Supplementary Material: 4D Panoptic LiDAR Segmentation

Mehmet Aygün[1*] Aljoša Ošep[1*] Mark Weber[1] Maxim Maximov[1]
Cyrill Stachniss[2] Jens Behley[2] Laura Leal-Taixé[1]
[1]Technical University of Munich, Germany [2]University of Bonn, Germany
[1]{mehmet.ayguen, aljosa.osep, leal.taixe, mark-cs.weber, maxim.maximov}@tum.de
[2]{firstname.lastname}@igg.uni-bonn.de

A. Implementation Details

In this section, we (i) provide details about the four different point propagation strategies we experimented with for forming a 4D point clouds and (ii) we detail the point overlap based association procedure we use to link 4D object instances across overlapping point clouds.

A.1. 4D Point Cloud Formation

Our method works on directly 4D volumes which constructed using consecutive lidar scans. However, due to memory constraints stacking all points is not feasible. To reduce memory usage, when we process the scan f_i together with previous scans $f_{i-\tau}$,..., f_{i-1}, we take all of the points from f_i and sub-sample points from other scans. Moreover, since we already processed previous scans $f_{i-\tau}$,..., f_{i-1} before, we know the semantic class and objectness scores of all points at time step f for that scans. We use three different strategy to sub-sample point from previous scans by leveraging these information.

Thing Propagation: In this strategy, we only sample points from previous scans if the points are assigned to a thing class. If the total number of points are exceeded the gpu memory limit, we randomly sub-sample again.

Importance Sampling: We select 10% of points from a previous scans using the objectness score predicted by the network in the previous time steps. Thus, points with higher objectness scores have a higher chance to be used in the clustering process in the following scans.

Temporal Decay: In this strategy, we use importance sampling using objectness scores again. However, instead of sampling 10% of points from each past scan, we select the percentage of points based on temporal proximity of scans. Given a temporal window size of τ, we select the number of points n_i as:

$$n_i = \frac{e^i}{\sum_{n=1}^{\tau-1} e^i}, \quad i = 1, 2, 3, \ldots, \tau - 1, \tag{1}$$

where $n_{\tau-1}$ is the closest scan to the current scan. In this strategy more points would be sampled from scans which are temporally close.

Temporal Stride: We used importance sampling in this strategy, but instead of using points from previous scans, *i.e.*, $i = 1, 2, 3, \ldots, \tau - 1$, we used every second scan, *i.e.*, $i = 1, 3, 5, \ldots, \tau - 1$. For the points from the remaining scans, we assigned predictions by looking at the closest points, which had class and instance prediction.

A.2. Clustering

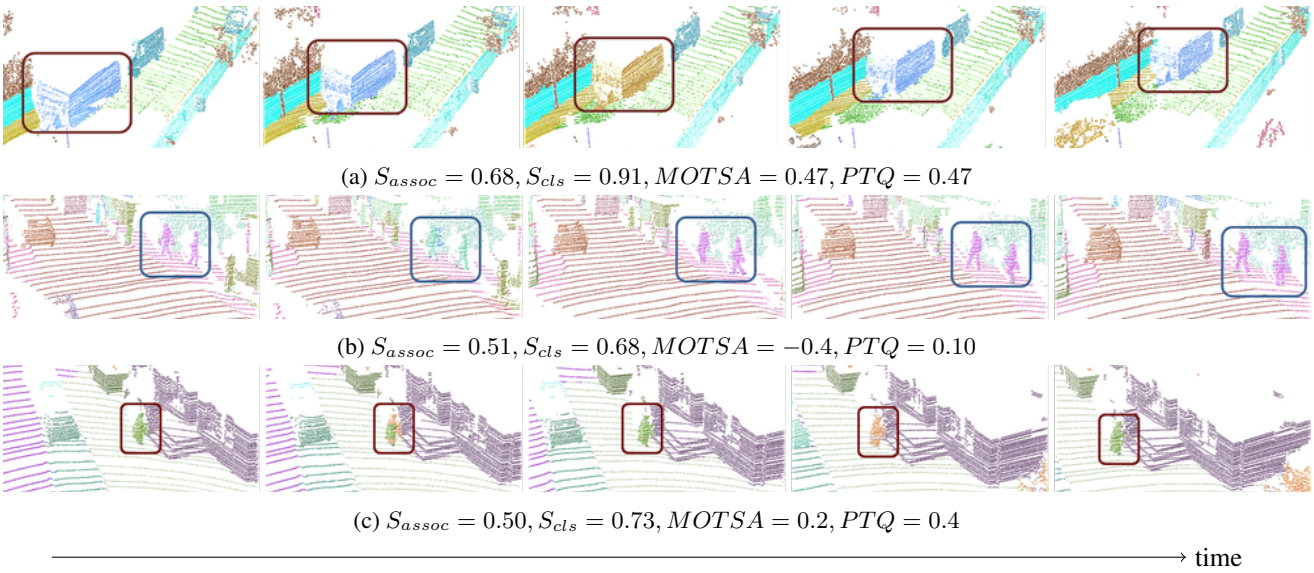
Our method can cluster points with different semantics and does not provide a single class label for a specific instance. This can be adapted depending on the requirements of the downstream application (*e.g.*, via majority vote). Moreover, if the number of points that assigned to a specific cluster is lower than a threshold, we eliminate that instance from the final prediction.

A.3. Tracking

As discussed in the main paper (Section 3), we process multiple scans together in an overlapping fashion. For a window size of τ, at time t, we process scans $f_{i-\tau}^t, \ldots, f_i^t$ together by overlapping them in a 4D point cloud. f_i^t represent the scan i which processed at time step t.

