 DOI: 10.3217/978-3-85125-663-5-46

Towards Object Detection and Pose Estimation in Clutter using only Synthetic Depth Data for Training

Stefan Thalhammer, Timothy Patten and Markus Vincze

***Abstract*— Object pose estimation is an important problem in robotics because it supports scene understanding and enables subsequent grasping and manipulation. Many methods, including modern deep learning approaches, exploit known object models, however, in industry these are difficult and expensive to obtain. 3D CAD models, on the other hand, are often readily available. Consequently, training a deep architecture for pose estimation exclusively from CAD models leads to a considerable decrease of the data creation effort. While this has been shown to work well for feature- and template-based approaches, real-world data is still required for pose estimation in clutter using deep learning. We use synthetically created depth data with domain-relevant background and randomized augmentation to train an end-to-end, multi-task network to detect and estimate poses of texture-less objects in cluttered real-world depth images of an arbitrary amount of objects. We present experiments and ablation studies on the architectural design choices and data representation with the LineMOD dataset.**

I. Introduction

Assembly systems in manufacturing are subject to increasing number of variants, smaller lot sizes and shorter life cycles. As such, the application of assistance or robotic systems is expected to reduce error rate and increase capacity [6]. Typically, the task of assistance systems in an industrial context is robust object detection as well as pose estimation. However, developing methods that deliver accurate estimates, especially for texture-less objects, is still an open research problem.

Recently deep learning advanced the state of the art for computer vision tasks, however, the advent of deep networks for 3D pose estimation has yet to be fully realized [9]. While deep networks achieve superior performance, they require a huge amount of training data [12]. Capturing and annotating these data is time and labour consuming, often requiring physical instances, which is problematic in fast paced manufacturing environments. Industrial applications typically have CAD data readily available, therefore, we propose to take advantage of this by directly training models for pose estimation of texture-less objects using only synthetic depth images for training.

Accurate pose estimation systems consist of multiple steps, firstly creating initial pose candidates and subsequently refining these using one or more refinement and verification steps. In this work we address the task of creating an initial pose estimate for further refinement.

[1]All authors are with the Faculty of Electrical Engineering and Information Technology, TU Wien, 1040 Vienna, Austria {sthalham, patten, vincze}@acin.tuwien.ac.at

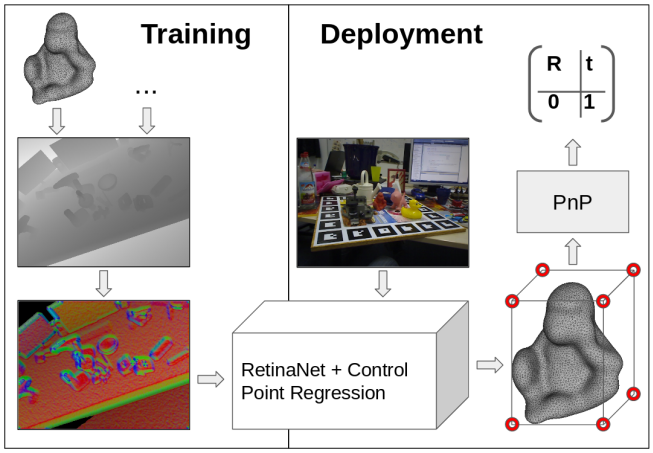

Fig. 1. Control point regression and pose computation in real-world images, trained using rendered and augmented data.

Feature- and template-based approaches for pose estimation employ meshes or point clouds to create templates or hash tables in order to detect objects and estimate their pose at runtime [2], [4], [10]. Consequently, these methods require no real-world data for training. Current deep learning approaches do not close the domain gap, i.e. traversing from synthetic to real-world data without a decrease in performance, and therefore need real-world data during training time. We address the task of training deep pose estimators only from synthetic depth data by rendering and augmenting these data in terms of background information and sensor noise through random shape perturbations.

