Supplementary Material: Category-Level Pose Retrieval with Contrastive Features Learnt with Occlusion Augmentation

Georgios Kouros[1]
georgios.kouros@esat.kuleuven.be
Shubham Shrivastava[2]
sshriva5@ford.com
Cédric Picron[1]
cedric.picron@esat.kuleuven.be
Sushruth Nagesh[2]
snagesh1@ford.com
Punarjay Chakravarty[2]
pchakra5@ford.com
Tinne Tuytelaars[1]
tinne.tuytelaars@esat.kuleuven.be

[1] PSI, ESAT
KU Leuven
Belgium

[2] Ford Greenfield Labs
Palo Alto, USA

Abstract

In this supplementary material, we present some additional background information about our work to give more insight into the data and performance of our approach. First, we present an overview of the data in PASCAL3D, OccludedPASCAL3D, and KITTI3D. Second, we evaluate the performance of our approach with different rendering types. Third, we present a more detailed view of the experiments on PASCAL3D and OccludedPASCAL3D and how our approach performed on each object category and for each occlusion level. Then we showcase three different reference set designs that were examined and evaluated to determine the optimal choice with regard to performance and inference speed. Afterwards, we provide a more detailed analysis on occlusion and bounding box augmentations, while finally, we include some qualitative results in the form of image retrievals and failure cases.

1 Data Overview

In this section we present an overview of the data in PASCAL3D and OccludedPASCAL3D starting with Table 1 that contains the number of samples and number of CAD models per object category and per occlusion category of PASCAL3D and OccludedPASCAL3D. Then we present some example images from all object categories and all occlusion categories in

Samples		aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Total
Train		986	661	1099	745	548	2763	526	1160	624	642	662	629	11068
Test	L0	969	645	1059	747	532	2712	507	1153	596	624	646	622	10812
	L1	951	634	921	732	527	2646	496	1145	578	542	633	616	10421
	L2	937	619	905	725	525	2612	483	1107	570	521	613	602	10219
	L3	903	601	887	720	521	2573	472	1075	555	507	583	586	9983
CADs		8	6	6	8	6	10	10	6	5	6	4	4	79

Table 1: Number of samples and number of CAD models per object category of test sets L0-L3 from PASCAL3D and OccludedPASCAL3D

Split	Frames	Object Instances by Occlusion Level			
		FullyVis.	PartlyOcc.	MostlyOcc.	Total
Train	3712	3036	1861	1541	6438
Val	3769	2895	1950	2035	6880
Test	7518	-	-	-	-

Table 2: Distribution of KITTI3D car instances (w,h > 40 px)

Figure 1, as well as renderings for all CAD models in Figure 2. Furthermore, we include a similar overview of KITTI3D in Table 2 and Figure 3.

2 Evaluation of Rendering Types

We evaluated five different rendering types, namely, RGB renderings, silhouettes, depth, normals, and triplets (concatenations of RGB, depth, and normals). According to the results in Table 3, we observe that surface normals are more robust to higher levels of occlusions, although, our approach achieves satisfactory performance in all cases. Overall, by observing the standard deviation, all renderings seem to perform similarly in *L*0, but as the difficulty increases towards *L*3 the performance varies more widely making the superiority of surface normal maps more easily observed.

3 Results per Object Category

In Tables 4-7, we present the performance of our approach per object category against the competing methods, StarMap and NeMo, on PASCAL3D (L0) and OccludedPASCAL3D (L1-L3). To compute the average across all object categories we use a weighted average, where the weight of each category is its number of samples divided by the total number of samples, as shown in Table 1. As observed in the aforementioned tables, we outperform the state-of-the-art methods in the vast majority of object categories and occlusion levels. In Table 8, we demonstrate the consistent performance of our approach across five models, trained from scratch on PASCAL3D with surface normal renderings. The standard deviation is less than 3% in all metrics and all five models outperform the state-of-the-art.

	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
Rendering	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
RGB	**99.3**	**97.7**	90.1	67.7	95.9	86.6	64.1	30.4	**3.0**	4.5	7.5	17.1
Silhouette	99.2	97.1	89.0	62.0	**96.3**	86.7	65.5	30.0	3.2	4.4	7.1	18.4
Depth	99.1	97.4	90.0	64.9	96.0	87.0	62.2	25.1	**3.0**	4.4	7.1	18.4
Normals	99.2	97.4	**91.5**	**69.6**	95.9	**88.3**	**68.8**	**32.6**	3.1	**4.2**	**6.7**	**16.1**
Triplet	99.2	97.2	90.8	67.9	96.0	86.8	64.3	27.9	3.1	4.3	7.5	18.0

Table 3: Comparison of rendering types on cars of PASCAL3D L0-L3

4 Evaluation of Reference Set Designs

To decide what is the best reference set design, we trained and evaluated on three distinct reference sets, namely TrainDB, CoarseDB, and FineDB. TrainDB contains the renderings that were generated based on the poses in the traininsg set, while CoarseDB and FineDB were generated using a discretization of the viewing sphere as shown in Table 10. To train more efficiently we sample renderings from CoarseDB and FineDB for every object instance in a batch. To run inference, we examine all possible combinations to find the optimal setting, as presented in Table 9. Out of the three designed reference sets, we found TrainDB to achieve the highest performance thanks to it being more representative of the data while also being the fastest, since it contains the least amount of renderings. Furthermore, we attribute the performance drop of CoarseDB and FineDB to our sampling scheme that does not utilize all the samples, but only the ones close to the training data. Therefore, the models are not trained on all available renderings in the database, so they are not able to project them correctly to the embedding space, thus resulting in the lower performance, seen in Table 9.

5 Occlusion Augmentation Results

In Figure 4, we provide graphs of all three evaluation metrics, not just $ACC_{\frac{\pi}{6}}$, for the experiments on our occlusion augmentation scheme. All three metrics follow the same trend indicating that the higher the occlusion scale s_{occ} during training, the higher the robustness across the increasingly occluded sets L0-L3. However, since the occlusion augmentation scheme is not sophisticated enough to avoid fully occluding the object of interest, using high a occlusion scale can actually make training harder and require many more epochs to converge. Due to this effect, we observe a slight drop in performance on L0, which could be alleviated by increasing the number of training epochs or by progressively decreasing the occlusion scale.

