Deep Hough Voting for 3D Object Detection in Point Clouds

ICCV 2019

Summary

This paper presents VoteNet, a novel architecture introducing *Deep Hough Voting* as a means to perform 3D Object Detection solely based on geometric information from point clouds. The method solves the problem of missing points at the center of an observed object by shifting the surface points towards the object center using a voting module. This improves performance specifically for objects with large distances between surface and object center. VoteNet achieves state-of-the-art performance on two indoor datasets, SUN RGB-D and ScanNet, outperforming previous arts by a large margin.

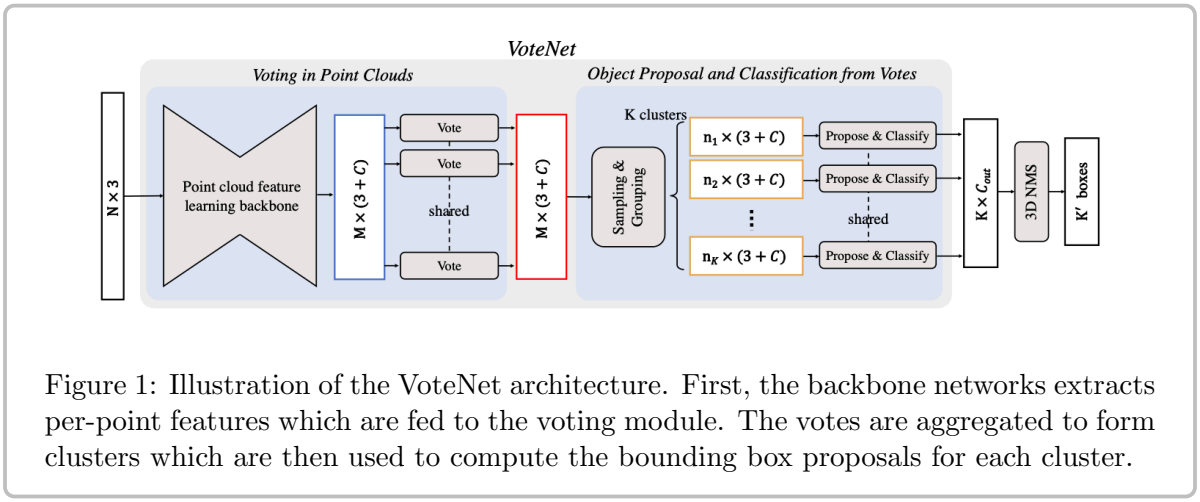

Figure 1: Illustration of the VoteNet architecture. First, the backbone networks extracts per-point features which are fed to the voting module. The votes are aggregated to form clusters which are then used to compute the bounding box proposals for each cluster.

Main Contributions

- **Deep Hough Voting** The network architecture can be divided into four main components: 1) the feature extraction backbone; 2) the voting module; 3) the clustering module and 4) the proposal module. The authors use a PointNet++ as the backbone which extracts a number of seed points with corresponding high-dimensional feature vectors. Each seed point generates a vote by predicting a residual pose and feature vector. The votes are then aggregated to form clusters for each of which a bounding box proposal is generated.
- **SOTA in 3D Object Detection** The authors report state-of-the-art results on indoor datasets, ScanNet and SUN RGB-D. The method is compared to a number of previous methods that use geometric information as well as RGB images. VoteNet outperforms all evaluated models by at least 3.7 and 18.4 mAP on SUN RGB-D and ScanNet, respectively.
- **Analysis of the Influence of Voting** The authors present BoxNet as a baseline model similar to VoteNet without Deep Hough Voting to illustate the effects of voting on the performance. The results show that VoteNet solves the problem of large point distances due to the surface restrictions by aggregating the votes around the center of the observed objects.

Implementation Details

- **Vote Generation** The input to the vote generation module is a sub-sampled set of key points including high-dimensional per-point feature vectors. For each of the so-called seed points a vote is generated using an MLP. The vote is implemented as a euclidean offset δx_i and a feature offset δf_i. The euclidean offset is explicitly supervised during training. Unlike the seed points, the votes are no longer restricted to the surface of the objects.
- **Vote Clustering** The set of votes is sub-sampled using *farthest point sampling* to determine the centers of the clusters. Clusters are then formed by considering a euclidean ball of specified radius around the cluster center.
- **Proposal Generation** For the final proposal generation, a PointNet-like module is applied. After locally normalizing the vote locations within their cluster, proposals are generated by passing the points of the cluster through a module consisting of two MLPs and a channel-wise max-pooling in between.

Evaluation

- **Effect of Voting** The effect of voting is evaluated by comparing VoteNet to BoxNet, a baseline model with the same backbone network but without voting module. The results of the experiment are illustrated in Figure 3. It can be seen that the performance gain by including voting is positively correlated with the normalized mean distance between the surface points and the object center. This underlines the potential of VoteNet especially for large objects.
- **Vote Aggregation Analysis** The authors test different methods for vote aggregation. Average-pooling, max-pooling and RBF weighting are tested as well as PointNet-like modules using average- and max-pooling, respectively. The results show that the learned aggregation modules outperforms the other methods by a large margin. Regarding the aggregation radius, the experiments confirm that the performance increases with the radius until 0.2m before decreasing due to the incorporation of clutter votes.
- **Sensitivity of Proposal Sampling** Three different sampling methods are tested for vote clustering, vote FPS, seed FPS and random sampling. The results indicate that the model is robust w.r.t. the sampling methods as the performance differences are negligible.

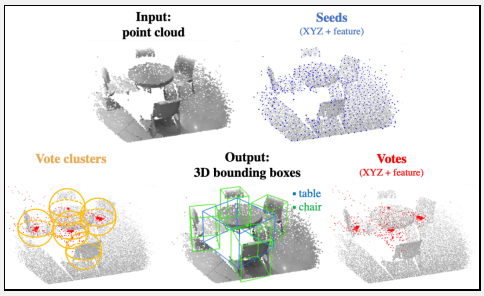

Figure 2: Illustration of the VoteNet pipeline.

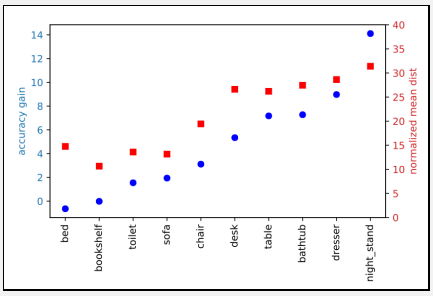

Figure 3: Illustration of the voting effect.

References

This summary is solely based on my understanding of the original paper. All images used here are taken from the original paper as well. The paper can be found under the following link:
`https://arxiv.org/pdf/1904.09664.pdf`