SPOT: Selective Point Cloud Voting for Better Proposal in Point Cloud Object Detection

Hongyuan Du, Linjun Li, Bo Liu, and Nuno Vasconcelos

Department of Electrical and Computer Engineering
University of California, San Diego
{hdu,lili,boliu,nvasconcelos}@ucsd.edu

Abstract. The sparsity of point clouds limits deep learning models on capturing long-range dependencies, which makes features extracted by the models ambiguous. In point cloud object detection, ambiguous features make it hard for detectors to locate object centers (Fig. 1) and finally lead to bad detection results. In this work, we propose Selective Point clOud voTing (SPOT) module, a simple effective component that can be easily trained end-to-end in point cloud object detectors to solve this problem. Inspired by probabilistic Hough voting, SPOT incorporates an attention mechanism that helps detectors focus on less ambiguous features and preserves their diversity of mapping to multiple object centers. For evaluating our module, we implement SPOT on advanced baseline detectors and test on two benchmark datasets of clutter indoor scenes, ScanNet and SUN RGB-D. Baselines enhanced by our module can stably improve results in agreement by a large margin and achieve new state-or-the-art detection, especially under more strict evaluation metric that adopts larger IoU threshold, implying our module is the key leading to high-quality object detection in point clouds.

1 Introduction

3D object detection is important for many applications, such as indoor robot navigation, augmented reality, and autonomous driving. While it can be performed using data from many sensing modalities, there has recently been interest in point clouds, due to their ability to accurately represent geometric information, their lightweight nature, and the popularity of LIDAR sensors. It is, however, challenging to implement object detection on point clouds, for two main reasons. First their non-Euclidean structure [5] makes them poorly suited for classic deep-learning architectures. Second, their sparsity increases the challenges of feature extraction. The first problem has received substantial interest in the recent computer vision literature, with the introduction of many deep architectures tailored for point clouds [25, 18, 17, 27, 29, 37, 13, 22]. However, considerably less progress has been observed on the second.

[0] First two authors had equal contribution.

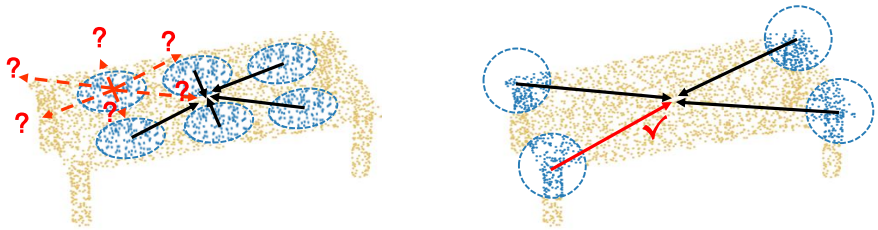


Fig. 1. Object localization from local shape measurements. Left: points on locations where the object surface has low dimensional structure, such as a table top, contribute ambiguous information for localization of the object center. **Right:** points with 3D structure, denoted *suspicious coincidences* due to the associated non-accidental confluence of geometric information (e.g. three lines that intersect at a point), are much more informative for this localization.

Modern point cloud architectures for object detection, attempt to mitigate the sparseness problem by aggregating information from multiple points [46, 43, 20, 33, 26, 34, 32]. An object is usually defined in terms of its center or a bounding box, which are detected by aggregating local shape information from the points on the object surface. This can be seen as a voting mechanism, where each point contributes information for both the localization and identification of the object. For example, the aggregation of geometric information from all points in the surface of each of the tables of Fig. 1 is what allows the perception of these point clouds as tables. However, the consolidation of the *local* measurements into a *global* object percept is a difficult problem, because not all points on an object are equally informative of object identity and location.

Consider, for example, the localization of the table of known dimensions of Fig. 1, from local shape measurements derived from sets of points on the surface of the object, such as those shown of the figure. As shown on the left, a neighborhood on the surface of the table is consistent with many object centers. This can be seen from the fact that any 2D translation along the tabletop leaves the neighborhood unchanged. Any amount of noise in the point cloud can originate a vote to an incorrect center or bounding box. Hence, such points are not reliable indicators of the object location. This is not the case for the neighborhood shown on the right, which is centered on a corner of the table. In this case, the neighborhood is only consistent with a center vote. Hence, the point is a reliable indicator of the object location.

For object class detection, the situation is obviously more complex, since the table can have any height and length. Nevertheless, it remains true that points where the object surface has 3D structure (e.g. table corners) are much more informative than points of 2D (table edges) or 1D structure (table top). This is similar to the *aperture problem* in optic flow estimation, where object corners are known to be more informative of object motion than other image points. In fact, the importance of these informative points for object recognition and

localization has long been pointed out in the vision literature. This dates back to at least the work Attneave [1] which equated the visual cortex to a detective that makes inductive inferences about the environment by looking out for "suspicious coincidences", such as the confluence of three 3D edges into a single point. This is what enables the recognition of a table from a hand-drawn sketch depicting some lines and corners. In computer vision, the detection of suspicious, or non-accidental coincidences has been proposed as a principle for perceptual organization by various authors [4, 23] and motivated a large literature on corner detection and interest points [14, 31].