Pose estimation is a non-trivial task for learning-based approaches, consequently strong approaches tend to train separate models for the detection and the subsequent pose estimation. Often the models for pose estimation are also trained separately for distinct classes [19], [22], [24]. However, end-to-end learning, i.e. training and deploying multiple stages of a vision pipeline at once, is desired to reach a high frame rate. Additionally, when deep architectures are trained on multiple objectives, i.e. in a multi-task fashion, the learned features are stronger, which has been shown to be beneficial for each individual task [5]. Especially when employing pre-trained models in a domain different from RGB, e.g. on depth data, retraining the backbone with additional guidance is desired to create stronger features. Our multi-task, end-to-end models for pose estimation are consequently trained with the capacity to estimate the poses of different classes simultaneously.

In summary, we propose a method for texture-less ob-

ject pose estimation in real-world depth images using only synthetic data for training. Figure 1 outlines our proposed approach.

The contributions are the following:

- We present an approach for simultaneous object detection, classification and pose estimation in a multi-task, end-to-end manner, of an arbitrary number of texture-less objects in real-world depth images. For training we only need meshes of the desired objects.
- We present our findings by evaluating on a standard dataset, the LineMOD [8].

The remainder of the paper is structured as follows. Section 2 summarizes related work. The approach is described in section 3. Section 4 presents the results and evaluation. Section 5 concludes with a discussion.

II. Related Work

The attention of pose estimation research has recently shifted to texture-poor or texture-less objects. Currently, the domain is dominated by template or hand-crafted feature-based approaches [9]. However, estimating object poses using deep architectures is gaining popularity due to the state-of-the-art performance for other computer vision tasks.

While there are manifold approaches to estimate the poses using colored images, depth data is usually only used for refinement [27]. Only very few employ depth data only [17], [16], [22]. However, depth data already gives strong cues about the shape and consequently also about the pose of the object. A major advantage of using only depth data to train networks for pose estimation is the possibility to exclusively train models using CAD data, independent of the color variations of the manufactured object.

A. Classical Approaches

Point-Pair features create a strong basis for pose estimation pipelines. Point pairs are matched between the test scene and the provided models then stored in a hash table. Votes are accumulated to create hypotheses, subsequently refined using ICP and non-maxima are suppressed. Hypotheses are favoured when the detected 3D edges match the model contours. These approaches do not use RGB data and have multiples stages, subsequently removing or refining pose hypotheses [4], [26].

Template matching methods can also exhibit strong pose estimation results. Hodan *et al.* [10] use a sliding window with cascading evaluation. Pre-filtering differentiates between the object and background. Hypotheses are generated for every window by hashing. Hypotheses verification consists of verifying size, normals, gradients, depth template and color. Object pose refinement is initialized from the verified hypotheses using particle swarm optimization.

While these approaches usually yield strong pose estimates, they are slow compared to end-to-end learning based approaches and lack high detection performance.

B. Learning-based Approaches

Learning-based approaches yield strong results for some pose estimation tasks, but they are currently not on par with classical approaches.

Random forest approaches can be used to sample pose hypotheses, which are used to choose and iteratively refine promising pose estimates [1], [23].

A common practice for pose estimation using deep learning is to treat the translation and the rotation part of the pose separately [11], [14], [19], [22], [27]. While the center of the detected bounding boxes already results in feasible translation estimates in image space [11], translation regression is desired when dealing with occlusions [22], [27]. When estimating the rotational part of the pose separately either regression [27] or classification [11], [22] can be employed. While regression of the rotation is computationally more efficient and natural due to the smooth representation space, classification yields better results in practice [11], [22].

For pose estimation using depth only, pixel-wise segmentation can be employed to create masks and then to be matched against previously computed templates [16] resulting in similar performance as classical approaches.

One of the strongest approaches for object pose estimation using deep learning is the regression of virtual control points [3], [19], [24], i.e. regressing a 3D bounding box projected into image space and alike. The regressed control points are used to solve Perspective-n-Points (PnP) in order to obtain a pose estimation. This approach is used for RGB images and models are trained separately for each object or even decoupled from detection. Considering the task of pose estimation as a translation regression problem is promising because CNNs exhibit translation equivariance between image and feature space.

