Pose Guided RGBD Feature Learning for 3D Object Pose Estimation

Vassileios Balntas[†], Andreas Doumanoglou[*], Caner Sahin[*], Juil Sock[*], Rigas Kouskouridas[‡], Tae-Kyun Kim[*]
[†]University of Oxford, UK   [‡]Scape Technologies, UK   [*]Imperial College London, UK

balntas@robots.ox.ac.uk   {a.doumanoglou12,c.sahin14,ju-il.sock08,tk.kim}@ic.ac.uk   rigas@scape.io

Abstract

In this paper we examine the effects of using object poses as guidance to learning robust features for 3D object pose estimation. Previous works have focused on learning feature embeddings based on metric learning with triplet comparisons and rely only on the qualitative distinction of similar and dissimilar pose labels. In contrast, we consider the exact pose differences between the training samples, and aim to learn embeddings such that the distances in the pose label space are proportional to the distances in the feature space. However, since it is less desirable to force the pose-feature correlation when objects are symmetric, we discuss the use of weights that reflect object symmetry when measuring the pose distances. Furthermore, end-to-end pose regression is investigated and is shown to further boost the discriminative power of feature learning, improving pose recognition accuracies. Experimental results show that the features that are learnt guided by poses, are significantly more discriminative than the ones learned in the traditional way, outperforming state-of-the-art works. Finally, we measure the generalisation capacity of pose guided feature learning in previously unseen scenes containing objects under different occlusion levels, and we show that it adapts well to novel tasks.

1. Introduction

Detecting objects and estimating their 3D pose is a very challenging task due to the fact that severe occlusions, background clutter and large scale changes dramatically affect the performance of any contemporary solution. State of the art methods make use of Hough Forests for casting patch votes in the 3D space [27, 7] or train CNNs to either perform classification into the quantized 3D pose space [13] or regress the object position [14] from local patches.

Another approach to the 3D object pose estimation problem involves transforming the initial problem into a nearest

---

Work done while VB and RK were at Imperial College London.

neighbour matching one, where extracted feature descriptors are matched with a set of templates via nearest neighbour search [9]. End-to-end deep networks for feature-based nearest neighbour matching entail training a classification network with a classifier layer which is subsequently removed, while the penultimate layer serve as a feature descriptor [24]. Direct feature learning for discrete object classes with deep neural networks [26, 12] demonstrated successful results by using siamese and triplet networks optimised for discriminative embeddings. The latter are learned in a way that ensures that features extracted from samples belonging to the same class are close in the learned embedding space, and samples from different classes are further apart. Wohlhart and Lepetit [29] adapted this framework to the problem of learning feature descriptors for 3D object pose estimation, by sampling the qualitative relation of pose similarity, and forming triplets consisting of similar and dissimilar poses.

It is apparent that moving from the continuous space of 3D object poses to the qualitative one of similar and dissimilar pose pairs, leads to inevitable information loss. Towards this end in this paper we are interested in creating a feature learning framework that directly uses the object pose in the optimization process. Our key idea is that by using the pose labels in the feature learning layer, we can devise a learning procedure that has inherit knowledge of the final goal (i.e. 3D pose estimation), thus allowing for a switch from a qualitative optimization (similar and dissimilar poses), to a quantitative one (directly computed distance in the pose space). In the proposed learning framework, the pose-feature correlation is established with the adjusted distances in the pose space. Direct 3D pose regression seems challenging due to ambiguities in appearance/pose space, the continuous nature of the multi-dimensional output space and the discrepancy between synthetic data used for training and real data used for testing. However, training an end-to-end pose regression network can still facilitate feature learning. Similar to [24], while evaluating our system's performance, we remove the regression layer and use the feature layer for nearest neighbour matching. The regression term along with the pose-guided feature one further improves the
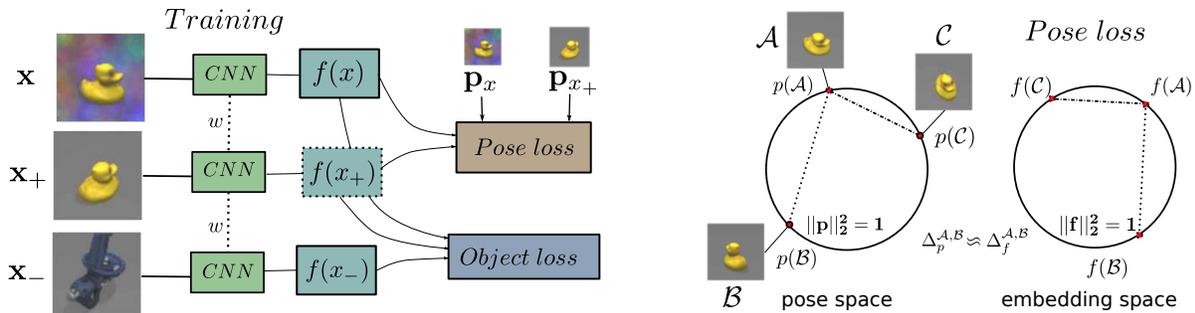
Figure 1. (**left**) The overview of our method. We use a triplet network to learn embeddings suitable for both object recognition and exact pose retrieval (**right**) Illustration of the proposed pose loss. Intuitively, our method aims to train a CNN in such a way that the distance in the resulting $D - dimensional$ embedding space is analogous to the pose differences.

discrimination power of features for pose recognition. In summary, this paper offers the following contributions:

Pose-guided feature learning: we propose a new optimisation term for learning 3D pose feature descriptors which enforces direct relationship between the learned features and pose label differences. The effect of symmetry can also be embedded into the learning procedure. Experiments provide evidence that the pose-guided learned features are significantly more discriminative than the ones learned with more traditional metric learning based techniques [12, 29].

