Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Learning Category-level Implicit 3D Rotation Representations for 6D Pose Estimation from RGB Images

Xiaocan Li[1,2], Yinghao Cai[1], Shuo Wang[1,2,3], Tao Lu[1]
{*lixiaocan2017, yinghao.cai, shuo.wang, tao.lu*}*@ia.ac.cn*

***Abstract*— We exploit the embedding ability of a de-noising autoencoder for an implicit 3D rotation representation learning at the *category level*. Contrast to the exact 3D reconstruction model of each instance-level physical object, we leverage the *inexact* CAD/Reconstruction models of an object as the *representative* model for some category. Under our assumptions that objects within the same category share resemble geometry, we train a de-noising autoencoder on synthetic 3D views of category-level objects to extract the homogenous features at the bottleneck layer. The latent representation is agnostic not only to heterogeneous textures, colors, and illuminations, but also ambiguous pose caused by object symmetry. To extend the instance-level 3D translation estimation to the category level, we considered the 3D diagonal length ratio between the source and target object. We achieved a frame rate of 17Hz.**

***Index Terms*— Pose Estimation, Autoencoder, 6D Object Detection, Category Level**

I. Introduction

6D object pose estimation has been an important research field in the realm of computer vision, a problem which determines the 3D translation and 3D orientation of an object described in the camera coordinate frame. Its implementation can be seen in virtual reality (VR) and augmented reality (AR) that are strongly related to the entertainment industry and medical care, 3D scene understanding, autonomous driving, and robotic grasping such as the Amazon Picking Challenge [1] and underwater manipulation [2]. The 6D object pose estimation can be quite challenging in many perspectives, such as severe occlusions, heavy clutterings, object pose ambiguity due to symmetry or self-occlusion, variations in illumination and appearance, etc.

Hinterstoisser et al. [3] fused color gradient and surface normal features, and generated synthetic views to perform template matching holistically. However, these hand-crafted features were usually susceptible to occlusion and cluttering. Thus, the descriptor or metric learning methods, which aim to learn a representation in a low-dimension manifold automatically, has emerged. Wohlhart et al. [4] designed a triplet loss to pull similar object poses together while pushing away dissimilar ones. Balntas et al. [5] improved [4] by weighting the loss with pose pair difference. This approach was usually more robust to occlusion and other types of noise such as Gaussian noises.

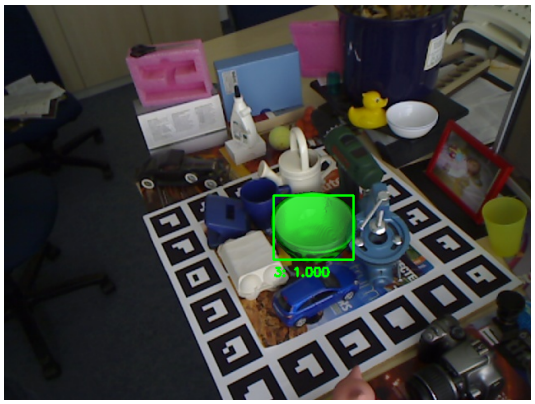


Fig. 1. 6D pose estimation of symmetric object, bowl. Source model: TOYOTALIGHT object id 10; Target model: LINEMOD object id 03.

Recently, convolutional neural network (CNN) has played its role in 6D object pose estimation. Xiang et al. [7] proposed PoseCNN that performs semantic segmentation and 6D pose estimation simultaneously. Similarly, Kehl et al. [8] introduced SSD-6D to perform object detection and pose regression. However, the prediction of 3D orientation is not easy due to its nonlinearity as a Special Orthogonal Group SO(3). To circumvent this nonlinearity issue, Tekin et al. [9] proposed a YOLO-based network to directly predict the 2D image locations of the projected vertices of the object 3D bounding box. Then, the object 6D pose can be calculated through a PnP algorithm. Similarly, Peng et al. [10] introduced PVNet to predict the 2D-3D correspondence by regressing pixel-wise vectors to keypoints and voting for keypoint locations and this work is robust to occlusion while running at a real-time frame rate. Nevertheless, most of these approaches required real pose annotations for training. Hence, Sundermeyer et al. [6] managed to evaluate the self-supervised Autoencoder embedding at the bottleneck layer for orientation learning and does not require depth information.

