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-VISICS
Department of Electrical Engineering
KU Leuven
Belgium

[2] Ford Greenfield Labs
Palo Alto
California, USA

Abstract

Pose estimation is usually tackled as either a bin classification or a regression problem. In both cases, the idea is to directly predict the pose of an object. This is a non-trivial task due to appearance variations between similar poses and similarities between dissimilar poses. Instead, we follow the key idea that comparing two poses is easier than directly predicting one. Render-and-compare approaches have been employed to that end, however, they tend to be unstable, computationally expensive, and slow for real-time applications. We propose doing category-level pose estimation by learning an alignment metric in an embedding space using a contrastive loss with a dynamic margin and a continuous pose-label space. For efficient inference, we use a simple real-time image retrieval scheme with a pre-rendered and pre-embedded reference set of renderings. To achieve robustness to real-world conditions, we employ synthetic occlusions, bounding box perturbations, and appearance augmentations. Our approach achieves state-of-the-art performance on PASCAL3D and OccludedPASCAL3D and surpasses the competing methods on KITTI3D in a cross-dataset evaluation setting. The code is currently available at https://github.com/gkouros/contrastive-pose-retrieval.

1 Introduction

Estimating the pose of a 3D rigid object is a fundamental task in numerous computer vision applications. For instance, a self-driving vehicle must be able to estimate the pose of other

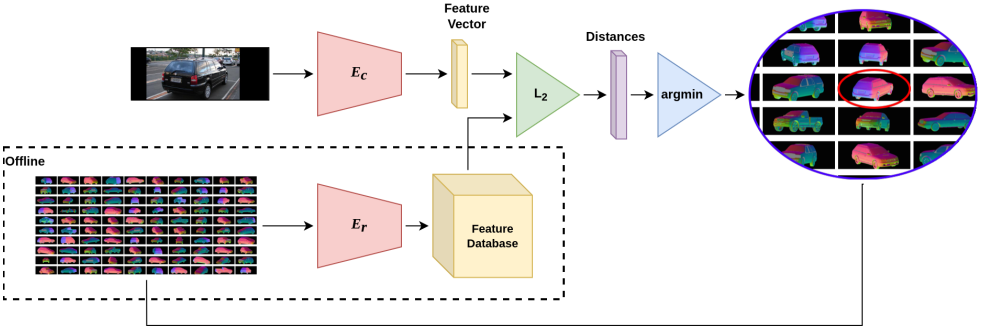


Figure 1: Using a properly learned metric, all it takes to estimate the pose of an object with state-of-the-art accuracy is a simple retrieval scheme that finds the most similar encoded rendering in a database. Two ResNet-50 encoders E_c and E_r are jointly trained in a contrastive manner to learn the mapping of query camera images and reference renderings to a feature space where their feature distance is proportional to their geodesic/pose distance. To ensure fast online inference after training, the reference set is encoded offline.

road users in its surroundings in order to navigate safely without endangering itself or others. Deep learning has revolutionized such pose estimation tasks especially for challenging monocular settings [1, 3, 10, 13, 15, 24, 25] compared to stereo [4, 5, 8, 19, 35] or RGB-D [28, 32] settings that leverage 3D information from their inputs. Two ongoing problems in the monocular setting are how to best extract 3D information from 2D image inputs and how to achieve real-time operation and robustness even in complex, cluttered and occluded real-world scenes.

Previous methods approach pose estimation as either a classification [33], regression [20, 30, 39, 41] or optimization problem [1, 3, 15]. Classification and regression have to directly predict the pose as either belonging to a bin or as a set of continuous values. On the other hand, optimization methods such as render-and-compare approaches iteratively optimize the pose. Comparing two images with regard to their pose can be considered a much easier task to learn. Nevertheless, such an iterative optimization approach, although accurate, may be too slow for real-time category-level pose estimation.

In this work, we propose a multimodal contrastive learning framework for extracting discriminative features from real-world images and renderings that can be used for comparing the two images with regard to their poses. Poses in this work refer to the 3D orientation of the camera with respect to an object expressed with the azimuth, elevation, and in-plane rotation angles. Rather than following a slow iterative approach similar to render-and-compare methods [1, 3], we utilize a common nearest neighbour retrieval scheme that compares the feature embedding of a query image with a reference set of embeddings from rendered objects in various poses. We also increase robustness to complex and cluttered scenes by augmenting training images with appearance variations, bounding box perturbations, and synthetic occlusions. Our main contributions can be summarized as follows:

- We propose a simple yet effective category-level pose retrieval framework based on learning discriminative features using contrastive learning with a dynamic margin.
- We show that strong data augmentation can enhance a simple pose estimation architecture to outperform more complex ones.

