Vol.2020-CG-180 No.9
Vol.2020-DCC-26 No.9
Vol.2020-CVIM-223 No.9
2020/11/5

SIGMA フレームワークにおける時空間映像データの管理手法

山﨑賢人[1]　関斯琨[1]　松木輝[1]　木村朝子[1]　柴田史久[1]

概要：我々は多種多様なセンサによって実世界を観測し，得られた映像・データを共有した上で，それらを様々な形で利用したシステムを構築可能なアプリケーションフレームワークの実現を目指している．本フレームワークでは時間的かつ空間的に広がりを持つ映像・データを一元管理し，ネットワークを介して様々なアプリケーションにおいて蓄積した映像・データを利用可能にするような仕組みを提供することを目標としている．本稿では，我々が目指すフレームワークにおけるデータの管理手法，およびそれらを利用したシステムの構築例について述べる．提案したデータの管理手法は，鎌倉駅旧駅舎時計台を撮影した画像を用いて確認した．

キーワード：ネットワークサービス，アプリケーションフレームワーク，画像・データ

Design of Spatiotemporal Imagery Data Management Method on SIGMA Framework

KENTO YAMAZAKI[†1]　GUAN SIKUN[†1]
AKIRA MATSUKI[†1]　ASAKO KIMURA[†1]　FUMIHISA SHIBATA[†1]

Abstract: This paper describes the design and implementation of versatile application framework for handling spatiotemporal imagery data. The goal of the framework is sharing the imagery data captured from a wide variety of sensors in the real world and making it easy for developers to implement applications. However, the captured data is scattered across time and space; thus, we designed the unified data management for handling a variety of sensor data based on cloud computing. We implemented a prototype of our framework based on the proposed design. We visualize the 3D model of Kamakura Station old station clock tower from captured images to check the operation of the data management.

Keywords: Network service, Application framework, Imagery data

1. はじめに

いたるところに設置された監視カメラ，自動車の車載カメラや各種センサなど，散在するセンサによって実世界を観測した映像・データを常時記録する社会が訪れようとしている．今後活躍が期待されている低空を安定して飛行できる UAV（Unmanned Aerial Vehicle; 俗称ドローン）などの自律型移動体にもセンサは欠かせない．今後も我々の暮らす社会に存在するセンサの数は，増加の一途をたどるであろう（**図 1**）．

これらのセンサは，単独での利用では単に画像やセンサデータを記録・表示するだけにとどまるが，複数のセンサを組み合わせることで，より高度な利用が可能となる．例えば，複数のカメラ映像から建造物の 3D モデルが作成できる[1]．この例は空間的に広がった映像を利用したものであるが，映像は空間だけではなく，時間的にも広がりを持つ．例えば，カメラで取得した映像を蓄積することで，Mapillary などのように，ほぼ同じ位置で撮影した過去の映像を見ることができる[2]．さらに現在の映像の一部に過去の映像を組み合わる試みも始まっている[3]．東日本大震災の被災地を対象として，震災前後の様子を仮想化空間で再現するような試みもある[4]．

1 立命館大学大学院 情報理工学研究科
Graduate School of Information Science and Engineering, Ritsumeikan University

このように実世界を観測する映像やデータを高度利用するためには，映像やデータを収集・蓄積し，それを簡便に利用できるような仕組みが，システムを構築する上で欠かせない．そこで我々の研究グループでは大規模な映像やデータを管理し，ネットワークを介して活用するためのシステム構築を目指し，基本アーキテクチャの概念設計を進めてきた[5]．ここでは，時間的・空間的に散在する大量の映像やセンサデータを時空間映像データと称し，この時空間映像データを利用可能なアプリケーションフレームワーク SIGMA (Spatiotemporal Images with Generalized Management Architecture) を提案する．

