OGC: Unsupervised 3D Object Segmentation from Rigid Dynamics of Point Clouds

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we study the problem of 3D object segmentation from raw point clouds. Unlike all existing methods which usually require a large amount of human annotations for full supervision, we propose the first unsupervised method, called OGC, to simultaneously identify multiple 3D objects in a single forward pass, without needing any type of human annotations. The key to our approach is to fully leverage the dynamic motion patterns over sequential point clouds as the supervision signals to automatically discover rigid objects. Our method consists of three major components, 1) the neural backbone to directly estimate multi-object masks, 2) the auxiliary self-supervised scene flow estimator, and 3) our core object geometry consistency component. By carefully designing a series of loss functions, we effectively take into account the multi-object rigid consistency and the object shape invariance in both temporal and spatial scales. This allows our method to truly discover the object geometry even in the absence of annotations. We extensively evaluate our method on three datasets, demonstrating the superior performance for object part instance segmentation and general object segmentation in both indoor and the challenging outdoor scenarios.

1 Introduction

Identifying 3D objects from point clouds is vital for machines to tackle high-level tasks such as autonomous planning and manipulation in real-world scenarios. Inspired by the seminal work PointNet [53], a plethora of sophisticated models [66, 71, 34] have been developed to accurately detect and segment individual objects from the sparse and irregular point clouds. Although these methods have achieved excellent performance on a wide range of public datasets, they primarily rely on a huge amount of human annotations for full supervision. However, it is extremely costly to fully annotate every objects in point clouds due to the irregularity of data format.

Very recently, a few works start to address 3D object segmentation in the absence of human annotations. By analysing 3D scene flows from sequential point clouds, Jiang *et al*. [25] apply the conventional subspace clustering optimization technique to identify moving objects from raw point cloud sequences. With the self-supervised learning of 3D scene flow, SLIM [3] is the first learning-based work to showcase that the set of moving points can be effectively learned as an object against the stationary background. Fundamentally, their design principle shares the key spirit of Gestalt theory [68, 65] developed exactly 100 years ago: the raw sensory data with similar motion are likely to be organized into an object. This is indeed true in our real world where solid objects usually have strong correlation in rigid motions. However, these methods can either effectively separate the foreground and background points, being unable to simultaneously learn to segment multiple interested 3D objects in one go, essentially due to their incapability of grouping diverse motion patterns into multiple objects.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

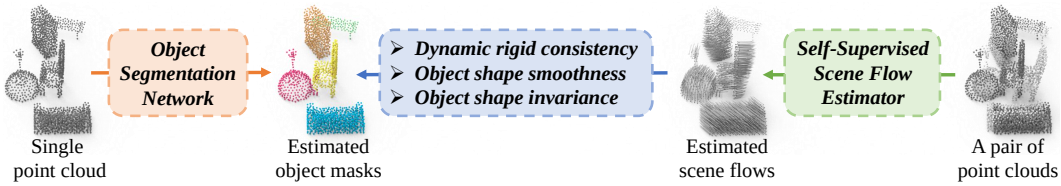

Figure 1: The general workflow and components of our framework.

Motivated by the potential of motion dynamics, this paper aims to design a general neural framework to simultaneously segment multiple 3D objects, without requiring any human annotations but the inherent object dynamics in training. To achieve this, a naïve approach is to train a neural network to directly cluster motion vectors into groups from sequential point clouds, which is widely known as motion segmentation in 2D images [72, 73]. However, such design requires that the input data points are sequential in both training and testing phases, and the trained model cannot infer objects from a single point cloud. Fundamentally, this is because the learned motion segmentation strategies simply cluster similar motion vectors instead of discriminating the diverse geometry of objects, and therefore such design is not general enough for real-world applications.

In this regard, we design a new pipeline which takes a single point cloud as input and directly estimates multiple object masks in a single forward pass. Without needing any human annotations, our pipeline instead leverages the underlying dynamics of sequential point clouds as supervision signals. In particular, as shown in Figure 1, our architecture consists of three major components: 1) a neural backbone to extract per-point features and estimate all object masks from a single point cloud, as indicated by the orange block; 2) an auxiliary self-supervised network to estimate per-point motion vectors from a pair of point clouds, as indicated by the green block; 3) a series of loss functions to fully utilize the motion dynamics to supervise the object segmentation backbone, as indicated by the blue block. For the first two components, we are actually flexible to adopt any of existing neural feature extractors [54] and self-supervised motion estimators [31]. Nevertheless, the third component is particularly challenging to design, primarily because we need to take into account not only the consistency of diverse dynamics of multiple objects in a sequence, but also the invariance of object geometry irregardless of different moving patterns.

To tackle this challenge, we introduce three key losses: 1) a multi-object dynamic rigid consistency loss, which aims to evaluate how coherently all estimated object masks (shapes) can fit the motion via rigid transformations; 2) an object shape smoothness prior, which regularizes all points of each estimated object to be spatially continual instead of fragmented; 3) an object shape invariance loss, which drives multiple estimated masks of a particular object to be invariant given different (augmented) rigid transformations. These losses together force all **o**bjects' **g**eometry to be **c**onsistent and represented by high-quality masks, purely from raw 3D point clouds without any human annotations. Our method is called **OGC** and our contributions are:

- We introduce the first unsupervised multi-object segmentation pipeline on 3D point clouds, without needing any human annotations in training.
- We design a set of geometry consistency based losses to fully leverage the object rigid dynamics and shape invariance as effective supervision signals.
- We demonstrate promising object segmentation performance on three datasets, showing significantly better results than classical clustering and optimization baselines.