We employ one model for detection, classification and pose estimation, independent of the amount of objects of interest. Our approaches modifies RetinaNet [13] to include pose estimation in their one-staged architecture. We use RetinaNet due to the very strong object detection performance on diverse datasets and its fast computation, running with approximately ten frames per second (fps). We consequently regard pose estimation as a multi-task, end-to-end learning approach, using only translation regression and subsequent PnP for pose calculation. Compared to other pose estimation approaches using depth images, our method is one-staged, uses no refinement and deals with diverse objects simultaneously at approximately five fps.

III. 6D Pose Estimation from Synthetic Data

We render synthetic depth data in Blender from a virtual scene resembling the area of deployment of our model. These data are subsequently augmented and annotated using a randomized noise model and are used for supervised training. We base our architecture on RetinaNet [13] and add an additional branch in order to enable multi-task, end-to-end 6D object pose estimation in complete scenes. The additional branch takes the features concatenated by the feature pyramid network as inputs and outputs n virtual control points as

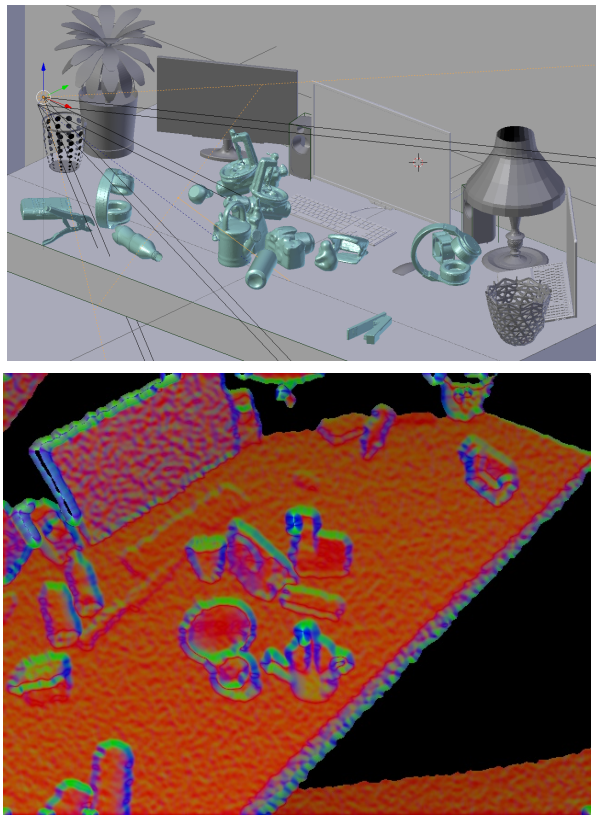

Fig. 2. Virtual scene to render synthetic training data from (top). Augmented synthetic depth image used for training (bottom).

defined in [3]. RetinaNet is currently considered one of the strongest object detectors and additionally exhibits tight bounding box estimates, thus ideally fitted for control point regression.

Since CNNs yield translation equivariance between image and feature space it is reasonable to regard pose estimation as regression tasks, in the context of deep learning. The authors of [24] showed that a similarly simple approach for RGB-data can achieve state-of-the-art results without limiting general applicability.

A. Training Dataset Creation

We use only synthetically created training data and deploy our model on real-world depth scans. We create synthetic depth data with a diverse scene setup and various background information and additionally apply noise heuristics in order to produce training data with high variation regarding views and occlusion patterns [25]. This has been shown to generate high quality data to train deep architectures for object detection and classification. An example for a virtual scene can be seen in the top image of Figure 2.

1) Data Rendering: We render 15,000 training images of virtual scenes exhibiting the expected variations of the area of deployment. For each image, we randomly place five to eight objects of interest with repetition. The objects are annotated with a bounding box, 6D pose and pixel-level class correspondences. The camera pose is sampled similar to the expected poses in the test set.

The output of the synthetic data creation step is a depth image, a binary mask indicating visible image regions and a mask indicating pixel-level class correspondences. The binary mask provides information about image regions with invalid depth values depending on the imaging geometry of infrared depth sensors.

2) Dataset Creation: The synthetic dataset used for training is created by combining the outputs of the rendering step.

