

## リスト1 必要なライブラリなどの準備

```
1 #Open3Dをインストール
2 !pip install open3d
3 #Googleドライブをマウント
4 from google.colab import drive
5 drive.mount('/content/drive')
6 # Open3DとNumpyをインポートします
7 import open3d as o3d
8 import numpy as np
9 # 点群ファイル (PLY) 読み込み
10 cloud = o3d.io.read_point_cloud("/content/drive/MyDrive/dataset/image_in/iruka.ply")
11 if cloud.is_empty(): exit()
12
13 # 読み込んだデータを確認します
14 print(cloud)
15 print(np.asarray(cloud.points))
16 # Matplotlibで点群ファイルを表示します
17 import matplotlib.pyplot as plt
18 from mpl_toolkits import mplot3d
19 # 点群データをnp.array形式に変換します
20 points = np.asarray(cloud.points)
21 # 色データをnp.array形式に変換します
22 colors = np.asarray(cloud.colors)
23 print(colors)
24 # 表示サイズを指定
25 import matplotlib.pyplot as plt
26 from matplotlib.pyplot import figure
27 plt.rcParams["figure.figsize"] = [10, 10] # 図の幅と高さをインチで指定する
28 # Matplotlibで表示
29 ax = plt.axes(projection='3d')
30 ax.view_init(elev=0, azim=0)
31 ax.axis("off")
32 ax.scatter(points[:,0], points[:,1], points[:,2], s=1, c=colors)
33 plt.show()
34 # 軸の仰角と方位を変更
35 ax = plt.axes(projection='3d')
36 ax.view_init(elev=90, azim=-45) # 軸の仰角と方位を（ラジアンではなく）度単位で設定
37 ax.axis("off")
38 ax.scatter(points[:,0], points[:,1], points[:,2], s=1, c=colors)
39 plt.show()
```

## リスト2 点群データ (PLYファイル) をOpen3DとMatplotlibで可視化

```
1 # メッシュタイプ (OBJファイル) のデータを読み込み
2 mesh = o3d.io.read_triangle_mesh("/content/drive/MyDrive/dataset/image_in/iruka.obj")
3 if mesh.is_empty(): exit()
4 # 法線を計算
5 mesh.compute_vertex_normals() # 頂点の法線を計算
6 mesh.compute_triangle_normals() # 三角形の法線を計算
7 # 頂点と三角形のデータをnp.array形式に変換
8 triangles = np.asarray(mesh.triangles)
9 vertices = np.asarray(mesh.vertices)
10 # 色を設定
11 colors = (0.5, 0.5, 0.5) + np.asarray(mesh.triangle_normals) * 0.5 # 三角形の法線データで色を変化させる
12 colors = tuple(map(tuple, colors))
13 # Plotlyをインストール
14 !pip install plotly
15 # Plotlyをインポート
16 import plotly.graph_objects as graph_objects
17 # Plotlyで表示。出力結果はマウスでインタラクティブに動かすことが可能です
18 fig = graph_objects.Figure(
19     data=[
20         graph_objects.Mesh3d(
21             x=vertices[:,0], # 頂点のX座標
22             y=vertices[:,1], # 頂点のY座標
23             z=vertices[:,2], # 頂点のZ座標
24             i=triangles[:,0], # 三角形の1番目の座標
25             j=triangles[:,1], # 三角形の2番目の座標
26             k=triangles[:,2], # 三角形の3番目の座標
27             facecolor=colors, # 面の色を設定
28             opacity=0.50) # 表面の不透明度を設定
29     ],
30     layout=dict(
31         scene=dict(
32             xaxis=dict(visible=False), # X軸非表示
33             yaxis=dict(visible=False), # Y軸非表示
34             zaxis=dict(visible=False) # Z軸非表示
35         )
36     )
37 )
38 fig.show() # 表示
```

## リスト3 メッシュデータ (OBJファイル) をOpen3dとPlotlyで可視化

```
1 #MMS の点群データ (Lasファイル) をLeafmapで可視化
2 # Leafmap, Open3dをインストール
3 !pip install leafmap[lidar] open3d
4 #Googleドライブをマウントします
5 from google.colab import drive
6 drive.mount('/content/drive')
7 # Open3DとNumpyをインポートします
8 import os
```

```

9 import leafmap
10 import open3d as o3d
11 import numpy as np
12 # Leafmapで点群データ (lasファイル) を読み込みます
13 %%time
14 las = leafmap.read_lidar("/content/drive/MyDrive/dataset/image_in/080F4060.las")
15 # 点群のポイント数を確認
16 las.header.point_count
17 # Matplotlibをインポート
18 import matplotlib.pyplot as plt
19 from mpl_toolkits import mplot3d
20 # データを3次元配列に変換
21 points = np.vstack((las.X, las.Y, las.Z))
22 colors = np.vstack((las.red, las.green, las.blue))
23 # 軸の順番を入れ替え
24 points = points.transpose() # 3次元座標
25 colors = colors.transpose() # RGB情報
26 # データを間引
27 step = 100 # データを1/100に間引く
28 decimated_points = points[::step]
29 decimated_colors = colors[::step] / 65535.0 #データを0.0-1.0で正規化します
30 # データのサイズを確認
31 len(decimated_points)
32 # 点データ確認
33 decimated_points
34 # 色情報確認
35 decimated_colors
36 # Matplotlib設定
37 import matplotlib.pyplot as plt
38 from matplotlib.pyplot import figure
39 plt.rcParams["figure.figsize"] = [15, 15] # 図の幅と高さをインチで指定する
40 # Matplotlibで点群表示
41 ax = plt.axes(projection='3d')
42 ax.view_init(elev=45, azim=0)
43 ax.axis("off")
44 ax.scatter(decimated_points[:,0], decimated_points[:,1], decimated_points[:,2], s=1, c=decimated_colors)
45 plt.show()

```

#### リスト4 PyPotreeで可視化

```

1 # PyPotreeのインストール
2 !pip install pypotree
3 # PyPotreeのインポート
4 import pypotree
5 # PotreeViewer を用いて3D点群を表示するための設定
6 cloudpath = pypotree.generate_cloud_for_display(decimated_points)
7 # generate_cloud_for_display で生成したパスに点群を表示
8 pypotree.display_cloud_colab(cloudpath)

```