課題がある。

さらに、高精度なテクスチャを取得するためには MMS（モービルマッピングシステム）の活用や徒歩での撮影が一般的であるが、これには高価な機材と専門的な技術が必要となり、3D 都市モデルを広範囲に作成するには、コスト面での課題が生じる。また、デジタルツインの社会実装を進めるためには、その利活用について、行政だけでなく民間も含めた幅広い領域で行うことが求められている。そのため、テクスチャの精度向上と整備コストの削減を両立し、デジタルツインの利便性を高めるための環境を整えることが期待される。



図 1-1 高精度なテクスチャ取得における課題のイメージ

1-2. 課題解決のアプローチ

- 本業務では、一般的な撮影機器及び撮影方法で取得可能な画像を活用し、従来よりもハイクオリティなデジタルツインデータを抽出・生成する方策を検討することで、3D 都市モデルを利用した開発環境のさらなる改善を目指す
- 具体的には、スマートフォンで撮影した一般的な画像を基に 3D 都市モデルの建物地物のテクスチャ（地物の外観）を抽出・生成し、3D 都市モデルのデータベースに登録・蓄積可能なシステム「PLATEAU SNAP」を開発する
- 同ツールをユーザー参加型の設計とすることで、クラウドソーシング型の高品質テクスチャ収集スキームを開発する
- 開発物の実証として、本システムを利用してデータベースに登録したテクスチャを 3D 都市モデル内の既存のテクスチャと置換し、3D 都市モデルを更新することが可能かを検証する。その際、更新した 3D 都市モデルを実際に画面上で表示し、新しいテクスチャが正しく適用されているか、表示に問題がないかを目視で確認する



図 1-2 ユースケースイメージ

1-3. 創出価値

- 本開発対象であるテクスチャ生成システム「PLATEAU SNAP」によって、測量の専門家ではない一般利用者の一般的なスマートフォンでのテクスチャ撮影による 3D 都市モデルのデータ整備・拡充が可能になる。これによって、テクスチャ精度・品質が相対的に低い空撮による計測や MMS 等を用いた高コストな地上測量とは異なる新たなデータ整備方式を提供する
- 自治体や民間事業者自身が必要なテクスチャを必要とする際に、独力で撮影・整備できるようになるほか、自治体や民間事業者がテクスチャデータ撮影を安価に発注し、一般利用者が受注・撮影を行うといったビジネスモデルへの発展も考えられる

1-4. 想定事業機会

表 1-1 想定事業機会

項目	内容
利用事業者	● 自治体職員 ● エンドユーザー
提供価値	● 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・整備費用の削減 ● エンドユーザーの巻き込みによる整備スピードの向上
サービス仮説	● 自治体職員自らが 3D 都市モデルを整備する機会の創出 ➢ 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、テクスチャ割当てを行う機能を整備することで、整備費用の削減を図ることができる ➢ 3D 都市モデルにおいて、すべての建物ではなく、テクスチャの不十分な建物のみを撮影することにより、整備費用削減や作業効率化なども可能となる ● エンドユーザーによる 3D 都市モデルの整備の促進 ➢ エンドユーザーによる地元エリアの建物の写真撮影・登録、というサイクルで 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

- 公共測量データを基に整備される 3D 都市モデルの建物や道路のテクスチャは、航空測量データを基に整備されているが、高い高度からの空撮では建物の側面が十分な精度で撮影されないため、建物側面のテクスチャ整備が遅れている
- 建物側面のテクスチャについては過去の公共測量データにも十分存在しているとは言えず、MMS や徒歩での測量による収集の必要があるが、高コストで十分な整備が難しい
- 本開発対象であるテクスチャ生成システム「PLATEAU SNAP」により、高コストな専用機器を用いた測量発注が不要となり、建物のテクスチャをスマートフォンでの撮影により高精細なものに改良することが可能になる

2-2. 実証フロー



図 2-1 実証フロー

2-3. 検証ポイント

- 本実証では以下の点に着目し、開発システムの利用者に対するアンケートやヒアリングを通じて検証を行うこととする
 - テクスチャ品質（有用性）
 - 使いやすさ（ユーザビリティ）

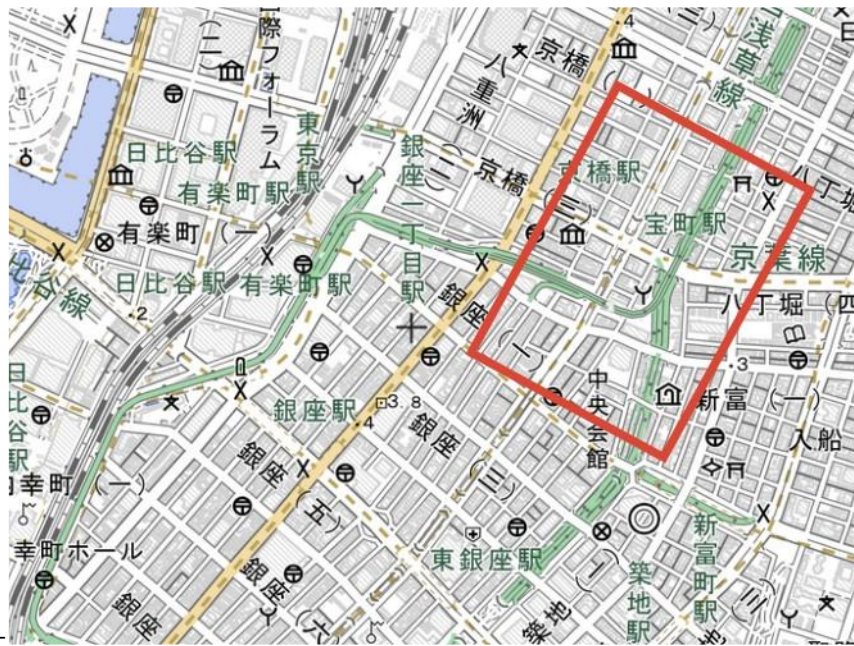
2-4. 実施体制

表 2-1 実施体制

役割	主体	詳細
全体管理	デジタル庁	● プロジェクト全体ディレクション
	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	● 全体管理 ● プロジェクト・ビジョン定義 ● ソフトウェア開発

2-5. 実証エリア

表 2-2 実証エリア

項目	内容
実証地	東京都中央区
面積	1 km ²
マップ (対象エリア は赤枠内)	

2-6. スケジュール

表 2-3 スケジュール

実施事項	2024 年				2025 年		
	9 月	10 月	11 月	12 月	1 月	2 月	3 月
1. 要件定義、実証計画	←→						
2. PLATEAU SNAP プロトタイプ開発			←→	←→	←→		
3. 実証実験						←→	
4. PLATEAU SNAP 正式版開発)						←→	
5. 成果の取りまとめ							←→

3. 開発スコープ

3-1. 概要

本プロジェクトでは、一般利用者がスマートフォンで建築物を撮影する場面において、VPS (Visual Positioning System) を活用して撮影地点と方位の情報を取得・計算し、撮影対象とする建築物と面を推定する機能の開発、撮影位置の情報と面の位置・法線方向から正射投影画像を生成する機能の開発等から、効率的で迅速な 3D 都市モデルにおけるテクスチャの整備を進める。

これにより、高価な専用機器や従来の測量手法 (MMS や徒歩測量) を使用することなく、スマートフォンなどの身近なデバイスを活用して建物側面のテクスチャを収集できるため、大幅なコスト削減が見込まれる。また、撮影した画像の品質を評価し、適切な建物面に自動で貼り付け可能な機能を活用することで、3D 都市モデルの精度の向上を見込むことができる。特に、これまで撮影が難しかった建物の側面に対して、細部まで高品質なテクスチャが適用できるため、リアルで詳細な 3D 都市モデルの作成が可能となる。一方、一般利用者が 3D 都市モデルやの作成に直接貢献することで、これらの技術への関心が高まり、社会全体での普及や活用が促進されと考えられる。

3-2. 開発内容

本プロジェクトでは、スマートフォンで撮影した一般的な画像を基に 3D 都市モデルの建物地物のテクスチャ (地物の外観) を抽出・生成し、3D 都市モデルのデータベースに登録・蓄積可能なシステム「PLATEAU SNAP」を開発した。具体的には、下記の機能を開発した。

1. カメラ映像を活用し、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に推定する VPS 機能
2. スマートフォンで撮影した際に、画像をテクスチャとして適用可能な面を計算し、その品質を評価する撮影可能面一覧取得・評価機能
3. クラウドから撮影可能な面情報と品質評価値を取得し、カメラ映像に AR 表示する AR での面表示機能
4. 建物の撮影対象面を選択し、その画像を取得できる建物撮影機能
5. 撮影した画像と貼り付け先の面情報をデータベースにアップロードする画像アップロード機能

