CenDerNet: Center and Curvature Representations for Render-and-Compare 6D Pose Estimation

Peter De Roovere[1,2(✉)], Rembert Daems[1,3,4], Jonathan Croenen[2],
Taoufik Bourgana[5], Joris de Hoog[5], and Francis wyffels[1]

[1] IDLab-AIRO, Ghent University – imec, Ghent, Belgium
peter.deroovere@ugent.be
[2] RoboJob, Heist-op-den-Berg, Belgium
[3] Department of Electromechanical, Systems and Metal Engineering,
Ghent University, Ghent, Belgium
[4] EEDT-DC, Flanders Make, Gent, Belgium
[5] DecisionS, Flanders Make, Leuven, Belgium

Abstract. We introduce CenDerNet, a framework for 6D pose estimation from multi-view images based on center and curvature representations. Finding precise poses for reflective, textureless objects is a key challenge for industrial robotics. Our approach consists of three stages: First, a fully convolutional neural network predicts center and curvature heatmaps for each view; Second, center heatmaps are used to detect object instances and find their 3D centers; Third, 6D object poses are estimated using 3D centers and curvature heatmaps. By jointly optimizing poses across views using a render-and-compare approach, our method naturally handles occlusions and object symmetries. We show that CenDerNet outperforms previous methods on two industry-relevant datasets: DIMO and T-LESS.

Keywords: Object detection · 6D object pose estimation · Industrial robotics · Render-and-Compare · Curvature maps

1 Introduction

1.1 Context

6D pose estimation is an essential aspect of industrial robotics. Today's high-volume production lines are powered by robots reliably executing repetitive movements. However, as the manufacturing industry shifts towards high-mix, low-volume production, there is a growing need for robots that can handle more variability [2]. Estimating 6D poses for diverse sets of objects is crucial to that goal.

Manufacturing use cases present unique challenges. Many industrial objects are reflective and textureless, with scratches or saw patterns affecting their appearance [4,31]. Parts are often stacked in dense compositions, with many occlusions. These densely stacked, reflective parts are problematic for existing depth sensors. In addition, object shapes vary greatly, often exhibiting symmetries leading to ambiguous poses. Many applications require sub-millimeter precision and the ability to integrate new, unseen parts swiftly.
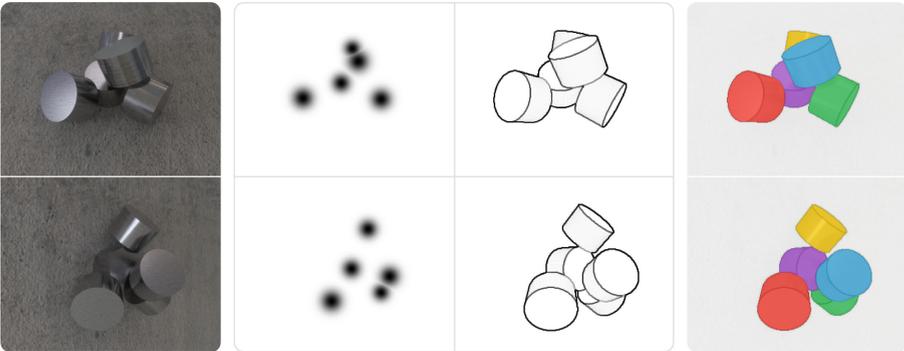

Fig. 1. Multi-view input images (left) are converted to center and curvature heatmaps (center), used to estimate 6D object poses (right).

This work presents a framework for 6D pose estimation targeted to these conditions. Our approach predicts object poses for known textureless parts from RGB images with known camera intrinsics and extrinsics. We use multi-view data as monocular images suffer from ambiguities in appearance and depth. In practice, images from multiple viewpoints are easily collected using multi-camera or hand-in-eye setups. This setup reflects many real-world industrial use-cases.

1.2   Related Work

Recent progress in 6D pose estimation from RGB images was demonstrated in the 2020 BOP challenge [10]. Convolutional neural networks (CNNs) trained on large amounts of synthetic data are at the core of this success. Many recent methods consist of three stages: (1) object detection, (2) pose estimation, and (3) refinement, using separate neural networks for each stage. While most approaches operate on monocular RGB images, some focus on multi-view data.

