

# LiDAR データの角度の差分と距離の差分による データ量の削減

山本 真也<sup>1</sup> 大沢 恭平<sup>2</sup> 串田 高幸<sup>1</sup>

**概要：**飲食業界では、人手不足と労働生産性向上のために、自律走行搬送ロボットの活用が進んでいる。このようなロボットは常に変化する店のレイアウトや予測できない人間の動き、障害物に対応するために LiDAR を利用している。LiDAR はリアルタイムで周囲の環境を点群データとして生成することができる。基礎実験より、LiDAR は 1[s] に約 178[KB] の測定データを出力する。このデータ量を送信するには時間がかかるため、データ量の削減が課題として挙げられる。本稿では、LiDAR によって測定された角度の差分と距離の差分を用いた可逆圧縮手法を提案する。LiDAR データを取得して整形を行い、処理を行うデータと 1 つ前のデータの、角度の差分と距離の差分を出力する。出力した、値は固定長ビットでバイナリにエンコードされ送信される。基礎実験では、LiDAR を使用して 30[s] 計測したデータと 60[s] 計測したデータのサイズを確認し、サーバーに送信しきるまでの時間を計測した。基礎実験の結果、30[s] 計測したデータでは約 5.1[MB] のサイズが出力され送信には約 440[ms] かかった。60[s] 計測したデータでは約 11[MB] 出力され、送信には約 960[ms] かかった。評価実験では LiDAR が移動した際のデータや周りの環境が変動するデータを圧縮し圧縮率や送信完了するまでの時間を評価する。

## 1. はじめに

### 背景

インダストリー 4.0 の台頭により、ロボット産業は重要な役割を持つ。自動車、航空、製造、倉庫、医療、飲食の分野で広く応用されつつある [1,2]。特に飲食業界では、人手不足や労働生産性向上に向けて、サービス分野におけるロボットの活用が進んでいる。特に、レストランにおいて配膳を行うロボットが用いられている [3]。また、自律走行搬送ロボット (AMR: *Autonomous Mobile Robot*) の導入が進んでいる。これによりレストラン内を移動して注文を受け、料理を配達するサービスの自動化を行っている [4]。レストランでは、障害物の配置に適應すること、予測できない人間の動きに対応することが課題となる [5-7]。AMR は Light Detection and Ranging (LiDAR) を活用することでこれらの課題を対処している [8]。LiDAR により周囲の環境をリアルタイムで把握し、障害物を回避しながら安全に移動することが可能である [6]。LiDAR とはリアルタイムで点群データを生成するセンサーの一種である [9]。レーザーを継続的に発射し、光がセンサーに戻ってくるま

での時間を測定する [10]。ターゲットとの距離、速度、方向に関する情報を非常に短時間かつ高精度で取得することができる [11]。しかし、短時間で大量にデータが取得できるため LiDAR から出力されるデータは大きい。そのため、処理に多くの計算リソースが必要になる [10]。通信を行う際に、クラウドコンピューティングのネットワークリソースを活用することで迅速に処理することが可能になる [12]。ネットワークリソースを使用することに加え、LiDAR から出力されるデータに圧縮技術を利用する方法が挙げられる [13]。例として、オープンソースで使用できる LASzip, PCL(*Point Cloud Library*), Draco, G-PCC, V-PCC, RLE と呼ばれるツールや圧縮技術がある [14-16]。

### 課題

LiDAR センサーはリアルタイムで環境をスキャンし、角度と距離が示されたデータを 1[s] に約 178[KB] 出力する。このデータ量をクラウドに送信するには時間がかかる点を挙げる。LiDAR データの送信には通信帯域が圧迫される [10,17]。図 1 は通信帯域が圧迫した際の影響を示す。サイズが大きくなれば送信にかかる時間は増加する。送信に時間がかかれば、リアルタイム性が重要な状況では悪影響を及ぼすことがある [18,19]。例として、ロボットが人や障害物を避ける際の動作に影響を及ぼすことが挙げられる。

<sup>1</sup> 東京工科大学コンピュータサイエンス学部  
〒192-0982 東京都八王子市片倉町 1404-1

<sup>2</sup> 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻  
〒192-0982 東京都八王子市片倉町 1404-1