4. 開発するシステム（機能要件）

4-1. アーキテクチャ

4-1-1. システム及びデータアーキテクチャ

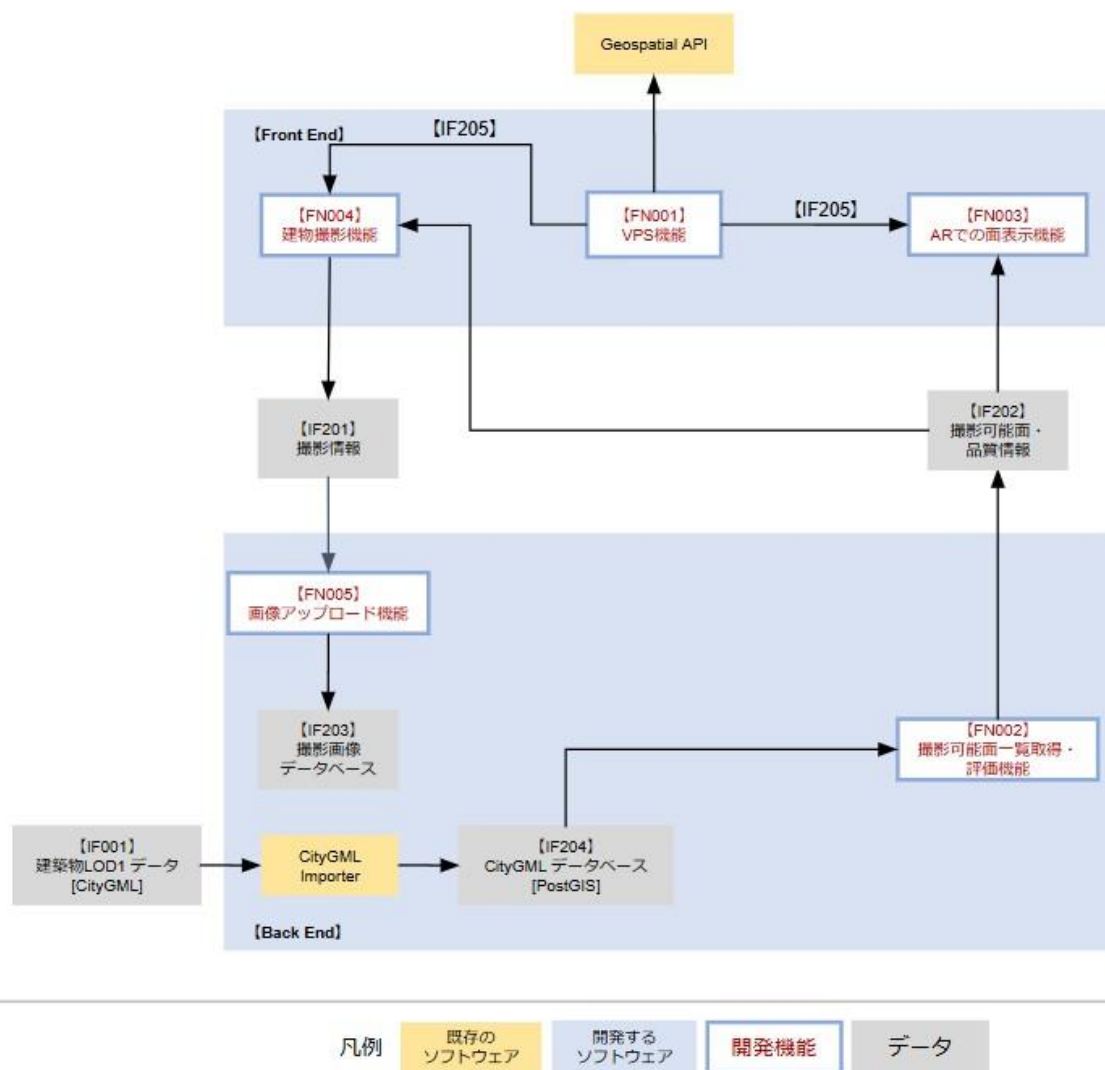


図 4-1 システム及びデータアーキテクチャ

4-1-2. ハードウェアアーキテクチャ

4-1-2-a. 利用するハードウェア一覧

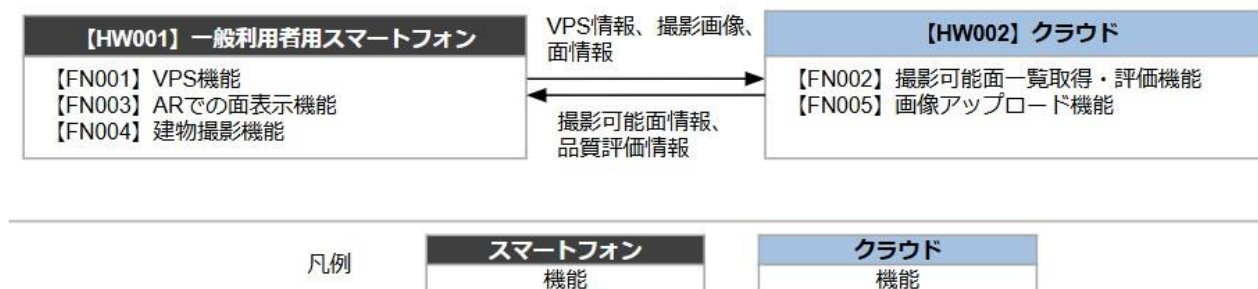


図 4-2 ハードウェアアーキテクチャ

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, FN003, FN004
HW002	クラウド	AWS	FN004, FN005

4-1-2-b. 利用するハードウェア詳細

【HW001】 iPhone 13 Pro

- 選定理由
 - 実証システムを利用するのに十分なスペックであるため
- 仕様・スペック
 - CPU 及び GPU： A15 Bionic
 - メモリ：6GB
 - ストレージ：256GB
 - OS：iOS 18.0.1（2024/10/28 時点での最新バージョン）
 - 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、HEIF
 - 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 4-3 iPhone 13 Pro¹

¹ 公式 HP より抜粋： <https://support.apple.com/ja-jp/111871>

【HW002】 AWS

- 選定理由
 - 実証システムを利用するのに十分なスペックであるため
- 仕様・スペック
 - EC2
 - ✧ バックエンドサーバーを構築する
 - ✧ スペック: 処理内容に応じて増減
 - RDS
 - ✧ CityGML データベースを構築する
 - ✧ スペック: 処理内容に応じて増減
 - S3
 - ✧ 撮影画像を保存する
 - ✧ スペック: 画像枚数に応じて増減

4-2. システム機能

4-2-1. システム機能一覧

表 4-2 システム機能一覧

ID	機能名	機能説明
FN001	VPS 機能	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	現在位置で撮影した際に、画像をテクスチャとして貼り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	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	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	画像アップロード機能	撮影画像と貼り付け先の面情報をデータベースにアップロードする機能

4-2-2. 利用するソフトウェア・ライブラリ

表 4-3 利用するソフトウェア・ライブラリ

ID	項目	内容
SL001	C#	- 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 - GitHub (https://github.com/dotnet/csharp-lang)
SL002	Unity	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 - 公式サイト (https://unity.com/ja)
SL003	Java	- Back End 開発用のプログラミング言語 - GitHub (https://github.com/openjdk)
SL004	Google ARCore XR Plugin	- Geospatial API を利用するためのプラグイン - GitHub (https://github.com/google-ar/arcore-unity-extensions)
SL005	CityGML Importer	- CityGML データをデータベースに読み込むためのツール - GitHub (https://github.com/3dcitydb/3dcitydb-docs/blob/release-v4.1/source/impexp/citygml-import.rst)
SL006	PostgreSQL	- データベース管理システム - GitHub (https://github.com/postgres/postgres)
SL007	PostGIS	PostgreSQL で地理情報システム (GIS) を扱えるようにするための拡張機能

		● GitHub (https://github.com/postgis/postgis)
SL008	NetTopologySuite	● .NET 用の GIS ライブラリ ● GitHub (https://github.com/NetTopologySuite/NetTopologySuite)
SL009	Npgsql	● ADO.NET ドライバ ● GitHub (https://github.com/npgsql/npgsql)
SL010	AWSSDK	● .NET 用の AWS にアクセスするための SDK ● GitHub (https://github.com/aws/aws-sdk-net)
SL011	Swashbuckle.AspNetCore	● .NET 用の Swagger ライブラリ ● GitHub (https://github.com/domaindrivendev/Swashbuckle.AspNetCore)
SL012	create_gpkg_for_city_boundaries	● 境界データから平面直角座標系番号を取得するための中間データを作成するツール ● GitHub (https://github.com/MIERUNE/create_gpkg_for_city_boundaries)
SL013	Docker	● 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 ● 公式サイト (https://www.docker.com/)
SL014	Docker Compose	● 複数の Docker コンテナを簡単に管理・起動できるツール ● GitHub (https://github.com/docker/compose)
SL015	AR Foundation	● AR 機能実装に使用 ● 公式サイト (https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/index.html)
SL016	ARCore Extensions	● AR 機能の拡張を提供 (ARCore が依存) ● GitHub (https://github.com/google-ar/arcore-unity-extensions)
SL017	UniTask	● 非同期タスクの実装に使用 ● GitHub (https://github.com/Cysharp/UniTask/releases/tag/2.5.10)
SL018	R3	● リアクティブな実装に使用 ● GitHub (https://github.com/Cysharp/R3/releases/tag/1.2.9)
SL019	Geometry	● メッシュ描画に使用 ● GitHub (https://github.com/iShapeUnity/Geometry/releases/tag/0.0.5)
SL020	Mesh2d	● メッシュ描画に使用 ● GitHub (https://github.com/iShapeUnity/Mesh2d/releases/tag/0.0.9)
SL021	Triangulation	● メッシュ描画に使用 ● GitHub (https://github.com/iShapeUnity/Triangulation/releases/tag/0.0.8)

		)
SL022	Addressables	- リソース管理に使用 - 公式サイト (https://docs.unity3d.com/Packages/com.unity.addressables@1.22/manual/index.html)
SL023	Localization	- 文字列のローカライズに使用 - 公式サイト (https://docs.unity3d.com/Packages/com.unity.localization@1.5/manual/index.html)
SL024	TextMeshPro	- 文字列 UI 描画（メッシュベース）に使用 - 公式サイト (https://docs.unity3d.com/ja/2022.3/Manual/com.unity.textmeshpro.html)
SL025	VContainer	- Dependency Injection に使用 - GitHub (https://github.com/hadashiA/VContainer/releases/tag/1.16.8)
SL026	Polly	- OpenAPI で生成されたコードのコンパイルに必要 - GitHub (https://github.com/App-vNext/Polly)

4-2-3. 開発機能の詳細要件

1) 【FN001】 VPS 機能

- 機能概要
 - 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
- フローチャート

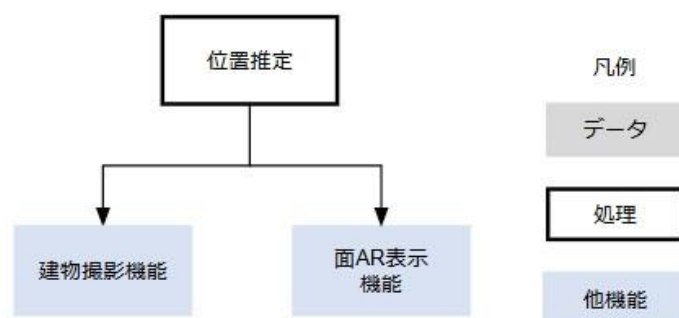


図 4-4 VPS 機能のフローチャート

- データ仕様
 - 入力

◇ なし

➤ 出力

◇ 位置情報

- 内容

- VPS により推定された位置・姿勢をフロントアプリ内で連携する

- 形式

- 関数呼出し

- データ詳細

- 【IF202】を参照

- 機能詳細

- 位置推定

- ◇ 処理内容

- Google Geospatial API を用いた位置・姿勢の推定処理は、スマートフォンのセンサー情報とカメラ映像を組み合わせで行う。まず、GPS によって現在のおおよその位置を特定する

- その後、カメラを利用して現実空間の映像をリアルタイムで取得し、画像解析によって映像内の特徴点（建物やランドマークなどの目印となるポイント）を検出する

- 検出した特徴点を Google が保持する膨大なストリートビュー画像データと照合し、カメラ映像に映るランドマークや建物とストリートビュー画像の一致点を特定することで、スマートフォンの正確な位置を推定する

- さらに、この位置情報に IMU（※）から得た角速度や加速度のデータを統合し、スマートフォンの姿勢（向きや傾き）を計算する。これらの処理はスマートフォンの移動に応じてリアルタイムで更新されるため、本機能では緯度、経度、高度といった位置情報に加え、方位角、ピッチ、ロールといった姿勢情報を継続的に提供する

