GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence Supplementary Material

Van Nguyen Nguyen[1] Thibault Groueix[2] Mathieu Salzmann[3] Vincent Lepetit[1]
[1]LIGM, École des Ponts [2]Adobe [3]EPFL

1. Ground-truth 2D-to-2D correspondences

As discussed in Section 3.2 of the main paper, we use the ground-truth 3D information of training sets provided by the BOP challenge [10], originally sourced from MegaPose [7] to create the 2D-to-2D correspondences for training.

For each 2D location i—the 2D center for a patch of size 14×14 of the query image—we aim to identify its corresponding location i^* in the nearest template. We achieve this through a straightforward re-projection process. For each 2D center in the query image, we first calculate its 3D counterpart using the query depth map and camera intrinsics. We then transform this 3D point into the camera view of the nearest template using the ground-truth relative pose, and re-project this 3D point into the template using template camera intrinsics. If the re-projected 2D location falls inside the template mask, we identify the nearest patch i^* among all patches within the template mask, as the corresponding location for the input query patch i.

We reverse the roles of the query and the template, then use the same process to establish 2D-to-2D correspondences for each patch of the template.

We use color augmentation to close the real-synthetic domain gap as done in [7], including: Gaussian blur, contrast, brightness, colors and sharpness filters from the Pillow library [2]. We show in Figure 1 eight training samples created from this process.

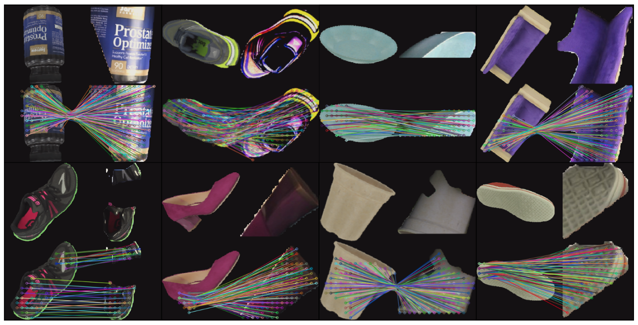
Figure 1. **Training samples.** For each training sample (2×2 images), we show the template image $\mathcal{T}_k$ (top left), the query image $\mathcal{Q}_k$ (top right) and the 2D-to-2D correspondences (bottom) as discussed in Section 1.

2. Recovering a 6D object pose

For simplicity, in this section, we denote the input template and the input testing image *before* the processing step (i.e., scaling, cropping, and padding) as $\mathcal{T}$ and $\mathcal{Q}$, respectively.

As mentioned in Section 3 of the main paper, we decompose the 6D object pose into out-of-plane rotation $\mathbf{R}_{\text{ae}}$ and the affine transform $\mathbf{M}_{t\rightarrow q}$ (including in-plane rotation α, 2D scale and 2D translation), which transforms the processed template to the processed query. We detail below how we can recover the 6D object pose from $\mathbf{R}_{\text{ae}}$ and $\mathbf{M}_{t\rightarrow q}$.

First, the 3 DoF for the rotation can be recovered by simply combining the out-of-plane rotation $\mathbf{R}_{\text{ae}}$, annotated alongside the nearest template, with the in-plane rotation, α, as predicted by the network $\mathbf{F}_{\text{ist}}$:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_\alpha \mathbf{R}_{\text{ae}} \\ &= \begin{bmatrix}\cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1\end{bmatrix} \mathbf{R}_{\text{ae}}\ .\end{aligned} \tag{1}$$