6 Inference Speed

Our approach can run inference on an object instance in approximately 30ms. Out of those, roughly 60% are spent on embedding the object instance and the rest 40% are spent on calculating the distances and finding the nearest neighbour. Further increasing the inference speed can be accomplished by training a smaller backbone (e.g. ResNet18), using an embedding size lower than 512, removing very similar poses from TrainDB, or using renderings of only one CAD model. In case of a larger database, in particular, it would also be beneficial to em-

		aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
$ACC_{\frac{\pi}{6}}$	Res50-A	83.0	79.6	73.1	87.9	96.8	95.5	91.1	82.0	80.7	97.0	94.9	83.3	88.1
	Res50-S	79.5	75.8	73.5	90.3	93.5	95.6	89.1	82.4	79.7	96.3	96.0	84.6	87.6
	StarMap	85.5	84.4	65.0	**93.0**	98.0	97.8	94.4	82.7	85.3	**97.5**	93.8	89.4	88.1
	NeMo	73.3	66.4	65.5	83.0	87.4	98.8	82.8	81.9	74.6	94.7	87.0	85.5	84.1
	NeMo-M	76.9	82.2	66.5	87.1	93.0	98.0	90.1	80.5	81.8	96.0	89.3	87.1	86.7
	NeMo-S	82.2	78.4	68.1	88.0	91.7	98.2	87.0	76.9	85.0	95.0	83.0	82.2	86.1
	PoseCon.	83.7	84.0	**82.5**	88.9	97.7	96.7	95.3	86.9	**87.2**	97.1	96.7	87.8	90.8
	Ours	**84.4**	**88.1**	**82.5**	91.7	**98.7**	**99.2**	**95.9**	**88.8**	85.6	97.0	**98.0**	**90.0**	**92.3**
$ACC_{\frac{\pi}{18}}$	Res50-A	31.3	25.7	23.9	35.9	67.2	63.5	37.0	40.2	18.9	62.5	51.2	24.9	44.6
	Res50-S	29.1	22.9	25.3	39.0	62.7	62.9	37.5	42.0	19.5	57.5	50.2	25.4	43.9
	StarMap	49.8	34.2	25.4	56.8	90.3	81.9	67.1	57.5	27.7	**70.3**	69.7	40.0	59.5
	NeMo	39.0	31.3	29.6	38.6	83.1	94.8	46.9	58.1	29.3	61.1	71.1	**66.4**	60.4
	NeMo-M	43.1	35.3	36.4	48.6	89.7	95.5	49.5	56.5	33.8	68.8	75.9	56.8	63.2
	NeMo-S	49.7	29.5	37.7	49.3	89.3	94.7	49.5	52.9	29.0	58.5	70.1	42.4	61.1
	PoseCon.	53.3	40.0	50.0	56.1	93.2	88.6	67.3	71.7	**37.7**	64.7	82.5	50.2	67.2
	Ours	**59.5**	**42.8**	**54.2**	**68.7**	**94.5**	**95.9**	**70.4**	**71.8**	33.9	69.9	**88.7**	58.7	**72.2**
MedErr	Res50-A	13.3	15.9	15.6	12.1	8.9	8.8	11.5	11.4	16.6	8.7	9.9	15.8	11.7
	Res50-S	14.2	17.3	15.4	11.7	9.0	8.8	12.0	11.0	17.1	9.2	10.0	14.9	11.8
	StarMap	10.0	14.0	19.7	8.8	3.2	4.2	6.9	8.5	14.5	6.8	6.7	12.1	9.0
	NeMo	13.8	17.5	18.3	12.8	3.4	2.7	10.7	8.2	16.1	8.0	5.6	**6.6**	9.3
	NeMo-M	11.8	13.4	14.8	10.2	**2.6**	**2.8**	10.1	8.8	14.0	7.0	5.0	8.1	8.2
	NeMo-S	10.1	16.3	14.9	10.2	3.2	3.2	10.1	9.3	14.1	8.6	5.4	12.2	8.8
	PoseCon.	9.3	12.0	10.0	8.8	3.1	3.5	7.1	**6.0**	**12.2**	7.8	4.8	9.9	7.1
	Ours	**8.2**	**11.6**	**9.4**	**7.1**	3.0	3.1	**6.7**	6.3	13.5	**6.5**	**3.9**	8.4	**6.6**

Table 4: Comparison with competing methods on unoccluded PASCAL3D (L0) per category

ploy a kd-tree to speed up the nearest neighbour search. This delay can further be reduced or even eliminated by incorporating a regression head and a regression loss term.

7 Bounding Box Augmentation Results

In Figure 5, we provide graphs of all three evaluation metrics for the experiments on our bounding box augmentation scheme. The three metrics follow a similar trend which denotes that training with higher bounding box noise results in more robustness to test time noise. However, training with too high β_{train} can result in a performance drop in cases of minimal test time noise, which could be potentially alleviated by training for more epochs. Nevertheless, training with $\beta_{train} \in [0.1, 0.25]$ seems a good trade-off between robustness to noise and accuracy in absence of noise. Better cross-dataset performance can be achieved by training with higher bounding box noise which leads to better generalization to cases without perfect center/scale alignment. For example training with $\beta_{train} = 0.75$ on PASCAL3D lead to better cross-dataset performance on KITTI3D.

		aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
$ACC_{\frac{\pi}{6}}$	Res50-A	57.3	56.8	51.4	78.3	82.5	80.0	62.3	63.1	61.1	84.9	87.8	69.8	70.4
	Res50-S	54.0	59.5	48.9	84.4	86.1	84.4	67.1	64.9	65.9	87.8	92.4	74.5	73.2
	StarMap	52.6	65.3	42.0	81.8	**87.9**	86.1	64.5	66.5	62.8	76.9	85.2	59.7	71.1
	NeMo	49.0	51.4	52.9	73.5	82.2	94.3	70.2	67.9	53.8	86.7	75.0	79.4	73.1
	NeMo-M	58.1	68.8	53.4	78.8	86.9	94.0	76.0	70.0	61.8	87.3	82.8	82.8	77.2
	NeMo-S	61.9	63.4	52.9	81.3	84.8	92.7	78.4	68.2	68.9	87.1	80.3	76.9	76.0
	PoseCon	57.7	66.6	56.9	**86.7**	87.1	83.6	66.9	74.2	72.3	90.6	89.4	78.2	76.2
	Ours	**71.4**	**79.2**	**70.6**	85.2	87.7	**97.4**	**87.2**	**81.9**	**78.4**	**94.1**	**96.5**	**80.0**	**85.7**
$ACC_{\frac{\pi}{18}}$	Res50-A	11.8	12.5	12.3	26.5	45.0	40.7	14.7	22.3	10.7	24.4	34.9	13.0	25.3
	Res50-S	12.4	10.7	13.8	30.2	46.9	44.8	21.2	24.0	10.4	28.0	40.6	17.9	28.1
	StarMap	15.6	15.1	10.8	36.2	66.6	58.1	26.6	32.0	14.4	23.8	47.4	13.0	34.4
	NeMo	18.5	19.9	19.1	24.0	72.1	82.0	25.8	35.7	12.6	44.3	54.0	49.0	45.1
	NeMo-M	25.4	23.3	22.9	36.7	**86.9**	84.8	33.1	36.8	20.8	46.5	61.0	**46.3**	49.9
	NeMo-S	29.3	18.0	24.3	41.5	76.1	80.5	27.2	31.4	19.4	39.9	55.1	32.0	46.3
	PoseCon.	23.4	24.9	27.7	48.8	73.1	67.4	30.8	48.5	21.8	45.2	65.7	29.2	46.4
	Ours	**35.1**	**25.9**	**34.7**	**51.8**	74.4	**88.3**	**44.4**	**53.1**	**23.9**	**59.0**	**81.4**	29.9	**56.7**
MedErr	Res50-A	25.3	24.5	29.0	14.9	10.6	11.2	22.4	18.1	23.3	15.5	11.7	21.1	17.9
	Res50-S	26.8	23.7	31.0	13.8	10.5	10.6	18.2	16.7	21.8	13.6	10.9	19.3	17.3
	StarMap	27.3	22.1	38.9	12.9	7.0	8.2	19.1	17.2	21.7	16.8	10.6	24.1	17.6
	NeMo	30.8	29.0	27.3	17.6	5.9	5.1	18.6	14.7	27.4	11.3	8.8	**10.2**	15.6
	NeMo-M	22.6	18.6	25.8	14.1	**4.7**	4.6	15.1	13.8	21.2	11.0	8.0	11.3	13.0
	NeMo-S	18.9	23.2	26.7	12.6	5.2	5.4	15.6	15.4	20.1	12.1	8.6	15.3	13.6
	PoseCon.	22.0	18.0	21.6	10.2	6.0	6.6	16.8	10.4	18.1	10.9	7.2	16.7	12.6
	Ours	**14.3**	**15.0**	**15.2**	**9.7**	4.8	**4.2**	**11.3**	**9.3**	**17.1**	**8.5**	**5.0**	14.8	**9.7**

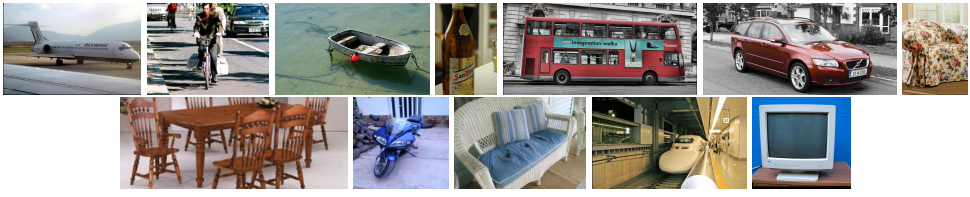
Table 5: Comparison with competing methods on OccludedPASCAL3D L1 per category

8 Qualitative Results

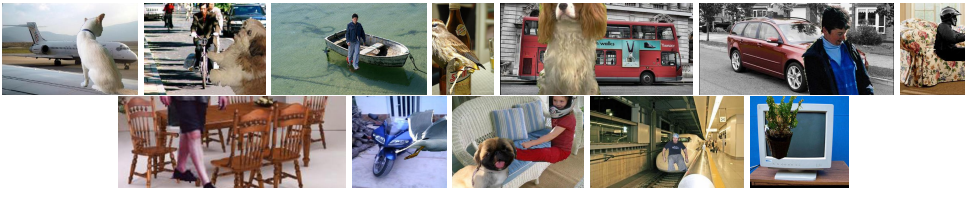
In this section, we present some additional qualitative results. In particular, Figures 6 and 7 present successful pose retrievals and failures cases for PASCSAL3D (L0) and OccludedPASCAL3D (L1-L3), while Figures 8 and 9 present similar cases for KITTI3D in all its occlusion levels, namely fully-visible, partly-occluded, and fully-occluded. We observe that the model has learnt to disregard the specific type of object type and focus more on its pose. In addition, thanks to the occlusion augmentation scheme, the model does not need to see the whole object to estimate its pose, but instead, it can estimate it adequately well even when seeing a small part of the object. On the other hand, observing the failure cases reveals that the model struggles with highly atypical cars, vastly different unseen poses, distinguishing between opposite directions, and large or same-category occlusions.

		aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
$ACC_{\frac{\pi}{6}}$	Res50-A	33.3	40.2	33.6	70.6	69.5	57.0	41.8	47.4	43.3	66.8	80.4	58.1	52.8
	Res50-S	36.3	44.9	36.1	76.1	73.1	65.5	53.2	49.5	45.4	72.7	88.3	65.0	58.4
	StarMap	28.5	38.9	21.3	65.0	61.7	59.3	37.5	44.7	43.2	55.1	56.4	36.2	47.2
	NeMo	38.2	41.2	39.6	58.3	72.6	84.7	50.7	51.1	34.9	70.1	60.0	64.6	59.9
	NeMo-M	43.1	**55.7**	43.3	69.1	**79.8**	84.5	58.8	58.4	43.9	76.4	64.3	**70.3**	65.2
	NeMo-S	43.4	49.6	43.6	76.0	71.2	83.8	61.9	55.9	50.9	78.3	63.1	68.6	63.9
	PoseCon.	38.5	51.2	39.2	**81.8**	69.5	61.8	49.3	57.6	56.1	74.1	82.4	61.0	59.3
	Ours	**54.6**	54.6	**55.4**	68.8	71.0	**91.5**	**66.5**	**67.8**	**57.9**	**84.4**	**93.1**	67.3	**72.7**
$ACC_{\frac{\pi}{18}}$	Res50-A	6.1	4.5	7.2	20.1	25.9	21.4	9.5	13.2	6.1	14.0	23.0	8.6	14.5
	Res50-S	5.7	6.9	8.0	25.5	33.9	29.1	13.0	11.6	6.8	18.4	32.0	13.8	18.6
	StarMap	3.8	5.8	2.4	19.7	30.5	24.5	7.7	9.6	5.1	9.6	21.5	5.8	13.9
	NeMo	10.7	10.5	11.3	13.9	55.8	60.6	9.3	20.3	6.3	26.1	34.6	32.1	30.2
	NeMo-M	12.8	**16.6**	16.8	21.9	**62.3**	64.6	17.2	20.3	12.3	32.4	38.2	**32.7**	34.5
	NeMo-S	14.9	11.1	15.6	18.2	56.0	62.4	17.4	18.7	10.2	30.5	36.4	22.4	32.0
	PoseCon.	11.3	13.6	17.3	**41.1**	46.3	38.2	16.4	28.4	12.3	26.7	44.9	17.9	28.1
	Ours	**19.0**	14.4	**22.3**	32.1	50.5	**68.8**	**22.2**	**29.5**	**13.3**	**35.7**	**67.5**	17.6	**38.9**
MedErr	Res50-A	49.3	42.5	58.5	17.7	15.9	21.3	35.4	32.0	36.1	20.3	15.2	25.3	30.4
	Res50-S	45.8	33.9	52.8	16.3	12.4	15.1	27.1	30.9	32.4	18.3	12.3	24.1	26.1
	StarMap	55.2	37.1	69.1	20.6	19.0	21.3	39.2	34.0	35.5	27.0	24.8	40.3	34.1
	NeMo	39.8	37.7	44.2	24.8	8.8	7.7	29.7	28.5	47.5	16.9	18.2	17.0	24.1
	NeMo-M	38.5	**26.4**	38.2	18.8	**7.0**	7.3	23.0	23.0	36.0	14.0	14.9	**16.1**	20.2
	NeMo-S	39.9	30.6	38.8	19.5	8.3	7.8	21.3	24.8	29.5	14.2	16.9	18.5	20.9
	PoseCon.	44.8	29.1	43.8	**12.0**	10.7	14.9	30.3	20.9	25.5	16.7	11.3	24.7	23.1
	Ours	**25.7**	26.7	**24.6**	15.2	9.7	**6.7**	**19.9**	**17.1**	**23.9**	**12.9**	**6.8**	22.5	**16**

