Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation

Markus Oberweger[1] Mahdi Rad[1] Vincent Lepetit[2, 1]

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria
[2] Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, Bordeaux, France
{oberweger,rad,lepetit}@icg.tugraz.at

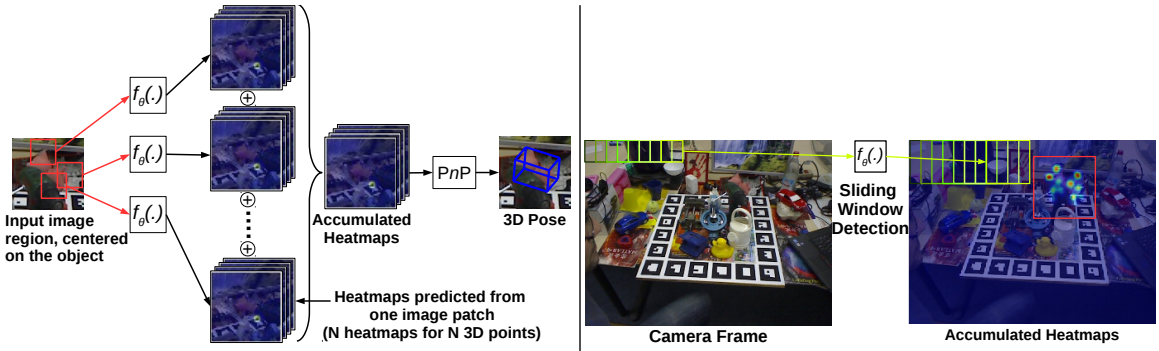
Abstract. We introduce a novel method for robust and accurate 3D object pose estimation from single color images under large occlusions. Following recent approaches [1–3], we first predict the 2D reprojections of 3D points related to the target object and then compute the 3D pose from these correspondences using a geometric method. Unfortunately, as our experiments show, predicting these 2D reprojections using a regular CNN or a Convolutional Pose Machine [4] is very sensitive to partial occlusions, even when these methods are trained with partially occluded examples. Our solution is to predict heatmaps from multiple small patches independently and to accumulate the results to obtain accurate and robust predictions. Training then becomes challenging because patches with similar appearances but different positions on the object correspond to different heatmaps. However, we provide a simple yet effective solution to deal with such ambiguities. We show that our approach outperforms existing methods on two challenging datasets: The Occluded LineMOD dataset, and the YCB-Video dataset, both exhibiting cluttered scenes with highly occluded objects.

Keywords: 3D object pose estimation, heatmaps, occlusions

1 Introduction

3D object pose estimation from images is an old but currently a highly researched topic, mostly due to the advent of Deep Learning based approaches and the possibility of using large datasets for training such methods. 3D object pose estimation from RGB-D already gives compelling results [3, 5–8], and the accuracy of methods that only require RGB images recently made huge progress [1–3, 6–9]. In particular, one way to obtain an accurate pose is to rely on a Deep Network to first predict the 2D projections of some chosen 3D points, and then compute the 3D pose of the object using a PnP method [10]. Such an approach was shown to be more accurate than directly predicting the pose in [1–3], and therefore we follow the same principle in this paper.

However, while Deep Learning methods work well to predict the pose of fully visible objects, they significantly suffer from occlusions, which are very common in practice: Parts of the target object can be hidden by other objects, or by a hand interacting with the object. A common *ad hoc* solution is to train the



(a) 3D pose estimation (b) Detection

Fig. 1: Overview of our method. (a) Given an image region centered on the target object, we sample image patches from which we predict heatmaps for the 2D projections of the corners of the object's 3D bounding box. This prediction is done by a Deep Network $f_\theta(\cdot)$. We aggregate the heatmaps and extract the global maxima for each heatmap, from which we compute the 3D object pose using a PnP algorithm. *We show that $f_\theta(\cdot)$ can be trained simply and efficiently despite the ambiguities that may arise when using small patches as input.* (b) To obtain the image region centered on the object, we apply the predictor in a sliding window fashion and accumulate the heatmaps for the full camera frame. We keep the image region with the largest values after accumulation.

Fig. 2: Our method can predict the 3D pose of objects even under heavy occlusions from color images. From left to right: The Cat, Duck, Can, Holepuncher, Driller objects from the Occluded LineMOD dataset [11]. The green bounding boxes correspond to the ground truth poses, the blue bounding boxes to our estimated poses.

network with occluded objects in the training data. As our experiments presented in this paper show, large occlusions and unknown occluders still decrease the accuracy of the predicted pose.

Instead of using the entire image of the target object as input to the network, and as illustrated in Fig. 1, we consider image patches, since at least some of them are not corrupted by the occluder. Using an image patch as input, we learn to predict heatmaps over the 2D projections of 3D points related to the target object. By combining the heatmaps predicted from many patches, we obtain an accurate 3D pose even if some patches actually lie on the occluder or the background instead of the object. We show results of our method in Fig. 2.

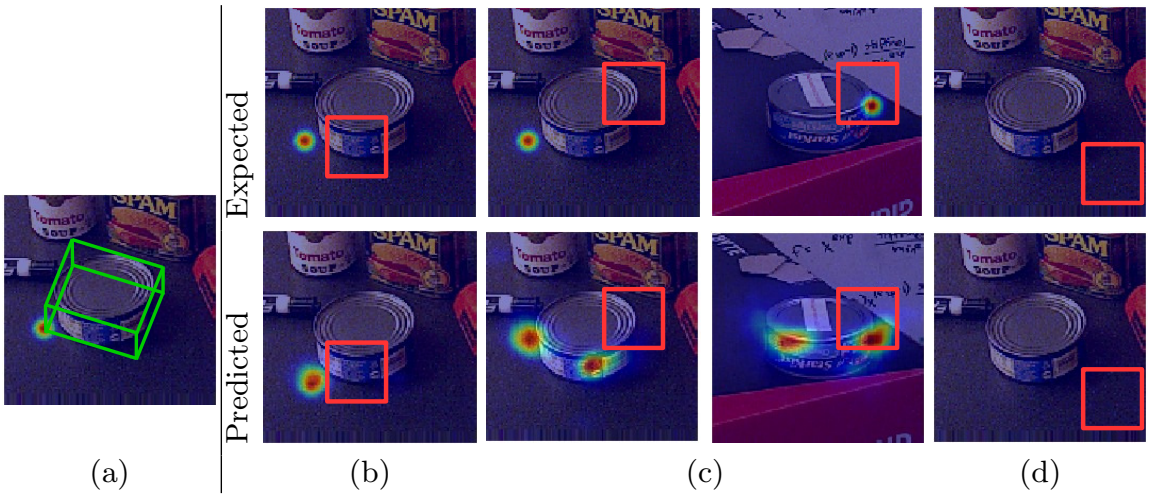


Fig. 3: Predicting heatmaps from image patches. In this example, we consider predicting the projection of the 3D corner highlighted in (a) for the *tuna fish can* object of the YCB-Video dataset [7]. The red boxes show the input patch of the predicted heatmap. (b) shows a patch from which the projection can be predicted unambiguously. (c) shows two patches that lie on two different positions of the can (notice that the can is flipped and rotated between the two images) while having a similar appearance. In presence of such patches, it is only possible to predict a distribution over the possible locations for the projection. (d) shows a patch on the background, from which we predict a uniform heatmap as it does not bring information. See text for details.

When moving to an image patch as input, the prediction becomes multi-modal. This is shown in Fig. 3: Some patches may lie on different parts of the target object but look similar. These patches are ambiguous, as they can correspond to different predictions. In such case, we would like to predict heatmaps with multiple local maxima, one for each possible prediction. The main challenge is that the ambiguities are difficult to identify: This would require to identify the patches that have similar appearances, over all the possible viewpoints and positions on the object.

[12] faced a similar problem in the context of 2D object detection when aiming to localize semantic parts from feature vectors of a convolutional layer computed by a CNN. As we discuss in Section 2, the method they propose is complex both for training and inference, and also inaccurate. The solution we propose is much simpler yet efficient: We train a network to predict heatmaps corresponding to a single solution for training image patches using a least-squares loss function. Thanks to the properties of the least-squares loss, this makes the network naturally predict the *average* of the possible heatmap solutions for a given patch. This is exactly what we want, because it is the best information we can get from a single patch even if it remains ambiguous. We then follow an ensemble approach and take the average of the heatmaps predicted for many patches as this resolves the ambiguities that arise with individual patches. We finally extract the global maximum from this average as the final 2D location.

