https://doi.org/10.2352/EI.2025.37.15.AVM-115

LiDAR Panoptic Segmentation for Autonomous Driving: A Survey

Aditya Dusi, Bassam Helou; Stanford University, Motional AD. Inc; adusi@stanford.edu, bassam.helou@motional.com

Abstract

This survey provides a comprehensive overview of LiDAR-based panoptic segmentation methods for autonomous driving. We motivate the importance of panoptic segmentation in autonomous vehicle perception, emphasizing its advantages over traditional 3D object detection in capturing a more detailed and comprehensive understanding of the environment. We summarize and categorize 42 panoptic segmentation methods based on their architectural approaches, with a focus on the kind of clustering utilized: machine learned or non-learned heuristic clustering. We discuss direct methods, most of which use single-stage architectures to predict binary masks for each instance, and clustering-based methods, most of which predict offsets to object centers for efficient clustering. We also highlight relevant datasets, evaluation metrics, and compile performance results on SemanticKITTI and panoptic nuScenes benchmarks. Our analysis reveals trends in the field, including the effectiveness of attention mechanisms, the competitiveness of center-based approaches, and the benefits of sensor fusion. This survey aims to guide practitioners in selecting suitable architectures and to inspire researchers in identifying promising directions for future work in LiDAR-based panoptic segmentation for autonomous driving.

Introduction

To operate safely, autonomous vehicles need to perceive and interpret complex real-world scenes in real time. Perception is tasked with this crucial responsibility. Many downstream subsystems in an autonomous vehicle software stack, such as localization and planning, rely on it. Most perception systems use LiDAR sensors because they provide high-resolution, three-dimensional point cloud data that captures the geometry and spatial distribution of objects in the environment. Most also use 3D object detection methods to identify and localize objects like cars and pedestrians with 3D bounding boxes. However, bounding boxes are coarse and may contain irrelevant background points (e.g., parts of the road or nearby objects) within the box. This can cause undesirable behavior from the autonomous vehicle around out-of-distribution or abnormally shaped objects, like the side supports of a crane or a long L-shape connected barrier as shown in figure 1. In that figure, small errors in the predicted bounding box lead to harsh and unnecessary braking.

An alternative richer representation of the environment is gaining popularity: panoptic segmentation. As we show in figure 2, this representation consists of classifying each point in a scene into 1 of 2 categories: "things" (distinct objects like cars, pedestrians, and cyclists) and "stuff" (background classes like road surfaces and vegetation) while also associating different points belonging to the same object to create an instance. This no longer limits our representation to cuboids and allows for a more free form, geomety-centric representation. In addition, it can be a more straightforward task than predicting bounding boxes. With segmentation, a network just has to pick one of a number of possible classes compared to regressing many attributes for bounding boxes, some of which are difficult to discern such as the orientation of an object. Semantic segmentation is an approach prioritized by Tesla in their occupancy network [1]. Instance segmentation is also important because, for the downstream tasks of tracking, prediction and planning, we want to discriminate between different instances of objects of a particular class for object association.

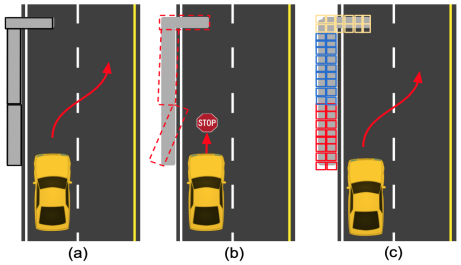
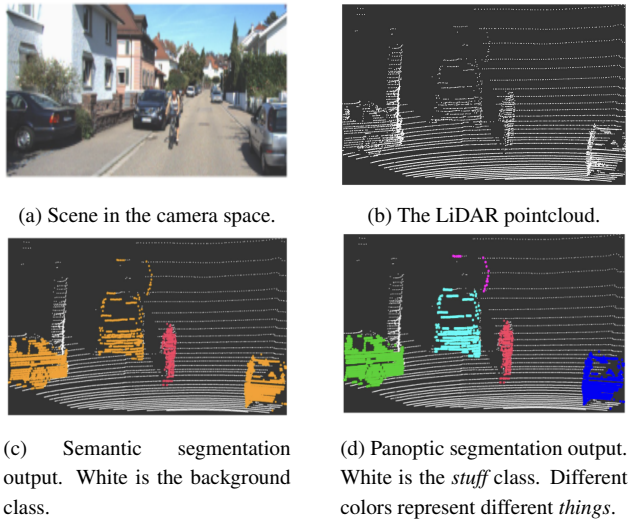


Figure 1: Pitfalls of using a bounding box as a universal representation. In this scenario, ego needs to make a lane change to avoid the barrier. Subfigure (a) the intended behavior, (b) what happens when barriers are detected as boxes and have some minor orientation errors and (c) how semantic and instance segmentation can aid in this scenario via per voxel prediction. Different colors represent different instances.



(a) Scene in the camera space.

(b) The LiDAR pointcloud.

(c) Semantic segmentation output. White is the background class.

(d) Panoptic segmentation output. White is the *stuff* class. Different colors represent different *things*.

