

## IoT 環境におけるデータベースを用いた点群管理の検討

## Point Cloud Management with a Database in IoT

松本 佳大  
Yoshihiro Matsumoto

杉浦 健人  
Kento Sugiyura

石川 佳治  
Yoshiharu Ishikawa

陸 可鏡  
Kejing Lu

## 1 はじめに

光の照射で物体との距離を計測する *LiDAR* (*light detection and ranging*) センサの普及により、点群データのリアルタイムでの取得及びその活用が可能となった。LiDAR はレーザー光を周囲に照射し、物体に反射して跳ね返るまでの時間を計測することで物体との距離や方向を測定する。具体的な活用例としては、飛行機に LiDAR センサを取り付けて観測を行い、撮影した地域の点群データによって地形調査を行った事例が挙げられる。また、GPS のデータを正確に得られない屋内でも点群データを獲得し、物体認識や自己位置推定が可能となる利点から自動運転の分野でも研究が進んでいる [1]。

一方、機器の普及に伴い異なる観点からの活用も期待され始めている。例として、LiDAR センサを用いた混雑状況の予測について述べる。ある街頭に LiDAR センサを設置したとき、得られる点群データをリアルタイムに解析することで点群の変動に基づく混雑状況の推定が行える。点群データには個人を特定できる情報が含まれないため、定点カメラのように画像や映像が残ってしまう方法に比べ、個人情報への配慮及びその管理コストを抑えられるという利点がある。

しかし、点群データのリアルタイムな管理及び処理を考えたとき、既存手法には性能やそもそのデータ保持方法に関して課題がある。LiDAR センサは周辺に存在する物体の表面情報を点群データとして取得するため、継続して周辺状況を監視する場合、データの取得間隔に応じて膨大な量の点群データが蓄積される。これに対し既存の点群を扱う技術はある特定のスナップショットを表す点群を主な処理対象としており、このような時系列点群データを効率的に扱えない。しかし、例えば先述した街頭での混雑予測であれば周辺の建物に該当する点群は時間変化に対してほぼ定常であり、その性質を利用した圧縮などが考えられる。

そこで、本研究では既存のデータベース技術を応用し、LiDAR センサから得られる時系列点群データの管理及び処理について提案する。特に本稿では、既存の点群データの管理手法についてまとめ、時系列点群データの効率的な管理方法について検討する。

## 2 関連研究

本章では関連研究として、ファイルに基づく点群管理とデータベースを用いた点群管理について述べる。

### 2.1 ファイルに基づく点群管理

ファイルを用いた管理方法について述べる。PLY や OBJ など、様々な企業や団体によって三次元オブジェクトを表すためのファイルフォーマットが開発されており、点群データもそれらのフォーマットで管理できる。ファイルに基づくデータ管理は伝統的な方法であり、使用できるソフトウェアやライブラリが多い、フォーマットに応じた圧縮方式が存在するなどの利点がある [2]。ただし、対応するソフトウェアさえあればデータの読み込みを簡単に行える一方、フォーマットが異なる点群データの管理が煩雑となるなどの問題もある。

検索効率の向上のために、一部のファイルフォーマットではファイル内に別ファイルへのリンクを埋め込むことによって、ファイル上で点群データの索引を構築できる。例えば、LiDAR センサで得られる点群データの標準的なフォーマットである LAS ファイルでは空間充填曲線と八分木の二つの索引を利用できる [3]。

一方で格納効率の向上のために、一部のファイルフォーマットでは効率的な圧縮手法も開発されている。例えば、LAS の圧縮フォーマットとして Rapidlasso の LAZ (LASzip) や ESRI ZLAS が存在する。ただし、こうした圧縮は処理するファイルの数やサイズの増加によるスケーラビリティの問題を持つ。

### 2.2 データベースでの点群管理

データベースを用いた点群管理として、pgpointcloud を使用した点群管理について述べる。pgpointcloud は OSS (open source software) データベースの一つである PostgreSQL の拡張モジュールであり、PcPoint と呼ばれるデータ型を定義する。PcPoint 型は複数の次元、または各次元に対して複数の属性を設定可能であり、LiDAR センサの持つ様々な変数を保持できる [4]。例えば各点の RGB カラーモデルなどを定義でき、他にも検出時間といった、LiDAR センサから得られる三次元位置情報以外のデータも効率的に格納できる。

更に、点群データをテーブル上で効率的に保持するために、PcPoint 型をまとめた PcPatch と呼ばれるデータ型も使用できる。データベースのテーブル上では一つのデータを一つのタブルで表すが、点群管理において点データを個別に扱うのは非効率的である。そのため、PcPatch 型は PcPoint 型を集約した点群ないし部分点群を一つのデータとして格納し、タブルとしての管理を容易にする。これにより、膨大な量のタブルを保持することで生じるオーバーヘッドの問題も回避できる [5]。また、pgpointcloud はデータの圧縮方法として次元データの配列として保存する Dimensional, LiDAR データを保存するための標準バイナリ形式を圧縮した LAZ をサポートしている [4]。

