

TsurugiDB ホワイトペーパー
応用事例

株式会社パスコ
2023/7

目次

1. アプリケーション概要	3
1.1. ユースケース概略	3
1.2. システム概要.....	3
1.2.1. 航空写真からの 3D 都市モデル作成	3
1.2.2. 3D モデル間の差分演算	3
1.2.3. 確認・閲覧システム	4
1.3. システムの特徴.....	4
1.3.1. 大容量データの高頻度更新・高可用性	4
1.3.2. Tsurugi と PostgreSQL の連携	5
1.3.3. 高効率・高速な 3D モデル作成	5
1.4. システムアーキテクチャ	6
1.4.1. Tsurugi	6
1.4.2. PostgreSQL	7
1.4.3. 分散処理システム (ParallelRunner)	7
1.4.4. OpenMVG、OpenMVS	8
2. サーバスペック	11
2.1. Tsurugi を用いたアプリケーション実装.....	11
2.2. 3D 都市モデル作成.....	11
3. ベンチマーキング.....	12
3.1. データベースの大容量書き込み	12
3.2. 3D 都市モデル作成.....	12
4. 導入の効果と提言	14
4.1. システム展開とスケール変更の柔軟性	14
4.2. 地理的に遠隔配置されたサーバ間での負荷分散.....	14
4.3. 空間検索の高速化と分散実行	14
4.4. 空間検索とその他検索の連携.....	14
5. 謝辞	16

1. アプリケーション概要

1.1. ユースケース概略

本プロトタイプアプリケーションは、災害発生時に被災地の航空写真から 3D 都市モデルを作成し、広く利用及び閲覧に供することを目的とする。

航空写真からの 3D モデル作成は、人工衛星やドローン等の観測手段に比較して、単位面積あたりのデータ取得効率と高解像度情報取得を同時に実現でき、特に被害が広域に生じる大規模地震等での情報収集に適した手法である。

作成した 3D 都市モデルは、現地状況の概要把握、差分抽出による被災箇所推定、モデル上の簡易計測に利用でき、各機能はインターネット経由で汎用ブラウザから操作が可能である。

1.2. システム概要

本プロトタイプアプリケーションは次の 3 つのシステムから構成される。

1.2.1. 航空写真からの 3D 都市モデル作成

航空オブリーク写真から、分散並列処理で 3D 都市モデルを作成するシステムである。

- 3D 都市モデル作成対象地域の航空オブリーク画像、カメラ位置姿勢・機器情報を入力データとして使用する
- 航空写真からの 3D 都市モデル作成には、オープンソースの SfM・MVS ライブラリの OpenMVG および OpenMVS を利用する
- 分散並列処理の単位は、国土基本図の図郭（レベル 250）、実行管理は Asakusa をベースとした分散処理システム（ParallelRunner）を利用する
- 生成された 3D モデルが一定の品質条件を満たさない場合、成果を破棄して再処理する。生成された 3D モデルが一定の品質条件を満たす場合、基盤地図（家屋形状、標高）による位置合わせ、インターネット閲覧用の軽量化を実施する

1.2.2. 3D モデル間の差分演算

2 時点の 3D 都市モデル間で演算を行い、差分を抽出するシステムである（図 1）。

- 災害の前後の差分は、地形や家屋形状の変化といった被災箇所の推定に役立つ
- 災害発生後の時系列での差分は、2 次災害や復旧活動の進捗を把握できる



図1 建物建設前後の差分（左：建設前、中：完成後、右：差分）

1.2.3. 確認・閲覧システム

汎用インターネットブラウザを用いて次の機能を利用できるシステムである（図2）。

- 3D 都市モデル作成の分散並列処理の進捗を確認する
- 作成した 3D 都市モデルの形状を目視確認し、再処理、公開・非公開を設定する
- 作成した 3D 都市モデルを用いて簡易計測する
- 背景地図は地理院タイルを使用、任意の GIS データ（GeoJSON 形式）を重ね表示可能

