

PyTorch3Dを用いた3次元点群の投影

2022-12-20 10:00 Hachiuma Ryo
AI, DataAnalytics

FORXAI Engineering Blogをご覧ください。誠にありがとうございます。

技術開発本部 FORXAI開発センター AI技術開発部の八馬です。私が所属するチームは主に**人行動**に関連するAI技術の開発を行っています。

過去には、**高速な人物行動認識手法の研究開発**をしておりました。

さて、今回の記事では打って変わって3次元点群処理に関する小ネタを紹介させていただきます。毎回3次元点群、特に座標系変換で頭を悩ませており、調べてもライブラリによって座標系の扱い方が異なっており、どのライブラリをどう使った方がいいか困っていたため、備忘録も含めた記事になります。

少なくとも日本語記事は執筆時点ですぐ検索で見つからなかったため、この記事が同じように検索した人の助けになれば幸いです

目次 (非表示)

1. 3次元点群の投影
2. 3次元点群処理ライブラリ
3. PyTorch3Dによる実装
 - 3.1. look_at_view_transform
 - 3.2. FoVOrthographicCameras
 - 3.3. transform_points_screen
4. おわりに
5. 引用

3次元点群の投影

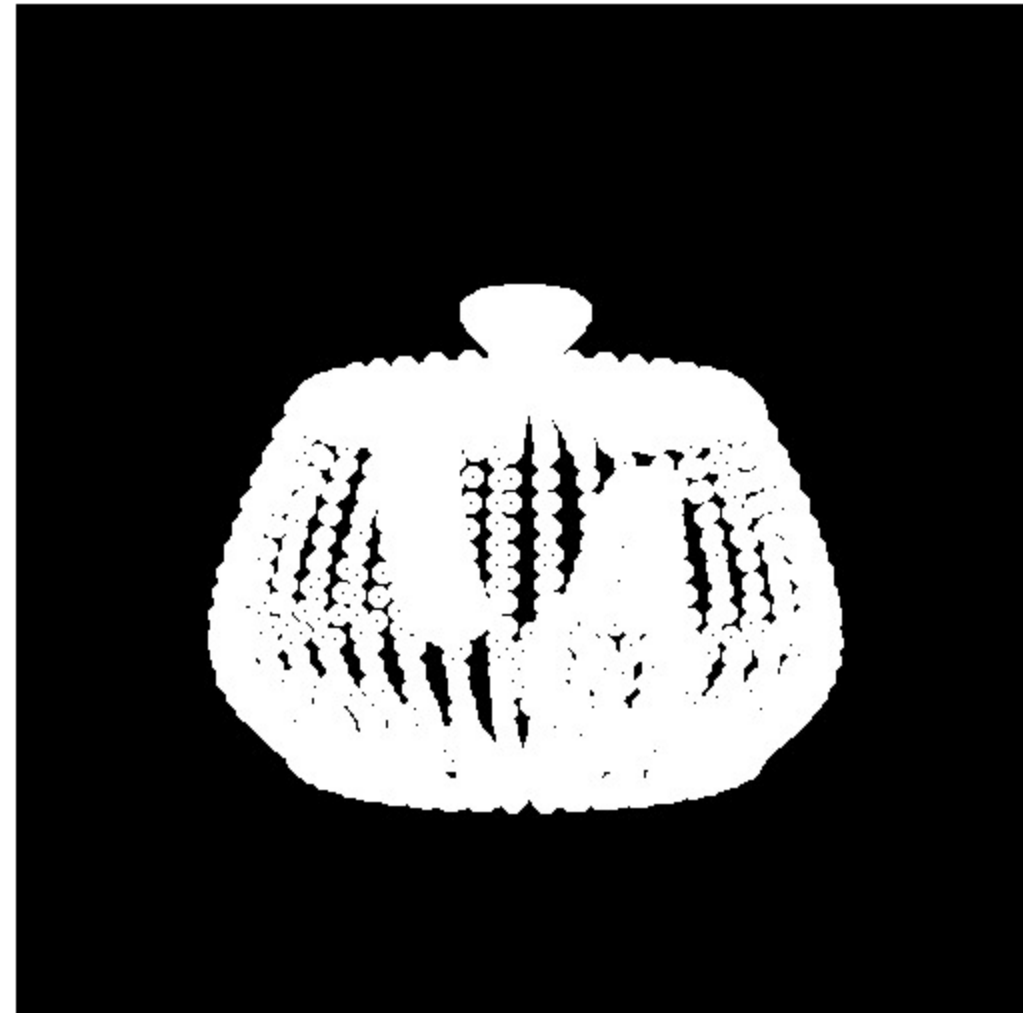
今回紹介する話は、3次元点群を2次元画像に投影する処理をPyTorch3Dで簡単にやってみた、というような内容になっています。