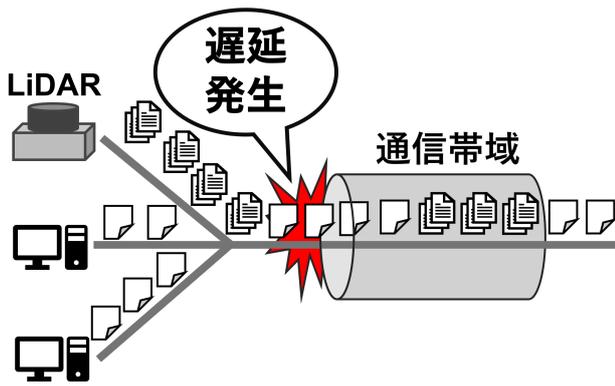


図 1: 通信帯域の圧迫による影響

## 各章の概要

第 2 章では、本論文の関連研究を説明する。第 3 章では、本研究の提案手法を説明する。第 4 章では、提案手法の実装を説明する。第 5 章では、実験環境と実験の結果の分析を説明する。第 6 章では、提案、実験、評価が本研究の課題を解決しているかを議論する。第 7 章では、本稿のまとめを行う。

## 2. 関連研究

取得した LiDAR データをセグメント化し、各セグメントの特徴の抽出を行い比較する。これにより障害物を特定する SegMatch と呼ばれるアルゴリズムがある [20]。SegMatch では、ロボットが位置を中心とした特定の範囲内の点を取得する。その点からノイズを除去し、残ったデータで点同士が近いものをクラスタごとに、セグメント化する。クラスタの特徴を抽出し、異なる時点で取得したクラスタと比較する。これにより、複雑な環境で物体や構造の一致を見つけることができる。この研究では、1[Hz]の周波数で正確な位置特定ができたとあるが、周波数が低いとレストランなどの環境内に移動する物体が多い場合、ノイズが多くなり障害物の特定が難しくなってしまう。

点群データに対し、時間的冗長性を活用した非可逆圧縮を提案する研究がある [21–23]。この研究では、オクトリーデータ構造という空間を 8 つの同じサイズの立方体に再帰的に分割し、点の分布を表現する方法を利用する。この構造は、点群データを管理し、分割した各空間の関係性を保持するために使用される。これにより分割した空間ごとに重複する点を特定し、取り出すことでデータの冗長性を利用して圧縮している。ただし、この提案では物体が少ない環境や静的な環境でのみ効果的な方法である。データに変動のある環境であれば差分が増え圧縮できるデータが少なくなってしまう。

LAS 形式の LiDAR 用ロスレス圧縮ツールである LASzip を提案する研究がある [14]。LiDAR から取得した点群データを専用のツールやソフトウェアを使用して LAS 形式のデータを LAZ 形式のデータに圧縮する。LAS 形式は AS-

PRS(American Society for Photogrammetry and Remote Sensing) 協会が策定した標準化フォーマットである。冗長性の削除やランレングスエンコーディングを用いることによって、ロスレス圧縮を行っている。この手法では、事前に特定のツールの準備が必要になる。本稿ではツールを使用する必要がないシンプルなデータ圧縮を提案する。

LiDAR が出力した距離データを直接エンコードする提案がある [24]。この提案では、点群データを画像として扱い既存の画像圧縮アルゴリズムを応用することでデータの圧縮を行う。圧縮アルゴリズムとしては MPEG, JPEG を用いている。画像圧縮方法が点群データ圧縮において有効であると示されている。しかし、MPEG, JPEG は圧縮時に一部のデータが削除されるため、品質が低下する。この研究では、非可逆圧縮方法は特定のアプリケーションに依存すると記述されているが具体的なアプリケーション名が挙げられていない。そのため、画像圧縮技術を用いた際の品質の低下がどの程度許されるのかが分からない。

## 3. 提案

本稿では、測定データ間の角度の差分と距離の差分を用いた、LiDAR データの可逆圧縮方法を提案する。提案方式では以下の 4 つの項目について説明する。

- 各測定データ間の差分計算
- 角度差と距離差の出力
- 出力データの送信
- 解凍方法

### 提案方式

- 各測定データ間の差分計算

LiDAR データの出力結果を一部抜粋したものを出力結果 1 に表す。

出力結果 1: LiDAR データ (一部抜粋)

```
1 Ultra simple LIDAR data grabber for SLAMTEC
  LIDAR.
2 Version: 2.1.0
3 SLAMTEC LIDAR S/N:
  B5F6E1F4C2E398C0BCEA9AF33B2D4806
4 Firmware Ver: 1.01
5 Hardware Rev: 18
6 SLAMTEC Lidar health status : 0
7   theta: 0.39 Dist: 02754.00 Q: 47
8   theta: 0.95 Dist: 02752.00 Q: 47
9   theta: 1.50 Dist: 02751.00 Q: 47
10  theta: 2.06 Dist: 02749.00 Q: 47
11  theta: 2.61 Dist: 02747.00 Q: 47
12  theta: 3.17 Dist: 02747.00 Q: 47
13 ...
14 ...
15 ...
```

