



# Implicit 3D Orientation Learning for 6D Object Detection from RGB Images

Martin Sundermeyer<sup>1</sup>(✉)(iD), Zoltan-Csaba Marton<sup>1</sup>(✉)(iD),  
Maximilian Durner<sup>1</sup>(✉)(iD), Manuel Brucker<sup>1</sup>(✉)(iD), and Rudolph Triebel<sup>1,2</sup>(✉)(iD)

<sup>1</sup> German Aerospace Center (DLR), 82234 Wessling, Germany  
{martin.sundermeyer,zoltan.marton,maximilian.durner,  
manuel.brucker,rudolph.triebel}@dlr.de

<sup>2</sup> Technical University of Munich, 80333 Munich, Germany

**Abstract.** We propose a real-time RGB-based pipeline for object detection and 6D pose estimation. Our novel 3D orientation estimation is based on a variant of the Denoising Autoencoder that is trained on simulated views of a 3D model using Domain Randomization.

This so-called Augmented Autoencoder has several advantages over existing methods: It does not require real, pose-annotated training data, generalizes to various test sensors and inherently handles object and view symmetries. Instead of learning an explicit mapping from input images to object poses, it provides an implicit representation of object orientations defined by samples in a latent space. Experiments on the T-LESS and LineMOD datasets show that our method outperforms similar model-based approaches and competes with state-of-the-art approaches that require real pose-annotated images.

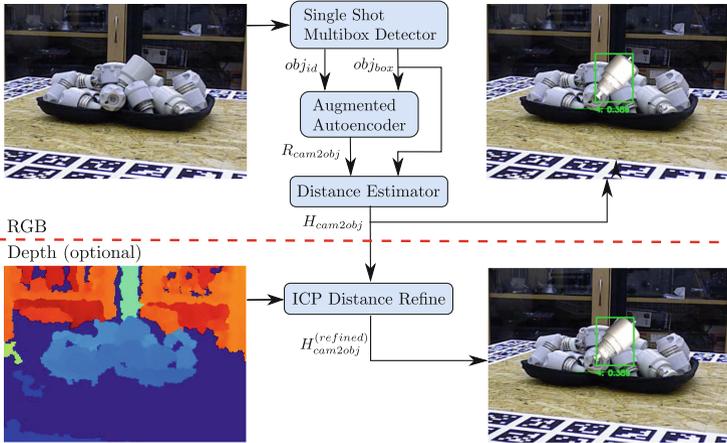
**Keywords:** 6D object detection · Pose estimation  
Domain Randomization · Autoencoder · Synthetic data  
Pose ambiguity · Symmetries

## 1 Introduction

One of the most important components of modern computer vision systems for applications such as mobile robotic manipulation and augmented reality is a reliable and fast 6D object detection module. Although, there are very encouraging recent results [12, 15, 17, 38, 40], a flexible, general, robust and fast solution is not available, yet. The reasons for this are manifold. First and foremost, current solutions are not robust enough against typical challenges such as object occlusions, different kinds of background clutter, and dynamic changes of the environment. Second, existing methods often require certain object properties

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-01231-1\\_43](https://doi.org/10.1007/978-3-030-01231-1_43)) contains supplementary material, which is available to authorized users.



**Fig. 1.** Our 6D object detection pipeline with homogeneous transformation  $H_{cam2obj} \in \mathcal{R}^{4 \times 4}$  (top-right) and depth-refined result  $H_{cam2obj}^{(refined)}$  (bottom-right)

such as enough textural surface structure or an asymmetric shape to avoid confusions. And finally, current systems are not efficient in terms of run-time and in the amount of annotated training data they require.

Therefore, we propose a novel approach that directly addresses these issues. Concretely, our method operates on single RGB images, which significantly increases the usability as no depth information is required. We note though that depth maps may be incorporated optionally to refine the estimation. As a first step, we apply a Single Shot Multibox Detector (SSD) [22] that provides object bounding boxes and identifiers. On the resulting scene crops, we employ our novel 3D orientation estimation algorithm, which is based on a previously trained deep network architecture. While deep networks are also used in existing approaches, our approach differs in that we do not explicitly learn from 3D pose annotations during training. Instead, we *implicitly* learn representations from rendered 3D model views. This is accomplished by training a generalized version of the Denoising Autoencoder [39], that we call ‘*Augmented Autoencoder (AAE)*’, using a novel Domain Randomization [36] strategy. Our approach has several advantages: First, since the training is independent from concrete representations of object orientations within  $SO(3)$  (e.g. quaternions), we can handle ambiguous poses caused by symmetric views because we avoid one-to-many mappings from images to orientations. Second, we learn representations that specifically encode 3D orientations while achieving robustness against occlusion, cluttered backgrounds and generalizing to different environments and test sensors. Finally, the AAE does not require any real pose-annotated training data. Instead, it is trained to encode 3D model views in a self-supervised way, overcoming the need of a large pose-annotated dataset. A schematic overview of the approach is shown in Fig. 1.

## 2 Related Work

Depth-based methods (e.g. using Point Pair Features (PPF) [12, 38]) have shown robust pose estimation performance on multiple datasets, winning the SIXD challenge 2017 [14]. However, they usually rely on the computationally expensive evaluation of many pose hypotheses. Furthermore, existing depth sensors are often more sensitive to sunlight or specular object surfaces than RGB cameras.

Convolutional Neural Networks (CNNs) have revolutionized 2D object detection from RGB images [20, 22, 29]. But, in comparison to 2D bounding box annotation, the effort of labeling real images with full 6D object poses is magnitudes higher, requires expert knowledge and a complex setup [15]. Nevertheless, the majority of learning-based pose estimation methods use real labeled images and are thus restricted to pose-annotated datasets [4, 28, 35, 40].