- 具体的な手法は、Google Geospatial API の内部実装に基づく

- ◇ 利用するライブラリ

- 【SL004】Google ARCore XR Plugin

- ◇ 利用するアルゴリズム

- なし

※IMU（ Inertial Measurement Unit：慣性計測装置）・・・ 物体の姿勢、加速度、角速度を測定する装置

2) 【FN002】撮影可能面一覧取得・評価機能

- 機能概要

- 現在位置で撮影した際に、画像をテクスチャとして貼り付け可能な面を計算し、画像の品質を評価する機能

- フローチャート

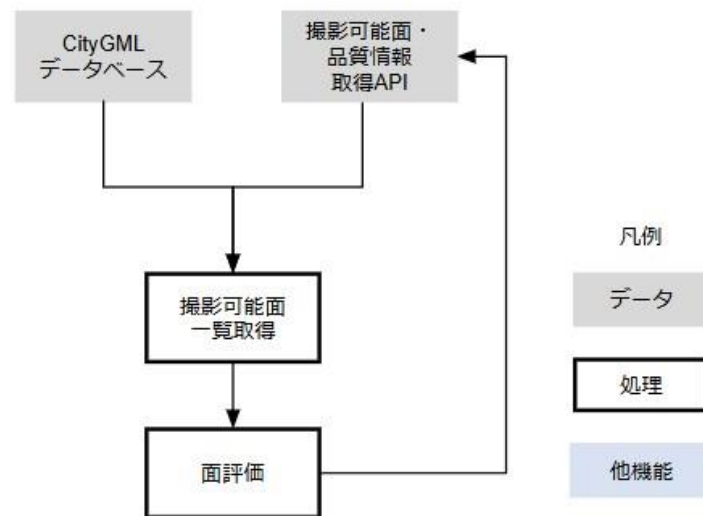


図 4-5 撮影可能面一覧取得・評価機能のフローチャート

- データ仕様
 - 入力
 - ◇ 撮影可能面・品質情報取得 API
 - 内容
 - 【FN001】により推定されたスマートフォンの位置情報及び方位情報（緯度、経度、高度及び方向）を送信し、撮影可能な面の一覧及び評価値を取得する
 - 形式
 - HTTP API
 - データ詳細
 - 【IF202】を参照
 - ◇ CityGML データベース
 - 内容
 - CityGML Importer により読み込んだ建築物 LOD1 の CityGML データ
 - 形式
 - PostGIS
 - データ詳細
 - 【IF204】を参照
 - 出力
 - ◇ 撮影可能面・品質情報取得 API
 - 内容
 - 【FN001】により推定されたスマートフォンの位置情報及び方位情報（緯度、経度、高度及び方向）を送信し、撮影可能な面の一覧及び評価値を取得する
 - 形式

- HTTP API
- レスポンスには JSON を含む
- データ詳細
 - 【IF202】を参照
- 機能詳細
 - 撮影可能面一覧取得
 - ◇ 処理内容
 - 具体的な処理内容は【AL001】を参照
 - ◇ 利用するライブラリ
 - 【SL007】 PostGIS
 - 【SL008】 NetTopologySuite
 - 【SL009】 Npgsql
 - ◇ 利用するアルゴリズム
 - 【AL001】 撮影可能面一覧取得
 - 面評価
 - ◇ 処理内容
 - 具体的な処理内容は【AL002】を参照
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - 【AL002】 面評価

3) 【FN003】AR での面表示機能

- 機能概要
 - 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
- フローチャート

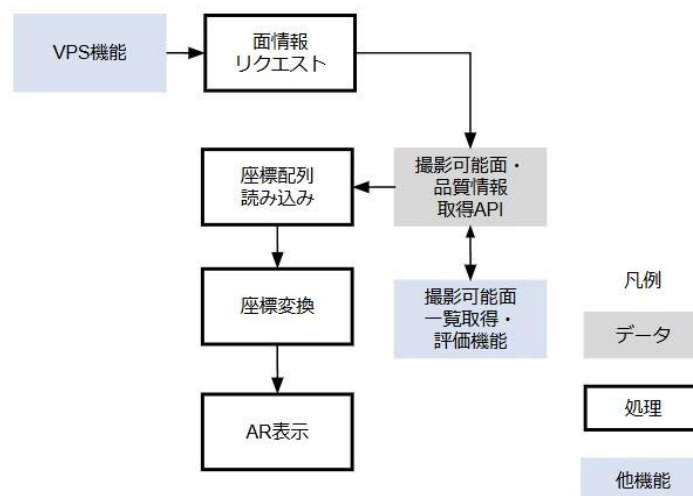


図 4-6 AR での面表示機能のフローチャート

- データ仕様
 - 入力
 - ◇ 撮影可能面・品質情報取得 API
 - 内容
 - 【FN001】により推定されたスマートフォンの位置情報及び方位情報（緯度、経度、高度及び方向）を送信し、撮影可能な面の一覧及び評価値を取得する
 - 形式
 - HTTP API
 - データ詳細
 - 【IF202】を参照
 - 出力
 - ◇ なし
- 機能詳細
 - 面情報リクエスト
 - ◇ 処理内容
 - 現在の位置で撮影可能な面の情報を GET リクエストでサーバーサイドから受信する
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
 - 座標配列読み込み
 - ◇ 処理内容
 - 面の頂点座標情報を読み込む
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし
 - 座標変換
 - ◇ 処理内容
 - 面の頂点座標を地理座標系（緯度、経度、高度）から AR 空間座標系に変換する
 - ◇ 利用するライブラリ
 - 【SL004】 Google ARCore XR Plugin
 - 【SL015】 AR Foundation
 - ◇ 利用するアルゴリズム
 - なし
 - AR 表示
 - ◇ 処理内容

- 撮影対象の建物の面の 3D モデルを実際の建物の面に重なるようにカメラ映像に表示する
- ✧ 使用するライブラリ
 - 【SL019】 Geometry
 - 【SL020】 Mesh2d
 - 【SL021】 Triangulation
- ✧ 使用するアルゴリズム
 - なし

4) 【FN004】建物撮影機能

- 機能概要
 - 撮影対象の面を選択し、建物の面の画像を撮影する機能
- フローチャート

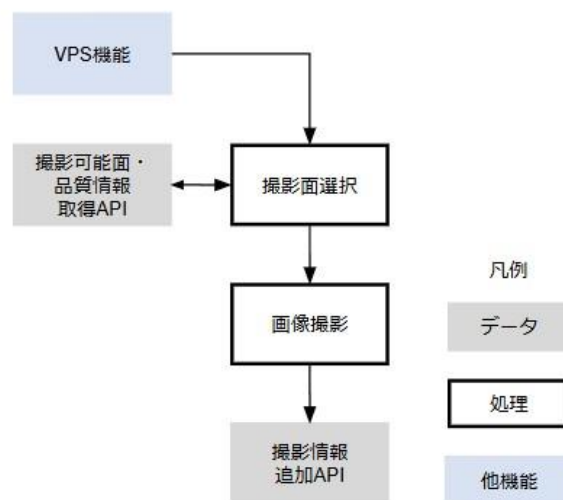


図 4-7 建物撮影機能のフローチャート

- データ仕様
 - 入力
 - ✧ 撮影可能面・品質情報取得 API
 - 内容
 - 【FN001】により推定されたスマートフォンの位置情報及び方位情報（緯度、経度、高度及び方向）を送信し、撮影可能な面の一覧及び評価値を取得する
 - 形式
 - HTTP API
 - データ詳細
 - 【IF202】を参照
 - ✧ 位置情報
 - 内容
 - VPSにより推定された位置・姿勢をフロントアプリ内で連携する

- 形式
 - 関数呼出し
- データ詳細
 - 【IF202】を参照
- 出力
 - ◇ 撮影情報追加 API
 - 内容
 - カメラで撮影した静止画
 - 撮影対象として選択した面を識別する情報
 - 形式
 - HTTP API
 - データ詳細
 - 【IF201】を参照
- 機能詳細
 - 撮影面選択
 - ◇ 処理内容
 - 撮影対象となる建物の面の 3D モデルを画面上でタップした際に、タップ位置からどの建物の面を選択したか特定する
 - ◇ 利用するライブラリ
 - 【SL019】 Geometry
 - 【SL020】 Mesh2d
 - 【SL021】 Triangulation
 - ◇ 利用するアルゴリズム
 - なし
 - 画像撮影
 - ◇ 処理内容
 - カメラで画像を撮影する
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - なし

5) 【FN005】画像アップロード機能

- 機能概要
 - 撮影画像と貼り付け先の面情報をクラウドにアップロードする機能
- フローチャート

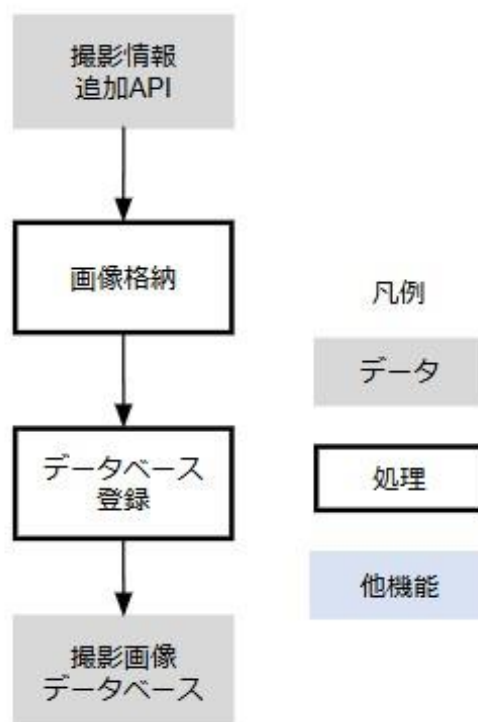


図 4-8 画像アップロード機能のフローチャート

- データ仕様
 - 入力
 - ◇ 撮影情報追加 API
 - 内容
 - 撮影した画像とそれに紐づく情報（撮影位置・姿勢）を追加するための API
 - 形式
 - HTTP API
 - データ詳細
 - 【IF201】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紐付けたデータベース
 - 形式

- PostgreSQL
 - データ詳細
 - 【IF203】を参照
- 機能詳細
 - 画像格納
 - ◇ 処理内容
 - 画像を AWS S3 にアップロードする
 - ◇ 利用するライブラリ
 - 【SL010】AWSSDK
 - 【SL013】Docker
 - 【SL014】Docker Compose
 - ◇ 利用するアルゴリズム
 - なし
 - データベース登録
 - ◇ 処理内容
 - 画像の URL と紐づく情報（対象面の gml:id、撮影位置・姿勢）をデータベースに登録する
 - ◇ 利用するライブラリ
 - 【SL008】NetTopologySuite
 - 【SL009】Npgsql
 - ◇ 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

なし。

4-3-2. 開発したアルゴリズム

1) 【AL001】撮影可能面一覧取得

- 本アルゴリズムを利用した機能
 - 【FN002】
- アルゴリズムの詳細
 - 【FN001】によって取得したカメラ位置を基に、カメラに映し出される建物の各面の頂点座標や法線ベクトルを CityGML データベースから取得する
 - 【FN001】によって取得したカメラの位置、方位、および視野角を基に、カメラの視野範囲を 3D 空間上に表現する視錐台を定義する
 - 建物の面とカメラとの距離が 2m から 30m 程度の範囲内にある面を絞り込む
 - 各面の法線ベクトルとカメラの方向ベクトルとのなす角を計算し、なす角が 120 度から 180 度程度の範囲に入る面を絞り込む
 - 上記の条件を満たす面を撮影可能な面としてリストアップする