Our main contribution is therefore a simple method that accurately predicts the 3D pose of an object under partial occlusion. We also considered Transfer Learning to exploit additional synthetic training data and improve performances. However, as we show, if the input to a network contains an occluder, the occlusion significantly influences the network output even when the network is trained with occlusion examples, and simply adding more training data does not help. In our case, some of the input patches used by our method will not contain occluders and Transfer Learning becomes useful. In practice, we use the Feature Mapping of [13], which maps the image features extracted from real images to corresponding image features for synthetic images. This step is not needed for our method to outperform the state-of-the-art, but provides an additional performance boost. As can be seen in Fig. 2, we can predict a very accurate pose even under drastic occlusions.

In the remainder of this paper, we first discuss related work, then present our approach, and finally evaluate it and compare it to the state-of-the-art methods on the Occluded LineMOD [11] and the YCB-Video [7] datasets.

2 Related Work

The literature on 3D object pose estimation is extremely large. After edge-based [14] and keypoint-based methods [15], Machine Learning and Deep Learning have become popular over the last years also for this problem [1–3, 6–9, 16]. Here, we will mostly focus on recent works based on RGB images. In the Evaluation section, we compare to the recent [1, 3, 6–8].

[8, 9] proposed a cascade of modules, where the first module localizes the target objects, and the second module regresses the object surface coordinates. They then predict the object pose through hypotheses sampling using a pre-emptive RANSAC [10]. However, surface coordinates are not adapted to deal with an axis of symmetry. [1] also first detects the target object, then predicts the 2D projections of the corners of the object's 3D bounding boxes, and finally estimates the 3D object pose from their 3D correspondences using a PnP algorithm. [3] integrated this idea into a recent object detector [17] to predict 2D projections of the corners of the 3D bounding boxes, instead of a 2D bounding box. Similarly, [2] predicts 2D keypoints in the form of a set of heatmaps as we do. However, it uses the entire image as input and it thus performs poorly on occlusions. It also requires training images annotated with keypoint locations, while we use virtual 3D points. [18] also relies on 2D keypoint detection. They consider partially occluded objects to infer the 3D object location from these keypoints. However, their inference adopts a complex model fitting and requires the target objects to co-occur in near-regular configuration.

[6] extends the SSD architecture [19] to estimate the objects' 2D locations and 3D rotations. In a next step, they use these predictions together with pre-computed information to estimate the object's 3D pose. However, this requires a refinement step to get an accurate pose, which is influenced by occlusions. [7] segments the objects, and estimates their 3D poses by predicting the trans-

lation and a quaternion for the rotation, refined by ICP. Segmenting objects makes their approach robust to occlusions to some extent, however, it requires a very complex model. [16] considers object parts in order to handle partial occlusion, by predicting a set of 2D-3D correspondences from each of these parts. However, the parts have to be manually picked, and it is not clear which parts can represent objects such as those we evaluate in this paper.

As mentioned in the introduction, our method is related to [12]. In the context of 2D object detection, [12] localizes semantic parts from neighbouring feature vectors using a spatial offset map. The offset maps are accumulated in a training phase. However, they need to identify which feature vectors support a semantic part, and complex statistical measures are used to identify such vectors. Our method is significantly simpler, as the mapping between the input patches and the 2D projections does not have to be established explicitly.

[20] already evaluated CNNs trained on occlusions in the context of 2D object detection and recognition, and proposed to modify training to penalize large spatial filters support. This yields better performance, however, this does not fully cancel the influence of occlusions. Some recent works also explicitly handle occlusions for 3D pose estimation when dealing with 3D or RGB-D data: Like us, [21] relies on a voting scheme to increase robustness to occlusions; [22] first segments and identifies the objects from a RGB-D image. They then perform an extensive randomized search over possible object poses by considering physical simulation of the configuration. [23] combines holistic and local patches for object pose estimation, using a codebook for local patches and applying nearest-neighbour search to find similar poses, in a way similar to [24, 25]. In contrast to these methods, we use only color images.

Our method is also related to ensemble methods, in particular Hough Forests [26], which are based on regression trees. Hough Forests also predict 2D locations from multiple patches, and are multimodal. Multimodal prediction is easy to perform with trees, as the multiple solutions can be stored in the tree leaves. With our method, we aim at combining the ability of Hough Forests for multimodal predictions, and the learning power of Deep Learning. [27] already reformulated a Hough Forest as a CNN by predicting classification and regression for patches of the input image. However, it requires to handle the detection separately and each patch regresses a single vector, which is not multimodal and requires clustering of the predicted vectors. In this paper, we show that multimodal prediction with Deep Networks is in fact easy for our problem.

3 Influence of Occlusions on Deep Networks

In this section, we evaluate how much a partial occlusion influences a Deep Network, be it a standard Convolutional Neural Network (CNN) or a Convolutional Pose Machine (CPM) [4].

For this experiment, depicted in Fig. 4, we use as input to a network an image centered on an object—here, the *Cat* object from the Occluded LineMOD dataset [11]. We then compare the layer activations in the absence of occlusion,

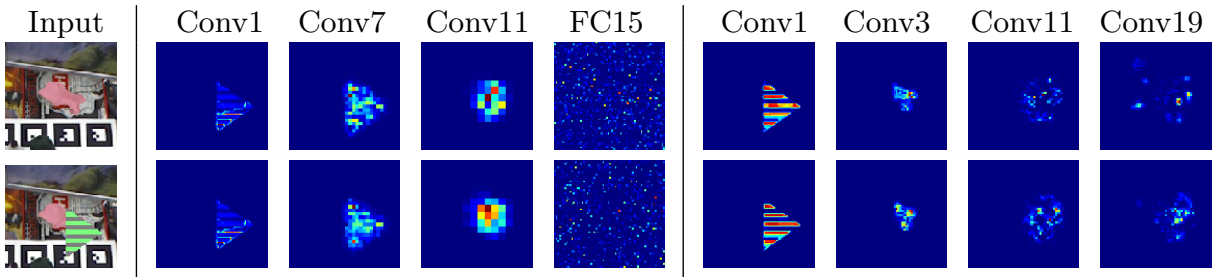


Fig. 4: Effect of occlusions on the feature maps of Deep Networks. (a) Input image without (top) and with (bottom) partial occlusion. (b-Top) Sums of the squared differences between the feature maps with and without occlusions for a CNN trained without occlusions. (b-Bottom) Same when the network is trained with occlusion examples. (c) Same for a Convolutional Pose Machine. The influence of the occlusion increases with the layers' depths, as receptive fields are larger in the deeper layers than in the first layers, even when the method is trained with occlusion examples. For more details we refer to the supplementary material. (Best seen on screen)

and when the object is occluded by an artificial object (here, a striped triangle). We consider two networks: A standard CNN trained to predict the 2D projections of 3D points as a vector [1], and a CPM [4] with 3 stages trained to predict a heatmap for each of the same 2D projections. For the 3D points, we use the corners of the 3D bounding box of the object.

As can be seen in Fig. 4, the occlusion induces changes in the activations of all the layers of both networks. For a standard CNN, the occlusion spreads to almost 25% in the last feature map, and after the first fully-connected layer, more than 25% of all activations are changed. In this case, all the predictions for the 2D projections, occluded or not, are inaccurate. A similar effect can be observed for CPMs: Here, the altered activations are more localized to the occluded region due to the convolutions. In this case, the predictions of the 2D projections are inaccurate when the 3D points are occluded. When the 3D points are not occluded, the predicted projections are sometimes correct, because the influence of the occluder spreads less with a CPM than with a standard CNN.

4 Minimizing the Effect of Occlusions

In this section, we first describe our training procedure given an input image region centered on the object, then the run-time inference of the pose. Finally, we explain how we identify the input image region in practice.