In consequence, some works [17, 40] have proposed to train on synthetic images rendered from a 3D model, yielding a great data source with pose labels free of charge. However, naive training on synthetic data does not typically generalize to real test images. Therefore, a main challenge is to bridge the domain gap that separates simulated views from real camera recordings.

### 2.1 Simulation to Reality Transfer

There exist three major strategies to generalize from synthetic to real data:

**Photo-Realistic Rendering** of object views and backgrounds has shown mixed generalization performance for tasks like object detection and viewpoint estimation [25, 26, 30, 34]. It is suitable for simple environments and performs well if jointly trained with a relatively small amount of real annotated images. However, photo-realistic modeling is always imperfect and requires much effort.

**Domain Adaptation (DA)** [5] refers to leveraging training data from a source domain to a target domain of which a small portion of labeled data (supervised DA) or unlabeled data (unsupervised DA) is available. Generative Adversarial Networks (GANs) have been deployed for unsupervised DA by generating realistic from synthetic images to train classifiers [33], 3D pose estimators [3] and grasping algorithms [2]. While constituting a promising approach, GANs often yield fragile training results. Supervised DA can lower the need for real annotated data, but does not abstain from it.

**Domain Randomization (DR)** builds upon the hypothesis that by training a model on rendered views in a variety of semi-realistic settings (augmented with random lighting conditions, backgrounds, saturation, etc.), it will also generalize to real images. Tobin et al. [36] demonstrated the potential of the Domain Randomization (DR) paradigm for 3D shape detection using CNNs. Hinterstoisser et al. [13] showed that by training only the head network of FasterRCNN [29]

with randomized synthetic views of a textured 3D model, it also generalizes well to real images. It must be noted, that their rendering is almost photo-realistic as the textured 3D models have very high quality. Recently, Kehl et al. [17] pioneered an end-to-end CNN, called ‘SSD6D’, for 6D object detection that uses a moderate DR strategy to utilize synthetic training data. The authors render views of textured 3D object reconstructions at random poses on top of MS COCO background images [21] while varying brightness and contrast. This lets the network generalize to real images and enables 6D detection at 10 Hz. Like us, for very accurate distance estimation they rely on Iterative Closest Point (ICP) post-processing using depth data. In contrast, we do not treat 3D orientation estimation as a classification task.

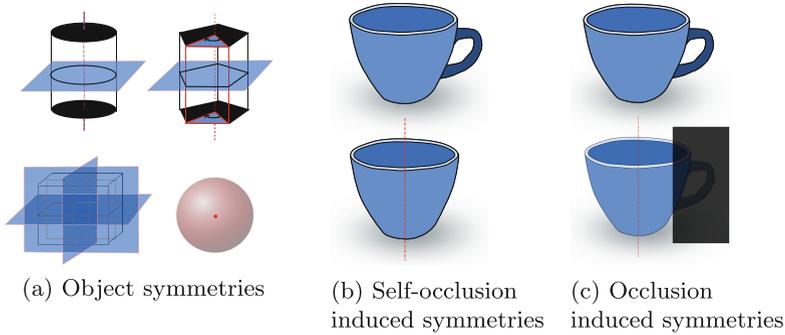
## 2.2 Learning Representations of 3D Orientations

We describe the difficulties of training with fixed  $SO(3)$  parameterizations which will motivate the learning of object-specific representations.

**Regression.** Since rotations live in a continuous space, it seems natural to directly regress a fixed  $SO(3)$  parameterizations like quaternions. However, representational constraints and pose ambiguities can introduce convergence issues [32]. In practice, direct regression approaches for full 3D object orientation estimation have not been very successful [23].

**Classification** of 3D object orientations requires a discretization of  $SO(3)$ . Even rather coarse intervals of  $\sim 5^\circ$  lead to over 50.000 possible classes. Since each class appears only sparsely in the training data, this hinders convergence. In SSD6D [17] the 3D orientation is learned by separately classifying a discretized viewpoint and in-plane rotation, thus reducing the complexity to  $\mathcal{O}(n^2)$ . However, for non-canonical views, e.g. if an object is seen from above, a change of viewpoint can be nearly equivalent to a change of in-plane rotation which yields ambiguous class combinations. In general, the relation between different orientations is ignored when performing one-hot classification.

**Symmetries** are a severe issue when relying on fixed representations of 3D orientations since they cause pose ambiguities (Fig. 2). If not manually addressed, identical training images can have different orientation labels assigned which can significantly disturb the learning process. In order to cope with ambiguous objects, most approaches in literature are manually adapted [9, 17, 28, 40]. The strategies reach from ignoring one axis of rotation [9, 40] over adapting the discretization according to the object [17] to the training of an extra CNN to predict symmetries [28]. These depict tedious, manual ways to filter out object symmetries (2a) in advance, but treating ambiguities due to self-occlusions (2b) and occlusions (2c) are harder to address. Symmetries do not only affect regression and classification methods, but any learning-based algorithm that discriminates object views solely by fixed  $SO(3)$  representations.



**Fig. 2.** Causes of pose ambiguities

**Descriptor Learning** can be used to learn a representation that relates object views in a low-dimensional space. Wohlhart et al. [40] introduced a CNN-based descriptor learning approach using a triplet loss that minimizes/maximizes the Euclidean distance between similar/dissimilar object orientations. Although mixing in synthetic data, the training also relies on pose-annotated sensor data. Furthermore, the approach is not immune against symmetries because the loss can be dominated by ambiguous object views that appear the same but have opposite orientations. Baltas et al. [1] extended this work by enforcing proportionality between descriptor and pose distances. They acknowledge the problem of object symmetries by weighting the pose distance loss with the depth difference of the object at the considered poses. This heuristic increases the accuracy on symmetric objects with respect to [40]. Our work is also based on learning descriptors, but we train self-supervised Augmented Autoencoders (AAEs) such that the learning process itself is independent of any fixed  $SO(3)$  representation. This means that descriptors are learned solely based on the appearance of object views and thus symmetrical ambiguities are inherently regarded. Assigning 3D orientations to the descriptors only happens after the training. Furthermore, unlike [1, 40] we can abstain from the use of real labeled data for training.