Regression-guided features: we plug in a regression term into the proposed framework and show that this further boost the performance levels of our system, especially when using real training data.

Experiments on unseen objects and occlusions: lastly, we investigate the performance of the feature learning methods when generalising from similar to dissimilar poses, and previously unseen objects under different level of occlusions.

The remainder of the paper is organized as follows. In Section 2 we briefly describe the previous state of the art methods for learning 3D pose descriptors, and explore in more detail the most relevant work to ours [29]. We discuss its limitations, motivate our method, and introduce our proposed approach in Section 3. In Section 4 we present an evaluation of our method compared to some state-of-the-art methods. Finally, in Section 5 we conclude with final remarks.

2. Related Work

Recognizing and detecting objects along with estimating their 3D pose has received a lot of attention in the recent literature. Early works made use of point-clouds to facilitate Point-to-Point matching [8, 19], while the advent of low-cost depth sensors [9, 18] provided additional data in favour of textureless objects. Hinterstoisser et al. [9] designed a powerful holistic template matching method

(LINEMOD) based on RGB-D data which however suffers in cases of occlusions. Inspired by this shortcoming, Tejani et al. [27] integrated LINEMOD into Hough Forests to tackle the problem of occlusions and clutter. The work of Brachmann et al. [4] along with its recent extension to RGB-only images [5] employ a new representation framework that jointly maps object coordinates and class labels. Hodan et al. [11] present a method that tackles the complexity of sliding window approaches, while fine 3D pose estimation is performed via a stochastic, population-based optimization scheme. In turn, in [25] exemplar SVMs are slid in the 3D space to perform object pose classification based on depth images.

Deep learning has only recently found application to the object pose estimation problem. Doumanoglou et al. [7] suggested using a network of stacked sparse autoencoders to automatically learn features in an unsupervised manner that are fed to a Hough Forest for object pose recovery and next-best-view estimation. Kehl et al [14], used regression autoencoders to learn patch representations for subsequent voting in the pose space. In [13] Johns et al. employ a CNN-based end-to-end learning framework for classification of object poses in the 3D space and next-best-view prediction. In turn, in [6] a CNN was used to learn projections of 3D control points for acute object tracking, while in [16] a CNN is utilized in a probabilistic framework to perform analysis-by-synthesis as a final refinement step for the object pose estimation. In the feature learning framework of Wolhart and Lepetit [29], 3D pose estimation is performed by a scalable Nearest Neighbour method on discriminative descriptors learned by a CNN. While their work, which we review below, is the most relevant to ours, the important difference is that in this paper we explore the direct usage of pose labels in the feature learning process.

Learning pose feature descriptors with qualitative similarity constraints. Wohlhart and Lepetit [29] use a metric learning approach that has been shown to be very powerful for learning robust features in several problems such
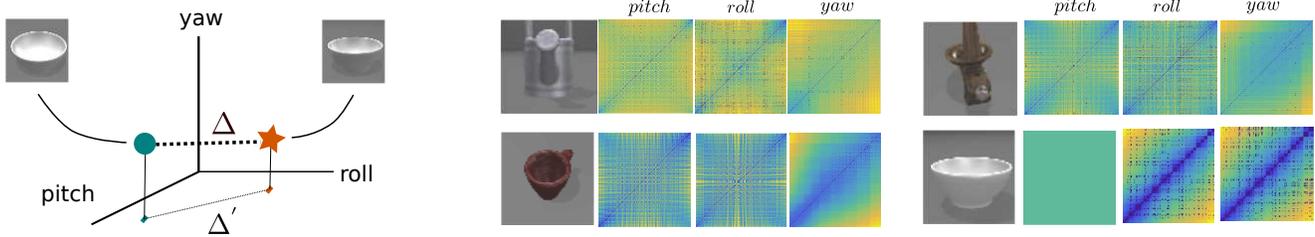
Figure 2. (**left**) Illustration of the case of $s_{yaw} = 0$ for a simple case of a fully rotationally invariant object for the yaw axis. (**right**) Illustration of the ϕ term (Equation 5) for all pose pairs individually per 3D axis. Note that the more symmetric objects present lower values than the non-symmetric ones.

as face recognition and local feature descriptors [22, 2] as well as 3D object pose. The method is based on a training dataset where each item is of the form $\{\mathbf{x}, y, \mathbf{p}\}$, with $\mathbf{x} \in \mathbb{R}^{W \times H \times 4}$ being the RGBD image, y the object class label, and $\mathbf{p}$ the 3D pose of the object. In addition, a set of synthetically generated (i.e. rendered views of the 3D object model) templates for each object is stored, in order to be used as reference frames for computing and matching descriptors based on a K-nearest neighbour search method.