Object Detection. In the first stage, CNN-based neural networks are used to detect object instances. Although these detectors typically represent objects by 2D bounding boxes [6,7,17,22], 2D center points can be a simple and efficient alternative [33].

Pose Estimation. In the second stage, the 6D pose of each detected object is predicted. Classical methods based on local features [1] or template matching [8] have been replaced by learning systems. CNNs are used to detect local

features [14,25,26,30] or find 2D-3D correspondences [18,19,29,32]. Crucial aspects are the parameterization of 6D poses [34] and how symmetries are handled [20].

Refinement. In the final stage, the estimated poses are iteratively refined by comparing object renders to the original image. This comparison is not trivial, as real-world images are affected by lighting, texture, and background changes, not captured in the corresponding renders. However, CNNs can be trained to perform this task [14,16,32].

Multi-view. Approaches for multi-view 6D object pose estimation extend existing single-view methods. First, poses hypotheses are generated from individual images. Next, these estimates are fused across views [14,15].

1.3   Contributions

We present CenDerNet, a framework for 6D pose estimation from multi-view images based on center and curvature representations. First, a convolutional neural network is trained to predict center and curvature heatmaps. Second, center heatmaps are used to detect object instances and find their 3D centers. These centers initialize and constrain the pose esimates. Third, curvature heatmaps are used to optimize these poses further using a render-and-compare approach (Fig. 1).

Our system is conceptually simple and easy to use. Many existing methods consist of multiple stages, each with different training and tuning requirements. Our framework is more straightforward. We use a single, fully-convolutional neural network to convert RGB images to interpretable representations. Next, we use classical optimization techniques that require little tuning.

Using a render-and-compare approach, we jointly estimate poses for all objects in a scene across all viewpoints. As a result, our method naturally handles occlusions and object symmetries. We provide a GPU implementation of our render-and-compare method that allows evaluating over 2,000 scene pose estimates per second.

We evaluate CenDerNet using DIMO and T-LESS, two challenging, industry-relevant datasets. On DIMO, our method outperforms PVNet by a large margin. On T-LESS, CenDerNet outperforms the 2020 ECCV results of CosyPose, the leading multi-view method.

2   CenDerNet

Our system consists of three stages:

1. A convolutional neural network predicts center and curvature heatmaps for multi-view input images.
2. The predicted center heatmaps are converted to 3D center points. These 3D centers initialize and constrain the set of predicted object poses.
3. Object poses are optimized by comparing curvature renders to the predicted curvature heatmaps.
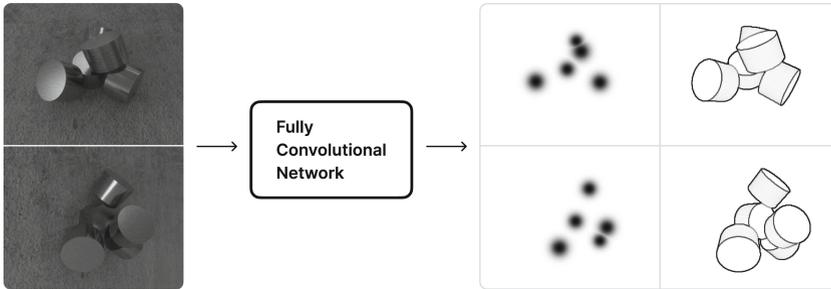
Fig. 2. Step 1: multi-view RGB images are converted to center and curvature heatmaps by a single, fully convolutional network.

2.1 From Images to Center and Curvature Heatmaps

This step eliminates task-irrelevant variations by converting images into center and curvature representations (Fig. 2). RGB images can vary significantly due to lighting, background, and texture changes. These changes, however, do not affect object poses. This step eliminates these effects by transforming images into representations that simplify pose estimation. As we want our system to apply to a wide range of objects, these representations should be category-independent. We identify centers and curvatures as suitable representations with complementary properties.

Centers. We use center heatmaps—modeling the probability of object center points—to detect objects and roughly estimate their locations. Previous work has shown that detecting objects as center points is simple and efficient [33]. Moreover, 2D center points can be triangulated to 3D, initializing object poses and enabling geometric reasoning. For example, center predictions located at impossible locations can be discarded. When predicting center points, there is a trade-off between spatial precision and generalization. Precise center locations can differ subtly between similar objects. This makes it difficult for a learning system to predict precise centers for unseen objects. As we want our system to generalize to unseen categories, we relax spatial precision requirements, by training our model to predict gaussian blobs at center locations. We use a single center heatmap for all object categories.

