Supplementary Material – PVN3D: A Deep Point-wise 3D Keypoints Voting Network for 6DoF Pose Estimation

Yisheng He[1] Wei Sun[2] Haibin Huang[3] Jianran Liu[2] Haoqiang Fan[2] Jian Sun[2]
[1]Hong Kong University of Science and Technology
[2]Megvii Inc. [3]Kuaishou Technology

1. Architecture Details.

The image embedding network in the Feature Extraction module consists of a standard ResNet-34 encoder pretrained with ImageNet and followed by 4 up-sampling layers as a decoder. The output RGB feature is of 128 channels. The point cloud embedding network is a PointNet++ [9] with Multi-scale Grouping (MSG), the output of which is also of 128 channels. In the DenseFuion block, for each sampled point, the 128 channels RGB feature is concatenated with the corresponding 128 channels point cloud feature and is fed into shared MLPs to raise the feature channels to 1024. The 128 channels RGB and point cloud feature are also fed into shared MLPs and raised to 256 channels respectively. They are then all concatenated together to form 1792 (128*2+256*2+1024) channels RGBD features. The 3D keypoint offset voting module $\mathcal{M}_{\mathcal{K}}$ consists of shared MLPs (1792-1024-512-128-N_{kp}*3) reducing the 1792 channels RGBD features to N_{kp} 3D keypoints offset. The semantic segmentation module $\mathcal{M}_{\mathcal{S}}$ consists of shared MLPs (1792-1024-512-128-N_{cls}) with N_{cls} the number of object classes. The center voting module $\mathcal{M}_{\mathcal{C}}$ also consists of shared MLPs (1792-1024-512-128-3). During training, Adam optimization algorithm is employed, with a mini-batch size of 24. We applied cyclical learning rates during training, the range of which is from 1e-5 to 1e-3.

2. Implementation: Models for LineMOD dataset.

In the LineMOD dataset, though there are multi objects in one scene, only the label of one target object in each scene is provided, making it hard to train a single model for all objects. Therefore, we follow PVNet[8] and trained models separately for each object. It means our semantic segmentation module $\mathcal{M}_{\mathcal{S}}$ only predicts two semantic labels for each object, label of the target object and the background.

3. Implementation: PVN3D+ICP

We introduce our implementation of PVN3D+ICP as follows. During training and inference of PVN3D, we randomly sample 12288 points (pixels) from the whole scene as input. Only these points are labeled with the semantic label by our instance semantic segmentation module, which is not enough for a good performance of the ICP algorithm. Therefore, for each unlabeled point in the whole point cloud scene, we find its closest labeled point and assigned that label to it. Points with the same instance semantic labels are then selected. To eliminate the effect of noise points, a MeanShift [2] clustering algorithm is also applied. The biggest cluster of points is then selected as the input of the ICP algorithm.

4. More Results

4.1. Instance semantic segmentation

We provide more results of instance semantic segmentation in Table 1.

4.2. Visualization of detected keypoints

The visualization of some detected keypoints is shown in Figure 1.

4.3. Evaluation on the Occlusion LineMOD Dataset

The Occlusion LineMOD dataset [1] is additionally annotated from the LineMOD datasets, objects of which are heavily occluded, making it harder for pose estimation. Evaluation results in Table 2 show that our PVN3D outperforms previous methods by a large margin.

4.4. Inference time

It takes 0.17 seconds for network forward propagation and 0.02 seconds for pose estimation of each object during inference. The overall runtime is 5 FPS on the LineMOD dataset.

	PoseCNN[10]	Mask R-CNN[3]	$\mathcal{M}_{\mathcal{S}}$	$\mathcal{M}_{\mathcal{S},\mathcal{K}}$	$\mathcal{M}_{\mathcal{S},\mathcal{K},\mathcal{C}}$
002_master_chef_can	86.9	87.7	88.0	**88.1**	87.7
003_cracker_box	87.5	85.9	88.0	88.2	**88.3**
004_sugar_box	92.0	91.1	**92.3**	91.4	91.5
005_tomato_soup_can	86.4	87.1	88.5	**88.7**	**88.7**
006_mustard_bottle	93.1	93.1	92.7	93.2	**93.3**
007_tuna_fish_can	89.2	88.8	89.2	89.5	**89.6**
008_pudding_box	73.9	82.5	86.7	90.7	**90.8**
009_gelatin_box	84.3	91.9	**92.6**	89.5	89.5
010_potted_meat_can	83.9	84.9	**88.4**	86.4	86.3
011_banana	90.0	**90.2**	89.0	89.0	89.1
019_pitcher_base	95.7	92.2	96.1	96.3	**96.4**
021_bleach_cleanser	88.2	90.2	87.1	**91.8**	91.7
024_bowl	90.3	90.8	**92.1**	79.0	78.8
025_mug	83.0	80.7	83.8	**84.1**	**84.1**
035_power_drill	86.4	87.8	87.2	**88.8**	**88.8**
036_wood_block	79.8	**83.2**	82.6	78.5	78.5
037_scissors	65.7	51.6	67.1	77.1	**77.2**
040_large_marker	69.7	73.4	**76.9**	74.5	74.6
051_large_clamp	43.1	48.4	58.5	62.5	**70.2**
052_extra_large_clamp	30.4	36.1	41.5	50.7	**69.0**
061_foam_brick	87.7	87.4	**89.3**	87.4	87.3
MEAN	80.3	81.2	83.7	84.1	**85.3**

Table 1. Instance semantic segmentation results (mIoU(%)) of different approaches.