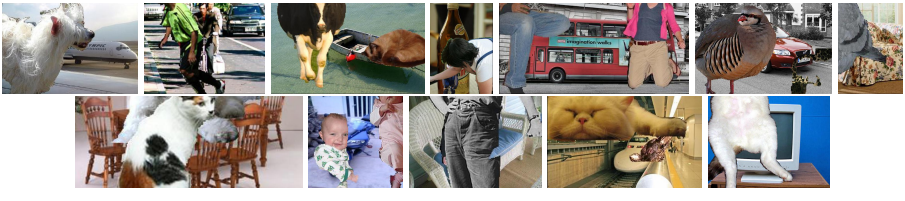
Table 6: Comparison with competing methods on OccludedPASCAL3D L2 per category



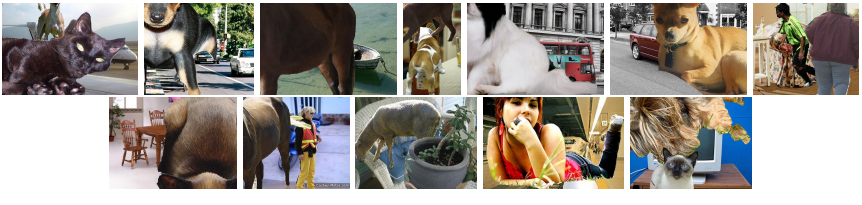
(a) L0



(b) L1



(c) L2



(d) L3

Figure 1: Sample images from the 12 categories of PASCAL3D (L0) and OccludedPASCAL3D (L1-L3) demonstrating the level of occlusion per occlusion category.

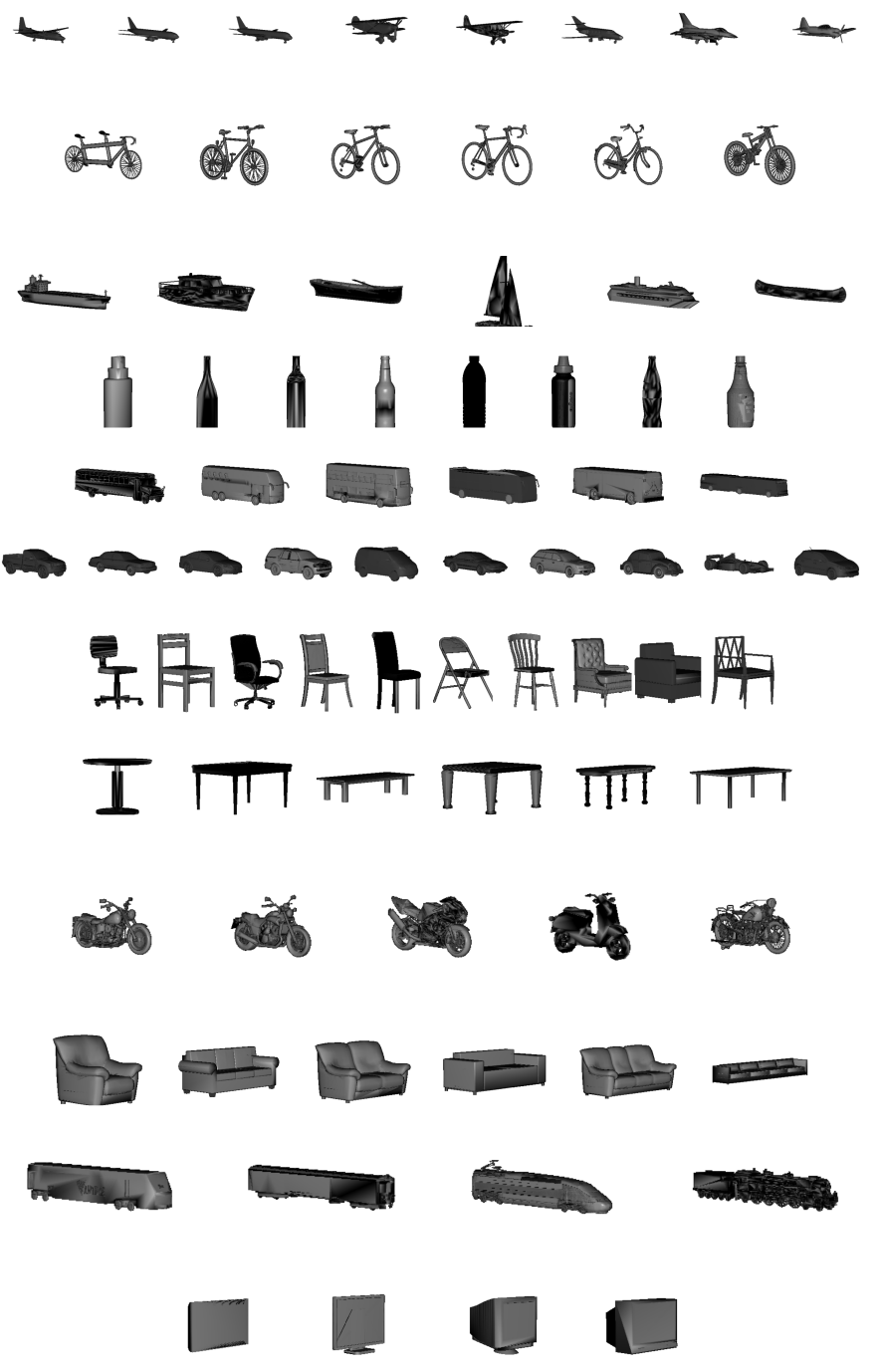
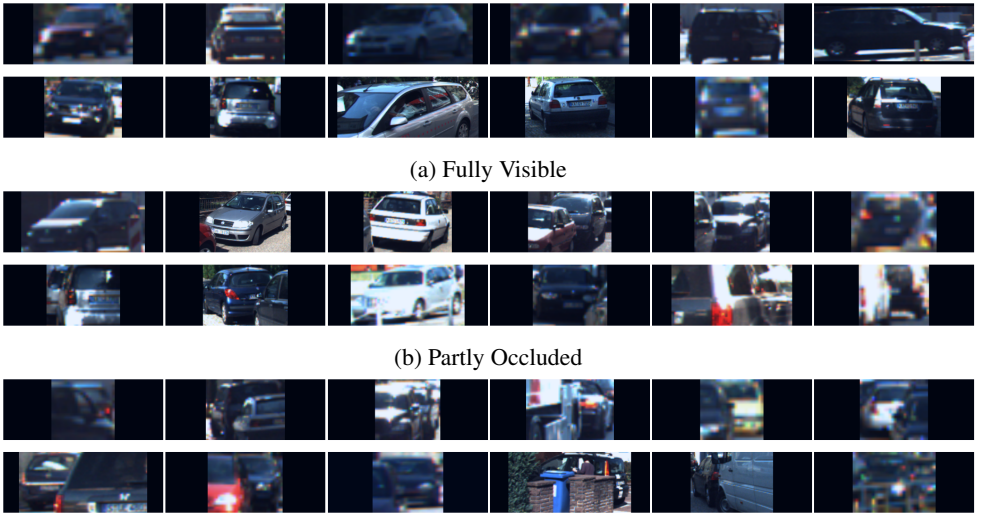


