Notes on Point-Based DL in 3D Point Cloud

Matthew Mo

2020 年 8 月 4 日

目录

1 背景

1.1 Datasets

- 3d 形状分类
  - synthetic/real-world
- 3d 对象检测/追踪
  - indoor/outdoor urban
- 3d 点云分割
  - sensors: MLS/ALS/TLS/RGBD Cams/Other 3D Scanners

1.2 Eval. Metrics

- 3d 形状分类
  - Overall Accuracy(OA)
  - Mean Class Accu.(mAcc)
- 3d 对象检测
  - Average Precision(AP). 是在 precion-recall 曲线下的面积
- 3d 对象追踪
  - Precision
  - Sucess
  - Multi Obj. Ãverage Multi-Object Tracking Accuracy(AMOTA)
  - Multi Obj. Ãverage Multi-Object Tracking Precision(AMOTP)
- 3d 点云分割
  - OA
  - mean Intersection over Union(mIoU)

- mAcc
- mean Avergae Precision(mAp)

2 3D 形状分类

常常学习每个点的 embedding, 最后在点云上用某种 aggregation 来得到最终结果.

- Multi-view based: 将点云投影在平面上
- Vol. based: 将点云转化为 3D 体积元表示
- Point based: 直接在点云上操作, 无需体素化/投影 ⇒ 本文的主要内容!

2.1 多视野 (multi-view) 方法

- MVCNN[1]: 先驱性工作, 简单地把 multiview 特征 pooling 得到 global desciptor ⇒info loss
- MHBN[2]: 使用 harmonic 双线性 pooling 得到紧的表示
- (Yang et al. [3]) 利用 relation-network 考虑 view 之间的关系
- View-GCN[4]把不同的 view 考虑成图的顶点, 并使用了一个有向图

2.2 Volumetric Methods

- VoxNet[5]
- ShapeNets[6]: 基于深度信度网络, 学习 shape 的分布, 用在 3d 中有点的概率表示

以上两种方法难以规模化, 应为是立方复杂度的. 之后, 结构性和紧的结构被引入来降低复杂度 (如 octree)

- OctNet[7]是第一个结构性划分点云的方法, 使用了混杂格点-八叉树结构, 使用位向量高效编码.
- (Wang et al.[8]) 是基于八叉树的 CNN 的 3d 形状分类方法. 在最细粒度的叶子上采样的 Average Normal Vec. 送到网络中,3DCNN 在对应网络象限计算.

[1] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multiview convolutional neural networks for 3D shape recognition," in ICCV, 2015.

[2] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in CVPR, 2018.

[3] Z. Yang and L. Wang, "Learning relationships for multi-view 3D object recognition," in ICCV, 2019.

[4] X. Wei, R. Yu, and J. Sun, "View-gcn: View-based graph convolutional network for 3D shape analysis," in CVPR, 2020.

[5] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in IROS, 2015.

[6] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapeNets: A deep representation for volumetric shapes," in CVPR, 2015.

[7] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in CVPR, 2017

[8] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "OCNN: Octree-based convolutional neural networks for 3D shape analysis," ACM TOG, 2017.

- Le et al.[9]聚合了点-格点表示, 在每个格点单元里常数个点被采样, 进而送进 3dCNN.
- Ben-Shabat et al.[10] 把点云用 3DmFV(3d-modified Fischer Vec.) 表示, 并用 CNN 计算表示.

2.3 点方法

1. Pointwise MLP
2. Conv. Based
3. Graph Based
4. Hierarchical Data Structure Based
5. Other

2.3.1 逐点 MLP

- 作为先驱性工作,PointNet[11]把 $N \times 3$ 的点云坐标送到 MLP 中 (共享权值), 然后得到 features $N \times M$,max-pooling 得到 feature $1 \times M$, 通过一个对称的函数来得到变换不变性.
- Deep Sets[12]达到平移不变性, 通过 features 加合起来, 并应用 non-linearity.
- PointNet++[13]利用了 PointNet 无法获取的局域关系信息: 采用有层次结构的 network.