A convolutional neural network is used to train feature embeddings based on the intuition that feature descriptors for similar poses of the same object should be closer in the learnt embedding space $\{\mathbf{x}, \mathbf{x}_+, \mathbf{x}_-\}$, where $\mathbf{x}$ is an arbitrarily chosen RGBD image of a specific pose, $\mathbf{x}_+$ is an image of the object with a similar pose as $\mathbf{x}$, and finally $\mathbf{x}_-$ contains an image of dissimilar pose compared to $\mathbf{x}$. Formally, the loss function for such a triplet is given as

$$\mathcal{L}_{triplets} = \sum_i \frac{\Delta_+^{(i)}}{\Delta_-^{(i)} + \mu} \qquad (1)$$

where $\Delta_+^{(i)}$ and $\Delta_-^{(i)}$ represent the feature distances of the similar and dissimilar data inside the ith triplet. Intuitively, the final goal of the triplet-based optimization process is to learn features such that the distance in the learned embedding space between the two similar poses is lower than the distance between the two dissimilar poses, within the limits of the margin μ. In addition, a pair-based term $\mathcal{L}_{pairs}$ is also included, in order to minimise distances between identical poses but different viewing conditions (e.g. background variability and illumination changes) with

$$\mathcal{L}_{pairs} = \sum_i \Delta_+^{(i)} \qquad (2)$$

where $\Delta_+^{(i)}$ denotes the feature distance between the data pair. Finally, the above two terms are combined in a common loss:

$$\mathcal{L} = \mathcal{L}_{triplets} + \mathcal{L}_{pairs} + \lambda ||w^T||_2^2 \qquad (3)$$

where w denotes the weights of the network.

A visualisation of the type of triplets and pairs that are used as input can be found in Figure 3 (left). Note that the triplets that are fed to $\mathcal{L}_{triplets}$, consist of both positive (i.e. all patches come from the same object, illustrated with a green border) and negative (i.e. two patches from the same object, and another from a different object, illustrated with a red border). This is done in order to ensure that the learned embeddings can be used for both object recognition and pose estimation. Furthermore, we can observe that the pairs that are optimized by the $\mathcal{L}_{pairs}$ term, consist of two patches with identical pose but varying backgrounds.

Despite the promising results, an important limitation of this work is that the similarity is only used as an indicator function in order to form the triplet and the pair training data. Thus, the final embedding space is built in terms of pairs and triplets of similar and dissimilar patches, which limits the discriminability of the embedding space. Note that such a constraint is unavoidable for methods that learn embeddings only using distinct classes (i.e. class label) [2, 12]. On the contrary, it is arbitrarily enforced in terms of dealing with continuous labels such as 3D poses.

3. Pose guided RGBD feature learning

Our key observation is that the pair based term, which acts on pairs of images extracted from the same object, can be used to optimise exact pose similarities by using the pose as a label. The creation of the embedding space directly using this exact pose label, could lead to a significantly more discriminative embedding space, which is the motivation behind the design of our method. We introduce the *Pose-guided Feature Similarity* term $\mathcal{L}_{pose}$, that aims to associate distances in the learnt embedding space with distances in the pose label space. More formally, for a data pair $\{(\mathbf{x}_1, \mathbf{p}_1), (\mathbf{x}_2, \mathbf{p}_2)\}$ where $\mathbf{x}$ represents the RGBD image and $\mathbf{p}$ the pose, the pair based loss that is to be minimised is given by

$\mathcal{L}_{triplets}$ $\mathcal{L}_{pairs}$ $\mathcal{L}_{object}$ $\mathcal{L}_{pose}$
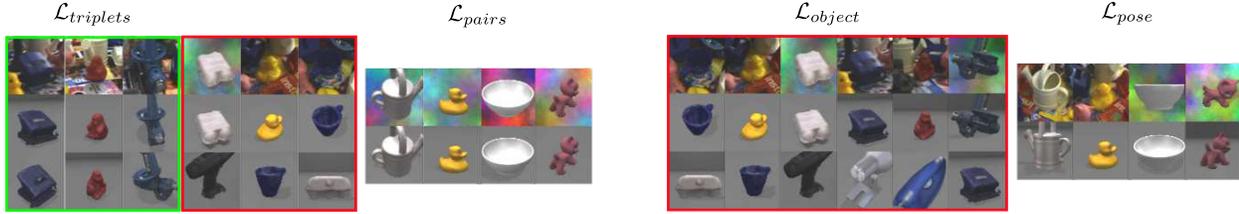
Figure 3. (**left**) Illustration of the training samples that are fed to the optimisation losses $\mathcal{L}_{triplets}$ and $\mathcal{L}_{pairs}$, for method from [29].(**right**) Illustration of the training samples that are fed to the optimisation losses $\mathcal{L}_{object}$ and $\mathcal{L}_{pose}$, for the proposed method.