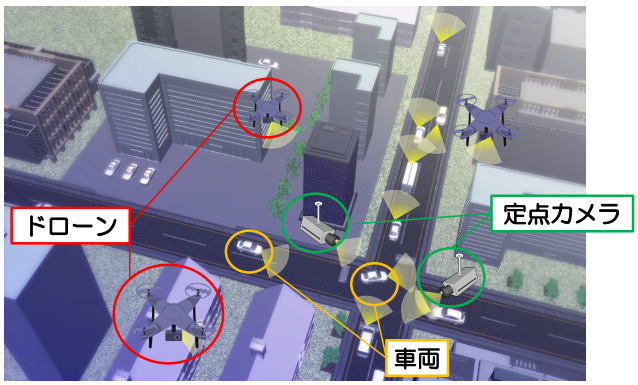

図 1 散在したセンサのイメージ
Fig.1 Image of sensors everywhere

Vol.2020-CG-180 No.9
Vol.2020-DCC-26 No.9
Vol.2020-CVIM-223 No.9
2020/11/5

本稿では，このフレームワークの実現を念頭に，以下の 2 つを要求仕様としたフレームワークの基本設計およびプロトタイプ実装の結果について報告する．

- 実世界を観測したセンサデータを時間軸と空間軸を画一的に扱うことが可能である
- データを共有でき，容易に取得できる仕組みを有する

2. 関連研究

実世界を観測した情報を蓄積し，共有するシステムは多く存在する．例えば，AWE(Augmented World Expo)の創設者のひとりである Inbar が提唱した AR Cloud もそのひとつである．AR Cloud では任意の座標系に配置した仮想物体と観測した点群を共有可能にすることを目指し，様々なプロジェクトが立ち上がっている[6]．また ITS（高度道路交通システム）分野でも同様に点群を自動運転などに活用している．このときの点群は 3 次元地図と称することが多い．3 次元地図とは，道路や建造物といった静的な物体を点群で表現する 3 次元の地図であり，モービルマッピングシステムなどによって作成される[7].

しかし静的な物体であっても時間経過によって形状が変化することがあるため，定期的な地図の更新は必要である．そこで Sakurada らは異なる時刻に撮影した画像から 3 次元形状を更新する方法を提案した[8]．他にも物体の 3D モデルを作成するにあたって，人物などの動体を，時間情報を用いて除外するなどの活用事例もある[9].

このように 3 次元のみを着目した場合では解決できなかった課題に対し，時間軸を含めた 4 次元で考えることによって様々な課題を解決する糸口を見出してきた．

そこで我々が提案するフレームワークでは実世界を観測したデータを高度利用できるようにするため，汎用的に 4 次元情報を共有し，活用できるシステムを構築する．

3. 時空間映像データの共有

3.1 基本設計

我々は文献[5]において，実世界を観測して得られた映像・データは広範な用途で利用するために，センサから取得したデータをそのまま保存し，利用方法はアプリケーション開発者に一任する方針を立てた．本フレームワークが扱うデータは一意の場所・時間で得られたデータではなく，様々な場所・時間に散在する．これらを共有するためには，いかにサーバで一元管理し，サーバ上のリソースデータベース（リソース DB）から任意のデータにアクセスできるかが重要視される．ただし，データに自由にアクセスする交換条件として本フレームワークで開発したアプリケーションは時空間映像データを自動的にアップロードすることが前提である．しかしアップロード機能自体は開発されたアプリケーションが提供する機能ではないため，アプリケーション開発者が実装せず，フレームワークに内包された機能としてバックグラウンドで実行されるものとする．

3.2 時空間映像データの管理方法

時空間映像データは実世界を観測して得られたものであるため，カメラの画像，LiDAR の点群，GNSS の測位情報などの多種多様な情報が想定される．これらの時空間映像データの内，アプリケーションが映像提示するためのものとして画像の取り扱いに焦点を当てる．さらに距離画像も画像のひとつとみなせるため，画像に準ずるものとして点群も想定する．