4.1 Training

Datasets for 3D pose estimation typically provide training images annotated with the objects' poses, and the 3D models of the objects. From this data, we

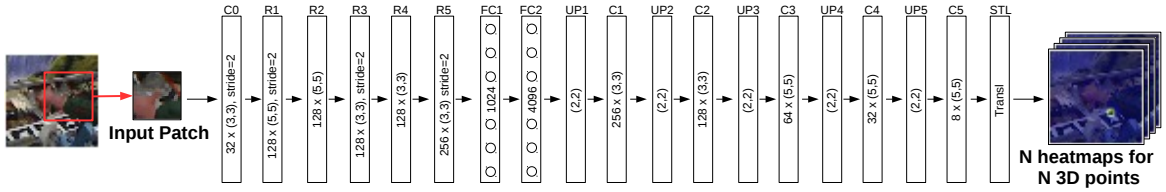


Fig. 5: Network architecture for $f_\theta(\cdot)$. C denotes a convolutional layer with the number of filters and the filter size inscribed, FC a fully-connected layer with the number of neurons, UP an unpooling layer [28], R a residual module [29] with the number of filters and filter size, and STL a Spatial Transformation Layer [30] used for translation. All layers have ReLU activations, and the output of the last layer is linear.

generate our training set $\{(I^{(i)}, \{\mathbf{p}_j^{(i)}\}_j, M^{(i)})\}_i$, where $I^{(i)}$ is the i-th training image, $\mathbf{p}_j^{(i)}$ the 2D projection of the j-th 3D corner, and $M^{(i)}$ the 2D mask of the object in image $I^{(i)}$. This mask can be obtained by projecting the 3D object model into the image using the object's pose.

The Unambiguous Case Let us first ignore the fact that some image patches can be ambiguous, and that the learning problem is actually multimodal. We train a network $f_\theta(\cdot)$ to predict a heatmap for each projection $\mathbf{p}_j$. The architecture we use for this network is shown in Fig. 5. $f_\theta(\cdot)$ takes an input patch of size $32 \times 32px$, and predicts a set of heatmaps of size $128 \times 128px$, and we train it by minimizing:

$$\min_\theta \sum_i \sum_{u,v} \|\mathcal{H}^{(i)} - \text{Transl}(f_\theta(\mathcal{P}(I^{(i)}, u, v)), -u, -v)\|^2 \,, \tag{1}$$

where:

- $\mathcal{P}(I^{(i)}, u, v)$ is an image patch centered on location (u, v) in image $I^{(i)}$;
- $\mathcal{H}^{(i)}$ is the set of expected heatmaps for $\mathcal{P}(I^{(i)}, u, v)$. It contains one heatmap for each 2D projection $\mathbf{p}_j^{(i)}$. We detail how $\mathcal{H}^{(i)}$ is defined below;
- $f_\theta(\mathcal{P})$ returns a set of heatmaps, one for each 2D projection $\mathbf{p}_j^{(i)}$.
- $\text{Transl}(H, -u, -v)$ translates the predicted heatmaps H by $(-u, -v)$. $f_\theta(\cdot)$ learns to predict the heatmaps with respect to the patch center (u, v), and this translation is required to correctly align the predicted heatmaps together. Such a translation can be efficiently implemented using a Spatial Transformation Layer [30], which makes the network trainable end-to-end.

The sum $\sum_{(u,v)}$ is over 2D locations randomly sampled from the image. The heatmaps in $\mathcal{H}^{(i)}$ are defined as a Gaussian distribution with a small standard deviation (we use $\sigma = 4px$ in practice) and centered on the expected 2D projections $\mathbf{p}_j^{(i)}$ when patch $\mathcal{P}(I^{(i)}, u, v)$ overlaps the object mask $M^{(i)}$. The top row of Fig. 3 shows examples of such heatmaps.

When the patch does not overlap the object mask, the heatmaps in $\mathcal{H}^{(i)}$ are defined as a uniform distribution of value $\frac{1}{W \cdot H}$, where $W \times H$ is the heatmaps' resolution, since there is no information in the patch to predict the 2D projections. In addition, we use patches sampled from the ImageNet dataset [31], and train the network to predict uniform heatmaps as well for these patches. Considering these patches (outside the object's mask or from ImageNet) during training allows to correctly handle patches lying on the background or on the occluders, and significantly reduces the number of false positives at run-time.

The Multimodal Case Let us now consider the real problem, where the prediction is multimodal: Two image patches such as the ones shown in Fig. 3(c) can be similar but extracted from different training images and therefore correspond to different expected heatmaps. In other words, in our training set, we can have values for samples i, i', locations (u, v) and (u', v') such that $\mathcal{P}(I^{(i)}, u, v) \approx \mathcal{P}(I^{(i')}, u', v')$ and $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$.

It may seem that in this case, training given by Eq. (1) would fail or needs to be modified. *In fact, Eq.* (1) *remains valid.* This is because we use the least-squares loss function: For image patches with similar appearances that correspond to different possible heatmaps, $f_\theta(\cdot)$ will learn to predict the average of these heatmaps, which is exactly what we want. The bottom row of Fig. 3 shows such heatmaps. At run-time, because we will combine the contribution of multiple image patches, we will be able to resolve the ambiguities.

4.2 Run-Time Inference

At run-time, given an input image I, we extract patches from randomly selected locations from the input image and feed them to the predictor $f_\theta(\cdot)$. To combine the contributions of the different patches, we use a simple ensemble approach and average the predicted heatmaps for each 2D projection. We take the locations of the global maxima after averaging as the final predictions for the 2D projections.

More formally, the final prediction $\widetilde{\mathbf{p}_j}$ for the 2D projection $\mathbf{p}_j$ is the location of the global maximum of $\sum_{u,v} \text{Transl}(f_\theta(\mathcal{P}(I, u, v)), -u, -v)[j]$, the sum of the heatmaps predicted for the j-th projection, translated such that these heatmaps align correctly. The sum $\sum_{u,v}$ is performed over randomly sampled patches. An evaluation of the effect of the number of samples is given in the supplementary material.

To compute the pose, we use a PnP estimation with RANSAC [10] on the correspondences between the corners of the object's 3D bounding box and the $\widetilde{\mathbf{p}_j}$ locations.

4.3 Two-Step Procedure

In practice, we first estimate the 2D location of the object of interest, using the same method as in the previous subsection, but instead of sampling random locations, we apply the network $f_\theta(\cdot)$ in a sliding window fashion and sum the

predicted heatmaps for each projection over the full camera frame, as illustrated in Fig. 1 (b). We extract the local maxima from the summed heatmaps, and keep the image region with the largest number of such local maxima. We then use this image region as the input to the method described in the previous subsection. We use a fixed size for this region as our method is robust to scale changes.

5 Evaluation

In this section, we evaluate our method and compare it to the state-of-the-art. For this, we use two datasets: The Occluded LineMOD dataset [11], and the YCB-Video dataset [7]. Both datasets contain challenging sequences with partially occluded objects and cluttered backgrounds. In the following, we first provide the implementation details, the used evaluation metrics, and then present the results for the two datasets, including an ablative analysis of our method.

5.1 Implementation Details

Training Data: The training data consists of real and synthetic images with annotated 3D poses and object masks, as was done in [7]. For rendering the synthetic objects, we use the models that are provided with the datasets. We crop the objects of interest from the training images and paste them on random backgrounds [32] sampled from ImageNet [31], to achieve invariance to different backgrounds. We augment the dataset with small affine perturbations in HSV color space.

Network Training: The network is optimized using ADAM [33] with default parameters using a minibatch size of 64, a learning rate of 0.001, and 100k iterations. We train one network per object starting from a random initialization.

Symmetric Objects: We adapted the heatmap generation to symmetric objects of the two datasets. For rotationally symmetric objects, *e.g.* cylindrical shapes, we only predict a single position around the rotation axis. For mirror-symmetric objects, we only train on half the range of the symmetry axis similar to [1].

Feature Mapping: Optionally, we apply the Feature Mapping method of [13] to overcome a lack of real training data. We apply the mapping between the FC1 and FC2 layers of Fig. 5. The mapping network uses the same architecture as in [13], but the weight for the feature loss is significantly lower (10^{-5}).

5.2 Evaluation Metrics

We consider the most common metrics. The 2D Reprojection error [9] computes the distances between the projections of the 3D model points when projected using the ground truth pose, and when using the predicted pose. The ADD metric [34] calculates the average distance in 3D between the model points,

after applying the ground truth pose and the predicted pose. For symmetric objects, the 3D distances are calculated between the closest 3D points, denoted as the ADI metric. Below, we refer to these two metrics as AD{D|I} and use the one appropriate to the object. The exact formulas for these metrics are provided in the supplementary material.

