

# あの日あの時あの場所へ：任意の時刻・場所の点群データを作る ための圧縮された点群の効率的マージ手法

渡邊真由子<sup>1</sup> 五十嵐勇<sup>1</sup> 谷田隆一<sup>1</sup> 木全英明<sup>1</sup>

**概要：**近年、ビルやインフラ設備などの構造物をレーザーの反射光を用いて3次元の座標と属性を持つ点の集合である点群データとして計測し、設備保守に利用することが検討されている。点群データを利用する際は対象とする時刻の特定の領域の点だけを使う必要がある。しかし、点群データは利用時の領域と無関係な範囲で様々なタイミングで取得された複数のデータの集合として集積されており、さらにデータ量が膨大であるため圧縮して保管されている。そのため、必要な点だけを効率よく取り出すのは容易ではない。本稿では、幅広い領域かつ膨大なデータ数を持ちうる点群データの解析処理を行いやすくするため、任意の時刻・場所の点群データを作るための圧縮された点群の効率的マージ手法を提案する。実験では、圧縮した古い点群データを新データで効率的に上書きするマージ手法の有効性を確認し、またここで用いた圧縮手法について既存技術に対する優位性を確認した。

## 1. はじめに

近年、レーザーの反射光から3次元の座標と属性を持つ点の集合である点群データを計測する Light Detection and Ranging (以下、LiDAR) などを用い、ビルや電信柱などの建築物の形状を計測・推定して劣化度合いや傾きなどの状態判定をすることが検討されている。これらの検討に関連し、計測位置を固定した LiDAR による単独のビルやインフラ設備の点群データの計測のみならず、車やドローンなどの移動体に LiDAR を搭載して計測する Mobile Mapping System (以下、MMS) や空撮 LiDAR などによる街全体の点群データの計測が進んでいる。点群データの計測機器の増加や点群利用のニーズの拡大に伴い、データの計測範囲の拡大や計測回数の増加などデータ量の増加が想定される。

LiDAR で計測する点群データはレーザーの反射波をもとに各点の3次元空間座標や属性(反射強度など)を計測したものであり、主に LASer (以下、LAS) 形式[1]で保存する。LAS 形式のデータ量は、点の数に応じて増加し、例えば点が30万個で約10MBになる。

LiDAR で点を計測する際は、計測区間のデータを全て1ファイルに取めるとファイルサイズが巨大になり処理に困るため細かく分割して保存される。その結果ファイル数が膨大になる。更に計測範囲が拡大し、計測する点数が拡大すると、ファイル数は更に増大する。尚、LAS データは一般的に点の計測順に各点の座標・属性を格納する。また、計測したデータを格納したファイルはファイルごとに計測領域が区切られず、複数のファイル間で点の範囲が重複する場合もある。ファイル数が膨大かつファイル間で各データの範囲が区切られず重複するようなデータは解析処理に適していない。

これに対し、従来、計測したデータを重ね合わせ空間内

を格子状にメッシュ分割して切り分けてファイルに保存する手法がある。以降、表現をわかりやすくするため、計測したデータを計測データ、計測データを重ね合わせメッシュ分割して切り分けたデータをメッシュ化マージ済データと呼ぶ。日時毎のメッシュ化マージ済データを生成して解析に用いることで設備保守のためのデータ解析が容易になる。しかし、異なる時刻の計測データを単純に重ね合わせてメッシュ化マージ済データを生成すると様々な時刻の点が混ざり正しい構造物の状態を推定できない、メッシュ化マージ済データを事前生成するとデータ量が増加するなどの問題がある。

本研究はこの問題解決を目指すものである。

### 1.1 本研究の貢献

本稿では、都市空間内の構造物を対象として計測する点群データに対し、任意の時刻・場所の点群データを作るため、圧縮されたデータを「空間内でメッシュ分割して切り分ける処理」(メッシュ処理)及び「重ね合わせる処理」(マージ処理)を提案する。具体的には、点群符号化処理で各計測データを共通のメッシュで分割して圧縮保存し、解析処理等で必要な領域を読み込む際に計測データの重複領域を考慮したマージ処理をしてメッシュ化マージ済データを生成する。