However, all the above mentioned approaches are all at instance level, which required exhausting ground truth pose annotations instance by instance. Besides, the object

[1]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; [2]University of Chinese Academy of Sciences, Beijing 100049, China; [3]Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

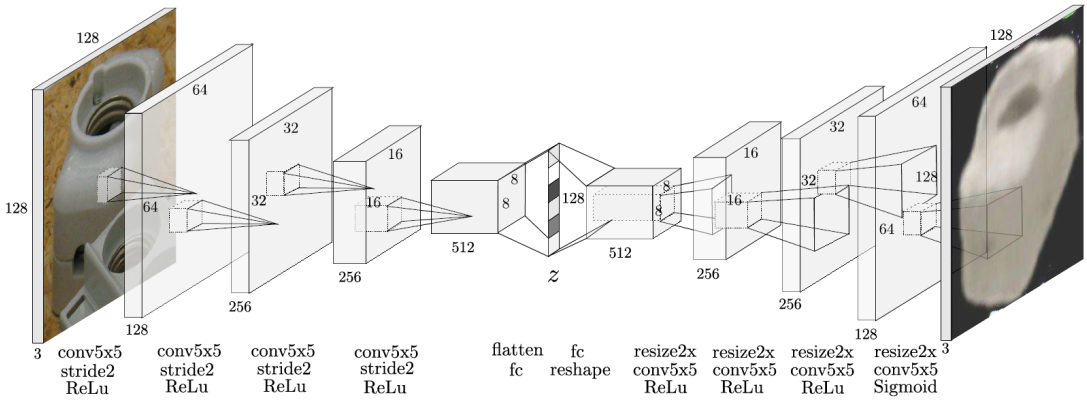


Fig. 2. Network architecture of de-noising autoencoder from Sundermeyer et al. [6]

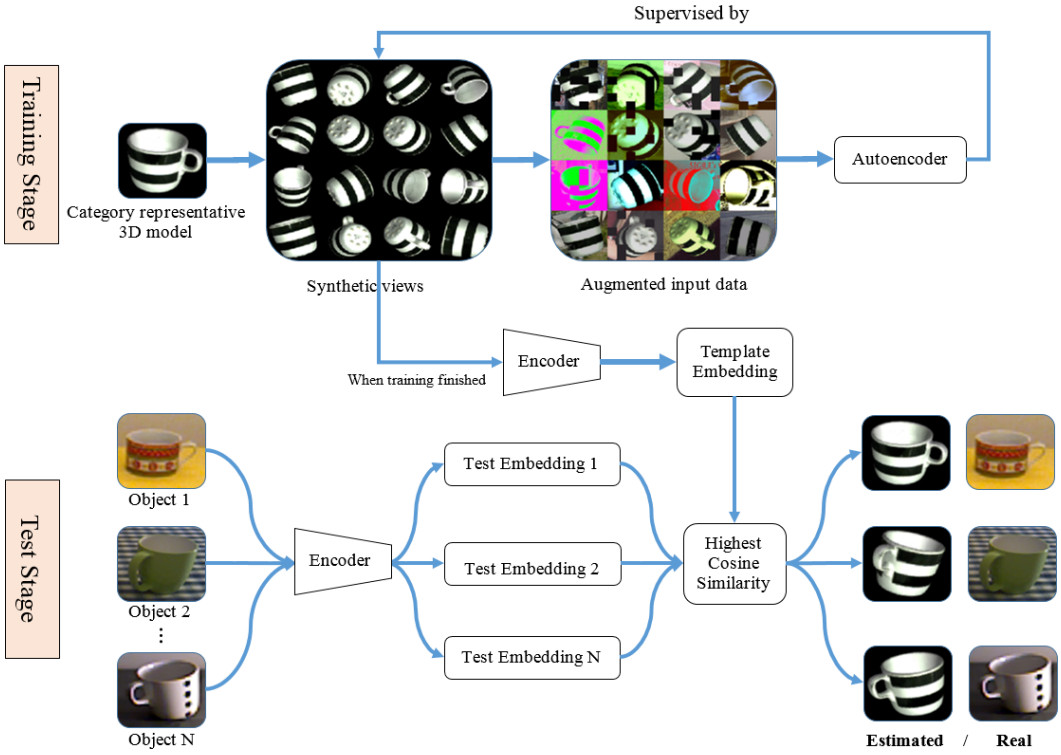


Fig. 3. Pipeline of training and testing.

detector may be confused by similar-looking objects, and consequently affect the pose estimation results. One possible solution to this problem is to perform detection at category level [11], [12]. The challenges of category-level pose estimation lie in the intra-class (intra-category) variation. The size of each instance would be different although they might share similar geometry or shape, which also results in the discrepancies in the depth map. In the color map, textures and colors of different instances would also largely be dissimilar. The aim of category-level pose estimation is to generate the homogeneous features of a category regardless of size, texture and color variations. Fig. 1 is an example of category-level pose estimation. In this paper, we further explore the de-noising autoencoder on the ability to extract category-level implicit rotation embedding as well as extend the translation prediction to the category level for objects of different sizes.

