Supplementary Material for OVE6D: Object Viewpoint Encoding for Depth-based 6D Object Pose Estimation

Dingding Cai[1], Janne Heikkilä[2], Esa Rahtu[1]
[1]Tampere University, [2]University of Oulu
{dingding.cai, esa.rahtu}@tuni.fi, janne.heikkila@oulu.fi

1. Qualitative examples

We illustrate qualitative pose estimation examples from the LINEMOD dataset for OVE6D and LatentFusion [6] in Figure 1. Note that the ground truth segmentation mask is used for LatentFusion by following [6], while OVE6D is evaluated using the predicted segmentation mask provided by Mask-RCNN [1].

2. Parameter configurations

The granularity of the discretized out-of-plane rotations (viewpoints) is determined by the parameter N. We conduct experiments to explore how the ADD(-S) recall is affected by the number of viewpoints N, the retrieving number of viewpoint hypothesis K, and the number of orientation proposal P. By increasing the number (N) of viewpoints, we reduce the average distance of adjacent viewpoints, which can result in a higher ADD(-S) recall, as shown in Table 1. In addition, retrieving more viewpoint hypotheses (K) from the codebook and taking more orientation proposals (P) could increase the probability of obtaining the correct pose for the subsequent stages. The experimental results are presented in Table 2 and Table 3. On the other hand, a finer discretization of the out-of-plane rotation leads to a larger memory footprint (more viewpoints), and more orientation proposals consume a longer verification time. We found $N = 4000, K = 50$, and $P = 5$ to be a good trade-off between the accuracy and the efficiency.

Sampling Number (N) (K = 1, P = 1)	1k	2k	4k	8k	16k
AAVD(°)	6.1	4.3	3.1	2.1	1.5
ADD(-S)(%)	74.1	75.0	75.7	75.8	76.6

Table 1. The average ADD(-S) recalls on the LINEMOD dataset in terms of the varying number of viewpoint sampling. "AAVD" in short for the Average Adjacent Viewpoint Distance.

Retrieving Number (K) (N = 4k, P = 1)	1	10	30	50	100
ADD(-S)(%)	75.7	80.7	81.5	81.6	81.5

Table 2. The average ADD(-S) recalls on the LINEMOD dataset in terms of the varying number of viewpoint retrieval.

Proposal Number (P) (N = 4k, K = 50)	1	3	5	10	20
ADD(-S)(%)	81.6	85.0	86.1	87.0	87.2

Table 3. The average ADD(-S) recalls on the LINEMOD dataset in terms of the varying number of orientation proposal.

3. Viewpoint codebook construction

In the main paper, we use the object 3D mesh model to construct the object viewpoint codebook (using synthesized data) to avoid the expensive 6D pose annotation. Nevertheless, the object viewpoint codebook can also be built using real-world training data (with the ground truth 6D object poses). To this end, we conduct additional experiments on the LINEMOD dataset where we build the viewpoint codebook using the real annotated images instead of the mesh model. The experimental results are presented in Table 4 in terms of the average ADD(-S) recall. We can observe a slight gain in the results compared to those obtained with the synthetic data. We attribute this to the alleviation of the domain gap between the object viewpoint codebook and the observed depth images.

4. Orientation decomposition

Our method decouples the complete 3D orientation into two components, *i.e.*, the out-of-plane rotation (viewpoint) and the in-plane rotation around the camera optical axis. Here, we provide more details about the factorization.

