Deep Hough Voting for 3D Object Detection in Point Clouds

ICCV 2019 KaiMing He.etc

Motivation

- 3D 点云具有精确的几何形状和对光照变化的鲁棒性。但是，点云是不规则的。因此，典型的 CNN 不太适合直接处理点云数据。
- 从之前的工作 Faster/Mask R-CNN 拓展迁移到三维目标检测
- Hough 变换在检测二维形状上具有良好表现和鲁棒性。
- 想要构建通用的，不依赖 2D 检测器的点云检测框架

Related Work

- 转换成 2D 鸟瞰图 (MV3D)
- 结合图像信息（ 3D-SIS ）
- 体素化处理丢失了很多细节信息

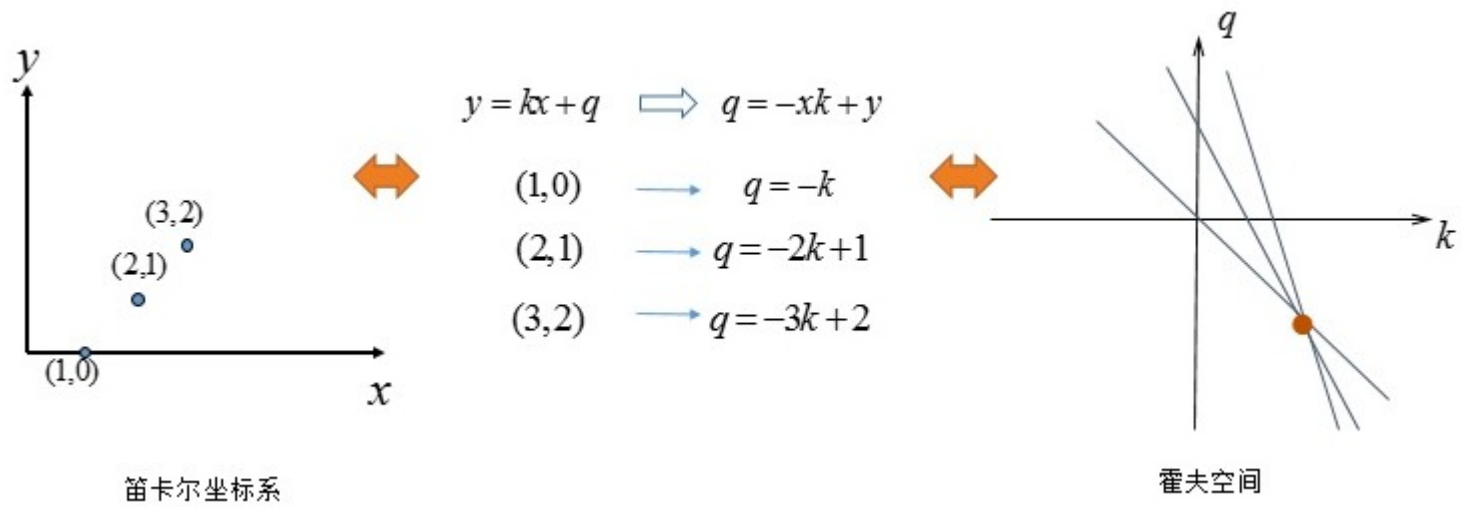
Contribution

- 重新组合并构建了基于霍夫投票和深度学习的微架构
- 在 SUNG RGB-D 和 ScanNet_v2 上达到了 State of Art
- 深度分析了运用 Hough 投票的重要性

Hough 变换

- Hough 变换将目标识别转换成了参数空间的投票问题，然后找出投票最多的参数点，认为其为具有最大可能的检测结果。
- 核心思想：
  - 将图像中的点集映射到高维空间，该点在高维空间中代表了图像中点集的图形参数，参数的范围被称为参数空间。
  - 举例来说，二维图像中的一个点可以映射成参数空间的一条曲线，将所有点都映射到参数空间，则有，在参数空间中，曲线的交点能够表征特征可能性比较大，选取满足阈值的几个点，可以实现对目标的检测，可以看成是一个聚类问题。