  由于 PointNet 的简洁性, 许多方法都基于它.
- MoNet 采用了有限个 patch.
- PATs(Point Attention Transformers[14]) 把每个点用绝对位置表示, 把关系用相对于这个点的位置表示, 进而学习一个高维表示,GSA(Graph Shuffle Attention) 随后被用于捕捉点之间的关系, 最后一个变换不变的, 可微的, 可学习的端到端 GSS(Gumbel Subset Sampling) 层被用于学习结构特征.
- 基于 PN++,PointWeb[15]利用了点的邻域上下文, 通过 AFA(Adaptive Feature Adjusting).
- (Duan et al.)SRN[16](Structural Relation Net) 用于在局部结构之间学习结构性关系特征.

[9]T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in CVPR, 2018.

[10]Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3D point cloud classification and segmentation using 3D modified fisher vector representation for convolutional neural networks," arXiv preprint arXiv:1711.08241, 2017.

[11]C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in CVPR,2017.

[12]M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in NeurIPS, 2017.

[13]C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deephierarchical feature learning on point sets in a metric space," in NeurIPS, 2017

[14]J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subsetsampling," in CVPR, 2019.

[15]J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subsetsampling," in CVPR, 2019.

[16]Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian, "Structural relational reasoning of point clouds," in CVPR, 2019.

- Lin et al.[17]通过在输入/函数两个空间上记录 loopkup table 加速推理 (32 倍速)
- SRINet 先利用投影得到旋转不变的表示, 在利用 PN 估价得到全局 feature, 再通过基于图的聚合得到局部特征
- PointASNL[18]利用了 Adaptive Sampling(AS) 模块来动态调整由 FPS 算法采样到的点, 并提出了 local-nonlocal(L-NL) 模块来捕捉采样的点的局部-长程依赖.

2.4 卷积方法

相比于 2D 网格结构的卷积, 点云卷积更难设计.

2.4.1 3D 连续卷积方法

在连续空间上做卷积 (嵌入在欧氏空间或在流形上?), 权重由到邻域中心点的位置关系 (距离 + 取向, 或者说坐标) 决定.

3D 卷积 $\Rightarrow$ 子集上的带权和,

- RS-CNN[62] 的核心卷积层 RSConv: 输入为一个几何邻域, 用 MLP 来学习低阶关系 (e.g. 欧式坐标, 相对位置) 到权重的关系.
- [63], 核元素 (邻域中心?) 在一个单位空间中随机选择 (s.t. 密度近似相等), 一个基于 MLP 的函数用于建立核的位置和点云的关系.
- DensePoint[64], 卷积使用单层感知机 (SLP)+ 非线性激活. 特征在之前所有层的特征拼接 (concat) 上学习.

65 , 一个 rigid/deformable 卷积算子 KPConv(Kernel Point Convolution), 使用一组可学习的核元素.

- ConvPoint[66], 卷积核分为空间-特征两部分, 空间部分在单位球中随机选择, 权函数由 MLP 学习得到.

使用既有算法实现卷积,

- PointConv[67], 卷积定义为 (关于重要度采样的)3D 卷积的 Monte-Carlo 估计. 卷积核通过权函数 (MLP 学习) 和密度估计组成 (核化密度估计 (kernelized density est.)+MLP 学习). 为了降低 TC/SC,3D 卷积简化为: 矩阵乘法和 2D 卷积, 降低内存消耗 64 倍.
- MCCNN[68], 卷积依赖于对样本密度函数 (MLP 学习) 的 MC 估计.Poisson Disk 采样接着用于构建点云结构.

[17]H. Lin, Z. Xiao, Y. Tan, H. Chao, and S. Ding, "Justlookup: One millisecond deep feature extraction for point clouds by lookup tables," in ICME, 2019.

[18]X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in CVPR, 2020.

- SpiderCNN[69], SpiderConv 的卷积是阶跃函数和在 kNN 上的 Taylor 展开的乘积. 阶跃函数捕捉粗粒度几何关系 (通过编码局部测地距离), 泰勒展开捕捉内蕴几何关系, 通过在立方体顶点上拟合函数.
- PCNN[70] 基于 RBF.