Scope: This paper neither aims to design a new self-supervised 3D scene flow estimation method, nor intends to replace existing fully-supervised 3D object segmentation approaches because the never-seen static objects are unlikely to be discovered due to the lack of supervision signals. In addition, estimating object categories or non-rigid objects is also out of the scope of this paper.

2 Related Works

Fully-supervised 3D Object Segmentation: To identify 3D objects from point clouds, existing fully-supervised solutions can be divided as 1) bounding box based object detection methods [78, 34, 57] or 2) mask based instance segmentation pipelines [66, 71, 64]. Thanks to the dense human annotations and the well developed backbones including projection-based [35, 7, 34], point-based [54, 61, 23]

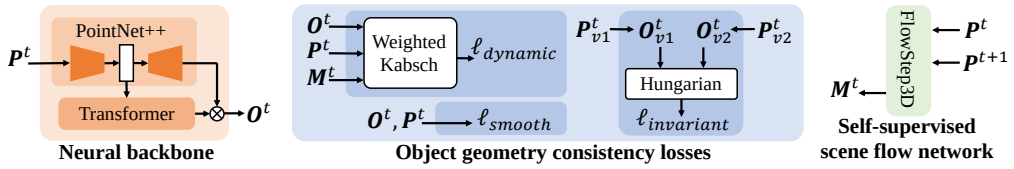

Figure 2: Components of our pipeline. The neural backbone consists of neural layers of PointNet++ and MaskFormer. FlowStep3D is adopted as the self-supervised scene flow network.

and voxel-based [19, 10] feature extractors, these methods achieve impressive performance on both indoor and outdoor datasets. However, manually labelling every object in large-scale point clouds is laborious and costly. To alleviate this burden, this paper aims to pioneer 3D object segmentation in the absence of human labels.

3D Scene Flow Estimation and Motion Segmentation: Given sequential point clouds, per-point 3D motion vectors, also known as scene flow, can be accurately estimated. Early works focus on fully-supervised scene flow estimation [39, 4, 21, 40, 52, 50, 67], whereas recent methods start to explore self-supervised motion estimation [47, 69, 31, 43, 76, 36]. Taking the scene flow as input, a number of works [74, 24, 60, 3] aim to group similar motion vectors, and then obtain bounding boxes or masks only for dynamic objects. Although achieving encouraging results, they either rely on ground truth segmentation for supervision or can only segment simple foreground and background objects, without being able to simultaneously segment multiple objects. In this paper, we leverage the successful self-supervised scene flow estimator as our auxiliary neural network to provide valuable supervision signals, so that multiple objects can be identified in a single forward pass.

Unsupervised 2D Object Segmentation: Inspired by the early work AIR [14], a large number of generative models have been proposed to discover objects from single images without needing human annotations, including MONet [5], IODINE [20], Slot-Att [41], etc. These methods are further extended to segment objects from video frames [32, 22, 46, 26, 79, 13]. However, all these approaches can only process simple synthetic datasets, and cannot discover objects from complex real-world images or videos yet. It is still elusive to apply these ideas on 3D point clouds where 3D objects are far more complicated and diverse in terms of geometry.

2D Scene Flow Estimation and Motion Segmentation: Given image sequences, pixel-level 2D scene flow, also known as optical flow, have been extensively studied in literature [16, 77]. The estimated flow field can be further grouped as objects [58, 59, 9, 42, 72, 38, 29]. Drawing insights from these works, this paper aims to segment multiple diverse objects in the complex 3D space.

3 OGC

3.1 Overview

As shown in Figure 2, given a single point cloud $\boldsymbol{P}^t$ with N points as input, *i.e.*, $\boldsymbol{P}^t \in \mathbb{R}^{N\times 3}$, where each point only has a location $\{x, y, z\}$ without color for simplicity, the **neural backbone** extracts per-point features and directly reasons a set of object masks, denoted as $\boldsymbol{O}^t \in \mathbb{R}^{N\times K}$, where K is a predefined number of objects that is large enough for a specific dataset. In particular, we firstly adopt PointNet++ [54] to extract the per-point local features. In order to infer object masks from the point features, we simply plug the recent MaskFormer [8] architecture which shows excellent performance in object segmentation in 2D images. Thanks to the powerful Transformer module [63], each object mask inferred by MaskFormer is effectively modeled over the entire point cloud.

In the meantime, we have the corresponding sequence of point clouds for supervision, denoted as $\{\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \cdots\}$. For simplicity, we only use the first two frames $\{\boldsymbol{P}^t, \boldsymbol{P}^{t+1}\}$ and feed them into the **auxiliary self-supervised scene flow network**, obtaining satisfactory motion vectors for every point in the first point cloud frame, denoted as $\boldsymbol{M}^t \in \mathbb{R}^{N\times 3}$, where each motion vector represents point displacement $\{\Delta x, \Delta y, \Delta z\}$. Among the existing self-supervised scene flow methods, we choose the recent FlowStep3D [31] which shows excellent scene flow estimation in multiple datasets.

Implementation details of above neural components are in appendix. Having the estimated object masks and scene flow at hand, the key component is the supervision mechanism to drive predicted masks to be true objectness, which is discussed in the following section.

3.2 Object Geometry Consistency Losses

Given the input cloud $\boldsymbol{P}^t$ and its estimated object masks $\boldsymbol{O}^t$ and motion $\boldsymbol{M}^t$, we introduce the following objectives to satisfy the geometry consistency on both frames $\boldsymbol{P}^t$ and $(\boldsymbol{P}^t + \boldsymbol{M}^t)$.