散在するセンサから得られる画像や点群を無秩序にサーバに蓄積した場合，実世界での関係性が失われてしまう．一般的に，多くのセンサはそれぞれが独立しているのではなく，他のセンサとの関係性が存在する．この関係性を維持するために本研究では，個々のセンサ間の関係を，木構造を用いて管理することにした．しかしセンサ自体を木構造のノードに見立てた場合，多層構造をイメージにしにくく，また，カメラの焦点距離が変更された場合のように，同一センサであってもパラメータが異なると同一センサとして扱う場合に不都合が生じる可能性がある．一方，時空間映像データ自身をノードにした場合，各ノードを時間や空間のエッジで結ぶと，木構造として成立せず，複雑なグラフになり，管理が難しくなる．そこで，センサではなく，独自に定義した抽象的な「Object」をノードとみなし木構造を構築した．また各ノードには画像や点群といった時空間映像データとメタ情報からなる「Resource」を紐づけする構造を設計した．つまり Resource には実世界を観測したときのセンサのパラメータ値もメタ情報として含む.

3.3 Object による管理

提案する木構造では，**図 2** に示すように，車両の前方に搭載したカメラや LiDAR がフロント Object に紐づき，リア Object とフロント Object は自動車 Object を親に持つ．このように Object は Resource を抽象化したものであるため，Resource は紐づいている Object の ID をメタ情報に有する．一方，Resource は複数のセンサが実世界を観測するごとに任意の Object に紐づくことから，Object と Resource は 1 対 n の関係であるため，Object は Resource の ID を持たない．

Object はアプリケーション開発者が想定する木構造を，リソース DB を管理している計算機にリクエストすることで登録される．本フレームワークのコンセプトでは異なるアプリケーション間であってもデータを共有可能なため，すべての Object の情報はアプリケーション開発者に開示される．**図 3** は，本フレームワークに登録された Resource を，サポートツールを用いて可視化したものである．このように，Resource から生成した点群を表示することで，実世界との関係も把握可能であり，さらに Object の構造が把握し

Vol.2020-CG-180 No.9
Vol.2020-DCC-26 No.9
Vol.2020-CVIM-223 No.9
2020/11/5

やすいように木構造の可視化も行う．また，任意の Object を選択することで，その Object をルートとする部分木のみを可視化することも可能である．Object の構造が把握しやすいようにグラフの可視化も行う．

これらのことから，既存の Object を確認しながら Object の新規登録および，既存の Object に親 Object や子 Object の追加登録も可能である．さらに複数の木構造のルート Object を子 Object にする新規 Object も追加であることから，複数の木構造を 1 つにまとめることもできる．つまり，アプリケーション開発者全員で Object の木を育てていくイメージである．

3.4 Geometry による検索

時空間映像データは Object と Resource によって管理されているが，任意の Resource を抽出するためには，指定した条件下の Object や Resource が検索できなければならない．

Object と Resource はともに ID による検索が可能であるが，さらに，本フレームワークでは 4 次元の情報を用いて検索ができるようにする．

Resource には画像や点群だけではなく，取得時のセンサの位置姿勢・時刻などのメタ情報が存在する．一方，Object は抽象的な存在であることから位置姿勢・時間を定義しなければならない．そこで Object の位置姿勢・時間の決め方は 2 パターン想定できる．

1 つ目はアプリケーション開発者が任意の値を与える方法である．例えば，あるビル内に複数の固定された監視カメラがあったとする．ビルを Object とした場合，位置姿勢・時間を，ビルの所在地や竣工日から与えることができる．2 つ目は「子 Object」や「紐づいている Resource」から計算する方法である．「子 Object」や「紐づいている Resource」が 1 つの場合の計算は容易である．一方，複数の場合については，適した値を計算可能かどうかについて先に検討する必要がある．