II. METHODOLOGY

A. *De-noising Autoencoder*

Autoencoder [14] has been quite successful in extracting the most informative embedding. One of its variants, de-

Fig. 4. 6D pose dataset example: LINEMOD, TEJANI, TLESS and TOYOTALIGHT.[13]

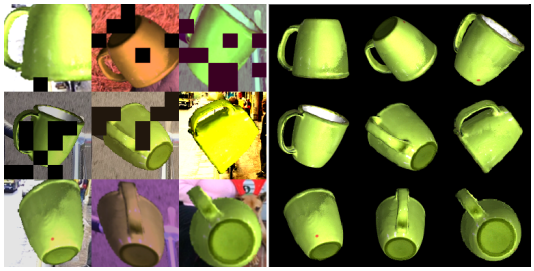


Fig. 5. Perspective transform causes some cups appeared deformed or larger (left) than the original ones (right). This could help the de-noising autoencoder aware of the problem of size inconsistency among objects in the same category.

noising autoencoder [15], was leveraged by Sundermeyer et al. [6] and Kehl et al. [16] to learn implicit rotation descriptor in a low dimension for template matching or voting, at the instance level. The key of de-noising autoencoder is to perform some augmentations to input data x like (1) while the output remains identical to the unaugmented input source.

$$x_{aug} = f_{aug}(x) \tag{1}$$

Like the regular autoencoder, de-noising autoencoder consists of two main components, the encoder and decoder, whose neural network structures mirror to each other. The most informative descriptor is generated at the bottleneck layer. Therefore, the output of encoder

$$z = f_{En}(x_{xug}) = f_{En} \circ f_{aug}(x) \tag{2}$$

is the latent representation of the input data.

The reconstruction of augmented input data is

$$\hat{x} = f_{De}(z) = f_{De} \circ f_{En} \circ f_{aug}(x) \tag{3}$$

and our goal is to minimize the difference between input data and output data, i.e.

$$\min L_2 = \Sigma_i \|x_{(i)} - \hat{x}_{(i)}\|_2 \tag{4}$$

B. Synthetic Dataset

The time-consuming issue of annotating real 6D pose dataset for any model can be mitigated by the photo-realistic rendering technique. He Wang et al. [17] used context-aware mixed reality technique to generate synthetic 6D pose datasets on real scenes. By contrast, Sundermeyer et al. [6] replaced the background image with Pascal VOC image [18], regardless of contexts.

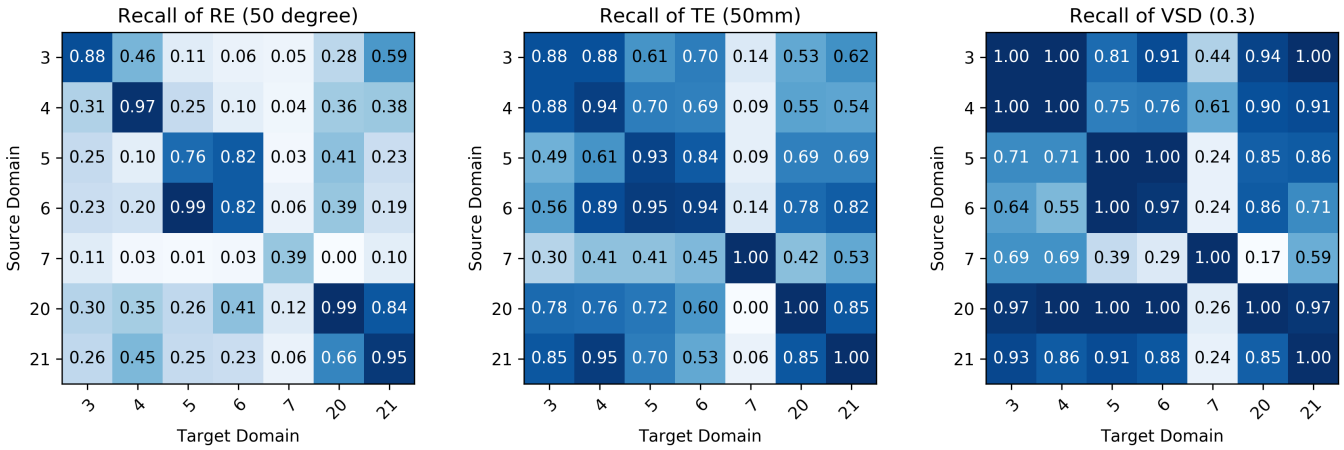
In this paper, we generate synthetic views of an object spherically, i.e. the camera is placed at a constant distance to the object while the azimuth and elevation angle varies, along with in-plane rotations to cover the rotation space as much as possible.