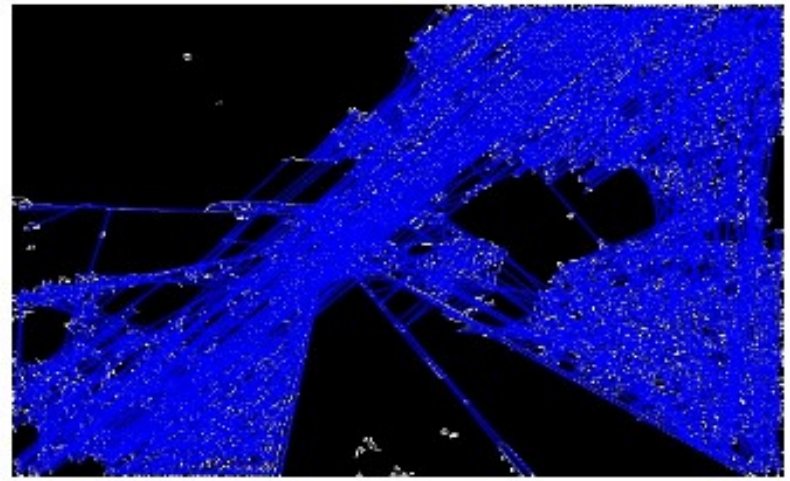
Hough 变换



笛卡尔坐标系　　霍夫空间

如图所示 给出点（ x1,y1 ） ,(x2,y2) 可以在二维坐标系下构建唯一的一条直线 y=kx+q
y1=kx1+q； y2=kx2+q
即 q=-kx1+y1； q=-kx2+y2
在参数空间（ k-q ）组成的二维坐标系内两条直线相交于一点（ k,q ），该点即代表了在 xy 坐标体系下的一条直线。通过记录一个累加数组即记录通过同一点的直线个数 K，得到 A(k,q)=K，在图像中检测直线时 会得到 K1,K2,...Kn, 通过阈值进行判定。
对于噪声的解决方案是在进行累加值峰值统计时，并不统计每个基本单元的点个数而得到最大值，而是对每个单元的一个邻域进行合并统计，可以帮助解决噪音导致的峰值分散的问题。

(a)　原始图像　　(b)　经Canny算子处理效果图



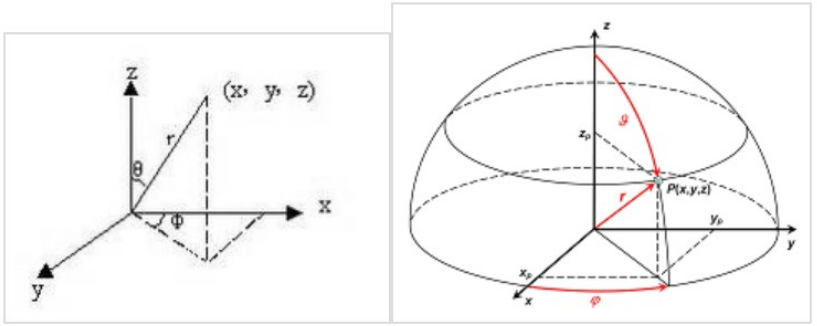
(c)　Hough Transform处理效果图

3D Hough

- 解决点云的平面检测问题
- 投票算法：
  - 已知点集 {p1,…,pn} 中存在平面以及一定数量的噪音，求解最好平面的参数 m 。拟合问题的解决思路可以用“投票 (voting)” 来概括：由点 pi 向其符合的模型 mx 投票，得票者最多的模型胜出成为“最好平面”。从理论上来说，这种方法非常通用，不过， n 个点的 pi 与数量不确定的模型之间的组合引发了可穷举性的问题，怎样确定模型的数量？霍夫变换解决的就是这个问题。

3D Hough

平面的方程:$Ax+By+Cz+D=0$,其中$\vec{v}=\begin{bmatrix}A\\B\\C\end{bmatrix}$为平面的法向量。D为原点$(0,0,0)$到平面的距离(有符号)。我们可以将法向量$\vec{v}$带入球极坐标系考虑。