Non-accidental coincidences are important for detection exactly because their non-accidental nature makes them *rare*. Hence, when objects are sampled sparsely, they are likely to either be missed or immersed in an ocean of less informative points. This increases the difficulty of recovering object identity and location. In this work, we seek to address this problem by focusing the attention of the object detector in points of suspicious coincidences. For this we introduce a *Selective Point clOud voTing* (SPOT) module, which seeks to increase the attention of the point cloud around points of suspicious coincidences and reduce it everywhere else. SPOT consists of a combination of two operations: 1) detection of locations of suspicious coincidences, and 2) voting synthesis in the neighborhood of these locations. The two operations are performed on the 3D interest points produced by popular detection architectures in the literature. The first is implemented by a softmax network and the second by a set of non-linear regressions. This allows the implementation of both operations with a simple module that can be easily integrated into most existing point cloud detectors, to enable end-to-end training. We demonstrate this by implementing SPOT on three point cloud object detectors, VoteNet [26], PointRCNN [33], and a self-implemented version of PointRCNN that uses the Sparse Convolutional Network [13] as backbone. Evaluation on two large datasets of indoor scenes, ScanNet [8] and SUN RGB-D [35], shows that a simple implementation of SPOT without any bells and whistles can enhance all the baseline models by a large margin. In particular, it is shown that SPOT improves performance under more strict evaluation metrics, using higher IoU thresholds. This suggests that selective voting is important for high quality point cloud object detection.

2 Related work

Feature Learning for Point Cloud Analysis. To deal with the irregular format of point cloud, one popular direction is to convert points into voxels in regular 3D grids and then utilize 3D CNNs for feature learning [42, 25, 28, 46]. Recent works adopt Sparse Convolutional Networks [13] to reduce the computation cost of 3D convolution so that much larger point cloud input can be processed for vision tasks like semantic segmentation [13] and object detection [43, 20, 47]. Another trend is to use neural networks specially formulated for point cloud data. PointNet [27] and PointNet++ [29] are pioneers in this area that take point coordinates as input and learn permutation invariant features by multi-

layer perceptrons and MaxPooling, showing strong performance on modeling point cloud geometry. In this work, we evaluate SPOT on both kinds of feature learning schemes, showing that our module is an universally useful structure for enhancing point cloud object detectors.

Point Cloud Object Detection. Due to the growing applications of high-resolution lidar sensors and the challenge of 2D-3D sensor fusion, recent methods are proposed to directly detect objects in 3D using point clouds. Some of these convert point clouds into voxels and use 3D CNNs to form backbones [46, 43, 20, 47]. PointRCNN [33] and VoteNet [26] utilize PointNet [27] and PointNet++ [29] to do detection on raw point clouds. More recently, several works [44, 6, 34, 32] explore the hybrid of voxel and point representation to take the advantages from both. Our work investigates the impact of suspicious coincidence on point cloud object detectors and proposes a method to make them more robust.

Hough Voting in 2D/3D Object Detection. Hough transform/voting is a good paradigm for bottom-up detection. Origin Hough transform [16] lets edge points vote in parameter space for detecting simple shapes like lines and circles. Generalized Hough transform [2] can detect arbitrary shapes, by recording a matching table of the mappings from an edge orientation to possible positions of a reference point on the shape. Leibe et al. [21] further extend this idea to general object detection and segmentation in images, by using more discriminative features and probabilistic voting that learns the likelihood of a vote being an object center in a data-driven manner. Improved methods also show success in 3D recognition problems [41, 19, 24, 38]. Recent works attempt embedding Hough voting in deep learning models for 3D object detection. [39, 9] cast votes according to the weights of convolutional kernels. VoteNet [26] includes a voting module to cast one-to-one votes, with each local feature voting for one object center. Similar schemes are implemented in PointRCNN [33], where one foreground point is used to predict a single proposal. In contrast, our work inherits and extends the idea of probabilistic Hough voting [21] that selectively allows a local feature to cast multiple votes with probability weighting and implements it in an end-to-end trainable style, showing strong performance on high quality object detection in point clouds.

3 Selective Point Cloud Voting

3.1 Overview

While the idea of selective voting can be of interest for many operations on point clouds, in this work we consider its deployment in the context of the two-stage object detection architecture show at the top of Fig. 2. This is a general architecture, implemented by several popular detectors in the literature. The first stage generates object proposals. Given an input of N points with XYZ coordinates, a backbone network is used to abstract the point cloud and learn

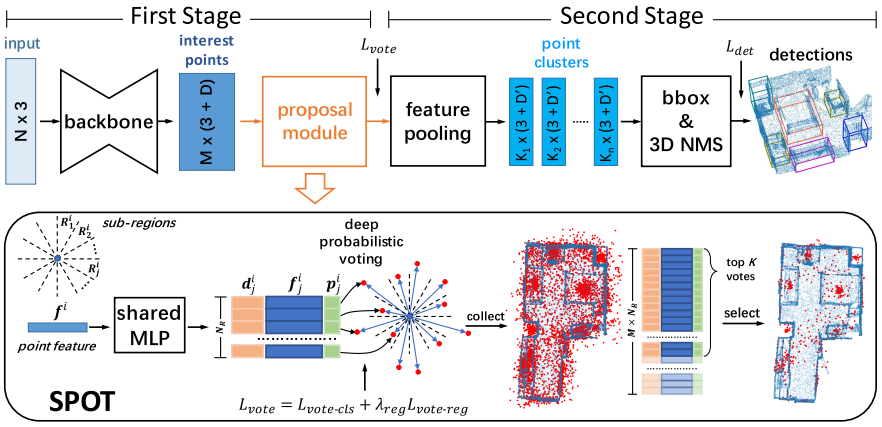


Fig. 2. Detection pipeline and Selective Point clOud voTing (SPOT). The proposal module of existing point cloud detectors is replaced by the proposed SPOT for better localization of object centers.