(1) Geometry Consistency over Dynamic Object Transformations

From time t to $t+1$, the rigid objects in point cloud frame $\boldsymbol{P}^t$ usually exhibit different dynamic transformations which can be described by matrices belonging to $SE(3)$ group. For the kth object, we firstly retrieve its (soft) binary mask $\boldsymbol{O}_k^t$, and then feed the tuple $\{\boldsymbol{P}^t, \boldsymbol{P}^t + \boldsymbol{M}^t, \boldsymbol{O}_k^t\}$ into the differentiable weighted-Kabsch algorithm [28, 18], estimating its transformation matrix $\boldsymbol{T}_k \in \mathbb{R}^{4\times 4}$.

In order to drive all object masks to fully explain the corresponding motion patterns within all masks, the following dynamic rigid loss is designed to minimize the discrepancy of every point scene flow between time t and $t+1$. Formally, it is defined as:

$$\ell_{dynamic} = \frac{1}{K}\sum_{k=1}^{K} \left\| \boldsymbol{O}_k^t * \left(\boldsymbol{T}_k \circ \boldsymbol{P}^t - (\boldsymbol{P}^t + \boldsymbol{M}^t)\right) \right\|_2 \tag{1}$$

where the operation $\circ$ represents the rigid transformation of point clouds. Intuitively, if one inferred object mask happens to include two sets of points with two different moving directions, the transformed point cloud can be only in favor of one moving direction, thereby resulting in higher errors. Therefore, the above constraint can push all object masks to fit the dynamic and diverse motion patterns. However, here arises a critical issue: a single rigid object may be assigned to multiple masks, *i.e.* oversegmentation. We alleviate this issue by a simple smoothness regularizer discussed below.

(2) Geometry Smoothness Regularization

The primary reason why a single object may be oversegmented is the lack of spatial connectivity between individual points. However, our common observation is that physically neighbouring points usually belong to a single object. In this regard, we simply introduce a geometry smoothness regularizer. Particularly, for a specific n^{th} 3D point $\boldsymbol{p}_n$ in the point cloud $\boldsymbol{P}^t$, we firstly search H points from its neighbourhood using either KNN or spherical querying methods, and then force their mask assignments to be consistent with the center point $\boldsymbol{p}_n$. Mathmatically, it is defined as:

$$\ell_{smooth} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{H}\sum_{h=1}^{H} d(\boldsymbol{o}_{p_n}, \boldsymbol{o}_{p_n^h})\right) \tag{2}$$

where $\boldsymbol{o}_{p_n} \in \mathbb{R}^{1\times K}$ represents the object assignment of center point $\boldsymbol{p}_n$, and $\boldsymbol{o}_{p_n^h} \in \mathbb{R}^{1\times K}$ represents the object assignment of its hth neighbouring point. The distance function $d()$ is flexible to choose the simple $L1$ and $L2$ or a more aggressive cross-entropy function.

(3) Geometry Invariance over Scene Transformations

With the above geometry constraints designed in (1)(2), the shapes of dynamic objects can be reasonably segmented. However, the learned object geometry may not be general enough. For example, a moving car can be well segmented, yet another similar parked car may not be discovered. To this end, we introduce an object geometry invariance constraint as follows:

- Firstly, given $\boldsymbol{P}^t$, we apply two transformations to get augmented point clouds $\boldsymbol{P}_{v1}^t$ and $\boldsymbol{P}_{v2}^t$.
- Secondly, we feed the two augmented point clouds into the neural backbone, obtaining two sets of object masks $\boldsymbol{O}_{v1}^t$ and $\boldsymbol{O}_{v2}^t$. Because the per-point locations in two point clouds are transformed differently, the position sensitive MaskFormer [63] naturally generates two different sets of masks.
- Thirdly, we leverage the Hungarian algorithm [33] to one-one match the individual masks in $\boldsymbol{O}_{v1}^t$ and $\boldsymbol{O}_{v2}^t$ according to the object pair-wise IoU scores. Basically, this is to address the issue that there is no fixed order for predicted object masks from the two augmented point clouds.
- At last, we reorder the masks in $\boldsymbol{O}_{v2}^t$ to align with $\boldsymbol{O}_{v1}^t$, and design the invariance loss as follows.

$$\ell_{invariant} = \frac{1}{N}\sum_{n=1}^{N} \hat{d}\big(\boldsymbol{o}_{v1}^n, \boldsymbol{o}_{v2}^n\big) \tag{3}$$

where $\boldsymbol{o}_{v1}^n$ and $\boldsymbol{o}_{v2}^n$ are the object assignments of the two augmented point clouds for a specific n^{th} point. The distance function $\hat{d}()$ is flexible to use $L1$, $L2$ or cross-entropy. Ultimately, this loss drives the estimated object masks to be invariant with different views of input point clouds.

3.3 Iterative Optimization of Object Segmentation and Motion Estimation

With the designed geometry consistency loss functions, the neural backbone for object segmentation is optimized by the combined loss: $\ell_{seg} = \ell_{dynamic} + \ell_{smooth} + \ell_{invariant}$. The auxiliary self-supervised scene flow network FlowStep3D [31] is independently trained by its own losses until convergence. Intuitively, with better and better object masks estimated, the estimated scene flow is also expected to be improved further if we use the masks properly. An ideal solution is to directly pass the estimated object masks into the scene flow network. However, this is non-trivial because we need to either adapt the scene flow network layers or add extra loss functions, and thus we leave it for our further exploration. In this paper, we instead propose a following Algorithm 1 to iteratively improve object segmentation and motion estimation.

Algorithm 1 Iterative optimization of object segmentation and scene flow estimation. Assume the whole train split has S point cloud pairs: $\{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1})_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1})_S\}$.

Stage 0: Initial scene flow estimation.