The object 3D orientation matrix $\mathbf{R}$ can be factorized into three separate rotations around each axis (x, y and z-axis) with respect to the object coordinate system. *i.e.* $\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$, where $\mathbf{R}_z$, $\mathbf{R}_y$ and $\mathbf{R}_x$ are the rotations around the z, y and x axis, respectively. Furthermore, we re-

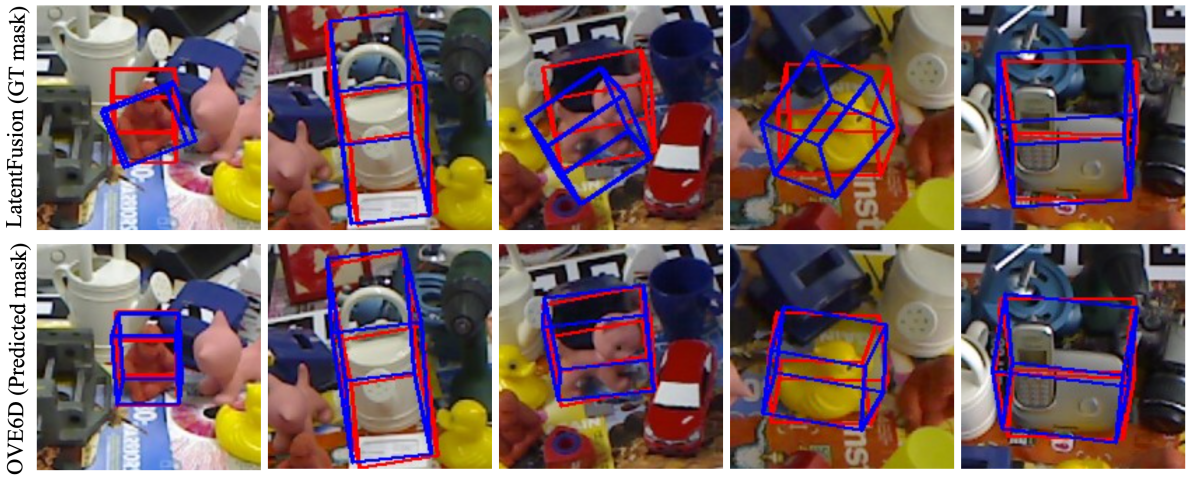

Figure 1. **Qualitative evaluation on LineMOD**. We show the qualitative results of LatentFusion [6] (first row) and OVE6D (second row). Red and blue 3D bounding boxes indicate the ground truth and the estimated poses, respectively.

Reference Data	Method	Input	ICP	ADD (-S)(%)
Mutli-View With Pose Annotation	LatentFusion(GT) [6]	RGBD		*87.1*
	OVE6D(GT)	D		*97.0*
	OVE6D(GT)	D	✓	**99.4**
	OVE6D(MRCNN)	D		86.5
	OVE6D(MRCNN)	D	✓	94.0
Object Mesh Model	OVE6D(GT)	D		*96.4*
	OVE6D(GT)	D	✓	***98.7***
	OVE6D(MRCNN)	D		86.1
	OVE6D(MRCNN)	D	✓	92.4

Table 4. Evaluation on **LINEMOD**. We report the average ADD(-S) recall. ICP refinement is performed for all pose proposals before pose selection. MRCNN and GT indicate using the masks provided by Mask-RCNN and the ground truth, respectively.

formulate the 3D orientation matrix as $\mathbf{R} = \mathbf{R}_\theta \mathbf{R}_\gamma$, where $\mathbf{R}_\theta = \mathbf{R}_z$ is the in-plane rotation around the camera optical axis (z axis) and $\mathbf{R}_\gamma = \mathbf{R}_y \mathbf{R}_x$ is the out-of-plane rotation. In the case of isometric orthographic projection, the histograms of object depth values is mainly determined by the out-of-plane rotation of the object, as illustrated in Figure 2. To this end, we uniformly discretize the out-of-plane rotation $\mathbf{R}_\gamma \in R^{3\times3}$ as a finite set of object viewpoints $\{\mathbf{R}_i^\gamma\}_{i=1}^N$ ($N = 4000$ in the main paper) and encode the object viewpoints into latent vectors. These latent viewpoint embeddings are invariant to the in-plane rotation around the camera optical axis. Moreover, the in-plane rotation $\mathbf{R}_\theta \in R^{3\times3}$ is formulated as,

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where θ is the rotating angle around the camera optical axis.