Kehl et al. [18] train an Autoencoder architecture on random RGB-D scene patches from the LineMOD dataset [10]. At test time, descriptors from scene and object patches are compared to find the 6D pose. Since the approach requires the evaluation of a lot of patches, it takes about 670ms per prediction. Furthermore, using local patches means to ignore holistic relations between object features which is crucial if few texture exists. Instead we train on holistic object views and explicitly learn domain invariance.

### 3 Method

In the following, we mainly focus on the novel 3D orientation estimation technique based on the Augmented Autoencoder (AAE).

### 3.1 Autoencoders

The original Autoencoder (AE), introduced by Hinton et al. [31], is a dimensionality reduction technique for high dimensional data such as images, audio or depth. It consists of an Encoder  $\Phi$  and a Decoder  $\Psi$ , both arbitrary learnable function approximators which are usually neural networks. The training objective is to reconstruct the input  $x \in \mathcal{R}^D$  after passing through a low-dimensional bottleneck, referred to as the latent representation  $z \in \mathcal{R}^n$  with  $n \ll D$ :

$$\hat{x} = (\Psi \circ \Phi)(x) = \Psi(z) \quad (1)$$

The per-sample loss is simply a sum over the pixel-wise L2 distance

$$\ell_2 = \sum_{i \in \mathcal{D}} \|x_{(i)} - \hat{x}_{(i)}\|_2 \quad (2)$$

The resulting latent space can, for example, be used for unsupervised clustering. *Denoising Autoencoders* [39] have a modified training procedure. Here, artificial random noise is applied to the input images  $x \in \mathcal{R}^D$  while the reconstruction target stays clean. The trained model can be used to reconstruct denoised test images. But how is the latent representation affected?

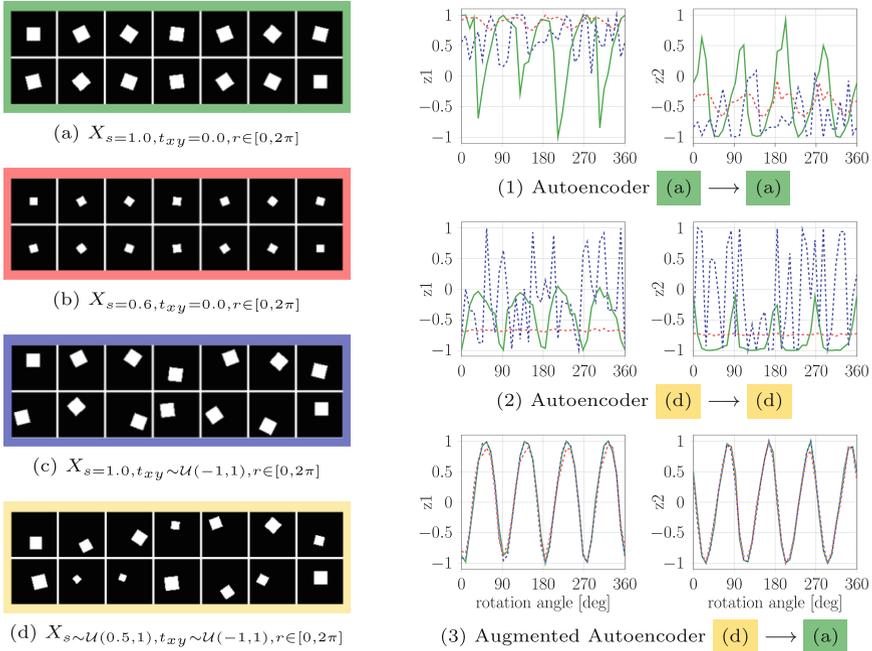
**Hypothesis 1:** *The Denoising AE produces latent representations which are invariant to noise because it facilitates the reconstruction of de-noised images.* We will demonstrate that this training strategy actually enforces invariance not only against noise but against a variety of different input augmentations. Finally, it allows us to bridge the domain gap between simulated and real data.

### 3.2 Augmented Autoencoder

The motivation behind the AAE is to control what the latent representation encodes and which properties are ignored. We apply random augmentations  $f_{augm}(\cdot)$  to the input images  $x \in \mathcal{R}^D$  against which the encoding shall become invariant. The reconstruction target remains Eq. (2) but Eq. (1) becomes

$$\hat{x} = (\Psi \circ \Phi \circ f_{augm})(x) = (\Psi \circ \Phi)(x') = \Psi(z') \quad (3)$$

To make evident that **Hypothesis 1** holds for geometric transformations, we learn latent representations of binary images depicting a 2D square at different scales, in-plane translations and rotations. Our goal is to encode only the in-plane rotations  $r \in [0, 2\pi]$  in a two dimensional latent space  $z \in \mathcal{R}^2$  independent of scale or translation. Figure 3 depicts the results after training a CNN-based AE architecture similar to the model in Fig. 5. It can be observed that the AEs trained on reconstructing squares at fixed scale and translation (1) or random scale and translation (2) do not clearly encode rotation alone, but are also sensitive to other latent factors. Instead, the encoding of the AAE (3) becomes invariant to translation and scale such that all squares with coinciding orientation are mapped to the same code. Furthermore, the latent representation is