2) 【AL002】面評価

- 本アルゴリズムを利用した機能
 - 【FN002】
- アルゴリズムの詳細
 - 以下の評価項目について、両方の基準を満たした場合に登録可能な画像として判定する
 - ◇ 撮影角度
 - 評価方法：撮影対象の面の法線ベクトルとカメラの方向ベクトルを基に、面とカメラのなす角（撮影角度）を計算する。面の真正面にカメラが向いている場合の撮影角度を 0 度とし、以下の基準で評価する
 - 撮影角度が±60 度以内の場合：「○」
 - 撮影角度が±60 度を超過している場合：「×」
 - ◇ 面の欠け
 - 評価方法：撮影対象の面の頂点がどれだけ画角に収まっているかを計算し、以下の基準で評価する
 - 全ての頂点が画角に収まっている場合：「○」
 - 一部の頂点が画角に収まっていない場合：「×」

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】 建築物 LOD1 モデル

撮影画像を適用する建築物モデル

4-4-2. ファイル出力インタフェース

なし。

4-4-3. 内部連携インタフェース

1) 【IF201】 撮影情報

スマートフォンのカメラで撮影した画像の情報を POST リクエストでサーバーサイドに送信する。

- 本インタフェースを利用する機能
 - 【FN004】、【FN005】
- エンドポイント：POST /api/building-image
- リクエスト：
 - multipart/form-data 形式で以下のパートを含む
 - ✧ image（必須）
 - 型：ファイル
 - 説明：建物面を撮影した画像ファイル
 - ✧ metadata（必須）
 - 型：文字列（JSON 形式）
 - 説明：以下の内容を含む画像に関連するメタデータ

```
{
  "gmlid": "string",
  "from": {
    "longitude": number, // カメラの緯度、-90.0 ～ 90.0
    "latitude": number, // カメラの経度、-180.0 ～ 180.0
    "altitude": number // カメラの高度、メートル単位
  },
  "to": {
    "longitude": number, // カメラの向く方向の緯度、90.0 ～ 90.0
    "latitude": number, // カメラの向く方向の経度、-180.0 ～ 180.0
    "altitude": number // カメラの向く方向の高度、メートル単位
  },
}
```

```
"role": number    // カメラの傾き
"timestamp": "timestamp" // 撮影日時 ISO 8601 形式
}
```

2) 【IF202】撮影可能面・品質情報

現在の位置で撮影可能な面の情報を GET リクエストでサーバーサイドから受信する。

- 本インタフェースを利用した機能
 - 【FN002】、【FN003】、【FN004】
- エンドポイント：GET /api/visible-surfaces
- リクエスト：

```
{
  "from": {
    "longitude": number, // カメラの緯度、-90.0 ~ 90.0
    "latitude": number,  // カメラの経度、-180.0 ~ 180.0
    "altitude": number    // カメラの高度、メートル単位
  },
  "to": {
    "longitude": number, // カメラの向く方向の緯度、90.0 ~ 90.0
    "latitude": number,  // カメラの向く方向の経度、-180.0 ~ 180.0
    "altitude": number    // カメラの向く方向の高度、メートル単位
  },
  "field_of_view": number, (任意、デフォルト値 45.0) // カメラの視野角（度数法）
  "max_distance": number   (任意、デフォルト値 30.0) // 検出する最大距離（メートル）
}
```

- レスポンス：

```
{
  "surfaces": [
    {
      "gmlid": "string",
      "coordinates": [
        [
          [number, number, number], // 面の頂点 1 の経度, 緯度, 高度
          [number, number, number], // 面の頂点 2 の経度, 緯度, 高度
          //その他の頂点
        ]
      ],
    },
    //その他の面
  ]
}
```



```

]
}

```

3) 【IF203】撮影画像データベース

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した機能

➤ 【FN005】

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

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

3DCityDB の仕様 (<https://www.3dcitydb.org/3dcitydb/>) 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した機能

➤ 【FN002】

5) 【IF205】位置情報

VPS から推定された位置・姿勢情報をフロントアプリケーション内で連携する。

- 本インタフェースを利用した機能

➤ 【FN001】、【FN003】、【FN004】

4-4-4. 外部連携インターフェース

なし。

4-5. 実証に用いたデータ

4-5-1. 活用したデータ一覧

1) 利用する 3D 都市モデル

- 年度：2023 年度
- 都市名：中央区
- ファイル名：13102_chuo-ku_city_2023_citygml_1_op
- メッシュ番号：中央区（LOD1 範囲のすべて）
- URL：<https://www.geospatial.jp/ckan/dataset/plateau-13102-chuo-ku-2023>

図 4-9 利用する 3D 都市モデルの仕様

地物	地物型	属性区分	属性名	データを利用した機能（ID）
建築物 LOD1	Building	空間属性	bldg:lod1Solid	FN002、FN003、FN004

4-5-2. 生成・変換したデータ

なし。

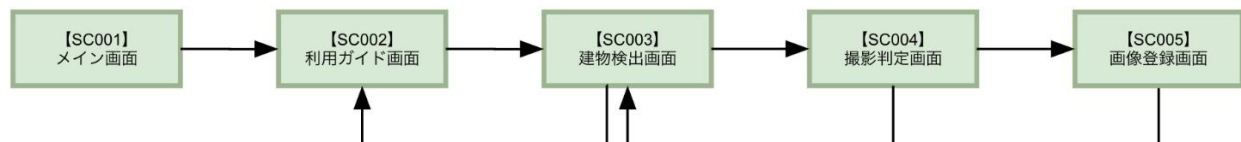
4-6. ユーザーインターフェース

4-6-1. 画面一覧

表 4-4 画面一覧

画面 ID	画面名	利用機能	画面説明
SC001	メイン画面	-	アプリ起動時に表示する画面
SC002	利用ガイド画面	-	アプリの基本的な操作方法や注意事項を表示する画面
SC003	建物検出画面	【FN001】 VPS 機能	カメラ映像及び撮影可能な面を AR 表示し、画像を撮影する画面
		【FN002】 撮影可能面一覧取得・評価機能	
		【FN003】 AR での面表示機能	
		【FN004】 建物撮影機能	
SC004	撮影判定画面	【FN002】 撮影可能面一覧取得・評価機能	画像とその品質を表示する画面
		【FN005】 画像アップロード機能	
SC005	画像登録画面	【FN005】 画像アップロード機能	画像の登録結果を表示する画面

4-6-2. 画面遷移図



凡例

画面

【SCxxx】
画面ID操作
→
画面遷移

図 4-10 画面遷移図

4-6-3. 各画面仕様詳細

1) スマートフォン用画面

1) 【SC001】メイン画面

- 画面の目的・概要
 - アプリ起動時に表示する画面
 - アプリ起動ボタンを押すと、建物検出画面に遷移する



図 4-11 メイン画面のイメージ

2) 【SC002】利用ガイド画面

- 画面の目的・概要
 - アプリの基本的な操作方法や注意事項を表示する画面
 - 「×」ボタンを押すと、建物検出画面へ遷移する

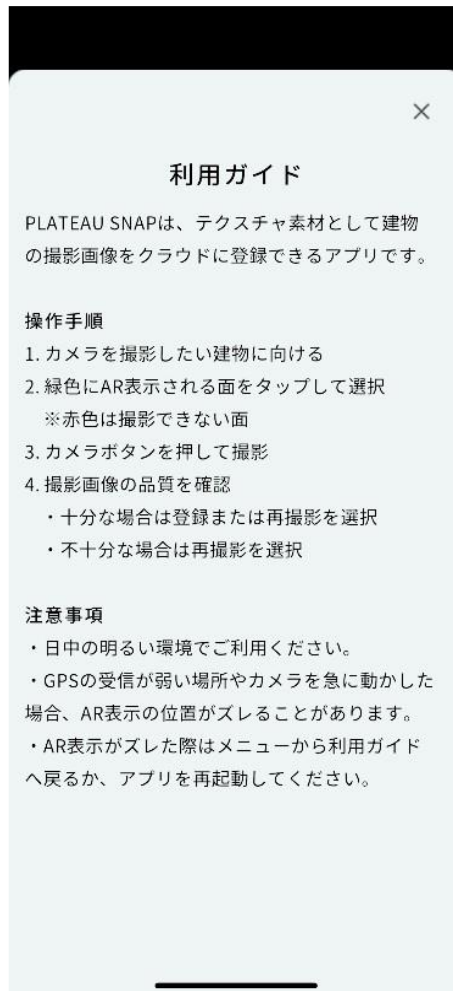


図 4-12 利用ガイド画面のイメージ

3) 【SC003】建物検出画面

- 画面の目的・概要
 - カメラ映像及び撮影可能な面を AR 表示し、画像を撮影する画面
 - カメラに建物を映すと、テクスチャの素材として撮影可能な面を青色で AR 表示し、撮影不可能な面を赤色で表示する
 - 青色の AR 面の中から撮影対象の面を選択すると、選択した AR 面が緑色に変化し、撮影ボタンが有効になる。撮影ボタンを押すと画像が撮影され、撮影判定画面に遷移する
 - メニューの利用ガイドボタンを押すと、利用ガイド画面に遷移する



図 4-13 建物検出画面のイメージ

4) 【SC004】撮影判定画面

- 画面の目的・概要
 - 画像とその品質を表示する画面
 - 登録ボタンを押すと、画像がデータベースにアップロードされ、画像登録画面に遷移する
 - 再撮影ボタンを押すと、画像は破棄され、建物検出画面に遷移する



図 4-14 撮影判定画面のイメージ

5) 【SC005】画像登録画面

- 画面の目的・概要
 - 画像の登録結果を表示する画面
 - 画像登録が完了した場合、再撮影ボタンを押すと、画像は破棄され、建物検出画面に遷移する
 - 画像登録エラーが発生した場合、登録ボタンを押すと、画像をデータベースに再アップロードする
再撮影ボタンを押すと、画像は破棄され、建物検出画面に遷移する