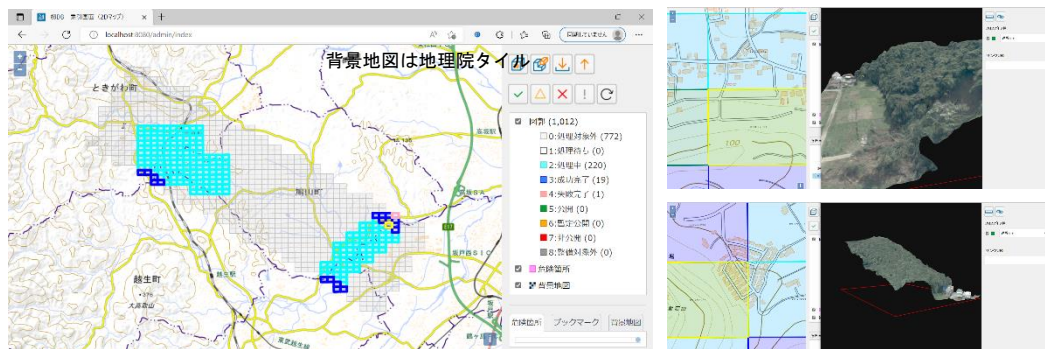


図2 システム画面（左：進捗確認、右：3D モデル閲覧）

1.3. システムの特徴

1.3.1. 大容量データの高頻度更新・高可用性

従来のデータベース技術では 3D 都市モデルを一定領域ごとに 1 レコードとして格納していたが、本プロトタイプアプリケーションでは、より細かい単位（個々の点ごとに 1 レコード）でデータベースに格納する。

また、一般に大規模データベースを用いたシステムでは、データ更新中はシステム利用が制限される場合が多い（夜間バッチ等）が、本プロトタイプアプリケーションでは、データ更新中でも最新データの読み出しが可能である。

上記の特徴により、災害対応において次の効果がある。

- 時々刻々と収集される被災地の情報を随時格納・更新しながら利用もできる
- 複数時点のデータの蓄積・比較により、災害の詳細な様態把握が可能になる
- 広域の災害に対応可能な容量の 3D 都市モデルを管理できる
- 任意の単位で 3D 都市モデルの更新・利用が可能になる

1.3.2. Tsurugi と PostgreSQL の連携

災害時利用を念頭に置いた 3D 都市モデル整備には、前項で述べた大容量データの高頻度更新・高可用性は必須である一方で、アプリケーション開発の自由度やデータ拡張性も確保する必要がある。そこで本プロトタイプアプリケーション開発においては、Tsurugi と PostgreSQL の 2 つの DBMS を連携して利用した。

上記により、災害対応アプリケーションの開発において次の効果がある。

- Tsurugi の特長を活かしつつ、PostgreSQL の幅広い対応データ型を活用できる
- PostgreSQL ベースで開発した差分演算機能（空間インデックス・ストアードプロシージャ・最適化済 SQL）を利用できる

1.3.3. 高効率・高速な 3D モデル作成

1.3.3.1. 分散並列化

従来手法においては 3D 都市モデル化対象の全ての航空写真データを一括して処理するが、処理対象地域を分割することで分散並列実行を可能にする。生成される 3D 都市モデルは対象地域毎に異なる（他の地域の影響を受けない）ので、対象地域をブロック状に分割して分散処理が可能である。処理の分割単位は国土基本図図郭（レベル 250）を標準とする。

分散並列化により、災害時の 3D 都市モデル作成において次の効果がある。

- ノードマシンの数を増やすことで、災害の規模に応じた処理能力をスケールできる
- 全ての撮影の完了を待つ必要がなく、画像が伝送できた区画から順に処理できる
- 優先して状況を把握したい一部の地域を優先して処理できる
- 部分的な撮影不良（雲入り、機体揺れなど）が他の地域に影響しない