The binary mask is applied to the synthetic depth images using randomized morphological operations. This results in missing image regions similar to real-world depth scans. Blur is added to minimize the discrepancy between depth gradients in the real-world and synthetic images. The synthetic depth values are rounded to the nearest quantization value based on the hypothesized sensor's depth resolution. This operation reduces the domain shift between synthetic and real depth images. Additional noise is added to the quantized depth values using an offset chosen randomly from a Gaussian distribution, assuming non-linearly increasing noise. Further randomness of the appearance of occluded scene parts, depth and lateral noise is added by warping the depth images through the application of pixel offsets using the Perlin noise technique [18], which was shown to significantly improve the performance of trained models [25]. The augmentation process is sampled twice per rendered image to create a dataset of approximately 30,000 images.

B. Network Architecture

We use RetinaNet[1] [13] with ResNet-50 [5] backbone and pretrained on ImageNet [20] as feature extractor and detector. We add an additional network branch for control point regression, parallel to the classification and detection branches.

1) Data Representation for Pose Estimation: We regress eight control points to exactly encapsulate the object's dimensions in 3D. In general, an arbitrary number of control points can be chosen and regressed. Those points are virtual, i.e. they do not represent actual object parts, thus can be chosen arbitrarily in the objects' coordinate frame. Using the camera intrinsics and the calculated corresponding object pose these points are projected into image space.

The design of the additional branch is based on RetinaNet's bounding box regression branch. We slightly modify it by adding *l2* regularization of the weights of every convolution layer with the hyperparameter set to 0.001. We perform experiments using other penalties, dropout and batch normalization but that resulted in decreased performance. The 16 values representing the x and y components of the eight control points are regressed for every object class (n) separately. The architecture of our control point estimation branch is shown in Figure 3.

2) Loss: The overall loss to minimize is defined as

$$L = L_{box} + L_{cls} + L_{box3D} \quad (1)$$

[1]https://github.com/fizyr/keras-retinanet

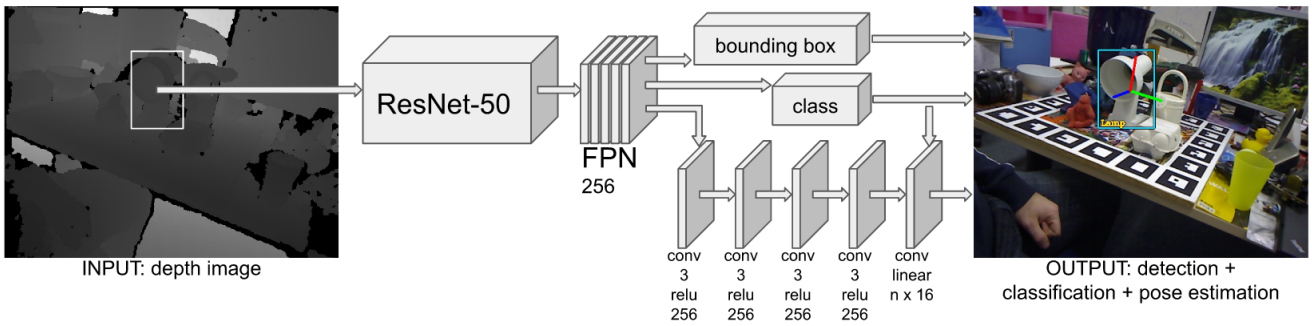

Fig. 3. Our multi-task, end-to-end network architecture.

where L_{box}, L_{cls} and L_{box3D} are the losses for bounding box regression, object classification and control point regression, respectively. We use smoothed *l1* loss for bounding box regression and focal loss for classification. The control point regression loss is formally, per image, defined as

$$L_P = \frac{1}{m} \sum_{o \in gt} \left|\left| Proj_T(cp) - Proj_{\hat{T}}(cp) \right|\right|^k \qquad (2)$$

where m is the total number of object instances o in the ground truth gt of an image, $Proj_T$ and $Proj_{\hat{T}}$ are the projection and the estimation projection of the control points cp onto the image plane and k the desired norm. As norm we use smooth *l1*.

