

基于邻域一致性的点云场景流传播更新方法

郑晗¹ 王宁¹ 马新柱² 张宏¹ 王智慧¹ 李豪杰¹

¹(大连理工大学-立命馆大学国际信息与软件学院 辽宁大连 116000)

²(悉尼大学电气与信息工程系 澳大利亚悉尼 2006)

(zheng_han@mail.dlut.edu.cn)

Point Cloud Scene Flow Propagation Update Method Based on Neighborhood Consistency

Zheng Han¹, Wang Ning¹, Ma Xinzh², Zhang Hong¹, Wang Zhihui¹, and Li Haojie¹

¹ (International School of Information Science and Engineering, Dalian University of Technology - Ritsumeikan University, Dalian, Liaoning 116000)

² (Department of Electrical and Information Engineering, The University of Sydney, Sydney, Australia 2006)

Abstract Scene flow is a 3D motion field between continuous dynamic scenes, which is widely applied in robotics and autonomous driving tasks. Existing methods ignore the correlation of point cloud points and focus only on the point-by-point matching relationship between the source point cloud and target point cloud, which is still challenging to estimate the scene flow accurately at the points with insufficient local feature information, because the matching relationship depends entirely on the feature information of the point cloud data. Considering the correlation property of the source point cloud's local regions, the NCPUM (neighborhood consistency propagation update method) is proposed to propagate the scene flow from high-confidence points to low-confidence points in local regions, so as to optimize the scene flow at the points with insufficient local feature information. Specifically, NCPUM consists of two modules: the confidence prediction module, which predicts the confidence of the source point cloud according to the priori distribution map of scene flow; the scene flow propagation module, which updates the scene flow of the low confidence point set based on the local area consistency constraint. We evaluate NCPUM on both challenging synthetic data from Flyingthing3D and real Lidar scans from KITTI, and experiment results outperform by previous methods a large margin in accuracy, especially on KITTI dataset, because the neighborhood consistency is more applicable with the a priori assumptions of real Lidar scans.

Key words scene flow; point cloud; neighborhood consistency; confidence; deep learning

摘要 场景流是连续动态场景之间的3D运动场,广泛应用于机器人技术和自动驾驶任务。现有方法忽略了点云点的相关性,仅关注源点云和目标点云逐点的匹配关系,由于匹配关系完全依赖于点云数据的特征信息,导致在局部特征信息不足的点上准确估计场景流仍然存在挑战。根据源点云相邻点具有相关性的特性,提出NCPUM (neighborhood consistency propagation update method)方法,在邻域内将场景流从高置信度点向低置信度点传播,从而优化局部特征信息不足点的场景流。具体来说, NCPUM 包含2个模块:置信度预测模块,根据场景流先验分布图,预测源点云逐点的置信度;场景流传播模块,根据局部区域一致

收稿日期: 2021-07-09; 修回日期: 2022-02-25

基金项目:国家自然科学基金项目(61976038, 61932020, 61772108, U1908210); 中央高校基本科研业务费专项资金(DUT20GF18)

This work was supported by the National Natural Science Foundation of China (61976038, 61932020, 61772108, U1908210) and the Fundamental Research Funds for the Central Universities (DUT20GF18).

通信作者: 王智慧 (zhwang@mail.dlut.edu.cn)

性的约束更新低置信度点集的场景流.NCPUM 在合成数据集 Flyingthings3D 和真实驾驶场景数据集 KITTI 上进行评估, 准确度上达到了国际先进水平. 由于邻域一致性更符合真实激光雷达场景的先验假设, 因此在 KITTI 数据集上的提升更加明显.

关键词 场景流; 点云; 邻域一致性; 置信度; 深度学习

中图法分类号 TP391

场景流(scene flow, SF)是定义在 2 个连续场景中表面点的 3D 运动场, 是一个感知动态场景的基本工具. 随着自动驾驶、人机交互等应用的大规模商业落地, 感知系统需要精确感知环境中的动态运动物体^[1-2], 因此精确估计场景流成为近年来的研究热点. 由 LiDAR 等 3D 传感器直接获得的点云数据可以得到场景中点的精确位置信息, 因此点云数据被广泛应用于场景流估计任务中. 点云数据仅包含 3D 点坐标, 因此在稀疏点、边缘点处会出现特征信息不足的现象, 在这些点上的匹配会更加困难, 这些匹配困难点严重影响场景流估计的整体准确度.

近几年的方法都是关注 2 个连续点云间对应的匹配关系来优化场景流的估计精度, FlowNet3D^[3] 在单尺度上获取匹配关系; HPLFlowNet^[4] 使用双边卷积层(bilateral convolutional layer, BCL), 在多个尺度上联合计算匹配关系^[5]; PointPWC-Net^[6] 在多个尺度上建立用于匹配的代价体(cost volume, CV)和重投影(warping)模块. 但这些方法仅考虑了点云间的匹配关系, 缺少优化匹配困难点的方式. 如图 1(a) 所示, 图片中的点为场景的一部分, 其中红色代表该点的端点误差(end point error, EPE)小于 0.05 m; 绿色代表该点的端点误差大于等于 0.05 m 且小于 0.3 m; 蓝色代表该点的端点误差大于等于 0.3 m. 在图 1(a) 虚线框中, PointPWC-Net 在一个局部邻域内(一个汽车整体)同时有匹配准确的红色点和匹配困难的蓝色点. 本文提出的基于邻域一致性的点云场景流传播更新方法(neighborhood consistency propagation update method, NCPUM)方法根据点云相邻点的相关性, 即属于源点云的场景流在足够小的邻域内很大程度上存在一致性, 将局部邻域中的准确场景流传播到匹配困难点上. 可以有效地减少匹配困难点场景流的误差, 提升整体准确度. 图 1(b) 为经过 NCPUM 方法优化后的效果, 可以看到在虚线框内的汽车点和匹配困难的蓝色点消失, 匹配较差的绿色点明显减少, 匹配准确的红色点明显增多.