Figure 2: Example of point-level panoptic segmentation. The image and pointcloud are adapted, with permission, from Ref. [2].

Most surveys on panoptic segmentation focus on images [3, 4, 5] and superficially discuss applications in autonomous driving. Ref. [6] focuses on autonomous driving, but is not as comprehensive and detailed as this survey in analyzing

and categorizing different LiDAR-based panoptic segmentation methods. They only discuss 5 LiDAR-based networks compared to 42 in this article. Our main contributions include providing a summary of most LiDAR based panoptic segmentation methods in the literature and comparison of these methods on the SemanticKITTI and panoptic nuScenes datasets. We also showcase key performance trends.

Datasets

Many LiDAR based panoptic segmentation datasets for autonomous driving exist, such as the Waymo open dataset [7] and WADS [8]. In this paper, however, we present details on the two datasets which are the most widely used to benchmark LiDAR panoptic segmentation.

SemanticKITTI

SemanticKITTI [9] is a large scale dataset with semantic labels for point clouds with 360° field of view around the vehicle. 22 scenes are provided, 11 for training and 11 for test, comprising of 23201 and 20351 samples respectively. There are 28 classes for semantic segmentation.

nuScenes

nuScenes [10, 11] comprises of 1000 scenes with 23 classes for detection and 32 classes for LiDAR panoptic segmentation. The scenes consist of 28130 samples for training, 6019 samples for validation, and 6008 samples for testing.

Metrics

Panoptic Quality (PQ) is the most commonly used metric to compare the performance of LiDAR-based panoptic segmentation methods. Other metrics used include average precision, average recall and mean intersection over union, but we only use PQ in this work because it is the most widely reported metric.

Panoptic Quality

Panoptic quality (PQ) is a single number aggregated over all classes that provides an overview of the panoptic segmentation performance of a model.

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}|}}_{\mathrm{SQ}} \times \underbrace{\frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}}_{\mathrm{RQ}}, \tag{1}$$

where p is a predicted set of points belonging to the same instance and category, and g is a set of points belonging to a ground-truth instance of the same category. SQ and RQ are segmentation quality and recognition quality, respectively. Intersection over union (IoU) is defined as

$$\mathrm{IoU} = \frac{p \cap g}{p \cup g}. \tag{2}$$

A pair (p,g) is said to be a true positive (TP), if the IoU between the ground truth and predicted set of points is ≥ 0.5. False positives (FP) are all predicted segments that don't have a ground truth to match to, and false negatives (FN) are all ground truths that don't have a prediction matched to them. $|\mathrm{TP}|$, $|\mathrm{FP}|$, $|\mathrm{FN}|$ are the number of TP, FP, and FN, respectively.

We also report $\mathrm{PQ}^{\mathrm{st}}$ and $\mathrm{PQ}^{\mathrm{th}}$ which refer to the panoptic quality of things and stuff, respectively.

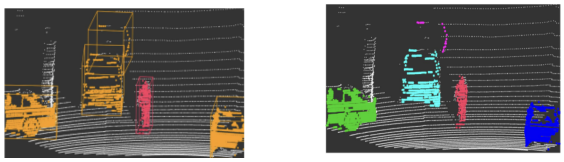
Direct Methods

Direct methods perform panoptic segmentation with minimal post-processing and usage of heuristic algorithms. Such methods fall into two major categories: two-stage and single-stage methods.

Two-stage methods

Two-stage methods first propose regions of interest, and then process these proposals to obtain semantic and instance labels. MOPT [12] and EfficientLPS [13] build on EfficientPS [14] and the popular two-stage Mask R-CNN [15] architecture to perform panoptic segmentation. Mask R-CNN processes a feature map to generate region proposals for objects. A branch processes the proposals to predict a binary mask of the object's shape. Mask R-CNN only provides instance level labels. To obtain panoptic labels, MOPT and EfficientLPS use the same strategy as EfficientPS [14]. An additional head computes semantic segmentation logits, which are combined with instance level logits through a panoptic fusion module.

MOPT introduces the novel task of multi-object panoptic tracking, which unifies semantic segmentation, instance segmentation, and multi-object tracking into a single framework. EfficientLPS uses many of the same architecture components as MOPT such as 2-way Feature Pyramid Networks [14] and separable convolutions. EfficientLPS also introduces many techniques to address challenges in LiDAR data, such as distance-dependent sparsity, scale variation, and occlusions. One such technique is a panoptic periphery loss that refines the boundaries between objects, ensuring more accurate separation of foreground objects from the background.



(a) Sem. seg. and detection bboxes. (b) Panoptic segmentation output.

Figure 3: Using bounding boxes and semantic segmentation to perform instance segmentation.

Single-stage methods

To reduce complexity and runtime, some methods directly predict instance IDs in a single stage. Most architectures are inspired by MaskFormer [16] and use learnable queries to directly predict binary masks and semantic classes for each instance.