• Independently and fully train the self-supervised scene flow network on the whole training data split, and obtain reasonable scene flow estimations: $\{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_S\}$.

for number of iteration rounds R **do**

Stage 1: Object segmentation optimization.

• Train the object segmentation network using ℓ_{seg} for a total E epochs on the whole training split: $\{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_S\}$.

• Estimate reasonable object masks: $\{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t, \boldsymbol{O}^t)_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t, \boldsymbol{O}^t)_S\}$.

Stage 2: Scene flow improvement.

• For each pair of data $(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t, \boldsymbol{O}^t)$, by drawing insights from the classical ICP [2], we propose an **object-aware ICP** algorithm to estimate new scene flow $\hat{\boldsymbol{M}}^t$ for point cloud $\boldsymbol{P}^t$.

• Update the new scene flow for next round training:

$\{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \boldsymbol{M}^t)_S\} \leftarrow \{(\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \hat{\boldsymbol{M}}^t)_1 \cdots (\boldsymbol{P}^t, \boldsymbol{P}^{t+1}, \hat{\boldsymbol{M}}^t)_S\}$

Empirically, setting the total number of rounds R to be 2 or 3 has a good trade off between accuracy and efficiency. Due to the limited space, details of object-aware ICP algorithm is in appendix.

3.4 Implementation

Our object segmentation network is implemented in PyTorch and Adam optimizer with a learning rate of 0.001 is adopted. All experiments are conducted on a single RTX3090 GPU. For the geometry smoothness regularization, the choice of neighbouring points H is specified in appendix for different datasets. For the geometry invariance loss, the augmentation method is also detailed in appendix.

4 Experiments

Our method is evaluated on three different scenarios: 1) part instance segmentation of articulated objects on SAPIEN dataset [70], 2) object segmentation of indoor scenes on our own synthetic dataset, and 3) object segmentation of real-world outdoor scenes on KITTI datasets [45]. For evaluation metrics, we follow [48] and report the **F1-score**, **Precision**, and **Recall** with an IoU threshold of 0.5. In addition, we also report the Average Precision (**AP**) score following COCO dataset [37] and the Panoptic Quality (**PQ**) score defined in [30]. The mean Intersection over Union (**mIoU**) score and the Rand Index (**RI**) score implemented in [24] are also included. Note that, all metrics are computed in a class-agnostic manner.

4.1 Evaluation on SAPIEN Dataset

The SAPIEN dataset [70] provides 720 simulated articulated objects with part instance level annotations. Each object has 4 sequential scans. The part instances have different articulating (moving) states. We follow [24] to use the training data generated from [75]. In particular, there are 82092 pairs of point clouds for training, 2880 single point cloud frames for testing. Each point cloud is downsampled to 512 points in both training and testing.

Since there is no existing unsupervised method for multi-object segmentation on 3D point clouds, we firstly implement two classical clustering methods: WardLinkage [27] and DBSCAN [15] to directly

Table 1: Quantitative results of our method and baselines on the SAPIEN dataset.

		AP	PQ	F1	Prec	Recall	mIoU	RI
Supervised Methods	PointNet++ [54]	-	-	-	-	-	51.2	65.0
	MeteorNet [40]	-	-	-	-	-	45.7	60.0
	DeepPart [74]	-	-	-	-	-	53.0	67.0
	MBS [24]	-	-	-	-	-	67.3	77.0
	OGC$_{sup}$	63.9	46.6	59.1	51.3	69.6	66.4	77.2
Unsupervised Motion Segmentation	TrajAffn [49]	6.4	15.1	22.5	16.7	34.5	46.0	60.3
	SSC [48]	9.5	20.8	28.7	21.3	43.8	50.5	65.8
Unsupervised Methods	WardLinkage [27]	17.4	26.8	40.1	36.9	43.9	49.4	62.2
	DBSCAN [15]	6.3	13.5	20.4	14.0	38.0	34.2	51.4
	OGC(Ours)	**49.7**	**46.5**	**61.5**	**59.2**	**64.0**	**60.0**	**71.6**

group 3D points from single point clouds into objects. Secondly, we implement two classical motion segmentation methods: TrajAffn [49] and SSC [48]. Note that, these two methods take the same estimated scene flows of FlowStep3D as input, while our method uses the estimated scene flows during training only but takes single point clouds during testing. In addition, we also include the excellent results of several fully-supervised methods (PointNet++ [54], MeteorNet [40], DeepPart [74]) reported in MBS [24]. Their experimental details can be found in MBS [24]. Lastly, we train our object segmentation backbone using single point clouds with full annotations, denoted as OGC$_{sup}$. All implementation details are in appendix.

Analysis: As shown in Table 1, our OGC surpasses the classical clustering based and motion segmentation methods by large margins on all metrics, showing the advantage of our method in fully leveraging both the motion patterns and various types of geometry consistency. Compared with the fully supervised baselines, our method is only inferior to the strong MBS [24] and OGC$_{sup}$. However, we observe that our OGC actually shows a higher precision score than OGC$_{sup}$, primarily because our method tends to learn better objectness thanks to the additional motion pattern constraints and avoid dividing a single object into pieces. Figure 3 shows qualitative results.