	PoseCNN [10]	Oberweger [6]	Hu et al. [5]	Pix2Pose [7]	PVNet [8]	DPOD [11]	PVN3D
ape	9.6	12.1	17.6	22.0	15.8	-	**33.9**
can	45.2	39.9	53.9	44.7	63.3	-	**88.6**
cat	0.9	8.2	3.3	22.7	16.7	-	**39.1**
driller	41.4	45.2	62.4	44.7	65.7	-	**78.4**
duck	19.6	17.2	19.2	15.0	25.2	-	**41.9**
eggbox	22.0	22.1	25.9	25.2	50.2	-	**80.9**
glue	38.5	35.8	39.6	32.4	49.6	-	**68.1**
holepuncher	22.1	36.0	21.3	49.5	39.7	-	**74.7**
average	24.9	27.0	27.0	32.0	40.8	47.3	**63.2**

Table 2. Quantitative evaluation of 6D Pose on ADD(S) [4] metric on the Occlusion LineMOD dataset. Objects with bold name are symmetric.

References

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. 1

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):603–619, 2002. 1

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[4] 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*, pages 548–562. Springer, 2012. 2

[5] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3385–3394, 2019. 2

[6] M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2

[7] K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise co-

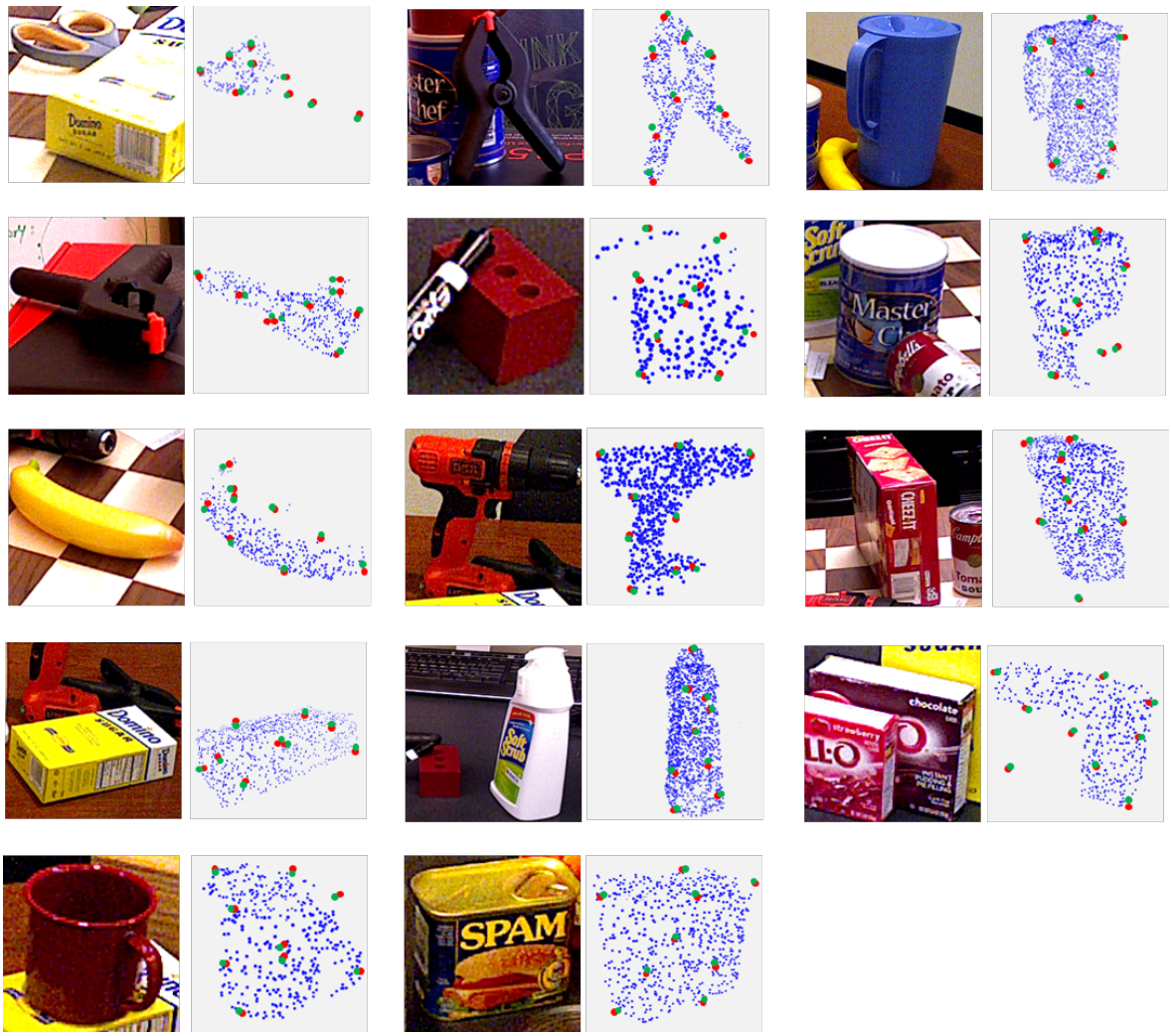

Figure 1. **Visualization of 3D keypoints in YCB-Video dataset.** Blue points are sampled point clouds from the scene. Red points are the ground truth 3D keypoints and green points are predicted 3D keypoints.

ordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7668–7677, 2019. 2

[8] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 1, 2

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1

[10] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 2

[11] S. I. Zakharov, S. and S. Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1941–1950, 2019. 2