Supplementary material
Neural Object Learning for 6D Pose Estimation Using a Few Cluttered Images

1. SMOT Dataset

Fig. 1 shows 2 training sequences and the reconstruction of each scene. Fig. 3 lists 11 test scenes of SMOT. All sequences are captured by an Asus Xtion Pro, 640×480 resolution, mounted on the head of a mobile robot. Eight target objects and their names are presented in Fig. 2. Statistics of the dataset are summarized in Table 1. For a test sequence, the pose of each object in a reference frame is manually annotated and the poses in other frames are computed using the relative camera poses that are jointly determined using the 2D markers [1] and the 3D reconstruction method. Additional manual adjustments are performed when poses are remarkably wrong. The dataset is available online: `https://www.acin.tuwien.ac.at/en/vision-for-robotics/software-tools/smot`.

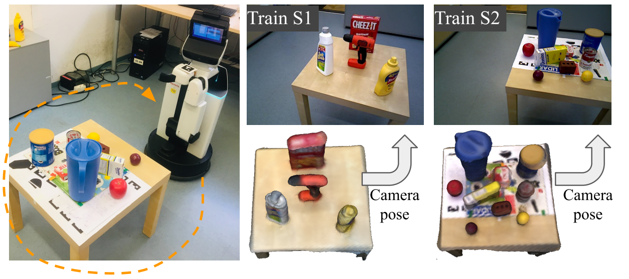


Figure 1. Training images of SMOT is collected using a mobile robot driving around the table

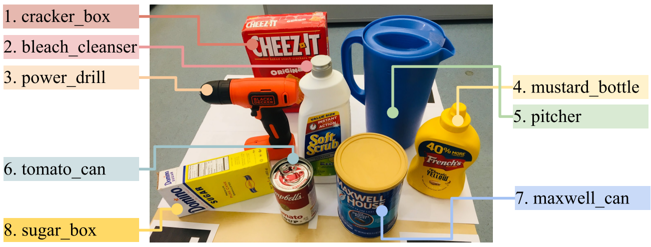


Figure 2. Target objects of SMOT

Table 1. Statistics of SMOT. Training images have a limited elevation range in comparison to the range of test images

Object	cracker_box	bleach	driller	mustard	pitcher	tomato_can	maxwell_can	sugar_box
No. Test Images	2155	2171	2118	2118	2090	2053	2106	2037
Train-Azimuth	(-180°, 180°)				(-180°, 180°)			
Train-Elevation	(38.3°, 40.0°)				(38.9°, 40.8°)			
Test-Azimuth	(-180°, 180°)							
Test-Elevation	(8°, 42°)							

2. Implementation Details

2.1. Examples of training batches

Examples of source images and target images are depicted in Fig. 4. Color augmentations are applied to source images and pose perturbations are applied to pose annotations of source images with the parameters in Table 2. No augmentation is applied to target images and target poses.

2.2. Training

The Adam optimizer with a learning rate of 0.001 is used and the learning rate is divided by a factor of 10 after 20 epochs during the training of 35 epochs. Both training and evaluation are performed using an Nvidia GTX 1080 with 8Gb memory