具体来说, NCPUM 假设利用点云内相邻点的相关性使场景流估计具有邻域一致性, 通过置信度图传播更新提升整体场景流估计的准确度. 基于该假设,

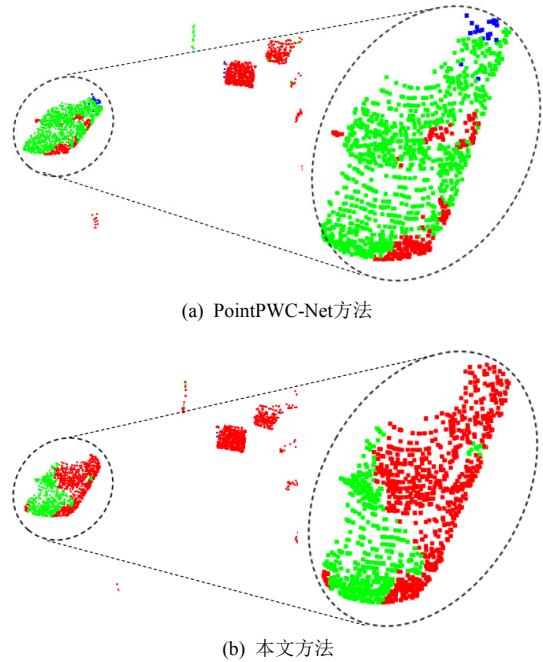


Fig. 1 Visual comparison of the two methods

图 1 2 种方法的可视化对比

NCPUM 设计了置信度预测模块和场景流传播模块, 对骨干网络输出的初始场景流预测置信度图, 经过场景流传播模块在具有一致性的邻域内将场景流从高置信度点集向低置信度点集传播, 改善邻域内匹配困难点的准确度. 本文的贡献有 2 方面:

1) 根据场景流的邻域一致性设计了场景流传播优化方法 NCPUM. 该方法使用场景流在局部邻域内传播的方式, 改善估计效果. NCPUM 的评估结果优于之前的工作, 证明了该方法的有效性.

2) NCPUM 在 Flyingthings3D 和 KITTI 数据集上的测试结果在准确度上达到国际先进水平, 并更换不同的骨干网络进行了测试, 验证了 NCPUM 对于不同的骨干网络都能明显提升其估计准确度.

1 相关工作

1.1 场景流估计

在 Vedula 等人^[7]工作中, 定义和介绍了场景流的概念, 之后许多工作^[8-12]在多种类型的数据集上进

行场景流的估计,随着最近几年基于点云的深度学习方法^[13-15]的发展,可以在点云上直接估计场景流。其中一个使用可学习的深度网络来估计点云上的场景流的方法是 FlowNet3D^[3],它将下采样的特征进行嵌入,得到点云间的运动信息,通过上采样方法回归得到对应点的场景流。FlowNet3D 只在单一尺度上进行了特征的嵌入,单一尺度的感受野无法在大尺度和小尺度运动上同时获得精确的估计结果。HPLFlowNet^[4] 使用双边卷积在多个尺度上联合计算匹配度,但限于内存使用限制无法在上采样过程中细化场景流。而 PointPWC-Net^[6] 遵循了光流估计的“由粗到细”(coarse to fine, CTF)的范式,在多个尺度的局部范围内使用 PointConv^[13] 建立用于匹配的代价体和重投影的模块。FLOT^[16] 通过最优传输(optimal transport),优化源点云和目标点云的匹配关系。这些关注于匹配关系的方法得到了很好的场景流估计结果。HALFlow^[17] 使用注意力机制,嵌入更多的位置信息,获得更准确的场景流估计结果。

文献[3-4,6,13,16-17]都是通过匹配连续点云间的特征回归出对应源点云的场景流,在匹配困难点处没有额外的优化措施。本文方法在源点云中根据相邻点的相关性,在邻域内改善匹配困难点的场景流,获得优于匹配方法的估计准确度。

1.2 邻域一致性

之前的场景流估计工作中都会注重在邻域内提取特征,根据提取到的特征来进行连续点云间的匹配^[3-4,6,17-19],回归出点云间的场景流。但这只是在提取的特征上包含了邻域信息,在邻域特征信息不足的点上会出现匹配困难的情况。在同样使用邻域信息进

行匹配的任务中^[20-21], LiteFlowNet3^[20] 根据局部光流一致性,在代价体上对邻域内的点进行优化,获得了相对于匹配方法更好的光流估计精度。受该想法的启发,我们合理假设在 2 个连续场景中,一个合适的邻域内的点共享相同的运动模式,因此在邻域内的场景流具有一致性。NCPUM 根据初始场景流显式的估计置信度,在邻域内的高置信度点向低置信度点进行传播更新。与现有方法不同的是, NCPUM 的更新操作是在场景流上而非在特征上,所依赖的也不是特征上的相关或者相似,而是点云邻域内场景流的一致性。

2 传播优化方法

NCPUM 从场景流邻域一致性假设出发,构建了一种对场景流在邻域内传播更新的优化方法。具体网络框架如图 2 所示,分别构建置信度预测模块和场景流传播模块实现 NCPUM 优化方法。首先是估计初始场景流的骨干网络,在得到初始场景流以及对应特征之后,送入置信度预测模块;然后在置信度预测模块中使用编码器-解码器(encoder-decoder)的网络结构,对输入的场景流进行置信度图的预测,置信度图用来表示各个场景流估计是否准确;最后在场景流传播模块中,根据预测得到的置信度图将场景流从高置信度点集向低置信度点集传播,更新低置信度点的场景流,降低匹配困难点对准确度的影响。