- We report state-of-the-art results on PASCAL3D and OccludedPASCAL3D, as well as superior cross-dataset performance on KITTI3D against the evaluated competing methods.

2 Related Work

Monocular pose estimation can be described as an ill-posed problem due to the lack of 3D information despite the fact that good empirical results have been obtained. Recovering 3D information is usually accomplished through either monocular depth prediction [9, 13, 18] or by incorporating prior hypotheses about the objects such as shape priors for template matching [1, 15, 43, 45] in render-and-compare or image retrieval settings. In this work, we utilize prior shape hypotheses in the form of CAD models for category-level pose estimation. We specifically target category-level methods to achieve a good trade-off between accuracy, generalization, and robustness to occlusions and clutter compared to instance-based and category-agnostic methods.

In the scope of render-and-compare approaches, RePose [15] runs faster than real-time by optimizing the pose of an object with learned deep textures, but is applicable only at the instance-level. Beker *et al*. [3] and Wang *et al*. [35] propose render-and-compare methods for estimating the pose and shape of cars using photometric, depth, or silhouette comparison, but without achieving real-time performance. NeMo [1], on the other hand, uses a generative neural mesh model and contrastive learning to first learn discriminative features that distinguish objects from occlusions and background clutter before optimizing the pose through a render-and-compare scheme for approximately 8 seconds per object. To avoid the overhead of render-and-compare optimization, we choose a simple yet efficient image retrieval setting.

Retrieval-based methods rely on a good comparison metric and deep metric learning with contrastive [6] or triplet-like [29] losses has been instrumental towards that end. Wohlhart and Lepetit [37] first proposed to use a triplet-like loss to optimize a CNN feature extractor for instance-level pose estimation tasks based on nearest neighbour retrieval. Zhakarov *et al*. [43] augmented the triplet-like loss with a dynamic margin that considers both the object instance and the pose distance between anchor/positive and negative samples. All aforementioned methods, however, discretize the pose space into bins with a negative impact on accuracy. Balntas *et al*. [2] incorporated a regression loss term during training that further improved performance. Papaioannidis and Pitas [25] added a regression loss term as well that enabled direct pose regression instead of slow nearest neighbour search. PoseContrast [40] is trained with a loss function, combining classification, regression, and contrastive loss terms, on real data with pose-aware augmentations and without prior geometry knowledge. In this work, we propose a simple contrastive loss with a dynamic margin and a continuous pose space achieved by choosing positive and negative pairs and optimizing according to pose distance rather than binning. In our case, each pair consists of a real image (anchor) and a rendering of a CAD model (positive/negative), for which we jointly train two individual feature extractor CNNs.

Achieving robustness to foreground occlusions and background clutter usually requires complex architectures and loss functions. For instance, NeMo [1] employs contrastive learning to learn how to distinguish between object features and background clutter or foreground occlusions. In a different approach, Sarandi *et al*. [31] propose to augment input images with synthetic occlusions to increase robustness in human pose estimation tasks. In this work, we follow the same approach and incorporate a synthetic occlusion augmentation scheme for

the pose estimation of 3D rigid objects rather than humans.

3 Method

3.1 Learning Discriminative Pose Features

The main idea in deep metric learning is to optimize a high-dimensional embedding feature space or manifold so that samples are pulled together or pushed apart depending on whether they belong to the same class or not. This task was first accomplished using the Contrastive Loss [6] which is defined as

$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\left[(1-y_i)\ ||f_{1,i}-f_{2,i}||_2^2 + y_i\ max(0,\ m-||f_{1,i}-f_{2,i}||_2^2)\right], \tag{1}$$

where N is the batch size or number of sample pairs, m is the margin, f is the embedding/encoding function, and y_i is a label that is 1 if the pair is positive and 0 if negative.