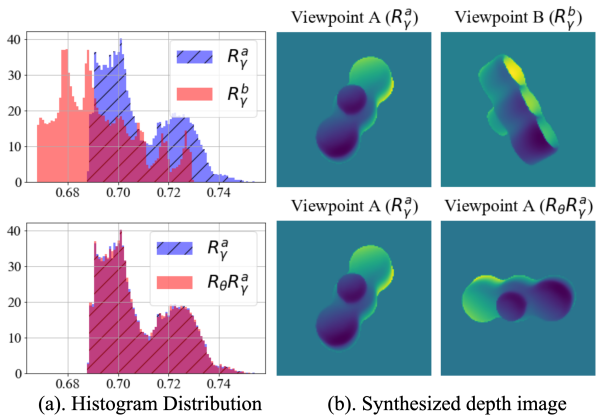

Figure 2. In the first row, we show the histograms of object depth values observed from two different viewpoints A ($\mathbf{R}_\gamma^a$) and B ($\mathbf{R}_\gamma^b$). In the second row, we show the histograms of object depth values from the same viewpoint A ($\mathbf{R}_\gamma^a$ and $\mathbf{R}_\theta \mathbf{R}_\gamma^a$). $\mathbf{R}_\theta$ is an in-plane rotation around the camera optical axis. We can observe that the (asymmetric) object depth images rendered from different viewpoints result in different distributions. In contrast, the depth images from the same viewpoint but with different in-plane rotations share similar distributions.

Equivalently, we construct the in-plane rotation matrix $\mathbf{R}_\theta$ as,

$$\mathbf{R}_\theta = \begin{bmatrix} \vartheta_1 & -\vartheta_2 & 0 \\ \vartheta_2 & \vartheta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where ϑ_1, ϑ_2 are scalar values of a unit vector $\Theta \in R^2$ predicted by the in-plane orientation regression network of OVE6D. As presented in Figure 3, we show some intermediate results $\mathbf{P}_{temp}$ without the regressed in-plane rotation and final complete 6D pose results $\mathbf{P}_{final}$ (with the

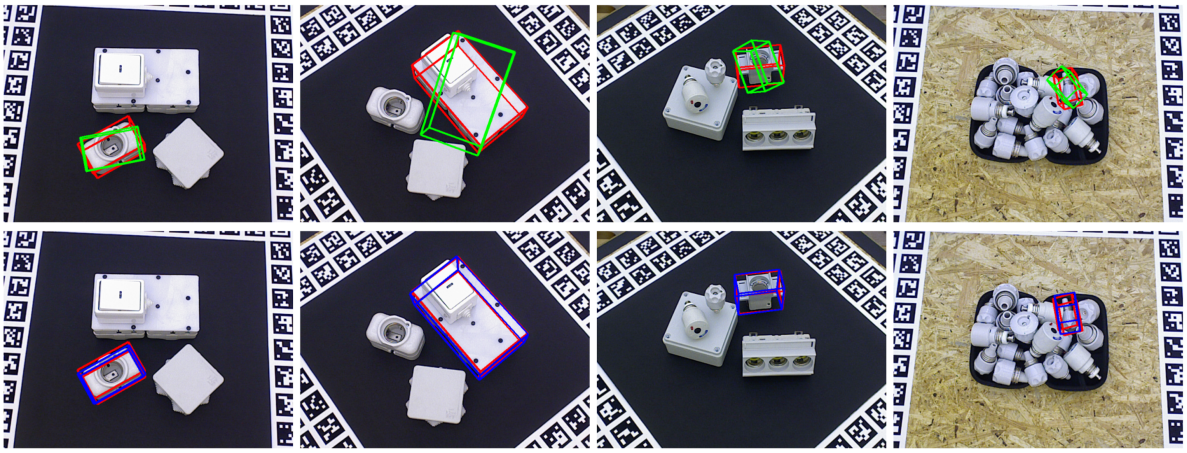

Figure 3. **Qualitative evaluation on T-LESS**. In the first row, we show the intermediate results $\mathbf{P}_{temp}$ (without the in-plane orientation regression). In the second row, we show the final complete 6D poses $\mathbf{P}_{final}$ (with the in-plane orientation regression). Red, green and blue 3D bounding boxes represent the ground truth, the intermediate and the final 6D poses, respectively.