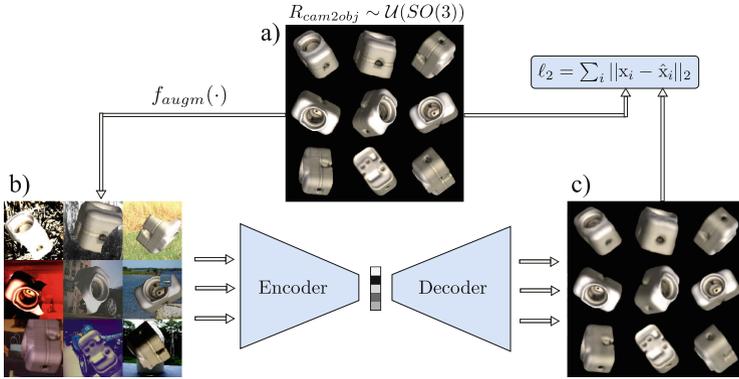


**Fig. 3.** Left:  $64 \times 64$  squares from four distributions (a, b, c and d) distinguished by scale ( $s$ ) and translation ( $t_{xy}$ ) that are used for training and testing [24]. Right: Normalized latent dimensions  $z_1$  and  $z_2$  for all rotations ( $r$ ) of the distribution (a), (b) or (c) after training ordinary AEs (1), (2) and an AAE (3) to reconstruct squares of the same orientation.

much smoother and the latent dimensions imitate a shifted sine and cosine function with frequency  $f = \frac{4}{2\pi}$  respectively. The reason is that the square has two perpendicular axes of symmetry, i.e. after rotating  $\frac{\pi}{2}$  the square appears the same. This property of representing the orientation based on the appearance of an object rather than on a fixed parametrization is valuable to avoid ambiguities due to symmetries when teaching 3D object orientations.

### 3.3 Learning 3D Orientation from Synthetic Object Views

Our toy problem showed that we can explicitly learn representations of object in-plane rotations using a geometric augmentation technique. Applying the same geometric input augmentations we can encode the whole  $SO(3)$  space of views from a 3D object model (CAD or 3D reconstruction) while being robust against inaccurate object detections. However, the encoder would still be unable to relate image crops from real RGB sensors because (1) the 3D model and the real object differ, (2) simulated and real lighting conditions differ, (3) the network can't distinguish the object from background clutter and foreground occlusions. Instead of trying to imitate every detail of specific real sensor recordings in simulation we



**Fig. 4.** Training process for the AAE; (a) reconstruction target batch  $\mathbf{x}$  of uniformly sampled  $SO(3)$  object views; (b) geometric and color augmented input; (c) reconstruction  $\hat{\mathbf{x}}$  after 30000 iterations

propose a Domain Randomization (DR) technique within the AAE framework to make the encodings invariant to insignificant environment and sensor variations. The goal is that the trained encoder treats the differences to real camera images as just another irrelevant variation. Therefore, while keeping reconstruction targets clean, we randomly apply additional augmentations to the input training views: (1) rendering with random light positions and randomized diffuse and specular reflection (simple Phong model [27] in OpenGL), (2) inserting random background images from the Pascal VOC dataset [6], (3) varying image contrast, brightness, Gaussian blur and color distortions, (4) applying occlusions using random object masks or black squares. Figure 4 depicts an exemplary training process for synthetic views of object 5 from T-LESS [15].

### 3.4 Network Architecture and Training Details

The convolutional Autoencoder architecture that is used in our experiments is depicted in Fig. 5. We use a bootstrapped pixel-wise L2 loss which is only computed on the pixels with the largest errors (per image bootstrap factor  $b=4$ ). Thereby, finer details are reconstructed and the training does not converge to local minima. Using OpenGL, we render 20000 views of each object uniformly at random 3D orientations and constant distance along the camera axis (700 mm). The resulting images are quadratically cropped and resized to  $128 \times 128 \times 3$  as shown in Fig. 4. All geometric and color input augmentations besides the rendering with random lighting are applied online during training at uniform random strength, parameters are found in the supplement. We use the Adam [19] optimizer with a learning rate of  $2 \times 10^{-4}$ , Xavier initialization [7], a batch size = 64 and 30000 iterations which takes  $\sim 4$  h on a single Nvidia Geforce GTX 1080.

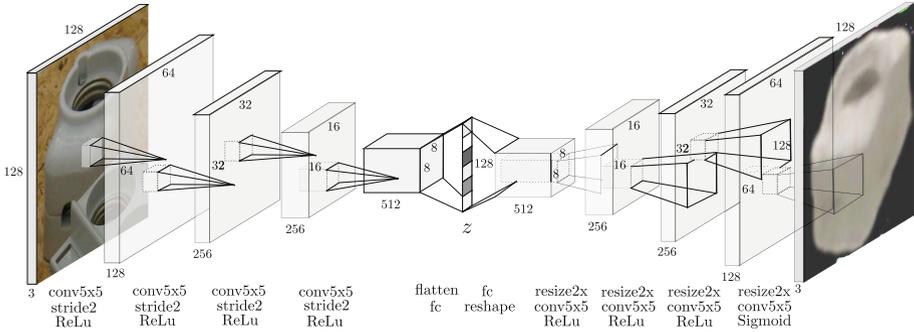


Fig. 5. Autoencoder CNN architecture with occluded test input

### 3.5 Codebook Creation and Test Procedure

After training, the AAE is able to extract a 3D object from real scene crops of many different camera sensors (Fig. 8). The clarity and orientation of the decoder reconstruction is an indicator of the encoding quality. To determine 3D object orientations from test scene crops we create a codebook (Fig. 6 (top)):