deep features. It outputs a subset of the input containing M *interest points* $\mathbf{q}^i = (\mathbf{z}^i, \mathbf{f}^i)$, each composed by a vector $\mathbf{z}^i$ of 3D coordinates and a D-dimensional descriptor $\mathbf{f}^i$ of the local object geometry. Interest points are all the information retained for object proposal generation. Proposals are generated by a *proposal module,* which maps the interest point descriptors $\mathbf{f}^i$ into a preliminary prediction of the locations of scene objects. The second stage performs a pooling or NMS of proposals to infer a refined set of descriptors. Finally, those are processed by a detection head that includes classification, bounding box regression, and NMS modules to output the final detection.

SPOT works on the proposal module in the first stage. Commonly used proposal module has slightly different implementations on different detectors. For example, PointRCNN [33] predicts 3D bounding boxes as proposals that is similar to region proposal [30] in image object detection; in VoteNet [26], object centers are regressed as proposals instead of whole bounding boxes, and the local shape descriptors $\mathbf{f}^i$ are propagated to the proposals for its second stage. Though implemented differently, a common behavior is that all interest points uniformly generate proposals, which gives no preference to the points of suspicious coincidences, such as the corner on the right of Fig. 1. Since these points are rare, they can be missed altogether, or have small contribution to the set of proposals considered in the subsequent stages of the detector. Instead, the large majority of the proposals available to the later stages originate from points that are much less informative of the object identity and location, such as the tabletop points on the left of Fig. 1. These proposals are likely to be less accurate than those rooted at locations of suspicious coincidences.

The two-stage detector is generally supervised by a combination of a proposal loss on the first stage and a detection loss on the second stage. The detection loss

L_{det} consists of objectness, bounding box regression and semantic classification. In our case, the first stage is supervised by a voting loss L_{vote} and the whole system is trained with the loss

$$L = L_{\text{vote}} + \lambda_{\text{det}} L_{\text{det}} \tag{1}$$

In this section, we will discuss the novel SPOT model trained with the addition of the voting loss L_{vote}.

3.2 SPOT

SPOT can be seen as an attention mechanism that aims to focus the proposal stage of Fig. 2 on interest points indicative of suspicious coincidences, such as the formation of a 3D corner by the confluence of three 3D lines on the same point[1]. Given a set of interest points $\mathbf{q}^i = (\mathbf{z}^i, \mathbf{f}^i)$, it produces a set of *votes* $\mathbf{v}_j^i = (\mathbf{d}_j^i, \mathbf{f}_j^i, p_j^i)$. Each interest point can contribute none or multiple votes, depending on how suspicious it is. Vote $\mathbf{v}_j^i$ is composed by a 3D coordinate $\mathbf{d}_j^i$, a D-dimensional descriptor $\mathbf{f}_j^i$ and a probability p_j^i. The 3D coordinate is the prediction of the object center, the descriptor is a refined version of the descriptor $\mathbf{f}^i$ provided by the input interest point, and the probability is the posterior of this vote being predicted as a valid object center. The goal of SPOT is to increase the attention of the point cloud around points of suspicious coincidences and reduce it everywhere else. This is performed by a sequence of two operations: 1) detection of locations of suspicious coincidences, and 2) selectively voting for object centers in the neighborhood of these locations.

Suspicious Coincidences. The central operation for the detection of suspicious coincidences is the estimation of the certainty with which the object center can be determined from the shape information contained in each interest point. This is inspired by Fig. 1. Interest points located at points of the object surface rich in 3D structure (such as corners) provide stronger constraints for localization of the object center than interest points at locations where the object surface has lower dimensional structure (such as table tops). Since the amount of 3D structure in the vicinity of the interest point can in principle be derived from the local shape descriptor $\mathbf{f}^i$, it should be possible to detect suspicious interest points by analyzing the shape descriptors.

SPOT implements this intuition as follows. Given interest point $\mathbf{q}^i = (\mathbf{z}^i, \mathbf{f}^i)$, a neighborhood of $\mathbf{z}^i$, composed by a series of pre-specified sub-regions $\mathcal{R}_j^i, j \in \{1, \ldots, N_R\}$, is defined as shown in Fig. 2. It is assumed that the object center is within one of the regions $\mathcal{R}_j^i$, identified by a label $y^i \in \{1, \ldots, N_R\}$. The posterior probability of this label is then predicted by a classifier

$$[p_1^i, \ldots, p_{N_R}^i]^T = [P(y^i = 1|\mathbf{f}^i), \ldots, P(y^i = N_R|\mathbf{f}^i)]^T = g(\mathbf{f}^i; \theta^g), \tag{2}$$

[1] While the description provided in this section is tailored to VoteNet, SPOT can be deployed on other detectors with minor modifications. Some variants are discussed in the experiments section.