Recovering 3 DoF for object translation in the test image involves additional transformations, $\mathbf{M}_\mathcal{Q}$ and $\mathbf{M}_\mathcal{T}$, for scaling, cropping, and padding the input testing image $\mathcal{Q}$ and the input template $\mathcal{T}$, respectively. These transformations are used to standardize the input images to a fixed size of 224×224, and can be defined as:

$$\mathbf{M}_\mathcal{T} = \begin{bmatrix}\mathbf{s} & 0 & \mathbf{t}_x \\ 0 & \mathbf{s} & \mathbf{t}_y \\ 0 & 0 & 1\end{bmatrix}\ , \tag{2}$$

where $\mathbf{s}$ is the scaling factor, and $[\mathbf{t}_x, \mathbf{t}_y]$ is the 2D translation, created from the scaling, cropping, and padding applied to the template. Similarly, we can define $\mathbf{M}_\mathcal{Q}$, the transformation applied to the input testing image $\mathcal{Q}$.

As shown in Figure 2, the transformation $\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$ transforming a 2D point in the template $\mathcal{T}$ to its 2D corresponding

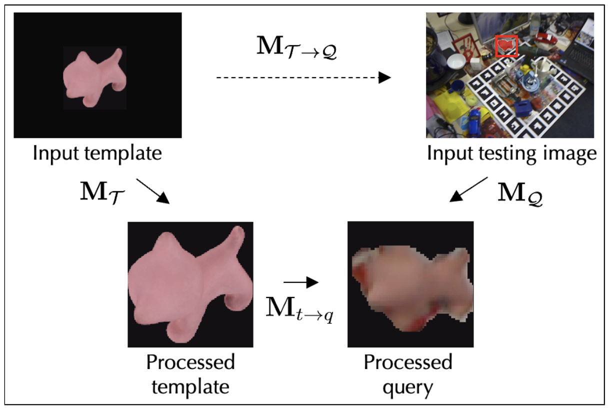
$\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$
Input template
Input testing image
$\mathbf{M}_{\mathcal{T}}$
$\mathbf{M}_{\mathcal{Q}}$
$\mathbf{M}_{t\rightarrow q}$
Processed template
Processed query

Figure 2. **Transformations from template $\mathcal{T}$ to query $\mathcal{Q}$.** We show all the transformations has been applied to transform the template $\mathcal{T}$ to the input testing image $\mathcal{Q}$, as discussed in Section 2.

point in the query $\mathcal{Q}$ can be defined explicitly as:

$$\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}} = \mathbf{M}_{\mathcal{T}}\mathbf{M}_{t\rightarrow q}\mathbf{M}_{\mathcal{Q}}^{-1} \, . \tag{3}$$

We can therefore use the transformation $\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$ to recover the 2D projection of the object's translation in the query image, $[\mathbf{c}_{\mathcal{Q},x}, \mathbf{c}_{\mathcal{Q},y}]$ given the 2D projection of object's translation in the template, $[\mathbf{c}_{\mathcal{T},x}, \mathbf{c}_{\mathcal{T},y}]$:

$$\begin{bmatrix}\mathbf{c}_{\mathcal{Q},x}\\ \mathbf{c}_{\mathcal{Q},y}\\ 1\end{bmatrix} = \mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}} \begin{bmatrix}\mathbf{c}_{\mathcal{T},x}\\ \mathbf{c}_{\mathcal{T},y}\\ 1\end{bmatrix} \, . \tag{4}$$

The only missing degree is the object's translation of the query image in Z axis, $\mathbf{t}_{\mathcal{Q},z}$, which can be deduced from $\mathbf{t}_{\mathcal{T},z}$, the object's translation of the template in Z axis, $\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$ and the focal ratio using the following formula:

$$\mathbf{t}_{\mathcal{Q},z} = \mathbf{t}_{\mathcal{T},z} \times \frac{1}{\text{scale}\left(\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}\right)} \times \frac{\mathbf{f}_{\mathcal{Q}}}{\mathbf{f}_{\mathcal{T}}} \, , \tag{5}$$

where scale $(\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}})$ is the 2D scale in $\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$, which is equal to the norm of first column of $\mathbf{M}_{\mathcal{T}\rightarrow\mathcal{Q}}$, and $\mathbf{f}_{(.)}$ is the focal length.