- (1) Render clean, synthetic object views at equidistant viewpoints from a full view-sphere (based on a refined icosahedron [8])
- (2) Rotate each view in-plane at fixed intervals to cover the whole  $SO(3)$
- (3) Create a codebook by generating latent codes  $z \in \mathcal{R}^{128}$  for all resulting images and assigning their corresponding rotation  $R_{cam2obj} \in \mathcal{R}^{3 \times 3}$

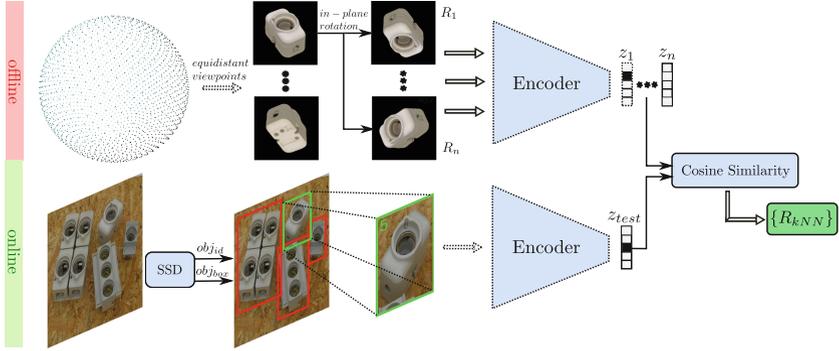
At test time, the considered object(s) are first detected in an RGB scene. The area is quadratically cropped and resized to match the encoder input size. After encoding we compute the cosine similarity between the test code  $z_{test} \in \mathcal{R}^{128}$  and all codes  $z_i \in \mathcal{R}^{128}$  from the codebook:

$$cos_i = \frac{z_i \cdot z_{test}}{\|z_i\| \|z_{test}\|} \quad (4)$$

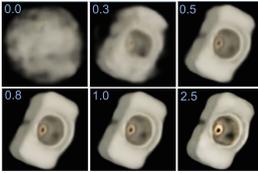
The highest similarities are determined in a k-Nearest-Neighbor (kNN) search and the corresponding rotation matrices  $\{R_{kNN}\}$  from the codebook are returned as estimates of the 3D object orientation. We use cosine similarity because (1) it can be very efficiently computed on a single GPU even for large codebooks. In our experiments we have 2562 equidistant viewpoints  $\times$  36 in-plane rotation = 92232 total entries. (2) We observed that, presumably due to the circular nature of rotations, scaling a latent test code does not change the object orientation of the decoder reconstruction (Fig. 7).

### 3.6 Extending to 6D Object Detection

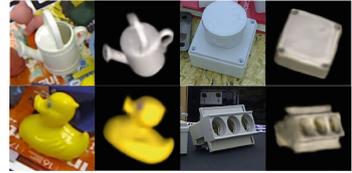
**Training the Object Detector.** We finetune SSD with VGG16 base [22] using object recordings on black background from different viewpoints which



**Fig. 6.** Top: creating a codebook from the encodings of discrete synthetic object views; bottom: object detection and 3D orientation estimation using the nearest neighbor(s) with highest cosine similarity from the codebook



**Fig. 7.** AAE decoder reconstruction of a test code  $z_{test} \in \mathcal{R}^{128}$  scaled by a factor  $s \in [0, 2.5]$



**Fig. 8.** AAE decoder reconstruction of LineMOD (left) and T-LESS (right) scene crops

are provided in the training datasets of LineMOD and T-LESS. We also train RetinaNet [20] with ResNet50 backbone which is slower but more accurate. Multiple objects are copied in a scene at random orientation, scale and translation. Bounding box annotations are adapted accordingly. As for the AAE, the black background is replaced with Pascal VOC images. During training with 60000 scenes, we apply various color and geometric augmentations.

**Projective Distance Estimation.** We estimate the full 3D translation  $t_{pred}$  from camera to object center, similar to [17]. Therefore, for each synthetic object view in the codebook, we save the diagonal length  $l_{syn,i}$  of its 2D bounding box. At test time, we compute the ratio between the detected bounding box diagonal  $l_{test}$  and the corresponding codebook diagonal  $l_{syn,max\_cos}$ , i.e. at similar orientation. The pinhole camera model yields the distance estimate  $t_{pred,z}$

$$t_{pred,z} = t_{syn,z} \times \frac{l_{syn,max\_cos}}{l_{test}} \times \frac{f_{test}}{f_{syn}} \quad (5)$$

**Table 1.** Inference time of RGB pipeline blocks

	4 CPUs	GPU
SSD	-	<b>~17 ms</b>
Encoder	-	<b>~5 ms</b>
Cosine similarity	2.5 ms	<b>1.3 ms</b>
Nearest neighbor	<b>0.3 ms</b>	3.2 ms
Projective distance	<b>0.4 ms</b>	-
	<b>~24 ms</b>	

**Table 2.** Single object pose estimation runtime w/o refinement

Method	fps
Vidal et al. [38]	0.2
Brachmann et al. [4]	2
Kehl et al. [18]	2
BB8 [28]	4
SSD6D [17]	12
OURS	42
Tekin et al. [35]	50

with synthetic rendering distance  $t_{syn,z}$  and focal lengths  $f_{test}$ ,  $f_{syn}$  of the test sensor and synthetic views. It follows that

$$\begin{pmatrix} t_{pred,x} \\ t_{pred,y} \end{pmatrix} = \frac{t_{pred,z}}{f_{test}} \begin{pmatrix} (bb_{cent,test,x} - p_{test,x}) - (bb_{cent,syn,x} - p_{syn,x}) \\ (bb_{cent,test,y} - p_{test,y}) - (bb_{cent,syn,y} - p_{syn,y}) \end{pmatrix} \quad (6)$$

with principal points  $p_{test}$ ,  $p_{syn}$  and bounding box centers  $bb_{cent,test}$ ,  $bb_{cent,syn}$ . In contrast to [17], we can predict the 3D translation for different test intrinsics.