一些算法为了解决旋转不变性而提出,

- [71], 3D Spherical CNN, 学习旋转协变的表示, 使用多值球函数作为输入. 局部化滤波器通过在带锚点的球谐谱上的参数化得到.
- [72], Tensor Field Net., 卷积是可学习的及径向函数和球谐函数的乘积 (分离系数?).
- [73], 基于球面交叉互相关 (cross-correlation), 使用推广的 FFT 实现.
- 基于 PCNN,SPHNet[74] 通过包含球谐核函数在容量函数的卷积过程中.

为了加速计算

- Flex-Conv. [75], 定义卷积为在 kNN 上的标量积, 可用 CUDA 加速.

2.4.2 3D 离散卷积方法

3D 卷积 $\Rightarrow$ 子集嵌入空间的网格上的带权和,

- [76] 把非均匀的点云转化为了均匀的格点, 并在格点上定义卷积核. 给定一点, 计算上一层在同一格网上的点的平均特征, 然后这些结果加权求和得到这一层的结果.
- [77] 学习了一个球面函数, 通过把球形邻域分割成块, 并且应用可学习的权重. 一个点的卷积核的输出 = 邻接点的值的权重和 + 激活函数
- GeoConv[78], 点与邻接点的关系显式通过六个基写出. 每个方向上的权分别学习, 并且每个方向上按角度再加权.
- PointConv[79], 将输入的点变为隐含核潜在可排序的, 通过 χ-conv 变换 (MLP 实现), 在应用经典的卷积算子.
- InterConv[80], 在邻接离散卷积**核-权**坐标上内插.
- RIConv[81], 是旋转不变的, 将低阶旋转不变几何特征作为输入, 通过 inning(翻译为块化?) 转化为 1D 卷积.
- A-CNN[82], 定义 annular conv., 通过在每个邻接点环上作与核大小相符的循环乘积.

为了降低 TC/SC

- Rectified Phase Volumn(ReLPV) 块 [83], 使用短时傅里叶变换 (STFT) 来从 3D 邻域提取 phase, 大大降低参数数量.
- SFCNN[84], 点云投影到正二十面体晶格上, 带有对齐的球形坐标. 通过把顶点上与邻接点上的特征 concat 后 conv.-maxpool.-conv. 来达成卷积.$\Rightarrow$ 旋转和变换不变性.

2.5 图方法

输入 ⇒ 图构建 ⇒ 图特征提取 ⇒ 输出

2.5.1 时域/空域图方法

边特征常常是两点间的几何属性, 点特征常常是坐标/颜色/激光强度等等. 卷积常常是空间邻域上的 MLP.

- [85] 点作为图顶点, 作有向邻接图.ECC(Edge-Conditioned Conv.)+VoxelGrid[86] 图粗化.
- DGCNN[87], 图在特征空间中构建, 并且每层动态更新.EdgeConv 中 MLP 用于学习边特征.
- 更进一步,LDGCNN[88] 去除了 DGCNN 中的 transformer 网络, 链接其不同层的结构特征, 以降低模型大小和提速.
- FoldingNet[89] 作为图 AE 网络, 用 concat 的向量化局部协方差矩阵和坐标作为输入.
- 由 Inception 和 DGCNN 启发,[91] 提出了多任务无监督 AE 用于提取特征. 其中编码器在多尺度图 (multi-scale graph) 上构建, 解码器用三个无监督任务 (聚类, 自监督分类, 重建) 构建, 使用分离的所任务 loss.
- Dynamic Points Agglomeration Module(DPAM, [92]), 通过图卷积来简化聚集操作 (取样/组队/池化) 为一步 ⇒ 聚集矩阵和点特征矩阵的乘积. 基于 PointNet 和多层 DPAM 建立了一种学习结构, 相较于 PointNet++, 动态利用了点关系和在语义空间聚集点.
- 为了利用局部几何结构,KCNet[93] 基于核互相关学习特征. 首先, 一个可学习的点集合, 称为核, 用于标识几何结构. 然后在每个点, 计算核和邻域的亲和度.
- G3D[94], 卷积定义为邻接矩阵的某个多项式, 图粗化是用粗化矩阵乘到 Laplacian 和定点特征矩阵上.
- ClusterNet[95] 使用严格旋转不变的模块在 kNN 图上找旋转不变的 feature. 通过无监督层次聚类 +ward-linkage criteria 建立点云的结构. 每个子聚类中的特征通过一个 EdgeConv 块后用 max-pooling 聚合.
- 为了解决 TC 问题, [97] 提出混合基于容积-点的方法.ModelNet 上的实验 (本文提出的)Grid-GCN 有 5x 加速.