2.1 问题定义

场景流估计任务的最终目标是估计 2 个连续点云之间的运动矢量,因此定义 2 个连续的 3D 点云场

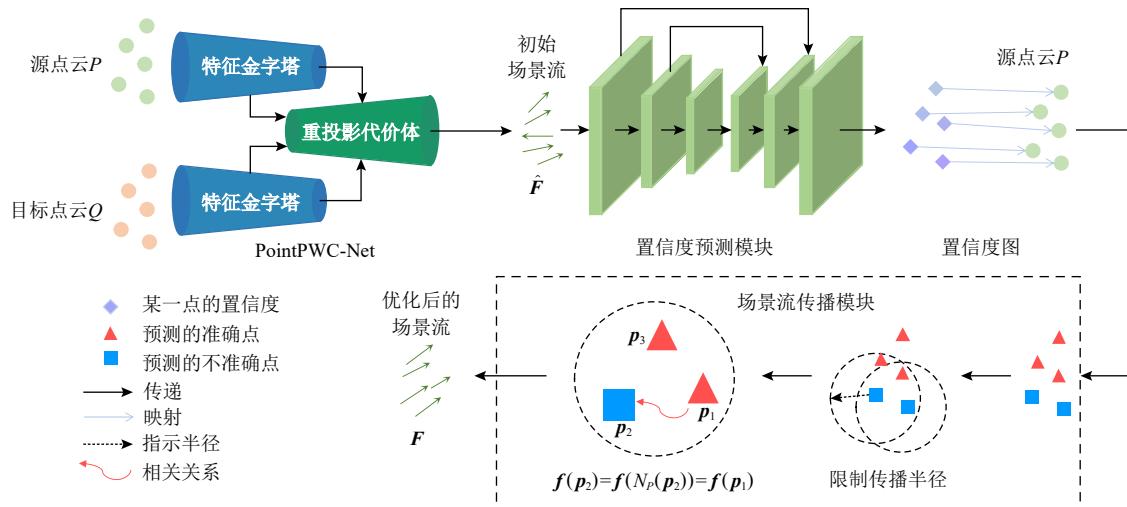


Fig. 2 Network structure

图 2 网络结构图

景：源点云 $P = \{\mathbf{x}_i | i = 1, 2, \dots, n_1\}$ ，和目标点云 $Q = \{\mathbf{y}_j | j = 1, 2, \dots, n_2\}$ ，其中 $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^3$ ，并且 i 并不必须与 j 相等。源点云 P 中的点运动到目标点云 Q 中的对应点的运动矢量场为 $F = (f_1 \dots f_{n_1})$ ，该运动矢量场即为最终估计的场景流。估计的场景流是基于源点云 P 的，因此场景流与源点云中的点一一对应。

2.2 估计初始场景流

在估计初始场景流时，使用的是 PointPWC-Net 作为骨干网络，该方法使用 2 个连续的点云作为输入，使用特征金字塔的结构，在每个分辨率尺度上都进行一次源点云 P 到目标点云 Q 的重投影，之后进行匹配度代价体的计算，代价体定义了逐点的匹配程度，PointPWC-Net 对代价体进行回归得到逐点的场景流。

在 PointPWC-Net 中，构建了 4 个尺度的特征金字塔，在得到 4 个尺度的点特征后，场景流的估计会从较粗的尺度进行，遵循由粗到细的规范。估计了当前尺度的场景流后，会上采样到更精细的尺度，将上采样的场景流对源点云进行重投影，在当前尺度上对重投影后的点云和对目标点云估计一个相对于上一个尺度场景流的残差，以完成对估计场景流的精细化。将整个重投影过程进行公式化定义：

$$P_w = \{\mathbf{p}_{w,i} = \mathbf{p}_i + \mathbf{f}_i | \mathbf{p}_i \in P, \mathbf{f}_i \in F^{up}\}_{i=1}^{n_1}, \quad (1)$$

其中 P 为源点云， P_w 为重投影后的点云， F^{up} 为从上一个尺度上采样的场景流。

在 PointPWC-Net 中，对 2 个点云以及对应的特征进行了代价体的构建。假设 $\mathbf{g}_i \in \mathbb{R}^C$ 是对应目标点云点 $\mathbf{p}_i \in P$ 的特征， $\mathbf{h}_j \in \mathbb{R}^C$ 是对应目标点云点 $\mathbf{q}_j \in Q$ 的特征，那么对应 2 个点之间的匹配度定义为：

$$Cost(\mathbf{p}_i, \mathbf{q}_j) = M(concat(\mathbf{g}_i, \mathbf{h}_j, \mathbf{q}_j - \mathbf{p}_i)), \quad (2)$$

使用多层感知机 (multilayer perceptron) M 将 2 点之间的潜在关系和点与点之间的距离串联后进行学习。在有了点对点的匹配度之后，将其组成当前尺度的代价体，PointPWC-Net 根据源点云点到目标点云邻域点的距离对代价体加权，即对于 1 个源点云的点 $\mathbf{p}_i \in P$ ，得到它在目标点云 Q 上的 1 个邻域 $N_Q(\mathbf{p}_i)$ ，再根据目标点云邻域中的每个点到源点云点的距离得到权重 C 。

$$C = \sum_{\mathbf{q}_j \in N_Q(\mathbf{p}_i)} W_Q(\mathbf{q}_j, \mathbf{p}_i) Cost(\mathbf{q}_j, \mathbf{p}_i), \quad (3)$$

$$W_Q(\mathbf{q}_j, \mathbf{p}_i) = M(\mathbf{q}_j - \mathbf{p}_i). \quad (4)$$

使用 PointPWC-Net 估计初始场景流时，沿用了多尺度监督损失，对应估计得到 4 个尺度的场景流，

对场景流真实值采样到同样的尺度，在对应的尺度上做不同权重 α 的 2 范数监督。

$$Loss_{sf} = \sum_{l=l_0}^L \alpha_l \sum_{\mathbf{p} \in P} \|\mathbf{F}^l(\mathbf{p}) - \mathbf{F}_{GR}^l(\mathbf{p})\|_2. \quad (5)$$