### 1.2 本稿の構成

次の2節では、点群の利用シーンを想定した場合の課題を述べ、3節では本提案手法に含まれる点群符号化技術について、既存の技術と本提案で用いる技術を紹介する。4節では提案手法を説明し、5節で実験内容と結果を述べる。実験では、点群データをマージできることを確認し、また圧縮したデータは、従来技術 LASzip[2]で圧縮した場合に比べデータ量が削減されることを確認した。

## 2. 課題

この節では、点群データの利用シーンを想定して課題を明らかにする。まず本研究で想定する備保守向けの点群デ

<sup>1</sup> 日本電信電話株式会社  
メディアインテリジェンス研究所

ータ計測・解析処理を前提として、3つのユースケースと課題を紹介する。その後従来行われてきた対策を紹介し、その課題について述べる。

### 2.1 ケース 1:一部重複するルートを同一日に複数回計測

一部重複するルートを同一日に LiDAR で複数回計測するケースを考える。複数の計測データを同一日に計測する場合、対象の形状は一定であるので、重複領域のデータをそのまま重ね合わせればよい。これをメッシュ単位で区切ることでメッシュ化マージ済データを生成できる。

### 2.2 ケース 2:一部重複するルートを期間を空け複数回計測

ケース 1 で期間を空けて計測する場合を考える。データを異なる日に計測すると、単純な重ね合わせでは解析処理に適したメッシュ化マージ済データを生成できない。新設・改築・立て壊しなどにより該当領域の状況が変化した場合、点群を重複領域もそのまま重ね合わせると正しい状況が不明確になるためである。このため、撮影日が異なる新旧の2データが重複する領域に対して、より計測日時が新しい新データで旧データを上書きすることが必要である(図 1)。これにより、新データが計測した領域は新データで上書きされ、旧データしかない領域は旧データが残り、各領域について、最も新しい計測データのみからなるデータを作成できる。また、データの撮影エリアの拡大や計測回数の増加につれ、データ量も増加していくため、点群データを効率的に圧縮することも必要となる。

### 2.3 ケース 3:街全体から収集した計測データから任意の時刻のメッシュ化マージ済データを生成

街全体から収集した計測データから任意の時刻のメッシュ化マージ済データを生成するケースを考える(図 2)。任意の時刻のメッシュ化マージ済データを生成するには、その時刻までの計測データを新データで上書きしながらマージすればよい。ただし、街全体の計測データから生成したメッシュ化マージ済データは、データ量が大きくなると想定され、全時刻分保存すると、計測データのみ保存する場合に比べデータ量が膨大になると考えられる。これらをそれぞれ圧縮保存すればデータ量削減が可能であるが、計測データのみを圧縮した場合に比べやはりデータ量は大きい。

### 2.4 従来の対応方法とその課題

従来の 2.1~2.3 のケースへの対応方法とその課題を述べる。従来は、データ計測後、計測データのメッシュ化処理・マージ処理を手動で実施してメッシュ化マージ済データを生成し、これらを共に圧縮保存している。しかし、この場合データ量が増加する。メッシュ化マージ済データを生成しない場合は、解析処理の際に都度メッシュ化マージ済データの生成処理を行うため、事前の処理が膨大になる。

圧縮保存した計測データから必要な領域のデータのみ取り出し、更にそのマージ処理が簡易に行えるならば、メ

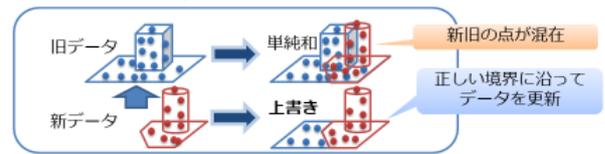


図 1. 複数の計測データの単純な重ね合わせ (右上) と重複領域を上書きする重ね合わせ (右下)

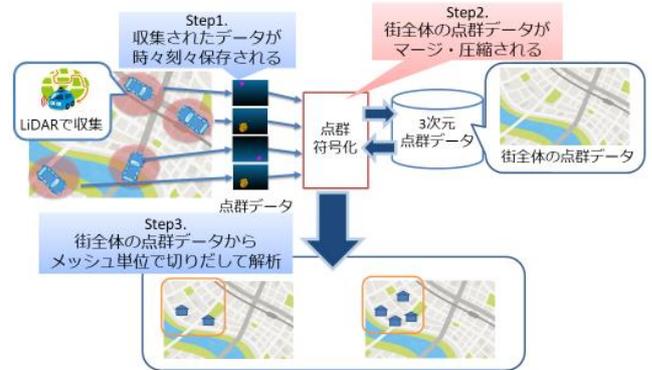


図 2. LiDAR を用いたデータ計測・蓄積

ッシュ化マージ済データの保存は不要となる。我々は、このような一般的な LiDAR データ蓄積・圧縮システムを、簡易に実現する機能を提案する。

## 3. 点群符号化技術

3 節では提案手法に含まれる点群符号化技術について、既存の圧縮技術である LASzip と本提案で用いる MPEG の Point Cloud Compression (以下、PCC) [3]について説明する。