2.5.2 频域图方法

卷积定义为频域的滤波器.

- RGCNN[100] 做一个全连接图, 每层更新 Laplacian, 损失函数带有邻域相似先验.
- 为了解决分散的图拓扑带来的问题,SGC-LL 层 (in AGCN[101]) 用于**学习**一个点邻接度量 (Mahalanobis dist.), 总的距离用高斯核和这个学习到的度量一起决定.

- HGNN[102], 在超图 (hypergraph) 上建立卷积 (谱卷积).
- LocalSpecGCN[103] 是 kNN 图上的端到端谱卷积网络. 无需 Laplacian 离线计算 (显式计算?) 和图粗化.
- PointGCN[104],kNN 图 + 高斯核边权 +ChebNet, 使用了全局池化-多分辨率池化, 来捕捉全局-局部信息.
- 3DTI-Net[105] kNN 图上的谱卷积. 使用相对欧式和方向距离来达到几何变换不变性.

2.5.3 层次数据结构方法

这些方法基于某些层次数据结构 (八叉树 (octree),kd-tree). 点特征按照结点层次向上学习.

- octree 引导的 CNN[77], 使用球面卷积核, 八叉树每一层对应一层卷积, 结点值是子节点值的平均.
- 不像 OctNet,Kd-Net[106] 基于多个 kd-tree, 它们在每次分割上有不同的方向. 基于自底向上原则, 非叶节点的特征由 MLP 给出, 并且在每层的相同结点分割类型上共享权重.
- 3DContextNet[107] 使用标准平衡 kd-tree. 点特征先在 local cue 和 global context cue 上用 MLP 学习. 然后非叶节点特征由子节点 MLP 和 max-pooling 得到.
- SO-Net[108],kNN by 点到节点搜索. 使用一个修改的, 轮换不变的 SOM 来建模点云空间分布. 点特征来自正则化的点-结点坐标 + 一系列 fc 层.SOM 中每个节点特征有和它相连的点的按 channel max-pooling 得到. 最终的特征类似于 PointNet 来学习.

2.5.4 其它方法

- RBFNet[113] 使用稀疏分布的 RBF 核 (位置核尺寸可学习), 并聚合特征.
- 3DPointCapsNet[112] MLP 学习点的独立特征, 使用逐点 MLP 和卷积层, 并用多个 max-pooling 来提取全局隐含表示. (?) 基于无监督动态 routing(这是个基于 capsule 的 net!), 接着学习 latent capsules.
- PointDAN[116], 端到端无监督 domain-adaptive net.
- [117], 使用自监督方法重建点云来捕捉语义信息, 且其部分被随机重排.
- PointAugment[118], 一个自动数据增强框架. 具体来讲, 对于每个输入样本, 学习按形状的变换和按点的替换. 网络根据优化它的增强器 (augmentor) 和分类器训练.
- ShapeContextNet[109], 组合了 affinity point sel.+ 紧特征聚合于软对齐操作 (soft align. op.), 使用点积形式的自注意力机制 [120].
- 为了处理噪声和 occuluion(遮挡?), [121] 使用了手写的点配对函数 (基于 4D 旋转不变 descriptor), 得到 4D 的 CNN.