Applying deep metric learning for pose estimation requires discretizing the pose space and assigning the pose labels to bins as in [37, 43]. However, this means that slightly different poses might fall in different bins and thus the network would be encouraged to separate them in feature space, which would negatively impact generalization to unseen poses. Consequently, similar to the Triplet-like Dynamic Margin Loss [43], we employ a dynamic margin that is proportional to the geodesic pose distance between two samples. In contrast to [43], we train our models for pose estimation on the category-level rather than the instance-level, and avoid the discretization of the pose-label space that negatively impacts accuracy. While they use the discretized pose labels to determine positive and negative sample pairs, we propose a continuous pose-label space and determine positive and negative pairs by applying a threshold on the pose distance. As a result, we redefine the Contrastive Loss from Equation 1 to our *Contrastive Pose Loss* expressed as

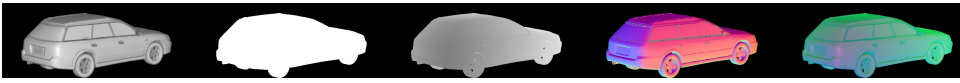
$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\left[(1-y_i)\ max(0,\ ||f^c_{1,i}-f^r_{2,i}||_2^2 - m\Delta\theta) + y_i\ max(0,\ m\Delta\theta - ||f^c_{1,i}-f^r_{2,i}||_2^2)\right], \tag{2}$$

where $\Delta\theta = 2\ cos^{-1}(|q_i \cdot q_j|)$ denotes the geodesic distance between two poses expressed as quaternions q_i, q_j. Furthermore, f^c and f^r denote the embedding functions of encoders E_c and E_r, respectively.

3.2 Sampling and Mining

According to numerous works [11, 14, 23, 27, 29, 36, 42], Deep Metric Learning performance is heavily influenced by the selection of samples in a mini-batch during training and thus a sophisticated scheme is essential to speeding up convergence.

Datasets often suffer from imbalance, such as typical car datasets having more sedan cars than vans, which may result in poorer performance in the latter subcategory. This can be alleviated by a sampling scheme that weights each sample inversely proportional to the number of occurrences of its subcategory, an idea inspired by the Focal Loss [21]. In addition, for every sample already chosen for a batch, the sampling scheme aims to include $N \geq 1$ additional samples with a similar pose (e.g. less than 5° difference) because they are more likely to have a small feature distance that needs to be optimized.



(a) RGB (b) Silhouette (c) Depth Map (d) Normals Map (e) Triplet Map

Figure 2: Five types of renderings that were evaluated for pose estimation.

We also designed a pose-aware miner that looks for pairs of samples violating the pose margin, resulting in higher losses and thus more efficient optimization. The sampler feeds the miner a mini-batch of N indices corresponding to N samples composed of a camera image and its rendered counterpart. All possible positive and negative pairs are constructed based on a threshold (e.g. 5°), of which all pairs that violate the pose margin are used to calculate the loss. The rest are dropped since they do not offer any value to the optimization.

3.3 Rendering

In contrast to NeMo [1] and similar to [43], we do not use a 3D generative model for producing the renderings, but rather employ a more conventional approach of generating 2D mesh renderings, silhouettes, surface normal maps, depth maps, or even multi-channel RGB-depth-normal combinations that were inspired from [10] and which we call triplets. It is our intuition that using such feature representations, as illustrated in Figure 2, preserves perspective information vital to pose estimation tasks as opposed to the 3D generative model used by NeMo. Rather than generating a set of renderings for the entire viewing sphere, we create a rendering database by generating one rendering per sample in the training set thus ensuring at least one positive per sample. This approach requires less space and less time for inference while avoiding the need to find a trade-off between discretization error, the size of the database, and inference time. Moreover, this naturally reflects the prior distribution over the viewing sphere.

3.4 Robustness to Occlusions

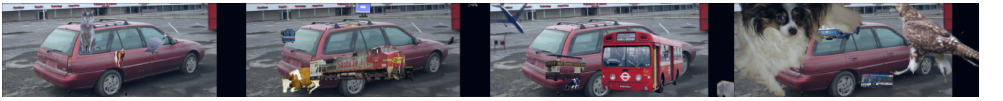
To increase robustness to occlusions and background clutter, we use data augmentation with synthetic occlusions [31] produced from PASCAL VOC 2012 [7]. This involves segmenting objects from PASCAL VOC to create a template set of occluders from 20 object categories. An input image is then augmented with one to eight randomly selected occluders which are visually, spatially, and geometrically augmented. When training with a specific object class we naturally filter out that class from the occluder set to avoid having objects from that class occluding the actual object of interest. Finally, for tuning purposes we use a tunable occlusion scale s_{occ} that is multiplied with a random resize factor $x \sim U[0,1]$ to produce the resize factor for a random occluder

$$f_x = f_y = s_{occ}\, x, \tag{3}$$

where f_x and f_y are the resizing factors for the horizontal and vertical dimensions, respectively. Figure 3 illustrates occlusion augmentations and the effect of s_{occ}.