5.3 Occluded LineMOD Dataset

The Occluded LineMOD dataset [11] consists of a sequence of 1215 frames, each frame labeled with the 3D poses of 8 objects as well as object masks. The objects show severe occlusions, which makes pose estimation very challenging. The sequences were captured using an RGB-D camera with $640 \times 480px$ images, however, we use *only the color images* for our method and all results reported here.

For training the heatmap predictors, we use the LineMOD dataset [34] that contains the same objects as the Occluded LineMOD dataset. This protocol is commonly used for the dataset [1, 3, 7, 8], since the Occluded LineMOD dataset only contains testing sequences. Fig. 6 shows some qualitative results. We give an extensive quantitative evaluation in the following.

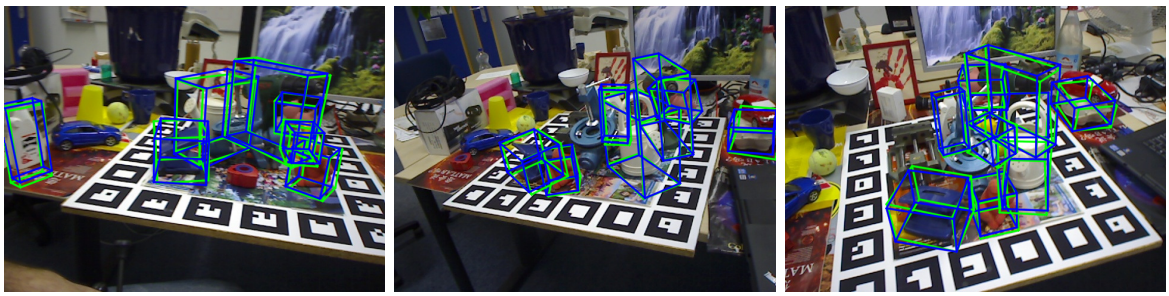


Fig. 6: Some qualitative results on the Occluded LineMOD dataset [11]. We show the 3D bounding boxes of the objects projected to the color image. Ground truth poses are shown in green, our predictions are shown in blue. More results are provided in the supplementary material. (Best viewed on screen)

Quantitative Results Fig. 7 shows the fraction of frames where the 2D Reprojection error is smaller than a given threshold, for each of the 8 objects from the dataset. A larger area under the curve denotes better results. We compare to several recent methods that also work with color images only, namely BB8 [1], PoseCNN [7], Jafari et al. [8], and Tekin et al. [3]. Note that the method of [1] uses ground truth detection, whereas ours does not. Our method performs significantly more accurately on all sequences. Notably, we also provide results for the *Eggbox* object, which, so far, was not considered since it was too difficult to learn for [1, 3, 8].

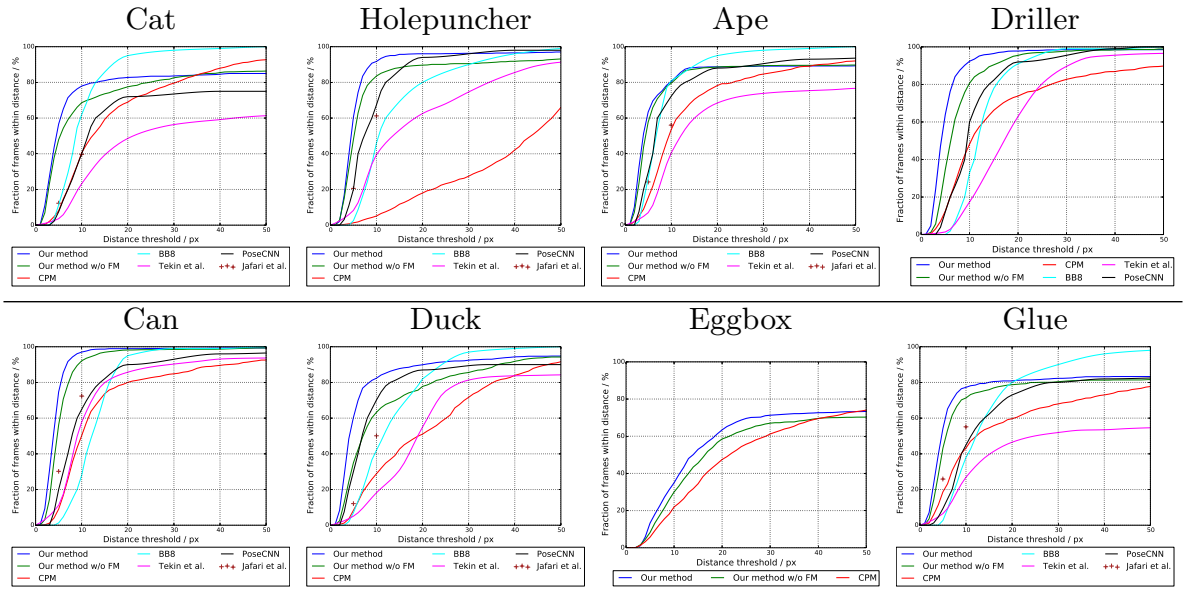


Fig. 7: Evaluation on the Occluded LineMOD dataset [11] using color images only. We plot the fraction of frames for which the 2D Reprojection error is smaller than the threshold on the horizontal axis. Our method significantly outperforms previous work. "w/o FM" denotes without Feature Mapping.

Adding Feature Mapping [13] improves the 2D Reprojection error for a threshold of $5px$ by 14.4% on average. We also tried Feature Mapping for the approach of [1], but it did not improve the results because the occlusions influence the feature maps too much when the network input contains occluders, as already discussed in the introduction.

Further quantitative results are given in Table 1, where we provide the percentage of frames for which the ADD or ADI metric is smaller than 10% of the object diameter, as [7] reported such results on the Occluded LineMOD dataset. This is considered a very challenging metric. We also give the percentage of frames which have a 2D Reprojection error of less than $5px$. Our method significantly outperforms all other methods on these metrics by a large margin.

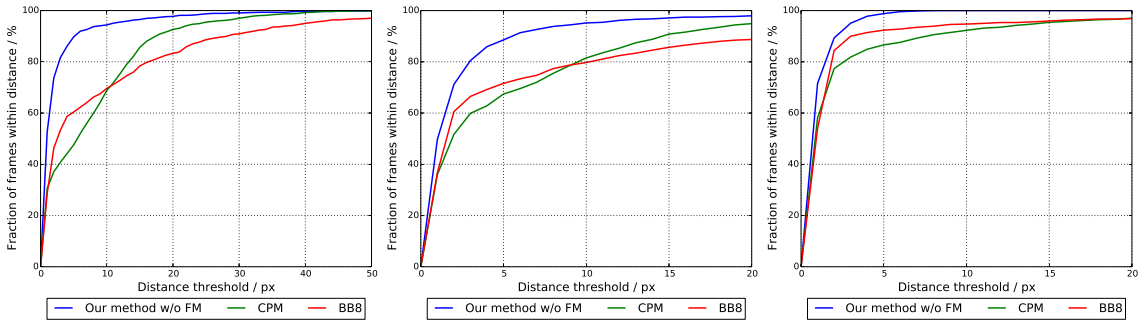
The Effect of Seeing Occlusions during Training We evaluate the importance of knowing the occluder in advance. [1, 3, 7] assume that the occluder is another object from the LineMOD dataset, and only use occlusions from these objects during training. However, in practice, this assumption does not hold, since the occluder can be an arbitrary object. Therefore, we investigated how performance varies if occlusions are used or not during training.

We compare our results (without Feature Mapping) to two state-of-the-art approaches, our reimplementations of BB8 [1] and CPM [4]. To avoid being biased by the limited amount of training data on the Occluded LineMOD dataset [11], we consider here synthetic images both for training and for testing.

We investigate three different training schemes: (a) No occlusions used for training; (b) Random occlusions by simple geometric shapes; (c) Random oc-

Table 1: Comparison on the Occluded LineMOD dataset [11] with color images only. We provide the percentage of frames for which the AD{D|I} error is smaller than 10% of the object diameter, and for which the 2D Reprojection error is smaller than 5px. Objects marked with a * are considered symmetric.