$$\mathcal{L}_{pose} = \sum_i \left\{ ||f(\mathbf{x}_1^{(i)}) - f(\mathbf{x}_2^{(i)})||_2^2 - \delta(\mathbf{p}_1^{(i)}, \mathbf{p}_2^{(i)}) \right\}^2 \quad (4)$$

where $f(\mathbf{x}) \in \mathbb{R}^D$ is the learnt embedding function, and $\delta(\mathbf{p}_1, \mathbf{p}_2)$ is a function encoding difference between two poses. For a pair that has minimum pose difference of 0, the optimisation process will enforce the distance between the embeddings to go to zero. Thus, in general, $\mathcal{L}_{pose}$ enforces a direct relation between normalised pose and normalised embedding distances. An illustration of this effect is shown in Figure 1 (right). Note that in such a loss, it is important for both the network feature layer, and the poses to be normalised to lie on the unit hyper-circle.

In its most basic form, the function δ can be the euclidean distance between two poses $\delta(\mathbf{p}_1, \mathbf{p}_2) = ||\mathbf{p}_1 - \mathbf{p}_2||_2^2$. However, this might be problematic in the case of embedding learning for heavily symmetric objects. Recent work focuses on the issues of evaluating pose estimation taking into consideration ambiguities arising from symmetric objects [10]. We propose to also incorporate such an idea about dealing with ambiguities into the training process in order to equip the learning proccess with a way to become more robust to symmetry issues.

One needs to consider cases where the object is non-symmetric or symmetric in a non-trivial way. Note that in this case it is desirable for the distance function δ to be learnt according to a probability distribution of symmetry, from different rendered views of the object. This allows dealing with objects that exhibit symmetry only at particular pose combinations, not all around a certain axis. In particular, we adopt an example of such learnt representation of the function δ which is to define it for each pair $(\mathbf{p}_1, \mathbf{p}_2)$ of poses as the difference in the depth channel of the rendered views of the model of the object, inspired by [10]. This is based on the idea that if two poses $(\mathbf{p}_1, \mathbf{p}_2)$ result in a two rendered depth images of the object $(\mathbf{x}_1, \mathbf{x}_2)$ that are very similar (i.e. $||\mathbf{x}_1 - \mathbf{x}_2|| \approx 0$), then their difference in the learnt embedding space should also be low, regardless of the magnitude of $\langle \mathbf{p}_1, \mathbf{p}_2 \rangle$. More formally, we weight the loss contribution of each ith sample in Equation 4 with the respective symmetric term

$$\phi(\mathbf{p}_1^{(i)}, \mathbf{p}_2^{(i)}) = ||s(\mathbf{p}_1^{(i)}) - s(\mathbf{p}_2^{(i)})||_2^2 \quad (5)$$

where $s(\mathbf{p}_1)$ and $s(\mathbf{p}_2)$ represent the renderd depth images at the poses $\mathbf{p}_1$ and $\mathbf{p}_2$. Practically, the values of ϕ can be stored in a look-up table, for all the possible training combinations of pose pairs. Note that the values in this lookup table, need to be normalised to exhibit values in the range $[0, 1]$. Since exact weighting might be undesirable, this term can also be thresholded such that that poses with similar renderings do not contribute at all to the loss.

In Figure 2 (left) we illustrate the case of a rotationally invariant object across a single axis and in Figure 2 (right) we show results for the phi term computed from the depth differences of the rendered views. For illustration purposes, we plot the generated 2D pose-pairs distributions separately for yaw, pitch and roll. Note that more symmetric objects such as the cup or the bowl, exhibit distributions with a higher percentage of lower distances (blue colour) in general. For the non-symmetrical objects, low values are expected only around the diagonals, since any pose change, will result in slight differences in the rendered view.

Triplet based object recognition loss. The pose-feature similarity term that was introduced, cannot be used by itself, since there is also a need for a loss that deals with the embeddings from different objects. This is to enforce embeddings that are computed from the same object, but arbitrary poses, to have lower distances compared to embeddings from different objects, irrespective of the pose differences. Since we cannot optimise such an objective in terms of pairs, we also use a triplet ranking loss similar to Equation 1, and a set of triplets of the form $\{\mathbf{x}, \mathbf{x}_+, \mathbf{x}_-\}$. A notable difference is that in our triplet term, there are only negative triplets (i.e. $\mathbf{x}, \mathbf{x}_+$ are randomly sampled patches from one object, and $\mathbf{x}_-$ from a different one), something that is illustrated in Figure 1 (right). Thus, we term our triplet loss $\mathcal{L}_{object}$. The combined loss to be optimised is

$$\mathcal{L}' = \mathcal{L}_{pose} + \mathcal{L}_{object} + \lambda ||w^T||_2^2 \quad (6)$$

Intuitively, such a combined multi-task loss is better equipped to deal with both the recognition and the pose estimation part of the problem, since they are dealt with two separate terms. On the other hand, the loss from Equation 3,
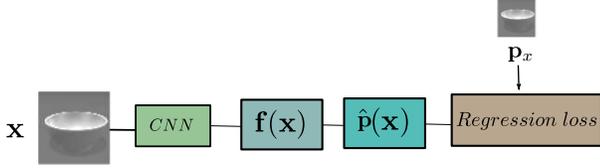
Figure 4. Illustration of the end-to-end feature learning and pose regression network.

jointly optimises both the object recognition term and the intra-object pose related embedding learning by utilising the triplet loss from Equation 1.