Curvature. We use curvature heatmaps to highlight local geometry and enable comparison between images and renders. Representations based on 3D geometry are robust to changes in lighting, texture, and background and can be created from textureless CAD files. Representations based on global geometry [29] or category-level semantics [5,19] do not generalize to unseen object types. Therefore, we focus on local geometry. Previous work has shown that geometric edges can be used for accurately estimating 6D poses [3,12,13]. We base our representation on view-space curvature. To obtain these view-space curvatures, we first

render normals in view-space. Next, we approximate the gradients for each pixel using the Prewitt operator [21]. Finally, we calculate the 2-norm of these gradients to obtain a per-pixel curvature value. Areas with high curvatures correspond to geometric edges or object boundaries and are visually distinct. Similarly, visually similar areas, like overlapping parallel planes, contain no curvatures values, as shown in Fig. 3b.
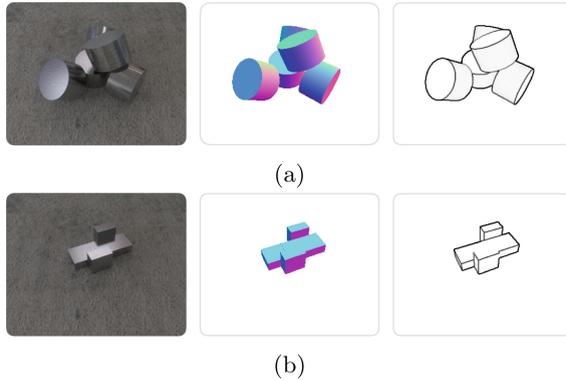

(a)

(b)

Fig. 3. (a) Curvatures are calculated by rendering normals in view space, approximating per-pixel gradients using the Prewitt operator, and calculating the 2-norm. (b) Visually similar areas, like overlapping parallel planes, exhibit no curvature values.

Model and Training. We use a fully convolutional network to predict center and curvature heatmaps. The same weights are applied to images from different viewpoints. The architecture is based on U-net [23] and shown in Fig. 4. A shared backbone outputs feature maps with a spatial resolution equal to the input images. Separate heads are used for predicting center and curvature heatmaps. Ground-truth center heatmaps are created by projecting 3D object centers to each image and splatting the resulting points using a Gaussian kernel, with standard deviation adapted by object size and distance. Curvature heatmaps are created as explained in Sect. 2.1. Binary cross entropy loss is used for both outputs.

2.2   From Center Heatmaps to 3D Centers

This step converts multi-view center heatmaps to 3D center points. First, local maxima are found in every heatmap using a peak local max filter [28] (Fig. 5). Each of these 2D maxima represents a 3D ray, defined by the respective camera intrinsics and extrinsics. Next, for each pair of 3D rays, the shortest mutual distance and midpoint are calculated [24]. When this distance is below a threshold d_t, the midpoint is added to the set of candidates. Within this set, points that are closer to each other than a distance d_c are merged. Finally, the remaining points are refined by maximizing their reprojection score across views, using
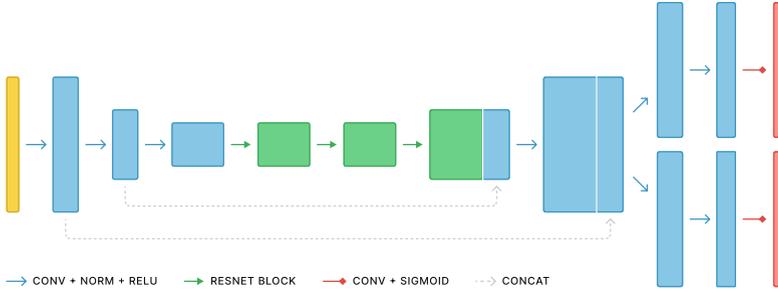
Fig. 4. The architecture of our fully convolutional network is based on U-net. Input images are first processed by a shared backbone. Afterwards, separate heads output center and curvature heatmaps.
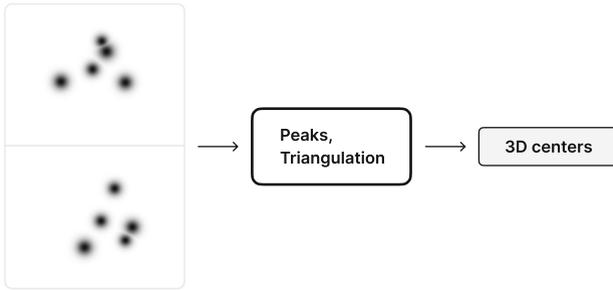

Fig. 5. Step 2: multi-view 2D center heatmaps are converted to 3D center points.