### 3.1 LASzip

LASzip は業界標準として用いられている LAS データの圧縮方式である。LASzip では、点群データを入力順に読み込み、その順に点と点の座標の差分から点の間隔を予測し、差分を符号化することで座標の冗長性を削減する。LASzip では、LAS データを入力順に圧縮する。そのため、圧縮データの部分的な領域のデータを抜き出す場合、全てのデータを復号し、全ての点を検索する必要があり、2.4 で述べたような部分的な領域のデータの取り出しには適さない。

### 3.2 PCC

PCC は MPEG で検討されている点群データを圧縮するための方式群である。PCC は、地図データや美術品などの静的なデータを対象とする Geometry based Point Cloud Compression (以下、GPCC)、人物などの動的なデータを対象とする Video based Point Cloud Compression (以下、VPCC) の 2 種類の方式にわけて検討されている。本研究が対象とする都市の状況や様々な構造物を計測した点群データの圧縮には VPCC より GPCC が適当であるため、以下では GPCC について述べる。

#### 3.2.1 GPCC

GPCC は、LASzip のような LiDAR で計測したデータに

特化した圧縮方式ではなく、3次元の座標と属性をもつような任意の条件で計測されたデータに対する圧縮方式として検討されている。特徴は、点群データを空間を $2 \times 2 \times 2$ の8個のブロックに再帰的に分割する8分木構造の葉の集合として表現することである。各点の情報は座標の代わりに再帰的に8分割したブロックの内部のどこに存在するか、で表現されるため点がブロック内の一部の領域に固まっているような場合には根からその領域に至る節点の情報は重複して持つ必要がなく、冗長性を削減できる。さらに、ブロックごとに符号化・復号できるため、ある領域のデータを抜き出す場合はその領域を含むブロックのみ復号し、マージ処理することが可能である。GPCCでは上記のブロックをタイルと呼ぶ。

### 3.2.2 GPCCによる圧縮データ

圧縮データは、ヘッダ情報とデータに分かれて格納される。ヘッダ情報には符号化範囲など、タイルのヘッダ情報としてタイルの範囲など、データとして座標の情報などが格納される。符号化範囲とは圧縮される範囲を意味する。ここで、GPCCでは符号化範囲の算出方法は規定されていないが、テストモデルTMC13[4]のエンコーダでは符号化範囲の定め方は以下のように、符号化範囲のサイズが最小となるように設定されている。

始点の座標  $(X0, Y0, Z0) = (\min x, \min y, \min z)$

終点の座標  $(X1, Y1, Z1) = (\max x, \max y, \max z)$

符号化範囲のサイズ

$$(\text{sizeX}, \text{sizeY}, \text{sizeZ}) = (X1 - X0, Y1 - Y0, Z1 - Z0)$$

(但し、上記の  $x, y, z$  は入力された点群の  $x, y, z$  座標)

尚、タイルについては始点、サイズをタイルごとに格納するが、テストモデルTMC13のエンコーダでは点群が存在する領域に対してタイルが設定される。このため、通常のタイルの間隔は等間隔ではなく、格子状に並んでいるわけではない。

### 3.2.3 座標の8分木構造

次に、3次元座標の8分木構造を用いた表現を説明する。図3左のブロックのように、点群を1つのブロックで覆い、そのブロックをxyzの3軸に沿ってそれぞれ2分割、つまり8個のサブブロックに分割する。各々のサブブロック内の点の有無(占有状態)を1と0で表現し、1のサブブロックを更に8分割する。これを繰り返す。8つに分割されたサブブロックそれぞれの占有状態は、あるノードに対する子ノードとして図3の右の木構造のように再帰的に表すことができる。これを繰り返すことで点の座標を8分木構造として表現することが可能である。このように座標を8分木構造として表現することで、特に空間内の一部に局所的に点が存在する場合、それらの座標の冗長性を削減でき、効率よく圧縮できる。尚、この8分木構造によりGPCCでは途中階層までの符号化・復号が可能、つまり非可逆な圧縮も可能だが、本稿では可逆圧縮に対する検討を行う。