図 4-15 画像登録画面のイメージ

4-7. 実証システムの利用手順

4-7-1. 実証システムの利用フロー

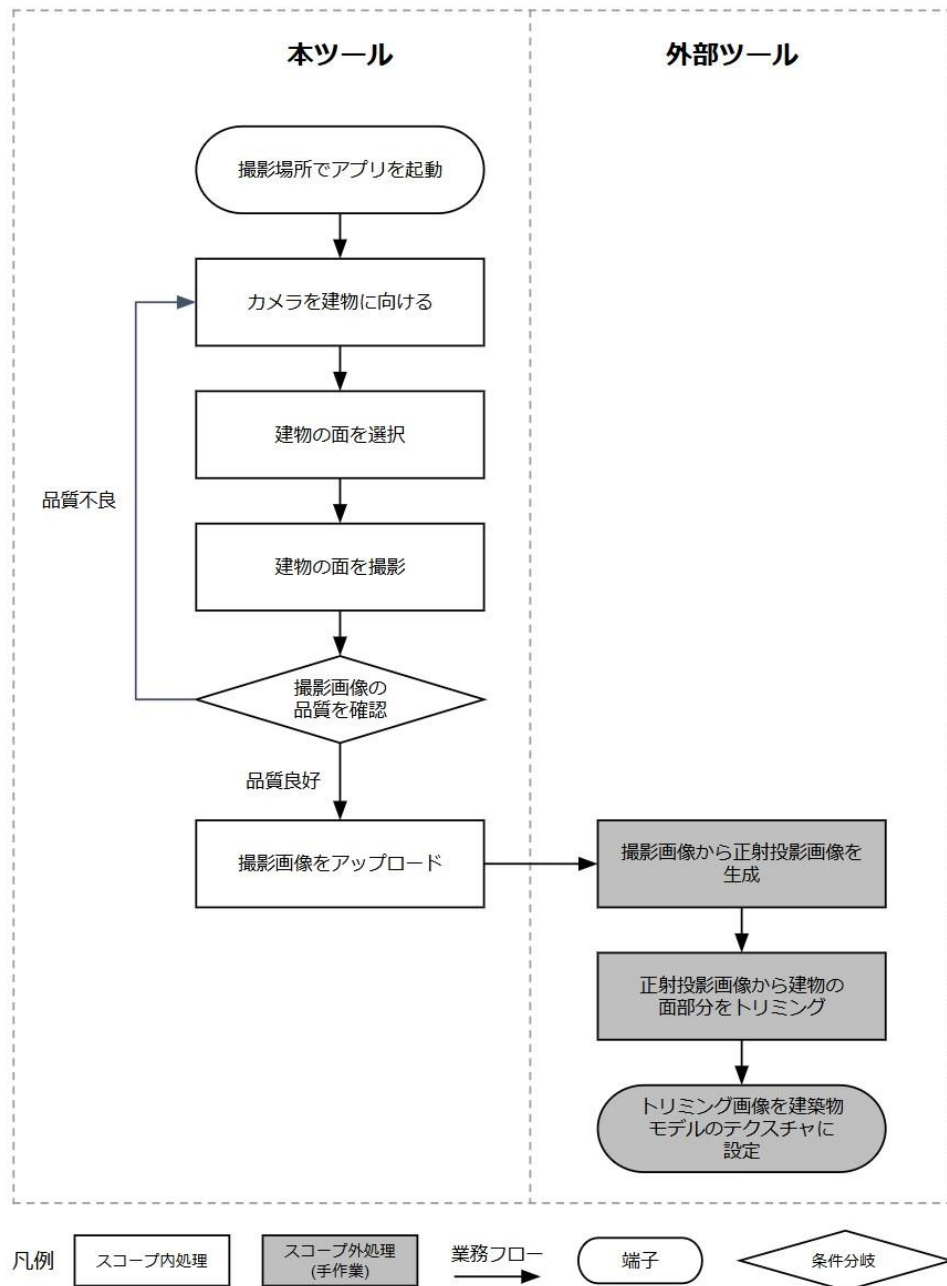


図 4-16 実証システムの利用フロー

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 4-17 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（下図赤枠）が画角に収まるようにカメラを向ける



図 4-18 建物検出画面（建物検出前）

3) 建物の面を選択

- 青色の AR 面の中から撮影対象の面をタップする



図 4-19 建物検出画面（建物検出後）

4) 建物の面を撮影

- タップした面が緑色に強調表示されたことを確認し、撮影ボタン（下図赤枠）を押す



図 4-20 建物検出画面（撮影対象面の選択後）

5) 撮影画像の登録（アップロード）

- 撮影画像の品質が十分な場合は、登録ボタン（下図赤枠）を押す
- 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 4-21 撮影判定画面（品質十分）

6) 登録結果の確認

- 画像の登録が完了した場合、再撮影ボタン（下図赤枠）を押す
- 画像の登録が失敗した場合、再度登録ボタンを押すか、再撮影ボタンを押す



図 4-22 画像登録画面（登録完了）

5. 開発するシステム（非機能要件）

5-1. 社会実装に向けた非機能要件

表 5-1 非機能要件一覧

カテゴリ	ID	項目	詳細
性能・拡張性	B001	動作環境	● 利用者が滞りなくシステムを利用できることを想定し、30 分以上の安定動作を行うこと

1) 【B001】動作環境

- 本非機能要件を適用するシステム
 - 撮影画像によるテクスチャ生成ツール
- 目標値
 - 安定動作時間 30 分
- 設定理由
 - 利用者が滞りなくシステムを利用するため
- 評価方法
 - 30 分の連続稼働を 2 回行い、フリーズ・クラッシュ等が発生しないことを確認する

6. 品質

6-1. 機能要件の品質担保

開発したシステムの総合テストを行い、システムの適格性及び有用性を評価する。この評価は、要件定義書に定める機能要件及び非機能要件の観点から行うことを原則とする。総合テストで適格性を確認した後、実証計画書に従い、システム実証を行う。

表 6-1 機能要件の品質担保方針

対象プロセス/ サブシステム	品質評価項目	目標値	期間の単位	アクティビティ
有用性	テクスチャ品質	既存の 3D 都市モデル (CityGML) 整備データ 内のテクスチャよりも高 品質なものに置換される こと	令和 7 年 1～2 月	既存の 3D 都市モデル (CityGML) と本実証で 作成した CityGML を可視 化し目視で比較

6-2. 非機能要件の品質担保

表 6-2 機能要件の品質担保方針

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
性能・拡張性	動作環境	安定動作時間 30 分	令和 7 年 1～2 月	総合テストによる検証

7. 開発するシステムの機能要件の検証

7-1. 撮影可能面一覧取得・評価機能および AR での面表示機能

7-1-1. 検証目的

現在位置とカメラの向きを基に、撮影可能な建物の面を取得し、正確に AR 表示できるかを検証する。

7-1-2. KPI

表 7-1 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法
1	撮影可能面の AR 再現率	80%	撮影可能な面を網羅的に AR 表示することがシステムにとって重要であり、高い精度が求められるため	撮影可能な面のうち、AR 表示されている面の割合を計測
2	AR 表示面の適合率	80%	撮影不可能な面を AR 表示しないことがシステムにとって重要であり、高い精度が求められるため	AR 表示されている面のうち、撮影可能な面の割合を計測

7-1-3. 検証方法

- 検証方法
 1. 撮影可能な建物の面が AR 表示されている画面（建物検出画面）でスクリーンショットを撮影
 2. スクリーンショット内で撮影可能な面を目視でリストアップ（リスト A）
 3. スクリーンショット内で青色に AR 表示されている面をリストアップ（リスト B）
 4. リスト A と B で一致している面の数を求め、撮影可能面の AR 再現率（一致面数 ÷ リスト A の面数）を算出
 5. リスト A と B で一致している面の数を求め、AR 表示面の適合率（一致面数 ÷ リスト B の面数）を算出
 6. 建物の種類やカメラの画角を変えた複数のケースで、同様の手順により評価
 7. 撮影可能面の AR 再現率および AR 表示面の適合率の平均値を算出

- 撮影可能な面の条件

- 以下 3 つの条件を満たす建築の側面を撮影可能な面とする
 - ✧ カメラの画角に建物の面がすべて収まる（街路樹などの障害物によって面の一部が隠れている場合も含む）
 - ✧ カメラと建物の面の角度が適切な範囲内である（建物の面の正面から撮影する角度を 0 度とし、水平方向に約-60～60 度の範囲内）
 - ✧ カメラと建物の面の距離が適切な範囲内である（約 2～30m）



図 7-1 撮影可能な面（赤枠）のイメージ

- 検証ケース

建物の種類やカメラの画角を変えた 10 ケースで検証を行う。なお、撮影は 2025 年 3 月 5 日 16～18 時に実施した。

表 7-2 検証対象の建物まとめ

ケース	建物の種類	建物の名称	建物の住所
1	店舗	水戯庵	東京都中央区日本橋室町 2 丁目 5 - 9
2	住宅ビル	パンテオン日本橋三越前	東京都中央区日本橋室町 2 丁目 5 - 8
3	神社	福德神社	東京都中央区日本橋室町 2 丁目 4 - 1 4
4	店舗	室町東駐輪場	東京都中央区日本橋室町 2 丁目 5 - 1
5	複合ビル	江戸ビル	東京都中央区日本橋本町 2 丁目 3 - 4
6	店舗	水戯庵	東京都中央区日本橋室町 2 丁目 5 - 9
7	オフィスビル	住友不動産日本橋ビル	東京都中央区日本橋本町 1 丁目 5 - 4

8	店舗	米洲日本橋	東京都中央区日本橋本町 1 丁目 5 - 3
9	店舗	米洲日本橋	東京都中央区日本橋本町 1 丁目 5 - 3
10	複合ビル	ナカヤビル	東京都中央区日本橋室町 1 丁目 1 3 - 8

7-1-4. 検証結果

建物の種類やカメラの画角を変えた 10 ケースを対象に検証を行った結果、撮影可能面の AR 再現率の平均は 92%、AR 表示面の適合率の平均は 100%となり、いずれも目標値の 80%を上回る結果となった。

表 7-3 検証データまとめ

ケース	撮影可能な面の数 (リスト A)	AR 表示されている面の数 (リスト B)	リスト A と B の一致数	撮影可能面の AR 再現率	AR 表示面の 適合率
1	1	1	1	100%	100%
2	2	2	2	100%	100%
3	1	1	1	100%	100%
4	2	2	2	100%	100%
5	1	1	1	100%	100%
6	2	2	2	100%	100%
7	1	1	1	100%	100%
8	5	4	4	80%	100%
9	4	3	3	75%	100%
10	3	2	2	67%	100%
平均	-	-	-	92%	100%

● 撮影可能面の AR 再現率

再現率の平均値は 92%であり、目標値の 80%を上回った。これは、撮影可能な面の約 9 割が正しく青色に AR 表示されていることを示しており、店舗やビル、神社の本殿など建物の種類が異なる場合や、カメラの画角を変えた場合でも、安定した精度が維持されていることがわかる。

一部のケースでは撮影可能な面が正しく青色に AR 表示されないことを確認した。主な原因は、VPS で取得したカメラの位置や傾きに誤差が生じたことで、AR 面が実際とは異なる位置や角度で表示されたことが原因と考えられる。その結果、カメラと面の撮影角度が閾値を超え、面が撮影対象から除外されるケースが発生する。対策として、VPS の誤差補正の閾値や頻度を適切に調整することや、IMU を定期的にキャリブレーションすることが考えられる。



図 7-2 撮影可能な面（赤枠）が青色に AR 表示されていない例（ケース 10）

- AR 表示面の適合率

適合率の平均値は 100%であり、目標値の 80%を上回った。これは、全てのケースで撮影不可能な面が一つも AR 表示されていないことを示しており、異なる建物の面や撮影条件に対して安定した精度が確保されていることがわかる。