Direct pose regression. Several recent works report good results by using direct regression objectives in order to infer the pose of the objects or cameras [14, 15, 17]. Inspired by this, in addition to the feature learning terms, we investigate a method that directly regresses the object pose $\hat{\mathbf{p}}$ given an input $\mathbf{x}$. The loss to be minimised is a standard RMS error over all samples

$$\mathcal{L}_{reg} = \sum_i \phi(\hat{\mathbf{p}}^{(i)}, \mathbf{p}^{(i)}) \cdot ||\hat{\mathbf{p}}^{(i)} - \mathbf{p}^{(i)}||_2^2 \qquad (7)$$

where the function ϕ is defined in Equation 5, and $\hat{\mathbf{p}}$ and $\mathbf{p}$ denote the regressed and the ground truth pose respectively. Note that in order for the ϕ term to be used, the pose space needs to be quantized [17]. This regression loss can be added in the collection of losses from Equation 6, with a suitable scaling parameter to alleviate any issues arising from different absolute numerical values.

4. Experimental results

In this section, we present experimental results, which show that significant improvements can expected be using our pose-guided feature terms over [29]. First, we present results by following the same experimental protocol of [29] in Section 4.2. In addition, we perform experiments using only real training data to understand how well the method generalises to unseen objects, and handle occlusions, in Section 4.3 and Section 4.4 respectively. Note that the results presented below are mainly with the loss collection of Equation 6, and the regression loss is added only when explicitly indicated as such. For the results in Section 4.2 and Section 4.3, we use the LINEMOD dataset, introduced in [9], as modified by the authors of [29]. For Section 4.4, we use an novel dataset of objects grasped by a human hand, collected by us.

4.1. Implementation details

Our convolutional network architecture can be described as follows $\{Input(4, W, H) - Conv(16, 8, 8) - Conv(7, 5, 5) - FC(256) - FC(32)\}$, where $Conv(N, M, M)$ represents a convolutional layer with

N filters of size $M \times M$, and $FC(D)$ a fully connected layer with D outputs. Note that this is identical to the architecture of [29], for a fair comparison. The final feature layer $FC(D)$ can be of a variable length, which allows a trade-off between the feature extraction size and performance. The experiments from [29] showed diminishing returns for $D > 32$. Thus, in all our experiments we fix $D = 32$. We use $ReLU$ as the non-linearity in all our convolutional and fully connected layers, together with max-pooling. In the end-to-end regression network, we use $Tanh$ as the final non-linearity. For training the network we use the stochastic gradient method [3], with 0.9 momentum and initial learning rate of 0.01. We also decay the learning rate in each epoch, to avoid oscillations around local minima.

Since the authors of [29] do not provide pre-trained networks, we use the code they provide with the default parameters described, in order to repeat the learning process. Note that in all our experiments, we use descriptors learnt on RGBD data, since this has been shown to provide a slight improvement over both the individual depth and colour channels [21, 29, 20].

4.2. Training with synthetic data

In this section, we follow the experimental protocol of [29], i.e. we train our network with rendered views of the objects, and we include a small portion of real images in the train data, in order to force the network to learn representation invariant to the background. We use the original training configuration samples as provided [1]. The interested reader can refer to [29] for more details.

Training samples. In Figure 3 (right), we show examples of the input given to each optimisation loss $\mathcal{L}_{object}$ and $\mathcal{L}_{pose}$. Note that compared with the inputs that were used in [29], we see that the triplets for the $\mathcal{L}_{object}$ only contain the negative samples where the third image comes from a different object. This allows the triplet term to more focus on distinguishing objects, delegating the pose related optimisations to the pose term, which acts on the pairs.

Pose versus feature similarity. In Figure 5 we plot for each object, the 2D histogram that indicates correlations between pose differences and descriptor distances. To compute this, we collect all items of the real test set together with all the rendered templates, and we compare all pairwise distances both in the pose, and in the learnt embedding space. As noted by [29], the ideal descriptor should have high values only in the diagonal of this plot, since significant correlation between small angles and large distances increases the probability of missing a correct template during NN matching, and high correlation between large angles and small distances leads to incorrect matches.