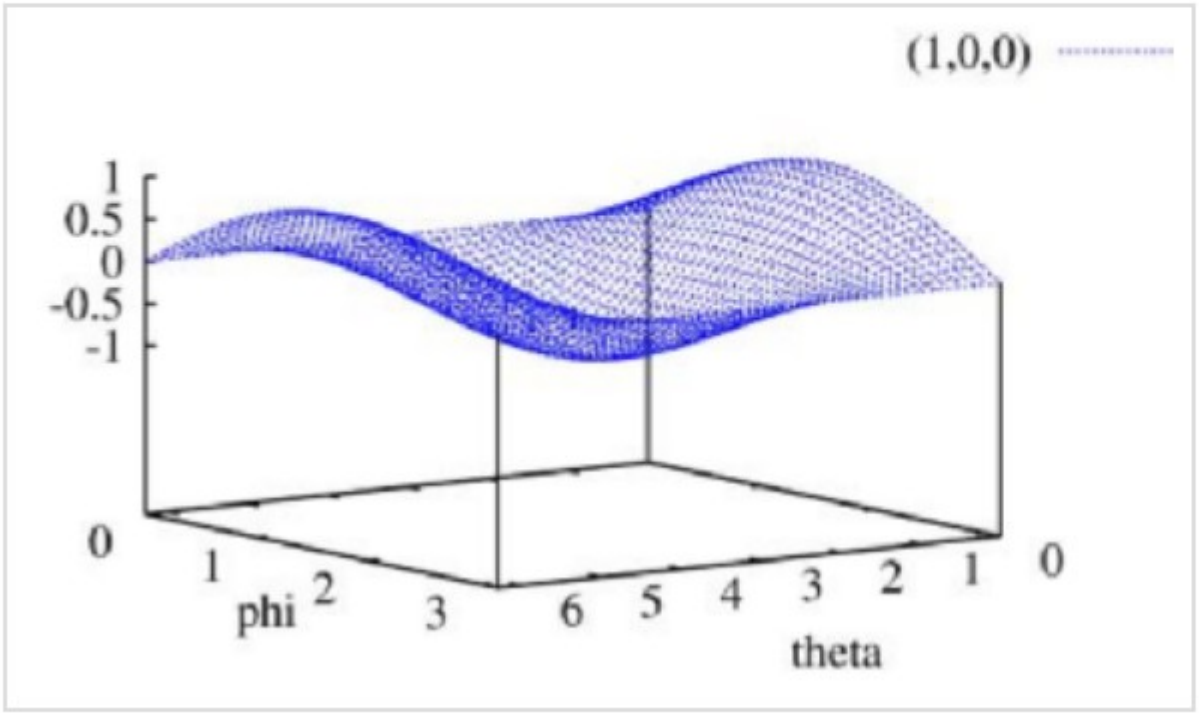
在极坐标系下

$$\vec{v}=\begin{bmatrix}cos\phi sin\theta\\sin\phi sin\theta\\cos\theta\end{bmatrix}$$

因为 $\theta\in[0,2\pi],\phi\in[0,2\pi]$,所以我们可以通过离散$\theta,\phi$以及原点到平面的距离$\rho$来实现对参数空间的离散化枚举。