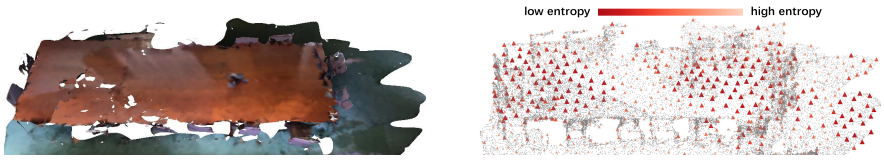


Fig. 3. Detection of suspicious coincidences. Left: A 3D scene composed of a large table. **Right:** Corresponding point cloud. For suspicious coincidences, the entropy of the distribution of center locations has low-entropy. These tends to happen around object regions informative for object identification and localization, such as table corners and edges.

where $g(;\theta^g)$ is a NN of parameters θ^g with softmax activation. During detector training, this classifier is optimized on a set of object interest points $\mathcal{Q} = \{(\mathbf{z}^i, \mathbf{f}^i, \mathbf{c}^i)\}$, of coordinate $\mathbf{z}^i$, descriptor $\mathbf{f}^i$, and ground-truth object center $\mathbf{c}^i$. For each interest point, a label y^i is set to the index of the region $\mathcal{R}^i_j$ that contains the object center, i.e. $y^i = j | \mathbf{c}^i \in \mathcal{R}^i_j$. The center location classifier is then optimized by minimizing the cross-entropy loss

$$L_{\text{vote-cls}} = -\sum_i \log g_{y^i}(\mathbf{f}^i; \theta^g). \tag{3}$$

During inference, given interest point $\mathbf{q}^i = (\mathbf{z}^i, \mathbf{f}^i)$, the classifier $g(.;\theta^g)$ is used to estimate the probabilities of (2). The detection of suspicious coincidences then follows from the intuition of Fig. 1. While interest points in the neighborhood of these coincidences (e.g. the centers of the patches on the right of the figure) should have distributions of low uncertainty, those consistent with many object centers (e.g. on the left of the figure) should generate high uncertainty. This is confirmed by Fig. 3, which shows a typical example of the the information entropy of the posterior distribution

$$\mathbf{H}(\mathbf{f}^i) = -\sum_{j=1}^{N_R} g_j(\mathbf{f}^i; \theta^g) \log g_j(\mathbf{f}^i; \theta^g) \tag{4}$$

for many interest points on an object surface. Points of lower dimensional structure, such as the center of a tabletop, the ground, etc., generate larger entropies than points on the vicinity of the 3D edges and corners that demarcate the table boundaries.

This suggests that the detection of suspicious coincidences can be framed as an instance of the problem of assessing classification confidence, which has received recent interest in the literature on calibration of deep classifiers [11, 12, 40, 7]. Methods based on the thresholding of the entropy of (4) or the largest probability at the classifier output

$$\mathbf{C}(\mathbf{f}^i) = \max_j g_j(\mathbf{f}^i; \theta^g) \tag{5}$$

are commonly used in this literature to assess whether the classifier is confident in the classification of a given example. However, the calibration of the network probabilities is known to be difficult. For example, some methods require the training of a secondary network just for this purpose [40, 12, 7], while others rely on computational expensive Monte-Carlo dropout [10] procedures.

In this work, we rely on an alternative approach, which aims to increase the robustness of suspicious coincidence detection. This is implemented as follows. Given a point cloud $\mathcal{P}$ of M interest points $\mathbf{q}^i$ of shape descriptor $\mathbf{f}^i$, the posterior probabilities from all interest points are stored in an array $\mathbf{G} \in [0,1]^{M \times N_R}$ such that $\mathbf{G}_{ij} = g_j(\mathbf{f}^i; \theta^g)$. The entries of $\mathbf{G}$ are then ranked in decreasing order and the *minimum confidence* threshold for suspicious coincidence detection set to the value of the K^{th} largest entry, i.e.

$$T(\mathcal{P}; K) = rank_K(\mathbf{G}). \tag{6}$$

The set of suspicious coincidences is then defined as the set of interest points for which the confidence measure of (5) is above this threshold, i.e.

$$\mathcal{S}(K) = \{\mathbf{q}^i \in \mathcal{P} \,|\, \mathbf{C}(\mathbf{f}^i) \geq T(\mathcal{P}; K)\}. \tag{7}$$

where $\mathbf{C}$ is the confidence measure of (5). Note that the computations above can be easily performed by standard neural network operations. $\mathbf{C}$ is a simple max-pooling operation and (6) is implemented by sorting the outputs of g_j in each iteration.

The parameter K controls the number $|\mathcal{S}(K)|$ of detected suspicious coincidences. $|\mathcal{S}(K)|$ is usually smaller than K, because suspicious coincidences can assign strong probability to more than one object center location. This is typically the case for complex scenes, where an interest point located at a suspicious coincidence, e.g. the corner of a table, may be located near another object, e.g. a chair. In this case, because the receptive field centered on the interest point overlaps with both objects, the interest point may confidently vote for both of them. In our experience, it is not uncommon for suspicious coincidences to vote for two or three different center locations.