Scipy's Nelder-Mead optimizer [27]. This leads to a set of 3D points, each with per-view heatmap scores. If information about the number of objects in the scene is available, this set is further pruned. 3D centers are sorted by their aggregated heatmap score (accumulated for all views) and removed if they are closer than a distance d_o to higher-scoring points.

2.3 6D Pose Estimation

This step optimizes the 6D poses for all detected objects, using a render-and-compare approach based on curvatures (Fig. 6). Object CAD models, camera intrinsics, and extrinsics are available. Consequently, curvature maps can be rendered for each set of 6D object pose candidates. We define a cost function that compares such curvature renders to the predicted curvature heatmaps. This cost function is used for optimizing a set of object poses, initialized by the previously detected 3D centers.
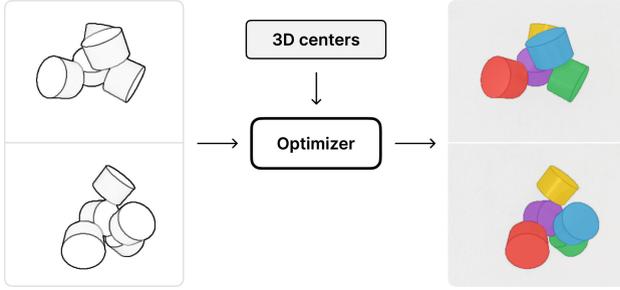
Fig. 6. Step 3: Multi-view curvature heatmaps and 3D object centers are used to find 6D object poses.

Cost Function. Predicted curvature heatmaps are converted to binary images with threshold t_b. Next, for each binary image, a distance map is created where each pixel contains the distance to the closest non-zero (true) pixel, using scikit-image's distance transform [28]. The resulting distance maps have to be calculated only once and can be reused throughout optimization (Fig. 7).
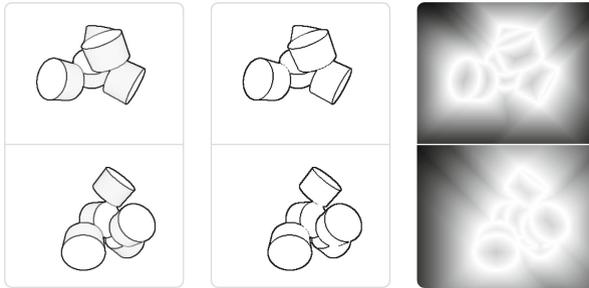

Fig. 7. Target curvature heatmaps (left) are converted to binary images (center). Next, distance maps (right) are calculated, where each pixel contains the distance to the closest non-zero (true) pixel.

Using these distance maps, curvature renders can be efficiently compared to the target heatmaps. Pixel-wise multiplication of a render to a distance map returns an image where each pixel contains the distance to the closest true curvature pixel, weighted by its curvature value. As a result, regions with high curvature weigh more, and regions with zero curvature do not contribute. The final cost value is obtained by summing the resulting image, and dividing by the sum of the rendered curvature map. This is done for each viewpoint. The resulting costs are weighed by view-specific weights w_v and summed, resulting in a final scalar cost value.
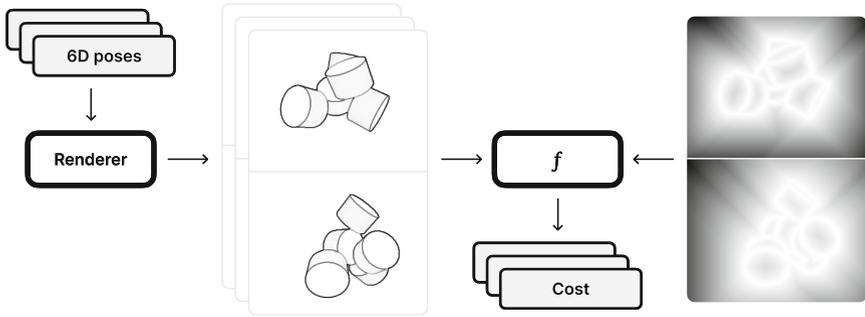
Fig. 8. Overview of the cost function. First, curvature maps are rendered for each set of 6D object pose estimates. These curvature maps are compared to a pre-computed distance map based on the target curvatures. This results in a cost value for each sample, calculated in parallel.