We weight the contribution of each of the loss parts differently. Our experiments showed that weighting L_{box3D} such that its magnitude is twice the magnitude of L_{box} and four times the magnitude of L_{cls}, results in good recall and precision regarding detections and reasonable pose estimates.

The estimated control points during test time are reprojected into 3D space and the object's pose is simultaneously estimated using PnP. For our purpose we use the iterative RANSAC based algorithm.

3) Data Augmentation: In order to prevent the network from overfitting to the limited amount of training data we apply extensive data augmentation of the training images. Every input image is randomly augmented online using a superposition of translation and scaling up to 20 percent each.

IV. Experiments

All the experiments are conducted on the LineMOD dataset [8]. LineMOD contains approximately 1,100 test images for each of the 13 dataset objects. Each object is placed in a heavily cluttered scene and annotated with bounding box, class and 6DoF pose. We provide ablation studies, specific for the task at hand. Comparing image preprocessing, regularization strategies and possible loss functions for the control point regression branch. In order to provide a reasonable comparison against the state of the art we compare against [4], [7] and [23].

A. Experimental Setup

For testing we use only the depth images of the LineMOD dataset that are captured using a Microsoft Kinect V1. Images are converted to three channel RGB images, coloured based on the depth gradient using the approach of Nakagawa *et al.* [15]. The ablation study shows the benefit of this. Image regions with missing depth values are inpainted using OpenCV[2] and depth cuts are applied to image regions farther than two meters.

Our networks are trained using the Adam optimizer with adaptive learning rate. Ablation studies are trained for 20 epochs using 10,000 images. Comparison against state of the art is trained for 100 epochs with 30,000 images. We use a batch size of one and an initial learning rate of 10^{-5}. We choose the best performing model after the above mention amount of epochs to provide comparisons. All networks are trained on a Nvidia GeForce GTX 1080.

B. Ablation Studies

We perform three studies to adapt RetinaNet to our needs. Firstly we compare different dataset augmentations and depth image representations, secondly we evaluate which loss to use for bounding box regression and thirdly we show an ablation study regarding regularization applied by the control point regression branch. All studies are performed on a validation set of 2400 images, taken uniformly from all classes of the LineMOD dataset.

1) Image Representation: We compare different options for of the augmentations applied to the depth images, as well as possibilities for depth to three channel image conversion.

Table I provides results in terms of recall and precision of detections with an Intersection over Union (IoU) higher than 0.5, and percentage of rotation estimates below a five degree deviation from the ground truth. *Depth* refers to repeating the depth images three times and converting it to eight bit, *rgb* refers to color coding the depth images based on the normal direction [15]. Options for augmentation are either *perlin*, which refers to only augmenting the synthetic training images by removing occluded image regions due to the imaging geometry and warping pixel locations using Perlin noise, and *full*, which refers to additionally adding blur and depth noise and quantizing depth values as described in section III-A.

Using color coded depth images with full augmentation applied shows best detection and rotation estimation results.

[2]https://opencv.org/

TABLE I

EXPERIMENTS REGARDING IMAGE DATA REPRESENTATION AND AUGMENTATION APPLIED.

Representation	*depth*		*rgb*	
Augmentation	*perlin*	*full*	*perlin*	*full*
Recall	83.61	82.72	86.67	**89.77**
Precision	**94.65**	89.54	93.91	93.59
Rotation $< 5°$	6.93	5.07	6.76	**7.33**

TABLE II

LOSS FUNCTIONS USED FOR CONTROL POINT REGRESSION AND THEIR INFLUENCE ON THE DERIVED ROTATION ESTIMATES.

Loss	*mse*	*l1*	*smooth l1*
Rotation $< 5°$	6.03	4.78	**7.33**

2) Loss Function: Using a pose regression branch similar to the bounding box regression branch suggests to use a similar loss function.

Table II presents results of different loss functions used for regressing the virtual control points. Care was taken to weight the individual loss parts in a way to preserve the above mentioned ratio between bounding box, classification and pose losses. The percentage of rotation estimates below a five degree deviation from the ground truth is provided for mean squared error (*mse*), absolute error (*l1*) and Huber loss (*smooth l1*). *Smooth l1* provides the strongest control point estimates evaluated using rotation estimation.