LiDAR データの構成について説明する。1 行目から 6 行目にヘッダー情報が記入される。LiDAR デバイスのバージョンやハードウェアの詳細、ステータス情報が含まれる。ヘッダー情報の後には、LiDAR が取得した測定点のデータが続く。データには「theta」、「Dist」、「Q」の 3 つのパラメータが出力される。

• theta

「theta」は LiDAR から放射されるレーザーの方向を表している。

• Dist

「Dist」は LiDAR から放射されたレーザーが物体に当たり、反射して LiDAR に戻るまで時間を計測し LiDAR から物体までの距離を示す。

• Q

「Q(quality)」は測定品質を表している。「Q」の値は変化がなく基本的に同じ値を出力する。「Dist」が正しく取得できていない場合、「Q」は 0 になる。

本稿で用いる LiDAR では、「theta」は 0.01 度刻みで「Dist」は 1[mm] で出力される。LiDAR が出力したデータの 0 度から 192 度の範囲のデータを読み込む。本提案で使用する角度については PUDA 社の「Bellabot」の検知角度を参考にした\*1。また、LiDAR は正面を 0 度として設定されている。0 度から 192 度の範囲で値を取得するために LiDAR 本体の向きを 96 度傾ける。測定データの取得範囲についてを図 2 に示す。

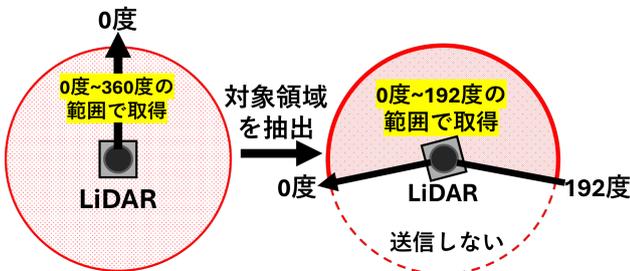


図 2: 測定データの取得範囲

LiDAR データを整形し各測定データの「theta」と「dist」の値のみを取得する。図 3 は、LiDAR データの整形例である。本稿で用いた LiDAR では、1 周あたり約 600 個のデータを出力する。加えて、0 度から 192 度の範囲では、約 340 個のデータを出力する。

次に、各連続する測定データの角度差と距離差を計算する。角度差は、取得した測定データの角度から 1 つ前の測定データの角度を引いた値として計算される。角度差は整数に変換して出力する。出力に少数が含まれることにより、値が整数か小数を判断するために追加ビットが必要になる。表現に必要なビット数を抑えるため整数で表現

\*1 <https://www.pudurobotics.com/jp/products/bellabot>

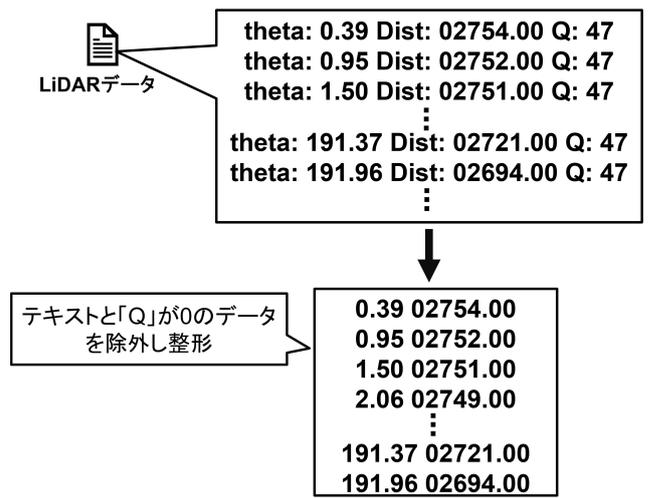


図 3: LiDAR データの整形

する。LiDAR データは 0.01 度刻みで「theta」が取得できるため、整数値に変換する際は「theta」に 100 を乗算する。距離差は、取得した測定データの距離から 1 つ前の測定データの距離を引いた値を符号付きで出力する。なお、LiDAR データの最初の測定点を初期値として扱う。出力結果 1 における初期値としては 39 と 2754 となる。