Figure 2: CAD Models per object category of PASCAL3D, from top to bottom, aeroplane, bicycle, boat, bottle, bus, car, chair, diningtable, motorbike, sofa, train, tvmonitor.



(a) Fully Visible

(b) Partly Occluded

(c) Largely Occluded

Figure 3: Sample images from KITTI3D demonstrating the level of occlusion per occlusion category.

		aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
$ACC_{\frac{\pi}{6}}$	Res50-A	18.3	20.8	21.2	62.1	57.0	36.9	31.1	32.2	24.3	56.2	64.5	53.4	37.8
	Res50-S	20.0	33.4	25.5	67.5	57.8	42.0	40.7	33.9	30.3	56.6	82.8	56.5	43.1
	StarMap	7.6	18.5	10.6	46.3	35.1	25.3	22.5	24.6	15.9	26.4	24.0	19.5	22.9
	NeMo	24.0	31.3	27.4	43.3	48.8	62.8	31.8	29.7	18.4	44.2	34.5	51.4	41.3
	NeMo-M	23.8	**34.3**	29.5	53.9	**56.0**	65.5	43.4	41.5	25.4	58.2	43.2	**54.1**	47.1
	NeMo-S	20.6	33.8	27.6	61.7	49.9	61.8	**44.7**	41.2	**35.3**	**62.9**	47.9	50.2	46.8
	PoseCon.	19.2	30.6	27.4	**73.5**	47	35.2	33.3	38.0	33.3	52.1	70.7	44.4	39.7
	Ours	**27.4**	28.8	**31.8**	43.3	41.3	**69.6**	40.9	**45.6**	32.1	62.1	**85.2**	47.8	**49.8**
$ACC_{\frac{\pi}{18}}$	Res50-A	1.6	2.3	2.9	11.9	14.4	7.6	3.8	5.7	3.1	7.9	12.7	8.9	6.7
	Res50-S	2.0	5.5	4.8	16.7	21.1	13.1	5.9	5.7	4.3	9.9	22.5	6.0	9.9
	StarMap	0.8	1.7	1.1	11.8	8.3	4.8	2.1	2.6	1.6	2.8	5.2	0.7	3.7
	NeMo	4.4	6.2	6.7	6.8	26.5	31.1	3.4	6.7	2.0	9.3	13.0	**16.7**	14.5
	NeMo-M	5.5	5.2	7.9	10.8	**34.2**	**37.4**	7.4	8.2	4.5	15.8	15.1	15.9	17.8
	NeMo-S	4.7	**6.7**	8.6	11.7	29.2	33.7	**11.0**	10.7	4.9	**17.8**	17.2	10.9	17.1
	PoseCon.	4.0	5.3	7.1	**26.8**	18.4	13.0	9.1	13.0	**5.0**	15.2	27.4	11.4	12.7
	Ours	**6.5**	3.3	**9.0**	16.9	19.2	32.6	7.2	**14.1**	3.6	15.4	**42.2**	7.0	**17.9**
MedErr	Res50-A	69.8	70.9	73.2	22.7	24.9	46.7	41.5	44.4	59.8	26.3	21.3	28.4	46.4
	Res50-S	65.8	47.1	75.8	20.9	18.5	46.6	35.9	49.9	56.3	26.4	15.3	26.5	44.0
	StarMap	87.0	67.6	90.2	32.6	51.3	64.0	60.7	53.2	73.4	51.0	52.7	54.7	63.0
	NeMo	**65.3**	48.4	65.2	34.5	34.9	17.2	44.6	55.7	74.3	33.7	47.6	29.3	41.8
	NeMo-M	69.8	49.6	63.0	28.2	**19.4**	**14.9**	35.4	39.9	60.0	23.7	38.1	**27.2**	**36.1**
	NeMo-S	74.8	**46.1**	70.1	24.5	30.2	16.3	**35.2**	37.5	50.5	**21.5**	31.7	29.9	36.5
	PoseCon.	66.8	61	64.7	**16.6**	34.2	51.5	41.4	42.5	50.7	28.7	16.7	33.3	45.5
	Ours	75.2	54.5	**61.9**	48.8	53.8	16.1	36.8	**34.2**	**49.2**	23.8	**11.8**	31.1	37.9