However, the synthetic image usually looks unreal. To bridge the gap between simulation and reality, we adopt the domain randomization (DR) [19] technique by performing random lights, contrast normalization, Gaussian blur, color channel shuffle and occlusion to synthetic view images. Sundermeyer et al. [6] justified DR to be effective in improving the estimation accuracy, but they did not perform the experiments at category level.

C. Category-level Pose Estimation

To mitigate the intra-class variation problem caused by heterogeneous textures, colors and illuminations, we augment our input data with domain randomization (DR) technique, so that the de-noising autoencoder will generalize to these characteristics. To address the problem of variations in object dimension, we perform perspective transformation as in Fig. 5, i.e. alter the aspect ratio of objects, so that the resemblance of geometry within a category is taken into account. In [6], the depth estimation is obtained by

$$z_{est} = z_{syn} \times \frac{l_{\text{syn\_max\_sim}}}{l_{\text{test}}} \times \frac{f_{\text{test}}}{f_{\text{syn}}} \tag{5}$$



(a) Recall of RE, threshold 50 degree. (b) Recall of TE, threshold 50 mm. (c) Recall of VSD, threshold 0.3.

Fig. 6. Recall of Rotational Error, Translational Error and VSD. Source domain target domain are both from TOYOTALIGHT cup, object id: 3, 4, 5, 6, 7, 20, 21. To increase the difficulty for calculating the Visible Surface Discrepancy, we use the source model on purpose, which is *inexact* for the target model.

(a) Estimated pose with top 3 similarity scores (b) Scene crops and de-noising autoencoder reconstruction

Fig. 7. Source domain LINEMOD bowl, object id 03. Target domain TOYOTALIGHT bowl, object id 11.

where z_{est} is the rendering distance of synthetic view. $l_{\text{syn\_max\_sim}}$ and l_{test} are the diagonal lengths (unit: pixel) of synthetic object with maximal similarity and test object, respectively. f_{test} and f_{syn} are the focal lengths of test camera and synthetic views.

However, this is only applicable at instance level, i.e. the source object and target object are of identical dimension. To facilitate the category-level depth estimation, we modify (5) to

$$z_{est} = z_{syn} \times \frac{l_{\text{syn\_max\_sim}}}{l_{\text{test}}} \times \frac{f_{\text{test}}}{f_{\text{syn}}} \times \frac{d_{\text{test}}}{d_{\text{syn}}} \tag{6}$$

which can be easily obtained from the pinhole camera model. d_{test} and d_{syn} (unit: mm) are the 3D diagonal lengths of test object (target domain) and synthetic object (source domain), respectively. Equation (6) has taken the dimension of source and target domain into consideration, making it possible to predict the depth of different objects within the same category.

III. Experiments

Fig. 3 illustrates the pipeline of our experiments.

A. Datasets

We train our de-noising autoencoder with the SIXD Challenge [20] Dataset TOYOTALIGHT object, such as box, bowl, can, milk, plate and cup. To train each de-noising autoencoder, we need only one object model as a *representative* for some object category, as the source domain. For the target domain, we manually pick objects listed in Fig. I from LINEMOD, TEJANI, and TOYOTALIGHT datasets.