• 角度差と距離差の出力

図 4 に角度差と距離差の出力について示す。初期値は 39 と 2754 となる。それ以降の測定データでは計算された角度差と距離差を出力する。これにより、測定データの角度

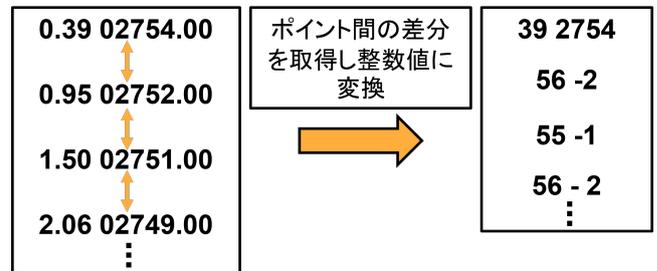


図 4: 角度差と距離差の出力

差と距離差の組み合わせが圧縮データとして生成される。また、圧縮されたデータは 1 行にまとめて出力される。出力結果 1 をもとに計算した結果を出力結果 2 に表す。

出力結果 2: 圧縮データ (一部抜粋)

1 39 2754 56 -2 55 -1 56 -2 55 -2 56 0 54 -1...

出力結果 2 では、初期値である「theta」の整数値である 39 と「Dist」の 2754 が出力される。続いて、次に測定された「theta」の 0.95 と初期値の「theta」の 0.39 の差を計算し整数値として 52 が出力される。また、次に測定された「Dist」の 2752 と初期値の 2754 の差を計算し-2 が出力される。以降は同様の手順で、処理する測定点と 1 つ前の

測定点の差を計算し出力していく。192度まで差を計算したら、次の0度の測定データから計算を繰り返す。

● 出力データの送信

測定データの角度差と距離差の組み合わせを出力する。また、圧縮されたデータは1行にまとめて出力される。0度から192度まで差を計算したら、サーバーへ角度差と距離差のデータを送信する。また、次の0度の測定データから計算を繰り返す。角度差と距離差のデータを送信する際に、固定長のビット数でバイナリデータにエンコードする。角度差と距離差の固定するビット数はそれぞれ異なる。本稿で用いるLiDARはステップ角が標準で0.72度であるため、整数値に変換すると72になる。したがって、0から72の範囲を表現できる必要がある。0から127を表現できる7ビットが適しているため、角度差は7ビット固定で表現される。また、最大で半径12[m]の範囲で値を取得できる。つまり、出力される距離差の最大値としては12000となる。したがって、0から12000の範囲を表現できる必要がある。0から16384を表現できる14ビットが適しているため、距離差は14ビット固定で表現される。以上の工程を踏まえて出力結果2に出力結果1を圧縮した結果を示す。

出力結果 3: 圧縮データ (一部抜粋)

```
1 0100111 01010110001010 0111000 11111111111110
   0110111 11111111111111...
```

● 解凍方法

図5に角度差と距離差の出力について示す。送信された

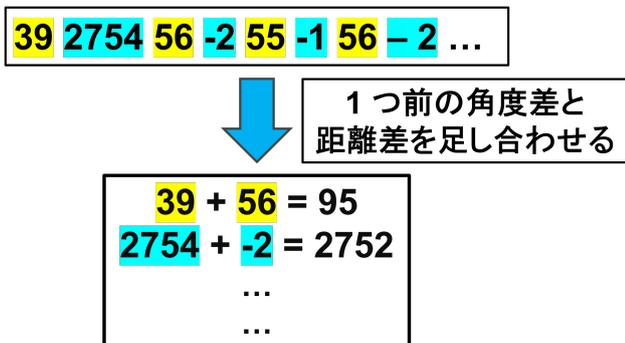


図 5: 圧縮データの解凍

データを解凍する際は、7ビット、14ビットの順で区切り解凍する。圧縮データの最初のペアである、「39 2754」は最初に測定された値なので、差分計算は行わない。「theta」を整数値からrawデータの単位に戻すため、値は「0.39 2754」となる。次に、圧縮データのペア「56 -2」を処理する。1つ前の角度差と距離差を足し合わせ、値を圧縮前に戻す。そのため値は「0.95 2752」となる。以降は、同様の手順で加えていく。