Figure 8 shows an overview of the cost function. We implement this function—including curvature rendering—on GPU. Our implementation runs at 2,000 calls per second on an NVIDIA RTX3090 Ti for six 256×320 images per call.

Optimization. We sequentially optimize 6D object poses by evaluating pose candidates anchored by the detected 3D centers. As scenes can consist of densely stacked objects with many occlusions, we optimize objects sequentially. We argue highly visible objects are easier to optimize and—crucially—should be taken into account when optimizing objects they occlude. For each 3D center, we use per-view center heatmap scores as a proxy for visibility. We optimize objects in order of decreasing visibility, and weigh the contribution of each view to the cost according to these scores. When estimating an object pose, we first evaluate a set of 2,000 pose candidates with random rotations and translations normally distributed around the 3D center. Afterwards, the best candidates are further optimized using a bounded Nelder-Mead optimizer [27].

3   Experiments

We evaluate our method on the 6D localization task as defined in the BOP challenge [10] on DIMO [4] and T-LESS [9], two industry-relevant datasets. On DIMO, we show our method significantly outperforms PVNet [19], a strong single-view baseline. On T-LESS, CenDerNet improves upon the 2020 ECCV results of CosyPose [14], the leading multi-view method.

3.1   DIMO

Dataset. The dataset of industrial metal objects (DIMO) [4] reflects real-world industrial conditions. Scenes consist of symmetric, textureless and highly reflective metal objects, stacked in dense compositions.

Baseline. We compare our method to PVNet with normalized pose rotations. PVNet [19] is based on estimating 2D keypoints followed by perspective n-point optimization. This method is robust to occlusion and truncation, but suffers from pose ambiguities caused by object symmetries. This is amended by mapping rotations to canonical rotations before training [20].

Experiments. We use a subset of the dataset, focusing on the most reflective parts, and scenes with objects of the same category. For training, only synthetic images are used. We report results on both synthetic and real-world test data. When predicting center and curvature heatmaps, the same output channel is used for all object categories.

Evaluation. We report average recall (AR) for two different error functions:

– MSPD, the maximum symmetry-aware projection distance, as it is relevant for evaluating RGB-only methods.
– MSSD, the maximum symmetry-aware surface distance, as it is relevant for robotic manipulation.

Strict thresholds of correctness θ_e are chosen, as manufacturing use-cases require high-precision poses. θ_{MSSD} is reported for 5% of the object diameter, θ_{MSPD} for 5px.

Our system predicts poses in world frame, whereas PVNet—a single-view method—predicts per-image poses that are relative to the camera. For better comparison, we transform our estimated poses to camera frames and compute per-image metrics.

Table 1. Results on DIMO for synthetic test images. CenDerNet significantly outperforms the PVNet baseline. The difference between MSSD and MSPD shows the importance of multi-view images for robotics applications where spatial precision is important.

	$\mathrm{AR_{MSPD}} < 5\mathrm{px}$	$\mathrm{AR_{MSSD}} < 5\%$
PVNet	0.577	0.079
CenDerNet (ours)	**0.721**	**0.639**

Results. As shown in Table 1 and Table 2, CenDerNet significantly outperforms the PVNet baseline. Despite rotation normalizations, PVNet still suffers from pose ambiguities, struggling to predict high-precision poses.

Table 2. Results on DIMO for real-world test images. Both methods are affected by the sim-to-real gap. While the results of PVNet plummet, CenDerNet is more robust.

