FlowNet3D: Learning Scene flow in 3D point cloud.

Summary

1. Propose a novel architecture called FlowNet3D that estimates scene flow from a pair of consecutive pt cloud end-to-end.

2. The network consists of 3 steps:
   - hierarchical pt cloud feature learning (with Set Conv)
   - Point mixture with flow embedding layer (flow embedding layer)
   - Flow Refinement with Set Upconv Layer (set upconv layer)

3. Prev methods focus on stereo and RGB-D images as ipt, few try to estimate scene flow directly from point clouds.

Problem Definition

1. Inputs are 2 sets of points sampled from a dynamic 3D scene. at 2 consecutive frames: $P = \{x_i \mid i = 1, \ldots, n_1\}$ x, y are XYZ coordinates.
   $Q = \{y_j \mid j = 1, \ldots, n_2\}$

2. Due to obj motion & viewpoint changes, the 2 pts cloud do not necessarily have the same no. of points or correspondence b/t the points.

3. Consider the point x_i moves to location x_i', let $d_i = x_i' - x_i$.

4. The goal is to recover the scene flow for every point in the first frame $D = \{d_i \mid i = 1, \ldots, n\}$ given P and Q.

There are 3 types of point cloud processing layers:

- set conv layer
- flow embedding layer
- set upconv layer

Set Conv layer

1. A set conv layer takes as input:
   - a pt cloud with n points, each point $p_i = \{x_i, f_i\}$
     - x_i : XYZ coordinates (R^3)
     - f_i : feature (R^c)

   and outputs:
   - a sub-sampled pt cloud with n' pts, each point $p_j' = \underbrace{\{x_j', f_j'\}}_{\text{updated coor \& features.}}$

2. Specifically, the layer:
   - samples n' regions from the input pts with farthest point sampling. (region centers are x_j')
   - then, for each region (defined by a radius neighborhood specified by radius r), local features are extracted with:

$$f_j' = \underset{\{i \mid \|x_i - x_j'\| \leq r\}}{MAX} \{ h(f_i, x_i - x_j') \}$$

where
- $h : R^{c+3} \to R^{c'}$ is a non-linear function.
- MAX is element-wise max pooling

Flow Embedding Layer

1. The layer takes as ipt:
   - a pair of pt clouds: $\{p_i = (x_i, f_i)\}_{i=1}^{n_1}$ & $\{q_j = (y_j, g_j)\}_{j=1}^{n_2}$

   and output $\{e_i\}_{i=1}^{n_1}$ where $e_i \in R^{c'}$

2. $e_i = \underset{\{j \mid \|y_j - x_i\| \le r\}}{MAX} \{h(f_i, g_j, y_j - x_i)\}$

Set Upconv Layer

1. The ipt are:
   - source points $\{p_i = \{x_i, f_i\} \mid i = 1, \dots, n\}$
   - target pts coordinate $\{x'_j \mid j = 1, \dots, n'\}$

   For each tgt location, output features f'_j.

2. The layer can be implemented by set conv layer, but with a diff. local region sampling strategy.

3. Instead of using farthest point sampling to find x'_j, they compute features on specified locations by the tgt points $\{x'_j\}_{j=1}^{n'}$. $(n' > n)$

Training

1. Training is done with GT scene flow supervision.
2. The GT is obtained from large-scale synthetic dataset.
3. The model trained on synthetic data generalizes well to real Lidar scans.

Qns

1. How are the pt features f_i computed for the first set conv layer ?
2. In the flow refinement step, how do they select the target point coordinates ?