The overall procedure can be seen as an attention mechanism that focuses the detector on the object locations most informative for object identification and detection. An additional benefit is that, in 3D object detection, the background is usually composed of planar structures, such as the ground or the walls of a room. Since, as shown in Fig. 3, interest points located on these structures are unlikely to be declared suspicious, SPOT tends to suppress background clutter. Hence, in addition to being an attention mechanism that highlights informative object features, it also acts as an object-level attention mechanism, which declares objects as overall salient from background walls and ground.

Selective Voting. Selective voting attempts to aggregate local features from the locations of suspicious coincidences while rejects those from elsewhere. This is done as follows. During training, given an interest point $(\mathbf{z}^i, \mathbf{f}^i)$ whose corresponding object centered at $\mathbf{c}^i$, the region $\mathcal{R}^i_j$ that contains the center is first

identified, which gives the label $y_i = j$. A ground truth offset $\Delta\mathbf{z}_*^i = \mathbf{c}^i - \mathbf{z}^i$ is then computed. The set of offsets associated with the same region label $y_i = j$ are then assembled into an offset training set $\mathcal{O}_j = \{\Delta\mathbf{z}_*^i | \mathbf{z}^i + \Delta\mathbf{z}_*^i \in \mathcal{R}_j^i\}$. This set is then used to train a *center location* regression function $\Delta\mathbf{z}_j^i = \phi_j(\mathbf{f}^i; \theta_j^\phi)$ for the region $\mathcal{R}_j^i$. The center location regression functions ϕ_j are learned by minimizing the regression loss

$$L_{\text{vote-reg}} = \sum_{j=1}^{N_R} \sum_{i|y_i=j} ||\phi_j(\mathbf{f}^i; \theta_j^\phi) - \Delta\mathbf{z}_*^i|| \tag{8}$$

Combined with (3), the whole voting loss is then implemented as

$$L_{\text{vote}} = L_{\text{vote-cls}} + \lambda_{\text{reg}} L_{\text{vote-reg}}. \tag{9}$$

During inference, given interest point $(\mathbf{z}^i, \mathbf{f}^i)$, the center location regressors ϕ_j are used to predict an estimate of the location of the center for each of the regions $\mathcal{R}_j^i$

$$\mathbf{d}_j^i = \mathbf{z}^i + \phi_j(\mathbf{f}^i; \theta_j^\phi). \tag{10}$$

Finally, a descriptor

$$\mathbf{f}_j^i = \mathbf{f}^i + \varphi_j(\mathbf{f}^i; \theta_j^\varphi) \tag{11}$$

is synthesized for each new center location $\mathbf{d}_j^i$. This is a refinement of the descriptor $\mathbf{f}^i$ of the interest point $(\mathbf{z}^i, \mathbf{f}^i)$, which accommodates variations of local geometry between $\mathbf{z}^i$ and $\mathbf{d}_j^i$. Since ground truth descriptors are not available, φ_j functions are learned end-to-end, using supervision from the second stage loss L_{det} of (1).

Overall, a single interest point produces multiple object center votes $\mathbf{v}_j^i = (\mathbf{d}_j^i, \mathbf{f}_j^i, p_j^i)$, where $\mathbf{d}_j^i$ is the center predicted by (10), $\mathbf{f}_j^i$ the shape descriptor predicted by (11) and p_j^i the probability $P(y^i = j|\mathbf{f}^i)$ of (2). $\mathcal{S}(K)$ of (7) then takes effect as a selection mechanism that only the votes of interest points in $\mathcal{S}(K)$ are passed to the subsequent stages of the network. Furthermore, the votes of these interest points are pruned by considering only those whose confidence is larger than the threshold $T(\mathcal{P}; K)$ of (6). This leads to a final set of votes

$$\mathcal{V} = \{(\mathbf{d}_j^i, \mathbf{f}_j^i, p_j^i) \,|\, p_j^i \geq T(\mathcal{P}; K)\}. \tag{12}$$

The functions g of (2), ϕ_j of (10), and φ_j of (8) are implemented with a shared MLP. The network implementation of SPOT is summarized at the bottom of Fig. 2.

4 Experiments

In this section, we discuss several experiments performed to evaluate the performance of SPOT.

4.1 Dataset and Evaluation Metrics.

SUN RGB-D [35] is a dataset of RGB-D images for scene understanding. It contains 10K RGB-D images densely annotated with 58,657 3D bounding boxes with orientations for 47 object categories. For fair comparison, we follow the evaluation protocol of [26], which prunes the dataset to ∼5K samples and the 10 most common categories. RGB-D images are converted to clouds of 20K points per image. ScanNetV2 [8] is an indoor scene dataset of 3D meshes reconstructed from RGB-D images. It contain 1,201 scans from hundreds of rooms, annotated with instance segmentation for 18 object categories. Following [26], we convert the meshes to clouds of 40K points per scene by sampling mesh vertices, and evaluate object detection performance on aligned circumscribed bounding boxes of instance segmentations.

4.2 Implementation Details

The impact of SPOT is evaluated on three detectors: VoteNet [26], PointRCNN [33] and a variant of PointRCNN based on Sparse Convolution [13].

VoteNet The original voting module is replaced by SPOT. This is implemented with interest point neighborhoods of 24 sub-regions $\mathcal{R}_1$ to $\mathcal{R}_{24}$. These are defined by 12 radial partitions, as illustrated in the bottom of Fig. 2, and one partition along the Z-axis. The rank parameter K of (6) is chosen so that the number of votes $|\mathcal{V}|$ of (12) is equal to $1,024$.