3.5 Robustness to Bounding Box Noise

Throughout the experiments we assume known scale and center of the objects similar to NeMo [1]. Practically, this is not realistic and although NeMo implies some basic toler-



(a) $s_{occ} = 0.25$ (b) $s_{occ} = 0.5$ (c) $s_{occ} = 0.75$ (d) $s_{occ} = 1.0$

Figure 3: Examples of synthetic occlusions for various scale factors. For examples of real occlusions we refer the reader to Figure 3 in the supplementary material.

(a) $IoU \geq 0.0$ (b) $IoU \geq 0.25$ (c) $IoU \geq 0.5$ (d) $IoU \geq 0.75$ (e) $IoU = 1.0$

Figure 4: Artificial bounding box noise with a lower boundary on IoU. Green denotes the original image borders and red denotes randomly perturbed bounding boxes.

ance to center/scale perturbations, it is designed in a way that works optimally with adequate alignment between camera images and renderings. In order to avoid this restriction we propose augmenting training samples with random bounding box noise with a lower boundary on IoU. To define this type of noise we express the deviation of the bounding box corners as a function of the lower IoU boundary. If w and h are the width and height of the bounding box and n is the maximum horizontal and vertical corner deviation in pixels, then

$$IoU_{min} = \frac{(w-2n)(h-2n)}{wh} \,. \tag{4}$$

By solving the quadratic equation we can calculate the maximum pixel deviation n as a function of IoU_{min} via the equation

$$n = \frac{h+w-\sqrt{(h+w)^2-4wh\beta}}{4} \,, \tag{5}$$

where $\beta = 1 - IoU_{min}$ is the noise scale parameter used in the experiments. Figure 4 presents a few examples of our bounding box noise scheme for different IoU lower boundaries.

3.6 Inference via Pose Retrieval

After jointly training the encoders E_c and E_r, we can predict poses through a simple image retrieval scheme, as shown in Figure 1. Our inference framework requires an offline step of generating a reference set of renderings which need to be embedded using encoder E_r and stored for online inference. Inferring the pose is then basically a two-step-approach composed of encoding a query image with encoder E_c, calculating the L_2 distance of the query embedding to all feature embeddings in the stored reference set, and finally finding the nearest neighbour, whose label corresponds with our predicted pose.

There are two main limitations with this approach. First, inference requires discretization to a set of sampled orientations, which can be either the orientations that are present in the training set or the orientations in a generated reference set that introduces a tradeoff between discretization and inference speed. In the end, we used the first approach to ensure fast training and inference speed. Second, comparing an encoded query image against a reference set introduces a delay not present in classification or regression approaches. Training a

regression layer on top of the E_c encoder, similar to [2, 40] would potentially eliminate both issues.

4 Experiments

4.1 Experimental Setup

Our framework is developed using *PyTorch* [26], *PyTorch Metric Learning* [24], and we also use *PyTorch3D* [16] for the generation of the renderings. We jointly train two ResNet50 [12] encoders E_c and E_r as shown in Figure 1. MLP heads are used to further reduce the dimensionality of the feature space from 2048 to 512. Each model was trained on an NVidia Titan V GPU with 12GB of memory for approximately 2-3 days depending on the object category and subsequent dataset size. For evaluation, we use the test set images as the query set and the training renderings as the reference set.

We train with a batch size of 32 sample pairs for 1000 epochs using the Adam Optimizer [17] with a learning rate of 10^{-4} for the ResNet50 encoders and 10^{-3} for the MLP heads. A weight decay equal to $5 \cdot 10^{-4}$ is used for both the backbone and MLP head. The embedding size is set to 512 and the loss margin is set to $m = 1$. For sampling and mining we use a positive/negative threshold $t_{\Delta\theta} = 5^{\circ}$. Finally, unless stated otherwise, we use $\beta_{train} = 0.1$ and $s_{occ} = 0.5$. To make a more fair evaluation, we train with occlusions produced from PASCAL VOC 2012, and not from MS-COCO [22] as in OccludedPASCAL3D. At the same time, we intentionally use smaller occluders compared to the L1-L3 sets as can be observed by comparing Figure 3 from this text and Figure 1 from the supplementary material.