Table 7: Comparison with competing methods on OccludedPASCAL3D L3 per category

Model	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
1	99.2	97.4	91.5	69.6	95.9	89.3	68.8	32.6	3.1	4.2	6.7	16.1
2	99.2	97.0	90.2	69.4	96.2	87.2	66.0	29.7	3.2	4.5	7.2	16.5
3	99.1	97.6	90.3	66.5	95.9	87.7	66.5	30.0	3.2	4.3	7.2	17.4
4	99.2	96.6	90.7	69.9	96.4	88.4	70.2	35.2	3.2	4.1	6.7	14.9
5	99.2	97.4	90.2	69.0	96.0	86.6	63.0	29.3	3.4	4.7	8.0	17.1
Mean	99.2	97.2	90.6	68.9	96.1	87.6	66.7	31.4	3.2	4.4	7.2	16.4
SD	0.04	0.4	0.55	1.37	0.22	0.76	2.84	2.51	0.11	0.24	0.53	0.98

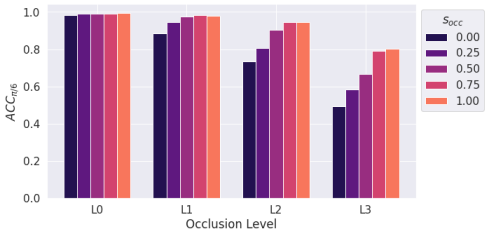
Table 8: Evaluation of performance consistency on PASCAL3D Cars

Training	Inference	fps ↑	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
			L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
TrainDB	TrainDB	35	99.2	**97.4**	**91.5**	**69.6**	**95.9**	**89.3**	**68.8**	**32.6**	**3.1**	**4.2**	**6.7**	**16.1**
CoarseDB	TrainDB	35	99.2	**97.4**	89.1	65.4	95.6	85.2	60.0	25.6	3.2	4.7	8.1	19.9
CoarseDB	CoarseDB	2.5	99.0	95.7	84.5	56.0	91.0	74.0	44.2	14.4	4.2	6.1	11.2	26.2
CoarseDB	Both	0.5	99.0	95.6	84.5	56.0	91.6	74.9	45.2	14.8	3.5	5.8	11.1	26.2
FineDB	TrainDB	35	**99.3**	96.7	90.9	68.9	95.8	86.6	64.3	30.3	3.2	4.6	7.4	17.4
FineDB	FineDB	0.5	97.6	92.5	81.5	52.0	90.7	75.7	48.0	15.7	4.3	6.2	10.4	28.3
FineDB	Both	0.5	98.0	92.9	82.0	52.3	91.7	76.6	48.9	16.1	3.6	5.8	10.3	28.1

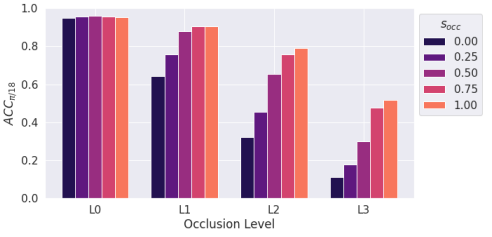
Table 9: Evaluation of the three designed reference sets

Set	Azimuth	Elevation	In-plane Rotation	Renderings
TrainDB	-	-	-	2.7k
CoarseDB	5°	5°	5°	178k
FineDB	1°	5°	5°	889k
Range	0°:360°	−30°:60°	−30°:30°	-

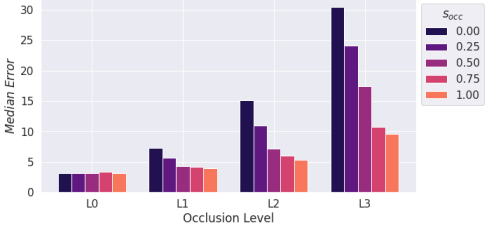
Table 10: Discretization of pose space per rendering set



(a) $ACC_{\frac{\pi}{6}}$

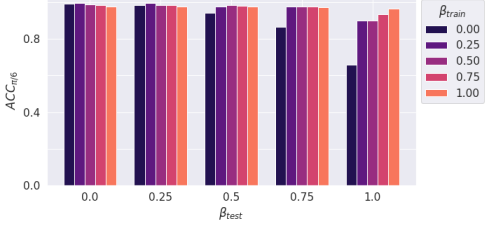


(b) $ACC_{\frac{\pi}{18}}$

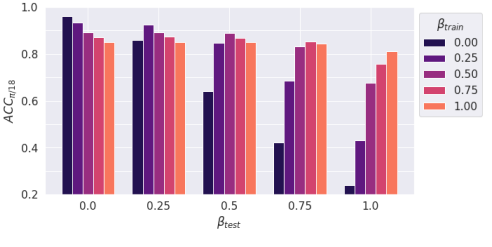


(c) *Median Error*

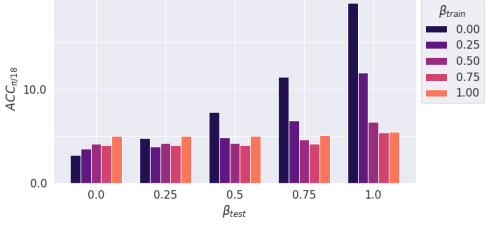
Figure 4: Evaluation of models trained on various occlusion scales s_{occ} on Cars of PASCAL3D and OccludedPASCAL3D.