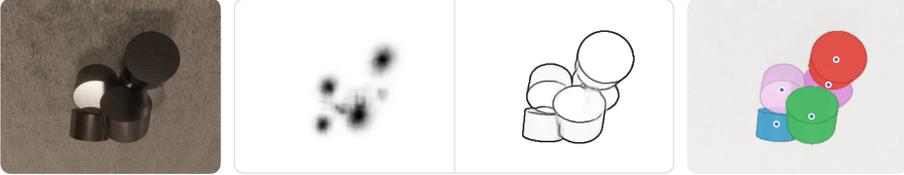|	$AR_{MSPD} < 5px$	$AR_{MSSD} < 5\%$
PVNet	0.016	0.000
CenDerNet (ours)	**0.516**	**0.403**

Fig. 9. Results on synthetic test image of the DIMO dataset. Despite severe reflections and shadows, our system recovers qualitative center and curvature representations and precise 6D poses.
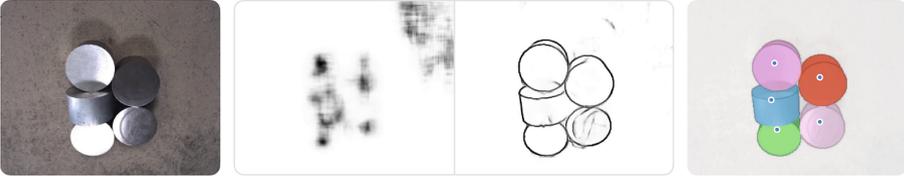

Fig. 10. Results on real-world test image of the DIMO dataset. The lower quality of the center and curvature heatmaps is due to the sim-to-real gap. Nevertheless, the estimated 6D poses are accurate.
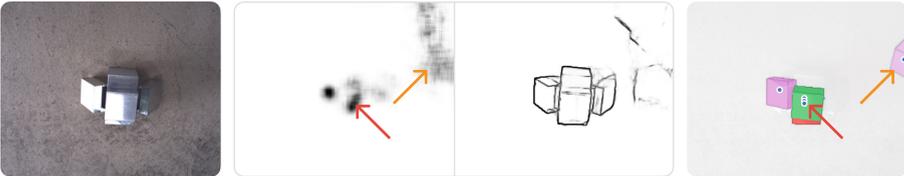

Fig. 11. Failure case on a real-world test image of the DIMO dataset. The unexpected output in the center heatmap (orange arrow) could be due to the sim-to-real gap. Objects cannot be differentiated when object centers align (red arrow). (Color figure online)

The large gap in MSPD en MSSD scores for PVNet shows the importance of multi-view images for robotics applications. While PVNet's MSPD score (in image space) is reasonable, its MSSD score (in world/robot space) is subpar.

Reflections and shadows significantly affect many test images, leading to challenging conditions, as shown in Fig. 9 and Fig. 10. Nevertheless, center and curvature predictions are often sufficient to find accurate object poses. The difference

in results between Table 1 and Table 2 shows the effect of the sim-to-real gap. Here, many of the keypoints predicted by PVNet are too far off to meet the strict accuracy requirements, causing results to plummet. CenDerNet is more robust.

An important advantage of our method is the interpretability of the predicted center and curvature representations. The top-right area of the center heatmap in Fig. 10 and Fig. 11 could be due to the sim-to-real gap. Adding more background variations while training could improve the robustness of the model.

Figure 11 shows a failure case. When object centers align, the system fails to differentiate between them.

3.2   T-LESS

Dataset. The T-LESS dataset provides multi-view images of non-reflective, colorless industrial objects. As part of the BOP benchmark, it allows easy comparison to state-of-the-art methods. For the 2020 BOP challenge [11], photorealistic synthetic training images (PBR) were provided.

Baseline. We compare our method to CosyPose [14], the multi-view method that won the 2020 BOP challenge. We use the ECCV 2020 BOP evaluation results for 8 views made available by the authors.

Experiments. We use only PBR images for training. As each scene contains a mix of object types, we extend our model to predict a separate center heatmap for each object category. This reduces the generality of our method, but allows us to select the correct CAD model when optimizing poses. As training images contain distractor objects, we take object visibility into account when generating ground-truth center and curvature heatmaps.

Evaluation. In addition to reporting $\text{AR}_{\text{MSPD}} < 5\text{px}$ and $\text{AR}_{\text{MSSD}} < 5\%$, we report the default BOP [10] evaluation metrics (VSD, MSSD and MSPD). We transform poses to camera-frames and calculate metrics for all images.

Results. As shown in Table 3, our method outperforms CosyPose in high-precision scenarios. Table 4 shows we also outperform the provided CosyPose predictions on the default BOP metrics. However, the CosyPose-variant that is currently at the top of the BOP leaderboard still performs significantly better.