但し、処理分割成果の利用には、3D 都市モデルの座標修正、分割単位間の境界トリミングが必要になる。

1.3.3.2. ランダム性への対応

航空写真からの 3D 都市モデル作成においては、ランダム性要因が含まれる。要因と対処について次に示す。

- OpenMVG OpenMVS の内部にランダム関数を使用されていることにより、入力データが同じであっても成果形状・所要時間・処理の成否が変動する。分散並列処理を投機的に実行することで対処する
- 3D 都市モデル化対象の地形や天候条件により、所要時間や成功率が変動する。様々な地形・季節のサンプルを用いたテストを行い、安定した成果を得られる処理パラメータを設定する

1.3.3.3. 入力データ（航空写真画像）の最適化

入力データ（航空写真画像）の読み込みデータ量（枚数・領域）を絞ることで、メモリ使用量と I/O 時間を削減する。具体的には次に示す方法を用いる。

- ・ 予め設定する 3D 都市モデル化対象範囲と、実際に撮影した航空写真の各画像が映し出す範囲、及び予め設定された処理分割単位の設定を基に、処理の分割単位ごとに必要な画像ファイルのリストを作成する。
- ・ 処理分割単位ごとに必要な画像ファイルを選択し、さらに当該画像ファイル内の必要領域のみを切り取って処理ノードに転送する。この際、カメラモデルを考慮して画像を投影変換する。

また、本プロトタイプアプリケーションの開発範囲内のみならず、開発範囲外の伝送コスト（航空機搭載 HDD から処理基盤まで）も最適化する。既存の業務用設備を活用したために必ずしもユースケースには最適化しないが可能な範囲で実施する。

- 航空機搭載 HDD からのデータ伝送は着陸毎とする。本来は逐次伝送が望ましく、災害対応ヘリ等で実用化されている対地直接伝送や衛星経由伝送での逐次伝送が望ましい。また仮に商用 5G 回線の送信時最大理論速度（1.1Gbps）であれば撮影ショット間隔にもよるが概ねリアルタイム伝送が可能である（実際には地上側設備や通信帯域専有可否などの課題がある）
- 航空写真の画像ファイルは圧縮形式（JPG 形式）に変換する

1.4. システムアーキテクチャ

1.4.1. Tsurugi

アプリケーションの主たる DBMS として Tsurugi を利用する。Tsurugi は大容量の高頻度更新、処理負荷分散とスケーラブルな展開が可能であり、データ容量が大きく必要能力の事前予測が困難な災害対应用途に好適である。

- 3D 都市モデル及びアプリケーションのシステムテーブルは原則として Tsurugi に格納する

- 3D 都市モデルの書き込みは Tsurugi が提供する 2 方式を使い分ける。分散並列処理成果の逐次書き込みでは Postgres レイヤーを、過去アーカイブ利用等を想定した一括書き込みでは API を利用する

1.4.2. PostgreSQL

1.4.2.1. Tsurugi と PostgreSQL 間の INNER JOIN

3D 都市モデルの骨格（頂点データ）は Tsurugi に格納し、テクスチャ画像データはバイナリ列データ型 (bytea) で PostgreSQL に格納する（図 3）。Tsurugi のフロントエンド連携機能 (FDW) により、Tsurugi と PostgreSQL に分かれて格納されたテーブル間で INNER JOIN 等が利用できる。

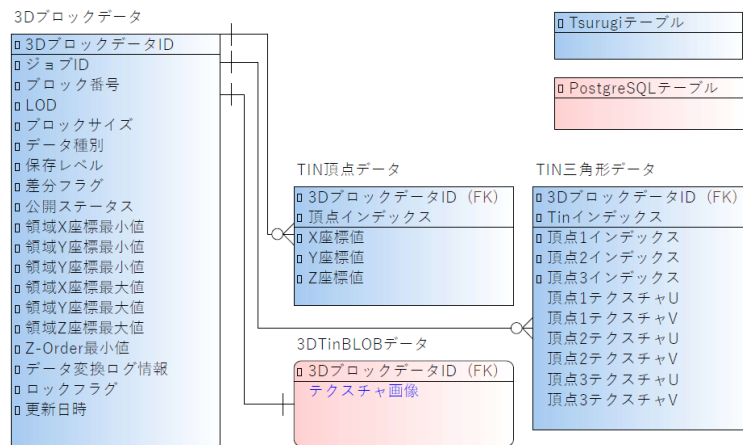


図 3 3D 都市モデルテーブルと格納先（青：Tsurugi、赤：PostgreSQL）

1.4.2.2. 差分演算

差分演算機能は PostgreSQL を利用する。

- 2 時点の 3D 都市モデルをそれぞれ点群に分解し、2 つの点群間で任意の距離よりも離れている点群を抽出する
- 空間充填曲線による空間索引の代替を使用する
- ストアドプロシージャとして実装する
- 分散並列処理による高速化を実施する