2.3 场景流置信度预测模块

在骨干网络输出初始场景流后，会经过置信度预测模块对初始场景流预测置信度图。该置信度定义为初始场景流相对于真实值的误差，即预测的误差值越小，表示该点在骨干网络中估计的初始场景流越准确，置信度值越高。置信度预测模块首先使用“编码器-解码器”的构造，以初始场景流的 3D 矢量信息作为输入，在编码器过程中进行点的下采样，可以扩大置信度预测模块的感受野，参考更多的相邻场景流推断置信度；然后在解码器的上采样过程中使用跳跃链接，串联编码过程中对应尺度的特征信息，为上采样提供更多精细尺度的特征，获得更精细的上采样结果，并且考虑骨干网络中输出的场景流特征；最后使用 sigmoid 函数输出 1 个 (0-1) 分布的置信度图，并将该置信度图用于之后的场景流传播模块中。

置信度预测模块使用的是有监督的训练方式，监督信息是初始场景流与场景流真实值的 2 范数二值化得到的先验分布图，该分布图为初始场景流相对于真实值的先验误差分布。设定阈值 θ ，当初始场景流与真实值的 2 范数小于 θ 时，设定为 0，否则设定为 1。由此得到的分布图为场景流先验的二分类分布图，用来监督置信度预测模块的输出。

$$GT_{conf} = \begin{cases} 0, |\mathbf{F} - \mathbf{GT}_{sf}| < \theta, \\ 1, |\mathbf{F} - \mathbf{GT}_{sf}| \geq \theta, \end{cases} \quad (6)$$

$$Loss_{conf} = -(GT_{conf} \times \ln confmap + (1 - GT_{conf}) \times \ln(1 - confmap)), \quad (7)$$

其中 $confmap$ 是置信度预测模块得到的置信图。 GT_{conf} 是场景流先验分布图，在式 (6) 中由初始场景流 F 和真实值 GT_{sf} 处理得到。估计置信度图的最后一层使用 sigmoid 函数将其转换为 0~1 之间的分布，因此就可以使用二分类交叉熵 (binary cross entropy, BCE) 进行监督。

2.4 场景流传播模块

经过场景流置信度图的预测，根据场景流置信度得到源点云中的高置信度点集和低置信度点集，由高置信度点集向低置信度点集进行限制半径的传播；根据邻域一致性的假设，如果高置信度点与低置信度点的传播半径不大于传播半径阈值，可以认为两点的场景流拥有一致性，可以使用高置信度点的场景流更新低置信度点的场景流。

$$\mathbf{p}_2 = KNN(\mathbf{p}_1), \quad \mathbf{p}_1, \mathbf{p}_2 \in P, \quad (8)$$

\mathbf{p}_1 和 \mathbf{p}_2 都属于源点云 P , 因为邻域一致性依赖于点云内相邻点的相关性, 所以距离最近的点最有可能存在场景流的邻域一致性, 式(8)中 KNN 为 K 最近邻(K -nearest neighbor)方法采样低置信度点 \mathbf{p}_1 在源点云的最近点.

$$f(\mathbf{p}_1) = f(\mathbf{p}_2), \quad \text{if } (\mathbf{p}_1 - \mathbf{p}_2) < \beta, \quad (9)$$

其中 \mathbf{p}_1 和 \mathbf{p}_2 分别为低置信度点和高置信度点, β 为传播半径阈值, 当两点的距离不大于传播半径时传播更新低置信度点的场景流. 这里传播半径阈值非常重要, 点云中的相邻点只有空间距离在一定阈值内才会具有相关性, 在点密度足够的情况下, 对于小邻域内的点的场景流具有一致性, 这个邻域的半径阈值设置不同的数值会影响到优化结果.

NCPUM 在优化初始场景流时, 会将反传的梯度在初始场景流处截断, 即训练置信度预测模块时不会影响到骨干网络 PointPWC-Net.

3 实验与结果分析

与之前的工作的训练过程^[4, 6, 16]类似, NCPUM 在合成数据集 Flyingthings3D^[22]上训练模型, 然后在 Flyingthings3D 和真实场景数据集 KITTI^[23]两个数据集上进行测试, 将测试结果与其他方法的结果在表 1 中进行对比.之所以使用这样的训练策略是因为很难在真实场景中确定出来一个场景流真实值, 这里使用的 KITTI 数据集只有 142 个场景, 而合成数据集有更大的数据量可供训练, 如 Flyingthings3D 数据集有

Table 1 Comparison of NCPUM and Other Methods

表 1 NCPUM 与其他方法的对比

数据集	方法	EPE/m	Acc3DS/%	Acc3DR/%	Outlier3D/%
Flyingthings3D	FlowNet3D ^[3]	0.114	41.3	77.0	60.2
	HPLFlowNet ^[4]	0.080	61.4	85.5	42.9
	PointPWC-Net ^[6]	0.059	73.8	92.8	34.2
	FLOT ^[16]	0.052	73.2	92.7	35.7
	HALFlow ^[17]	0.049	78.5	94.7	30.8
	NCPUM	0.060	76.1	93.9	30.7
KITTI	FlowNet3D ^[3]	0.177	37.4	66.8	52.7
	HPLFlowNet ^[4]	0.117	47.8	77.8	41.0
	PointPWC-Net ^[6]	0.069	72.8	88.8	26.5
	FLOT ^[16]	0.056	75.5	90.8	24.2
	HALFlow ^[17]	0.062	76.5	90.3	24.9
	NCPUM	0.070	78.1	91.5	22.3

注: 黑体数字表示最优结果.

