

3次元点群データの3D Tiles化とWeb配信のパフォーマンス評価

柳下 大*・嘉山 陽一**

3D point cloud data conversion to 3D Tiles and Web delivery performance evaluation

Dai Yagishita*, Yoichi Kayama**

3D point cloud data has come to be used in GIS for landscape expression, simulation and others. The point cloud data with high measurement density enables a precise representation of the features. However, the large volume of 3D point cloud data is unsuitable for handling as it is because the WEB distribution is affected by communication. As a mechanism to solve this, there is a technology called Level of Detail (LOD) in the 3D Tiles format. This is a mechanism for smoothing the display by controlling the number of points to be read according to the distance from the viewpoint. This time, using this LOD technology, the 3D point cloud data of the whole 23 wards of Tokyo was converted into 3D Tiles, and it was verified to display it on a Web browser. This article describes the procedure for creating LOD data and the evaluation of display performance for each library of Web 3D rendering.

Keywords: 点群(Point Cloud), WEB, 3DTiles, Level of Detail (LOD)

1. はじめに

3次元点群データは、景観の表現やシミュレーション用途などでGISでも活用されるようになってきた。3次元点群データは計測密度が高いと地物の精細な表現が可能となるが、データの容量増大に比例して読み込みや表示のパフォーマンスに影響するという課題がある。特にWebで取り扱う場合は通信を介するため影響が大きく、大容量の3次元点群データをそのまま扱うことは不向きである。この課題を解決する仕組みとして、Level of Detail (LOD)という技術がある。視点からの距離に応じて読み込む点数を制御することで、表示をスムーズにする仕組みである。Webの3次元データの標準的なフォーマットの一つである3D Tilesでは、データを階層化することでLODを実現している。今回このLOD技術を用い、東京23区全域の3次元点群データを3D Tiles化し、Webブラウザで表示する検証を行った。本稿ではLODのためのデータ階層化の手順と、Web 3Dレンダリングのライブラリごとの表示パフォーマンスの評価について解説する。

2. 3次元点群データについて

2.1. 3次元点群データの諸元

検証に用いた3次元点群データは、GPSとIMU(慣性計測装置)で位置を把握している航空機からレーザを発射し、3次元形状を取得したデータである。データのモデルは土地や建物・植生を含めた表面を含んだDSM(Digital Surface Model)で、計測した状態と同様の不規則に配置されているランダムデータである。データの形式はXYZ値のスペース気切りのテキストファイルで、XYは平面直角座標、Zは標高値(m単位)である。データの範囲は東京23区を網羅する、図1の範囲である。データは東西400、南北300mの図郭に分けられており、計6811ファイルで合計のデータサイズは226GBとなる。

2.2. 色付き点群の作成

3次元点群データは3DTilesへ変換する中間データとしてLASフォーマットへ変換した。また、その際に景観表現のため同範囲の航空写真(位置情報付き画像ファイル)を用意し、3次元点群データへの航空

* 賛助会員 朝日航洋株式会社 (Aero Asahi Corporation)
〒350-1165 埼玉県川越市南台 3-14-4 E-mail : dai-yagishita@aeroasahi.co.jp

** 正会員 朝日航洋株式会社 (Aero Asahi Corporation)

写真色情報の付加を行なった。変換および色情報の付加は、FME というデータ変換エンジンを用いた。FME ではトランスフォーマーと呼ばれる変換モジュールを組み合わせて図2のフローを構築して実施した。なお、変換後のファイル数は変わらず 6811 ファイルであり、LAS フォーマットはバイナリ形式であるため、色情報を付加してもファイルサイズは合計 227GB でほぼ変化なしであった。



図-1 データの範囲

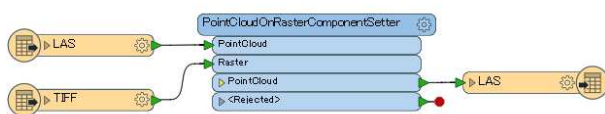


図-2 LAS への変換フロー

3. 変換の仕組み

3.1. 3DTiles のデータ構造

3DTiles は、3 次元モデルや点群などの大規模な 3 次元地理空間コンテンツをストリーミングおよびレンダリングするために設計されている。LOD を実現するために細かく分割された実データと、tileset.json と命名される JSON 形式の階層化の定義ファイルで構成されている。点群の場合、実データは Point Cloud 形式(pnts 拡張子)のバイナリファイルとなる。モデルの場合は Batched 3D Model 形式(b3dm 拡張子)のファイルである。tileset.json は入れ子構造で階層化され、階層ごとのデータファイル名とそのパラメータの記述で構成されている(図3)。パラメータの一つである Geometric Error は、視点からの距離に応じて読み込むデータを制御するために用いられる。設定の効果は、2 次元のタイルのズームレベルの概念に

