FlowNet3D: Learning Scene Flow in 3D Point Clouds

承子杰 202228000243001

---

1. 主要内容

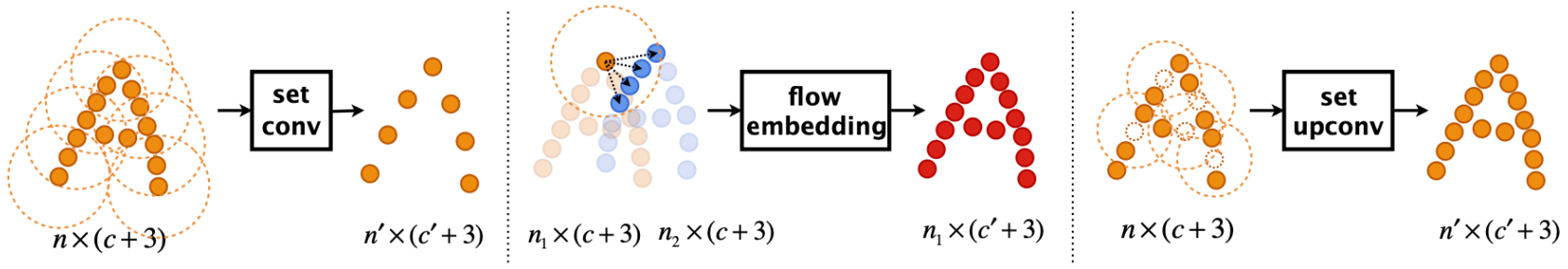
场景流是点在动态环境中的三维运动信息。相比于在二维图像中传统的光流算法，由于对象运动和视点的变化会带来运动前后点云中点数量的不一致，以及点与点之间缺乏明确的对应关系，在三维点云中如何计算和使用场景流仍然是一个巨大的研究挑战。该论文提出了一种新的网络架构 `FlowNet3D` 用于对点云中的场景流进行端到端的学习，并使用 `FlyingThings3D` 数据集与 `KITTI` 激光雷达扫描数据对该网络的性能进行评估。

2.网络架构

2.1 问题定义

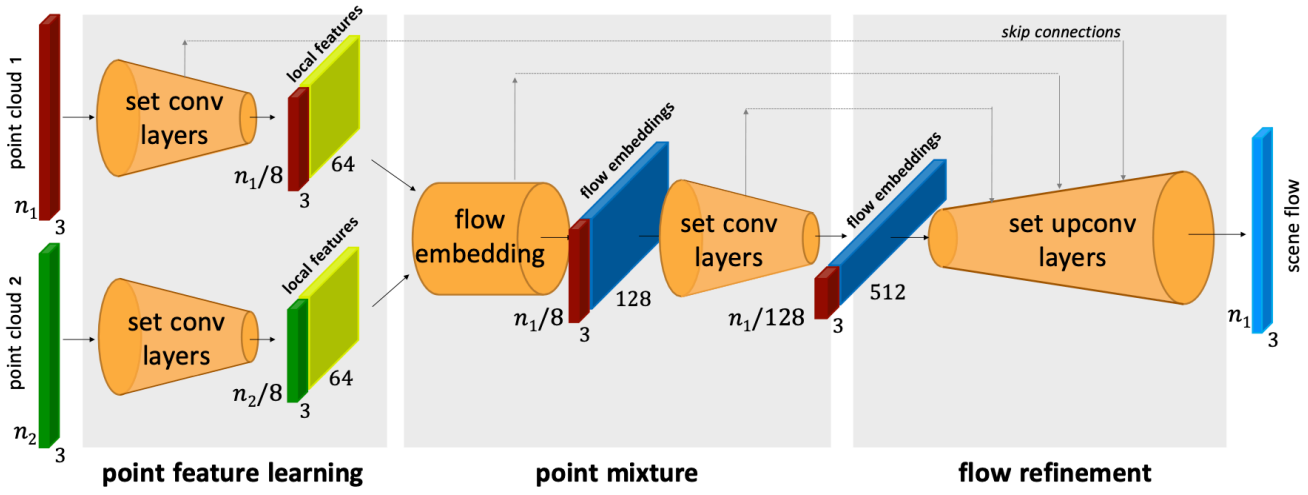
考虑存在同一点云的前后连续两帧Q, P，假设我们在Q中任意采样一个点x_i，在下一帧P中改点变化到了x_i'，则我们定义该点的场景流为$d_i = x_i' - x_i$。我们需要解决的问题是获得前后两帧P, Q中所有采样点对应的场景流$D = \{d_i | i = 1, 2, \ldots, n\}$。