● 圧縮率

LiDARから取得した測定データに対し、0度から192度の

範囲の測定点を抽出し差分を取り圧縮を行った。圧縮前のデータサイズと圧縮後のデータサイズを表1に示す。圧縮前のデータサイズは、ファイル内に半角文字しかないため文字数がバイト数になる。圧縮後のデータサイズは、各測定データのビット数に、測定データ数をかけて合計ビット数を算出する。各測定データのビット数は角度差と距離差のビット数の合計で表される。合計ビット数を8で割った数がバイト数になる。

表 1: LiDAR データの圧縮前と圧縮後

|        | 圧縮前のデータサイズ | 圧縮後のデータサイズ |
|--------|------------|------------|
| データサイズ | 2062[byte] | 880[byte]  |

圧縮の結果より、約42.6%までデータサイズを削減することができる。

ユースケース・シナリオ

レストランで使用されるケータリングロボットを想定する。図6にユースケースシナリオを示す。ケータリングロ

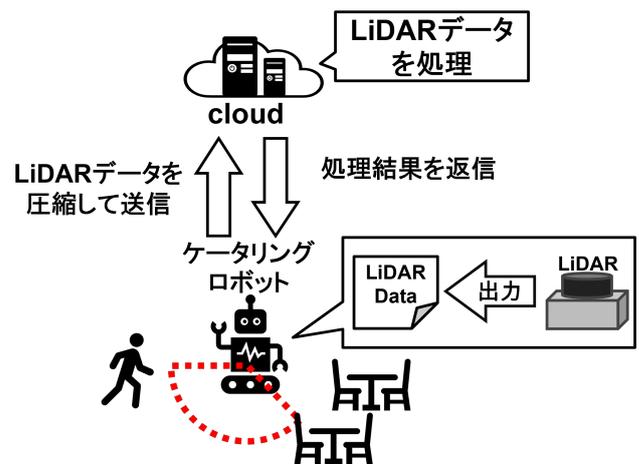


図 6: ユースケース

ボットには障害物を検知して回避するためにLiDARを搭載する。モニタリングやLiDARデータを処理して回避の判断をするためにクラウドにデータを送信することを想定する。また、クラウドではLiDARデータを受け取り処理した結果をロボットに返す。LiDARは短時間で大量のデータを生成することができる。しかし、データ量が大きいと送信やデータの処理に時間がかかり、障害物検知や回避が遅れる。本提案を適用することにより、大量に生成されたLiDARデータが圧縮され通信にかかる時間が少なくなる。これにより、リアルタイムで障害物検知や回避が可能になる。

## 4. 実装

圧縮を行うソフトウェア「compression」と解凍を行うソフトウェア「decompression」はPythonを使用して実装した。図7に実装したソフトウェアの概要を示す。このソ

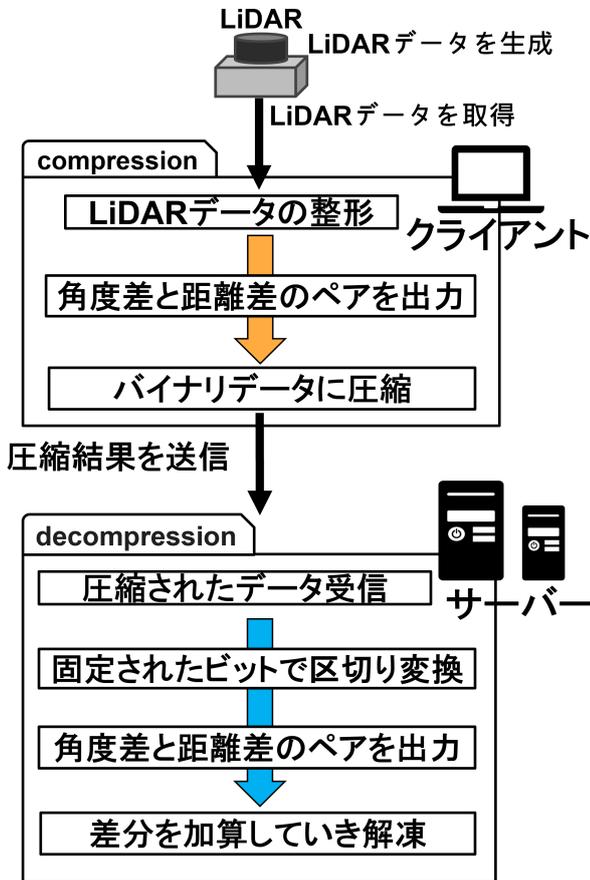


図7: 提案手法の実装

フトウェアでは、LiDAR データを読み取り、「theta」と「dist」の値のみを取得し、ポイント間の角度差を計算する。計算した角度差と「dist」のペアを1行で出力する。また、読み取る LiDAR データは RPLiDAR C1 を用いて取得する。LiDAR を使用するにあたって SLAMTEC 社の SDK を利用した\*2。