- [122] 首先在单位球里均匀采样一个基点集合, 然后把点云作为到基点集合的距离来编码 (相当于多极坐标). 这样点云中就变成了一个短定长向, 在用传统方法处理.
- RCNet[115], RNN+2D-CNN 以建立一个轮换不变的网络. 点云分割为平行的束 (beams), 然后按某特定维度排序, 然后每个束送到一个共享的 RNN 中. 学到的特征再传到一个 2DCNN 中进行层级特征聚合.RCNet-E 采用聚合多个不同方向和不同分割的 RCNet 模型来提高性能.
- Point2Sequences[114] 也基于 RNN, 捕捉不同局部区域间的互相关. 它考虑各局部区域在多尺度上学习到特征, 并且视为序列送到基于 RNN 的 E-D(编码器-解码器) 结构上来聚合局部区域特征.

一些方法同时在 3D 点云和 2D 图像中学习,

- PVNet[110], 从 multi-view 图像中提取的高阶特征通过一个嵌入网络 (embedding net.) 被投影到点云的子空间里, 并通过一个软注意力 (soft attention)mask 和点云特征嵌合 (fuse). 最后, 一个 residual 层应用在在嵌合特征和多视图特征上来进行形状分类.
- PVRNet[111], 通过一个关系打分模块, 利用点云和 2D 视图们的关系.

2.6 总结

Model10/40 数据集是最常用的 3D 形状分类数据集, 一些观察:

- 逐点 MLP 常常是其他网络的基本模块.
- 基于卷积的方法能在不规则 3D 点云上达到很好的性能. 我们应该更多关注离散/连续卷积网络.
- 由于其内蕴的处理不规则数据的能力,GNN 这几年用得多了起来. 但是谱域方法难以跨域是一个大问题.

3 3D 对象检测和追踪

3.1 3D 对象检测

输入:3D 场景点云, 输出: 有取向的 3D 范围盒 (bounding box). 算法可分为

1. 基于区域提案的 (region proposal-based): 首先提出一些可能的区域 (称为提案), 再每区域地提取特征, 来决定 label. 根据提案生成方法, 分为: 基于多视图的, 基于分割 (segmentation) 的, 基于 (?) 视锥体的 (frustum).
2. 单步的 (single-shot): 直接通过一个阶段的网络得出结果, 通过直接预测类属的概率和回归 3D 范围盒, 不需要提案生成和后处理, 所以可以高速推理. 分类: 基于 BEV 的, 基于离散化的, 基于点的.

3.1.1 基于区域提案的方法

基于多视图这些方法每提案地把不同的视图中的特征嵌合 (e.g. LiDAR 正面视图,BEV(Bird's Eye View), 图像) 来得到 3D 取向盒,TC 常常很高.

[4] 从 BEV 图上生成高精度的候选盒子, 再投影到多视图的特征图上 (如 RGB 图,LiDAR 正面图), 最终结合这些每区域的特征来预测取向的 3D 盒子. 这个方法达到了 99.1% 的召回率 (recall) 以及再仅仅 300 个提案上 0.25 的 IoU! 但是非常慢, 难以进入使用领域, 故发展从两个方面发展别的方法来改进.

1. 提出了快速嵌合不同模态的信息的方法
    - [126] 多模态融合的区域提案方法. 从 BEV 和图像视图使用剪切和缩放提取等大小的特征, 再用每元素-pooling 嵌合特征.
    - [127] 使用了连续卷积来嵌合不同分辨率的 LiDAR 特征图, 具体上说, 对于每一个 BEV 空间的点, 提取了最近的相应图像特征 (nearest corresponding image feature), 使用双线性内插来和到 BEV 平面的投影来得到紧密的 BEV 特征图 (dense BEV feat. map).
    - [128] 提出多任务-多感知器的 3D 对象检测网络. 多个任务被用于增强性能 (2D 对象检测, 深度补全和地面估计) 多模态! 实验证明这提高了性能 (2D/3D 检测和 BEV 检测).
2. 不同的方法用于从输入中提取健壮的特征.
    - [39] 探索了多尺度上下文信息, 通过 SCA 模块 (Spatial Chan. Atten.) 在一个场景中捕捉全局和多尺度上下文来高亮有用的特征. 提出了 ESU(Extended Spatial Unsample) 模块来得到 rich-spatial 的高阶 repr. 和低阶表示的融合. 这很花时间, 因为要为每个 proposal 进行特征池化.
    - [131] 接着提出了 pre-RoI 来提高池化效率. 具体地说, 他们把卷积全部移到了池化之前, 池化对于所有提案执行一次. 11.5fps, 5 倍于 MV3D[4].