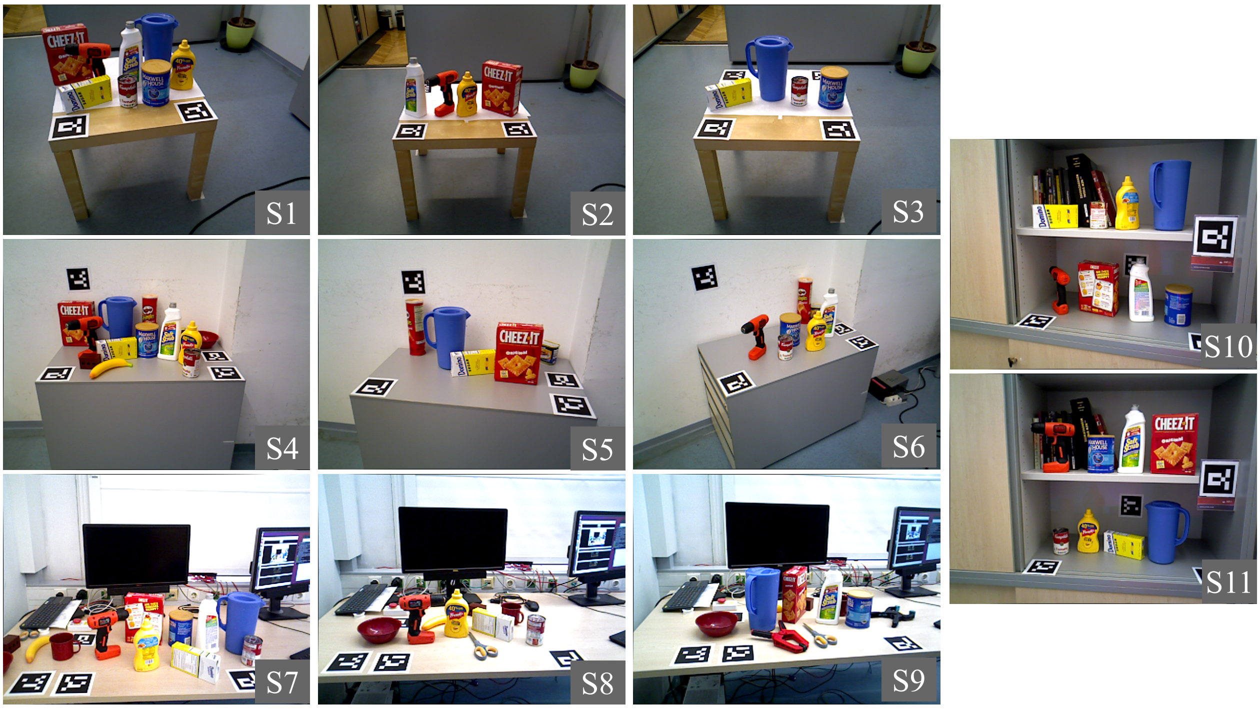
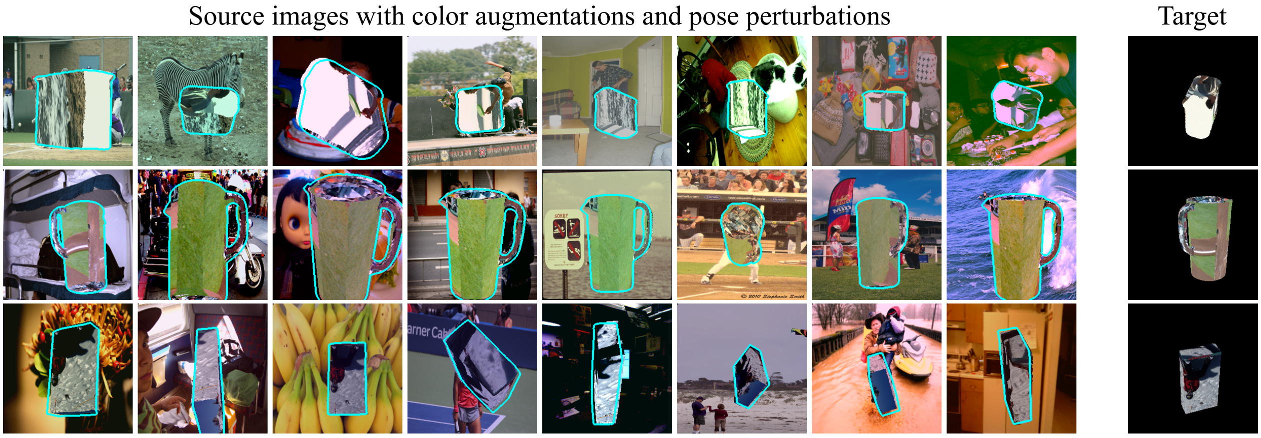


Figure 3. Test sequences of SMOT


Source images with color augmentations and pose perturbations

Target

Figure 4. Examples of training batches

and i7-6700 CPU. Due to the limitation of our GPU memory, weights are updated after every 10 iterations using average gradient values of the last 10 iterations, which is equivalent to 10 mini batches per iteration. Table 2 reports ranges of color augmentations and pose perturbations used for training.

Color augmentation					Pose augmentation	
Color add	Contrast norm	Multiply	Gaussian blur	Addictive noise	ΔTranslation(m)	ΔRotation(rad)
$\mathcal{U}(-15, 15)$	$\mathcal{U}(0.8, 1.3)$	$\mathcal{U}(0.8, 1.2)$	$\mathcal{U}(0.0, 0.5)$	$\mathcal{N}(0, 10)$	$\mathcal{U}(-0.01, 0.01)$	$\mathcal{U}(-0.05, 0.05)$

Table 2. Parameters of color augmentations and pose perturbations

2.3. Network architectures

Fig. 5 and Fig. 6 show architectures of each module in the NOL pipeline.