## 5. 評価実験

評価実験として、LiDAR 本体が動いた際のデータや周りの環境が変動するデータ、環境が変動しないデータを用いて、圧縮を行う。圧縮率や送信して解凍するまでの時間を計測し評価する。加えて、LiDAR から出力される raw データを圧縮せずに送る場合と提案手法で圧縮したデータを送った場合で、送信時間を計測し評価を行う。

\*2 [https://github.com/Slamtec/rplidar\\_sdk](https://github.com/Slamtec/rplidar_sdk)

## 実験環境

実験では、以下の環境で実施する。

### • LiDAR

SLAMTEC 社の RPLidar C1 を使用し、LiDAR が動いた場合のデータ、周りの環境が変動するデータ、環境が変動しないデータを取得する。

### • サーバー

送信先のサーバーには、Ubuntu24.04LTS をインストールした仮想マシンを用意する。受信したデータを保存するデータベースには MongoDB を用いる。

## 基礎実験

LiDAR から出力された環境データを使用する。図8に基礎実験の概要を示す。

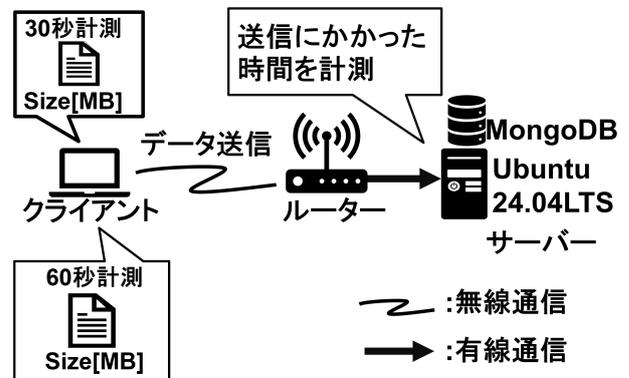


図8: 基礎実験の概要

基礎実験では、以下の環境で実施する。

### • LiDAR

SLAMTEC 社の RPLidar C1 を使用し、周囲の環境データを取得する。データは 30[s] 計測したデータと 60[s] 計測したデータを用意する。

### • クライアント

Dell 社の「Inspiron 5402」を用いて LiDAR が出力したデータをサーバーに送信する。データの送信には、Wi-Fi を使用する。

### • Wi-Fi ルーター

ASUS 社の「ASUS TUF-AX5400」を用いる。Wi-Fi ルーターとサーバーは 1[Gbps] のケーブルで有線接続されている。

### • サーバー

送信先のサーバーには、Ubuntu24.04LTS をインストールした仮想マシンを用意する。受信したデータを保存するデータベースには MongoDB を用いる。

実験目的は、raw データのサイズや実際に遅延するかを確認するため実験を行う。30[s] 計測したデータと 60[s] 計測したデータを記録する。それに加えて、取得した2つの LiDAR データをサーバーに送信し、送りきるまでの時間

を計測する。実験は各データ 10 回送信する。

## 実験結果

実験の結果を表 2 に示す。

表 2: 各時間の計測したデータサイズ

|        | 30[s] 計測したデータ | 60[s] 計測したデータ |
|--------|---------------|---------------|
| データサイズ | 5.1[MB]       | 11[MB]        |

30[s] 計測したデータのデータサイズは約 5.1[MB] となり、60[s] 計測したデータのデータサイズは約 11[MB] となった。

送信時間を計測した結果をヒストグラムにし、図 9 に示す。

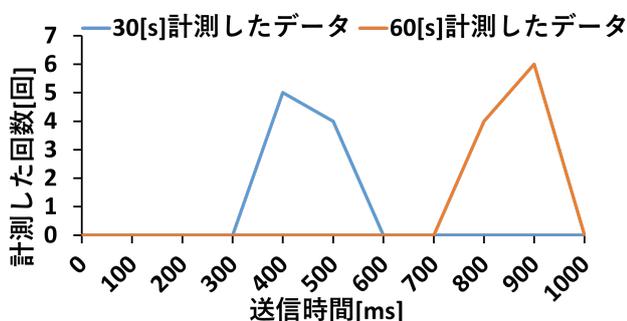


図 9: 計測したデータの送信時間

30[s] 計測したデータの送信時間の最頻値は約 440[ms] となった。次に、60[s] 計測したデータの送信時間の最頻値は約 960[ms] となった。

## 6. 議論

本提案では差分データをバイナリに変換する際に、固定長ビットを使用している。固定長にすることで、データの読み書きや処理が単純化される。しかし、全てのデータを同じビット長で表すため、実際に必要なビット数よりも多くのビットを使用してしまう。これにより、データ量が増加してしまう。バイナリデータは 0 と 1 のみで表される。そのため 0 や 1 が連続する箇所が発生する。RLE 圧縮を用いることにより、連続した値を少ないデータ量で表すことができる。