To associate instances at time t and $t+1$, we look at instance intersections in scans which are common in both time steps. For instance, with temporal window size of two, we would process scans f_1^1 and f_2^1, next we would process f_2^2 and f_3^2 together. To transfer ids from the previous time to the current scan (f_3^2), we would look the instance intersections in scans which processed on both time step (f_2^1 and f_2^2). Since the instance ids are same for the scans which processed together (f_2^2 and f_3^2), the association would be finished between overlapping 4D volumes.

For the intersection, we consider all common scans. When there is a conflict (i.e, one instance has overlap with two instance in the next step), we pick the instance pair which have higher intersection-over-union. If any of the



(a) $S_{assoc} = 0.68, S_{cls} = 0.91, MOTSA = 0.47, PTQ = 0.47$

(b) $S_{assoc} = 0.51, S_{cls} = 0.68, MOTSA = -0.4, PTQ = 0.10$

(c) $S_{assoc} = 0.50, S_{cls} = 0.73, MOTSA = 0.2, PTQ = 0.4$

Figure 1: Comparison of evaluation metrics for some failure cases. Respective instances which we calculate the metrics are depicted with bounding boxes. In (a) ID recovery is punished by MOTSA and PTQ. In (b) two instances predicted as single instance and in (c) ID switch happened and in the second scan the instance is not segmented correctly.

# Scans	*LSTQ*	S_{assoc}	S_{cls}	IoU^{St}	IoU^{Th}
1	51.92	45.16	59.69	64.60	60.40
2	59.86	58.79	**60.95**	64.96	**63.06**
3	61.74	62.65	60.85	65.16	62.53
4	**62**.74	**65.11**	60.46	**65.36**	61.26
6	61.52	64.28	58.88	65.32	57.38
8	59.09	62.30	57.68	65.23	54.52

Table 1: Panoptic Tracking on SemanticKITTI valid. set.

intersections do not surpass IoU of 0.5, we create a new ID for the instance.

B. Additional Results

B.1. Ablation on the Temporal Window Size

In Tab. 1, we highlight the performance of our method for temporal window size $\tau = 1, 2, 3, 4, 6, 8$. As can be seen, the association accuracy is increasing up to $\tau = 4$ and then saturates, while classification accuracy saturates a $\tau = 2$; however, it only decreases marginally.

B.2. Per-class Evaluation

In this section, we analyze the performance on the validation split (Tab. **??**) through the lens of several evaluation metrics and analyze per-class performance in Tab. 2 (this table extends Tab. **??** from the main paper). While our 4-scan variant performs better than the 2-scan variant in terms of *LSTQ*, we observe a significant drop in the MOTSA score. Our analysis shows that this is because we obtain negative MOTSA scores on some classes due to a decrease in precision while having fewer ID switches. This unintuitive behavior of MOTSA can be further validated when analysing performance for class, *e.g.*, *other-vehicle*. For this class the IDS reduces ($162 \rightarrow 99$), the precision drops ($0.68 \rightarrow 0.47$), while recall improves from ($0.36 \rightarrow 0.47$). In our metric, this is reflected in the decrease of S_{cls} ($0.56 \rightarrow 0.55$) and increase in S_{assoc} ($0.17 \rightarrow 0.38$) while MOTSA unintuitively drops from 0.12 to -0.1, even though association capabilities improve.

We visualize such cases in Fig. 1. As can be seen, the difference is due to the semantic interpretation of the points and not due to the segmentation and tracking quality at the instance level. This confirms the nonintuitive behavior of MOTSA, while our metric provides insights on both semantic interpretation and instance segmentation and tracking. As shown in Figure 1a-1c, our method successfully recovers the ID of the instance. This behavior is penalized by both MOTSA and PTQ, but not by the association score of our metric S_{assoc}. Moreover, while the instances tracked reasonably well in Figure 1b, MOTSA and PTQ scores decrease substantially due to poor segmentation of the instances.

Finally, we acknowledge that our method works very well on the most frequently occurring object classes (*car*), however, segmenting and tracking objects that appear in the long tail of the object class distribution remains challenging.

Category	# Scans	# Instances	% Instances	TP	FP	FN	IDS	Prec.	Recall	MOTSA	S_{assoc}	S_{cls}
Car	2	29255	0.80	27553	687	1702	1204	0.98	0.94	0.88	0.72	0.96
	4			27401	845	1854	720	0.97	0.94	0.88	0.77	0.96
Truck	2	1253	0.03	447	226	806	90	0.66	0.36	0.10	0.15	0.38
	4			496	331	757	52	0.60	0.40	0.09	0.20	0.39
Bicycle	2	792	0.02	435	132	357	64	0.77	0.55	0.30	0.36	0.72
	4			574	230	218	43	0.71	0.72	0.38	0.59	0.71
Motorcycle	2	255	0.01	209	151	46	31	0.58	0.82	0.11	0.56	0.88
	4			231	747	24	9	0.24	0.91	-2.06	0.81	0.74
Other-vehicle	2	2138	0.06	778	362	1360	162	0.68	0.36	0.12	0.17	0.56
	4			1022	1131	1116	99	0.47	0.48	-0.10	0.38	0.55
Person	2	1975	0.05	1183	282	792	203	0.81	0.60	0.35	0.31	0.65
	4			1180	346	795	143	0.77	0.60	0.35	0.35	0.63
Bicyclist	2	816	0.02	720	39	96	33	0.95	0.88	0.79	0.63	0.89
	4			750	39	66	28	0.95	0.92	0.84	0.69	0.91
Motorcyclist	2	78	0.01	0	0	78	0	0.00	0.00	0.00	0.10	0.00
	4			0	0	78	0	0.00	0.00	0.00	0.16	0.00

Table 2: Per-class tracking evaluation on Semantic-KITTI validation set (2 and 4 scan versions).