1.4.3. 分散処理システム (ParallelRunner)

分散処理の実行管理にはオープンソースの分散処理環境用フレームワークの Asakusa

Framework をベースとする分散処理システム ParallelRunner を利用する。3D 都市モデルの素材となる航空写真および中間生成ファイルは分散ファイルシステムである HDFS を利用してストレージ I/O に対する負荷を分散する。

分散処理を用いて実行した主な処理は下記の通り。

- 航空写真画像を選択・部分抽出・東映変換してノードに配布する
- 機能呼出し（OpenMVG・OpenMVS のコマンド、位置合わせ、軽量化）
- 投機的実行、任意のタイムアウト時間・リトライ回数の設定
- 成果（3D 都市モデル）が不十分な場合は再処理の待ち行列へ戻

1.4.4. OpenMVG、OpenMVS

航空写真からの 3D 都市モデル作成には、オープンソースの SfM・MVS ライブラリの OpenMVG および OpenMVS を利用する。

1.4.4.1. パラメータ設定

パラメータ設定は、目的に応じて複数のパラメータセットを使い分ける。

例として、処理の速さに優れたパラメータセット A・速度では劣るが 3 次元形状の再現性に優れた同 B を表 1 に、両設定による 3D 都市モデル形状の例を図 4 に示す。

表 1 パラメータ設定の例

使用コマンド		パラメータ設定	
		速度優先 (A)	3次元形状の再現優先 (B)
1	openMVG_main_SfMInit_ImageListing	-P -c 1	-P -c 1 -g 0
2	openMVG_main_ComputeFeatures	-p ULTRA -m SIFT -n 64 -u 1	-p ULTRA -m SIFT -n 64 -u 0
3	openMVG_main_ComputeMatches	-g e -r 0.6	
4	openMVG_main_GlobalSfM	-P -f NONE	
5	DensifyPointCloud	--fusion-mode -1	
6	DensifyPointCloud	--fusion-mode -2	
7	ReconstructMesh	--quality-factor=2 --min-point-distance=1.5 --decimate=0.3	
8	RefineMesh	--decimate=1 --max-face-area=32	
9	TextureMesh	--cost-smoothness-ratio=1 --patch-packing-heuristic=0	



図4 パラメータ設定と形状例（左：速度優先(A)、右：形状優先(B)）

1.4.4.2. 多様なサンプルデータ

OpenMVG 及び OpenMVS が出力する 3D 都市モデルの品質は、入力する航空写真データの内容にも左右される。そこでいくつかの状況を想定し、それぞれについてサンプルデータを取得して撮影仕様と天候等の要素が及ぼす影響を検討した。

取得対象の選定において想定した状況は下記の通りである。

- 災害時の被害把握（土砂崩れ・家屋倒壊・倒木、それらに伴う道路閉塞等）
- 特徴的な都市形状（山間部の道路、密集市街地、大規模河川と流域、高層ビル）
- 災害が発生した地域（旧山古志村、阿武隈川流域、房総半島、安倍川流域、熱海）
- 災害直後の撮影手配～システム動作までの実証実験（2022年8月埼玉県大雨）

サンプルデータの一覧を表 2 に示す。撮影機材は、直下視および斜方視の画像を取得できるオブリークカメラの Leica 社 RCD30 Oblique を航空機に搭載して使用した。