PointRCNN To minimize the changes to the original PointRCNN model, we modify its bin-based localization to SPOT. After the point cloud is segmented into background and foreground, each foreground point is considered as an interest point, predicting the object center coordinates along the X and Y axes. These axes are binned into 6 segments, forming a set of square regions $\mathcal{R}_1, \ldots, \mathcal{R}_{36}$. The probability of the object center being located $\mathcal{R}_j$ is then computed as $p_j = p_{xj} \cdot p_{yj}$, where p_{xj} and p_{yj} are the probabilities computed by the original network for the X and Y bins corresponding to $\mathcal{R}_j$.

PointRCNN-SC To investigate the effectiveness of SPOT on different backbone networks, we have also implemented a variant of PointRCNN using the submanifold sparse U-Net [13] as backbone to extract pointwise features. The remaining components are unaltered. This is denoted as PointRCNN with sparse convolutions (PointRCNN-SC).

4.3 Ablation Studies

We start with a series of experiments using VoteNet and ScanNet V2 to ablate several parameters of SPOT.

Definition of suspicious coincidences. SPOT defines suspicious coincidences as in (7), from which the set of center votes $\mathcal{V}$ of (12) is extracted. Table 1 compares this strategy to other possibilities for proposal selection. These are

	total votes	min votes/I.P.	max votes/I.P.	AR@0.5	mAP@0.5
VoteNet (regression)	1024	1	1	53.3	33.5
best 1 per I.P.	1024	1	1	56.0	38.7
best 2 per I.P.	2048	2	2	55.0	38.4
best 3 per I.P.	3072	3	3	54.3	37.1
SPOT	1024	0	3	**57.8**	**40.4**

Table 1. Ablation study of vote selecting method. I.P. means interest point. Our method outperforms others by dynamically selecting votes based on the spatial probability.

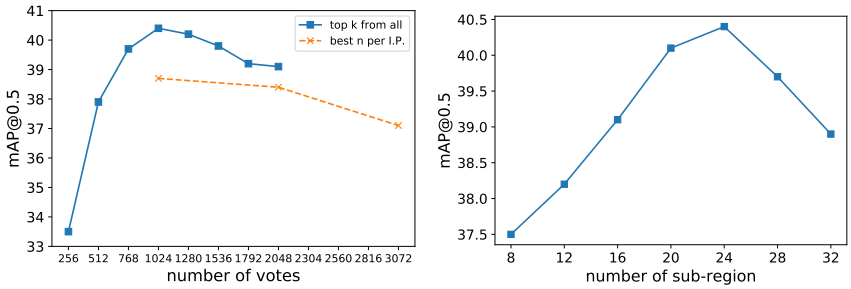
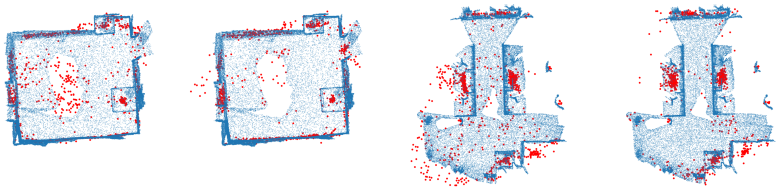


Fig. 4. Analysis of number of votes and sub-regions. Left: Number of votes v.s. mAP performance, under the case of 24 sub-regions. **Right:** Number of sub-regions v.s. mAP performance, under the case of 1024 votes.

the simple regression of the baseline VoteNet, and three alternative approaches that keep the best 1, 2, and 3 votes of largest probability from each interest point. Selecting the best vote from each interest point outperforms the baseline, confirming the advantages of sub-region center votes as compared to a single center regression. On the other hand, selecting the best 2 or 3 votes per interest point degrades performance. This is because these votes are not reliable for interest points that are not co-located with suspicious coincidences. In fact, as discussed in Fig. 1, even the top vote is usually unreliable when this is the case. The detection of suspicious coincidences by SPOT eliminates such ambiguous interest points, allowing the detector to focus attention on the ones that most informative of object center locations. Hence, for the same number of votes as the best 1 strategy, and significantly less than the others, SPOT enables the best detection performance. When compared to the baseline, it enables significant gains of more than 3 points under both AR@0.5 and mAP@0.5 metrics.

Number of votes. We next investigated the impact of the number of votes $|\mathcal{V}|$ on overall detector performance. This is shown in Fig. 4 a) for mAP@0.5. Performance increases until $|\mathcal{V}| = 1,024$ and decreases after that. Also shown is the performance of the three best n per I.P. strategies. For $|\mathcal{V}| = 512$ the performance of SPOT is already superior to the baseline detector and for $|\mathcal{V}| >$



(a) VoteNet (b) SPOT (c) VoteNet (d) SPOT

Fig. 5. Votes from baseline model and SPOT. Red dots are center votes.

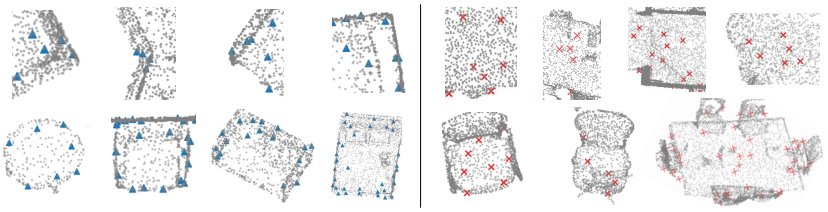


Fig. 6. local geometry of different interest points. Left (blue): Interest points that cast deterministic votes. **Right (red):** Interest points that cast multiple votes or filtered out.