We evaluate our approach on PASCAL3D [38], its synthetically occluded counterpart OccludedPASCAL3D [34], and KITTI3D [9]. Unless stated otherwise, we use surface normal maps in all experiments. An evaluation of the various rendering types is included in the supplementary material. Similar to [1] we assume known center and distance for all samples, however, we train to achieve robustness to bounding box perturbations and avoid over-reliance on 2D detection accuracy by employing training time bounding box augmentations. As a result, our models learn to disregard distance information when comparing camera images and renderings and instead solely focus on pose information. We further increase the training data variance through horizontal flipping, color jittering, gaussian blurring, bounding box perturbations, and synthetic occlusions.

We compare the performance of our approach against a category-agnostic classifier Res-50-A and a category-specific classifier Res50-S from [1] as well as three state-of-the-art competing methods, namely StarMap [45], NeMo [1], and PoseContrast [40]. For NeMo, in particular, we compare against all three variations termed as NeMo, NeMo-MultiCuboid (NeMo-M), and NeMo-SingleCuboid (NeMo-S). We follow the exact same preprocessing and evaluation methodology as in NeMo and thus borrow their results and the results for Res50-A, Res50-S, and StarMap. PoseContrast, however, was originally trained with more data, so we had to retrain it with the same amount of data as the rest of the methods to ensure a fair comparison. Similar to StarMap and NeMo, we perform the evaluation using three metrics, namely pose accuracy with a threshold of 10° ($ACC_{\frac{\pi}{18}}$) and 30° ($ACC_{\frac{\pi}{6}}$), as well as *Median Error*. Furthermore, we evaluate the inference speed of our approach and compare it against the best competing methods, namely NeMo and PoseContrast.

	Categ. aware	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
		L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
Res50-A †		88.1	70.4	52.8	37.8	44.6	25.3	14.5	6.7	11.7	17.9	30.4	46.4
Res50-S †	✓	87.6	73.2	58.4	43.1	43.9	28.1	18.6	9.9	11.8	17.3	26.1	44.0
StarMap †		89.4	71.1	47.2	22.9	59.5	34.4	13.9	3.7	9.0	17.6	34.1	63.0
NeMo †	✓	84.1	73.1	59.9	41.3	60.4	45.1	30.2	14.5	9.3	15.6	24.1	41.8
NeMo-M †	✓	86.7	77.2	65.2	47.1	63.2	49.9	34.5	17.8	8.2	13.0	20.2	**36.1**
NeMo-S †	✓	86.1	76.0	63.9	46.8	61.0	46.3	32.0	17.1	8.8	13.6	20.2	36.5
PoseCon		90.8	76.2	59.3	39.7	67.2	46.4	28.1	12.7	7.1	12.6	23.1	45.5
Ours	✓	**92.3**	**85.7**	**72.7**	**49.8**	**72.2**	**56.7**	**38.9**	**17.9**	**6.6**	**9.7**	**16.0**	37.9

Table 1: Evaluation against the state-of-the-art on PASCAL3D (L0) and OccludedPASCAL3D (L1-L3). The results are averaged across the 12 object categories and the symbol † denotes results taken from [1].

Occl.	Method	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	Mean
L0	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
	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**
L1	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
	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**
L2	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
	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**
L3	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
	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**

Table 2: $ACC_{\frac{\pi}{6}}$ per PASCAL3D category against the two best competing methods [1, 40].

4.2 Robust and Efficient 3D Pose Estimation

In Tables 1 and 2 we present our performance against the competing methods from [1, 40, 45]. The results in Table 1 are averaged over all object categories for the levels of synthetic occlusion 0% (L0), 20-40% (L1), 40-60% (L2), and 60-80% (L3), respectively. Similar to [1], we use a weighted average that takes into account the number of samples per object category. Overall, our approach outperforms the competing methods across all occlusion levels showcasing the benefit of strong data augmentation compared to complex and specialized architectures.