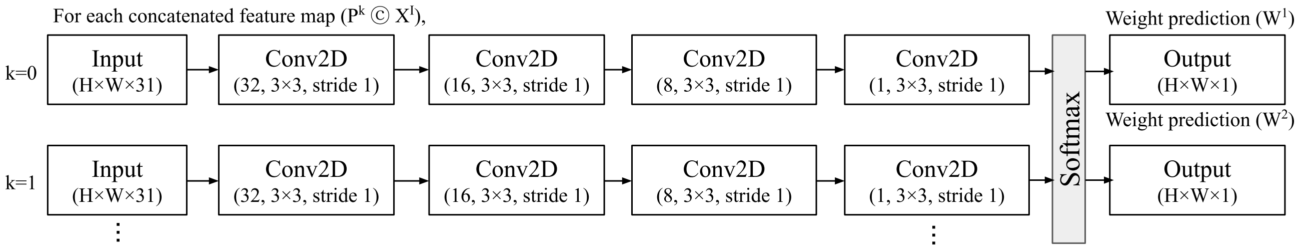

For each concatenated feature map (P^k ⓒ X^l),
Weight prediction (W^1)
k=0
Input (H×W×31)
Conv2D (32, 3×3, stride 1)
Conv2D (16, 3×3, stride 1)
Conv2D (8, 3×3, stride 1)
Conv2D (1, 3×3, stride 1)
Softmax
Output (H×W×1)
Weight prediction (W^2)
k=1
Input (H×W×31)
Conv2D (32, 3×3, stride 1)
Conv2D (16, 3×3, stride 1)
Conv2D (8, 3×3, stride 1)
Conv2D (1, 3×3, stride 1)
Output (H×W×1)

Figure 5. The architecture of the weight prediction block

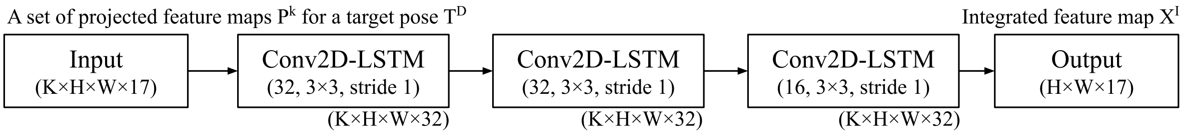

A set of projected feature maps P^k for a target pose T^D
Integrated feature map X^l
Input (K×H×W×17)
Conv2D-LSTM (32, 3×3, stride 1)
(K×H×W×32)
Conv2D-LSTM (32, 3×3, stride 1)
(K×H×W×32)
Conv2D-LSTM (16, 3×3, stride 1)
(K×H×W×32)
Output (H×W×17)

Figure 6. The architecture of the LSTM block that integrates projected feature maps

3. Object-Wise Results of The SMOT Evaluation

Table 3. Object-wise results of the SMOT evaluation. The ADD metric is used except for symmetric objects marked with (*) that are evaluated with the ADI metric

Type	RGB				RGB-D (ICP3D)			
3D Model	Precise			Recont	Precise			Recont
Train source	Real	G2Ltex	**NOL**	**NOL**	Real	G2Ltex	**NOL**	**NOL**
cracker_box	30.8	24.8	49.5	45.4	85.2	92.5	96.3	88.6
bleach_cleanser	19.1	27.5	32.7	12.8	94.0	89.9	93.6	64.9
driller	23.8	2.3	19.8	26.0	87.8	53.9	96.4	91.2
mustard	2.0	33.2	25.9	19.0	88.3	73.8	89.7	82.0
pitcher*	25.9	21.7	30.9	34.8	93.3	92.9	88.7	96.1
tomato_can*	36.7	17.5	41.3	11.1	86.9	79.0	84.9	71.7
maxwell_can*	37.6	40.9	54.7	18.3	95.3	94.2	93.3	84.6
sugar_box	23.7	37.9	29.0	12.3	60.8	79.9	76.9	55.2
Average	25.0	25.7	35.5	22.5	86.5	82.0	90.0	79.3

4. Sensitivity to Different Shapes

Fig. 7 provides experimental results that show the sensitivity of NOL with different shapes (cylindrical, box) and pose errors of input images. A synthetic setup is used to ensure precise GT poses and 3D models. For a target pose of an object, we render 5 source images from different viewpoints. 3D models of *cracker_box* and *mastershef_can* (in SMOT, we label this object as *maxwell_can* since it has a different texture) in YCB-V are used to compare the results for different shapes. 50 sets (a set consists of a GT image, a target pose, and 5 source images) are created for each object. We directly measure the image quality using perceptual similarity [2] between GT and NOL images while increasing the range of pose errors of source images.

The results clearly show that the quality of images becomes worse with larger errors. On the other hand, the proposed refinement step successfully reduces the gap between rendered and target images regardless of the shapes of objects.