One exception is Panoster [17] which directly predicts instance ids and semantic labels from a traditional CNN architecture and with loss functions inspired by confusion matrices. PUPS [18] also uses a unique architecture consisting of point-level classifiers. PUPS uses bipartite matching during training to ensure unique assignments for each instance without the need for post-processing. Transformer decoders refine the classifiers by querying point features.

MaskRange [19] uses the range view to transform the LiDAR pointcloud to a pseudo-image that can be directly processed by a MaskFormer like architecture with small modifications (such as a novel data augmentation technique). Range view is a dense and compact 2D representation of pointclouds. Unlike bird's eye view

(BEV) methods, its runtime does not increase with the detection range. However, in the range view, object shapes are not distance-invariant, and objects may overlap heavily with each other making occlusion harder to deal with.

The following methods employ different strategies to tailor MaskFormer to LiDAR pointclouds. MaskPLS [20] (and its extension to tracking Mask4D [21]) map multi-scale voxel features into point features that go into transformer decoders. P3Former [22] leverages novel position embeddings, that are a mixture of cartesian and polar coordinates, to better differentiate objects (especially small objects). 4D-Former [23] uses fusion with images to improve performance. DQFormer [24] initializes things and stuff queries differently because things are typically concentrated in local regions whereas stuff have distributed pointclouds with distinct geometries (e.g. cars vs road surface).

These methods have limitations. There is a limit on the number of masks and instances that the network can predict. Moreover, methods based on the MaskFormer architecture might require network pruning, quanitization or compression steps before they can be deployed on the car. Perception from LiDAR point clouds for autonomous driving utilize networks that have to balance multiple tasks, such as object detection and 3D map element detection, on a strict runtime budget. Thus, adding a large panoptic segmentation head might be infeasible. Furthermore, image based methods have to deal with perspective projection, whereas BEV LiDAR methods have the advantage of preserving object sizes. Methods that leverage clustering algorithms in one form or another can take advantage of the geometric properties of pointclouds, have competitive performance, and can be added to existing LiDAR-based networks without adding too much overhead.

Clustering-based Methods

In this section, we categorize methods that perform any kind of clustering as a heuristic post-processing operation that is not part of a neural network.

Proposals from bounding boxes

The earliest panoptic segmentation methods leveraged existing semantic segmentation and object detection methods to obtain simple baselines. Figure 3 captures the main idea of such methods, which first generate proposals from 3D bounding boxes and then group points based on their features. The simplest methods, Refs. [25] and [26], choose the features to be the semantic label of each point and the distance of each point to the center of the proposal region.

To improve performance and showcase the capability of weakly supervised methods, VIN [27] adds an additional post-processing step to resolve inconsistencies between bounding box and semantic segmentation predictions. LidarMultiNet [28] refines the semantic segmentation and bounding box classification scores through a second stage network which uses BEV and sparse voxel features from the first stage.

Non-learned heuristic clustering

A simple baseline is to cluster a pointcloud with heuristic, non-learned, algorithms. Ref. [29] surveys a few such algorithms. The algorithms make use of per point class labels obtained from a neural network, Cylinder3D [30], to filter out background points so that only foreground points are fed to the clustering algorithms. The same authors in Divide-and-Merge [31] propose a range view based clustering algorithm. They divide the image into small regions and perform local clustering. By voting on the edges of objects, they decide if different object clusters should be merged.

Such algorithms can be easy to implement. However, they can have a lot of parameters to tune. Performance is also limited and, unlike some of the learned methods we present later, does not improve with higher volume of data.

In the coming section, we will discuss methods that mitigate these drawbacks by producing learnable features that can be more easily clustered with heuristic algorithms.

Clustering with center offsets

Many methods predict center offsets to help cluster instances. Center offsets are 2D or 3D vectors predicted per pillar or voxel or point to the center of an object, which is typically the center of the bounding box of the object.

With center heatmap

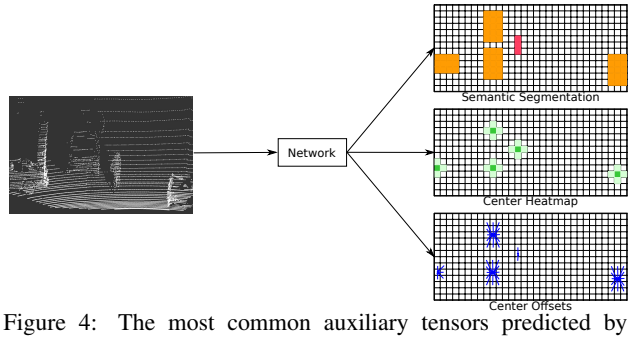


Figure 4: The most common auxiliary tensors predicted by LiDAR based networks for panoptic segmentation. For ease of illustration, we choose pillars as the output modality. The different shades of green in the center heatmap represent the confidence of the network in a pillar being the center of a thing with darker shades denoting higher confidence. We do not show all center offsets to reduce clutter.

As we show in figure 4, many approaches predict a center heatmap along with center offsets. A center heatmap is a probability distribution across all locations, indicating the likelihood of the center of an object being present at each point. These centers can be used to associate different voxels, pillars, or points to a single instance.