点群データを操作する際は、この PcPatch 型に対してフィルタリングや索引の作成といった処理を施す。PcPatch 型の利用により、例えば数十億個の点群データも数百万程度の PcPath 型として扱える。つまり、B+ 木や R 木といった PostgreSQL に元々備わっている索引を利用する際にも、削減されたタプルに対して効率的に索引を構築できる。また、共通のフレームワークの提供や索引サイズの削減といったメリットもある。

### 3 時系列点群データの管理

本章では、時系列点群データをデータベース上で扱う際に必要となる機能について述べる。以下では圧縮、検索、及び分析支援の三つの機能について述べる。

#### 3.1 時系列情報に基づく圧縮機能

時系列点群データは時間軸に対して細かい粒度で点群データが取得されるため、時間変化の少ない点群を効率的に圧縮する機能が必要となる。一般に点群データのデータ量は大きく、時系列で取得される点群を全てデータベース中に保持するのは現実的ではない。しかし、例えば街頭のモニタリングを考えた場合、往来する歩行者以外にもビルのような時間変化が少ない点群データも存在すると考えられる。したがって、そのような時間変化の少ない点群データを時間軸方向に圧縮することで、ストレージ使用量の大幅な削減が可能だと考える。

一方で、点群データの時間変化のモデル化による圧縮も考えられる。例えば、平行移動する部分点群など、線形関数などでモデル化可能な点群を考える。この場合、平行移動の開始時刻と終了時刻、及びそのモデルのみを保存することで部分点群は復元可能となる。これは前述した時間変化のない点群の圧縮の一般化として考えられるが、モデル化した部分点群以外の点群との兼ね合いをどう処理するかなど、技術的な課題も多い。

#### 3.2 時空間検索機能

時系列点群データの検索には点群内の各点の位置だけでなく、時系列情報にも対応可能な時空間索引が必要となる。ここでは、典型的な検索として 1) ある時刻におけるスナップショット点群の検索、及び 2) 連続した時区間の点群データの検索について考える。

スナップショット点群の検索は時間軸に対する点検索及び三次元空間に対する範囲検索であり、特定の時刻に撮影された点群を効率的に検索する索引が必要となる。このような検索は同地点における時間別の点群データの比較などの際に必要となる。単純には時刻を基にパーティショニングを行う要領で索引を構築すればよく、単体であれば構築は難しくない。

一方で、連続した時区間の点群データの検索は時空間全てに対する範囲検索である。このような検索は、例えばある興味深いオブジェクトが存在した時区間の点群を確認したい場合に必要となる。前述のスナップショット点群の検索に対してこちらでは時空間の合計 4 次元を考慮した索引構築が必要となり、時間及び空間の両方の性質を適切に処理できる索引構造が望ましい。

更に、以上の索引を構築する際は前述した点群データの圧縮についても考慮しなければならない。圧縮後の点群データはある時間幅を持つデータとなり、時刻のみを持つ点群データとス

キーマ上でも差異が生まれる。そのため、圧縮・非圧縮のデータを分離して格納する、もしくは空間データにおける点データとポリゴンデータのように統一して扱えるような索引を使用するなど、適切な手法の選択が必要となる。

#### 3.3 時系列点群データの分析支援機能

圧縮や検索に加えて、時系列点群データの分析支援も必要な機能である。例えば、点群データを用いた分析としては差分抽出やそのための点群の位置合わせが考えられる。これらの処理は頻繁に使用されることが予想されるため、共通して必要な処理をデータベース側で行うなどの補助が考えられる。

また、点群データのフィルタリングや可視化、他のデータベースのデータとの統合機能なども分析を進めるためには必要になると考えられる。時系列上で変動が少ない点群データを圧縮の有無を基準にフィルタリングで取り除き、変動の多いデータを抽出する。検証したい点群データを可視化して比較や検討を実行しやすくするほか、統合機能によって他のデータとも照らし合わせることが出来る。

### 4 おわりに

本稿では LiDAR センサから得られる点群データとして時系列点群データを想定し、その管理手法について検討した。関連研究として pgpointcloud やそれを用いたデータベースでの管理、そしてファイルに基づく管理方法を述べた。今後の方針としては本稿で述べた各機能の詳細な検討や、データベース上での実装を行う予定である。

### 謝辞

本研究は JSPS 科研費(16H01722, 20K19804)の助成、及び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(JPNP16007)から得られた結果による。

### 参考文献

- [1] 伊藤 敏夫, *自動運転のための LiDAR 技術の原理と活用法*. 科学情報出版株式会社, 2020.
- [2] S. Psomadaki, "Using a database for dynamic point cloud data management," Master's thesis, Delft University of Technology, 2016. (Graduation Plan).
- [3] F. Alvanaki, R. Goncalves, M. Ivanova, M. Kersten, and K. Kyziarakos, "GIS navigation boosted by column stores," *PVLDB*, vol. 8, pp. 1956–1959, 2015.
- [4] "GitHub - pgpointcloud/pointcloud." <https://github.com/pgpointcloud/pointcloud>.
- [5] R. Cura, J. Perret, and N. Paparoditis, "Point cloud server (PCS): Point clouds in-base management and processing," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W5, pp. 531–539, 2015.