具体的には、ただ投影された画像を作りたいただけではなく、その投影された2次元座標を取得したい、また、3次元の原点中心にカメラを回し様々な方向から自動で撮影したい、といった条件がありました。

結果として3次元点群を入力とし、以下のような結果を得ることができます。(実際には、投影された点座標を取得することができます。) 入力には、Utah teapod [1]をお借りしました。



3次元点群



投影された画像

3次元点群処理ライブラリ

Pythonで3次元点群を扱うライブラリとしては、BlenderのPythonスクリプトである**box, Open3D, PyTorch3D**など様々なライブラリがあります。(私個人としては、学生時代**Point Cloud Library**を主に使って3次元点群処理のコードを書いていました。)

しかし、ライブラリによっては、カメラが原点を向きながら任意の角度から撮影するコードが一筋縄ではない、カメラの配置が難解、またその投影された点の座標を取得することが困難、座標系の扱い方が複雑などといったことがあり、インターネットで探してもなかなか見つかりませんでした。

PyTorch3Dによる実装

ほとんどの場合、各ライブラリで定義されている座標系を使って座標系毎に幾何変換を施していき、3次元点を画像座標まで変換する必要がありますが、PyTorch3Dを用いるとたった3つの関数でカメラの設定・座標の投影までできました。その関数は以下の3つです。

- [look_at_view_transform](#)
- [FoVOrthographicCameras](#)
- [transform_points_screen](#)

最初2つの関数は、カメラをどの位置に置くか、やカメラ自身のインスタンスを生成する関数であり、実際の投影は最後のtransform_points_screenの関数1つで行っています。

順番に紹介していきます。

look_at_view_transform

この関数では、カメラの位置を定義します。引数としては、カメラの仰角・方位角・原点からの距離を取り、カメラのRotation, translationを出力してくれます。

```
R, T = look_at_view_transform(dist=cam_dist, elev=cam_elev, azim=cam_azim, degrees=True)
```

それ以外にも、look_atでカメラの視線方向も決められるのですが、今回投影対象の物体は原点にあるとしているので、入力していません。

FoVOrthographicCameras

次の関数では、その得られたRTを用いてカメラインスタンスを生成します。

```
cam = FoVOrthographicCameras(R=R, T=T)
```

transform_points_screen

最後に生成されたカメラインスタンスを用いて3次元点を2次元座標上(screen space)に投影し、その2次元点の座標を出力してくれます。この時、画像サイズ(IMG_SIZE)を引数として入力することで、その画像サイズでの2次元点を出力してくれるようになります。

```
points2d = cam.transform_points_screen(points3d, image_size=(IMG_SIZE, IMG_SIZE))
```

これらの関数を使って、任意の仰角・方位角から投影することで、1で示したような動画を作成(2次元点を取得)することができるようになります。

おわりに

この記事では、PyTorch3Dを使って3次元点を任意の角度のカメラから投影し2次元点を簡単に取得する方法をご紹介させていただきました。過去3次元点群の座標系や、カメラをどう配置すればよいかなど頭を悩ませてきたのですが、とても簡単に書いて驚きました。本記事が皆様にとって有益な情報であれば、幸いです。

引用

- [1] [CS184 AS6: Triangle Meshes and Speed-up Structures for your Raytracer](#)

コニカミノルタは画像IoTプラットフォームFORXAIを通じて、お客様やパートナー様との共創を加速させ、技術・ソリューションの提供により人間社会の進化に貢献してまいります。

新卒採用情報 - 採用情報 | コニカミノルタ



コニカミノルタの新卒採用サイトです。募集要項や募集職種などの採用情報から、プロジェクト紹介、社員インタビューなどを掲載しています。ぜひご覧ください。
KONICA MINOLTA

中途採用に関する情報については以下の採用情報ページをご覧ください。

キャリア採用情報 - 採用情報 | コニカミノルタ



コニカミノルタキャリア採用情報 現在の募集職種にはこちらからエントリー可能です。募集要項、先輩インタビュー、人事部からのメッセージなど掲載。
KONICA MINOLTA

Hachiuma Ryo

FORXAI事業統括部 AI技術開発部 所属 人行動領域を中心とする機械学習のモデル開発などを行っています

前の記事



ASPIC IoT-AI-クラウドアワード2022受賞まで

次の記事



最新論文調査 Self-Supervised Anomaly Detectionの出版

- シェアする: X, Post, Facebook, Pocket, LINE, etc.