Panoptic-PolarNet [32] is one of the first methods that applies predicted center offsets to each BEV pillar, and then groups the displaced pillars based on the closest center heatmap peak. They were inspired by Ref. [33] which first applied this idea to images.

In addition to center offsets and a center heatmap, EvLPSNet [34] predicts uncertainties in the semantic output per voxel. To reduce discretization errors due to the BEV grid structure, EvLPSNet then selects the most uncertain voxels and refines their labels with a KPconv-based network [35].

With temporal tracking of panoptic segmentation IDs as their main aim, Ref. [36] presents a technique to modify a network so it better satisfies the symmetries of center heatmaps and center offsets. Specifically, if we rotate the pointcloud of an object, the

object's center remains the same, and the center offsets at each point of the object form an equivariant field which rotates in the same way as the object's pointcloud.

To reduce oversegmentation, GMM-PanopticSeg [37] presents a method that can be integrated into networks that predict center offsets and instance centers. Oversegmentation is the issue where a single object (such as a bus) is divided up into multiple clusters. GMM-PanopticSeg models intra-instance variance through Gaussian mixture models (GMM). This enables the model to handle challenges like noise and occlusion more effectively. Clustering is done by associating points with the most likely instance on the basis of their GMM distribution.

Some methods leverage range view for panoptic segmentation. In P-RangeFormer [38], the authors propose novel augmentations to address the dearth of data. Their contributions include the idea of splitting one range image into multiple non-overlapping sub-images, and a novel rasterization method to mitigate deformation issues in the range image. Ref. [39] produces centerness and center offsets which are used by an iterative algorithm for instance segmentation. They also propose two modules. One helps with label assignment for object detection. The second is an additional BEV regression head to improve 3D bounding box predictions.

Multi-modal sensor fusion between LiDAR and cameras helps overcome pitfalls of each sensor. Images are rich in semantic information but lack depth whereas pointclouds provide accurate geometry and depth but lack color and texture. LCPS [40] proposes different feature alignment modules to fuse camera features to voxelized LiDAR features. These are learnt modules as opposed to the heuristic modules conventionally employed in sensor fusion works. In addition to just using sensor fusion, there are methods that utilize multiple LiDAR views. Multi-view fusion improves performance by providing the network complementary information from multiple perspectives of the pointcloud, such as range view and cartesian voxels. UniSeg [41] leverages multi-modal sensor fusion (camera and LiDAR) and multi-view fusion to predict a center heatmap, center offsets and class labels.

Without center heatmap

Some methods do not directly predict a center heatmap but instead, predict offsets to the instance center, and use them, if needed, to find instances' centers.

Ref. [61] encodes a pointcloud into a bird's eye view feature map, and then predicts for each cell offsets to the center of the nearest object. If two cells predict offsets to nearby centers, then the points inside these two cells are merged into the same instance.

LPS [42] predicts center offsets and semantic labels in the range view. For each pixel in the range view, they predict an embedding vector, center offsets and a radius to be used for clustering. Things points are then sampled iteratively and the embedding vector and center offsets are used to perform clustering in a given radius

Panoptic PH-Net [47] utilizes a voxel encoder as well as a BEV encoder. A k-NN transformer is applied to estimate instance center offsets per voxel. Centers are derived from these offsets.

SMAC-Seg [44] proposes a clustering architecture that is computationally efficient, and that mitigates the information loss from projecting a pointcloud to the range view. SMAC-Seg first applies an instance mask to keep only foreground points, and then predicts center offsets for these points. Foreground points are shifted using the predicted offsets, and then projected to the BEV because it preserves spatial relationships making it easier to cluster points. The shifted points are then further refined through an attention module. A breadth-first search algorithm is finally applied to obtain object clusters. To ensure that the clusters are well-separated, SMAC-Seg introduces a centroid-aware repel loss. The loss penalizes points that are too close to the centroids of neighboring objects, and pulls objects closer to their respective object centroids.

A contrastive loss can be added to ensure that features of voxels belonging to the same instance have similar features as compared to objects of another instance. Authors in PVCL [48] apply a contrastive loss with an anchor, positive and negative to the output features. Positives and negatives are picked based on semantic segmentation outputs. This is akin to a triplet loss.

DS-Net [46] addresses the limitations Mean Shift faces in dealing with non-uniform point clouds and varying instance sizes (for example, compare the size of a compact car to that of a truck), which often results in oversegmentation. In Mean Shift, a fixed kernel is applied to iteratively shift points toward cluster centers. Instead of using a single kernel, DS-Net adaptively selects kernel bandwidths for each point. DS-Net learns multiple candidate bandwidths and combines them based on learned weights to shift each point toward their correct instance centers.

Authors in Contrastive instance association [62] use the same backbone as in DS-Net [46] but propose a contrastive learning module to better track and associate instances across different frames, improving temporal consistency of dynamic objects.

In CFNet [49], the network predicts center offset and center offset confidence scores to denote the accuracy of the center offsets regression. Instance centers are estimated from offsets which are then used to generate final clusters. An iterative center de-duplication module is applied to associate points to an instance and suppress low confidence centers.