(a) $ACC_{\frac{\pi}{6}}$

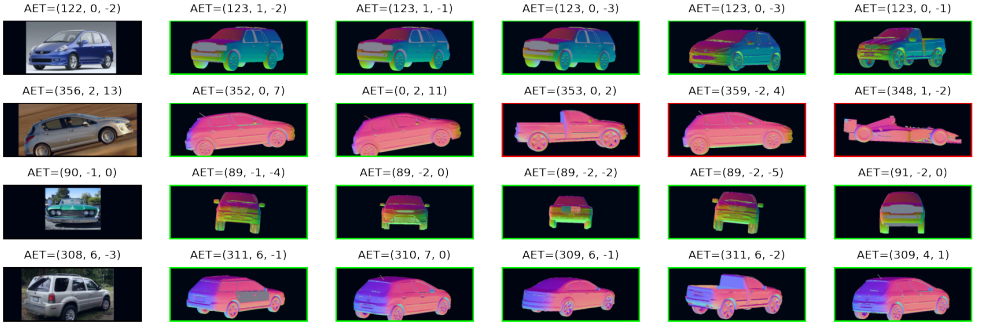


(b) $ACC_{\frac{\pi}{18}}$

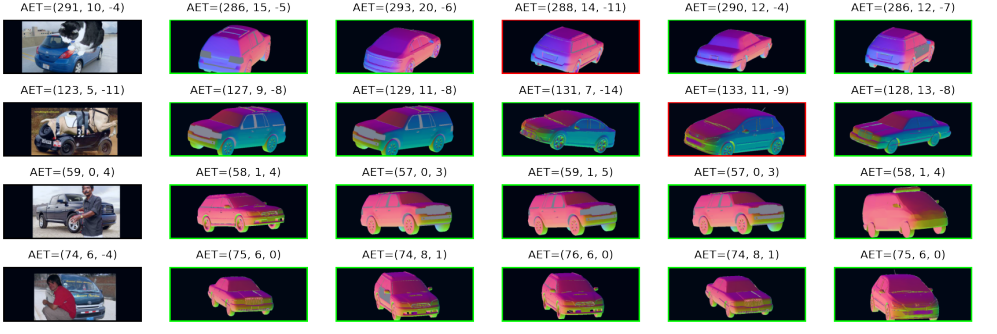


(c) *Median Error*

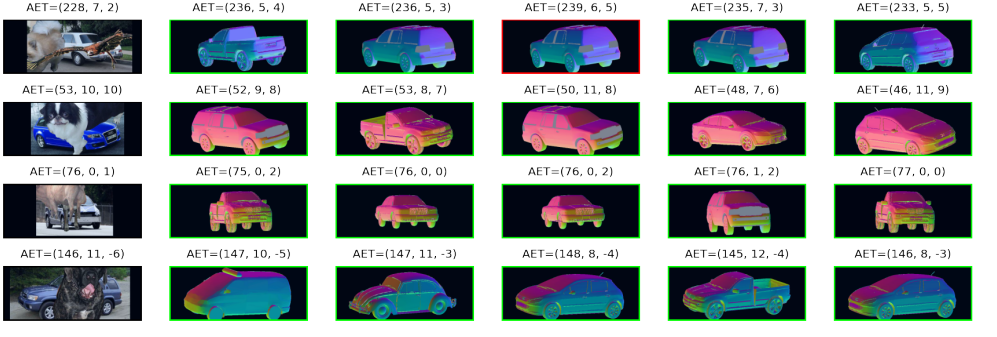
Figure 5: Evaluation of models trained on various levels β_{train} of bounding box augmentation on PASCAL3D L0 Cars with various levels β_{test} of test-time bounding box augmentation.



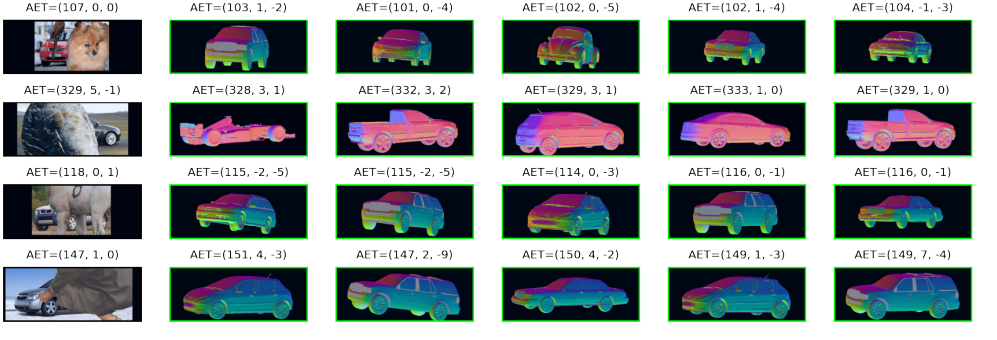
(a) L0



(b) L1

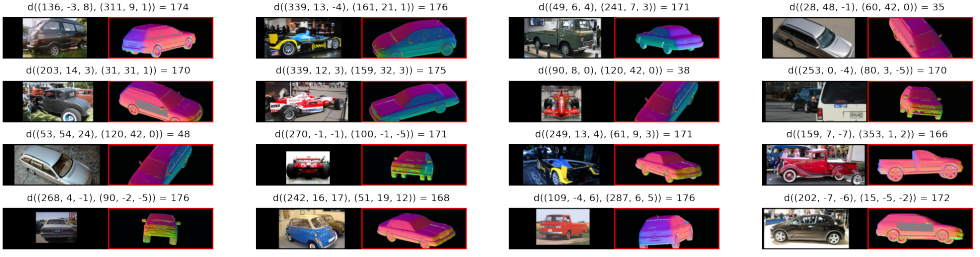


(c) L2

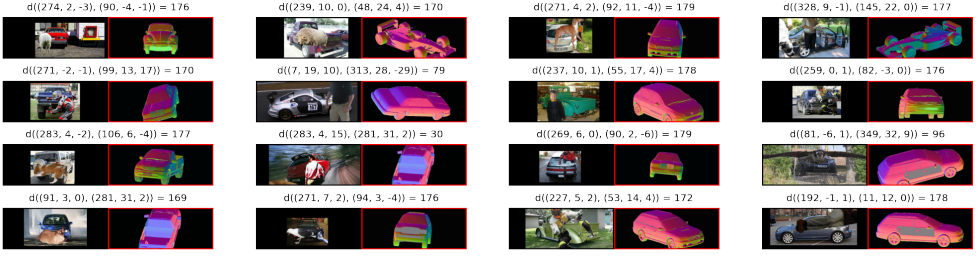


(d) L3

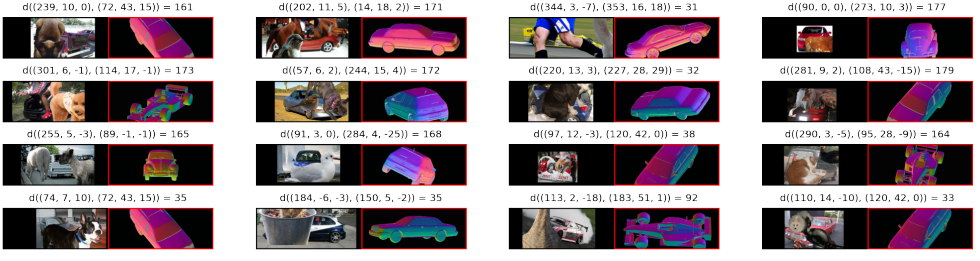
Figure 6: Retrieval of nearest neighbours for occlusion levels L0-L3 in PASCAL3D Cars.