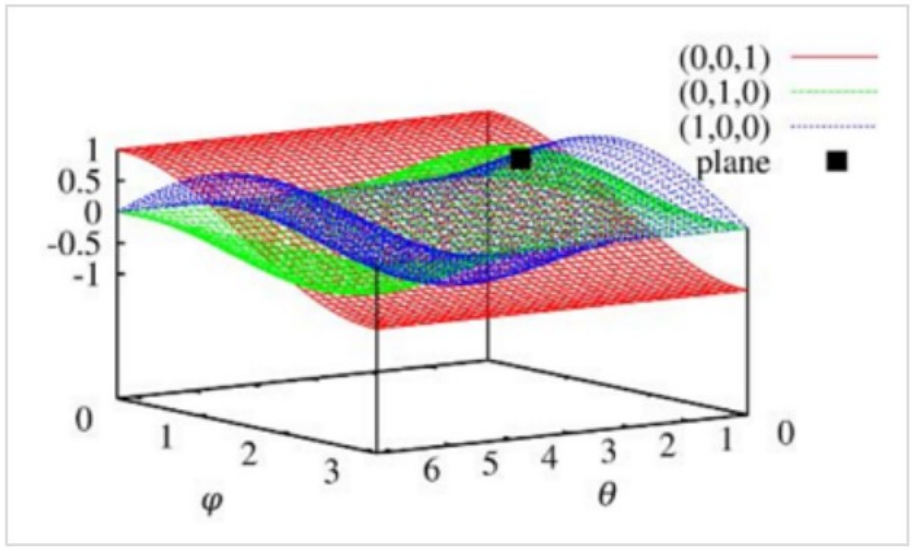
3D Hough

直线方程为：$(cos\phi sin\theta)x + (sin\phi sin\theta)y + cos\theta + \rho = 0$，对于点$p_i$，其针对所有的$(\theta, \phi)$带入方程均可求得对应的$\rho$，因此每个点$p_i$都可在参数空间形成了对应参数曲面如下图所示：