4.2 Evaluation on Indoor Scene Dataset

We further evaluate our method to segment objects in indoor 3D scenes. Considering that the existing dataset FlyingThings3D [44] tends to have unrealistically cluttered scenes with severely fragmented objects and it is originally introduced for scene flow estimation, we turn to synthesize a new dynamic room dataset, called **OGC-DR**, that suits both scene flow estimation and object segmentation. In particular, we follow [51] to randomly place $4 \sim 8$ objects belonging to 7 classes of ShapeNet [6] {chair, table, lamp, sofa, cabinet, bench, display} into each room. In total, we create 3750, 250, and 1000 indoor rooms (scenes) for training/validation/test splits. In each scene, we create rigid dynamics by applying continuous random transformations to each object and record 4 sequential frames for evaluation. Each point cloud frame is downsampled to 2048 points. Note that, we follow [11] to split different object instances for train/val/test sets. More details of the dataset are in appendix and this dataset will be released.

Analysis: As shown in Table 2, our method outperforms all classical unsupervised methods including the clustering based and the motion segmentation based methods. Since our synthetic rooms all have complete 3D objects, and the generated point cloud sequences are of high quality. Therefore, the performance gap between our OGC and the supervised OGC$_{sup}$ is nearly closed. This shows that the rigid dynamic motions can indeed provide sufficient supervision signals, if not more than the human annotations, to identify objects. Figure 3 shows qualitative results.

4.3 Evaluation on KITTI Benchmarks

We additionally evaluate our method on the challenging real-world outdoor dataset. In particularly, we conduct the following two groups of experiments on two benchmarks of KITTI dataset.

- Group 1: Officially, KITTI-SF (Scene Flow) dataset [45] consists of 200 (training) pairs of point clouds from real-world traffic scenes and an online hidden test for scene flow estimation. In

Table 2: Quantitative results of our method and baselines on our OGC-DR dataset.

		AP	PQ	F1	Pre	Rec	mIoU	RI
Supervised Methods	OGC$_{sup}$	91.6	84.1	88.9	86.0	92.0	89.4	97.7
Unsupervised Motion Segmentation	TrajAffn [49]	32.3	43.6	54.4	66.3	46.2	44.4	78.5
	SSC [48]	70.1	77.4	82.3	87.1	78.1	75.4	92.2
Unsupervised Methods	WardLinkage [27]	68.3	73.2	82.0	**93.4**	73.1	69.1	94.0
	DBSCAN [15]	66.3	75.8	81.3	85.6	77.4	74.6	91.4
	OGC(Ours)	**91.3**	**83.5**	**88.5**	84.9	**92.5**	**89.2**	**97.2**

our experiment, we train our pipeline on the first 100 pairs of point clouds, and then test on the remaining 100 pairs (200 single point clouds). We observe that in the 100 training pairs, the moving objects are only cars and trucks. Therefore, in the testing phase, we only keep the human annotations [1] of cars and trucks in every single frame to compute the scores. All other objects are treated as part of background. Note that, the whole background is not ignored, but counted as one object in our evaluation.

- Group 2: Given our well trained model in Group 1, we directly test it on the popular single frame based KITTI 3D Object Detection benchmark [17]. In this benchmark, there are officially 3712 point clouds for training, 3769 for validation. Again, we only keep the ground truth object masks obtained from the bounding boxes for the car category in each frame. All other objects are treated as part of background. For comparison, we download the official pretrained models of three fully-supervised methods PointRCNN [56], PV-RCNN [55] and Voxel-RCNN [12] to directly test on the validation split using the same settings as ours. In addition, we also directly use the well trained OGC$_{sup}$ in Group 1 to directly test for comparison, denoted as OGC*$_{sup}$. At last, we train the OGC$_{sup}$ on the training split (3712 frames) from scratch and test it on the remaining 3769 frames using the same evaluation settings, denoted as OGC$_{sup}$.

Table 3: Quantitative results of our method and baselines on the KITTI-SF dataset.

		AP	PQ	F1	Pre	Rec	mIoU	RI
Supervised Methods	OGC$_{sup}$	58.9	52.4	63.7	66.3	61.3	64.1	94.8
Unsupervised Motion Segmentation	TrajAffn [49]	22.4	34.7	42.8	40.7	45.2	48.8	83.5
	SSC [48]	0.6	5.1	7.5	6.2	9.4	19.3	33.2
Unsupervised Methods	WardLinkage [27]	0.4	2.4	3.8	2.2	14.4	26.9	15.8
	DBSCAN [15]	12.6	29.8	32.8	46.5	25.3	31.1	84.7
	OGC(Ours)	**50.1**	**40.5**	**51.1**	**46.5**	**56.8**	**62.0**	**93.1**

Table 4: Quantitative results on KITTI-Det (* denotes the model trained on KITTI-SF).

		AP	PQ	F1	Pre	Rec	mIoU	RI
Supervised Methods	PointRCNN [56]	94.9	79.9	88.9	81.3	98.0	91.2	97.2
	PV-RCNN [55]	95.4	77.2	84.4	73.7	98.7	92.6	97.1
	Voxel-RCNN [12]	95.8	79.5	87.2	78.1	98.9	92.5	97.3
	OGC$_{sup}$	79.9	68.2	77.7	70.3	87.0	85.4	97.0
	OGC*$_{sup}$	48.2	41.5	49.0	46.9	51.3	63.9	90.3
Unsupervised Method	**OGC(Ours)***	**41.0**	**30.9**	**37.7**	**31.4**	**47.0**	**59.9**	**85.0**

Analysis: As shown in Table 3, the classical unsupervised clustering based and motion segmentation based baselines catastrophically fail on the challenging KITTI-SF dataset. Primarily, this is because SSC cannot handle the extreme imbalance of 3D points between objects and background. WardLinkage and DBSCAN favor spatially separated objects, but the background and objects in KITTI-SF are always connected because of the Earth's gravity. By comparison, our method obtains superior segmentation scores, being very close to our fully-supervised counterpart OGC$_{sup}$. As shown in Table 4, our method can directly generalize to the single frame based 3D object segmentation with satisfactory results, also being close to the fully-supervised counterpart OGC*$_{sup}$. It is understandable that the other three fully-supervised models have a clear advantage over ours, because they are fully trained on the similar training split while ours does not. We hope that our method can serve the first

baseline and inspire more advanced unsupervised methods in the future to close the gap. Figure 3 shows qualitative results.

