

Week 2 Report

Previous week issues.

- Normally, FlyingThings3D has annotation for scene flow (it is the dataset used for supervised training in FlowNet3D). But it is still possible that author remove annotation and train it in a self-supervised manner. The author did not state clearly how they pretrain the model. [Github issues on this topic](#)
- About the output of model. It is the flow. More specifically, the input of the network F is two point clouds P, Q , the output is flow D (translational motion vectors for each point), such that ideally $P + F(P, Q) = Q$. (ref. FlowNet3D)
- About the inference of "reversed flow". After forward flow we have our estimated point clouds for 2nd frame $Q' = P + F(P, Q)$. Then the reversed flow is computed as $D' = F(Q', P)$. The Cycle consistency loss term is simply $\|D + D'\|$. (ref. FlowNet3D)

This week

KITTI Scene Flow

- **Info**: originally designed for RGB stereo based method. **Raw data** of 3D Velodyne point clouds are used in this paper (following FlowNet3D (Liu et al., CVPR 2019))
- **Makeup**: Original KITTI contains 200 scene, where 150 for train and 50 for testing. However, 50 test scene are not associated with raw point cloud data. Hence, only 150 scene (100 train, 50 test) are used in this project.
- metric: End Point Error (EPE), Acc (0.05): percentage of scene flow prediction with an EPE $< 0.05\text{m}$ or relative error $< 5\%$ (also following FlowNet3D)
- preprocessing: ground point are removed.
- **** Details:****
 - Velodyne scans (i.e., the point clouds) are stored as floating point binaries, each point is stored with (x, y, z) coordinate and reflectance value (r) , (however this project does not use (r))
 - **Format**: each "scene" are associated with 3 attributes. `pos1`; `pos2`; `gt`, where `pos` are Cartesian coordinates of points in R^3 , `gt` is the displacement vector (label), and `pos1 + gt = pos2`. (unit meter)

```
X["pos2"]

array([[ -9.42635918,  -3.51228802,   0.7285
5435],
       [ -9.41987018,  -3.54181502,   0.7289
0535],
       [ -9.36450018,  -3.43048602,   0.7288
4935],
       ...,
       [ -9.31897918,  -2.99916502,  -0.7229
6165],
       [ -9.36137818,  -2.99597502,  -0.7140
3265],
       [ -9.36182418,  -3.00256602,  -0.7140
7665]])

X["pos2"].shape

(33599, 3)
```

About the NuScenes dataset

- A larger dataset compared with KITTI, but no scene flow label is available for point clouds
- Actually, the authors preprocessed the NuScenes dataset but they did not release the processed dataset.
- However, we can infer from the source code that the format of point clouds are similar as KITTI. *The major difference is that the point in this dataset has additional RGB color attributes.* (see `nuscenes_dataset_self_supervised_cycle.py` for more details)

Source Codes

- The major part of codes are edited from pointnet++ and flownet3d.
- load dataset: `kitti_dataset_self_supervised_cycle.py`
- model definition `model_concat_upsa.py` (FlowNet3D).
`model_concat_upsa_cycle.py` (self-supervised)
- training: `train_lnn_cycle_nuscenes.py` (self-supervised.)

next week

code detail. flow embedding, dataset. find "problems" in the codes.

important: data processing, net arch, loss; regularization, opt (train)(hyper-param), metric.

when fail - analysis