3) Regularization: Regularization reduces the generalization error, thus reducing the model's performance discrepancy between the training and the validation/test set. Since our data domains for source and target are very different it is not straight forward to decide which regularization strategy to apply. Table III provides information about regularization applied and their influence on the networks performance. All regularization strategies applied here are only applied to the control point regression branch. Batch normalization (*bn*) is applied to each convolution layer's output except from the last, weight decay (*wd*) is applied to all the weights of the convolution kernels and dropout (*do*) tested here is applied to the inputs of the last convolution layer with a probability of 20 percent.

The metric *5cm 5°* refers to the metric defined in [21], *6D pose* refers to the metric defined by [8], where we evaluate on ten percent of the mesh-model diameter, and *proj. 2D* refers to the reprojection of the object mesh to the image using the estimated pose. The pose is considered as true if the average pixel difference is smaller than a threshold. For this we use five pixels. Results show that only *l2* weight decay with a hyperparameter of 0.001 improves results on the validation set.

C. Comparison Against the State of the Art

For evaluation against the state of the art we use the metric defined in [8] as well as the F1-score from the harmonic mean of the precision and recall as in [23]. Unlike [23] we train only one model and not separate models for every object. While [23] and [7] use RGB and depth, [4] only uses depth data. Consequently, we consider [4] as the most relevant method to compare against. Table IV provides a comparison of our approach against the state of the art.

TABLE III

REGULARIZATION APPLIED BY THE NETWORK, TESTED ON DIFFERENT METRICS.

Metric	*bn*	*wd(0.01)*	*wd(0.001)*	*wd(0.0001)*	*do(0.2)*
Rot. $< 5°$	0.0	6.84	**7.85**	7.03	1.83
5cm 5	0.0	3.36	**5.02**	3.76	0.0
6D pose	0.0	4.82	**6.62**	6.17	0.35
proj. 2D	0.0	18.06	22.33	**23.48**	0.5

TABLE IV

F1-SCORE COMPARISON OF OUR METHOD AGAINST COMMON STATE-OF-THE-ART METHODS.

Method	LINEMOD[7]	Drost[4]	Tejani[23]	ours
Ape	53.3	62.8	85.5	34.0
Benchvise	84.6	23.7	96.1	52.2
Driller	69.1	59.7	90.5	31.6
Cam	64.0	51.3	71.8	52.4
Can	51.2	51.0	70.9	51.2
Iron	68.3	40.5	73.5	46.5
Lamp	67.5	77.6	92.1	26.0
Phone	56.3	47.1	72.8	66.2
Cat	65.6	56.6	88.8	60.6
Holepuncher	51.6	50.0	87.5	46.6
Duck	58.0	31.3	90.7	44.6
Eggbox	86.0	82.6	74.0	54.0
Glue	43.8	38.2	67.8	30.5
Average	63.0	51.7	81.7	46.8

Our method exhibits comparable results to Drost *et al.* [4] when taking the false-positive rate into account. When considering detections above 0.5 IoU as true our method exhibits a recall and precision of 96.71 and 94.43 percent respectively on the LineMOD dataset.

Figure 4 shows control point estimation of the object *Glue* on the left and the corresponding ground truth on the right. A severely distorted 3D bounding box estimation of the object is visible, the box appears to vanish in one dimension. This happens often for the object *Glue*, leading to the conclusion that for objects with a small size along one dimension the control points have to be chosen significantly higher than the corresponding dimension.

Figure 5 shows the 3D bounding box of the object *Lamp*, defined by the estimated control points, on the left and again the corresponding ground truth on the right. A detection with good alignment of the estimated 3D box and pose, with respect to the ground truth, is visible.