表2 サンプルデータ一覧

年度	撮影地区	解像度(m)	コース数	写真枚数	図郭数(1/250)	面積(km ²)	備考
2018	長岡	0.12	25	10585	3856	115.68	
2019	阿武隈川	0.12	20	11215	5660	169.80	
2019	館山	0.07	39	10575	928	27.84	南北15、東西24
2019	南房総	0.07	34	7495	748	22.44	南北13、東西21
2019	東京2020	0.13	30	8610	4348	130.44	2回撮影
2019	渋谷	0.05	11	1155	64	1.92	2回撮影
2019	長岡	0.12	25	10585	3856	115.68	2回撮影
2020	安倍川	0.12	13	2735	432	12.96	
2020	朝倉	0.08	10	1530	140	4.20	
2020	上阿多古	0.07	10	1275	80	2.40	
2020	山古志	0.12	19	5620	1012	30.36	
2020	伊豆下田	0.12	12	2920	484	14.52	
2020	東京	0.18	11	2585	2120	63.60	
2020	南房総	0.07	13	3440	748	22.44	
2020	館山	0.07	9	2840	500	15.00	
2020	阿武隈川	0.12	10	5530	2360	70.80	
2020	渋谷	0.05	11	1155	64	1.92	
2020	長岡	0.12	25	10585	3856	115.68	
2020	京都	0.07	28	10965	1700	51.00	2回撮影
2020	大阪	0.15	13	2975	1820	54.60	2回撮影
2021	渋谷	0.12	10	750	64	1.92	
2021	京都	0.07	28	10965	1700	51.00	
2021	山古志	0.12	19	5620	1012	30.36	
2021	熱海	0.11	9	1080	52	1.56	2回撮影
2022	埼玉	0.10	10	4010	1012	30.36	実証実験
2022	横浜	0.07	15	2625	192	5.76	
2022	神戸	0.07	21	5190	1172	35.16	
2022	長岡	0.12	25	10585	3856	115.68	

2. サーバスペック

本プロトタイプアプリケーション開発では、Tsurugi を用いたアプリケーション実装（1.3.1、1.3.2 に対応）と 3D 都市モデル作成の分散処理（1.3.3 に対応）は別環境で実施した。プロジェクトの共有環境を利用したので本プロトタイプアプリケーションのためのスペック検討（サーバとネットワーク帯域のサイジング検討、ネットワーク構成の検討）は行っていない。

2.1. Tsurugi を用いたアプリケーション実装

以下のスペックのサーバ 1 台で構成した。

- CPU: Xeon(R) Platinum 8276(28C/56T, 2.2GHz) × 4
- メモリ: 512GB DRAM(32GB 2933 RDIMM × 4)
- Disk(HDD): 13TByte
- Disk(SSD): 440GB

2.2. 3D 都市モデル作成

利用したサーバ環境は下記の通り。サーバ 56 台（クライアントマシン 1 台、ノードマシン 55 台）で構成した。

- CPU: Xeon Gold 6130(16C/32T, 2.1GHz) × 2
- メモリ: DDR4-2666 LRDIMM 64GiB × 24(合計 1.536TiB)
- Disk(HDD): 10TB SATA 7,200rpm × 7
- Disk(SSD): 250GB SATA × 2 (RAID1)

3. ベンチマーキング

3.1. データベースの大容量書き込み

本項では Tsurugi のバルクロード API のベンチマークについて述べる。

分散並列処理によって生成される 3D 都市モデル 1 単位 (330,452 レコード) を書き込み用のサンプルとして parquet ファイル化 (10,581,067byte、軽量化加工済) し、これを 1,000 単位に複製して php スクリプトで Tsurugi のバルクロード機能 (parquet-loader) を連続して実行した。

書き込み先テーブルの仕様を表 3、処理時間とメモリ使用量のグラフを図 5、図 6 に示す。

表 3 書き込み先テーブル仕様

No.	項目和名	項目名	データ型	PK
1	3DTin データ ID	tin_id	int	1
2	頂点インデックス	vertex_idx	int	2
3	X 座標値	x	float8	
4	Y 座標値	y	float8	
5	Z 座標値	z	float8	