4.4 Ablation Study

(1) Geometry Consistency Losses: To validate the choice of our design, we firstly conduct three groups of ablative experiments on the SAPIEN dataset [70]: 1) only remove the dynamic rigid loss $\ell_{dynamic}$, 2) only remove the smoothness loss ℓ_{smooth}, and 3) only remove the invariance loss $\ell_{invariant}$. As shown in Table 5, combining the proposed three losses together gives the highest segmentation scores. Basically, the dynamic rigid loss serves to discriminate multiple objects from different motion patterns. Without it, the network tends to assign all points to a single object as the shortcut to minimize the other two losses. However, we observe in Table 5 that without $\ell_{dynamic}$, the network still works to some extend. This is because the synthetic SAPIEN dataset tends to have a number of point cloud frames with only 2 or 3 objects, thus assigning all points to a single object can still get plausible scores. This issue is further validated by conducting additional ablation experiments on curated SAPIEN dataset. More details are in appendix.

Table 5: Ablation studies about loss designs on SAPIEN.

	AP	PQ	F1	Pre	Rec	mIoU	RI
w/o $\ell_{dynamic}$	35.4	35.4	54.2	**91.3**	38.6	28.6	52.7
w/o ℓ_{smooth}	19.7	18.6	27.3	19.3	46.5	52.3	64.5
w/o $\ell_{invariant}$	44.5	43.1	58.0	56.9	59.2	56.5	69.6
Full OGC	**49.7**	**46.5**	**61.5**	59.2	**64.0**	**60.0**	**71.6**

(2) Iterative Optimization Algorithm: We also conduct ablative experiments to validate the effectiveness of our proposed Algorithm 1. We set the number of iterative rounds R as $\{1, 2, 3\}$. As shown in Table 6, after 2 rounds, satisfactory segmentation results can be achieved, although we expect better results after more rounds with longer training time. This shows that our iterative optimization algorithm can indeed fully leverage the mutual benefits between object segmentation and motion estimation.

Table 6: Iterative optimization on SAPIEN.

	Object Segmentation						
$\#R$	AP	PQ	F1	Pre	Rec	mIoU	RI
1	39.9	39.2	53.6	48.8	59.5	57.2	69.3
2	49.7	46.5	61.5	59.2	64.0	60.0	71.6
3	52.4	47.0	62.0	60.4	63.7	58.8	71.8

4.5 Pushing the Boundaries of Unsupervised Scene Flow Estimation and Segmentation

In addition to the improvement of object segmentation from our iterative optimization algorithm, the scene flow estimation can be naturally further improved from our estimated object masks as well. Given our well trained model on the KITTI-SF dataset in Section 4.3, we use the estimated object masks to further improved the scene flow estimation. As shown in Table 7, following the exact evaluation settings of FlowStep3D [31], our method, not surprisingly, significantly boosts the scene flow accuracy, surpassing the state-of-the-art unsupervised FlowStep3D [31] and other baselines in all metrics.

Table 7: Scene flow estimation on the KITTI-SF dataset.

	EPE3D	AccS	AccR	Outlier
Ego-motion [62]	41.54	22.09	37.21	80.96
PointPWC-Net [69]	25.49	23.79	49.57	68.63
FlowStep3D [31]	10.21	70.80	83.94	24.56
OGC(Ours)	**7.11**	**78.51**	**87.35**	**23.98**

In fact, our object segmentation backbone is also flexible to take the scene flow as input instead of point xyz to segment objects. This is commonly called motion segmentation. We replace the single point clouds by (estimated) scene flow vectors as our network inputs, and train the network from scratch using the same settings on the KITTI-SF dataset. As shown in Table 8, we can see that our network can still achieve superior results regardless of the modality of inputs, demonstrating the generality of our framework.

Table 8: Motion vs points based segmentation on KITTI-SF.

Input	AP	PQ	F1	Pre	Rec	mIoU	RI
scene flow	49.1	**42.2**	**52.7**	**53.1**	52.3	56.7	90.7
point cloud	**50.1**	40.5	51.1	46.5	**56.8**	**62.0**	**93.1**