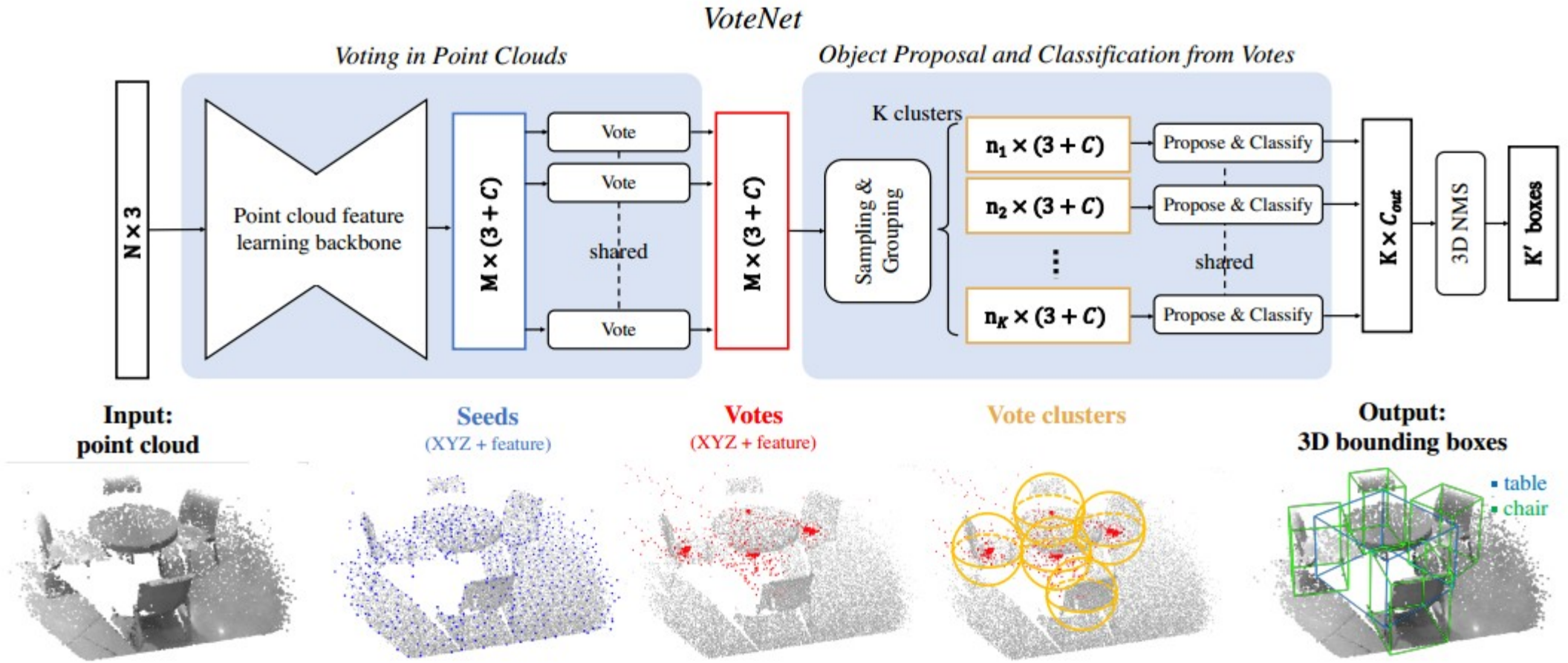
3D Hough

多个点对应的参数曲面会形成多个交点，其中交点最多的参数(θ, ϕ, ρ)即对应的点数最多的平面。如下图所示：



从实现的角度考虑，我们可以构建一个三维的数组，第三维保存符合平面的点的个数，称为累加器。针对每个点p_i，均对其对应的(θ, ϕ, ρ)进行累加。

VoteNet Framework



VoteNet Framework

- 学习如何在点云上进行投票
  - 学习点云特征：
    - 使用 pointnet++ 进行 vote 初始 seed 的提取。
      - input: 原始点云（ N×3 ）
      - output: 种子点（ M×（ 3+C ））xyz+c 维特征
  - 使用深度网络进行霍夫投票
    - 原始的三维霍夫投票需要有密码本来记录各种形状的峰值特征
    - 深度方法 简单直接准确率高 (MLP with fc & Relu & BN)
      - input：种子点 {Si}(i∈[1,M]),si=[xi;fi](即 xyz+c 维特征)
      - output:vote-->vi=[yi;gi] yi=xi+△xi （与 xi 的欧氏距离） gi=fi+△fi （与特征的偏差）（投票点）
      - Loss: $L_{\text{vote-reg}} = \frac{1}{M_{\text{pos}}} \sum_i \|\Delta x_i - \Delta x_i^*\| \mathbb{1}[s_i \text{ on object}], \quad (1)$ xi
      - 用来表征种子点是否在平面上 $\mathbb{1}[s_i \text{ on object}]$.
      - Mpos 表示物体表面的种子点个数，△ xi* 表示在 gt 中，从种子点到 bbox 中心点的距离
    - 使用监督使得生成的 vote 不再为表面的种子点而是 位于目标中心的投票点。

VoteNet Framework

- 从投票点出发得到目标的 bbox 和语义标签
  - 投票聚簇：均匀采样 -> 空间聚类
    - 从投票点中使用最远点采样得到 K 个点，然后每个点找到紧邻点并聚簇。
  - 从聚簇的投票点中生成语义和 bbox
    - 本质上还是一团点云（带有高维特征）xyz+c 维特征 (进行局部归一化)
    - 使用 shared PointNet 作为工具
    - input:{wi} wi=[zi;hi]zi:xyz, hi:pointnet++ feature
    - output: 多维向量 p p=[目标性打分， bbox 参数，语义得分]
    - Loss: L （ obj-cls ） + λ2 L （ box ） + λ3 L （ sem-cls ）
      - obj-cls: 交叉熵 loss
      - box ： 与 gt 进行比较计算 loss
      - sem-cls ： 交叉熵 loss

Experiments

	Input	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
DSS [42]	Geo + RGB	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG [38]	Geo + RGB	58.3	63.7	31.8	62.2	**45.2**	15.5	27.4	51.0	**51.3**	70.1	47.6
2D-driven [20]	Geo + RGB	43.5	64.5	31.4	48.3	27.9	25.9	41.9	50.4	37.0	80.4	45.1
F-PointNet [34]	Geo + RGB	43.3	81.1	**33.3**	64.2	24.7	**32.0**	58.1	61.1	51.1	**90.9**	54.0
VoteNet (ours)	**Geo only**	**74.4**	**83.0**	28.8	**75.3**	22.0	29.8	**62.2**	**64.0**	47.3	90.1	**57.7**

Table 1. **3D object detection results on SUN RGB-D val set.** Evaluation metric is average precision with 3D IoU threshold 0.25 as proposed by [40]. Note that both COG [38] and 2D-driven [20] use room layout context to boost performance. To have fair comparison with previous methods, the evaluation is on the SUN RGB-D V1 data.