In order to better cluster large objects which only have points on their surface, far from the object's center, SCAN [45] voxelizes and processes a point cloud to produce features at different scales. SCAN then uses a novel attention mechanism to align these features. It also uses a sparse representation, and sparse convolutions, to maintain computational efficiency.

4D-PLS [43] efficiently performs panoptic segmentation and tracking by dividing 4D (temporal 3D) point clouds into overlapping 4D volumes that are processed in parallel. For each point in the 4D volume, 4D-PLS predicts its proximity to its instance center which they define to be the mean point of all thing points in it. They then use learned embeddings to cluster the points in a single volume, followed by greedy association across volumes. 4D-StOP [63] uses KP-conv to extract rich features per point. Unlike 4D-PLS, which models instances as Gaussian probability distributions in 4D space-time, 4D-StOP generates instance proposals through a per-point voting-based mechanism. Finally, 4D-StOP aggregates these instance proposals using learned geometric features and DBSCAN.

Other methods

In this section, we discuss works that do not fall into the categories mentioned above.

PillarAffinity [52] uses a lightweight head and a novel

Table 1: Panoptic Quality of different methods on SemanticKITTI and panoptic nuScenes val and test set. We report panoptic quality (PQ), panoptic quality for things class (PQ^{th}) and panoptic quality for stuff class (PQ^{st}). Methods are grouped together and categorized. Methods of each category are also sorted in chronological order. Numbers in **bold** denote the best performance on a given metric (column).

Dataset	SemanticKITTI						Panoptic nuScenes				
Method[e]	Val			Test			Val			Test[c]	Attn.[d]
	PQ	PQ^{th}	PQ^{st}	PQ	PQ^{th}	PQ^{st}	PQ	PQ^{th}	PQ^{st}	PQ	
KITTI Panoptic$^{\phi}$ [25]	–	–	–	44.5	32.7	53.1	–	–	–	–	×
LidarMultiNet [28]	–	–	–	–	–	–	**81.1**	–	–	–	✓
PanopticTrackNet $^{\theta}$ [12]	40	29.9	47.4	–	–	–	–	–	–	–	×
EfficientLPS [13]	59.2	58	**60.9**	57.4	53.1	60.5	–	–	–	–	×
LPS ‡[42]	–	–	–	38	25.6	47.1	–	–	–	–	×
4DPLS [43]	–	–	50.3	–	–	–	–	–	–	–	×
SMAC-Seg [44]	–	–	–	56.1	53	58.4	–	–	–	–	✓
SCAN [45]	–	–	–	61.5	61.4	61.5	65.1	60.6	72.5	–	✓
DS-Net [46]	57.7	61.8	54.8	55.9	55.1	56.5	–	–	–	–	×
Panoptic PH-Net [47]	61.7	69.3	–	61.5	63.8	59.9	74.7	74	75.9	**80.1**	✓
PVCL [48]	–	–	–	59.1	59.8	58.6	64.9	59.2	67.6	–	×
CFNet [49]	62.7	70	57.3	–	–	–	–	–	–	–	×
Panoster $^{\wedge}$ [17]	55.6	56.6	–	52.7	49.4	55.1	–	–	–	–	×
MaskRange [19]	–	–	–	53.1	44.9	59.1	–	–	–	–	✓
MaskPLS-M [20]	–	–	–	–	–	–	58.2	55.7	60	–	✓
MaskPLS-C [20]	–	–	–	–	–	–	57.4	63.6	52.9	–	✓
PUPS [18]	**64.4**	**73**	58.1	62.2	65.7	59.6	74.7	75.4	73.6	–	✓
P3Former [22]	–	–	–	64.9	67.1	63.3	75.9	76.9	75.4	–	✓
4D-Former [23]	–	–	–	–	–	–	77.3	–	–	78	✓
DQFormer [24]	63.5	–	–	63.1	–	–	77.7	77.8	**77.5**	73.9	✓
Panoptic-PolarNet † [32]	59.1	65.7	54.3	54.1	53.3	54.8	–	–	–	–	×
EvLPSNet [34]	58	62.7	54.6	–	–	–	–	–	–	–	✓
P-RangeFormer [38]	–	–	–	64.2	63.6	64.6	–	–	–	–	✓
Eq-4D-StOP [36]	61.2	66.1	57.5	–	–	–	–	–	–	–	×
LCPS (Baseline)[a] [40]	55.7	–	–	–	–	–	72.9	72.8	73	72.8	✓
LCPS (Full) [40]	59	–	–	–	–	–	79.8	**82.3**	75.6	79.5	✓
UniSeg [41]	–	–	–	**67.2**	**67.5**	**67**	–	–	–	78.4	✓
Panoptic-PolarNet+DEM [37]	60.3	68.6	54.3	–	–	–	69.6	68.3	71.8	68.9	×
GP-S3Net $^{\vee}$ [50]	63.3	70.2	58.3	60	65	56.4	–	–	–	–	✓
CPSeg [51]	–	–	–	56.9	54.7	58.5	–	–	–	73.2	✓
(Cartesian) affinity [52]	–	–	–	–	–	–	76.7	78.5	73.6	–	×
(Polar) affinity [52]	–	–	–	–	–	-	77.9	80	74.3	–	×
PANet [53]	61.7	–	–	58.5	59.7	57.6	69.2	69.5	68.7	–	✓
SAL[b] [54]	24.8	17.4	30.2	–	–	–	38.4	47.5	29.2	–	✓
Cylinder3D * [30]	56.4	58.8	54.8	–	–	–	–	–	–	–	×
LRPS [55]	–	–	–	54.6	54	55.1	–	–	–	–	×
SoftGroup++ [56]	–	–	–	57.2	57.1	57.3	–	–	–	–	✓
AOP-Net [57]	–	–	–	–	–	–	–	–	–	68.3	×
TARL [58]	56.6	–	–	–	–	–	–	–	–	–	✓
LPST [59]	63.1	68.7	58.9	61	58.1	63.2	77.1	79.3	73.6	76.1	×
Cylinder3D+SLR $^{\diamond}$ [29]	–	–	–	56.0	51.8	59.1	–	–	–	–	×
Divide-and-Merge [31]	–	–	–	56.5	52.9	59.1	–	–	–	–	×