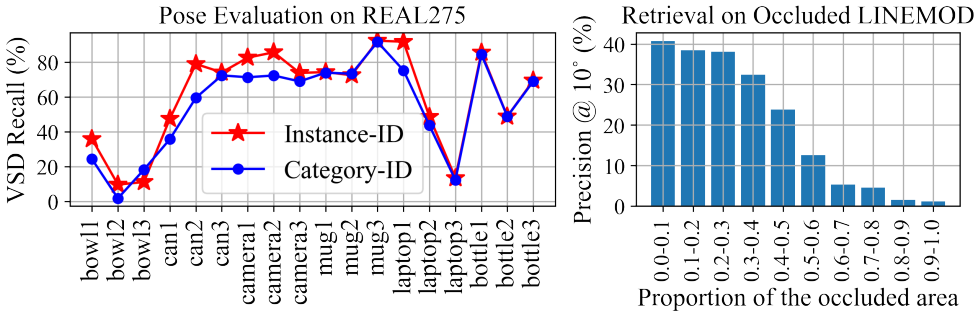

Figure 4. L: Results for REAL275 (18 objects from 6 categories) using category or instance-level IDs. In category case, we retrieve from all object codebooks belonging to the selected category. R: Viewpoint retrieval results with respect to the occlusion size.

estimated in-plane rotation), *i.e.*, $\mathbf{P}_{temp} = [\mathbf{R}_i^{\gamma}|\mathbf{t}]$ and $\mathbf{P}_{final} = [\mathbf{R}_i^{\theta}\mathbf{R}_i^{\gamma}|\mathbf{t}]$, where $\mathbf{R}_i^{\gamma} \in R^{3\times3}$ is the rotation matrix of the retrieved object viewpoint, $\mathbf{t} \in R^3$ is the estimated object 3D translation, and $\mathbf{R}_i^{\theta}$ is the estimated in-plane rotation for the retrieved viewpoint.

5. Data augmentation

We apply the commonly used training data augmentation techniques to improve the generalization of our model. In particular, we first downscale the synthetic depth image with a random factor and then augment the downscaled depth image (see Tab. 5) before re-scaling it to the original size. The imgaug [3] library is employed to achieve this.

6. Object category-level / instance-level ID

We follow the standard practice and use the class labels predicted by the off-the-shelf Mask-RCNN detector as the object IDs. The IDs are used to index the viewpoint codebook and, therefore, a wrong or non-optimal ID could damage the performance as an inadequate codebook would be used in the retrieval. To gain further insight, we evaluated OVE6D using the category-level 6D dataset REAL275 [8]. The results (Fig. 4 left) show that OVE6D achieves comparable performance using object category IDs instead of the object instance IDs. We believe this is due to the fact that OVE6D is a shape-based method and objects within a category often share similar shapes.

7. Sensitivity to occlusion

We performed an additional experiment to examine the viewpoint retrieval performance on the Occluded LINEMOD dataset using ground truth segmentation masks in terms of varying percentage of object visibility. The results in Figure 4 right indicate that the performance remains almost intact up to 30 % occlusion and declines smoothly after that.