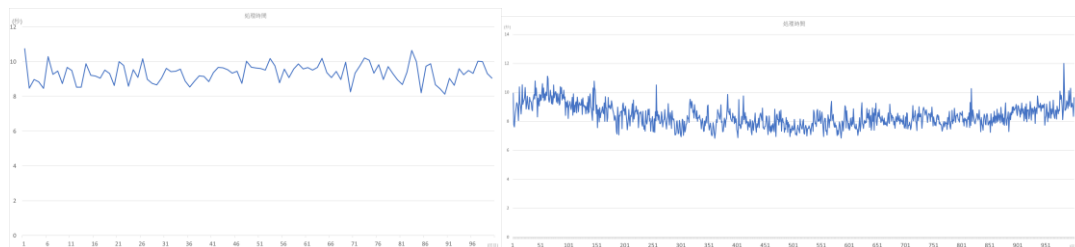


図 5 処理時間グラフ (左: 100 件、右: 1,000 件)

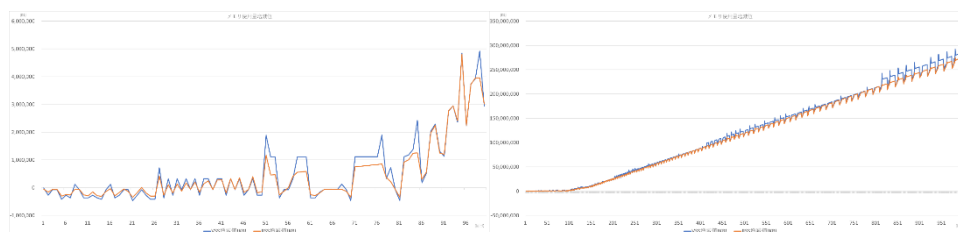


図 6 メモリ使用量増減値グラフ (左: 100 件、右: 1,000 件)

3.2. 3D 都市モデル作成

本項では 3D 都市モデル作成のベンチマークについて述べる。

分割処理の単位あたりの実行時間は、航空写真の直下視方向地上解像度が 10~12cm 程

度の郊外市街地では平均 2～5 時間、同 7cm の密集市街地では 7～9 時間程度であった。

4. 導入の効果と提言

従来技術では、3D 都市モデルのように多数の座標から成る GIS データは一定の面積等で区切ったバイナリ形式での格納が主であり、不必要な I/O や待ち時間等ファイルベース管理と同様の課題があった。本プロトタイプアプリケーションの開発では、3D 都市モデルデータを個々の点に分解し、分解した点ごとに 1 レコードとしてデータベースに格納した上で想定ユースケースを実現可能なパフォーマンスを確認できた。

Tsurugi は、本来は現実空間をよりリアルタイムに反映すべき GIS データをあるべき姿でメンテナンス・利活用できるデータベース技術であると言える。よって本プロトタイプアプリケーションで対象とした災害時のみならず、より一般的な状況下のユースケースへの適用を展望し、GIS 分野においては次のような能力の充実を期待する。

4.1. システム展開とスケール変更の柔軟性

災害時を想定する場合、発災直後に最大のパフォーマンスが必要となると考えられる。必要処理能力やタイミングを事前に予測することは困難であるため、迅速なシステムの展開と、必要に応じた柔軟なスケールが望ましい。

4.2. 地理的に遠隔配置されたサーバ間での負荷分散

本プロトタイプアプリケーションの実用上の課題が、航空機から処理環境までの入力データ（航空写真）を伝送時間である。高能力設備の常時・複数維持は困難であり、遠隔配置されたサーバ間で負荷分散できれば対応可能な災害規模・地域が広がる。

4.3. 空間検索の高速化と分散実行

本プロトタイプアプリケーションでは 3D 都市モデルを図郭毎に区切って格納したが、将来的には図郭を意識しない利用が望ましく、これには高速な空間検索が必要になる。また、空間系の検索・空間演算の負荷分散も望ましい。座標が十分離れた座標空間での検索処理は相互に影響しないので単純な分散実行が可能と考えられるが、近い座標空間での検索処理は相互の影響を考慮したマージを考慮する必要がある。