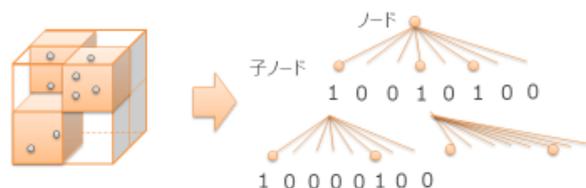


図3. 座標の8分木表現

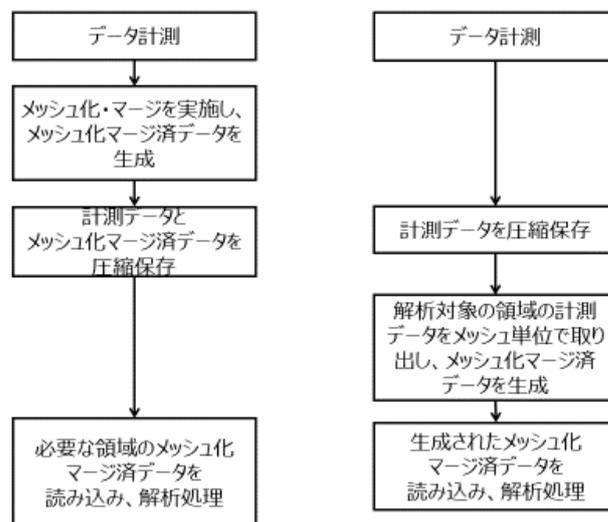


図4. 従来手法(左)と提案手法(右)の処理比較

## 4. 提案手法

2節で述べたケースを想定し、圧縮保存した点群データから特定の領域のデータのみ取り出し、マージ処理を効率的に行う点群データ復元方法を提案する。具体的には、メッシュ単位で圧縮した新旧の計測データに対し、旧データを新データで上書きしたメッシュ化マージ済データを生成する。

図4に従来手法と提案手法の処理フローの違いを示す。提案手法では計測データを圧縮保存した後、解析処理時に必要な領域のデータのみ取り出してマージ処理するため、メッシュ化マージ済データの保存が不要になる。従来はLASzipによるLASデータ圧縮が一般的でありLASzipの特性から、事前にメッシュ化マージ済データを生成する必要がある。提案手法ではデータをGPCCで圧縮することによりメッシュ分割をして切り分けることを可能にし、さらにGPCCで圧縮した座標の8分木構造を用いた階層的なマージ処理により、処理コストを抑えたメッシュ化マージ済データ生成を実現する。

### 4.1 全体の処理

全体の処理として、点群符号化及びマージ処理の手順を述べる。まず圧縮処理は、新旧の計測データをGPCCのタイルによりどちらも共通のメッシュ単位となるよう処理する。位置が共通のタイル内の8分木構造において、同じノ

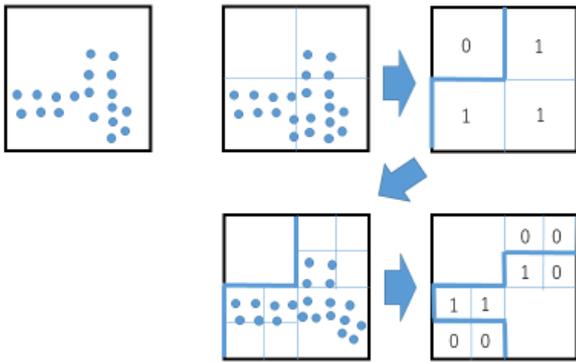


図 5. 新データの領域の削りだし方法（左側から削りだす場合）

ードは同じ場所を表し、ノードの交換によりデータの置き換えが可能になる。次に、新データの 8 分木を xy 平面に射影して 4 分木構造で表現し、タイル単位で次のようなマージ処理を行う。