Table 3. Results on TLESS. CenDerNet outperforms CosyPose on both metrics. The stark difference in MSSD score is relevant for high-precision robotics applications.

	$\text{AR}_{\text{MSPD}} < 5\text{px}$	$\text{AR}_{\text{MSSD}} < 5\%$
CosyPose (ECCV 2020)	0.499	0.250
CenDerNet (ours)	**0.543**	**0.544**

Table 4. Results on TLESS for the default BOP evaluation metrics. CenDerNet outperforms the provided CosyPose results. However, the scores reported on the BOP leaderboard are still significantly higher.

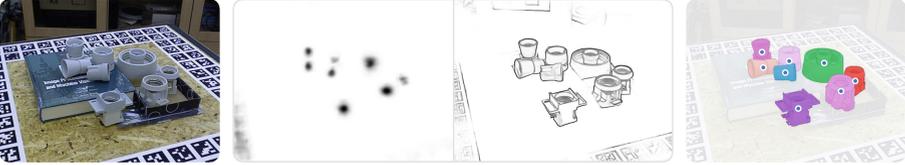|	AR	AR_{MSPD}	AR_{MSSD}	AR_{VSD}
CosyPose (ECCV 2020)	0.617	0.686	0.610	0.557
CenDerNet (ours)	0.713	0.715	0.717	0.707
CosyPose (BOP leaderboard)	**0.839**	**0.907**	**0.836**	**0.773**

Fig. 12. Results on T-LESS.Our model is able to eliminate irrelevant scene elements, like backgrounds and distractor objects.
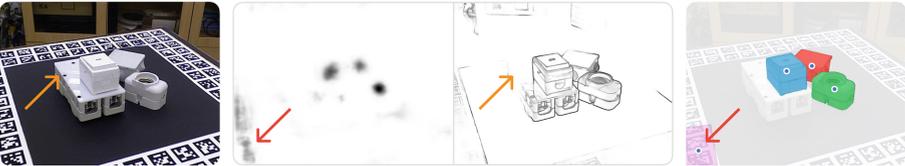

Fig. 13. Failure case on T-LESS. The model is confused about the charuco tags around the scene, leading to an incorrect 3D center. The pose optimization process cannot recover from this error.

Figure 12 shows predicted heatmaps, object centers and poses. In many situations, our method is capable of eliminating irrelevant backgrounds and objects, despite being trained only on synthetic data. There is, however, still a sim-to-real gap, as shown in Fig. 13. Again, the predicted representations provide hints on how to improve the system. In this case, the model is confused by charuco tags on the edge of the table.

4   Conclusions

We present CenDerNet, a system for multi-view 6D pose estimation based on center and curvature representations. Our system is conceptually simple and therefore easy-to-use. First, a single neural network converts images into interpretable representations. Next, a render-and-compare approach, with GPU-optimized cost function, allows for jointly optimizing object poses across multiple viewpoints, thereby naturally handling occlusions and object symmetries. We demonstrate our system on two challenging, industry-relevant datasets and

show it outperforms PVNet, a strong single-view baseline, and CosyPose, the leading multi-view approach. In future work, we plan to explore ways to further decrease the processing time of our method. In addition, we will investigate improvements to the robustness of our pipeline. For example, fusing information from multiple viewpoints could improve center and curvature predictions. We also look forward to evaluating our method in a real-world setup.

References

1. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008). https://doi.org/10.1016/j.cviu.2007.09.014
2. Commission, E.: Directorate-General for Internal Market, Industry, E., SMEs: a vision for the European industry until 2030 : final report of the Industry 2030 high level industrial roundtable. Publications Office (2019). https://doi.org/10.2873/34695
3. Daems, R., Victor, J., Baets, P.D., Onsem, S.V., Verstraete, M.: Validation of three-dimensional total knee replacement kinematics measurement using single-plane fluoroscopy. Int. J. Sustain. Constr. Des. **7**(1), 14 (2016). https://doi.org/10.21825/scad.v7i1.3634
4. De Roovere, P., Moonen, S., Michiels, N., Wyffels, F.: Dataset of industrial metal objects. arXiv preprint arXiv:2208.04052 (2022)
5. Florence, P.R., Manuelli, L., Tedrake, R.: Dense object nets: learning dense visual object descriptors by and for robotic manipulation. arXiv preprint arXiv:1806.08756 (2018)
6. Fu, C.Y., Shvets, M., Berg, A.C.: RetinaMask: learning to predict masks improves state-of-the-art single-shot detection for free. arXiv preprint arXiv:1901.03353 (2019)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
8. Hinterstoisser, S., et al.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 International Conference on Computer Vision. IEEE, November 2011. https://doi.org/10.1109/iccv.2011.6126326
9. Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: an RGB-d dataset for 6d pose estimation of texture-less objects. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, March 2017. https://doi.org/10.1109/wacv.2017.103