In Table 3, we present our results on KITTI3D. Since the test set of KITTI3D does not provide labels, we split the training set based on the 50-50 split proposed by [4]. NeMo-MultiCuboid requires car type labels and StarMap requires object keypoints which are not provided by KITTI3D, so we evaluate solely against NeMo-SingleCuboid and PoseContrast. Even without retraining or fine-tuning on KITTI3D, our approach exhibits similar performance as in PASCAL3D, outperforming the competing methods in the Fully-Visible (FV), Partly-Occluded (PO) and Largely-Occluded (LO) evaluation categories. At the same time, we demonstrate robustness to out-of-distribution occlusions considering that we trained on

	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
	FV	PO	LO	All	FV	PO	LO	All	FV	PO	LO	All
NeMo-S	88.1	72.4	34.9	67.9	70.3	40.4	7.5	43.7	7.3	11.6	46.1	20.0
PoseContrast	97.8	88.5	48.6	80.6	81.6	62.4	18.6	57.5	6.6	8.6	33.0	15.0
Ours	**98.1**	90.0	56.1	83.4	92.8	70.6	21.0	65.3	3.2	5.4	24.8	10.2
Ours-2	97.9	**90.6**	**66.5**	**86.5**	**94.2**	**74.4**	**34.4**	**70.9**	**2.9**	**5.3**	**15.5**	**7.3**

Table 3: Evaluation on cars of KITTI3D without retraining or fine-tuning. Ours-2 was trained with higher bounding box noise $\beta_{train} = 0.75$ showcasing the benefit of this augmentation technique to cross-dataset performance.

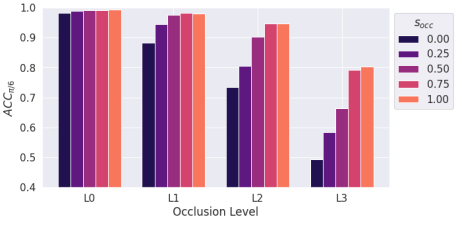


Figure 5: Comparison of models trained with different levels s_{occ} of synthetic occlusion and evaluated on L0-L3.

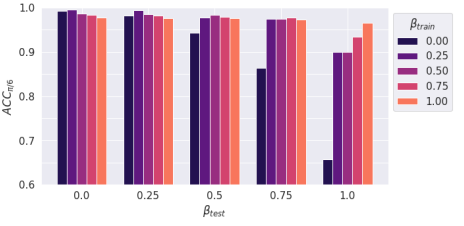


Figure 6: Comparison of models trained with different bounding box noise levels β_{train} on perturbed L0 by β_{test}.

cars without same-category occlusions.

To evaluate the effect of occlusion augmentation in our approach, we trained five models with different occlusion scales s_{occ} and evaluated them on $L0$-$L3$. As shown in Figure 5, the higher the occlusion scale s_{occ} during training, the more robust the model becomes to higher levels of occlusions. We also note that even on $L0$ the models trained with synthetic occlusions outperform the ones without demonstrating robustness to real occlusions and clutter.

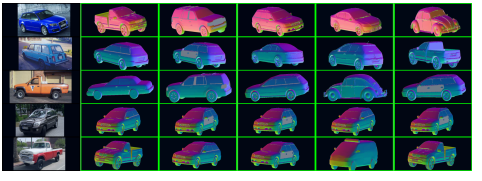
To evaluate the effect of bounding box augmentation we train five models with different levels of bounding box noise and evaluate each one on increasing levels of test time augmentation. Based on the graphs in Figure 6, training with bounding box augmentations leads to increased robustness to higher β_{test} noise values. However, performance drops slightly for larger β_{train} values when evaluating on the unperturbed datasets ($\beta_{test} = 0$).

To evaluate the inference speed of our approach we count the duration of embedding a query image and retrieving the closest neighbour. We ran this experiment on a consumer-grade GPU, namely NVidia GTX 1050, and averaged the measurements over one thousand runs. Our approach runs at $35\,fps$ or requires approximately $29ms$ per object instance, significantly faster than the 8-second inference of NeMo, but still almost double the $15ms$ inference time of PoseContrast that uses a mix of classification and regression.

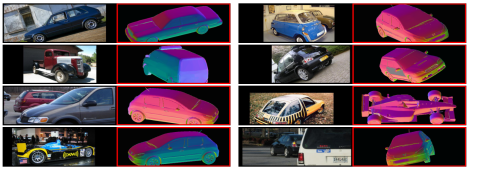
In Figure 7a, we present examples of queries with their corresponding rendered retrievals demonstrating pose estimation accuracy and feature extraction invariant to specific CAD models. In Figure 7b we present failure cases of queries with wrong retrievals which have an angle error higher than 30°. We observe that most failure cases are due to confusion between opposite directions, same-category occlusions, rarely-seen poses, or atypical vehicles. More examples are included in the supplementary material.