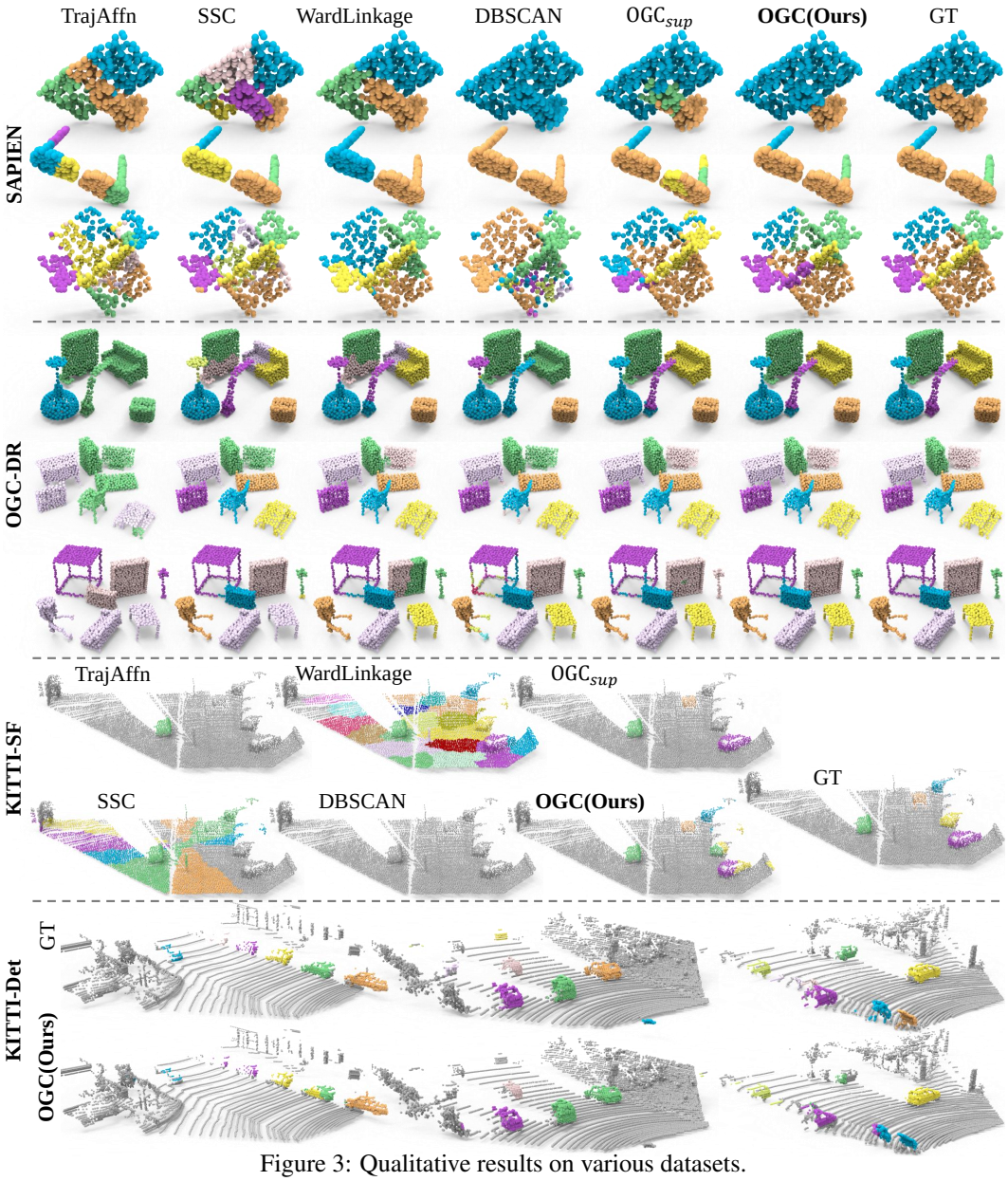

Figure 3: Qualitative results on various datasets.

5 Conclusion

In this paper, we demonstrate for the first time that 3D objects can be accurately segmented using an unsupervised method from raw point clouds. Unlike the existing approaches which usually rely on a large amount of human annotations of every 3D object for training networks, we instead turn to leverage the diverse motion patterns over sequential point clouds as supervision signals to automatically discover the objectness from single point clouds. A series of loss functions are designed to preserve the object geometry consistency over spatial and temporal scales. Extensive experiments over multiple datasets including the extremely challenging outdoor scenes demonstrate the effectiveness of our method. One limitation of our method is the lack of capability to discover unseen object types, and we leave it for our future exploration.

Broader Impact: The proposed OGC learns 3D objects from raw point clouds without requiring human annotations for supervision. We showcase the effectiveness for some basic applications including object part instance segmentation, indoor object segmentation and outdoor moving vehicle identification. We also believe that our method can be general for other domains such as AR/VR.

References

[1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *IJCV*, 126:961–972, 2018.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *TPAMI*, (5):698 – 700, 1987.

[3] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. SLIM: Self-Supervised LiDAR Scene Flow and Motion Segmentation. *ICCV*, 2021.

[4] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. PointFlowNet: Learning Representations for 3D Scene Flow Estimation from Point Clouds. *CVPR*, 2019.

[5] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv:1901.11390*, 2019.

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv*, 2015.

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-View 3D Object Detection Network for Autonomous Driving. *CVPR*, 2017.

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. *NeurIPS*, 2021.

[9] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. SegFlow: Joint Learning for Video Object Segmentation and Optical Flow. *ICCV*, 2017.

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. *CVPR*, 2019.

[11] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. *ECCV*, 2016.

[12] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. *AAAI*, 2021.

[13] David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. Attention over learned object embeddings enables complex visual reasoning. *NeurIPS*, 2021.

[14] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. *NIPS*, 2016.

[15] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 1996.

[16] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: A survey. *CVIU*, 134(1-21), 2015.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. *CVPR*, 2012.

[18] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multiview 3D point cloud registration. *CVPR*, 2020.

[19] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *CVPR*, 2018.

[20] Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loie Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *ICML*, 2019.

[21] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. HPLFlowNet: Hierarchical Permutohedral Lattice FlowNet for Scene Flow Estimation on Large-scale Point Clouds. *CVPR*, 2019.

[22] Jun Ting Hsieh, Bingbin Liu, De An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *NIPS*, 2018.

[23] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *CVPR*, 2020.

[24] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi Min Hu, and Leonidas Guibas. MultiBodySync: Multi-Body Segmentation and Motion Estimation via 3D Scan Synchronization. *CVPR*, 2021.

[25] Cansen Jiang, Danda Pani Paudel, David Fofi, Yohan Fougerolle, and Cedric Demonceaux. Moving Object Detection by 3D Flow Field Analysis. *TITS*, 22(4):1950–1963, 2021.