近く、適切な Geometric Error 値を設定することにより 3DTiles 全体のデータ容量が増大しても影響を受けずに表示がスムーズに行うことが可能となる。

```
{
  "root": {
    "geometricError": 72.402,
    "refine": "ADD",
    "content": {
      "boundingVolume": {
        "box": [-3947293.64, 3422210.49, 3646716.96, 179
      ],
      "url": "r0.pnts"
    },
    "children": [

```

図-3. tileset. json の構成

3.2. 3DTiles への変換

las からの 3DTiles への変換は、オープンソースの変換ツールである py3dtiles を用いる。この変換ツールではデータのサイズに応じて最適化された階層構造の 3DTiles が作成される。また、入力 of las データは複数ファイルに対応しており、入力および出力については座標系を設定する。なお、後述する 3 次元ビューアエンジンによって対応している座標系が異なるので留意が必要である。変換に要した時間は今回の 227GB のデータ量で 8 時間程度であった。変換作業は Amazon Web Service(AWS) の EC2 インスタンスを使用し、インスタンスタイプは t2xlarge, OS は ubuntu16.04 である。さらに変換後の 3DTiles は AWS S3 にアップロードし、4.2 のパフォーマンス検証を実施した。

4. 表示ビューアとパフォーマンス

4.1. 3D ビューアエンジンについて

Web 上での 3 次元描画は、ウェブブラウザでプラグインを使用せずに 3 次元グラフィックスを描画するための標準仕様である WebGL が用いられている。WebGL をベースした 3 次元ビューアエンジンは多数あるが、GIS としての活用を想定して以下の条件を設定した。

- ・ 3DTiles で点群が表示可能なこと
- ・ 緯度経度の座標系を持ち、2D 地図が背景として表示可能であること
- ・ 低コストで検証・運用が可能なオープンソースであること

これらを満たす3次元ビューアエンジンとして、今回の検証ではCesiumJS、iTown、MapboxGLJSの3種類を選定した。各ビューアエンジンの特徴について表1に記す。

表-1 3次元ビューアエンジンの特徴

	CesiumJS	iTown	MapboxGLJS (mapbox-3dtiles)
ライブラリ	Cesium.js	Three.js	Three.js
地形表現	あり	あり	なし
座標系	EPSG:4978	EPSG:4978	EPSG:3857
最新バージョン	1.72 20/08/01	2.24.0 20/07/30	1.12.0 20/08/14
その他	時系列管理		2Dベクトル タイル表示

表1の3次元ビューアエンジンのうち、iTownとMapboxGLJSの3DTilesのレンダリング(プラグインmapbox-3dtilesで対応)はWebGLのJavaScriptライブラリとして広く普及しているThree.jsで構成されているのに対し、CesiumJSはWebGLベースの独自のJavaScriptライブラリで構成されている。いっぽう、地形表現の有する点と座標系はCesiumJSとiTownが共通である。いずれも頻繁にバージョンの更新がされていて、今後の発展が期待できる。

4.2. 表示パフォーマンス

4つの観点から表示パフォーマンスの検証を実施した。以下のその結果を記す。

① ビューアエンジンごとの描画パフォーマンス

各ビューアで同一の表示位置および範囲で描画パフォーマンス比較を行った。ブラウザは普及率の最も高いGoogle Chromeを使用した。表示範囲はエリア全域(2次元地図のズームレベル11相当)、ズームレベル15相当、ズームレベル17相当とした。また、視点を斜めにするなどの操作の追従性も確認した。エリアは新宿新都心周辺を選定した。キャッシュをクリアした状態と、キャッシュのある状態での描画完了までの時間を3回ずつ計測し、平均値を算出した(表2)。MapboxGLJSはプラグイン内で視点が遠い場合も描画密度が向上するようgeometric Errorに固

定の係数(20倍)が設定されていたが、本データに対しては過剰描画となりパフォーマンスが著しく低下したため、係数を2倍とした。MapboxGLJSはキャッシュがクリアの状態だと描画に時間がかかる傾向があったが、読込後の操作感はいずれのエンジンも良好であった。

表-2 ビューアエンジンごとの結果(単位:秒)

表示範囲	キャッシュ	Cesium JS	iTown	Mapbox GLJS
全域	なし	7.0	7.7	22.3
	あり	5.3	4.6	9.7
ズーム15	なし	24.3	14.7	47.3
	あり	15.3	9.7	12.7
ズーム17	なし	13.3	19.0	35.7
	あり	10.7	13.7	11.7
視点操作		○	○	○