Experiments

	Input	mAP@0.25	mAP@0.5
DSS [42, 12]	Geo + RGB	15.2	6.8
MRCNN 2D-3D [11, 12]	Geo + RGB	17.3	10.5
F-PointNet [34, 12]	Geo + RGB	19.8	10.8
GSPN [54]	Geo + RGB	30.6	17.7
3D-SIS [12]	Geo + 1 view	35.1	18.7
3D-SIS [12]	Geo + 3 views	36.6	19.0
3D-SIS [12]	Geo + 5 views	40.2	22.5
3D-SIS [12]	Geo only	25.4	14.6
VoteNet (ours)	Geo only	**58.6**	**33.5**

Table 2. **3D object detection results on ScanNetV2 val set.** DSS and F-PointNet results are from [12]. Mask R-CNN 2D-3D results are from [54]. GSPN and 3D-SIS results are up-to-date numbers provided by the original authors.

Analysis

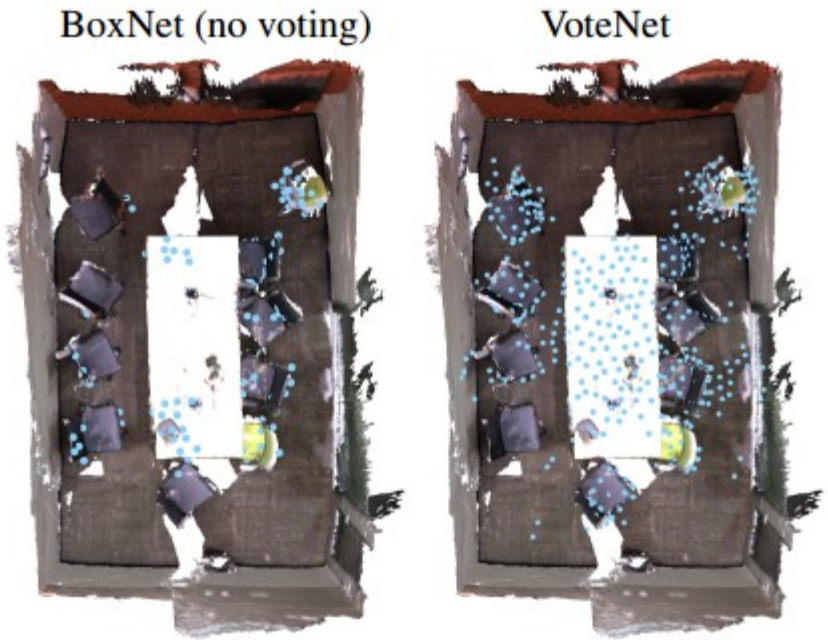


Figure 3. **Voting helps increase detection contexts.** Seed points that generate good boxes (BoxNet), or good votes (VoteNet) which in turn generate good boxes, are overlaid (in blue) on top of a representative ScanNet scene. As the voting step effectively increases context, VoteNet demonstrates a much denser cover of the scene, therefore increasing the likelihood of accurate detection.

Method	mAP@0.25	
	SUN RGB-D	ScanNet
BoxNet (ours)	53.0	45.4
VoteNet (ours)	**57.7**	**58.6**

Table 3. **Comparing VoteNet with a no-vote baseline.** Metric is 3D object detection mAP. VoteNet estimate object bounding boxes from vote clusters. BoxNet proposes boxes directly from seed points on object surfaces without voting.

Qualitative Results

Method	Model size	SUN RGB-D	ScanNetV2
F-PointNet [34]	47.0MB	0.09s	-
3D-SIS [12]	19.7MB	-	2.85s
VoteNet (ours)	11.2MB	0.10s	0.14s

Table 4. **Model size and processing time (per frame or scan).** Our method is more than 4× more compact in model size than [34] and more than 20× faster than [12].