	AD{D|I}-10%									2D Reprojection Error-5px								
Method	Ape	Can	Cat	Driller	Duck	Eggbox*	Glue*	Holepunch.	Average	Ape	Can	Cat	Driller	Duck	Eggbox*	Glue*	Holepunch.	Average
PoseCNN [7]	9.6	**45.2**	0.93	41.4	**19.6**	22.0	38.5	**22.1**	24.9	34.6	15.1	10.4	7.40	31.8	1.90	13.8	23.1	17.2
Tekin *et al.* [3]	–	–	–	–	–	–	–	–	–	40.4	57.8	23.3	17.4	18.2	–	26.9	39.5	31.9
BB8 [1]	–	–	–	–	–	–	–	–	–	28.5	1.20	9.60	0.00	6.80	–	4.70	2.40	7.60
Jafari *et al.* [8]	–	–	–	–	–	–	–	–	–	24.2	30.2	12.3	–	12.1	–	25.9	20.6	20.8
CPM [4]	4.33	20.3	1.43	25.5	2.12	0.83	15.0	0.91	8.81	15.1	10.4	9.34	12.3	7.52	1.16	18.5	1.16	9.45
Our method w/o FM	*14.2*	36.9	*8.82*	*46.6*	11.1	*22.9*	*39.7*	20.3	*25.1*	*57.7*	*55.9*	*47.9*	*34.4*	*35.0*	*9.85*	*42.9*	*48.5*	*41.5*
Our method	**15.3**	*44.7*	**9.33**	**55.4**	**19.6**	**23.0**	**41.4**	*20.4*	**28.7**	**63.4**	**74.5**	**56.8**	**64.2**	**60.2**	**13.5**	**54.2**	**60.4**	**55.9**



(a) (b) (c)

Fig. 8: Evaluation on synthetic renderings of scenes from the Occluded LineMOD dataset (see text) using the 2D Reprojection error. (a) training without occlusions; (b) training with random geometric occlusions; (c) training with occluding objects from the LineMOD dataset [34]. Knowing the occluders in advance significantly improves the performances of BB8 [1] and CPM [4], however, this knowledge is often not available in practice. Our method does not require this knowledge.

clusions with the same objects from the dataset, as in [1, 3, 8]. We compare the different training schemes in Fig. 8. Training without occlusions clearly performs worse for BB8 and CPM, whereas our method is significantly more robust. Interestingly, when adding random geometric occlusions, the performances decrease for all the methods, since the networks learn to ignore only these occlusions. Using occluders from the dataset gives the best results, since the networks learn to ignore specific features from these occluders. This, however, is only possible when the occluders are known in advance, which is not necessarily the case in practice.

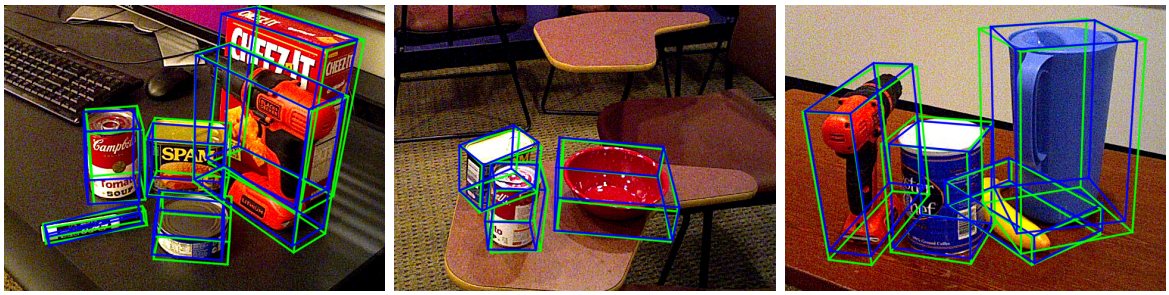


Fig. 9: Qualitative results on the YCB-Video dataset [7]. The green bounding boxes correspond to the ground truth poses, the blue ones to our estimated poses. More results are provided in the supplementary material.

Number of Patches In the supplementary material, we study in detail the influence of the number of patches on the predicted pose accuracy. The main conclusions are that the accuracy starts to flatten when more than 64 patches are used, and that if some preprocessing algorithm would provide a segmentation mask, we could reduce the number of patches to achieve the same accuracy.

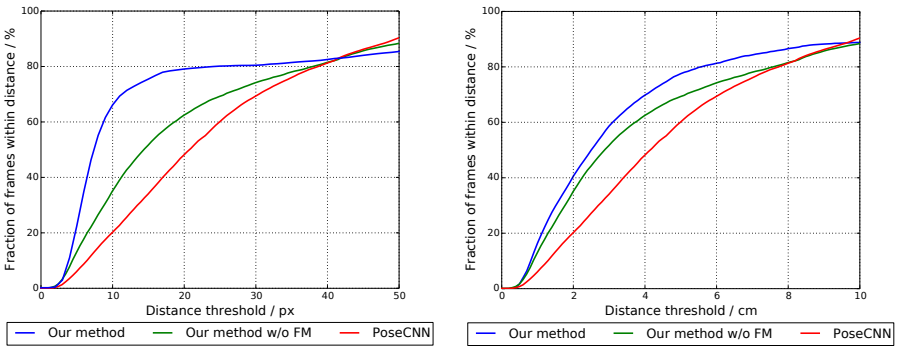
Runtime We implemented our method in Python on an Intel i7 with 3.2GHz and 64GB of RAM, and an nVidia GTX 980 Ti graphics card. Pose estimation is $100ms$ for 64 patches, and detection takes $150ms$ on a 640×480 camera frame. Predicting the heatmaps for a single patch takes $4ms$, and total run-time could thus be significantly reduced by processing the individual patches in parallel.

5.4 YCB-Video Dataset

The recently proposed YCB-Video dataset [7] consists of 92 video sequences, where 12 sequences are used for testing and the remaining 80 sequences for training. In addition, the dataset contains 80k synthetically rendered images, which can be used for training as well. There are 21 objects in the dataset, which are taken from the YCB dataset [35] and are publicly availably for purchase. The dataset is captured with two different RGB-D sensors, each providing 640×480 images, but we only use the color images. The test images are very challenging due to significant image noise and illumination changes. Each image is annotated with the 3D object poses, as well as the objects' masks. Fig. 9 shows some qualitative results. We give an extensive quantitative evaluation in the following.

Quantitative Results Fig. 10(a) and (b) plot the fraction of frames where the 2D Reprojection error and the AD{D|I} metrics averaged over all the objects are smaller than a given threshold. Our approach clearly performs better. [7] used the area under the accuracy-threshold curve as a metric, which we also provide in Table 2, in addition to the other metrics.[1] Again, our approach performs better according to these metrics.

[1] The 10% and $5px$ metrics are calculated from the poses the authors provide at https://github.com/yuxng/YCB_Video_toolbox.



(a) 2D Reprojection error. (b) AD{D|I} error.

Fig. 10: Evaluation on YCB-Video dataset [7]. We plot the fraction of frames where (a) the 2D Reprojection error and (b) AD{D|I} metrics are smaller than a threshold.

Table 2: Comparison on the YCB-Video dataset [7]. We refer to the supplemental material for the object-specific numbers. Our method outperforms the baseline.

Method	PoseCNN [7]			Our method w/o FM			Our method		
	AUC	AD{D|I}-10%	2D Repr-5*px*	AUC	AD{D|I}-10%	2D Repr-5*px*	AUC	AD{D|I}-10%	2D Repr-5*px*
Average	58.7	16.2	3.32	*59.7*	*30.9*	*15.0*	**66.1**	**37.1**	**28.1**

6 Discussion and Conclusion

In this paper, we introduced a novel method for 3D object pose estimation that is inherently robust to partial occlusions of the object. To do this, we consider only small image patches as input and merge their contributions. Because we chose to compute the pose by first predicting the 2D projections of 3D points related to the object, the prediction can be done in the form of 2D heatmaps. Since heatmaps are closely related to density functions, they are convenient to capture the ambiguities that arise when using small image patches as input. We showed that training a network to predict the heatmaps under such ambiguities is much simpler than it may sound. This results in a very simple pipeline, which outperforms much more complex methods on two challenging datasets.

Our approach can be extended in different ways. The heatmaps could be merged in a way that is more robust to erroneous values than simple averaging. The pose could be estimated by considering the best local maxima rather than only the global maxima. Sampling only patches intersecting with the object mask, which could be predicted by a segmentation method, would limit even more the influence of occluders and background in the accumulated heatmaps. Predicting the heatmaps could be performed in parallel. However, our method was shown to work already very well, and it would be interesting to apply it to other computer vision problems where occlusions may be an issue, such as object category detection.