**ICP Refinement.** Optionally, the estimate is refined on depth data using a standard ICP approach [41] taking  $\sim 200$  ms on CPU. Details in supplement (Table 2).

**Inference Time.** SSD with VGG16 base and 31 classes plus the AAE (Fig. 5) with a codebook size of  $92232 \times 128$  yield the average inference times depicted in Table 1. We conclude that the RGB-based pipeline is real-time capable at  $\sim 42$  Hz on a Nvidia GTX 1080. This enables augmented reality and robotic applications and leaves room for tracking algorithms. Multiple encoders and corresponding codebooks fit into the GPU memory, making multi-object pose estimation feasible.

## 4 Evaluation

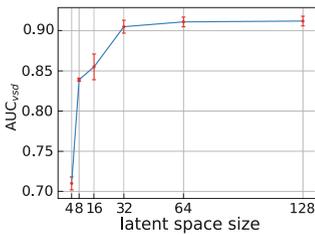
We evaluate the AAE and the whole 6D detection pipeline on the T-LESS [15] and LineMOD [10] datasets. Example sequences are found in the supplement.

### 4.1 Test Conditions

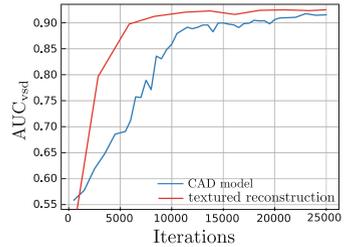
Few RGB-based pose estimation approaches (e.g. [17, 37]) only rely on 3D model information. Most methods make use of real pose annotated data and often even

**Table 3.** Ablation study on color augmentations for different test sensors. Object 5, all scenes, T-LESS [15]. Standard deviation of three runs in brackets.

Train RGB	Test RGB	Dyn. light	Add	Contrast	Multiply	Invert	$AUC_{vsd}$
3D reconstruction	Primesense	✓					0.472 ( $\pm 0.013$ )
		✓	✓				0.611 ( $\pm 0.030$ )
		✓	✓	✓			0.825 ( $\pm 0.015$ )
		✓	✓	✓	✓		0.876 ( $\pm 0.019$ )
		✓	✓	✓	✓	✓	<b>0.877</b> ( $\pm 0.005$ )
			✓	✓	✓		0.861 ( $\pm 0.014$ )
Primesense	Primesense		✓	✓	✓		0.890 ( $\pm 0.003$ )
3D reconstruction	Kinect	✓					0.461 ( $\pm 0.022$ )
		✓	✓				0.580 ( $\pm 0.014$ )
		✓	✓	✓			0.701 ( $\pm 0.046$ )
		✓	✓	✓	✓		0.855 ( $\pm 0.016$ )
		✓	✓	✓	✓	✓	0.897 ( $\pm 0.008$ )
			✓	✓	✓		<b>0.903</b> ( $\pm 0.016$ )
Kinect	Kinect		✓	✓	✓		0.917 ( $\pm 0.007$ )



(a) Effect of latent space size, standard deviation in red



(b) Training on CAD model (bottom) vs. textured 3D reconstruction (top)

**Fig. 9.** Testing object 5 on all 504 Kinect RGB views of scene 2 in T-LESS

train and test on the same scenes (e.g. at slightly different viewpoints) [1, 4, 40]. It is common practice to ignore in-plane rotations or only consider object poses that appear in the dataset [28, 40] which also limits applicability. Symmetric object views are often individually treated [1, 28] or ignored [40]. The SIXD challenge [14] is an attempt to make fair comparisons between 6D localization algorithms by prohibiting the use of test scene pixels. We follow these strict evaluation guidelines, but treat the harder problem of 6D detection where it is unknown which of the considered objects are present in the scene. This is especially difficult in the T-LESS dataset since objects are very similar.

## 4.2 Metrics

The Visible Surface Discrepancy ( $err_{vsd}$ ) [16] is an ambiguity-invariant pose error function that is determined by the distance between the estimated and ground truth visible object depth surfaces. As in the SIXD challenge, we report the recall of correct 6D object poses at  $err_{vsd} < 0.3$  with tolerance  $\tau = 20$  mm and  $>10\%$  object visibility. Although the Average Distance of Model Points (ADD) [11] metric can't handle pose ambiguities, we also present it for the LineMOD dataset following the protocol in [11] ( $k_m = 0.1$ ). For objects with symmetric views (eggbox, glue), [11] computes the average distance to the *closest* model point. In our ablation studies we also report the  $AUC_{vsd}$ , which represents the area under the ‘ $err_{vsd}$  vs. recall’ curve:  $AUC_{vsd} = \int_0^1 recall(err_{vsd}) d err_{vsd}$

## 4.3 Ablation Studies

To assess the AAE alone, in this subsection we only predict the 3D orientation of Object 5 from the T-LESS dataset on Primesense and Kinect RGB scene crops. Table 3 shows the influence of different input augmentations. It can be seen that the effect of different color augmentations is cumulative. For textureless objects, even the inversion of color channels seems to be beneficial since it prevents overfitting to synthetic color information. Furthermore, training with real object recordings provided in T-LESS with random Pascal VOC background and augmentations yields only slightly better performance than the training with synthetic data. Figure 9a depicts the effect of different latent space sizes on the 3D pose estimation accuracy. Performance starts to saturate at  $dim = 64$ . In Fig. 9b we demonstrate that our Domain Randomization strategy even allows the generalization from untextured CAD models.