- 検証に用いたスクリーンショット一覧



図 7-3 ケース 1 のスクリーンショット



図 7-4 ケース 2 のスクリーンショット

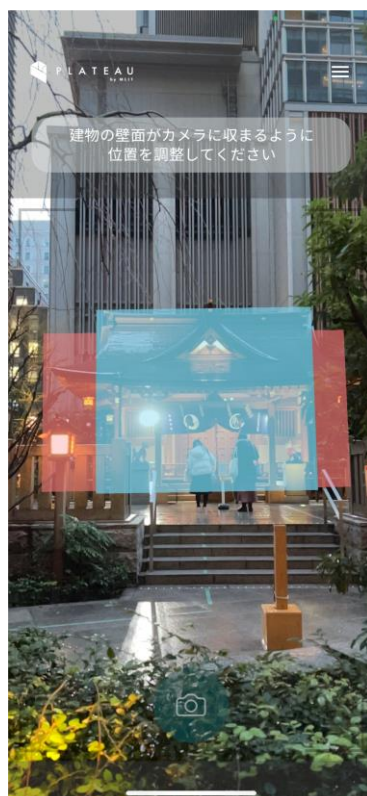


図 7-5 ケース 3 のスクリーンショット



図 7-6 ケース 4 のスクリーンショット

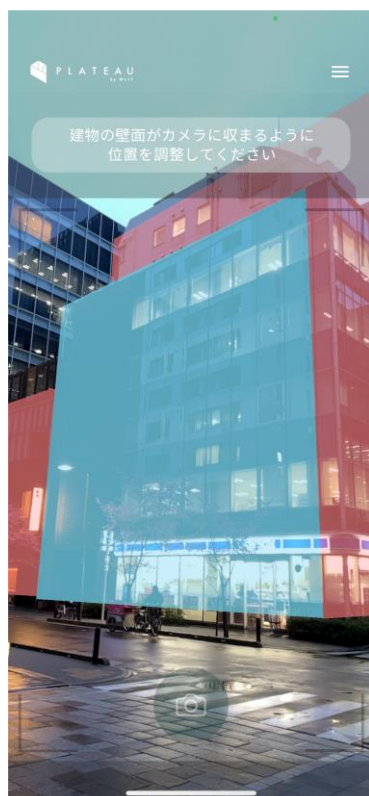


図 7-7 ケース 5 のスクリーンショット

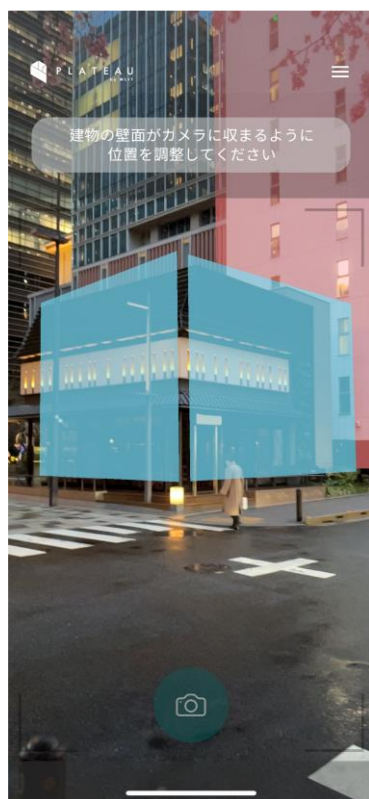


図 7-8 ケース 6 のスクリーンショット

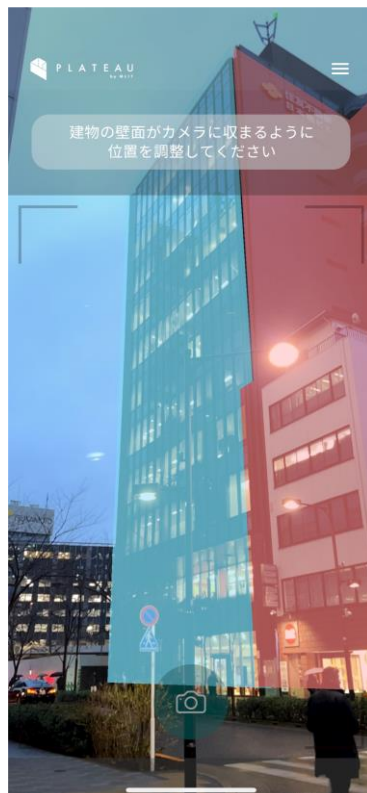


図 7-9 ケース 7 のスクリーンショット



図 7-10 ケース 8 のスクリーンショット



図 7-11 ケース 9 のスクリーンショット



図 7-12 ケース 10 のスクリーンショット

8. 開発するシステム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

8-2-1. 検証方法と検証シナリオ

表 8-1 非機能要件の一覧

カテゴリ	ID	項目	詳細
性能・拡張性	NR001	動作環境	「撮影画像によるテクスチャ生成ツール」はユーザーの標準的な動作環境に準拠した動作環境に対応する。

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
PLATEAU SNAP	動作環境	安定動作時間 30 分	2025 年 1～2 月	総合テストによる検証

8-2-2. 検証結果

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

9. 有用性検証

9-1. 検証目的

スマートフォンを活用したテクスチャ整備の実現性および有効性を検証する。

9-2. 検証方法

検証方法としては、被検者に対してデモンストレーションを取り入れたヒアリング・アンケートを実施する。
(ヒアリング・アンケートの項目については「9-4.ヒアリング・アンケートの詳細」にて記載)

- 実施会場
 - 日本橋レンタル会議室／中央区日本橋
- 機材：体験・デモ用に以下のスペックのスマートフォン／PC を用意する
- スマートフォン／iPhone13Pro
 - CPU 及び GPU： A15 Bionic
 - メモリ：6GB
 - ストレージ：256GB
 - OS：iOS 18.0.1 (2024/10/28 時点での最新バージョン)
 - 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
- PC／ROG Zephyrus G14 GA401QC
 - CPU：AMD Ryzen 7 5800HS
 - GPU：NVIDIA GeForce RTX 3050
 - メモリ：24GB
 - ストレージ (SSD)：512GB
 - OS：Windows11 Home
 - 通信環境：シナスタジア社で用意された社用 Wi-Fi

9-3. 被験者

表 9-1 被験者一覧

分類	具体名称	部署	担当業務	人数
事業者	デジタル庁	国民向けサービスグループ	プロジェクトオーナー	1 名
事業者	国土交通省	都市局	プロジェクトオーナー	1 名
事業者	アクセントチュア株式会社	ビジネスコンサルティング本部	プロジェクト統括	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	作業手順のデモ、説明	10 分
4	操作の体験	30 分
6	操作の確認	15 分
7	品質の確認	10 分
6	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	詳細
1	実験の目的を説明	● 本実証実験でアプローチする背景や課題の説明 ● 本実証実験で用いるシステムの提供価値 ● システムの全体像の説明
2	作業手順のデモ、説明	● システムの操作方法の説明
4	操作の体験	● 福德神社～三井本館～三越前福島ビル～東京建物日本橋ビルのルートで、本システムを用いてスマートフォンで建築物を撮影
6	操作の確認	● テクスチャ置換のデモ実施。作業工程を確認
7	品質の確認	● 既存の 3D 都市モデル（LOD2）の建築物テクスチャと、本システムで撮影したテクスチャを比較し、品質を確認
6	アンケート回答	● Google フォームでアンケート回答を依頼し、その場で回答（10 分）、送信を依頼

9-4-3. 検証項目と評価方法

本システムの性能・拡張性及びユーザビリティ評価を検証項目とし、それぞれ定量・定性的に評価する。

表 9-4 検証項目と評価方法

検証観点	No	検証項目	定量評価	定性評価
1) テクスチャ品質の評価	1	テクスチャ品質はよりリアルなテクスチャに改善されているか	選択肢は 1.以前より悪くなった 2.以前と変わらない 3.以前と比べてやや改善した 4.以前と比べて大幅に改善した	アンケートの各設問に自由記入欄を設定
2) ユーザビリティの評価	2	画面表示（UI）がわかりやすいか	選択肢は 1.わかりづらい	
	3	操作が直感的でわかりやすいか	2.ややわかりづらい 3.普通 4.わかりやすい 5.とてもわかりやすい	
	4	読みこみ時間が遅いなどの不自由なく操作できたか	選択肢は 1.使いづらかった 2.やや使いづらかった 3.普通 4.使いやすかった 5.とても使いやすかった	

9-4-4. 実証実験の様子

被験者が AR 表示と本ツールの UI/UX を確認している様子。



図 9-1 AR 表示の確認



図 9-2 AR 表示画面

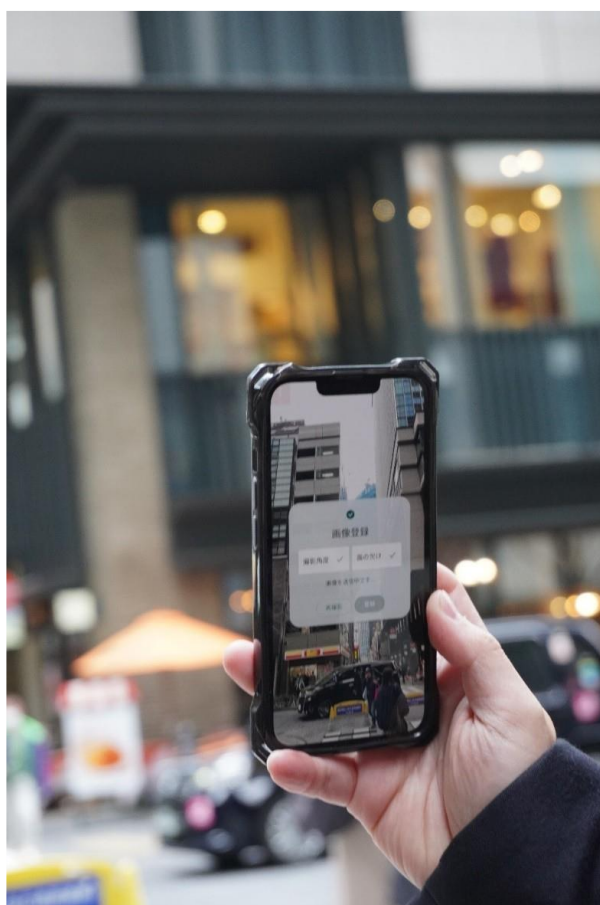


図 9-3 登録画面の確認

9-5. 検証結果

本システム「PLATEAU SNAP」の有用性を検証するため、日本橋において実機 3 台を用いた実証実験を実施し、参加者 3 名からのフィードバックを収集した。特に、3D 都市モデルのテクスチャ品質の向上、およびユーザービリティの観点から評価を行った。

1) テクスチャ品質の評価

本システムを使用して撮影・登録されたテクスチャを 3D 都市モデルに適用し、従来の CityGML のテクスチャと比較する評価を実施した。評価は、どちらのテクスチャが CityGML のものか、本システムで生成されたものを明示しないブラインドテストの形式で行った。その結果、対象にした 8 つの建物において、7 つの建物で本システムのテクスチャがより高品質であると評価された。特に、建物のディテールや質感の表現力が向上している点が高く評価され、本システムが 3D 都市モデルの視覚的品質を向上させる有効な手段であることが示された。