B. Training Specifications

We use the network architecture of de-noising autoencoder in Fig. 2 from Sundermeyer et al. [6], with Xavier weight initialization [21], Adam optimizer [22] at the fixed learning rate of 0.0002. We rendered 10000 synthetic views for each 3D reconstruction model uniformly at a distance of 700mm. It takes approximately 1 hour on one NVIDIA GeForce GTX

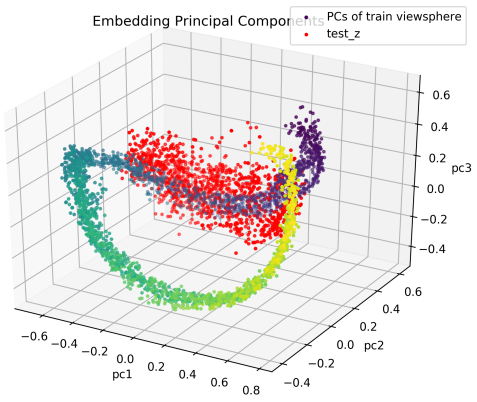


Fig. 8. Principal Component Analysis on embedding. Red dot stands for target domain (test embedding) LINEMOD bowl, object id 03. The dots in other colors stand for source domain (template embedding), TOYOTALIGHT bowl, object id 11. Bowl is an object with one axis of symmetry.

1080 Ti GPU for 10000 iterations with batch size of 64 for each 3D model.

C. Metric and Evaluation

The error for an estimated pose is calculated by the Visible Surface Discrepancy (VSD) [23], which computes the error over the visible model surface. By definition, this error can handle ambiguous pose by treating them as the same. Instead of using the exact target model to compute the VSD, we deliberately use the source model, which is *inexact* for the target object. Hence, our evaluation of VSD is **more stringent** than [13]. We also compute the translational error (TE) and rotational angular error (RE) as in [9]. Although the rotational error can not properly take care of the ambiguous pose, we still report it for reference.

For evaluation, we calculate the recall of VSD with depth tolerance $20mm$ and 10% object visibility. The pose estimation is considered correct if the value of VSD is less than 0.3. For TE and RE, the estimated pose will be regarded as correct if the value is less than $50mm$ and 50 degree respectively.

D. Analysis

Generally speaking, from Fig. 5 and Table. I we can see that when the source and target model are identical, the recall is higher than that of when they are different. This is the case due to instance-level pose estimation imposes more constraints than the category-level one. In Fig. 6 we can see that TOYOTALIGHT cup with object id 7 is estimated not so well as other cups, because the appearance of which is far from similar to the rest cups. Furthermore, this cup does not have a cup handle, the part which could help reduce the pose ambiguity.

In Table. II we compare our recall of VSD metric with other instance-level methods [13]. Although we use *inexact* models for calculating VSD and the unfairness of comparing

TABLE I

RECALL FOR DIFFERENT ERRORS, SOURCE AND TARGET OBJECTS

TYO: TOYOTALIGHT; TLS:TLESS; TEJ:TEJANI; LM:LINEMOD.

Source	Target	RE	TE	VSD
Box, TYO16	TLS25	4.97	53.65	23.94
	TLS26	5.57	50.4	23.56
	TLS27	12.06	11.03	67.73
	TLS28	13.24	30.11	81.17
	TLS29	2.81	46.79	21.04
	LM10	2.63	60.97	36.95
Bowl, TYO10	TYO10	43.75	100.0	100.0
	TYO11	10.0	75.0	100.0
	LM03	23.2	72.91	99.92
Bowl, TYO11	TYO10	37.5	72.5	76.25
	TYO11	72.5	80.0	100.0
	LM03	26.03	78.02	99.92
Can, TYO17	TYO17	88.0	97.33	86.67
	TYO18	17.65	85.88	98.82
Can, TYO18	TYO17	20.0	81.33	94.67
	TYO18	83.53	100.0	100.0
Milk, TYO15	TEJ04	24.15	31.46	34.47
Plate, TYO12	TYO12	18.75	97.5	100.0
	TYO13	10.0	85.0	100.0
Plate, TYO13	TYO12	10.0	90.0	100.0
	TYO13	28.75	83.75	100.0
Cup, TYO03	LM07	2.82	30.24	20.97
Cup, TYO04		4.6	33.71	19.03
Cup, TYO05		4.6	58.06	15.32
Cup, TYO06		14.44	73.31	13.23
Cup, TYO07		15.48	26.61	47.9
Cup, TYO20		8.31	70.0	14.03
Cup, TYO21		5.56	53.95	14.35

category-level method to the instance-level one, our results still outperform others for some objects.