Technique	Parameter	Description
Rescale	$0.2 \sim 0.8$	Downscale the original image with a random ratio and then upscale to the original size.
LaplaceNoise	$0.0 \sim 0.01$	Add the Laplace noise to the downscaled image with a random deviation.
Cutout	$0.01 \sim 0.1$	Cutout rectangular area from the downscaled image with a random area ratio.
GaussianBlur	$0.0 \sim 1.5$	Apply random Guassian blurring on the downscaled image.
RandomOcclusion	0.2	Apply a random square or circle occlusion mask on the downscaled image.

Table 5. Data augmentation techniques and parameters applied on the training data.

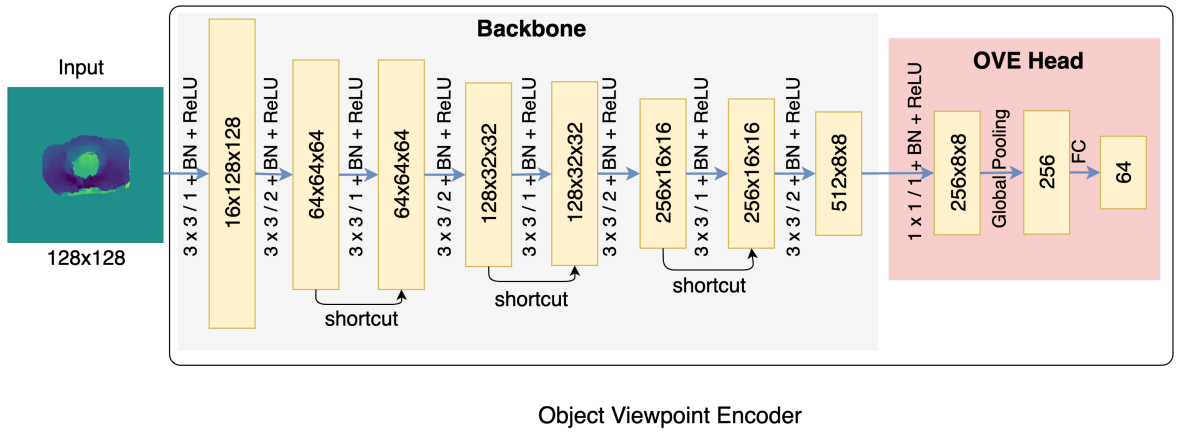

Figure 5. Network structure of the proposed object viewpoint encoder.

8. Structure of the object viewpoint encoder

Figure 5 illustrates the network architecture of the proposed object viewpoint encoder invariant to the in-plane rotation around the camera optical axis. Every convolution layer (3×3 / s where s denoting stride) is followed by the batch normalization (BN) and ReLU activation layers. Besides, skip connections are added between the feature maps with the same dimensionality.

9. Training details of Mask-RCNN

We employ Mask-RCNN [1] from Detectron2 [9] with the backbone ResNet50-FPN [4] to predict the segmenation masks for the objects in the LINEMOD and LINEMOD-Occlusion datasets. We use the physically-based rendered (PBR) images provided by BOP Challenge 2020 [2] to train the network.

Specifically, we apply two steps to finetune the Mask-RCNN to overcome the domain gap between the real and synthetic images. In the first step, we freeze the backbone of Mask-RCNN initialized with the pretrained weights (on MSCOCO dataset [5]) and train 50k iterations on the training data using the default *WarmupMultiStepLR* learning schedule with the learning rate $lr = 0.001$, decayed by 10 at the iteration steps 30k and 40k, respectively. In the second step, we unfreeze the backbone and separately train additional 50k iterations for the 13 objects of LINEMOD dataset as well as 50k iterations for the 8 objects of LINEMOD-Occlusion dataset using the *CosineAnnealingLR* learning schedule with the learning rate $lr = 0.001$. While for the TLESS dataset we directly employ the segmentation results provided by Multi-Path Encoder [7] for a fair comparison.

References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 4

[2] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 4

[3] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 3

[4] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[6] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[7] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13916–13925, 2020. 4

[8] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 3

[9] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4