[a]Model without image features, features fusion module and foreground object selection module. [b]Note that for better call SAL, we report zero shot results. [c] For panoptic nuScenes test set, we only report PQ as that is most widely reported. [d] Use of attention as described in Attention Is All You Need [60] [e] Category of a method- †: Center offsets and heat map , ‡: Center offsets only, ϕ: Proposal from bounding box, θ: Two-stage, $\wedge$: Single-stage, $\diamond$: Non learnt heuristic clustering, $\vee$: Others, $*$: Not categorized in this paper but added to show metrics.

Caveats: Some works, such as EfficientLPS [13], report results on the original nuScenes dataset which was released without instance segmentation labels [10]. We exclude such results, and only report results on the panoptic nuScenes dataset because it is the official source for panoptic segmentation annotations. We also exclude results on autonomous driving datasets that are rarely used for panoptic segmentation. Moreover, although computationally efficiency is an important requirement for autonomous vehicles, we do not present methods' inference time latency because it is seldom reported and is hardware dependent which makes it hard to normalize across different papers.

decoding algorithm. The network outputs a flag signaling the start of a new instance, and a zigzag decoding is then used to group pillars to an instance.

CPSeg [51] performs instance segmentation on a range image. They predict semantic labels and an embedding vector per point. They also have a module that pillarizes the embedding vectors for foreground points and applies pair wise connectivity on these pillars to group and associate different pillars to an instance. These pillar instance IDs are then reprojected back onto the pointcloud.

PANet [53] also assigns instance IDs in two steps. First, PANet heuristically clusters the pointcloud by using shifting algorithms inspired by Mean Shift and DS-Net [46]. Second, PANet merges certain clusters if their affinity exceeds a predetermined threshold. PANet calculates these affinities by running a k-NN transformer on embeddings from each of the instances obtained in the first step.

Clustering using only class labels and points' positions suffers from oversegmentation. GP-S3Net [50] addresses this by first using HDBSCAN [64] to cluster a pointcloud. GP-S3Net then constructs a graph where each cluster is a node. Finally, a graph convolutional neural network processes the edges of this graph to decide whether two clusters should be merged.

Some methods leverage self-supervised learning as collecting panoptic segmentation annotations for pointclouds is tedious and expensive, and because it might help with detecting rare objects that are not present in the training set thereby improving safety for autonomous vehicles. Authors in [65] clusters non-ground points with HDBSCAN and then refine them by constructing a graph representation of each instance proposal region. Self-supervised features extracted from a pre-trained network are used to weigh the edges between the nodes. Finally, GraphCut [66] is used on each graph to refine its corresponding cluster. AutoInst [67] also constructs a graph from a pointcloud with points as nodes and connectivity as edges. The graph is used to generate instance proposals, which are then refined through a self-trained neural network.

Recently, some methods have shown how to use foundational models to perform panoptic segmenation. Authors in Ref. [54] use vision foundation models along with text prompts. Using foundational models like CLIP [68], SAM [69] trains their model on unlabeled text and image data while using off the shelf models to perform pseudo-labeling in a zero shot manner, showcasing the extensibilty of these large models.

Results

Table 1 present results along with caveats of different LiDAR-based panoptic segmentation methods that report panoptic quality on any of the validation and test sets of SemanticKITTI [25], and panoptic nuScenes [11]. This table will be analyzed in upcoming sections and trends will be highlighted.

Performance Trends

The superiority of data driven approaches is evident in their performance when compared to fully non-learned heuristic clustering methods. The best heuristic method, Ref. [31] attains a PQ of 56.5 on the SemanticKITTI test set, whereas data-driven methods attain a PQ of up to 67.2 (about 19 % better).

Methods that leverage center offsets are popular, and we observe that they usually perform well and are close to the best model, if not the best in most cases (such as Panoptic PH-Net [47], CFNet [49], SCAN [45]). Analyzing methods that leverage center heatmap in addition to offsets, Eq-4D-StOP [36], Uniseg [41], LCPS [40] lead the pack in PQ.