To show how the embedding learned by the de-noising autoencoder could handle ambiguous pose caused by symmetry or self-occlusion, we perform Principal Component Analysis (PCA) [24] to map the embedding to 3-dimension space. Fig. 8 shows the PCA result of training viewsphere for the bowl, an object with one axis of symmetry. The "yellow-green-purple strip" is the synthetic views' embedding after PCA. We colorize each training viewpoint according to the azimuth and elevation angle value, hence we expect points with similar color should be close while dissimilar ones be far. At the meantime, these points should distribute continuously on some manifold since we generate azimuth and elevation angle uniformly.

Apparently the "strip" satisfies our expectation: **1) similar color points are grouped together while dissimilar ones are far away. 2) distribution in continuity.** Therefore, the ambiguous poses will be estimated as almost the same pose thanks to the adjacency of embedding for similar poses. Our test image embedding is colored red and it lies near our "strip", which means the de-noising autoencoder does learn

TABLE II

RECALL COMPARISON FOR VSD

We compare our category-level result with the instance-level one [13], but we still manage to exceed some results.

Source	Box, TYO16						Bowl, TYO10,11	Milk, TYO15
Method\Target	TLS25	TLS26	TLS27	TLS28	TLS29	LM10	LM03	TEJ04
Vidal-18 [13]	43	58	62	69	69	**97**	91	98
Drost-10-edge [13]	**47**	69	61	80	**84**	94	94	**100**
Drost-10 [13]	28	51	32	60	81	96	89	**100**
Hodan-15 [13]	47	**72**	45	73	74	97	79	74
Brachmann-16 [13]	16	27	17	13	6	79	76	88
OURS	23.94	23.56	**67.73**	**81.17**	21.04	36.95	**99.92, 99.92**	34.47

a generalizable embedding at the bottleneck layer.

IV. CONCLUSIONS

In this paper, we have explored the embedding ability of de-noising autoencoder for the 3D rotation at the category level. To estimate the 3D translation, we also extended the depth estimation from the instance level to the category level by considering the 3D diagonal length ratio between the source and the target objects. We evaluated our approach on some SIXD Challenge datasets and even under our stringent criteria we outperformed some previous results. Analysis of embedding manifold explained why de-noising autoencoder can resist pose ambiguity to some extent. Future work may include using the depth map to perform category-level Iterative Closest Point (ICP) [25].

ACKNOWLEDGMENT

This work is supported in part by the National Natural Science Foundation of China under Grant 81671854, 61773378, U1713222, and U1806204. National Key R&D Program of China (grant 2017YFB1300202). Science and Technology on Space Intelligent Control Laboratory for National Defense, No.KGJZDSYS-2018-09; in part by Equipment Pre-research Field Fund under Grant 61403120407.

REFERENCES

[1] "Amazon picking challenge," http://amazonpickingchallenge.org/.

[2] Y. Wang, R. Wang, S. Wang, M. Tan, and J. Yu, "Underwater bio-inspired propulsion: From inspection to manipulation," *IEEE Transactions on Industrial Electronics*, 2019.

[3] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012.

[4] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *Computer Vision & Pattern Recognition*, 2015.

[5] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T. K. Kim, "Pose guided rgbd feature learning for 3d object pose estimation," in *IEEE International Conference on Computer Vision*, 2017.

[6] M. Sundermeyer, Z. C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," 2018.

[7] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *Robotics: Science and Systems (RSS)*, 2018.

[8] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *IEEE International Conference on Computer Vision*, 2017.

[9] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," 2017.

[10] S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou, "Pvnet: Pixel-wise voting network for 6dof pose estimation," 2018.

[11] C. Sahin and T.-K. Kim, "Category-level 6d object pose recovery in depth images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[12] S. Song and J. Xiao, "Sliding shapes for 3d object detection in depth images," in *European conference on computer vision*. Springer, 2014, pp. 634–651.

[13] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, and X. a. Zabulis, "Bop: Benchmark for 6d object pose estimation," 2018.

[14] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," pp. 1096–1103, 2008.

[16] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," pp. 205–220, 2016.

[17] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," 2019.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," pp. 23–30, 2017.

[20] "Sixd challenge," http://cmp.felk.cvut.cz/sixd/challenge_2017/.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," pp. 249–256, 2010.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[23] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis *et al.*, "Bop: Benchmark for 6d object pose estimation," pp. 19–35, 2018.

[24] I. T. Jolliffe, "Principal component analysis," *Journal of Marketing Research*, vol. 87, no. 100, p. 513, 2002.

[25] P. J. Besl and N. D. Mckay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.