例えば，自動車に搭載した複数のセンサは，ドアとともに動く電子ミラー用カメラであっても自動車から離れていくことはない．ビル内を自由に動き回る監視ロボット群はビル内から出ることはない．これらのことから，複数の場合の「子 Object や紐づいている Resource」は法則性を持った群集とみなせるため，計算は複雑化するが，この法則性を基に Object に適した値を計算することができる．

しかし，Object 自身に複数の位置姿勢・時間をメタ情報として有することはできない．また，検索対象を一本化することで検索処理を統一できることから，Object や Resource を検索するために時間・位置を組み合わせた「Geometry」を定義する．Geometry は位置姿勢・時間を保持しており，これらの各パラメータが画一的に扱えるように設計している．

実世界を観測した Resource は即時アップロードされるため，位置姿勢の情報を付加してアップロードすることは難しい．これに対処するため，Geometry には 2 種類の位置姿勢を格納できるようにした．1 つ目は推定値であり，2 つ目は実測値である．前者は SfM や ICP アルゴリズムなどのように，様々な手法で推定した値を想定している．後者は GNSS の測位情報のように物理センサの値を想定した．推定値を付加する前にアップロードされる場合は，初期値として実測値を入れておく．ただし，より高精度な位置姿勢の結果を得られた場合は推定値の更新を許可する．こうす

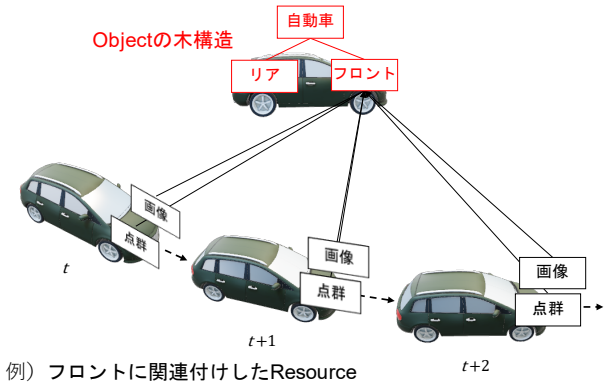

図 2 木構造における Object と Resource
Fig.2 “Object” and “Resource” on a tree structure

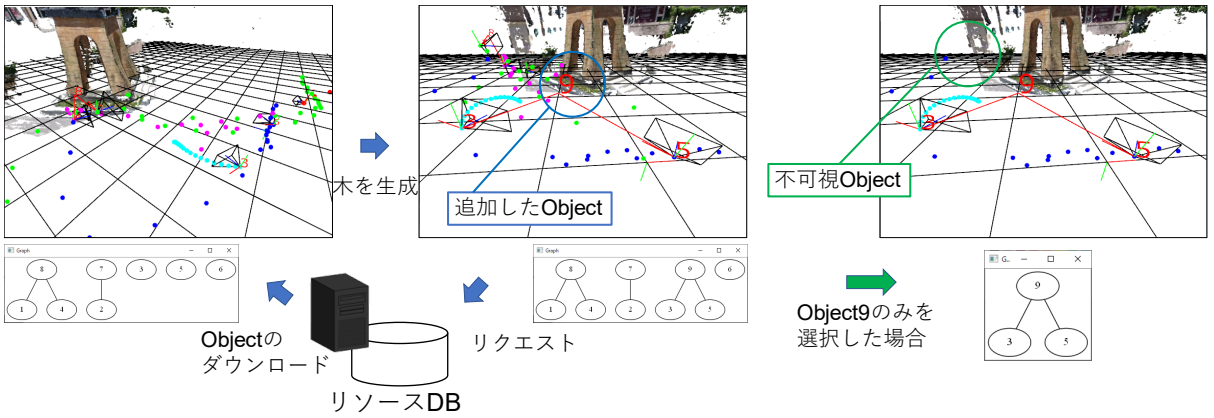

図 3 Object の設定方法
Fig.3 Method of Object management

Vol.2020-CG-180 No.9
Vol.2020-DCC-26 No.9
Vol.2020-CVIM-223 No.9
2020/11/5

ることで，検索は実測値ではなく，推定値のみに着目すれば必要条件を満たす．

3.5 リソースデータベースの設計

Resource を DB に蓄積し，要求に応じて検索するためには，Resource が関連付けられている Object の構造を損ねることなく蓄積する必要がある．これらの要求を満たすため，関係型データベース (Relational Database; RDB) を用いることにした．