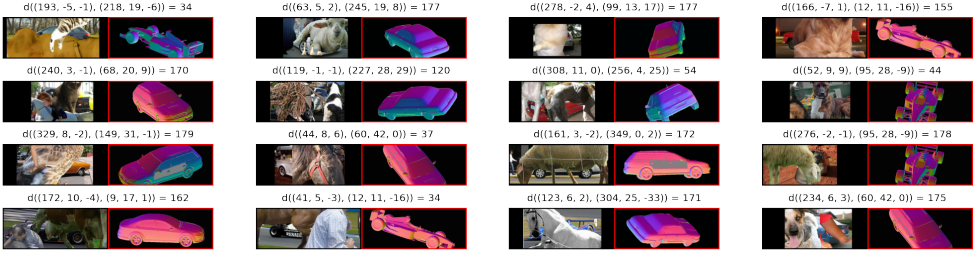
(a) L0



(b) L1

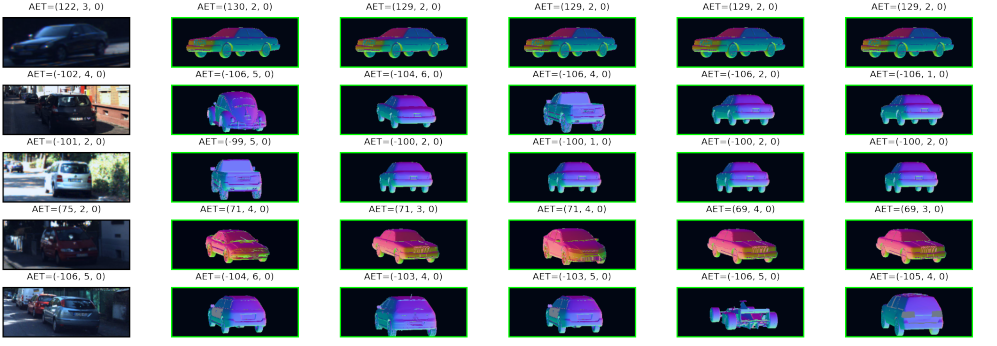


(c) L2

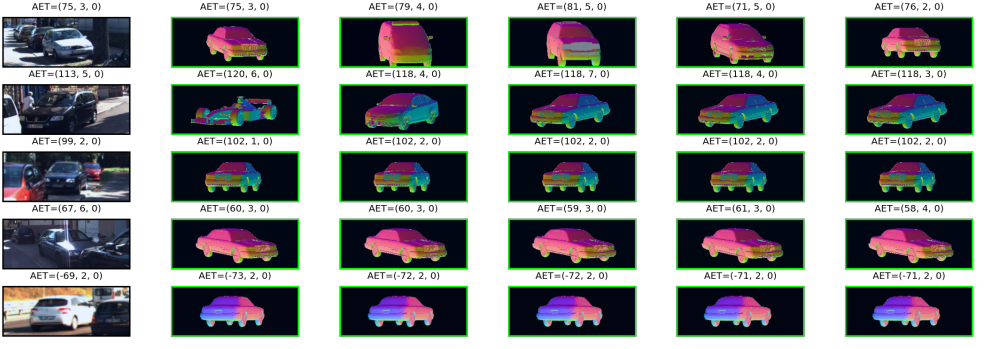


(d) L3

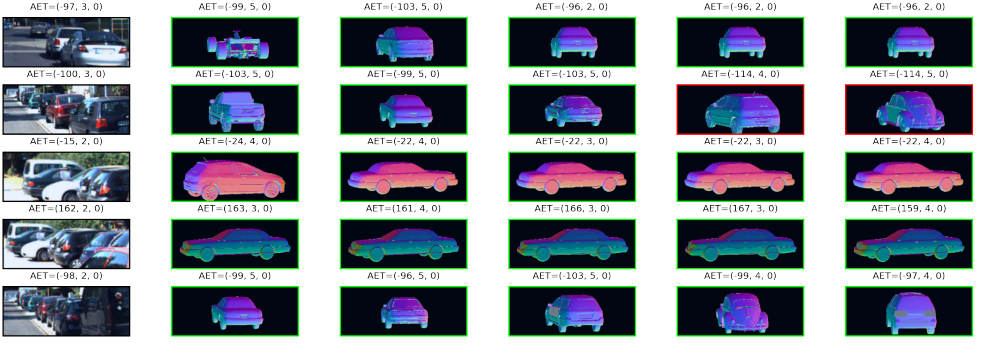
Figure 7: Failure cases for four different levels of occlusion L0-L3 in PASCAL3D Cars.



(a) Fully Visible Objects

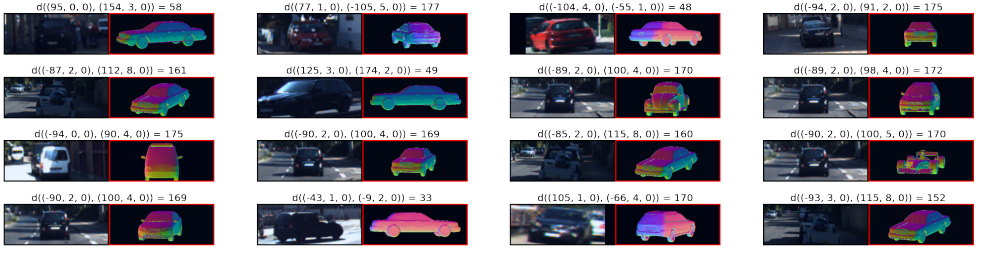


(b) Partly Occluded Objects

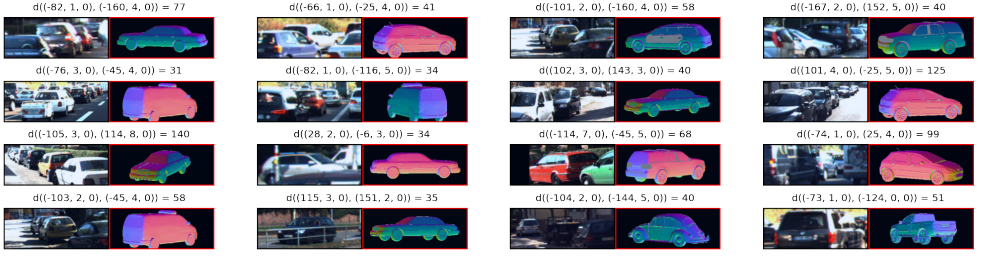


(c) Largely Occluded Objects

Figure 8: Retrieval of nearest neighbours for three different levels of occlusion in KITTI3D Cars.



(a) Fully Visible



(b) Partly Occluded

(c) Largely Occluded

Figure 9: Failure cases for three different levels of occlusion in KITTI3D Cars.