基于分割的方法: 这些方法先用分割的方法移除大多数的背景点, 再产生大量高质量的提案, 来节省计算. 和多视图方法比, 有更高召回率, 更适合复杂和封闭-拥挤的场景.

- [132] 使用 2D 分割网络来预测前景像素, 并把他们投影到点云上来去除背景点. 然后提出基于这些点提出提案, 并设计了 PointsIoU 来减少提案的重复性和模糊性.
- [133] 接着提出了 PointRCNN 框架, 他们直接分割点云来得到前景点, 再把语义特征和局部空间特征嵌合来得到高质量 3D 盒子.
- 接着 [133] 的 RPN(Region Proposal Net.),[134] 提出了使用 GCN 来进行 3D 对象检测的先驱性工作. 具体地说, 提出了两个模块来改善提案:R-GCN 使用提案中的所有提案来得到每提案的特征聚合,C-GCN 用于嵌合所有提案的每帧的信息来回归一个精确的盒子.
- [135] 把点云投影到 2D 分割网络的输出, 把语义预测分数加到点上. 改进的点云送到已有的检测器 [133,136,137] 里来得到显著性能提升.

- [138] 每个点联系一个球面锚点 (spherical anchor), 并且每个点的语义分数用于移除多余的锚点. 这个方法大大降低计算复杂度 (和 [132, 133] 相比). PointsPool 层被提出来从提案的内点学习紧的特征, 一个平行的 IoU 分支用于改善局域化精度和检测精度.

基于锥体的方法 这些方法先利用 2D 对象检测器生成 2D 候选区域, 再每候选区域生成一个 3D 锥体区域提案. 快速, 但性能有限.

- F-PointNets[139(F-PointNets)] 先驱性工作, 为每个 2D 区域生成锥体提案, 用 PointNet(or PN++) 来学习点云表示, 来进行无模态 (amodal) 盒子估计.
- Point-SENet[140(SIFRNet)] 模块来预测一些尺度因子载每特征上, 并用 PointSIFT[141] 模块来捕捉点云的取向, 来达到对形状放缩的健壮性. 相比于 F-PointNets, 大大提高了性能.
- [142(PointFusion)] 使用 2D 图片区域和对应的锥体区域来精确回归 3D 盒子. 使用了一个全局嵌合网络来嵌合点云特征和图像特征来回归盒子的角落点. 还提出了一个 dense-net. 来预测盒子角落的偏移.
- [143(RoarNet)] 先从 2D 图像估计 3D 对象的 2D 边界盒和 3D 姿势, 再从中提取多个集合可行的对象候选. 这些候选送到一个 3D 盒子回归网络来预测精确 3D 盒子.
- [144(F-ConvNet)] 每 2D 区域生成了一系列沿着锥轴的锥体, 再用 PN 提取每个锥体的特征. 锥级的特征重组生成 2D 特征图, 再送到全卷积网络 (fully convolutional net.) 来估计 3D 盒子. 这个方法再当时是基于 2D 图片的 SOTA(KITTI 的排行榜).
- [145(Patch Refinement)] 先再 BEV 获得初步检测结果, 再基于这些预测提取小点集 (i.e. patches). 一个局部 refinement net. 用于学习 patch 的局部特征并预测高精度的 3D 盒子.