一般的な方法として，Object ID や Resource の ID で検索も可能であるが，本節では本フレームワーク特有の検索方法について述べる．まず基本的なものとして，Geometry を用いて 4 次元時空間を「範囲」や「距離」によって検索する方法を用意した．

次に，本フレームワークの特徴である時空間映像データ特有の検索方法について述べる．「範囲」や「距離」による検索方法はカメラや LiDAR などのセンサの観測位置に依存するものであり，キャプチャした画像・点群の被観測物と一致するとは限らない．例えば，カメラで撮影した被写体は，カメラの位置から数 m 離れた位置に存在することが一般的である．しかし，被観測物ベースで Resource を検索したい場合がある．被観測物をベースに検索するには，キャプチャされた画像・点群に何がキャプチャされているかを調査する必要がある．リソース DB の検索において，それらを調査してから結果を返すのは非現実的である．そこで解決策として，指定した位置に向いているセンサでキャプチャしたものに限定する手法を提案する．しかし，向いているだけでは検索範囲には無数の画像・点群が存在する．各センサの解像度/分解能の差を無視するならば，指定した位置に近いセンサのほうが，より高解像度/高分解能になる．つまり Resource を抽出するために次の条件での検索手法も追加した．

- 指定した位置を向いているセンサが観測したもの
- 指定した位置から任意の範囲のもの

これらの検索方法が実現するため，RDB に 5 つのテーブルを用意した（**図 4**）．

Resource テーブルにおける種類は登録された Resource の種類であり，1 を画像，2 を点群とデフォルトで設定した．

Geometry テーブルの時間は UNIX 時間と同じように任意のタイミングからの経過時間で表すものを使用する．これはスケールの違いはあっても空間（位置姿勢）と同じように 10 進数で表すためである．しかし UNIX 時間では粒度が秒であることから，一般的なセンサのフレームレートより粗い．本フレームワークでは ROS Time のようなミリ秒まで扱えるものを想定している．

Resource テーブルにおける位置姿勢は推定値と実測値を保持しているが，どちらも同じ座標系で扱う．本フレームワークを基に開発されるアプリケーションが取り扱う空間の規模は任意である．しかし，リソース DB に蓄積する値には基準となる座標系が必要である．そこで，UTM 座標系と標高を基準に，位置はゾーンと(x, y, z)，姿勢は四元数(x, y, z, w)で表す．パラメータは画像・点群が取得したときのセンサのパラメータを指す．リソースのパスは画像や点群が保存されている補助記憶装置のファイルパスである．画像は png 形式，点群は pcd 形式などの可逆圧縮が望ましい．