19 640 对点云可以进行训练.在训练之前, 遵循 HPL-FlowNet 和 PointPWC-Net 的方式对数据进行了预处理, 点云场景中没有被遮挡的点.

在接下来的内容中, 将介绍 NCPUM 实现的细节, 然后将测试结果与之前的方法进行对比, 证明了 NCPUM 的有效性.并且我们认为 Flyingthings3D 数据集与 KITTI 数据集差异较大, 将 NCPUM 在 KITTI 数据集前 100 对数据上进行微调, 在后 42 对数据上进行了测试, 更换不同骨干网络微调的测试结果在表 3 中展示, 证明 NCPUM 基于邻域一致性假设的传播更新方法更适用于真实场景, 并进行了消融实验, 对传播半径阈值进行更多的对比实验.

3.1 网络实现细节

NCPUM 的训练设置与骨干网络 PointPWC-Net 相同, 对输入的点云采样到 8 192 个点, 为其构建 4 层的特征金字塔, 每一层的损失权重 α 设置为 $\alpha_0 = 0.02, \alpha_1 = 0.04, \alpha_2 = 0.08, \alpha_3 = 0.16$, 训练时的初始学习率设置为 0.001, 训练 800 个周期, 每 80 个周期学习率减少一半.在对 KITTI 进行微调时, 参数设置与 NCPUM 设置一致, 只是将训练周期减少为 400 个, 并且学习率每 40 个周期减少一半.

置信度预测模块设置了 3 个尺度, 分别为 2048, 512, 256 个点, 特征通道数都为 64.在经过 3 次下采样的特征提取之后得到了 3 个尺度的场景流特征, 再经过 3 次上采样将特征传播到原尺度.前 2 个尺度的上采样都同时考虑了前一尺度的特征和下采样时对应的特征; 后 1 个尺度的上采样考虑原尺度的场景流, 在上采样到原尺度之后, 与骨干网络输出的场景流特征串联在一起, 经过 2 个 1D 卷积, 再经过 sigmoid 得到 0~1 分布的置信度.

3.2 评价标准

本文沿用了场景流估计任务一直以来的评价标准^[3-4, 6, 18], 假设 \mathbf{F} 代表场景流估值, \mathbf{F}_{GT} 代表场景流真实值, 各个评价标准的定义为:

1) EPE3D. $\|\mathbf{F} - \mathbf{F}_{GT}\|_2$ 表示点云上的逐点误差平均值.

2) Acc3DS. $EPE3D < 0.05$ m 或者相对误差 $< 5\%$ 的点占比.

3) Acc3DR. $EPE3D < 0.1$ m 或者相对误差 $< 10\%$ 的点占比.

4) Outlier3D. $EPE3D > 0.3$ m 或者相对误差 $> 10\%$ 的点占比.

3.3 在 Flyingthings3D 上的训练和测试

Flyingthings3D 可用于训练的数据集共有 19 640

对点云,可用于测试的数据集有3 824对点云.与FlowNet3D、PointPWC-Net模型方法的设置相同,本文只使用深度小于35 m的点,大部分的前景物体都能包含在这个范围内.对每个点云采样得到8 192个点用于训练和测试,训练设置参数在3.1节网络实现细节中进行了介绍.

表1展示了在Flyingthings3D数据集上NCPUM在2个准确度指标和离群值占比指标上均优于骨干网络PointPWC-Net.尤其是Acc3DS指标,对比PointPWC-Net有2.3%的提升.但在EPE指标上略低于PointPWC-Net.其原因是在场景流传播模块中会使低置信度点的场景流与高置信度点的场景流相等,对于估计偏差大的点,会传播更新一个准确的场景流,对于估计误差

小的低置信度点,传播更新操作反而带来了精度上的扰动.但是准确度是统计 $EPE3D < 0.05$ m或者相对误差<5%的点占比,所以我们的方法能够优化估计偏差大的匹配困难点,提升准确度.本文在表2中整理出更新点的统计信息(图2中 \hat{F} 与 F 的比较),其中包含更新点占全部点的比例(更新点占比)、更新点中精度提升的占比(改进点占比)、更新点中精度下降的占比(扰动点占比)、精度提升的平均值(改进均值)、精度下降的平均值(扰动均值).可以看到,有一半以上的点其实是产生了扰动,并且产生扰动的均值大于改进的均值,因此在精度上NCPUM确实会产生一定的扰动,但是在准确度指标和离群值占比指标上有大幅提升.

Table 2 Statistical Information of Update Points on Flyingthings3D Dataset

表2 Flyingthings3D数据集更新点统计信息

方法	更新点占比/%	改进点占比/%	扰动点占比/%	改进均值/m	扰动均值/m	EPE/m	Acc3DS/%	Acc3DR/%	Outlier3D/%
PointPWC-Net						0.059	73.8	92.8	34.2
NCPUM	9.73	46.98	53.02	0.003	0.004	0.060	76.1	93.9	30.7

注:黑体数字表示最优结果.

3.4 在KITTI上测试和微调

KITTI Scene Flow 2015数据集包括200个训练场景和200个测试场景,遵循之前工作的方法,在具有真实值并且可用的142对点云数据集上进行测试,并且与之前的方法保持一致^[4,6],删除了点云中高度小于0.3 m的地面点.在KITTI数据集上,先使用在Flyingthings3D数据集上训练的模型直接对KITTI数据集进行测试而不经过任何微调,以评估NCPUM的泛化能力.从表1中可以看到NCPUM在Acc3DS和Acc3DR评价指标上优于PointPWC-Net,在Acc3DS指标上有5.3%的提升,在Acc3DR指标上有2.7%的提升,提升幅度大于Flyingthings3D数据集,表现出优秀的泛化性能.不经过微调而在真实数据集上表现更加优秀的原因在于真实数据集KITTI的点云数据物体之间的距离大于Flyingthings3D数据集中物体之间的距离,NCPUM的邻域一致性假设更适用于这种数据特性,所以NCPUM在真实场景上会有更加优秀的表现.本文统计了表1中FLOT和NCPUM的场景流估计时间,FLOT在一秒钟内可以处理2.15对连续点云,而NCPUM在一秒钟内可以处理5.09对连续点云,NCPUM处理速度约是FLOT的2.37倍.在真实使用场景中,准确场景流在总体场景流中的占比比场景流的绝对平均误差值更有意义,拥有更多的准确场景流代表该方法为真实应用带来更好的稳定性.