センシングする環境によって、スキャンデータの密度や分布が変化する。この変化により、角度差と距離差の値が増減する。例えば、物体が密集した環境では、角度差と距離差は小さくなるため圧縮効率が向上する。一方で、物体が散在した環境では、角度差と距離差の値は大きくなるため圧縮率が低下すると考えられる。すでに送信したデータと比較して変更点があった部分のみ送信することにより環境に左右されずに圧縮することができる。

## 7. おわりに

飲食業界では人手不足や労働生産性向上に向けて、サービス分野におけるロボットの活用が進んでいる。ロボットには障害物や人の検知のために LiDAR と呼ばれるセンサーを用いる。LiDAR は短時間で大量のデータを出力する。課題は、通信帯域を圧迫してしまい、それにより送信時間が遅くなり全体的な処理が遅くなってしまうことである。基礎実験として、30[s] 計測したデータと 60[s] 計測したデータのサイズを記録し、サーバーに送信完了するまでの時間を計測した。その結果、30[s] 計測したデータではサイズが約 5.1[MB] で送信時間は約 440[ms] となり、60[s] 計測したデータではサイズが約 11[MB] で送信時間は約 960[ms] であった。課題を解決するために、本稿では LiDAR データのポイント間の差分を用いた可逆圧縮手法を提案する。評価は LiDAR が移動した際のデータや周りの環境が変動するデータを圧縮し圧縮率や送信完了するまでの時間を評価する。

## 参考文献

- [1] Guo, P., Shi, H., Wang, S., Tang, L. and Wang, Z.: An ROS Architecture for Autonomous Mobile Robots with UCAR Platforms in Smart Restaurants, *Machines*, Vol. 10, No. 10 (online), DOI: 10.3390/machines10100844 (2022).
- [2] Tallava, jhula, A.: Lidar Simulation for Robotic Application Development: Modeling and Evaluation., PhD Thesis, Carnegie Mellon University, USA (2018).
- [3] Shimmura, T., Ichikari, R., Okuma, T., Ito, H., Okada, K. and Nonaka, T.: Service robot introduction to a restaurant enhances both labor productivity and service quality, *Procedia CIRP*, Vol. 88, pp. 589–594 (online), DOI: <https://doi.org/10.1016/j.procir.2020.05.103> (2020). 13th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 17-19 July 2019, Gulf of Naples, Italy.
- [4] Moshayedi, A. J., Roy, A. S., Liao, L., Khan, A. S., Kolahdoz, A. and Eftekhari, A.: Design and Development of FOODIEBOT Robot: From Simulation to Design, *IEEE Access*, Vol. 12, pp. 36148–36172 (online), DOI: 10.1109/ACCESS.2024.3355278 (2024).
- [5] Garcia-Haro, J. M., Oña, E. D., Hernandez-Vicen, J., Martinez, S. and Balaguer, C.: Service Robots in Catering Applications: A Review and Future Challenges, *Electronics*, Vol. 10, No. 1 (online), DOI: 10.3390/electronics10010047 (2021).
- [6] Qaisar, M. W., Shakeel, M. M., Kedzia, K., Machado, J. M. and Jan, A. Z.: Localization-based waiter robot for dynamic environment using Internet of Things, *International Journal of Information Technology*, (online), DOI: 10.1007/s41870-023-01723-8 (2024). In press.
- [7] Piazza, E., Lima, P. U. and Matteucci, M.: Performance Models in Robotics With a Use Case on SLAM, *IEEE Robotics and Automation Letters*, Vol. 7, No. 2, pp. 4646–4653 (online), DOI: 10.1109/LRA.2022.3151162 (2022).
- [8] Jeyabal, S., Sachinathana, W. K. R., Bhagya, S. M., Samarakoon, P., Elara, M. R. and Sheu, B. J.: Hard-to-