Result

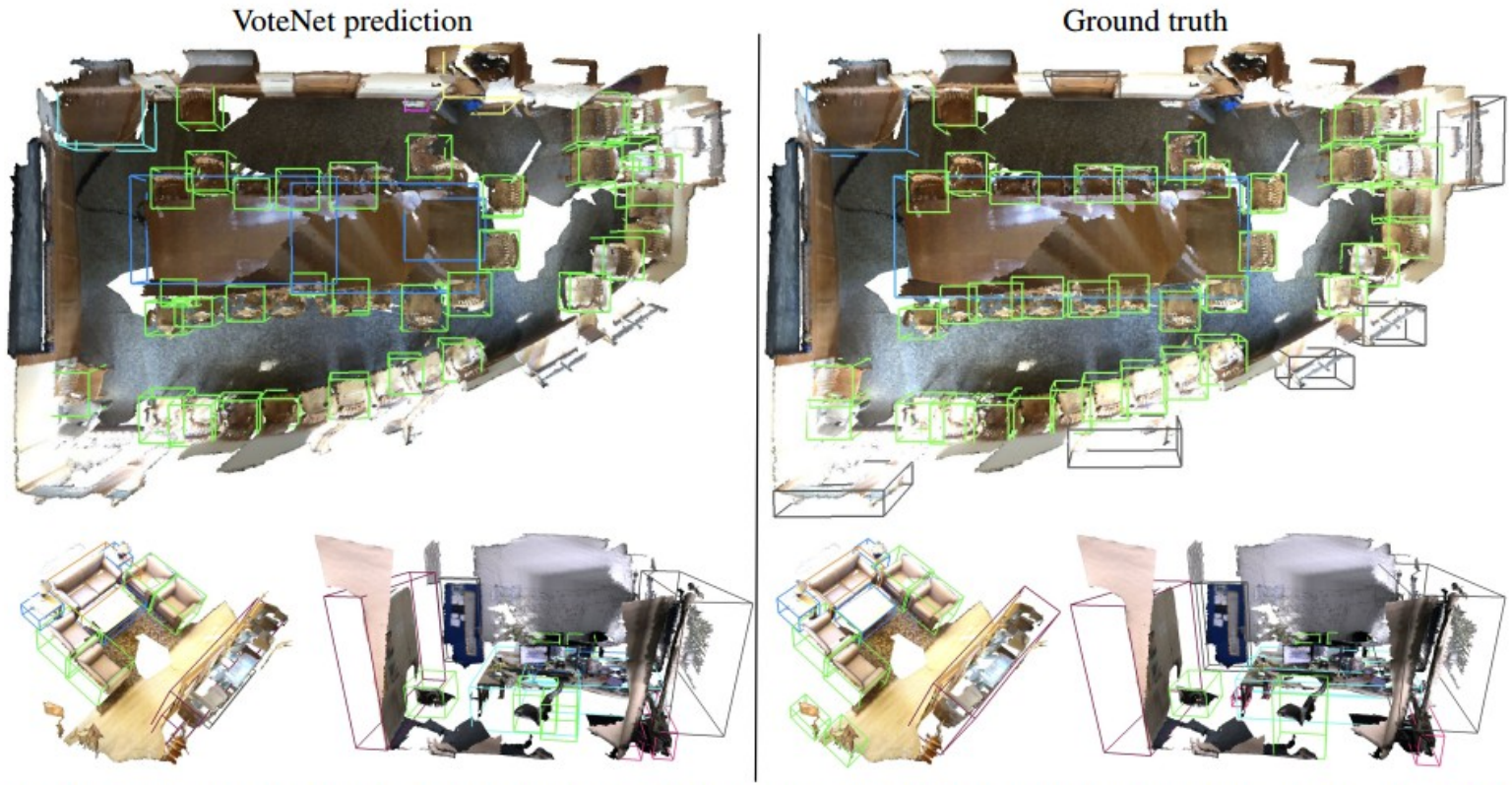


Figure 6. **Qualitative results of 3D object detection in ScanNetV2.** Left: our VoteNet, Right: ground-truth. See Section 5.3 for details.