NCPUM在Acc3DS和Acc3DR准确度指标上都有可观的提升,尤其在真实数据集上的场景流Acc3DS指标超过PointPwc的Acc3DS指标7.28%,超过HALFlow的Acc3DS最佳结果2.09%,对比之前的方法,NCPUM的处理速度和准确度表现出更大的应用潜力.

因为Flyingthings3D数据集和KITTI数据集存在较大的差异,直接使用在Flyingthings3D数据集上预训练的模型在KITTI数据集上测试并不能完全展示NCPUM在KITTI数据集上的性能.所以本文将KITTI数据集拆分,使用前100对场景微调NCPUM,并在剩余的42对场景上进行测试.分别将FlowNet3D,PointPWC-Net,FLOT 3种骨干网络在KITTI数据集上进行微调,然后进行NCPUM的微调,将微调后的3种骨干网络做对比.在微调之后,3种骨干网络的NCPUM可以获得更好的效果,如表3所示,微调后的NCPUM对比微调后对应的骨干网络,在4个评价标准上都有提升,与泛化能力测试结果不同的是,NCPUM在EPE指标上也有了一定的提升,我们认为,Flyingthings3D是虚拟数据集,场景中物体间的距离较小,对某一物体边缘的低置信度点传播更新,可能采样到距离较近的其他物体上的准确点,而不是同一物体中距离较远的准确点,例如图3所示,绿色点为更新场景流的低置信度点,红色的点是传播场景流的高置信度点,黄色的连线为传播更新关系.在图3(a)

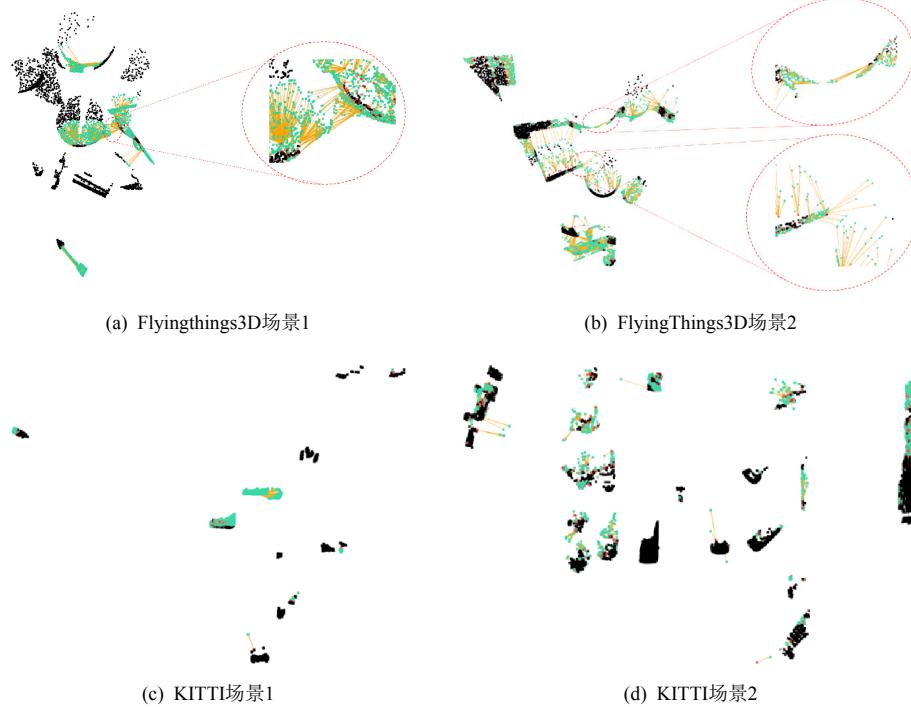


Fig. 3 Comparison of Flyingthings3D and KITTI on replacement details

图3 Flyingthings3D 与 KITTI 替换细节对比

和图3(b)中都出现了采样到其他物体的现象; KITTI 是真实数据集, 物体之间的距离较大, 如图3(c)所示, 不容易出现采样错误的情况, 只有在如图3(d)中的远端离群点上可能出现采样不准确的情况, 因此 KITTI 相较于 Flyingthings3D 数据集更容易符合邻域一致性的假设.

Table 3 Fine Tuning and Testing on KITTI Dataset

表3 在 KITTI 数据集上微调测试

骨干网络	方法	EPE/m	Acc3DS/%	Acc3DR/%	Outlier3D/%
FlowNet3D ^[3]	w/o ft	0.173	27.6	60.9	64.9
	w ft	0.102	33.6	70.3	44.9
	NCPUM	0.094	38.6	74.1	37.1
PointPWC-Net ^[6]	w/o ft	0.069	72.8	88.8	26.5
	w ft	0.045	82.7	95.1	25.3
	NCPUM	0.043	87.5	96.9	24.3
FLOT ^[16]	w/o ft	0.056	75.5	90.8	24.2
	w ft	0.029	89.4	96.8	18.0
	NCPUM	0.028	89.9	97.0	17.5

注: 黑体数字表示最优结果; w/o 表示 with/without; ft 表示 fine-funing.