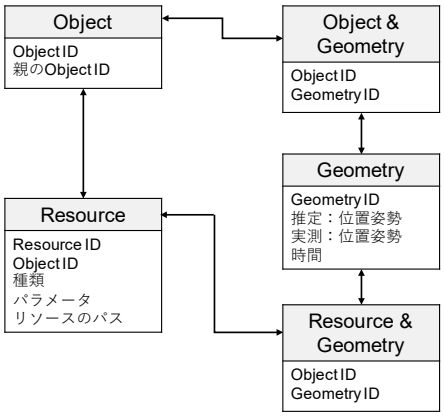

図 4 テーブルの関係
Fig.4 Relation of the table

この RDB から Geometry を用いて，Resource や Object の「範囲」「距離」を検索する．両検索手法ともに中心座標値(x, y, z, t)と，「範囲」ならば 3 次元で例えると立方体の辺の長さ，「距離」ならば 3 次元で例えると楕円球の径の長さ(a, b, c, d)を入力に検索結果を得ることが可能である．また視点によって検索する場合は，センサが視点を向いていても，センサの光軸の延長線上に視点があるとは限らない．そこで許容範囲として視点の座標値とセンサの光軸とセンサ中心から視点への直線とのなす角の閾値から検索できる．例えば，**図 5** に示すとおり，条件 1 に合うのはカメラ A でキャプチャした画像とカメラ B でキャプチャした画像となる．閾値thが$\theta < th < \phi$の場合，条件 2 に合うのはカメラ C のキャプチャ画像となり，検索結果はカメラ C のキャプチャ画像となる．

4. 試作

本評価用にリソース DB にアクセス可能な三次元再構成アプリケーションを開発した．このアプリケーションを用いて DB から任意の画像をロードできているかを確認する．開発したアプリケーションは得た画像群から SfM を用いてカメラの位置姿勢と疎な点群を構成した後，PatchMatch Stereo をベースに密な点群を構成するものである[10].

Vol.2020-CG-180 No.9
Vol.2020-DCC-26 No.9
Vol.2020-CVIM-223 No.9
2020/11/5

まず，事前準備として鎌倉駅旧駅舎時計台を周回しながら写真を 2 種類のカメラで撮影し，撮影画像 104 枚をリソース DB に登録した．本評価は散在したセンサで撮影した画像を用いての機能確認であり，三次元再構成そのものの機能確認ではない．そのため，三次元再構成の精度は確認対象とはしない．

そして，任意の範囲を指定することで，TCP/IP をリソース DB のある HTTP サーバから Resource としての画像とメタ情報をダウンロードした．メタ情報には関連する Object ID が既知であることから，ダウンロードした画像を撮影したセンサごとに時系列で並べることができるため，Resource のメタ情報のみの使用時よりもカメラモーションをロバストに推定できる．このように Object を活用して生成した広場の三次元再構成結果を**図 6** 左に示す．

次に，視点のよる検索の動作確認のため，時計台のみを三次元再構成する．そのために，時計台の 2 点に指定し，HTTP サーバから Resource をダウンロードした．改めてダウンロードした結果を用いて三次元再構成した結果を図 6 右に示す．赤枠内の案内板などがなくなっていることがわかる．このことからリソース DB と設計した検索手法の動作を確認できた．

このとき指定した角度の閾値は，使用したカメラの画角に近い 60° であり，ダウンロードできた画像は 37 枚であった．本確認において 2 点選んだのは，被写体が近い撮影画像しかなく，1 点では時計台全体を再構成できなかったためである．したがって，近くから撮影した場合は点ではなく，異なる指標を用いて Resource を検索する必要があることがわかった．

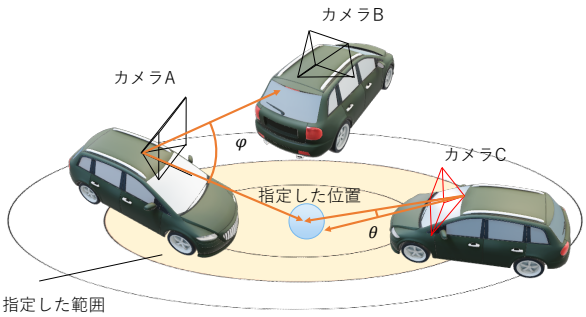

図 5 視点による検索
Fig.5 Search from viewpoint

5. むすび

本稿では，時空間映像データの将来的な高度利用を念頭に，実世界を観測した大量のデータを収集・蓄積することで時空間映像データを利用可能なアプリケーションフレームワークの基本設計について述べた．この基本設計を基に時空間映像データを共有するためのリソース DB を試作した．試作したリソース DB を用いて，視点を限定したリソース検索機能が有効に働くかを確かめ，抽出したカメラ画像から対象を限定した任意の風景を三次元再構成可能であることを確認した．今後は，時空間映像データの高度利用のための機能を拡充するとともに，フレームワークを利用したアプリケーションの事例を増やす予定である．