768 SPOT is always superior to all other strategies. While the choice of the threshold of (6) has some impact on the mAP@0.5 performance, SPOT has the best performance for a large range of thresholds. These results illustrate its robustness.

Number of sub-regions. The impact of the number of sub-regions N_R on detection performance was also investigated, by varying the number of spatial sectors $\mathcal{R}_j^i$ under a bird's eye view. The results are shown in Fig. 4 b). When the number of sub-region is too small detection performance degrades. This is not surprising, because the sub-regions $\mathcal{R}_j^i$ become less selective for center location. On the other hand, too many sub-regions can also lead to a drop in performance, because classifier labels become noisier and there are fewer examples per region, increasing the difficulty of learning all the classification and regression functions of SPOT. While the careful section of the number of sub-regions can make a significant difference, in the example of the figure a gain of almost 3 points for $N_R = 24$, the curve is not very sharp and there is some flexibility in this parameter.

4.4 Detection Results

The detector baselines enhanced by SPOT were compared to the original versions and several enhancements recently proposed in the literature. The results are summarized in Table 2, which shows that SPOT improves the performance of all

	Scan @0.25	Scan @0.5	SUN @0.25	SUN @0.5
DSS [36]	15.2	6.8	42.1	-
F-PointNet [6]	19.8	10.8	54.0	-
GSPN [45]	30.6	17.7	-	-
3D-SIS[15]	40.2	22.5	40.2	22.5
PRCNN[33]	53.0	25.4	53.7	23.4
PRCNN+SPOT	55.2	27.8	57.6	25.3
PointRCNN-SC	57.0	31.8	53.0	24.5
PointRCNN-SC+SPOT	57.4	33.1	59.5	27.7
VoteNet [26]	58.7	33.5	57.7	32.3
VoteNet+SPOT	**59.8**	**40.4**	**60.4**	**36.3**

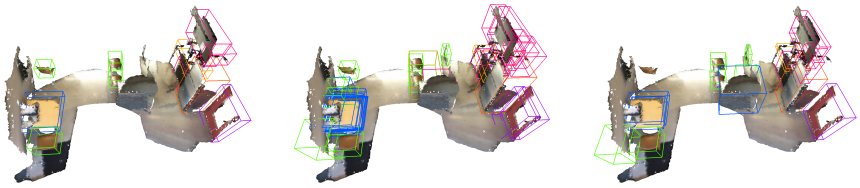
Table 2. 3D object detection results on SUN RGB-D and ScanNet V2 validation set with 3D IoU threshold 0.25 and 0.50. DSS, F-PointNet and 3D-SIS results are from [15], GSPN are from [45], VoteNet are from [26], PointRCNN is implemented base on [33] and Sparse Conv backbone is implemented base on [13].

	bed	table	desk	refrigerator	bathtub	counter
AP gain	+7.3	+7.4	+7.2	+7.6	+6.2	+4.8
mean size	(1.9, 1.8, 1.2)	(1.0, 1.2, 0.6)	(1.0, 1.4, 0.9)	(0.7, 0.7, 1.3)	(1.2, 1.1, 0.5)	(1.4, 1.9, 0.3)
	chair	toilet	sink	garbagebin	picture	cabinet
AP gain	+1.7	+1.4	+1.3	+1.1	+0.2	-1.5
mean size	(0.6, 0.6, 0.7)	(0.6, 0.6, 0.7)	(0.5, 0.5, 0.3)	(0.5, 0.5, 0.6)	(0.2, 0.4, 0.5)	(0.8, 0.8, 0.9)

Table 3. AP gains v.s. object sizes. The best 6 gains (top) and the worst 6 gains (bottom) out of 17 object categories of ScanNet V2 along with objects' mean sizes.

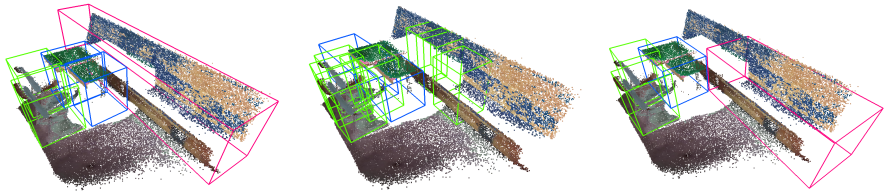
detectors. For most combinations of dataset and detector, the gains are between 2 and 3 map@0.5 points, with the larger gains being observed for the strongest baseline, which is VoteNet. This and the improvements of the PointRCNN with different backbone designs suggest that SPOT should improve the quality of other point cloud detectors.

