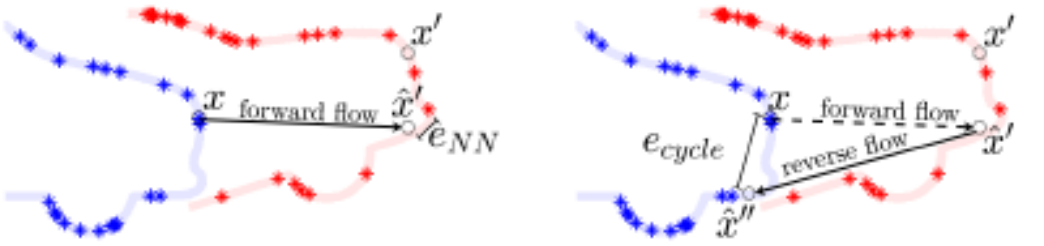
week 1 report

What I have done

- read through chapter 9 (Motion Estimation) of *Computer Vision: Algorithms and Applications*, understand the goal of scene flow estimation and previous techniques.
- Also read *A survey on deep learning methods for scene flow estimation, Pattern Recognition 2020*. to get some understanding of the scene flow estimation problem.
- It took me about 3 days to configure the training environment. (I set up a Linux workstation locally)
- now I'm trying to understand the source code.

About the Paper.

- Motivation: hand-crafted label are expensive for sceneflow estimation problem
- Problem definition: (Lossly) find a dense correspondence of points between two frames
- Method: proposed a self-supervised method for sceneflow estimation by *introducing two loss functions (nearest neighbor and cycle consistency)*.
- Firstly, to compensate for the absence of supervision signal, use the nearest neighbor of the transformed point as an approximation for the true correspondence. More specifically, the "nearest neighbor loss" is the Euclidian distance of the transformed point with its closest neighbor.
-



(a) Nearest Neighbor Loss (b) Cycle Consistency Loss

- However, nearest neighbor loss alone is not sufficient in 3 ways:
-
    - 1: the nearest neighbor is just an approximation and may not be the "real point" after displacement.
-
    - 2: Due to the sparsity of points cloud, a transformed point may not correspond to any point in the next frame.

    - 3: this loss does not penalize degenerated solution where all points are transformed to exactly same location between two frames.
- To solve above issues, "cycle consistency loss" is used, possibly inspired by CycleGAN. Roughly speaking, a point will be transformed back and forth, and the loss value is the Euclidian distance between the two points.
- However, cycle consistency loss comes from two phases, forward flow and backward flow. Optimizing the network minimize those two error simultaneously, but only forward flow should be optimized in this sceneflow estimation problem. The author use "anchoring point" to restrict the backward flow and push the network to learn forward flow only.
- Lastly, the overall loss is a combination of the two losses
- Augmentation: reverse the flow (Although sceneflow are point-to-point estimation, perhaps reversed flow may introduce some problem that may not happen in real-world where time only go forward?).
- Architecture: FlowNet3D, HPLFlownet (Seems that the loss is independent of architectures?)
- All model are pretrained on Flyingthings3D (contradict with "purely self-supervised "?)
- Experiment: Purely self-supervised, self-supervised with supervised fine-tuning.

Tentative Plan

- Try to fully understand the source code written in Tensorflow 1.x
- dataset structure
- paper detail.