謝辞 本研究の一部は，科研費：基盤研究(B)課題番号 17H01747 による．

参考文献

[1] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R.: Building rome in a day, *Communications of the ACM*, Vol. 54, No. 10, pp. 105 - 112 (2011).
[2] Juh´asz, L. and Hochmair, H. H.: Cross-linkage between Mapillary street level photos and OSM edits, Geospatial Data in a Changing World, *Springer*, pp. 141 - 156 (2016).
[3] Suzuki, K., Wakisaka, S., and Fujii, N.: Substitutional reality system: a novel experimental platform for experiencing alternative reality, *Scientific reports*, Vol. 2, p. 459 (2012).
[4] 池内克史，大石岳史，小野晋太郎，岡本泰英，鎌倉真音：まちと震災のいま・過去を「仮想化空間」で伝える，生産研究，Vol. 68, No. 2, pp. 123‐126 (2016).
[5] 山﨑賢人，有富友紀，关斯琨，木村朝子，柴田史久：実世界観測による時空間映像データの高度利用 (1) —基本アーキ

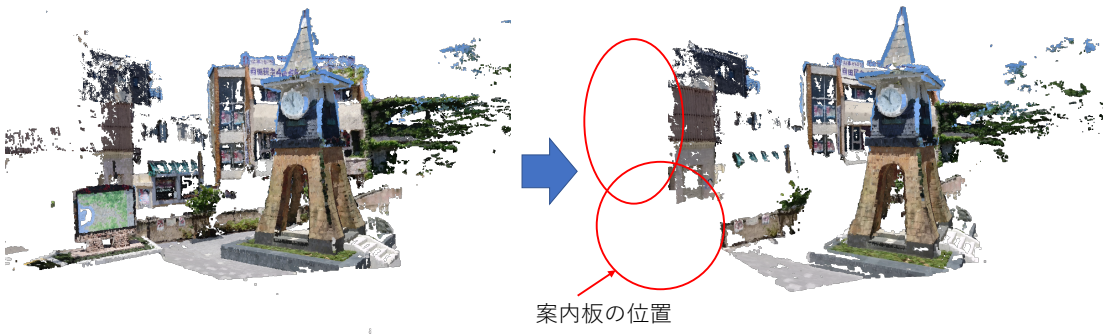

図 6 三次元再構成結果
Fig. 6 Result of 3d reconstruction

テクチャの概念設計と第 1 次システム試作―，日本 VR 学会複合現実感研究会，MR2019-15，日本 VR 学会研究報告，Vol. 22, No. 2, pp. 49 - 53 (2019).

[6] 原田昌亮： Future ICT Trend : AR クラウドが産み落とす第三の巨大な波: 実現へと邁進するミラーワールド, InfoCom T&S world trend report:世界の情報通信サービスの情報誌， No. 367, pp. 20 - 25 (2019).

[7] Manandhar, D. and Shibasaki, R.: Vehicle-borne laser mapping system (VLMS) for 3-D GIS, IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium, Vol. 5, IEEE, pp. 2073 - 2075 (2001).

[8] Sakurada, K., Okatani, T., and Deguchi, K.: Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 137 - 144 (2013).

[9] Furukawa, Y. and Ponce, J.: Accurate, dense, and robust multiview stereopsis, *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, Vol. 32, No. 8, pp. 1362 - 1376 (2009).

[10] Bleyer, M., Rhemann, C. and Rother, C.: PatchMatch Stereo - Stereo Matching with Slanted Support Windows, Proceedings of the British Machine Vision Conference pp. 14.1 - 14.11, (2011).