Qualitative Results. To understand how SPOT improves detection accuracy, we visualize the votes produced by it and compare with those generated by the baseline model. Fig. 5 shows how SPOT focus the attention of the detector on suspicious coincidences indicative of object presence. Note the much smaller number of votes on the ground or table tops and the concentration of votes on object surfaces. Fig. 6 shows the local geometry of different kinds of interest points. Interest points that cast deterministic votes, i.e., one vote with high probability score, tend to gather around 3D structures of objects like corners and edges. On the other hand interest points that cast multiple votes or are prunned by SPOT tend to gather on low dimensional structures, such as flat surfaces, or locations where multiple object intersect. Table 3 summarizes AP



(a) Ground Truth (b) VoteNet (c) VoteNet w/ SPOT

Fig. 7. Qualitative results for ScanNetV2



(a) Ground Truth (b) VoteNet (c) VoteNet w/ SPOT

Fig. 8. Qualitative results for SUN RGB-D

gains per object class, showing that SPOT benefits the most the detection of large objects, such as beds, tables, or desks. This is because these objects contain larger surface areas of low dimensional structure than small objects and are more likely to produce interest points unaligned with suspicious coincidences. Fig. 7 and Fig. 8 show how the addition of SPOT affects the object detection of VoteNet on ScanNetV2 and SUN RGB-D. As shown in these figures, the addition of SPOT leads to a reduction of false positives. For example, in Fig. 7, the vote clustering of VoteNet produces a set of detections around each object, even after NMS. With SPOT the results are much more accurate, due to the attention of votes around suspicious coincidences.

5 Conclusion

In this work, we considered point cloud object detection, and proposed a procedure for selective point cloud voting (SPOT). This can be seen as an attention mechanism, which increases the attention of the point cloud in the neighborhood of suspicious coincidences, i.e. features that are most informative of object identity and location. SPOT was shown to be a valuable addition to several state of the art detectors based on different architectures, achieving state-of-the-art results on both ScanNet and SUN-RGBD. All of these observations confirm long-standing arguments for the importance of suspicious coincidences in object recognition [1, 3], and suggest that selective point cloud voting should be useful for future object detector designs. **Acknowledgment:** This work was partially funded by NSF awards IIS-1637941, IIS-1924937, and NVIDIA GPU donations.

References

1. Attneave, F.: Information aspects of visual perception. Psychological Rev. **61**, 183–193 (1954)
2. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern recognition **13**(2), 111–122 (1981)
3. Barlow, H.: Cerebral cortex as a model builder. Models of the Visual Cortex pp. 37–46 (1985)
4. Binford, T.: Inferring surfaces from images. Artificial Intelligence **17**, 205–244 (1981)
5. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
6. Chen, Y., Liu, S., Shen, X., Jia, J.: Fast point r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 9775–9784 (2019)
7. Corbière, C., Thome, N., Bar-Hen, A., Cord, M., Pérez, P.: Addressing failure prediction by learning model confidence. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2902–2913 (2019)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5828–5839 (2017)
9. Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I.: Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1355–1361 (2017)
10. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning (ICML). pp. 1050–1059 (2016)
11. Geifman, Y., El-Yaniv, R.: Selective classification for deep neural networks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 4878–4887 (2017)
12. Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. In: International Conference on Machine Learning (ICML). pp. 2151–2159 (2019)
13. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9224–9232 (2018)
14. Harris, C.G., Stephens, M., et al.: A combined corner and edge detector. In: Alvey vision conference. vol. 15, pp. 10–5244 (1988)
15. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4421–4430 (2019)
16. Hough, P.V.: Machine analysis of bubble chamber pictures. In: Proceedings of the International Conference on High Energy Accelerators and Instrumentation. pp. 554–556 (1959)
17. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3d segmentation of point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2626–2635 (2018)

18. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 863–872 (2017)
19. Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L.: Hough transform and 3d surf for robust three dimensional classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 589–602. Springer (2010)
20. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12697–12705 (2019)
21. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. International Journal of Computer Vision (IJCV) **77**(1-3), 259–289 (2008)
22. Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8895–8904 (2019)
23. Lowe, D.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence **31**, 355–395 (1987)
24. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1038–1045 (2009)
25. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 922–928 (2015)
26. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 9277–9286 (2019)
27. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
28. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5648–5656 (2016)
29. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 5099–5108 (2017)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 91–99 (2015)
31. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. International Journal of Computer Vision (IJCV) **37**, 151–172 (2000)
32. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10529–10538 (2020)
33. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–779 (2019)

34. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
35. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 567–576 (2015)
36. Song, S., Xiao, J.: Deep sliding shapes for amodal 3d object detection in rgb-d images. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR). pp. 808–816 (2016)
37. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J.: Splatnet: Sparse lattice networks for point cloud processing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2530–2539 (2018)
38. Velizhev, A., Shapovalov, R., Schindler, K.: Implicit shape models for object detection in 3d point clouds. In: International Society of Photogrammetry and Remote Sensing Congress. vol. 2, p. 2 (2012)
39. Wang, D.Z., Posner, I.: Voting for voting in online point cloud object detection. In: Robotics: Science and Systems. vol. 1, pp. 10–15607 (2015)
40. Wang, P., Vasconcelos, N.: Towards realistic predictors. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 36–51 (2018)
41. Woodford, O.J., Pham, M.T., Maki, A., Perbet, F., Stenger, B.: Demisting the hough transform for 3d shape recognition and registration. International Journal of Computer Vision (IJCV) **106**, 332–341 (2014)
42. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1912–1920 (2015)
43. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
44. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1951–1960 (2019)
45. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3947–3956 (2019)
46. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4490–4499 (2018)
47. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)

.