I. 高さの低い建物のテクスチャ比較

高さの低い建物のテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は 3 名だった。



図 9-4 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

II. 店舗外観のテクスチャ比較

店舗外観のテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は 3 名だった。



図 9-5 (左) 本システムで生成したテクスチャ_A／(右) CityGML のテクスチャ (LOD2) _B

III. 神社仏閣のテクスチャ比較

神社仏閣のテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は3名だった。



図 9-6 (左) 本システムで生成したテクスチャ_A／(右) CityGML のテクスチャ (LOD2) _B

IV. 複合ビルのテクスチャ比較

複合ビルのテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は3名だった。



図 9-7 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

V. オフィスビルのテクスチャ比較

オフィスビルのテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は2名、「どちらも変わらない」と評価した参加者が1名だった。

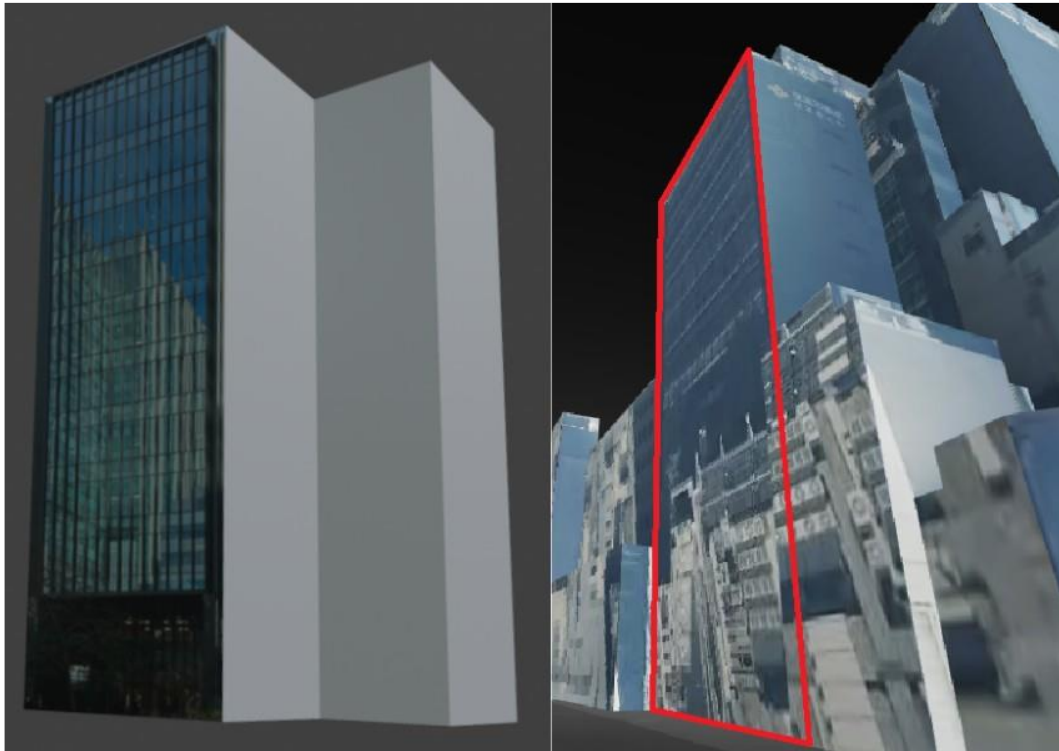


図 9-8 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

VI. オフィスビル（中層階）のテクスチャ比較

オフィスビル（中層階）のテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は 3 名だった。



図 9-9 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

VII. 大型複合ビルのテクスチャ比較

大型複合ビルのテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は 3 名だった。

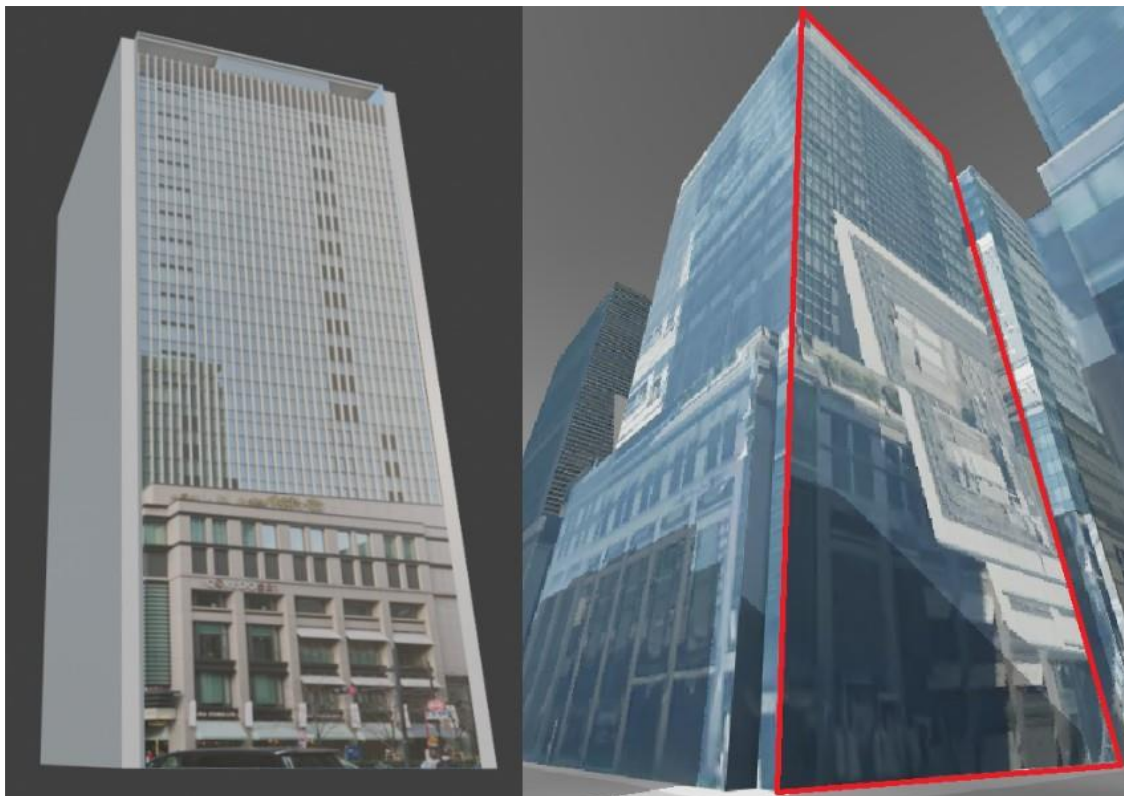


図 9-10 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

VIII. 幅の大きい建物のテクスチャ比較

幅の大きい建物のテクスチャ比較において、「本システムで生成したテクスチャ_A」の品質が高いと評価した参加者は 3 名だった。

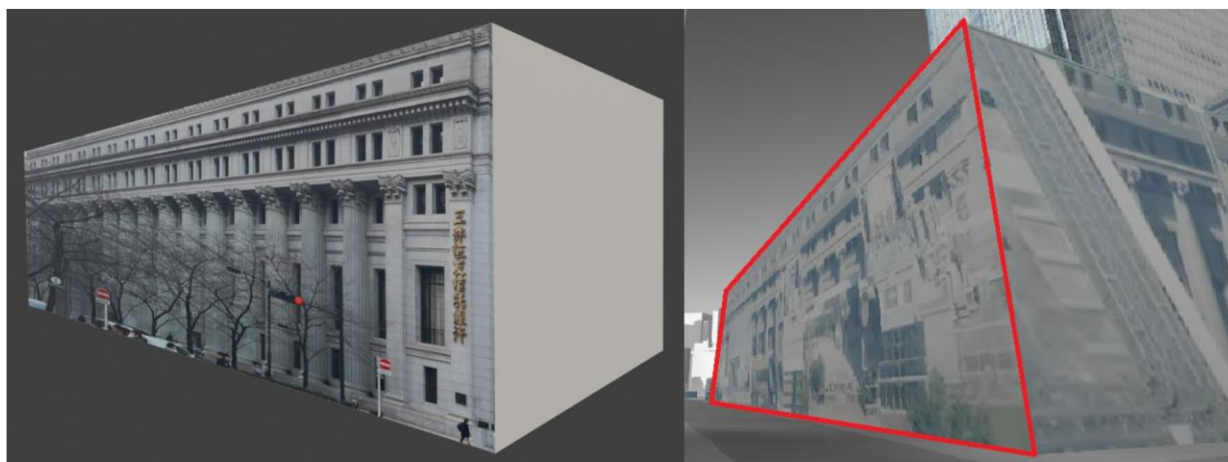


図 9-11 （左）本システムで生成したテクスチャ_A／（右）CityGML のテクスチャ（LOD2）_B

2) 撮影角度によるテクスチャ品質の評価

本システムを用いた撮影時の角度によるテクスチャ品質の変化についても評価を行った。2 つの建物を対象に、正面から撮影した場合と斜めから撮影した場合のテクスチャを比較するブラインドテストを実施した。

その結果、1つの建物では全員が正面から撮影したテクスチャの方が高品質であると評価し、もう1つの建物では「どちらも変わらない」との回答が1名、「正面から撮影したものが良い」との回答が2名であった。これにより、正面からの撮影がより高品質なテクスチャを得る上で有利であることが確認されたものの、斜めからの撮影でも一定の品質を維持しながらテクスチャを生成できることが示された。

I. 撮影角度の違いによるオフィスビルのテクスチャ比較

オフィスビルのテクスチャ比較において、「本システムで正面から撮影したテクスチャの品質が高い」と回答した参加者が2名、「どちらも変わらない」と回答した参加者が1名だった。