Acknowledgment

This work was funded by the Christian Doppler Laboratory for Semantic 3D Computer Vision.

References

1. Rad, M., Lepetit, V.: BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In: International Conference on Computer Vision. (2017)
2. Pavlakos, G., Zhou, X., Chan, A., Derpanis, K.G., Daniilidis, K.: 6-DoF Object Pose from Semantic Keypoints. In: International Conference on Intelligent Robots and Systems. (2018)
3. Tekin, B., Sinha, S.N., Fua, P.: Real-Time Seamless Single Shot 6D Object Pose Prediction. In: Conference on Computer Vision and Pattern Recognition. (2018)
4. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional Pose Machines. In: Conference on Computer Vision and Pattern Recognition. (2016)
5. Krull, A., Brachmann, E., Michel, F., Yang, M.Y., Gumhold, S., Rother, C.: Learning Analysis-By-Synthesis for 6D Pose Estimation in RGB-D Images. In: International Conference on Computer Vision. (2015)
6. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In: International Conference on Computer Vision. (2017)
7. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. arXiv Preprint (2018)
8. Jafari, O.H., Mustikovela, S.K., Pertsch, K., Brachmann, E., Rother, C.: iPose: Instance-Aware 6D Pose Estimation of Partly Occluded Objects. CoRR **abs/1712.01924** (2017)
9. Brachmann, E., Michel, F., Krull, A., Yang, M.M., Gumhold, S., Rother, C.: Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In: Conference on Computer Vision and Pattern Recognition. (2016)
10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
11. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D Object Pose Estimation Using 3D Object Coordinates. In: European Conference on Computer Vision. (2014)
12. Wang, J., Xie, Q., Zhang, Z., Zhu, J., Xie, L., Yuille, A.L.: Detecting Semantic Parts on Partially Occluded Objects. In: British Machine Vision Conference. (2017)
13. Rad, M., Oberweger, M., Lepetit, V.: Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images. In: Conference on Computer Vision and Pattern Recognition. (2018)
14. Harris, C., Stennett, C.: RAPID-a Video Rate Object Tracker. In: British Machine Vision Conference. (1990)
15. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **20**(2) (2004)
16. Crivellaro, A., Rad, M., Verdie, Y., Yi, K., Fua, P., Lepetit, V.: Robust 3D Object Tracking from Monocular Images Using Stable Parts. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Conference on Computer Vision and Pattern Recognition. (2016)
18. Hueting, M., Reddy, P., Kim, V., Carr, N., Yumer, E., Mitra, N.: SeeThrough: Finding Chairs in Heavily Occluded Indoor Scene Images. In: arXiv Preprint. (2017)
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. CoRR **abs/1512.02325** (2016)
20. Osherov, E., Lindenbaum, M.: Increasing CNN Robustness to Occlusions by Reducing Filter Support. In: International Conference on Computer Vision. (2017)
21. Buch, A.G., Kiforenko, L., Kraft, D.: Rotational Subgroup Voting and Pose Clustering for Robust 3D Object Recognition. In: International Conference on Computer Vision. (2017)
22. Mitash, C., Boularias, A., Bekris, K.E.: Improving 6D Pose Estimation of Objects in Clutter via Physics-aware Monte Carlo Tree Search. In: International Conference on Robotics and Automation. (2018)
23. Zhang, H., Cao, Q.: Combined Holistic and Local Patches for Recovering 6D Object Pose. In: International Conference on Computer Vision Workshops. (2017)
24. Doumanoglou, A., Balntas, V., Kouskouridas, R., Kim, T.: Siamese Regression Networks with Efficient Mid-Level Feature Extraction for 3D Object Pose Estimation. ARXIV (2016)
25. Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. In: European Conference on Computer Vision. (2016)
26. Gall, J., Lempitsky, V.: Class-Specific Hough Forests for Object Detection. In: Conference on Computer Vision and Pattern Recognition. (2009)
27. Riegler, G., Ferstl, D., Rüther, M., Bischof, H.: Hough Networks for Head Pose Estimation and Facial Feature Localization. In: British Machine Vision Conference. (2014)
28. Zeiler, M., Krishnan, D., Taylor, G., Fergus, R.: Deconvolutional Networks. In: Conference on Computer Vision and Pattern Recognition. (2010)
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Conference on Computer Vision and Pattern Recognition. (2016)
30. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: Advances in Neural Information Processing Systems. (2015) 2017–2025
31. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Conference on Computer Vision and Pattern Recognition. (2009)
32. Dwibedi, D., Misra, I., Hebert, M.: Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In: International Conference on Computer Vision. (2017)
33. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: International Conference on Machine Learning. (2015)
34. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In: Asian Conference on Computer Vision. (2012)
35. Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., Dollar, A.M.: Yale-CMU-Berkeley Dataset for Robotic Manipulation Research. International Journal of Robotics Research **36** (2017) 261 – 268

1 Supplementary Material

1.1 Evaluation Metrics

For the evaluation of the 3D object pose, we apply the most common metrics [1–3]: Firstly, the 2D Reprojection error. This error computes the distances between the projected 3D model points $\mathcal{M}$ in 2D. Hereby, $\mathbf{Rt}$ denotes the ground truth pose, $\widehat{\mathbf{Rt}}$ denotes our estimated pose, and $\mathbf{K}$ denotes the intrinsic camera calibration matrix. The 2D Reprojection error can be calculated as:

$$\frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \|\mathbf{K} \cdot \mathbf{Rt} \cdot \mathbf{x} - \mathbf{K} \cdot \widehat{\mathbf{Rt}} \cdot \mathbf{x}\| \tag{1}$$

Similarly, the ADD metric calculates the average distance in 3D between the model points transformed by ground truth pose and our estimated pose:

$$\text{ADD} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \|\mathbf{Rt} \cdot \mathbf{x} - \widehat{\mathbf{Rt}} \cdot \mathbf{x}\| \tag{2}$$

Further, for symmetric objects, the 3D distances are calculated between the closest 3D points, referred to as the ADI metric:

$$\text{ADI} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|\mathbf{Rt} \cdot \mathbf{x}_1 - \widehat{\mathbf{Rt}} \cdot \mathbf{x}_2\| \tag{3}$$

1.2 Visualization Occluded Feature Maps

Fig. 1 shows the effect of occlusions on the feature maps of CNNs. We show two examples: in the top part of the figure an example with small occlusion and in the bottom part of the figure an example with a larger occlusion. In each part of the figure, we show the feature maps with and without occlusions together with the sums of the squared differences between them. Further, we compare when the network is trained with or without occlusion examples. The left column shows the results for a feedforward network [4] and the right column shows the same for a Convolutional Pose Machine [5]. The influence of the occlusion increases with the layers' depths, as receptive fields are larger in the deeper layers than in the first layers, even when the method is trained with occlusion examples.