10. Hodaň, T., et al.: BOP challenge 2020 on 6D object localization. In: Bartoli, A., Fusiello, A. (eds.) ECCV 2020. LNCS, vol. 12536, pp. 577–594. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66096-3_39

11. Hodaň, T., et al.: Photorealistic image synthesis for object instance detection. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 66–70 (2019). https://doi.org/10.1109/ICIP.2019.8803821

12. Jensen, A., Flood, P., Palm-Vlasak, L., Burton, W., Rullkoetter, P., Banks, S.: Joint track machine learning: an autonomous method for measuring 6dof tka kinematics from single-plane x-ray images. arXiv preprint arXiv:2205.00057 (2022)

13. Kaskman, R., Shugurov, I., Zakharov, S., Ilic, S.: 6 DoF pose estimation of textureless objects from multiple RGB frames. In: Bartoli, A., Fusiello, A. (eds.) ECCV 2020. LNCS, vol. 12536, pp. 612–630. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66096-3_41

14. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: CosyPose: consistent multi-view multi-object 6D pose estimation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12362, pp. 574–591. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58520-4_34

15. Li, C., Bai, J., Hager, G.D.: A unified framework for multi-view multi-class object pose estimation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11220, pp. 263–281. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01270-0_16

16. Li, Y., Wang, G., Ji, X., Xiang, Yu., Fox, D.: DeepIM: deep iterative matching for 6D pose estimation. Int. J. Comput. Vis. **128**(3), 657–678 (2019). https://doi.org/10.1007/s11263-019-01250-9

17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, October 2017. https://doi.org/10.1109/iccv.2017.324

18. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, October 2019. https://doi.org/10.1109/iccv.2019.00776

19. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: pixel-wise voting network for 6dof pose estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2019. https://doi.org/10.1109/cvpr.2019.00469

20. Pitteri, G., Ramamonjisoa, M., Ilic, S., Lepetit, V.: On object symmetries and 6D pose estimation from images. In: 2019 International Conference on 3D Vision (3DV). IEEE, September 2019. https://doi.org/10.1109/3dv.2019.00073

21. Prewitt, J.M., et al.: Object enhancement and extraction. Pict. Process. Psychopictorics **10**(1), 15–19 (1970)

22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017). https://doi.org/10.1109/tpami.2016.2577031

23. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

24. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer Science & Business Media, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-1-84882-935-0

25. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, June 2018. https://doi.org/10.1109/cvpr.2018.00038
26. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. In: 2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29–31 October 2018, Proceedings. Proceedings of Machine Learning Research, vol. 87, pp. 306–316. PMLR (2018)
27. Virtanen, P., et al.: SciPy 1.0 contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2
28. van der Walt, S., et al.: The scikit-image contributors: scikit-image: image processing in python. PeerJ **2**, e453 (2014). https://doi.org/10.7717/peerj.453
29. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6D object pose and size estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2019. https://doi.org/10.1109/cvpr.2019.00275
30. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: a convolutional neural network for 6d object pose estimation in cluttered scenes. In: Robotics: Science and Systems XIV. Robotics: Science and Systems Foundation, June 2018. https://doi.org/10.15607/rss.2018.xiv.019
31. Yang, J., Gao, Y., Li, D., Waslander, S.L.: ROBI: a multi-view dataset for reflective objects in robotic bin-picking. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, September 2021. https://doi.org/10.1109/iros51168.2021.9635871
32. Zakharov, S., Shugurov, I., Ilic, S.: DPOD: 6d pose object detector and refiner. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, October 2019. https://doi.org/10.1109/iccv.2019.00203
33. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. CoRR abs/1904.07850 (2019)
34. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2019. https://doi.org/10.1109/cvpr.2019.00589