3.5 传播半径阈值设置的影响

因为 NCPUM 是基于邻域一致性假设进行构建的, 因此传播半径阈值设置十分重要, 不同的半径阈值设置带来的效果是不一样的, 甚至不合适的半径阈值会降低 NCPUM 优化效果. 当半径阈值设置过大

时, 高置信度点的场景流会传播到不具有一致性的低置信度点上, 出现扰动; 当半径设置过小时, 只会有少部分低置信度点会被更新. 数据集的不同也会影响到传播半径的设置, 对比虚拟数据集, 真实数据集因为物体间的距离更大, 更容易设置一个合适的传播半径进行传播, 这也是 NCPUM 泛化到真实数据集表现更好的原因. 表4 对 2 个数据集上设置的不同传播半径进行对比, NCPUM 在 Flyingthings3D 数据集上的半径设置为 0.4 时达到最好效果, 而在 KITTI 数据集上的半径设置为 3.0 时达到最好效果. 这个数据的差异表现出在真实场景上对传播的约束更加小, 传播更新可以影响到更多的点, 从而带来更好的改进效果.

Table 4 NCPUM Tested with Different Radius Threshold

表4 NCPUM 在不同半径阈值下的测试

数据集	半径阈值	EPE/m	Acc3DS/%	Acc3DR/%	Outlier3D/%
Flyingthings3D	1.0	0.062	75.2	93.5	31.7
	0.4	0.060	76.1	93.9	30.7
	0.2	0.060	76.0	93.8	30.7
KITTI	5.0	0.054	85.8	95.9	24.1
	3.0	0.043	87.5	96.9	24.3
	1.0	0.043	85.0	95.4	25.5

注: 黑体数字表示最优结果.

3.6 在不同骨干网络上测试

为了证明 NCPUM 方法的泛化性能,本文尝试在不同的骨干网络上进行优化.我们分别以 FlowNet3D, PointPWC-Net, FLOT 为骨干网络构建使用置信度预测模块和场景流传播模块,在 KITTI 数据集上进行微调和使用 NCPUM 优化方法改进.测试结果如表 3 所示,在对 FlowNet3D, PointPWC-Net, FLOT 方法使用 NCPUM 优化方法后,4 个指标上都有明显的提升,展示了 NCPUM 优化方法对不同骨干网络的泛化性.

在图 4 中可视化了 NCPUM 的传播更新过程,绿

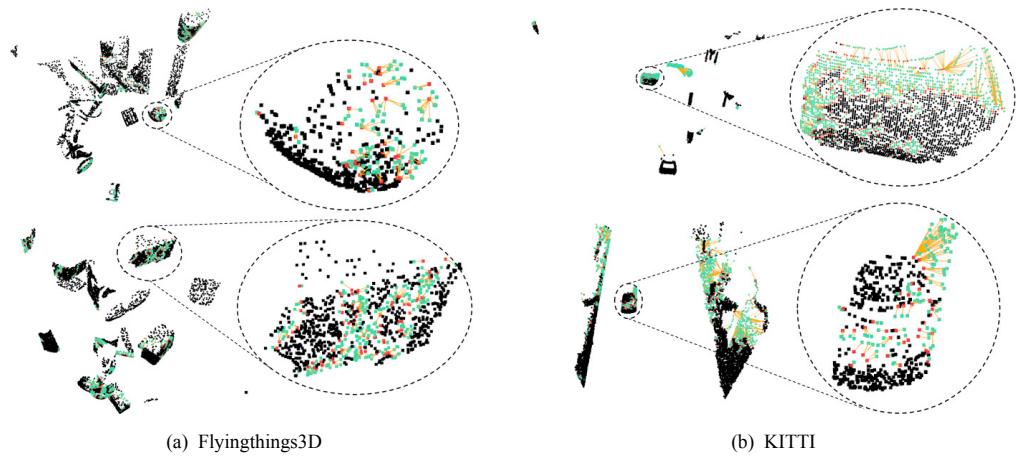


Fig. 4 Visualization of replacement details

图 4 替换细节可视化

4 结 论

本文提出了一种根据邻域一致性的传播更新方法 NCPUM 来优化估计场景流的精度.该方法通过置信度预测模块对骨干网络估计出的初始场景流预测置信度图来判断场景流准确度;然后通过场景流传播模块在有限制的传播半径内从匹配困难点寻找匹配准确点,将匹配准确点的场景流传播到匹配困难点上,以提高场景流的估计精度.NCPUM 在不同的数据集 Flyingthings3D 和 KITTI 上都体现出了优于之前工作的性能.并且通过在真实数据集上的微调实验和不同传播半径的实验展现出 NCPUM 在真实场景数据有更加优秀的表现.

作者贡献声明: 郑晗负责提出模型、编写代码、实施实验过程和论文撰写;王宁负责修订和完善论文;马新柱负责数据分析和数据解释;张宏负责理论和实验设计;王智慧负责理论设计;李豪杰提供论文写作指导.

色点为更新场景流的低置信度点,红色点是传播场景流的高置信度点,黄色的连线为传播更新关系.可以看到 KITTI 数据集中具有一致性的汽车表面会出现估计不准确的绿色低置信度点,这些点更多是位于距离激光雷达更远的低密度点和邻域信息单一的边缘点上,若只关注连续点云间匹配的方法容易在这些点上出现较大的误差, NCPUM 对匹配困难点寻找一个匹配准确点进行更新,和相邻的准确点保持一致,从而提高整体的估计准确度;同时传播过程要限制传播半径阈值,避免引入扰动.