其它方法

- 启发于图片中对象检测的轴对齐 IoU 的成功, [146(3D IoU Loss)] 把两个 3D 旋转盒子的 IoU 结合到 SOTA 检测器上 ([134, 137, 158]) 来得到一致的性能提升.
- [147(Fast Point R-CNN)] 提出两阶段网络, 使用点云和体素表示. 首先, 体素表示送到 3D 骨架网络 (3D backbone net.) 来产生初始检测结果, 然后初始预测的内点特征用于进一步得到精细的盒子边界. 虽然概念上很简答, 但是和 [133] 有类似性能 16.7fps
- [148] 提出 PointVoxel-RCNN(PV-RCNN), 使用 3D-CNN 和基于 PN 的集合抽象 (set abstraction) 来学习点云特征. 具体上, 体素化的点云被送到 3D-稀疏 CNN 上来得到高质量提案. 之后学到的每体素的特征编码成一个关键点的小子集, 使用体素集合抽象模块 (voxel set abstraction). 此外, 还提出了关键点到格点 (keypoint-to-grid) 的 ROI 抽象模块, 来提取丰富的上下文信息, 用于盒子精细化. 在 Car 类 KITTI 3D 检测中排行第一 (2020.6.12)
- 启发于基于 Hough Voting 的 2D 对象检测,[124] 提出 VoteNet 来直接投票出一个对象的点云的虚拟中心, 以及用聚合投票特征 (vote feature) 来得到一组高质量的提案. 它显著超越了其它只用几何信息的方法 (在室内数据集 ScanNet, SUN RGB-D), 但是在部分阻挡的物体上不稳定.

- [149] 进一步加入了方向向量的辅助分支, 用于改进虚拟中心点的预测精度和 3D 候选盒子, 并建立了 3D 对象提案间的关系图来对于精确目标检测强调有用的特征.
- [150] 提出了 ImVoteNet 检测器, 通过嵌合 2D 对象检测线索 (cue)(如几何线索和语义-纹理线索) 到 3D 投票管线中. 受到 3D 对象 ground truth 盒子提供了精确的内-对象组件的精确位置.
- [151] 提出了 Part-A^2 网络, 由 part-aware 步和 part-aggregation 步组成.part-aware 阶段使用 UNet-like[163] net. 和稀疏卷积-反卷积来学到点特征用预测和内-对象的部件位置的粗略生成.part-aggr. 阶段使用 RoI-aware 池化来聚合得到预测的部件位置用于盒子精细化.

3.1.2 一次性方法 (Single-Shot Methods)

这些方法使用一阶段网络直接预测 (在点上) 类概率和回归 3D 盒子 ⇒ 无需提案, 高速. 三类: 基于 BEV, 基于离散化, 基于点.

基于 BEV 这些方法把 BEV 表示作为输入.

- [129] 把点云用均匀的格子离散化, 再把反射率 (reflectance) 做同样处理, 得到一个正规表示 (reg. repr.). 一个 FCN(Fully Conv. Net.) 用于估计对象的位置和方向角, 这个方法已经在表现上超过了大多数一步法 (VeloFCN[154], 3D-FCN[155], Vote3Deep[156]),28.6fps.
- [152] 利用了几何和语义先验 (来自 High-Definition/HD maps) 来提高 [129] 的健壮性和性能. 具体上, 它们从 HD map 中得到了基准点 (ground points) 的坐标, 然后使用 BEV 表示到基准点的距离来克服地面斜率带来的平移偏差. 此外,concat 了二值路面遮罩 (binary road mask) 到了 BEV 表示的 channels 上来只关注移动的对象.HD map 不总是可用的, 他们还提出了一个通过 LiDAR 点云估计图先验的在线图预测模块. 这个 map-aware 方法显著优于基线 (on TOR4D,KITTI). 对于不同密度的点云的泛化能力效果差.
- 进一步, [153] 使用正则化图 (map) 来考虑不同 LiDAR 传感器的差异. 正则化图是和 BEV 图一样分辨率的 2D 格子, 编码了每个单元包含的最大点数. 显著提高了泛化能力.

基于离散化把点云转换成离散表示, 然后使用 CNN 来预测对象的 3D 盒子和类别.

3.2 3D 对象追踪

3.3 3D 场景流估计

3.4 总结

4 3D 点云分割

4.1 3D 语义分割

4.2 3D 实例分割

4.3 3D 部分分割

4.4 总结

4.5 略语表

- repr. ⇒representation
- net. ⇒network
- feat. ⇒feature
- conv. ⇒convolution/convolutional
- reg. ⇒regular/regularized
-