2.2 基本设计



图像 1

`FlowNet3D` 是一个基于点云的端到端的场景流估计模型，主要有三个模块组成(图像1)：点特征学习(`point feature learning`)，点信息融合(`point mixture`)和点云流上采样(`flow refinement`)。具体而言，在三个模块中还分别融合了三个重要的点云处理层(图2)：集合卷积层(`set conv layer`)，流嵌入层(`flow embedding`)和集合上卷积层(`set upconv layer`)。下面我么就每个模块与其相对应的处理层进行一一介绍。



图像 2

`point feature learning ( set cone layers )`

在模块中，作者采用了 `PointNet++` 结构中的 `feature learning layer` 。具体地，该模块采用 `farthest point sampling` 技术对输入点云的前后两帧分别进行下采样操作(区域中心点记为x_j')，再对每个采样点的r-邻域使用如下的对称函数f进行局部特征提取。

$$f_j' = \mathrm{MAX}_{\{i | \|x_i - x_j'\| \le r\}} \{h(f_i, x_i - x_j')\}.$$

其中，$h : \mathbb{R}^{c+3} \to \mathbb{R}^{c'}$是一个非线性算子函数（以一个多层感知机实现），$\mathrm{MAX}$是逐元素的最大池化。

`point mixture ( flow embedding layers  +  set conversion layers )`

对于点云而言，由于存在采样的稀疏性、物体遮挡和视点变换等多种问题，我们经常无法明确前后两帧中任一采样点的对应关系（甚至可能前后不一定存在）。对此作者提出了一种对应的 软策略 ,即对于前一帧的任一点在下一帧中寻找其一系列的柔和对应点，在采用投票法对其场景流进行估计。具体地，假设点云前一帧为Q，后一帧为P，将点云坐标与前一层提取的特征向量进行两两配对作为输入$\{p_i = (x_i, f_i)\}_{i=1}^{n_1}$和$\{q_j = (y_j, g_j)\}_{j=1}^{n_2}$，通过使用点对的几何特征相似性和空间关系，对于$Q$中任一点$x_i$在$P$中进行最近邻查找，设查找结果中的点为$y_j$，再使用一个神经网络层对各近邻投票进行聚合，得到最后的 `flow embedding` ，具体的计算公式如下

$$e_i = \mathrm{MAX}_{\{j | \|y_j - x_i\| \le r\}} \{h(f_i, y_j - x_i)\}.$$

`flow refinement ( set upconv layers )`

在 point feature learning 模块中，由于我们使用 FPS 算法进行了下采样，而目标最后需要获得全局的场景流信息，因此我们还需要进行一步上采样操作来估计每一个点。借鉴二维卷积的上采样，我们可以通过再做一次类似于 point feature learning 中公式(1)的操作，但这次我们是使用加权的方法来聚合多个点的特征。

最后，我们给出 FlowNet3D 的具体的网络架构(图像 3)。

Layer type	r	Sample rate	MLP width
set conv	0.5	0.5×	[32, 32, 64]
set conv	1.0	0.25×	[64, 64, 128]
flow embedding	5.0	1×	[128, 128, 128]
set conv	2.0	0.25×	[128, 128, 256]
set conv	4.0	0.25×	[256, 256, 512]
set upconv	4.0	4×	[128, 128, 256]
set upconv	2.0	4×	[128, 128, 256]
set upconv	1.0	4×	[128, 128, 128]
set upconv	0.5	2×	[128, 128, 128]
linear	-	-	3*

图像 3

2.3 训练与推断技巧

作者使用了平滑的L_1损失(huber loss)与 cycle-consistency 正则化进行监督训练，具体的损失函数公式如下

$$L(P, Q, D^*, \Theta) = \frac{1}{n_1}\sum_{i=1}^{n_1}\{\|d_i - d_i^*\| + \lambda\|d_i' + d_i\|\}$$

其中d_i为预测的场景流，d_i^*为给定的标签，d_i'为逆场景流。

此外由于下采样和上采样恢复操作会给最后的预测结果中引入噪声，作者采用了重采样操作来稳定结果，缓解该问题。具体而言，即进行多次重复采样，最后平均各次结果作为最后采样结果。

3. 实验结果

由于目前基本不存在带有标注的3D场景流信息数据集，作者采用了合成数据集 FlyingTings 3D 进行训练和预测，并微调后在 KITTI 数据集上进行使用。作者在最后还指出场景流的预测在多种下游任务(配准，分割，检测等)中有着广泛的应用。