---

[1]https://cvarlab.icg.tugraz.at/members/wohlhart.php
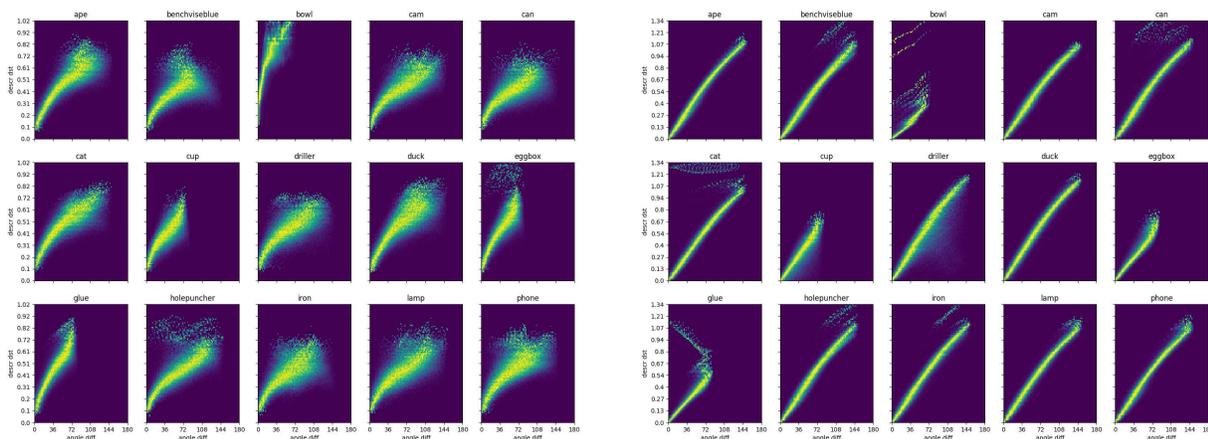
Figure 5. Illustration of the correlations between pose differences and learnt embedding distances, between real test images and rendered templates for the 15 objects of the LINEMOD dataset [9]. (**left**) Learning with qualitative similarity constraints [29]. (**right**) Learning with our pose-guided feature learning. We observe that the distances in the feature space are significantly more correlated with the differences in the pose space.

Note that using our learnt embedding space (right) leads to much more robust representations. On the other hand, the features from [29] (left) exhibit much more expanded distributions, which indicates that items with similar poses might have representations that significantly vary. This is a direct effect of optimising the descriptors using only qualitative pose labelling (similar or dissimilar) since even representations that are even slightly compliant with the ordering inside a triplet yield zero loss functions. On the contrary, due to the effect of Equation 4, there is a direct and linear relation between feature and pose similarity.

K-nearest neighbour pose recognition accuracy. Following the protocol of [29], in Figure 6 we plot the pose recognition accuracy ($y - axis$) across different thresholds of acceptable pose difference in terms of maximum error ($x - axis$). The accuracy at pose threshold t is defined as the percentage of test images for which the best template matching angle error is below t. The value of K indicates the number of nearest neighbours that are used for template pose retrieval and selection of the best error.

Note that the learnt representations using our method, outperform the previous work by a large margin, especially in the high accuracy area (5°). In terms of matching with only a single NN, we can examine a performance boost in the high accuracy 5° area from 54% to 72%. This is a significant improvement, since it allows very high accuracy results for a large majority of the test cases with only a single nearest neighbour, which leads to much improved computational efficiency. In general, our method performs as good as the prior state-of-the-art, using only one nearest neighbour instead of two.
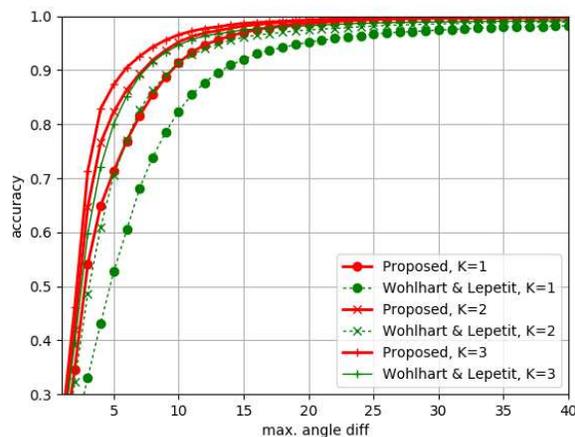
Figure 6. Pose recognition accuracy for different levels of maximum accepted pose difference. K denotes the number of nearest neighbours. The proposed method significantly outperforms the state of the art RGBD deep pose descriptor.

Table 1. Object recognition rate when the pose of the retrieved object is ignored.

LINEMOD	Wohlhart [29]	Ours %
83.70	99.31	99.98

Object recognition rate. In Table 1, we show the object recognition rates for different methods. The recognition rate measures the percentage of cases in which the nearest retrieved template was from the same object class, ignoring the pose difference. We can observe that there is a clear

boost using our method, likely because the proposed triplet loss $\mathcal{L}_{object}$ solely focuses on the recognition aspect of the learnt embedding space.

Evaluation of different optimisation terms. In Table 2, we show the performance of our methods using different combinations of the terms. The pose estimation accuracy is measured in terms of percentage of test images that were correctly matched to their closest template. The proposed pose-guided feature term significantly boosts the accuracy of [29], and the regression term when it is used in addition to the pose-guided feature term tops up the accuracy meaningfully. We can also see that incorporating the symmetry constraints into the learning function improves the results, especially for symmetric objects. It has to be noted that the authors of [29] employ complex bootstrapping and fine-tuning schemes, in order to aid the learning process. Since the authors did not report results without such schemes, we use their implementation to report the performance without bootstrapping. On the contrary, our method does not employ such complicated schemes, something that further indicates the superiority of the pose guided feature learning terms. It has to be noted that the use of bootstrapping and fine tuning increases the training time, and thus our method is also characterised by significantly decreased training computational burden. In addition to [29], we show results for two further baseline methods LINEMOD [9] and LCHF [27]. Note that for these two baselines, the evaluation is done without any aspect of object recognition, since the templates to be matched are only limited to the ones from the same class. This is due to the fact that these pose estimators are individually trained per object and include no recognition aspect. Note that LCHF [27] performs similarly with the deep descriptors from [29], even though the deep feature descriptor is learnt simultaneously for all the objects, something that indicates the great learning capacity of CNNs.