- Detect Obstacle Mapping by Fusing LIDAR and Depth Camera, *IEEE Sensors Journal*, pp. 1–1 (online), DOI: 10.1109/JSEN.2024.3409623 (2024).
- [9] Zhang, X., Gao, W. and Liu, S.: Linear model based geometry coding for LiDAR acquired point clouds, *arXiv preprint arXiv:2001.03871* (2020).
- [10] Sato, K., Shinkuma, R., Sato, T., Oki, E., Iwai, T., Kanetomo, D. and Satoda, K.: Prioritized Transmission Control of Point Cloud Data Obtained by LIDAR Devices, *IEEE Access*, Vol. 8, pp. 113779–113789 (online), DOI: 10.1109/ACCESS.2020.3003753 (2020).
- [11] Zhang, R., Li, S., Lu, W., Sun, J., Zhang, Y. and Ge, W.: Coherent heterodyne FMCW lidar based on combined single/double sideband modulation detection technology, *Infrared Physics Technology*, Vol. 137, p. 105181 (online), DOI: <https://doi.org/10.1016/j.infrared.2024.105181> (2024).
- [12] Bui, G., Calyam, P., Morago, B., Antequera, R. B., Nguyen, T. and Duan, Y.: LIDAR-based virtual environment study for disaster response scenarios, *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 790–793 (online), DOI: 10.1109/INM.2015.7140377 (2015).
- [13] Roriz, R., Silva, H., Dias, F. and Gomes, T.: A Survey on Data Compression Techniques for Automotive LiDAR Point Clouds, *Sensors*, Vol. 24, No. 10 (online), DOI: 10.3390/s24103185 (2024).
- [14] Isenburg, M.: LASzip: lossless compression of LiDAR data, *Photogrammetric engineering and remote sensing*, Vol. 79, No. 2, pp. 209–217 (2013).
- [15] Rusu, R. B. and Cousins, S.: 3D is here: Point Cloud Library (PCL), *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4 (online), DOI: 10.1109/ICRA.2011.5980567 (2011).
- [16] Graziosi, D., Nakagami, O., Kuma, S., Zaghetto, A., Suzuki, T. and Tabatabai, A.: An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC), *APSIPA Transactions on Signal and Information Processing*, Vol. 9, p. e13 (online), DOI: 10.1017/AT-SIP.2020.12 (2020).
- [17] Otsu, R., Shinkuma, R., Sato, T. and Oki, E.: Data-Importance-Aware Bandwidth-Allocation Scheme for Point-Cloud Transmission in Multiple LIDAR Sensors, *IEEE Access*, Vol. 9, pp. 65150–65161 (online), DOI: 10.1109/ACCESS.2021.3075275 (2021).
- [18] Ojanperä, T., Mäkelä, J., Majanen, M., Mämmelä, O., Martikainen, O. and Väisänen, J.: Evaluation of LiDAR data processing at the mobile network edge for connected vehicles, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2021, No. 1, p. 96 (online), DOI: 10.1186/s13638-021-01975-7 (2021).
- [19] Houshiar, H. and Nüchter, A.: 3D point cloud compression using conventional image compression for efficient data transmission, *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*, pp. 1–8 (online), DOI: 10.1109/ICAT.2015.7340499 (2015).
- [20] Dubé, R., Dugas, D., Stumm, E., Nieto, J., Siegwart, R. and Cadena, C.: SegMatch: Segment based place recognition in 3D point clouds, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272 (online), DOI: 10.1109/ICRA.2017.7989618 (2017).
- [21] Kammerl, J., Blodow, N., Rusu, R. B., Gedikli, S., Beetz, M. and Steinbach, E.: Real-time compression of point cloud streams, *2012 IEEE International Conference on Robotics and Automation*, pp. 778–785 (online), DOI: 10.1109/ICRA.2012.6224647 (2012).
- [22] Moreno, C., Chen, Y. and Li, M.: A dynamic compression technique for streaming kinect-based Point Cloud data, *2017 International Conference on Computing, Networking and Communications (ICNC)*, pp. 550–555 (online), DOI: 10.1109/ICCNC.2017.7876188 (2017).
- [23] Mekuria, R., Blom, K. and Cesar, P.: Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 27, No. 4, pp. 828–842 (online), DOI: 10.1109/TCSVT.2016.2543039 (2017).
- [24] Tu, C., Takeuchi, E., Miyajima, C. and Takeda, K.: Compressing continuous point cloud data using image compression methods, *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1712–1719 (online), DOI: 10.1109/ITSC.2016.7795789 (2016).