V. CONCLUSION

In this paper we presented a deep learning architecture for multi-task, end-to-end 6D object pose estimation for an arbitrary number of objects from only depth images. The architecture was trained entirely from synthetic data that is generated to resemble real-world data. Experiments

Fig. 4. Warped 3D box detection of the object *glue*

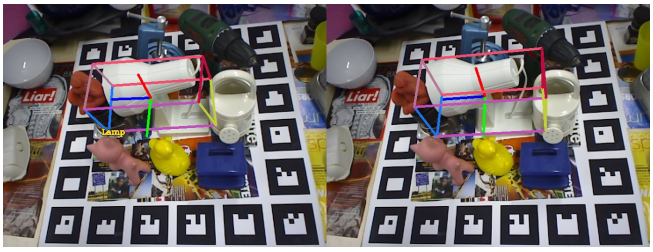

Fig. 5. 3D box detection of the object *lamp*

with the LineMOD dataset showed promising results. Our ablation studies provide valuable information for detection, classification and pose estimation of texture-less objects in clutter.

Future work will tackle the improvement of the 3D bounding box regression results. Experiments with other data modalities will also be conducted. We will furthermore investigate the benefits of enforcing orthogonality on the boxes. Additional architecture modifications will be tested to disentangle the control point estimation per object further in order to enhance pose estimation. Other directions for future work include addressing object symmetries and tuning the parameters for the generation of the synthetic training data to randomize the applied noise more specifically to the desired sensor.

REFERENCES

[1] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, *et al.*, "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*, 2016, pp. 3364–3372.

[2] A. G. Buch, L. Kiforenko, and D. Kraft, "Rotational subgroup voting and pose clustering for robust 3d object recognition," 10 2017, pp. 4137–4145.

[3] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "A novel representation of parts for accurate 3d object detection and tracking in monocular images," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4391–4399.

[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 998–1005.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] S. Hinrichsen, D. Riediger, and A. Unrau, "Assistance systems in manual assembly," in *Proceedings 6th International Conference on Production Engineering and Management*, 2016, pp. 3–13.

[7] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *2011 international conference on computer vision*. IEEE, 2011, pp. 858–865.

[8] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[9] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "Bop: Benchmark for 6d object pose estimation," in *Computer Vision – ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 19–35.

[10] T. Hoda, X. Zabulis, M. Lourakis, . Obdrlek, and J. Matas, "Detection and fine 3d pose estimation of texture-less objects in rgb-d images," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2015, pp. 4421–4428.

[11] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision*, pp. 2999–3007, 2017.

[14] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6d pose refinement in rgb," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 800–815.

[15] Y. Nakagawa, H. Uchiyama, H. Nagahara, and R.-I. Taniguchi, "Estimating surface normals with depth image gradients for fast and accurate registration," in *3D Vision, International Conference on*. IEEE, 2015, pp. 640–647.

[16] K. Park, T. Patten, J. Prankl, and M. Vincze, "Multi-task template matching for object detection, segmentation and pose estimation using depth images," in *International Conference on Robotics and Automation*, 2019.

[17] K. Park, J. Prankl, M. Zillich, and M. Vincze, "Pose estimation of similar shape objects using convolutional neural network trained by synthetic data," in *Proceedings of the OAGM-ARW Joint Workshop*, 5 2017, pp. 87–91.

[18] K. Perlin, "Improving noise," in *ACM Transactions on Graphics*, vol. 21. ACM, 2002, pp. 681–682.

[19] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *The IEEE International Conference on Computer Vision*, Oct 2017.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[21] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2930–2937.

[22] J. Sock, K. Kim, C. Sahin, and T. Kim, "Multi-task deep networks for depth-based 6d object pose and joint registration in crowd scenarios," in *Proceedings of British Machine Vision Conference*, 7 2018.

[23] A. Tejani, R. Kouskouridas, A. Doumanoglou, D. Tang, and T. Kim, "Latent-class hough forests for 6 dof object pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 119–132, Jan 2018.

[24] B. Tekin, S. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," 06 2018, pp. 292–301.

[25] S. Thalhammer, K. Park, T. Patten, M. Vincze, and W. Kropatsch, "Sydd: Synthetic depth data randomization for object detection using domain-relevant background." TUGraz OPEN Library, 2019, pp. 14–22.

[26] J. Vidal, C.-Y. Lin, and R. Marti, "6d pose estimation using an improved method based on point pair features," 04 2018, pp. 405–409.

[27] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2018.