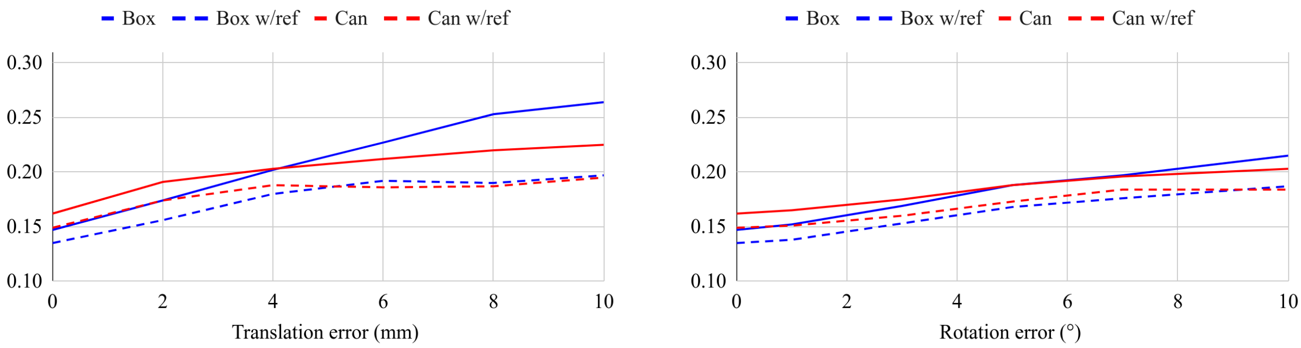


Figure 7. Perceptual similarity (smaller is better) of rendered images with translation and rotational errors in source images

5. Geometrical Errors in Reconstructed Models of SMOT Objects

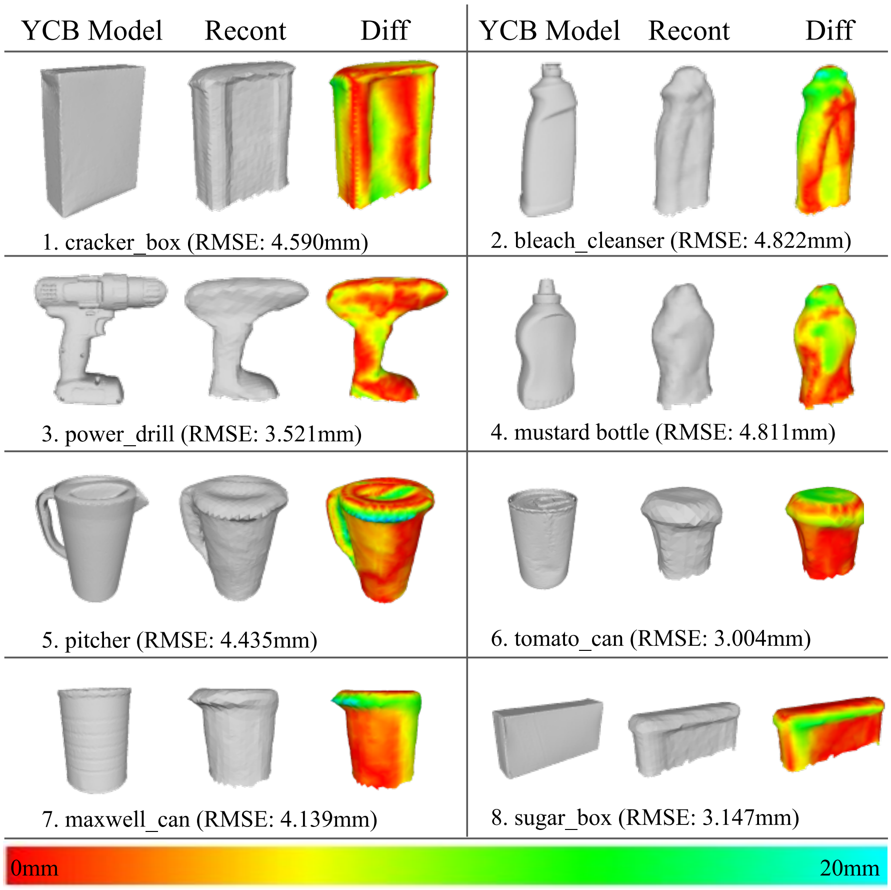

YCB Model
Recont
Diff
YCB Model
Recont
Diff
1. cracker_box (RMSE: 4.590mm)
2. bleach_cleanser (RMSE: 4.822mm)
3. power_drill (RMSE: 3.521mm)
4. mustard bottle (RMSE: 4.811mm)
5. pitcher (RMSE: 4.435mm)
6. tomato_can (RMSE: 3.004mm)
7. maxwell_can (RMSE: 4.139mm)
8. sugar_box (RMSE: 3.147mm)
0mm
20mm

Figure 8. Geometrical errors measured with the Hausdorff distance. The quality of NOL images drops significantly with geometrical errors.

References

[1] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. 1

[2] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3