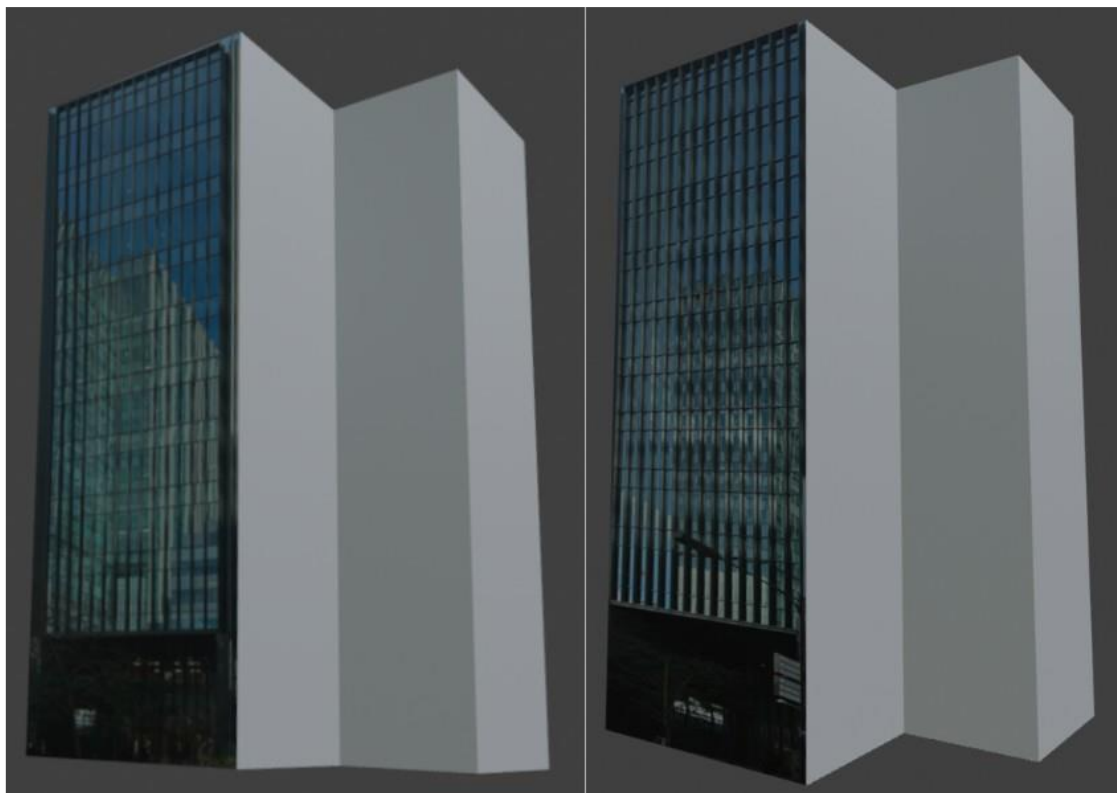


図 9-12 （左）本システムで正面から撮影し生成したテクスチャ_A／（右）本システムで斜めから撮影し生成したテクスチャ_B

II. 撮影角度の違いによるオフィスビル（中層階）のテクスチャ比較

オフィスビルのテクスチャ比較において、「本システムで正面から撮影したテクスチャの品質が高い」と回答した参加者が3名だった。



図 9-13 本システムで正面から撮影し生成したテクスチャ_A／（右）本システムで斜めから撮影し生成したテクスチャ_B

2) ユーザビリティの評価

ユーザビリティに関しては、UI のわかりやすさについてすべての回答者が「とてもわかりやすい」と評価し、視認性の高さが強く支持された。また、操作の直感性については、3 名中 2 名が「とてもわかりやすい」、1 名が「わかりやすい」と回答しており、全体的に高い評価を得た。これらの結果から、本システムは専門知識を必要とせず、誰でも直感的に操作できる設計となっていることが確認された。一方で、さらなる改善の余地として、特定の操作フローの簡略化や補助機能の追加が望まれる可能性がある。

3) 動作環境の評価

動作環境については、読み込み時間の遅延や操作上の不自由さに関する評価を実施した。その結果、操作がしづらいつと感じた参加者はいなかったものの、3 名中 1 名が「普通」と回答し、動作の快適さに関しては一定の改善の余地があることが示唆された。一方で、2 名の参加者は「操作しやすかった」と評価しており、多くのユーザーが問題なく利用できる環境であったことが確認された。今後は、より快適な操作性を実現するため、システムの最適化や処理速度の向上が求められる可能性がある。

4) 総合的な評価

テクスチャ品質の総合的な評価として、これまでの比較結果を踏まえたアンケートを実施した。この際、どちらのテクスチャが本システムで生成されたものかを明示した上で回答を得た。その結果、全ての回答者が「本システムによってテクスチャの品質が向上した」と回答し、本システムの有用性が明確に示されたと考えられる。

表 9-14 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	テクスチャ品質はよりリアルなテクスチャに改善されているか	● 樹木等の写り込みが気になるので、AI の画像編集機能等で削除できると良い ● 街路樹や自動車の映り込みや人の写り込みが気になる ● 対象建物前にある障害物（車や人等）を取り除ける機能が付くとよりリアルなものになると思う
2	画面表示（UI）がわかりやすいか	● 撮影可能面が少しオフセットされて映るため、どこを撮っているかわからないときがあった ● 初回のみ赤青緑の違いが凡例として表示されるとよい ● 画面上のシャッターボタンはもう少し大きくてもいいかと思う
3	操作が直感的でわかりやすいか	● 撮影可能面を選択し撮影するプロセスに沿って必要なボタンが適切な順序および位置で表示されており使いやすい ● 現在地の取得に読み込みに時間がかかる場合に改善方法のガイドが表示されるとよい（画面を左右に振った方がよいとか、縦向きで撮影した方がよい） ● 本日の実証では、画面上で面が捉えられない場合は右上のアイコンを押してリセットしていましたが、このリセットのアイコンをもう少し分かりやすくなったらいいと思う
4	読みこみ時間が遅いなどの不自由なく操作できたか	● 場所によりサーバーとの接続が切れるケースがあったようなので、動作安定性は改善ポイントかと思った ● たまに表示可能面がうまく表示されず、撮影まで時間がかかることがあった ● 現在地の取得さえうまくいけば読み込みは速いと思った

10. 成果と課題

10-1. 本実証で得られた成果

表 10-1 本実証で得られた成果

大項目	小項目	本実証で得られた成果
機能	VPS 機能及び AR での面表示機能	● 建物の頂点座標情報をもとにメッシュを作成し、撮影可能な面を建物の実際の面に沿って正確に AR 表示できた
	撮影可能面の取得及び評価機能	● カメラの視野内に頂点がすべて収まり、他の面に遮蔽されていない建物の面を正確に絞り込むことができた
	テクスチャ品質	● 従来に比べて大幅に向上した ● 品質の低下が予想された遠距離からの撮影や斜めの角度からの撮影でも、高品質のテクスチャを生成できた

10-2. 実証実験で得られた課題と対応策

表 10-2 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
機能	VPS 機能及び AR での面表示機能	● AR 表示された面と実際の建物の面に 1～3m 程度のズレが発生する場合がある ● AR 表示のズレが目視では判別しづらい場合がある	● VPS の誤差補正の閾値や頻度を適切に調整することや、IMU を定期的にキャリブレーションすることが考えられる ● AR 表示のズレをシステムで検出し、ユーザーに分かりやすく示す
	テクスチャ品質	● 街路樹や通行人などの不要なものが映り込む場合が多い	● 画像処理で不要な要素を除去し、欠損部分を自然な建物のテクスチャとして補完する ● 同じ面を異なる時間帯や視点から撮影した複数の画像を組み合わせ、不要な要素を排除したテクスチャを生成する
UI・UX	利用ガイド	● 直感的かつ最小限の操作で活用できるよう設計したものの、操作に不慣れなユーザーでも容易に扱えるよう、利用ガイドの改善が求められる ● 特に、AR 表示されるメッシュの色ごとの違いや、適切なカメラの動かし方についての説明が不足しているため、視覚的にわかりやすいガイドの追加が重要である	● 利用ガイド画面で、メッシュの色ごとの違いや、建物検出画面に遷移する際は建物にカメラを向けてなるべく動かさないことを、文字だけでなく画像や動画を用いて説明する ● カメラの急な動きを検知し、ユーザーに対してカメラをゆっくり動かすよう促すガイドを提供する

10-3. 今後の展望

今回の実証実験では、スマートフォンを活用したテクスチャ生成技術の有用性と社会実装の可能性を検証した。従来の CityGML のテクスチャと比較し、本ツールによって生成されたテクスチャの品質向上が確認されたことは、大きな成果の一つである。また、ユーザビリティに関しても、直感的な操作性が評価され、一般ユーザーによる 3D 都市モデルの更新が容易になる可能性が示された。

さらに、AR 表示の精度検証では、撮影可能面の再現率が平均 92%、AR 表示面の適合率が平均 100%と、目標値の 80%を大きく上回る結果となった。これは、異なる建物の面や撮影条件に対して、撮影可能な面の約 9 割が正しく AR 表示され、撮影不可能な面が一切 AR 表示されないことをそれぞれ示している。一方、一部のケースでは撮影可能面の再現率が 80%にとどまる例もあり、VPS によるカメラの姿勢推定の誤差が影響を与えている可能性が示唆された。

今後の展望として、テクスチャの生成・貼り付け作業をシステムに統合し、手動による調整を大幅に削減する。さらに、街路樹や通行人などの不要な映り込みを除去し、自然な仕上がりに補正することで、より高品質なテクスチャ生成を実現する。AR 表示については、IMU の定期的なキャリブレーションを行い、VPS の誤差補正の頻度やタイミングを最適化することで、さらなる精度向上を目指す。

将来的には、自治体や民間事業者がテクスチャデータの撮影を安価に発注し、一般利用者が受注・撮影を行うビジネスモデルの構築も視野に入れる。また、自治体や民間事業者自身が必要なテクスチャデータを独力かつ低コストに撮影・整備出来る仕組みを確立する。

一方で、一般利用者が撮影したデータの品質や信頼性にはばらつきが生じやすくなるという課題が浮上した。これに対しては、テクスチャ品質検査ツールの開発によって解決が図られる可能性がある。このツールは、3D 都市モデルの標準製品仕様書に新たに規定されるテクスチャ品質基準をもとに、撮影・アップロードされたテクスチャデータの画質や推定撮影位置に紐づく対象建物 ID や面 ID の適切性等を半自動又は全自動で検査し、検査基準を満たすデータのみを 3D 都市モデルのデータサーバに蓄積・配信する仕組みを想定する。

また、現在の PLATEAU CMS (Contents Management System) に相当するデータ管理システム (CMS) の導入により、自治体や企業などのテクスチャデータ利用者が一般利用者にテクスチャの撮影を依頼できる仕組みを構築する。さらに、アップロードされたデータの検査・管理・編集・ダウンロードを行うことを可能とすることで、品質の担保だけでなく、データの流通や活用の効率化も進むことが期待される。

11. 用語集

A) アルファベット順

※下記は記載例

表 11-1 用語集（アルファベット順）

No.	用語	説明
1	デジタルツイン	現実世界の物理空間をデジタル上に再現する技術
2	テクスチャ	3D 都市モデルの視覚的なリアリティを向上させるために、建物や地物の表面に適用される画像データ

B) 五十音順

表 11-2 用語集（五十音順）

No.	用語	説明
1	AWS（Amazon Web Services）	Amazon が提供するクラウドサービスの総称。計算、ストレージ、データベース、機械学習など多様なサービスを提供し、スケーラブルなアプリケーション開発を支援する
2	Google Geospatial API	:Google の位置情報技術を活用し、現実世界とデジタルデータを統合する API 群。AR やマップベースのアプリ開発に使用される
3	IMU（Inertial Measurement Unit）	加速度計やジャイロ스코プを組み合わせたセンサー。デバイスの位置や姿勢を高精度に測定し、ナビゲーションやモーションキャプチャに利用される
4	MMS（Mobile Mapping System）	走行しながら周囲の地理情報を計測する移動型計測システム。LiDAR やカメラ、GNSS を活用し、高精度な 3D 地図作成に用いられる
5	VPS（Visual Positioning System）	画像解析を利用した位置推定技術。GPS が使えない環境でも、カメラ映像をもとに正確な位置を特定する

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
技術検証レポート

発 行：2025 年 3 月

委託者：デジタル庁

受託者：株式会社シナスタジア