Result

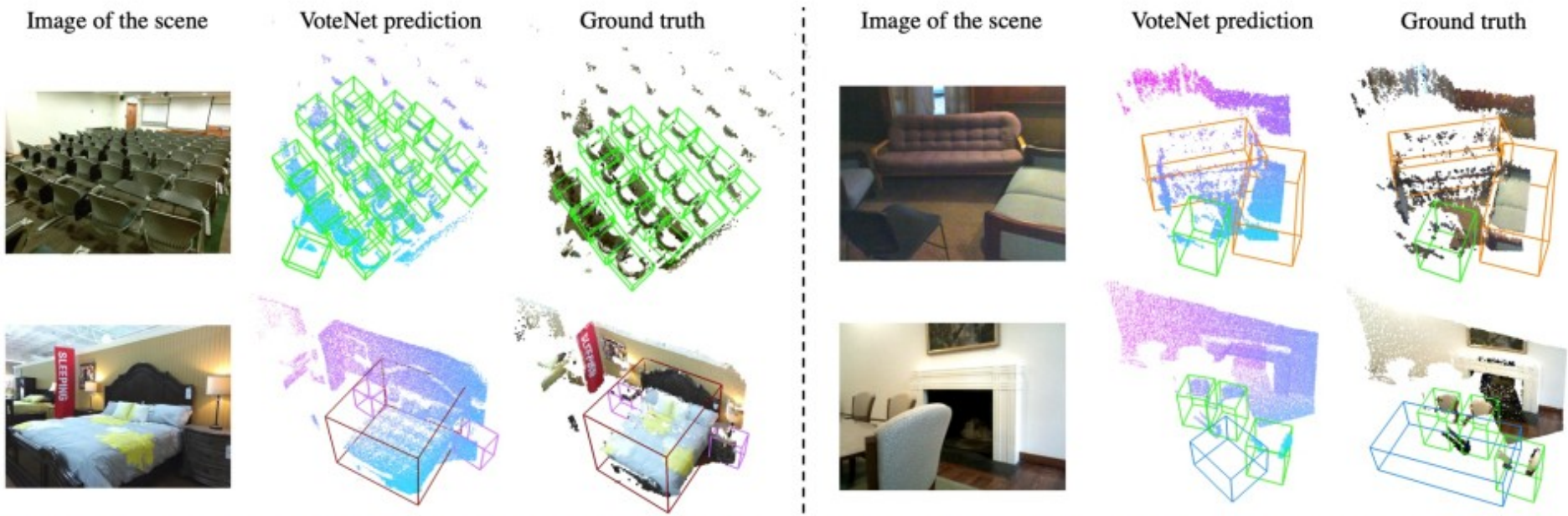


Figure 7. **Qualitative results on SUN RGB-D.** Both left and right panels show (from left to right): an image of the scene (not used by our network), 3D object detection by VoteNet, and ground-truth annotations. See Section 5.3 for details.

Conclusion

- 由于在稀疏的 3D 点云中，现有的场景点往往远离目标的中心点，直接提出的 proposal 可能置信度较低或不准确。相反，投票让这些较低的置信点更接近，并允许通过聚合来强化它们的假设。
- 可以看出，当目标点远离边界框中心时，投票会起到更大的作用。
- VoteNet 利用了点云的稀疏性，避免在空的空间搜索。与以前的最佳方法相比，该模型比 F-PointNet 小 4 倍，在速度上比 3D-SIS 快 20 倍。
- 该模型仅使用 3D 点云，与之前使用深度和彩色图像的方法相比，有了显著的改进。且在复杂且混乱的点云数据上也表现除了较好的鲁棒性。

Reference

- https://www.cnblogs.com/nowgood/p/houghtransfrom.html
- http://www.whudj.cn/?p=877
- Object Detection using a Max-Margin Hough Transform