② ビューアエンジンごとの設定変更による効果

ビューアごとに描画に効果のあるパラメータ設定が用意されており、その効果について検証をおこなった。CesiumJSにはScreenSpaceErrorを設定するいくつかのパラメータがあるが、本検証において有効だったのはmaximumScreenSpaceError(デフォルト値は16)である。視覚的に適度な間引きと描画パフォーマンスのバランスが取れる値がデフォルトになっている。この値を小さくすると点密度は上がるが描画時間が長くなり、値を小さくすると描画時間が短くなるが、点密度が下がる(図4)。iTownではsseThreshold(デフォルト値は16)パラメータが同様の効果がある。なお、MapboxGLJSでは設定可能なパラメータはなく、前述の係数で調整する。

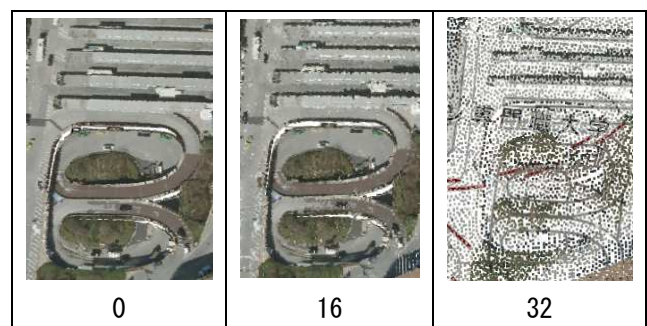


図-4 maximumScreenSpaceError 値による差

③ tileset.json の設定による効果

tileset.json の設定値は py3dtiles で最適化がなされていると推察されるが、geometric error 値が及ぼす影響について検証をおこなった。結果、大きな値を取ると視点に対し点群の表示密度が過剰になり、小さな値の場合密度が不足となった。密度の調整は②のパラメータで可能なので、基本的には geometric error 値を変更する必要はない。

④ ブラウザによる表示パフォーマンス

ブラウザごとの表示パフォーマンスについて、描画速度や操作感、メモリの使用量に着目して検証を実施した。なお、検証に使用した PC のメモリは 8GB である。

・ Google Chrome

描画速度は表 2 の通り。メモリは最大 2GB 程度使用されるが描画、操作感とも良好で安定する。

・ Firefox

メモリ使用量や操作感は Chrome 同様であるが、描画にかかる時間に差異があったので表 3 に示す。CesiumJS と iTowns の描画が Chrome より若干時間がかかる傾向があり、いっぽう MapboxGLJS はキャッシュがクリアな状態の描画時間を短縮しており、ビューアエンジン間の差は小さくなった。

表-3 Firefox での描画時間（単位:秒）

表示範囲	キャッシュ	CesiumJS	iTowns	MapboxGLJS
全域	なし	13.0	12.0	16.0
	あり	9.0	7.7	10.0
ズーム 15	なし	34.0	16.0	27.7
	あり	23.3	12.7	12.0
ズーム 17	なし	18.3	22.7	23.0
	あり	14.7	18.7	11.7

・ Edge / Internet Explorer

最新版の Edge や Internet Explorer では WebGL に対応しているが、全域表示の途中で描画が停止し、操作も不能となるなど非常にパフォーマンスが悪い。パラメータ設定で大幅に表示を間引くことで操作可能となる。

5. まとめ

検証の結果、大規模 3 次元点群データの WEB 配信は、データに適した 3 次元ビューアエンジンやブラウザを用いることで、高いパフォーマンスで可能であることが確認できた。

5.1 課題と解決策

変換およびデータの転送に要する時間がデータサイズに比例して大きくなるのが課題である。今回の検証作業では、変換とアップロードの手順で複数のツールを用いているため、元データ 226GB に対し、計 4 日かかる計算となる。解決にはワークフローによる自動化やデータ管理の一元化(アップロードの削減)などが考えられる。さらに即時性が求められる場合は、対象データを絞り込む必要もある。

5.2 今後の活用

3 次元点群データはオープンデータ化も進んでおり、入手がしやすくなっている。また本稿でも記したようにオープンソースを用いたデータ変換も可能である。3 次元点群データは景観シミュレーション、維持管理、水害・土砂災害予測などさまざまな活用が期待できる。広く普及している WEB ブラウザでパフォーマンスに問題なく取り扱えることは、3 次元点群データの活用拡大に貢献していくことと思われる。

謝辞

本稿を検証するにあたり、東京 23 区の点群データを WEB 配信で公開することを想定して G 空間情報センターの AWS 環境を利用させていただきました。一般社団法人社会基盤情報流通推進協議会 (AIGID) 大伴真吾様ならびに関係された方々に感謝いたします。

参考文献

Open Geospatial Consortium(2018)

3D Tiles Specification 1.0

<https://docs.opengeospatial.org/cs/18-053r2/18-053r2.html>