4.4. 空間検索とその他検索の連携

本プロトタイプアプリケーションでは 3D 都市モデルに属性(図 7)を付与しなかったが、本研究開発プロジェクトでは AI による建物抽出(図 8)、ハイパースペクトルセンサによる地表分類(図 9)を別途で検討した。また、近年の航空オプリークカメラには LiDAR が搭載されており、光学画像と点群を同時に取得できる。これらのセマンティクス情報と空間情報を組み合わせた検索が高速に実行可能になれば、例えば避難場所や緊急輸送道路から一定距離内の地形変動などを迅速な抽出が可能と考えられる。

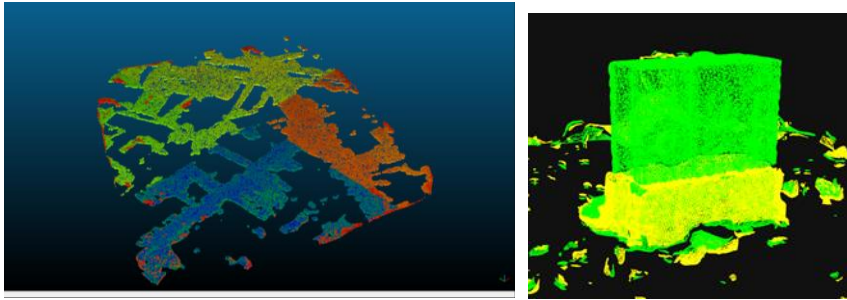


図7 3D都市モデルからの情報・属性抽出（左：平地抽出、右：差分抽出）

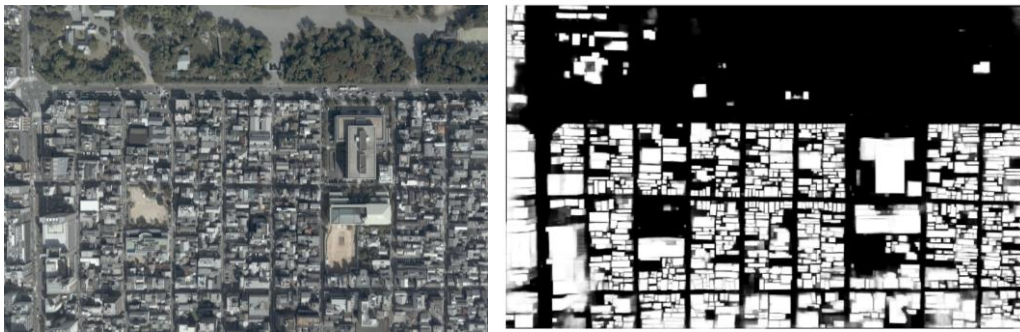


図8 AIを用いた建物抽出（左：直下視画像、右：抽出結果）

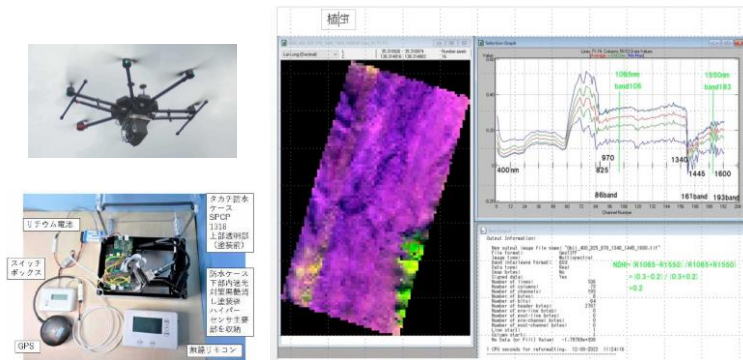


図9 ドローン搭載ハイパースペクトルセンサによる地表分類

5. 謝辞

本資料は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務「高効率・高速処理を可能とするA Iチップ・次世代コンピューティングの技術開発／次世代コンピューティング技術の開発／実社会の事象をリアルタイム処理可能な次世代データ処理基盤技術の研究開発」（JPNP16007）で得られたものです。