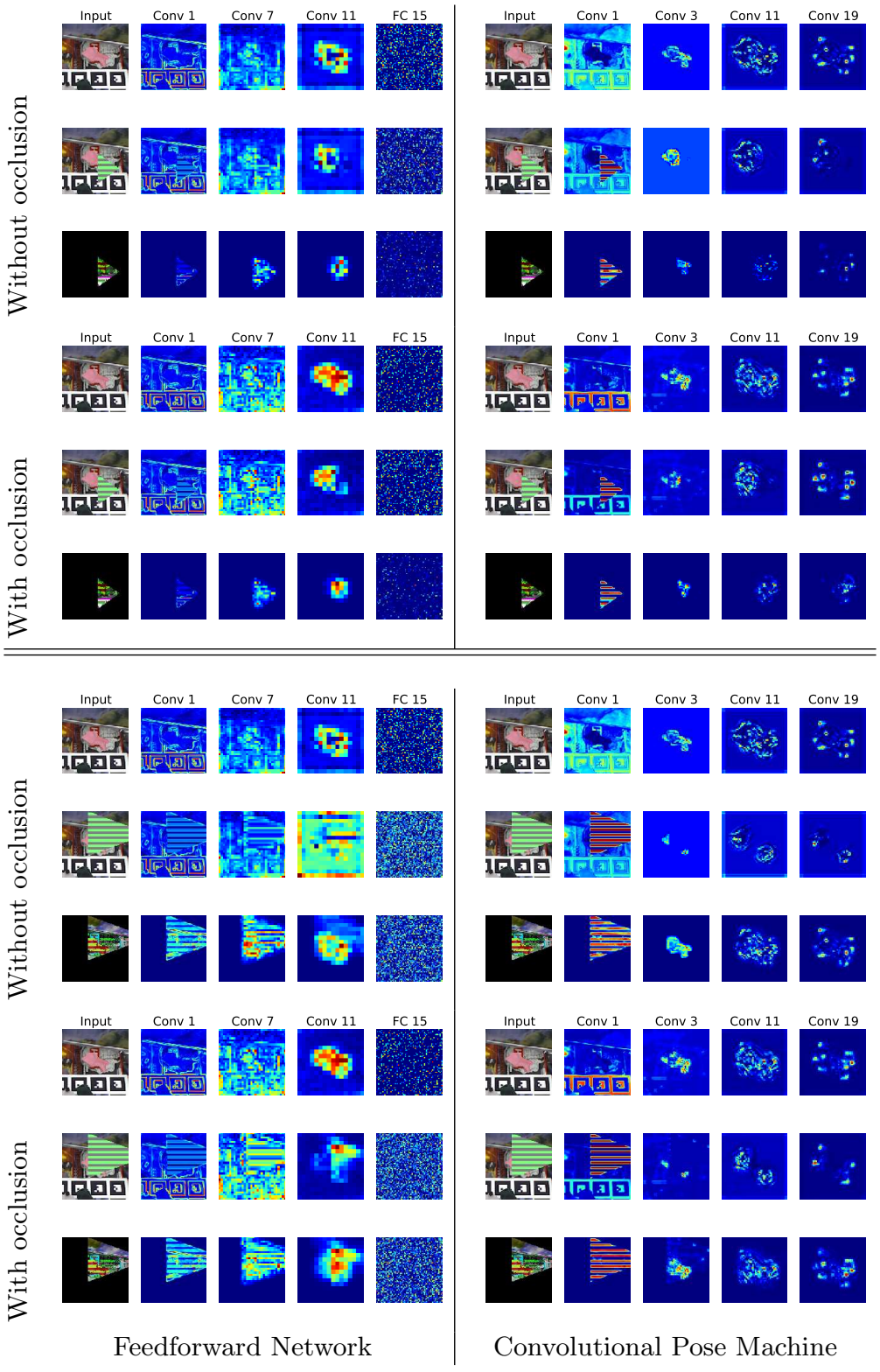


Fig. 1: Effect of occlusions on the feature maps of CNNs. See text for details. (Best viewed on screen)

1.3 Number of Patches

Table 1 shows the aggregated heatmaps for one input image when varying the number of patches.

Table 1: Effect of the number of patches used at run-time. While for small numbers of patches, the resulting heatmaps are strongly effected by the sampling, the aggregation of more heatmaps forms more accurate distributions robust to the patches sampled from occluded regions. (Best viewed on screen)

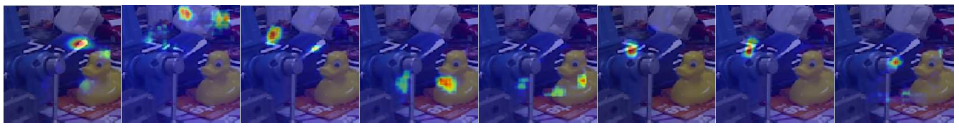
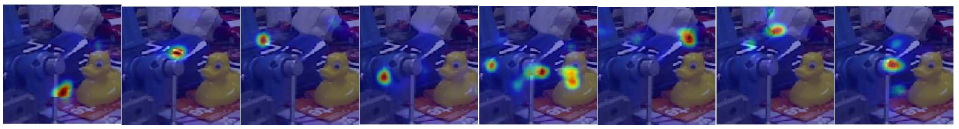
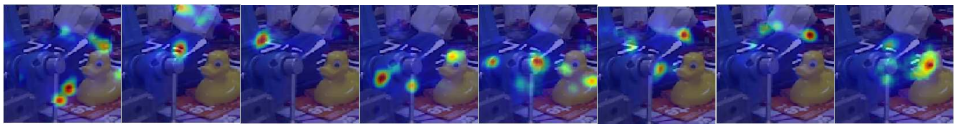
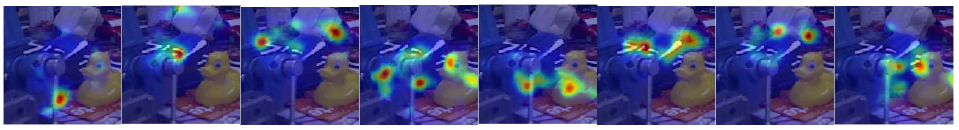
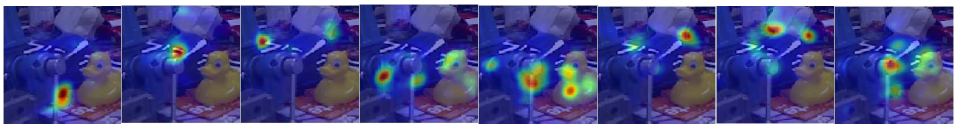
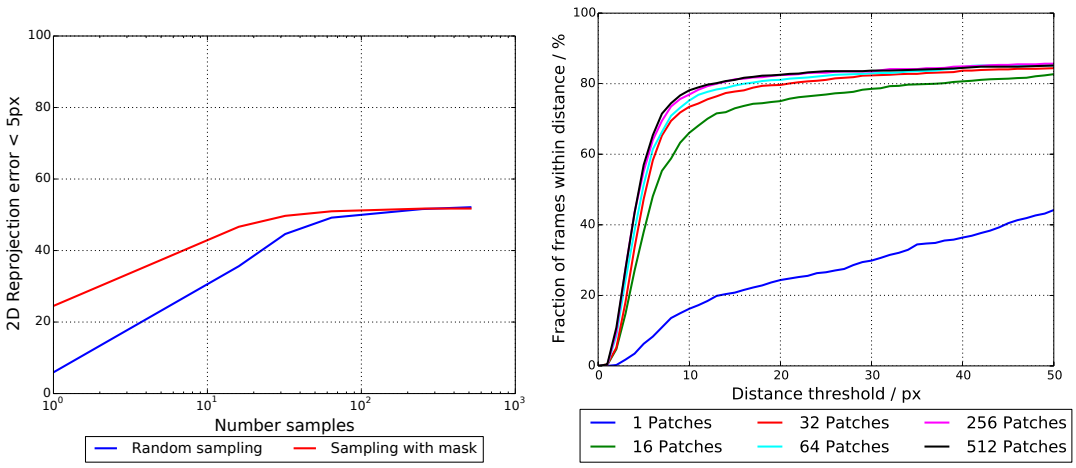
# Patches	Heatmaps
GT	
$N = 1$	
$N = 16$	
$N = 32$	
$N = 64$	
$N = 512$	

Fig. 2(a) shows the influence of the sampling of the patches on the 2D Reprojection error for different sampling strategies. Since we do not have a segmentation mask of the object given during runtime, we rely on random sampling. However, if we sample enough patches, *i.e.* > 64, we can achieve the same performance as sampling the patches only on the object mask that we created for the test sequences. So, if some preprocessing algorithm would provide a segmentation mask as well, we could reduce the number of patches to achieve the same accuracy. Fig. 2(b) shows the influence of the number of patches that we randomly sample. The more patches we sample, the more accurate the results get, however, the accuracy starts to flatten for > 64 patches.



(a) Sampling patches within the object mask versus random sampling.

(b) Influence of number of samples.

Fig. 2: (a) Evaluation of the sampling strategy and (b) the number of sampled patches on the Occluded LineMOD dataset [6] for the *Cat* object. We plot the fraction of frames where the 2D Reprojection error is smaller than a threshold.

1.4 Quantitative Evaluation YCB-Video

A detailed evaluation of the different objects of the YCB-Video dataset is shown in Table 2. Our method performs better for almost all objects, and significantly better on all metrics on average.

Table 2: Comparison on the YCB-Video dataset [3]. We denote the area under the accuracy-threshold curve (AUC), the percentage of frames which have a 2D Reprojection error of less than 5px, and the percentage of frames where the 3D ADD or ADI error is less than 10% of the object diameter. The objects marked with * are considered symmetric.