	$ACC_{\frac{\pi}{6}} \uparrow$				$ACC_{\frac{\pi}{18}} \uparrow$				$MedErr \downarrow$			
	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
Ours	**99.2**	**97.4**	91.5	69.6	95.9	**89.3**	**68.8**	**32.6**	3.1	**4.2**	**6.7**	**16.1**
w/o multi-CAD	99.0	97.3	**91.8**	70.1	**96.2**	88.1	65.4	30.2	**3.0**	4.3	7.3	16.2
w/o data augment.	**99.2**	97.2	90.3	**70.7**	94.9	85.0	62.9	29.9	3.5	4.7	7.8	16.2
w/ same cat. occl.	99.0	97.1	90.8	68.2	95.8	87.4	64.5	28.6	3.6	4.7	7.5	17.3
w/o contin. labels	98.6	95.5	86.4	59.7	90.4	78.0	54.0	23.0	5.2	6.3	9.3	22.3
w/ Triplet Loss [43]	98.3	94.8	84.6	60.6	93.6	83.6	62.5	31.6	4.2	5.3	7.7	17.1
w/o syn. occlusions	97.8	88.3	74.9	54.6	94.7	67.8	37.7	13.8	3.2	6.9	13.3	25.9
w/o dyn. margin	36.9	36.5	35.8	33.4	25.5	19.2	13.7	7.4	57.9	56.7	53.7	50.0

Table 4: Ablation study results on cars of PASCAL3D L0-L3



(a) Successful retrievals ($\Delta\theta < 10°$).



(b) Failed retrievals ($\Delta\theta > 30°$).

Figure 7: Examples of nearest neighbour retrievals.

4.3 Ablation Study

We conducted an ablation study in which we evaluated our approach for seven distinct cases, as presented in Table 4. First, we trained with only a single CAD model arbitrarily chosen as the sedan for the car category. Second, we trained without appearance augmentations (color jitter, gaussian blur, horizontal flipping). Third, we evaluated our approach with same-category occlusions. Fourth, we used discretized pose labels. Fifth, we used a Triplet loss with a dynamic margin rather than our proposed contrastive loss. Sixth, we trained without synthetic occlusions and in the final case, we removed the dynamic margin. In all cases we observed a non-negligible decrease in performance.

5 Conclusion

Object Pose estimation in a monocular setting is a non-trivial task, especially when dealing with occlusions, clutter, and appearance variations that make handcrafted approaches more prone to error or lack of accuracy. Therefore, we propose learning a pose alignment metric using a contrastive loss with a dynamic margin for comparing object images and renderings with regard to their pose. We reinforce the robustness of the metric using synthetic occlusions and other appearance augmentations. The metric learnt with our Contrastive Pose Loss can be used for pose estimation in an efficient real-time image retrieval setting and achieves state-of-the-art performance on PASCAL3D and OccludedPASCAL3D, as well as high cross-dataset performance on KITTI3D.

Acknowledgements

We gratefully acknowledge funding support from the Sim2Real project, in the context of the Ford-KU Leuven alliance program.

References

[1] Wang Angtian, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. In *Proceedings International Conference on Learning Representations (ICLR)*, 2021.

[2] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. Pose guided rgbd feature learning for 3d object pose estimation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3876–3884, 2017.

[3] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Monocular differentiable rendering for self-supervised 3d object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58589-1.

[4] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/6da37dd3139aa4d9aa55b8d237ec5d4a-Paper.pdf.

[5] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12533–12542, 2020. doi: 10.1109/CVPR42600.2020.01255.

[6] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012. URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[8] Divyansh Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Wasserstein distances for stereo disparity estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

[10] Alexander Grabner, Yaming Wang, Peizhao Zhang, Peihong Guo, Tong Xiao, Peter Vajda, Peter M. Roth, and Vincent Lepetit. Geometric correspondence fields: Learned differentiable rendering for 3d pose refinement in the wild. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 102–119, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58517-4.

[11] B. Harwood, V. Kumar B.G., G. Carneiro, I. Reid, and T. Drummond. Smart mining for deep metric learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2840–2848, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. doi: 10.1109/ICCV.2017.307. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.307.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[13] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:8409–8416, 07 2019. doi: 10.1609/aaai.v33i01.33018409.