4.3. Training with real data

In this section, we present a different experimental protocol, in order to examine how similar and dissimilar training/testing sets affect the pose recognition performance of the deep learning baselines. Since it is hard to qualify/quantify the gap between the synthetic rendered images and real images (Section 4.2), we use real training images only, which are chosen via a set of different criteria leading to the protocols 1, 2 and 3 described below. For our experiments, we use the LINEMOD dataset [9] and use as an evaluation metric the average pose difference between the query and the top $K = 5$ retrieved nearest neighbours [28, 23].

Protocol 1 - Random train/test split. We split the real imaages of the LINEMOD dataset into train (30%) and test (70%) samples randomly. This guarantees that similar

Figure 7. Results for training and testing with real data. First column indicates the query image, subsequent columns show the retrieved 5 nearest neighbours using our pose-guided features.

Figure 8. Results for training and testing on real data. For description of the three protocols please refer to the text.

poses are included in both train and test sets, thus protocol examines how well a feature descriptor performs in terms of identifying poses that are very close to the ones already seen. In Figure 8 (right) we present the mean angle errors of different methods across the test data. It is clear that our proposed method of combining the pose-guided feature term and regression term leads to improved results.

Protocol 2 - Dissimilar train/test poses. We split the data into train and test by combining sets of similar poses into the same category. With this protocol, poses that are very similar will be either on the train data, or the test data, but could not be in both. This allows us to investigate how well different methods adapt to unseen poses. In Figure 8 (left), we present results for this evaluation protocol. Our proposed method of combining all optimisation terms leads to the best results. One important observation is that the regression term does not perform as well as in Protocol 1, something that indicates that the regression term can boost the performance only when it has seen similar poses in the training set.

Protocol 3 - Unseen objects. While protocols 1 and 2 aim to examine the performance degradation from training and testing on similar poses to dissimilar poses, it is not very clear how such descriptors perform when evaluated in unseen objects. We form our training and test data in such a way that for every test object the training process is done on triplets that do not contain any samples from this object. In Figure 8 (middle), we present results for the *ape*

Table 2. Performance in terms of % of test images that were correctly matched to their nearest template in the pose space. Note that the proposed methods do not use bootstrapping, and that both the proposed methods and [29] do both object recognition as well as pose estimation. On the contrary [9] and [27] learn individual pose estimators for each object.

	ape	benchvise	cam	can	cat	driller	duck	eggbox	glue	holepunch.	iron	lamp	phone	*average*
$LINEMOD$ [9]	32.01	28.64	-	24.45	33.74	21.22	30.16	-	-	32.01	26.82	22.81	23.25	27.51
$LCHF$ [27]	35.51	46.92	-	35.25	35.51	43.38	37.50	-	-	43.33	38.80	36.59	32.01	38.48
$\mathcal{L}_{triplets} + \mathcal{L}_{pairs}$ [29] (no bootstrapping)	18.71	20.12	21.99	18.32	20.06	16.41	15.09	15.85	14.14	23.23	20.19	20.30	17.60	18.61
$\mathcal{L}_{triplets} + \mathcal{L}_{pairs}$ [29]	30.99	35.45	37.44	31.37	26.95	27.66	27.00	23.91	25.14	32.72	28.92	32.69	34.19	30.34
$\mathcal{L}_{object} + \mathcal{L}_{pose}\, \delta(p_1^{(k)},p_2^{(k)})$	49.51	51.43	58.03	48.92	55.37	42.58	47.41	47.34	49.78	53.29	55.63	52.84	51.19	51.02
$\mathcal{L}_{object} + \mathcal{L}_{pose}\, \phi(\boldsymbol{p}_1,\boldsymbol{p}_2)$	51.61	51.70	57.21	49.84	56.66	45.16	50.51	58.20	53.14	57.51	58.32	57.76	58.66	54.32
$\mathcal{L}_{object} + \mathcal{L}_{pose}\, \phi(\boldsymbol{p}_1,\boldsymbol{p}_2) + \mathcal{L}_{reg}$	53.26	52.89	59.73	51.59	57.46	46.71	51.58	59.29	55.51	55.84	60.31	57.91	61.35	55.64

sequence. Results for other objects are similar. It is worth noting that the addition of pose regression hurts the performance, which is in contrast with the two previous protocols. This indicates that directly optimising the pose regression term, can be problematic in terms of generalization to novel objects. The best performing method is the pose-guided feature learning, which shows that a combination of discriminative feature learning and enforcing pose-feature similarity, can be beneficial even for unseen objects.