1. 新データの領域を特定
2. 旧データのうち 1 で特定した領域を新データに置き換え

上記の 1 は、新データを含む領域全体からデータのない領域を削る方針で実現しており、その削り方を 4.2 節で説明する。

#### 4.2 点群データマージ処理

ここではマージ処理について、説明を簡単にするため、新データの領域を左側から削る例を示す。新データを真上から見下ろした時の点の分布が図 5 左上のような場合を想定する。このブロックを 4 分割すると占有状態は、左上のみ 0、残りは 1 となる。この時に新データの占める領域を左側から削ると図 5 右上のようになる。更に、残った領域内で、左側の境界に接するサブブロックを階層的に 4 分割し、左側から削る。ここで削られる領域は、あくまでこのサブブロック内に閉じ、右隣に隣接するサブブロックは削られない。これを繰り返すことで、新データが存在する領域を階層的に削りだすことができる。

提案手法においては、この処理をタイルを囲む 4 辺それぞれから行い、新データの領域は、削りだされた領域の積集合として表現する。ブロック内部を全て再帰的に 4 分割し、新データの領域を分割されたブロック単位で行うと、分割が細かくなると点群の分散性から新データの領域が分散的になり、旧データが本来不要な領域に残る可能性がある。しかし、上記の手法では上の階層から順に判定するため、境界に接するブロックのみ細分化されたサブブロック毎に削ることができ、新データに囲まれた領域に旧データが含まれることを防ぐことができる。また本手法では 1 列あたりの最小サブブロックの数に対して線形の計算量で演算が実現できて効率的である。

尚、境界判定を終了する階層は、対象とする点群の計測

精度、タイルのサイズに応じて変更可能である。

#### 4.3 提案手法のメリット

通常、境界を正しく判定するには、点同士の関係や点が表す物体などを明らかにする必要がある。例えばセグメンテーションなどで推定は可能であるが、処理時間がかかる。提案手法では点をブロックとしてマージ処理するため、点が表す物体の推定等を行うことなく高速に境界判定し、メッシュ化マージ済データを生成できる。

### 5. 評価実験

#### 5.1 実験目的

提案手法による圧縮された計測データの重ね合わせの結果、及びその際の圧縮率への影響を実験により確認する。

#### 5.2 実験条件

実験条件は以下の通りである。

- ・実験データ：新旧 2 データ（詳細は下記）
- ・点数：新旧データそれぞれ 300 万点
- ・データの種類：座標のみ、属性なし
- ・データ計測精度（単位）：0.001[m]
- ・入力形式：LAS→PLY（ascii 形式）に変換
- ・符号化範囲の原点：(0, 0, 0)
- ・タイルサイズ：32,768×32,768
- ・マージの階層数：5～10

##### 5.2.1 実験用ソフトウェア

計測データの圧縮には GPCC のテストモデル TMC13 を使い、エンコーダにおいて以下の修正を行った。タイル共通化のため、符号化範囲の原点から指定された間隔でタイルを敷き詰めるよう修正し、原点・タイルサイズを共通で設定することにより一定のタイルサイズを格子のサイズとするメッシュ分割エンコード可能にした。これにより、本来より符号化範囲が拡大し圧縮時のデータ量が増える恐れがあるため、LASzip で圧縮した場合と比較し、データ量への影響を確認する。

##### 5.2.2 実験データ

今回の実験では、街中で MMS により計測した 2 つの LAS データを用いた。尚、本稿の実験に用いた点群データは株式会社アジア航測が撮影したものである。図 6 は、各データを真上から見下ろしたものである。このデータはある地点を撮影後、再度その一部が重複するよう隣接地点を撮影したものである。図 6 では区別しやすくするためそれぞれ別の色で着色してあるが、今回の実験では座標のみのデータを圧縮対象とする。尚、今回は簡単のため上記のデータの撮影期間は開けずに撮影を行った。また、計測時のデータは連続して撮影したものを 30 万点毎に保存するため、このファイルを 10 個連結したものをそれぞれ新データ、旧データとした。