## 4.4 6D Object Detection

First, we report RGB-only results consisting of 2D detection, 3D orientation estimation and projective distance estimation. Although these results are visually appealing, the distance estimation is refined using a simple cloud-based ICP to compete with state-of-the-art depth-based methods. Table 4 presents our 6D detection evaluation on all scenes of the T-LESS dataset, which contains a lot of pose ambiguities. Our refined results outperform the recent local patch descriptor approach from Kehl et al. [18] even though they only do 6D localization. The state-of-the-art (in terms of average accuracy in the SIXD challenge [14]) from Vidal et al. [38] performs a time consuming search through pose hypotheses (average of 4.9s/object). Our approach yields comparable accuracy while being much more efficient. The right part of Table 4 shows results with ground truth bounding boxes yielding an upper limit on the pose estimation performance. The appendix shows some failure cases, mostly stemming from missed

**Table 4.** T-LESS: Object recall for  $err_{vsd} < 0.3$  on all Primesense test scenes <sup>a</sup>

Object	6D Detection - SSD		6D Detection - retina		6D localization		w/GT 2D BBs	
	OURS RGB	OURS +Depth (ICP)	OURS RGB	OURS +Depth (ICP)	Kehl [18] RGB-D +ICP	Vidal [38] Depth +ICP	OURS RGB	OURS +Depth (ICP)
1	5.65	15.79	8.87	22.32	-	43	12.33	28.05
2	5.46	22.14	13.22	29.49	-	47	11.23	37.30
3	7.05	32.65	12.47	38.26	-	69	13.11	46.15
4	4.61	18.58	6.56	23.07	-	63	12.71	35.30
5	36.45	69.39	34.80	76.10	-	69	66.70	90.29
6	23.15	61.32	20.24	67.64	-	67	52.30	88.28
7	15.97	68.45	16.21	73.88	-	77	36.58	81.75
8	10.86	43.18	19.74	67.02	-	79	22.05	82.65
9	19.59	67.12	36.21	78.24	-	90	46.49	84.38
10	10.47	58.61	11.55	77.65	-	68	14.31	83.12
11	4.35	32.52	6.31	35.89	-	69	15.01	57.26
12	7.80	40.53	8.15	49.30	-	82	31.34	73.75
13	3.30	29.31	4.91	42.50	-	56	13.60	65.01
14	2.85	26.12	4.61	30.53	-	47	45.32	76.05
15	7.90	52.34	26.71	83.73	-	52	50.00	90.56
16	13.06	61.64	21.73	67.42	-	81	36.09	70.57
17	41.70	77.46	64.84	86.17	-	83	81.11	90.49
18	47.17	81.08	14.30	84.34	-	80	52.62	87.47
19	15.95	45.48	22.46	50.54	-	55	50.75	82.50
20	2.17	7.60	5.27	14.75	-	47	37.75	53.84
21	19.77	38.98	17.93	40.31	-	63	50.89	72.10
22	11.01	25.42	18.63	35.23	-	70	47.60	61.74
23	7.98	30.24	18.63	42.52	-	85	35.18	54.65
24	4.74	49.48	4.23	59.54	-	70	11.24	81.34
25	21.91	50.00	18.76	70.89	-	48	37.12	88.54
26	10.04	57.85	12.62	66.20	-	55	28.33	90.66
27	7.42	47.22	21.13	73.51	-	60	21.86	77.63
28	21.78	44.80	23.07	61.20	-	69	42.58	67.10
29	15.33	53.71	26.65	73.05	-	65	57.01	87.68
30	34.63	86.34	29.58	92.90	-	84	70.42	96.45
Mean	<b>14.67</b>	<b>46.51</b>	<b>18.35</b>	<b>57.14</b>	<b>35.9</b>	<b>66.3</b>	<b>36.79</b>	<b>72.76</b>

<sup>a</sup> Since the 3D reconstructions of the T-LESS plugs (Objects 19–23) are missing the pins, we instead use their untextured CAD models

detections or strong occlusions. In Table 5 we compare our method against the recently introduced SSD6D [17] and other methods on the LineMOD dataset. SSD6D also trains on synthetic views of 3D models, but their performance seems quite dependent on a sophisticated occlusion-aware, projective ICP refinement step. Our basic ICP sometimes converges to similarly shaped objects in the vicinity. In the RGB domain our method outperforms SSD6D.

**Table 5.** LineMOD: object recall (ADD [11] metric) using different training and test data, results taken from [35]

Test data	RGB						+Depth (ICP)	
Train data	RGB w/o real pose labels		RGB with real pose labels				+Depth	
Object	SSD6D [17]	OURS	Brachmann [4]	BB8 [28]	Tekin [35]	OURS	SSD6D [17]	
Ape	0.00	<b>3.96</b>	-	<b>27.9</b>	21.62	20.55	<b>65</b>	
Benchvise	0.18	<b>20.92</b>	-	62.0	<b>81.80</b>	64.25	<b>80</b>	
Cam	0.41	<b>30.47</b>	-	<b>40.1</b>	36.57	63.20	<b>78</b>	
Can	1.35	<b>35.87</b>	-	48.1	<b>68.80</b>	76.09	<b>86</b>	
Cat	0.51	<b>17.90</b>	-	<b>45.2</b>	41.82	<b>72.01</b>	70	
Driller	2.58	<b>23.99</b>	-	58.6	<b>63.51</b>	41.58	<b>73</b>	
Duck	0.00	<b>4.86</b>	-	<b>32.8</b>	27.23	32.38	<b>66</b>	
Eggbox	8.90	<b>81.01</b>	-	40.0	<b>69.58</b>	98.64	<b>100</b>	
Glue	0.00	<b>45.49</b>	-	27.0	<b>80.02</b>	96.39	<b>100</b>	
Holepuncher	0.30	<b>17.60</b>	-	42.4	<b>42.63</b>	<b>49.88</b>	49	
Iron	8.86	<b>32.03</b>	-	67.0	<b>74.97</b>	63.11	<b>78</b>	
Lamp	8.2	<b>60.47</b>	-	39.9	<b>71.11</b>	<b>91.69</b>	73	
Phone	0.18	<b>33.79</b>	-	35.2	<b>47.74</b>	70.96	<b>79</b>	
Mean	2.42	<b>28.65</b>	32.3	43.6	<b>55.95</b>	64.67	<b>79</b>	