4.4. Experiments with occlusions

The real images in the LINEMOD dataset in the previous sections has been used primarily for testing in literature, and synthetically rendered images were used for training. In this section, we use a novel dataset called Hand-Object dataset, which consists of fully annotated 11 objects, and is characterized by occlusions of significant parts of the object by humans that are grasping the object. We obtain the object pose annotations either by fitting 3D models of the objects to a scene with the help of a tracker (the belt object) or a magnetic sensor `trakSTAR` that captures the 3D locations and orientations of a sensor, that we attach to an object [1] (10 other objects). In Figure 9, we present some samples of our dataset. For each object a human freely manipulates it without any severe hand occlusions, denoted as the *clean* data, and subsequently grasps in such a way that introduces significant occlusions and manipulates the object, denoted as the *occluded* data. We split the videos into 60% train and 40% test images. More details are given in the supplementary material.

We modify the pose-guided feature term to include one sample fully visible, and one obscured as follows $\mathcal{L}_{occ} = \sum_i \left\{ ||f(\mathbf{x}_{occ}^{(i)}) - f(\mathbf{x}_{clean}^{(i)})||_2^2 - ||\mathbf{p}_{occ}^{(i)} - \mathbf{p}_{clean}^{(i)}||_2^2 \right\}^2$. By adding this term to the learning process, we ensure that two images with similar poses, will be projected close in the feature space, despite the fact that parts of the obscured. Table 3 shows the average angle error between the retrieved NN and the ground truth NN with and without the occlusion term on our hand-object occlusion dataset. We can observe that the proposed method leads to impoved results.

Figure 9. (left) Objects from our real captured occlusion dataset (right) Results for the tracker-based occluded dataset. 1st column shows a real RGBD image, 2nd shows the image rendered using the tracking annotation, 3rd shows the rendered object corresponding to the occluded image, 4th shows the nearest neighbour retrieved using our pose-guided features.

Table 3. Occlusion dataset results. The network is able to learn representations that are significantly more invariant to occlusions. Reported is the average angle error from the ground truth.

	w\o $\mathcal{L}_{occ}$	w\ $\mathcal{L}_{occ}$
belt object (tracker labels)	14.3	11.8
avg. of 10 objects (sensor labels)	36.3	27.7

5. Conclusions

We examine the use of object poses as guidance for learning robust features used for 3D object pose estimation. Previous works have focused on learning embeddings based only on the qualitative binary distinction of similar and dissimilar pose labels. In contrast, we consider the exact pose differences between the training samples such that the distances in the pose label space are proportional to the distances in the feature space. Symmetry of objects and pose regression were further investigated to better guide the feature learning process. The proposed methods yielded more discriminative pose features than the ones learned in the traditional way, outperforming the state-of-the-art.

References

[1] Ndi trackstar sensor manual. http://www.ascension-tech.com/products/trakstar-2-drivebay-2/. 8

[2] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016. 3

[3] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*. 2012. 5

[4] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*. 2014. 2

[5] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*. 2016. 2

[6] A. Crivellaro, M. Rad, Y. Verdie, K. Moo Yi, P. Fua, and V. Lepetit. A novel representation of parts for accurate 3d object detection and tracking in monocular images. In *ICCV*, 2015. 2

[7] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *CVPR*. 2016. 1, 2

[8] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010. 2

[9] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012. 1, 2, 5, 6, 7, 8

[10] T. Hodan, J. Matas, and Š. Obdrzálek. On evaluation of 6d object pose estimation. *ECCV Workshops*, 2016. 4

[11] T. Hodan, X. Zabulis, M. Lourakis, S. Obdrzalek, and J. Matas. Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *IROS*, 2015. 2

[12] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition*. 2015. 1, 2, 3

[13] E. Johns, S. Leutenegger, and A. J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, 2016. 1, 2

[14] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *ECCV*. Springer, 2016. 1, 2, 5

[15] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 5

[16] A. Krull, E. Brachmann, F. Michel, M. Ying Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *ICCV*, 2015. 2

[17] F. Massa, R. Marlet, and M. Aubry. Crafting a multi-task cnn for viewpoint estimation. *arXiv preprint arXiv:1609.03894*, 2016. 5

[18] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *ICCV*, 2013. 2

[19] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009. 2

[20] C. Sahin, R. Kouskouridas, and T.-K. Kim. Iterative hough forest with histogram of control points for 6 dof object registration from depth images. In *IROS*, 2016. 5

[21] C. Sahin, R. Kouskouridas, and T.-K. Kim. A learning-based variable size part extraction architecture for 6d object pose recovery in depth. *arXiv preprint arXiv:1701.02166*, 2017. 5

[22] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3

[23] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013. 7

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 1

[25] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *ECCV*. 2014. 2

[26] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014. 1

[27] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *ECCV*. 2014. 1, 2, 7, 8

[28] T. Vatahska, M. Bennewitz, and S. Behnke. Feature-based head pose estimation from images. In *IEEE-RAS Conference on Humanoid Robots*, 2007. 7

[29] P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *CVPR*, 2015. 1, 2, 4, 5, 6, 7, 8