[26] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. SCALOR: Generative World Models with Scalable Object Representations. *ICLR*, 2020.

[27] Joe H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[28] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[29] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. *ICLR*, 2022.

[30] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CVPR*, 2019.

[31] Yair Kittenplon, Yonina C. Eldar, and Dan Raviv. FlowStep3D: Model Unrolling for Self-Supervised Scene Flow Estimation. *CVPR*, 2021.

[32] Adam R. Kosiorek, Hyunjik Kim, Ingmar Posner, and Yee Whye Teh. Sequential attend, infer, repeat: Generative modelling of moving objects. *NeurIPS*, 2018.

[33] Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[34] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. *CVPR*, 2019.

[35] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle Detection from 3D Lidar Using Fully Convolutional Network. *RSS*, 2016.

[36] Ruibo Li, Guosheng Lin, and Lihua Xie. Self-Point-Flow: Self-Supervised Scene Flow Estimation from Point Clouds with Optimal Transport and Random Walk. *CVPR*, 2021.

[37] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *ECCV*, 2014.

[38] Runtao Liu, Zhirong Wu, Stella X Yu, and Stephen Lin. The Emergence of Objectness : Learning Zero-Shot Segmentation from Videos. *NeurIPS*, 2021.

[39] Xingyu Liu, Charles R. Qi, and Leonidas J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. *CVPR*, 2019.

[40] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. MeteorNet: Deep Learning on Dynamic 3D Point Cloud Sequences. *ICCV*, 2019.

[41] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. *NeurIPS*, 2020.

[42] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J. Crandall, and Steven C. H. Hoi. Learning Video Object Segmentation From Unlabeled Videos. *CVPR*, 2020.

[43] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-Supervised Pillar Motion Learning for Autonomous Driving. *CVPR*, 2021.

[44] Nikolaus Mayer, Eddy Ilg, H Philip, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A Large Dataset to Train Convolutional Networks for Disparity , Optical Flow , and Scene Flow Estimation. *CVPR*, 2016.

[45] Moritz Menze, Andreas Geiger, and T Mpi. Object Scene Flow for Autonomous Vehicles. *CVPR*, 2015.

[46] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee. Unsupervised Learning of Object Structure and Dynamics from Videos. *NeurIPS*, 2019.

[47] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. *CVPR*, 2020.

[48] Urbano Miguel Nunes and Yiannis Demiris. 3D motion segmentation of articulated rigid bodies based on RGB-D data. *BMVC*, 2018.

[49] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 36(6):1187–1200, 2014.

[50] Bojun Ouyang and Dan Raviv. Occlusion guided scene flow estimation on 3D point clouds. *CVPR*, 2021.

[51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional Occupancy Networks. *ECCV*, 2020.

[52] Gilles Puy, Alexandre Boulch, and Renaud Marlet. FLOT: Scene Flow on Point Clouds Guided by Optimal Transport. *ECCV*, 2020.

[53] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *CVPR*, 2017.

[54] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NIPS*, 2017.

[55] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *CVPR*, 2019.

[56] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. *CVPR*, 2019.

[57] Weijing Shi, Ragunathan, and Rajkumar. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. *CVPR*, 2020.

[58] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *NIPS*, 2010.

[59] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered segmentation and optical flow estimation over time. *CVPR*, 2012.

[60] Hugues Thomas, Ben Agro, Mona Gridseth, Jian Zhang, and Timothy D. Barfoot. Self-Supervised Learning of Lidar Segmentation for Autonomous Indoor Navigation. *ICRA*, 2021.

[61] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. *ICCV*, 2019.

[62] Ivan Tishchenko, Sandro Lombardi, Martin R. Oswald, and Marc Pollefeys. Self-supervised learning of non-rigid residual flow and ego-motion. *3DV*, 2020.

[63] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A N. Gomez, L Kaiser, and I Polosukhin. Attention Is All You Need. *NIPS*, 2017.

[64] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D Instance Segmentation on Point Clouds. *CVPR*, 2022.

[65] Johan Wagemans, James H. Elder, Michael Kubovy, Stephen E. Palmer, Mary A. Peterson, Manish Singh, and Rüdiger von der Heydt. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychological Bulletin*, 138(6):1172–1217, 2012.

[66] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. *CVPR*, 2018.

[67] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. PV-RAFT: Point-Voxel Correlation Fields for Scene Flow Estimation of Point Clouds. *CVPR*, 2021.

[68] Max Wertheimer. Untersuchungen zur Lehre yon der Gestalt. *Psychologische Forschung*, (4):301–350, 1923.

[69] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. PointPWC-Net: Cost Volume on Point Clouds for (Self-)Supervised Scene Flow Estimation. *ECCV*, 2020.

[70] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A SimulAted Part-based Interactive ENvironment. *CVPR*, 2020.

[71] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. *NeurIPS*, 2019.

[72] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised Video Object Segmentation by Motion Grouping. *ICCV*, 2021.

[73] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. *CVPR*, 2019.

[74] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *SIGGRAPH Asia*, 37(6), 2018.

[75] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 2016.

[76] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. CorrNet3D: Unsupervised End-to-end Learning of Dense Correspondence for 3D Point Clouds. *CVPR*, 2021.

[77] Mingliang Zhai, Xuezhi Xiang, Ning Lv, and Xiangdong Kong. Optical Flow and Scene Flow Estimation: A Survey. *Pattern Recognition*, 2021.

[78] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *CVPR*, 2018.

[79] Daniel Zoran, Rishabh Kabra, and Danilo J Rezende. PARTS: Unsupervised segmentation with slots , attention and independence maximization. *ICCV*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported because it would be too computationally expensive
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]