3.1 FlyingThings 3D

Method	Input	EPE	ACC (0.05)	ACC (0.1)
FlowNet-C [9]	depth	0.7887	0.20%	1.49%
	RGBD	0.7836	0.25%	1.74%
ICP [4]	points	0.5019	7.62%	21.98%
EM-baseline (ours)	points	0.5807	2.64%	12.21%
LM-baseline (ours)	points	0.7876	0.27%	1.83%
DM-baseline (ours)	points	0.3401	4.87%	21.01%
FlowNet3D (ours)	points	**0.1694**	**25.37%**	**57.85%**

图像 4

> EM-baseline ： 在一开始就混合两个点云，使用一个特殊向量标记区分两个点云
>
> LM-baseline :先计算两个点云的全局特征，然后连接这一特征作为混合点云的方式
>
> DM-baseline ： 采用插值的方式传播点的特征

作者通过计算预测向量和真实向量之间的距离(EPE)与指定终点区域的流向量范围以判定预测结果是否误判(ACA)来衡量场景流预测的性能，并采用 FlowNet-C 和 ICP 进行对比实验。此外，作者还对不同的点云操作(图像 4)、网络架构与正则化技术等(图像 5)方面进行了丰富的实验尝试，给出了一些 baseline 为后续的研究提供参考。最后的实验结果如下

Feature distance	Pooling	Refine	Multiple resample	Cycle-consistency	EPE
dot	avg	interp	✗	✗	0.3163
dot	max	interp	✗	✗	0.2463
cosine	max	interp	✗	✗	0.2600
learned	max	interp	✗	✗	0.2298
learned	max	upconv	✗	✗	0.1835
learned	max	upconv	✓	✗	0.1694
learned	max	upconv	✓	✓	**0.1626**

图像 5

通过实验可以发现如下几个规律：

- 最大值池化操作与平均池化操作相比，存在着显著的优势
- 使用 `feature distance functions` 对于性能存在一定的提升
- 使用多次重采样技术和 `cycle-consistency` 正则化对网络性能也有一定的提升

3.2 KITTI

作者采用了多种模型作为对比组，在 `KITTI` 数据集上进行了去除地面与保留地面两种情况下的实验，具体的实验情况如下所示。

Method	Input	EPE (meters)	outliers (0.3m or 5%)	KITTI ranking
LDOF [5]	RGB-D	0.498	12.61%	21
OSF [17]	RGB-D	0.394	8.25%	9
PRSM [31]	RGB-D	0.327	6.06%	3
	RGB stereo	0.729	6.40%	
Dewan et al. [8]	points	0.587	71.74%	-
ICP (global)	points	0.385	42.38%	-
ICP (segmentation)	points	0.215	13.38%	-
FlowNet3D (ours)	points	**0.122**	**5.61%**	-

图像 6(去除地面)

> `LDOF` ： 使用变分模型获得光流，并将深度作为一个额外的特征维度。
>
> `OSF` ： 假设物体为刚体，在超像素上使用条件随机场
>
> `PRSM` ： 对刚性移动段的能量进行最小化，并对多个属性进行联合估算
>
> `ICP (global)` ： 对整个场景进行全局的刚体运动估计
>
> `ICP (segmentation)` ： 对整个场景分块进行刚体运动估计

Method	PRSM [31] (RGB stereo)	PRSM [31] (RGB-D)	ICP (global)	FlowNet3D (without finetune)	FlowNet3D + ICP (without finetune)	FlowNet3D (with finetune)
3D EPE	0.668	0.368	0.281	0.211	0.195	**0.144**
3D outliers	6.42%	**6.06%**	24.29%	20.71%	13.41%	9.52%

图像 7（保留地面）

通过实验结果可以发现，保留地面对所有方法的检测精度都存在一定的影响，其中 `PRSM` 的 `3D outliers` 最少，而微调的 `FlowNet3D` 具有更高的EPE。

4. 总结

该论文的主要贡献具体有如下几点：

1. 提出了一种可用于在给定点云的连续两帧间进行端到端场景流信息估计的新型网络架构 `FlowNet3D` 。
2. 在点云的操作处理上创新性地引入了两个学习层： `flow embedding layer` 与 `set upconv layer` 。
3. 在合成数据集 `FlyingTings 3D` 和 `KITTI` 数据集上预测训练了 `FlowNet3D` 网络，与传统方法相比获得了性能的巨大提升。