The best performing methods in table 1 use attention as described in Ref. [60]. Attention can boost the performance of a model by making it easier to focus on the most critical features of an object and to capture complex geometric relationships in a point cloud. This performance gain can come at the cost of higher compute and memory requirements. Methods like flash attention [70] can alleviate the memory cost of the attention module. The best performing methods notably also leverage sensor fusion, pointing to the benefit of multi-modal inputs. Despite compute being a limiting factor in the BEV pointcloud representation, most methods still use it owing to the orthographic geometry of this representation which leads to better box localization.

Conclusion

In this survey, we provided a comprehensive review of panoptic segmentation methods for LiDAR point clouds collected by autonomous vehicles. We summarized the methods, and categorized them based on whether they use non-learned heuristic clustering techniques at any point in their architecture. We also described the most common datasets and metrics currently in use. We reported performance of different methods on these datasets, and extracted performance trends.

We hope that our guide helps practitioners choose the most suitable architecture for their specific requirements, and helps researchers identify gaps in the field to inspire new innovations.

References

[1] T. Ashok Elluswamy, "Occupancy network," in *Tesla AI day*, 2022.

[2] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4603–4611.

[3] O. Elharrouss *et al.*, "Panoptic segmentation: A review," 2021. [Online]. Available: https://arxiv.org/abs/2111.10250

[4] X. Li and D. Chen, "A survey on deep learning-based panoptic segmentation," *Digital Signal Processing*, vol. 120, p. 103283, 2022.

[5] P. Nagaraju and S. Sudha, "A novel survey on panoptic segmentation using deep learning approaches," in *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 2023, pp. 1179–1184.

[6] Y. Li and L. Xu, "Panoptic perception for autonomous driving: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2408.15388

[7] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2443–2451.

[8] J. P. Bos *et al.*, "Autonomy at the end of the Earth: an inclement weather autonomous driving data set," in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2020*, vol. 11415. International Society for Optics and Photonics, 2020, pp. 36 – 48.

[9] J. Behley *et al.*, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9296–9306.

[10] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[11] W. K. Fong *et al.*, "Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3795–3802, 2022.

[12] J. V. Hurtado, R. Mohan, and A. Valada, "Mopt: Multi-object panoptic tracking," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Scalability in Autonomous Driving*, 2020.

[13] K. Sirohi *et al.*, "Efficientlps: Efficient lidar panoptic segmentation," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1894–1914, 2022.

[14] R. Mohan and A. Valada, "Efficientps: Efficient panoptic segmentation," *International Journal of Computer Vision*, vol. 129, pp. 1551 – 1579, 2020.

[15] K. He *et al.*, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[16] B. Cheng *et al.*, "Per-pixel classification is not all you need for semantic segmentation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[17] S. Gasperini *et al.*, "Panoster: End-to-end panoptic segmentation of lidar point clouds," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, p. 3216–3223, Apr. 2021.

[18] S. Su *et al.*, "Pups: point cloud unified panoptic segmentation," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, 2023.

[19] Y. Gu *et al.*, "Maskrange: A mask-classification model for range-view based lidar segmentation," 2022. [Online]. Available: https://arxiv.org/abs/2206.12073

[20] R. Marcuzzi *et al.*, "Mask-based panoptic lidar segmentation for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1141–1148, 2023.

[21] M. Rodrigo *et al.*, "Mask4d: End-to-end mask-based 4d panoptic segmentation for lidar sequences," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, 2023.

[22] Z. Xiao *et al.*, "Position-guided point cloud panoptic segmentation transformer," 2023. [Online]. Available: https://arxiv.org/abs/2303.13509

[23] A. Athar *et al.*, "4d-former: Multimodal 4d panoptic segmentation," in *Proceedings of the 2023 Conference on Robot Learning*, 2023.

[24] Y. Yang *et al.*, "Dqformer: Towards unified lidar panoptic segmentation with decoupled queries," 2024. [Online]. Available: https://arxiv.org/abs/2408.15813

[25] J. Behley, A. Milioto, and C. Stachniss, "A benchmark for lidar-based panoptic segmentation based on kitti," in *A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI*, 05 2021, pp. 13 596–13 603.

[26] Q. Chen, S. Vora, and O. Beijbom, "Polarstream: Streaming object detection and segmentation with polar pillars," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[27] Y. Zhong, M. Zhu, and H. Peng, "Vin: Voxel-based implicit network for joint 3d object detection and segmentation for lidars," in *British Machine Vision Conference*, 2021.

[28] D. Ye *et al.*, "Lidarmultinet: towards a unified multi-task network for lidar perception," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, 2023.

[29] Y. Zhao *et al.*, "A technical survey and evaluation of traditional point cloud clustering methods for lidar panoptic segmentation," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[30] X. Zhu *et al.*, "Cylindrical and asymmetrical 3d convolution networks for lidar-based perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6807–6822, 2022.