また、TMC13 の入力形式は PLY であるため、LAS から PLY への変換を行った。PLY は、3 次元データを格納する

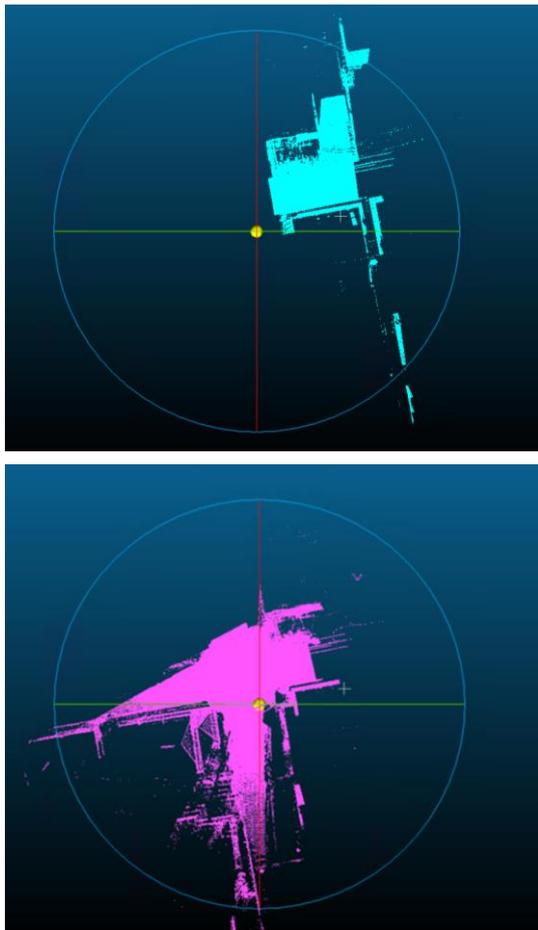


図 6. 実験データ（上：新データ，下：旧データ）

ための形式であり，LAS と同様に点の座標，属性を格納することが可能である．大きな違いとして，LAS ではデータのヘッダ情報で座標のスケールとオフセットを定義し座標は整数値のみ格納するが，PLY ではスケールとオフセット処理済みの座標を格納するため，小数値も含まれる．今回，スケール処理による座標の桁落ちを防ぐため，LAS のスケール，オフセットを除いた整数値のみの変換とした．LAS のヘッダのスケールが計測精度と同じ 0.001 であるため，実質的な点の精度に影響はない．LAS のスケール，オフセットは PLY には残らないが，変換時にヘッダファイル（データ量約 1[KB]）として切り出すことにより，PLY から LAS に再変換する際に戻すことも可能である．

### 5.2.3 実験内容

今回の実験では，圧縮された計測データのマージ処理について，新データで旧データを上書きできるかの確認に加え，境界を判定する階層をいくつかに分けた場合の影響を確認した．タイルサイズは，PLY 上での整数値座標に対するサイズであり，実寸上は約 32m 四方の領域で符号化単位を分けている．ここで判定階層が  $N$  とは， $32.768/2^N$  m 単位のブロックまで境界を判定するという意味である．例えば，判定階層が 10 であれば 3.2cm 四方のブロックまで境界判定

を行っている．

また，提案手法における圧縮性能を確認するため，新旧の計測した LAS データに対し，LAS→LASzip と LAS→PLY →提案方式（TMC13 ベース）による圧縮後のデータ量を比較した．また，提案方式で生成されたマージデータを圧縮した際のデータ量及びこのマージデータを再度 LASzip で圧縮した場合のデータ量の比較も行った．

### 5.3 実験結果・考察

図 7 はマージ結果を真上から見た図である．旧データを新データでほぼ上書き可能なことを確認できた．但し，判定階層が 5 のように浅い（値が小さい）場合，特に境界が斜めのときに削除される境界が粗くなる．判定階層 10 のように増やすとより細かな精度で境界判定を行うため，スムーズになることが確認できる．（図 8）

表 1 は新旧データ，及びマージデータの圧縮後のデータ量である．いずれにおいても提案手法でデータ量が抑えられることを確認した．尚，新旧データに対し，マージデータでは点が一部削減されるため，それらの和よりデータ量は減少する．また，マージデータでは，新旧データに比べ