Object \ Method	PoseCNN [3]			Our method w/o FM			Our method		
	AUC	AD{D|I}-10%	2D Repr-5px	AUC	AD{D|I}-10%	2D Repr-5px	AUC	AD{D|I}-10%	2D Repr-5px
002_master_chef_can	46.7	11.9	3.18	*59.1*	*29.9*	*3.57*	**69.0**	**31.2**	**36.0**
003_cracker_box	59.6	21.5	0.00	*68.8*	*61.2*	*5.41*	**80.2**	**75.0**	**15.3**
004_sugar_box	59.9	29.3	2.54	*70.2*	*42.7*	*18.6*	**76.2**	**47.2**	**34.7**
005_tomato_soup_can	**71.5**	29.3	16.3	69.4	*29.4*	*35.6*	*70.0*	**30.2**	**41.2**
006_mustard_bottle	*80.0*	*54.9*	3.08	71.6	39.5	*29.1*	**84.8**	**72.5**	**47.6**
007_tuna_fish_can	*51.7*	0.44	*1.91*	34.8	*2.09*	0.87	49.4	**4.31**	**31.7**
008_pudding_box	75.9	13.1	3.27	*81.4*	*47.2*	*36.9*	**82.2**	**48.3**	**67.3**
009_gelatin_box	77.8	6.54	1.86	*81.6*	*36.9*	*46.7*	**81.8**	**37.2**	**51.9**
010_potted_meat_can	61.4	18.3	13.8	*64.5*	*39.7*	*30.5*	**66.2**	**40.3**	**47.7**
011_banana	**61.5**	1.05	0.00	39.3	*5.81*	*1.93*	*52.9*	**6.20**	**25.9**
019_pitcher_base	60.7	24.4	0.00	*69.3*	*52.8*	*1.93*	**69.9**	**53.8**	**2.01**
021_bleach_cleanser	58.8	30.7	0.87	*69.3*	*56.9*	*6.71*	**73.3**	**57.2**	**15.9**
024_bowl*	*74.2*	*28.1*	0.00	57.7	10.7	*8.37*	**80.3**	**49.5**	**17.5**
025_mug	*47.1*	1.73	3.45	33.5	*10.2*	*14.9*	**50.5**	**10.5**	**15.0**
035_power_drill	56.7	22.4	0.00	*66.6*	*45.2*	*22.3*	**78.3**	**63.0**	**35.0**
036_wood_block*	*63.9*	0.00	0.04	63.0	*47.9*	*7.85*	**65.2**	**48.2**	**9.21**
037_scissors	**43.9**	0.00	0.00	28.0	0.00	0.00	28.2	**0.55**	**5.52**
040_large_marker	44.2	0.62	0.00	*46.5*	*2.78*	0.00	**48.2**	**11.7**	**31.5**
051_large_clamp*	34.3	6.60	0.00	*46.7*	*10.4*	*1.41*	**47.2**	**12.2**	**8.92**
052_extra_large_clamp*	38.6	3.66	0.00	*46.7*	*14.5*	0.00	**47.5**	**17.3**	**7.52**
061_foam_brick*	82.0	35.1	19.1	*85.5*	*62.8*	*42.7*	**85.6**	**63.8**	**42.8**
Average	58.7	16.2	3.32	*59.7*	*30.9*	*15.0*	**66.1**	**37.1**	**28.1**

1.5 Qualitative Results Occluded LineMOD

We show qualitative results on the Occluded LineMOD dataset [6] in Fig. 3.

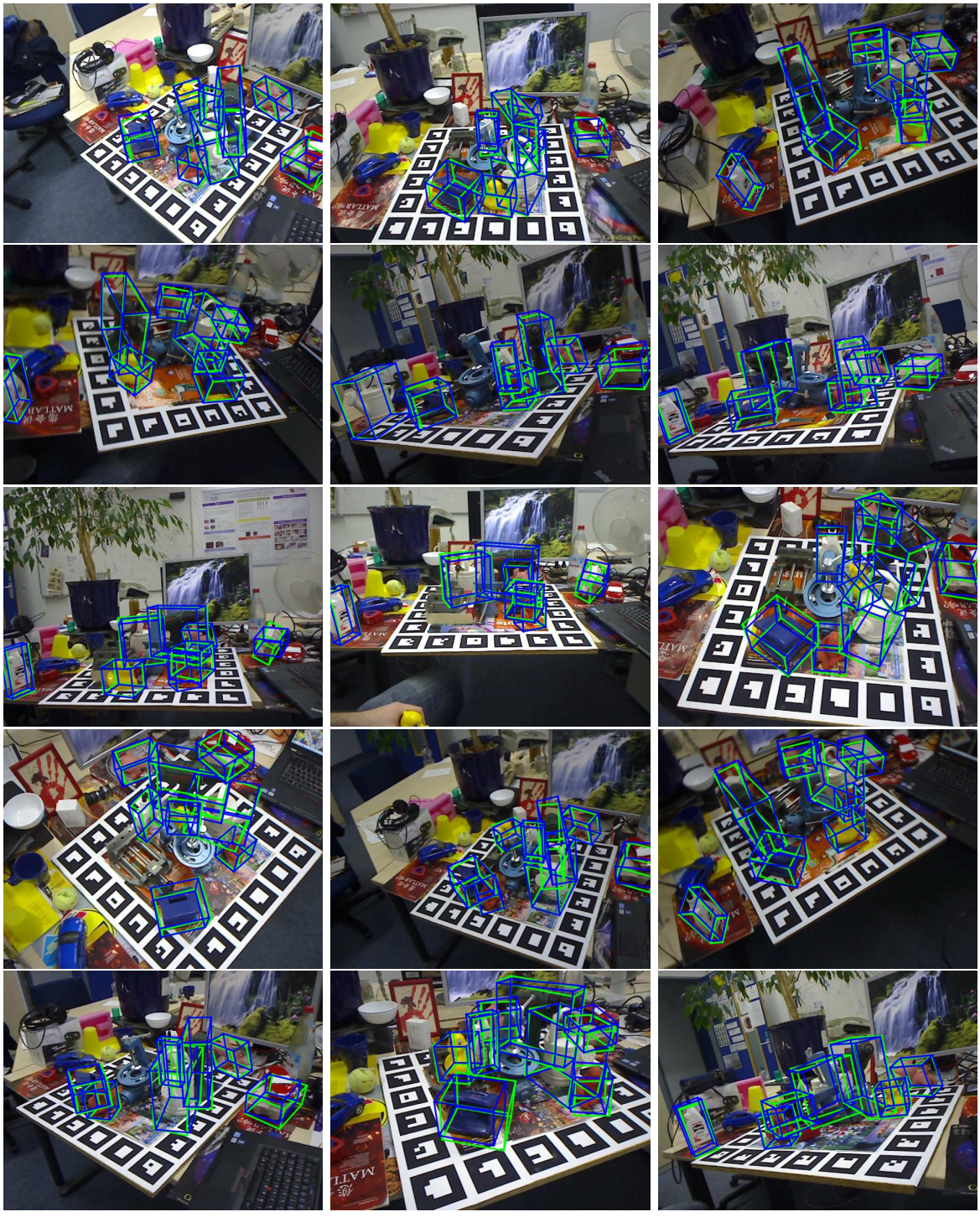


Fig. 3: Qualitative results on the Occluded LineMOD dataset [6]. We show the 3D bounding boxes of the objects projected to the color image. Ground truth poses are shown in green, our predictions are shown in blue. (Best viewed on screen)

1.6 Qualitative Results YCB-Video

We show qualitative results on the YCB-Video dataset [3] in Fig. 4. Further, we would like to refer to the supplementary video for more results on the YCB-Video dataset.

Fig. 4: Qualitative results on the YCB-Video dataset [3]. We show the 3D bounding boxes of the objects projected to the color image. Ground truth poses are shown in green, our predictions are shown in blue. (Best viewed on screen)

References

1. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In: Asian Conference on Computer Vision. (2012)
2. Brachmann, E., Michel, F., Krull, A., Yang, M.M., Gumhold, S., Rother, C.: Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In: Conference on Computer Vision and Pattern Recognition. (2016)
3. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. arXiv Preprint (2018)
4. Rad, M., Lepetit, V.: BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In: International Conference on Computer Vision. (2017)
5. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional Pose Machines. In: Conference on Computer Vision and Pattern Recognition. (2016)
6. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D Object Pose Estimation Using 3D Object Coordinates. In: European Conference on Computer Vision. (2014)