[31] Y. Zhao, X. Zhang, and X. Huang, "A divide-and-merge point cloud clustering algorithm for lidar panoptic segmentation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7029–7035.

[32] Z. Zhou, Y. Zhang, and H. Foroosh, "Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 189–13 198.

[33] B. Cheng *et al.*, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *CVPR*, 2020.

[34] K. Sirohi *et al.*, "Uncertainty-aware lidar panoptic segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8277–8283.

[35] T. Hugues *et al.*, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] M. Zhu *et al.*, "4d panoptic segmentation as invariant and equivariant field prediction," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 22 431–22 441.

[37] Y. Wang *et al.*, "Panoptic segmentation of 3d point clouds with gaussian mixture model in outdoor scenes," *Visual Intelligence*, vol. 2, no. 1, p. 10, 2024.

[38] L. Kong *et al.*, "Rethinking range view representation for lidar segmentation," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 228–240.

[39] Q. Meng *et al.*, "Small, versatile and mighty: A range-view perception framework," 2024. [Online]. Available: https://arxiv.org/abs/2403.00325

[40] Z. Zhang *et al.*, "Lidar-camera panoptic segmentation via geometry-consistent and semantic-aware alignment," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 3639–3648.

[41] Y. Liu *et al.*, "Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 21 605–21 616.

[42] A. Milioto *et al.*, "Lidar panoptic segmentation for autonomous driving," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8505–8512.

[43] M. Aygun *et al.*, "4d panoptic lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.
[44] E. Li *et al.*, "Smac-seg: Lidar panoptic segmentation via sparse multi-directional attention clustering," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9207–9213.
[45] S. Xu *et al.*, "Sparse cross-scale attention network for efficient lidar panoptic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022.
[46] F. Hong *et al.*, "Lidar-based panoptic segmentation via dynamic shifting network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 090–13 099.
[47] J. Li *et al.*, "Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[48] M. Liu *et al.*, "Prototype-voxel contrastive learning for lidar point cloud panoptic segmentation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022.
[49] X. Li *et al.*, "Center focusing network for real-time lidar panoptic segmentation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 425–13 434.
[50] R. Razani *et al.*, "Gp-s3net: Graph-based panoptic sparse semantic segmentation network," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 16 056–16 065.
[51] Y. X. Enxu Li, Ryan Razan and B. Liu, "Cpseg: Cluster-free panoptic segmentation of 3d lidar point clouds," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8239–8245, 2021.
[52] Q. Chen and S. Vora, "Proposal-free lidar panoptic segmentation with pillar-level affinity," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 4528–4535.
[53] J. Mei *et al.*, "Panet: Lidar panoptic segmentation with sparse instance proposal and aggregation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7726–7733.
[54] A. Ošep *et al.*, "Better call sal: Towards learning to segment anything in lidar," in *European Conference on Computer Vision (ECCV)*, 2024.
[55] W. Wang, X. You, J. Yang, M. Su, L. Zhang, Z. Yang, and Y. Kuang, "Lidar-based real-time panoptic segmentation via spatiotemporal sequential data fusion," *Remote Sensing*, vol. 14, no. 8, p. 1775, 2022.
[56] T. Vu *et al.*, "Scalable softgroup for 3d instance segmentation on point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 1981–1995, 2024.
[57] Y. Xu *et al.*, "Aop-net: All-in-one perception network for lidar-based joint 3d object detection and panoptic segmentation," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–7.
[58] L. Nunes *et al.*, "Temporal consistent 3d lidar representation learning for semantic perception in autonomous driving," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
[59] A. Agarwalla *et al.*, "Lidar panoptic segmentation and tracking without bells and whistles," *IROS*, 2023.
[60] A. Vaswani *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.
[61] F. Zhang *et al.*, "Instance segmentation of lidar point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9448–9455.
[62] R. Marcuzzi *et al.*, "Contrastive instance association for 4d panoptic segmentation using sequences of 3d lidar scans," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1550–1557, 2022.
[63] L. Kreuzberg *et al.*, "4d-stop: Panoptic segmentation of 4d lidar using spatio-temporal object proposal generation and aggregation," in *European Conference on Computer Vision Workshop*, 2022.
[64] R. Campello *et al.*, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–172.
[65] L. Nunes *et al.*, "Unsupervised class-agnostic instance segmentation of 3d lidar data for autonomous vehicles," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 8713–8720, 2022.
[66] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International Journal of Computer Vision - IJCV*, vol. 70, pp. 109–131, 11 2006.
[67] C. Perauer *et al.*, "Autoinst: Automatic instance-based segmentation of lidar 3d scans." *CoRR*, vol. abs/2403.16318, 2024.
[68] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021.
[69] A. Kirillov *et al.*, "Segment anything," in *Int. Conf. Comput. Vis.*, 2023.
[70] T. Dao *et al.*, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Author Biography

Aditya Dusi received his MS in Electrical Engineering from Stanford University. Since then, he has been working as a research engineer in perception for autonomous driving.

Bassam Helou received his PhD in Applied Physics from Caltech. Since then, he has worked on a variety of autonomous driving ML projects.