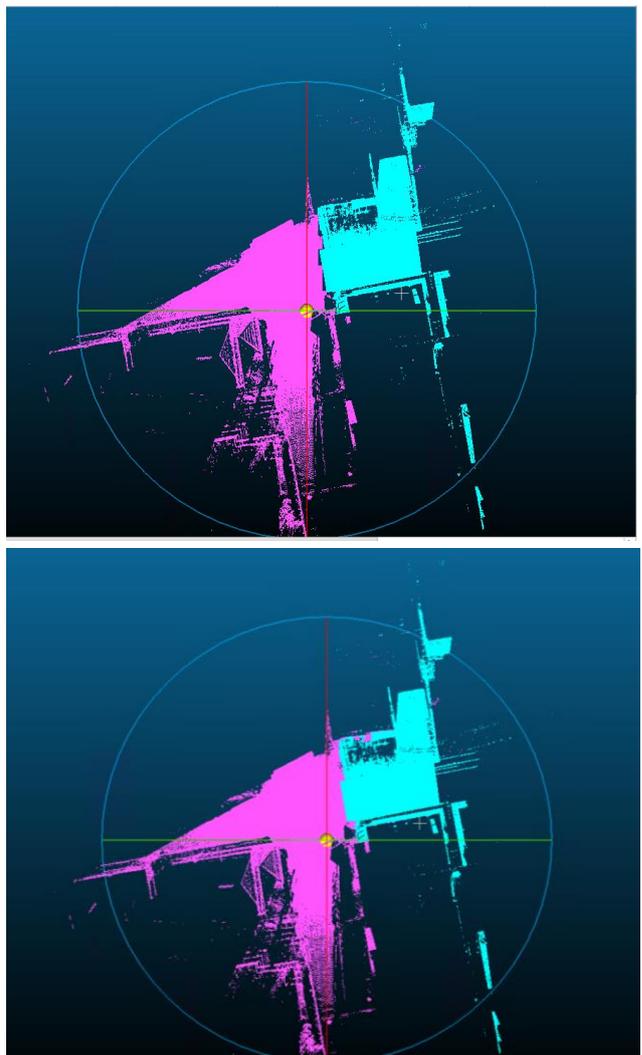


図 7. マージ結果（上：判定階層 5，下：判定階層 10）

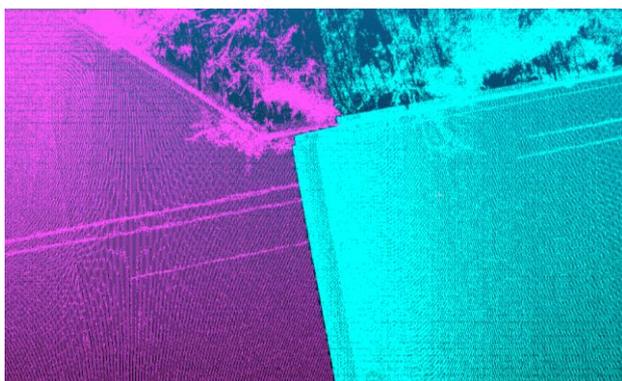


図 8. 境界領域の拡大結果 (判定階層 : 10)

表 1. 圧縮後のデータ量[KB]

	提案手法 (TMC13 ベース)	LASzip
新データ (300 万点)	3, 335	3, 690
旧データ (300 万点)	3, 585	4, 181
マージデータ (4, 984, 032 点)	5, 689	7, 807

LASzip に対する圧縮性能が向上している. この理由は以下であると考えられる. LASzip で LiDAR が一定間隔で計測した点群データを圧縮した場合, 座標の予測精度が高まる. しかし, 本実験で作成したマージデータのように複数ファイルの LAS データが混在する場合, 点の間隔が計測時と異なるため, 予測精度が低下し圧縮性能も低下したと想定される.

## 6. まとめ

本稿では, 都市空間内の構造物から計測され, 幅広い領域かつ膨大なデータ数を持ちうる点群データに対し, 任意の時刻・場所の点群データを作るための圧縮された点群の効率的マージ手法を提案した. 新旧 2 つの点群データが圧縮されるとき, 圧縮した古い点群データを圧縮した新データで簡易に上書きするマージ手法の有効性及びその圧縮性能の既存技術に対する優位性を確認した.

## 参考文献

- [1] ASPRS LAS 1.4 Format Specification R15 July 9 2019
- [2] M. Isenburg, "LASzip: Lossless compression of LiDAR data," Photogrammetric Eng. Remote Sens., vol. 79, no. 2, pp. 209–217, Feb. 2013.
- [3] S. Schwarz et al., "Emerging MPEG standards for point cloud compression," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 133–148, Mar. 2019.
- [4] <https://github.com/MPEGGroup/mpeg-pcc-tmc13>