参 考 文 献

- [1] Ouyang Wanli, Zeng Xingyu, Wang Xiaogang, et al. DeepID-Net: Deformable deep convolutional neural networks for object detection[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2016, 39(7): 1320–1334
- [2] Liao Ruijie, Yang Shaofa, Meng Wenxia, et al. SegGraph: An algorithm for loop-closure detection in outdoor scenes using 3D point clouds[J]. *Journal of Computer Research and Development*, 2019, 56(2): 338–348 (in Chinese)
(廖瑞杰, 杨绍发, 孟文霞, 等. SegGraph: 室外场景三维点云闭环检测算法[J]. *计算机研究与发展*, 2019, 56(2): 338–348)
- [3] Liu Xingyu, Qi C R, Guibas L J. FlowNet3D: Learning scene flow in 3D point clouds[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2019: 529–537
- [4] Gu Xiuye, Wang Yijie, Wu Chongruo, et al. HPLFlowNet: Hierarchical permutohedral lattice FlowNet for scene flow estimation on large-scale point clouds[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2019: 3254–3263
- [5] Zhang Yanyong, Zhang Sha, Zhang Yu, et al. Multi-modality fusion perception and computing in autonomous driving[J]. *Journal of Computer Research and Development*, 2020, 57(9): 1781–1799 (in Chinese)

- (张燕咏, 张莎, 张昱, 等. 基于多模态融合的自动驾驶感知及计算[J]. 计算机研究与发展, 2020, 57(9): 1781–1799)
- [6] Wu Wenxuan, Wang Zhiyuan, Li Zhuwen, et al. PointPWC-Net: Cost volume on point clouds for (self-)supervised scene flow estimation[C] //Proc of European Conf on Computer Vision. Piscataway, NJ: IEEE, 2020: 88–107
- [7] Vedula S, Baker S, Rander P, et al. Three-dimensional scene flow[C] //Proc of Int Conf on Computer Vision. Piscataway, NJ: IEEE, 1999: 722–729
- [8] Jiang Huaizu, Sun Deqing, Jampani V, et al. SENSE: A shared encoder network for scene-flow estimation[C] //Proc of Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 3194–3203
- [9] Teed Z, Deng Jia. RAFT-3D: Scene flow using rigid-motion embeddings[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2021: 8375–8384
- [10] Menze M, Geiger A. Object scene flow for autonomous vehicles[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2015: 3061–3070
- [11] Ma Weichiu, Wang Shenlong, Hu Rui, et al. Deep rigid instance scene flow[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2019: 3614–3622
- [12] Quiroga J, Brox T, F Devernay, et al. Dense semi-rigid scene flow estimation from RGBD Images[C] //Proc of European Conf on Computer Vision. Piscataway, NJ: IEEE, 2014: 567–582
- [13] Wu Wenxuan, Qi Zhongang, Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2019: 9621–9630
- [14] Qi C R, Su Hao, Mo Kaichun, et al. PointNet: Deep learning on point sets for 3D classification and segmentation[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2017: 77–85
- [15] Qi C R, Yi Li, Su Hao, et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space[C] //Proc of Neural Information Processing Systems. Cambridge, MA: MIT Press, 2017: 5099–5108
- [16] Puy G, Boulch A, Marlet R. FLOT: Scene flow on point clouds guided by optimal transport[C] //Proc of European Conf on Computer Vision. Piscataway, NJ: IEEE, 2020: 527–544
- [17] Wang Guangming, Wu Xinrui, Liu Zhe, et al. Hierarchical attention learning of scene flow in 3D point clouds[J]. IEEE Transactions Image Process, 2021(30): 5168–5181
- [18] Gojcic Z, Litany O, Wieser A, et al. Weakly supervised learning of rigid 3D scene flow[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2021: 5692–5703
- [19] Chen Yuhua, Gool L V, Schmid C, et al. Consistency guided scene flow estimation[C] //Proc of European Conf on Computer Vision. Piscataway, NJ: IEEE, 2020: 125–141
- [20] Hui T W, Chen C L. LiteFlowNet3: Resolving correspondence ambiguity for more accurate optical flow estimation[C] //Proc of European Conf on Computer Vision. Piscataway, NJ: IEEE, 2020: 169–184
- [21] Li Tong, Ma Wei, Xu Shibiao, et al. Task-adaptive end-to-end networks for stereo matching[J]. Journal of Computer Research and Development, 2020, 57(7): 1531–1538 (in Chinese)

(李瞳, 马伟, 徐士彪, 等. 适应立体匹配任务的端到端深度网络[J]. 计算机研究与发展, 2020, 57(7): 1531–1538)

- [22] Mayer N, Ilg E, Hausser P, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation[C] //Proc of Computer Vision & Pattern Recognition. Piscataway, NJ: IEEE, 2016: 4040–4048
- [23] Menze M, Heipke C, Geiger A. Joint 3D estimation of vehicles and scene flow[J]. International Society for Photogrammetry and Remote Sensing Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015, II-3/W5: 427–434



Zheng Han, born in 1997. Master. His main research interests include 3D vision and 3D scene understanding.

郑晗, 1997年生. 硕士. 主要研究方向为3D视觉和3D场景感知.



Wang Ning, born in 1994. PhD candidate. Her main research interests include artificial intelligence application and image processing.

王宁, 1994年生. 博士研究生. 主要研究方向为人工智能应用和图像处理.



Ma Xinzhu, born in 1994. PhD candidate. His research interests include image-based 3D scene understanding and 3D object detection.

马新柱, 1994年生. 博士研究生. 主要研究方向为基于图像的3D场景理解和3D目标检测.



Zhang Hong, born in 1993. PhD candidate. Her main research interests include depth estimation and stereo matching.

张宏, 1993年生. 博士研究生. 主要研究方向为深度估计和立体匹配.



Wang Zhihui, born in 1982. PhD, professor. Her main research interests include information hiding and image compression.

王智慧, 1982年生. 博士, 教授. 主要研究方向为信息隐藏和图像压缩.



Li Haojie, born in 1973. PhD, professor. Member of CCF. His main research interests include social media computing and multimedia information retrieval.

李豪杰, 1973年生. 博士, 教授. CCF会员. 主要研究方向为社交媒体计算和多媒体信息检索.