## 5 Conclusion

We have proposed a new self-supervised training strategy for Autoencoder architectures that enables robust 3D object orientation estimation on various RGB sensors while training only on synthetic views of a 3D model. By demanding the Autoencoder to revert geometric and color input augmentations, we learn representations that (1) specifically encode 3D object orientations, (2) are invariant to a significant domain gap between synthetic and real RGB images, (3) inherently regard pose ambiguities from symmetric object views. Around this approach, we created a real-time (42 fps), RGB-based pipeline for 6D object detection which is especially suitable when pose-annotated RGB sensor data is not available.

**Acknowledgement.** We would like to thank Dr. Ingo Kossyk, Dimitri Henkel and Max Denninger for helpful discussions. We also thank the reviewers for their useful comments.

## References

1. Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R., Kim, T.K.: Pose guided RGBD feature learning for 3D object pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3856–3864 (2017)
2. Bousmalis, K., et al.: Using simulation and domain adaptation to improve efficiency of deep robotic grasping. arXiv preprint [arXiv:1709.07857](https://arxiv.org/abs/1709.07857) (2017)
3. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, p. 7 (2017)

4. Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., Rother, C.: Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3364–3372 (2016)
5. Csurka, G.: Domain adaptation for visual applications: a comprehensive survey. arXiv preprint [arXiv:1702.05374](https://arxiv.org/abs/1702.05374) (2017)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge 2012 (VOC 2012) results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 249–256 (2010)
8. Hinterstoisser, S., Benhimane, S., Lepetit, V., Fua, P., Navab, N.: Simultaneous recognition and homography extraction of local patches with a simple linear classifier. In: Proceedings of the British Machine Conference (BMVC), pp. 1–10 (2008)
9. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(5), 876–888 (2012)
10. Hinterstoisser, S., et al.: Multimodal templates for real-time detection of textureless objects in heavily cluttered scenes. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 858–865. IEEE (2011)
11. Hinterstoisser, S., et al.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-37331-2\\_42](https://doi.org/10.1007/978-3-642-37331-2_42)
12. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K.: Going further with point pair features. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 834–848. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46487-9\\_51](https://doi.org/10.1007/978-3-319-46487-9_51)
13. Hinterstoisser, S., Lepetit, V., Wohlhart, P., Konolige, K.: On pre-trained image features and synthetic images for deep learning. arXiv preprint [arXiv:1710.10710](https://arxiv.org/abs/1710.10710) (2017)
14. Hodan, T.: SIXD Challenge (2017). [http://cmp.felk.cvut.cz/sixd/challenge\\_2017/](http://cmp.felk.cvut.cz/sixd/challenge_2017/)
15. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: an RGB-D dataset for 6D pose estimation of texture-less objects. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2017)
16. Hodaň, T., Matas, J., Obdržálek, Š.: On evaluation of 6D object pose estimation. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 606–619. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-49409-8\\_52](https://doi.org/10.1007/978-3-319-49409-8_52)
17. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: making RGB-based 3D detection and 6D pose estimation great again. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1521–1529 (2017)
18. 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: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 205–220. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46487-9\\_13](https://doi.org/10.1007/978-3-319-46487-9_13)
19. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint [arXiv:1708.02002](https://arxiv.org/abs/1708.02002) (2017)

21. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
22. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
23. Mahendran, S., Ali, H., Vidal, R.: 3D pose regression using convolutional neural networks. arXiv preprint [arXiv:1708.05628](https://arxiv.org/abs/1708.05628) (2017)
24. Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dSprites: disentanglement testing sprites dataset (2017). <https://github.com/deepmind/dsprites-dataset/>
25. Mitash, C., Bekris, K.E., Boularias, A.: A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 545–551. IEEE (2017)
26. Movshovitz-Attias, Y., Kanade, T., Sheikh, Y.: How useful is photo-realistic rendering for visual learning? In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 202–217. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-49409-8\\_18](https://doi.org/10.1007/978-3-319-49409-8_18)
27. Phong, B.T.: Illumination for computer generated pictures. *Commun. ACM* **18**(6), 311–317 (1975)
28. 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: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), pp. 91–99 (2015)
30. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: ground truth from computer games. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 102–118. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46475-6\\_7](https://doi.org/10.1007/978-3-319-46475-6_7)
31. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Technical report, California University, Institute for Cognitive Science, San Diego, La Jolla (1985)
32. Saxena, A., Driemeyer, J., Ng, A.Y.: Learning 3D object orientation from images. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 794–800. IEEE (2009)
33. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2242–2251. IEEE (2017)
34. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2686–2694 (2015)
35. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6D object pose prediction. arXiv preprint [arXiv:1711.08848](https://arxiv.org/abs/1711.08848) (2017)
36. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 23–30. IEEE (2017)

37. Ulrich, M., Wiedemann, C., Steger, C.: CAD-based recognition of 3D objects in monocular images. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol. 9, pp. 1191–1198 (2009)
38. Vidal, J., Lin, C.Y., Martí, R.: 6D pose estimation using an improved method based on point pair features. arXiv preprint [arXiv:1802.08516](https://arxiv.org/abs/1802.08516) (2018)
39. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(Dec), 3371–3408 (2010)
40. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3D pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3109–3118 (2015)
41. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **13**(2), 119–152 (1994)