[14] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. URL http://arxiv.org/abs/1703.07737.

[15] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3303–3312, October 2021.

[16] Justin Johnson, Nikhila Ravi, Jeremy Reizenstein, David Novotny, Shubham Tulsiani, Christoph Lassner, and Steve Branson. Accelerating 3d deep learning with pytorch3d. In *SIGGRAPH Asia 2020 Courses*, SA '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450381123. doi: 10.1145/3415263.3419160. URL https://doi.org/10.1145/3415263.3419160.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[18] Huifang Kong, Tiankuo Liu, Jie Hu, Yao Fang, and Jixing Sun. Unsupervised monocular depth and pose estimation using multiple masks based on photometric and geometric consistency. In *2020 Chinese Automation Congress (CAC)*, pages 3558–3563, 2020. doi: 10.1109/CAC51589.2020.9326951.

[19] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7636–7644, 2019. doi: 10.1109/CVPR.2019.00783.

[20] Kyaw Zaw Lin, Weipeng Xu, Qianru Sun, Christian Theobalt, and Tat-Seng Chua. Learning a disentangled embedding for monocular 3d shape retrieval and pose estimation. *CoRR*, abs/1812.09899, 2018. URL http://arxiv.org/abs/1812.09899.

[21] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 42(02):318–327, feb 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2858826.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[23] R. Manmatha, Chao-Yuan Wu, Alexander Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2859–2867, 10 2017. doi: 10.1109/ICCV.2017.309.

[24] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Pytorch metric learning, 2020. URL https://github.com/KevinMusgrave/pytorch-metric-learning.

[25] Christos Papaioannidis and Ioannis Pitas. 3d object pose estimation using multi-objective quaternion learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2683–2693, 2020. doi: 10.1109/TCSVT.2019.2929600.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[27] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CR1XOQ0UTh-.

[28] Lounès Saadi, Bassem Besbes, Sébastien Kramm, and Abdelaziz Bensrhair. Optimizing rgb-d fusion for accurate 6dof pose estimation. *IEEE Robotics and Automation Letters*, 6(2):2413–2420, 2021. doi: 10.1109/LRA.2021.3061347.

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. doi: 10.1109/cvpr.2015.7298682. URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

[30] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2686–2694, 2015. doi: 10.1109/ICCV.2015.308.

[31] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. How robust is 3d human pose estimation to occlusion? In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS) Workshops*, 2018.

[32] Meng Tian, Liang Pan, Marcelo H Ang Jr, and Gim Hee Lee. Robust 6d object pose estimation by learning rgb-d features. In *International Conference on Robotics and Automation (ICRA)*, 2020.

[33] S. Tulsiani, J. Carreira, and J. Malik. Pose induction for novel object categories. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 64–72, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.16. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.16.

[34] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654, 2020.

[35] Rui Wang, Nan Yang, Jörg Stückler, and Daniel Cremers. Directshape: Direct photometric alignment of shape priors for visual vehicle pose and shape estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11067–11073, 2020. doi: 10.1109/ICRA40945.2020.9197095.

[36] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5022–5030, 2019.

[37] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3109–3118, 2015. doi: 10.1109/CVPR.2015.7298930.

[38] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. doi: 10.1109/WACV.2014.6836101.

[39] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017. URL http://arxiv.org/abs/1711.00199.

[40] Y. Xiao, Y. Du, and R. Marlet. Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning. In *2021 International Conference on 3D Vision (3DV)*, pages 74–84, Los Alamitos, CA, USA, dec 2021. IEEE Computer Society. doi: 10.1109/3DV53792.2021.00018. URL https://doi.ieeecomputersociety.org/10.1109/3DV53792.2021.00018.

[41] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3D objects. In *British Machine Vision Conference (BMVC)*, 2019.

[42] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 126–142, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58568-6.

[43] Sergey Zakharov, Wadim Kehl, Benjamin Planche, Andreas Hutter, and Slobodan Ilic. 3d object instance recognition and pose estimation using triplet loss with dynamic margin. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017. doi: 10.1109/iros.2017.8202207. URL http://dx.doi.org/10.1109/IROS.2017.8202207.

[44] Sergey Zakharov, Ivan S. Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1941–1950, 2019.

[45] Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. *European Conference on Computer Vision (ECCV)*, 2018.