Finally, we calculate the object's translation in the query image $[\mathbf{t}_{\mathcal{Q},x}, \mathbf{t}_{\mathcal{Q},y}, \mathbf{t}_{\mathcal{Q},z}]$ using the query camera intrinsic $\mathbf{K}_{\mathcal{Q}}$:

$$\begin{bmatrix}\mathbf{t}_{\mathcal{Q},x}\\ \mathbf{t}_{\mathcal{Q},y}\\ \mathbf{t}_{\mathcal{Q},z}\end{bmatrix} = \mathbf{t}_{\mathcal{Q},z} \times \left(\mathbf{K}_{\mathcal{Q}}^{-1} \begin{bmatrix}\mathbf{c}_{\mathcal{Q},x}\\ \mathbf{c}_{\mathcal{Q},y}\\ 1\end{bmatrix}\right) \, . \tag{6}$$

3. "2D" version of the Kabsch algorithm

The classic Kabsch algorithm [5] has been commonly used for points in a three-dimensional space. In this work, we use a "2D" version for two-dimensional space. This allows us to recover the affine transformation $\mathbf{M}_{t\rightarrow q}$, including the 2D scale $\mathbf{s}$, in-plane rotation $\mathbf{R}_{\alpha}$, and 2D translation $\mathbf{t}$ from two 2D-to-2D correspondences.

Let denote $\{(\mathbf{p}_{\mathcal{T}}^1, \mathbf{p}_{\mathcal{Q}}^1), (\mathbf{p}_{\mathcal{T}}^2, \mathbf{p}_{\mathcal{Q}}^2)\}$ two correspondences that we obtain from the nearest neighbor search with the features of $\mathbf{F}_{\text{ae}}$. Our goal is to find $\{\mathbf{s}, \mathbf{R}_{\alpha}, \mathbf{t}\}$ which transforms $\mathbf{p}_{\mathcal{T}}^1$ to $\mathbf{p}_{\mathcal{Q}}^1$, and $\mathbf{p}_{\mathcal{T}}^2$ to $\mathbf{p}_{\mathcal{Q}}^2$:

$$\begin{aligned}\mathbf{p}_{\mathcal{Q}}^1 &= \mathbf{s} \times \mathbf{R}_{\alpha}\mathbf{p}_{\mathcal{T}}^1 + \mathbf{t} \, ,\\ \mathbf{p}_{\mathcal{Q}}^2 &= \mathbf{s} \times \mathbf{R}_{\alpha}\mathbf{p}_{\mathcal{T}}^2 + \mathbf{t} \, .\end{aligned} \tag{7}$$

First, we calculate the scale $\mathbf{s}$ from the size of two vectors $\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1$ and $\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1$:

$$\mathbf{s} = \frac{||\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1||}{||\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1||} \, . \tag{8}$$

The rotation matrix $\mathbf{R}_{\alpha}$ is composed of $\cos(\alpha)$ and $\sin(\alpha)$, and is defined as:

$$\mathbf{R}_{\alpha} = \begin{bmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{bmatrix} \, , \tag{9}$$

where $\cos(\alpha)$ and $\sin(\alpha)$ are the dot product and cross product respectively of vectors $\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1$ and $\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1$:

$$\cos(\alpha) = \frac{\left(\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1\right)^T . \left(\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1\right)}{||\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1||.||\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1||} \, , \tag{10}$$

$$\sin(\alpha) = \frac{\left(\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1\right)^T \wedge \left(\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1\right)}{||\mathbf{p}_{\mathcal{T}}^2 - \mathbf{p}_{\mathcal{T}}^1||.||\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{p}_{\mathcal{Q}}^1||} \, . \tag{11}$$

Given the predicted scale $\mathbf{s}$ and rotation matrix $\mathbf{R}_{\alpha}$, we can deduce translation $\mathbf{t}$:

$$\mathbf{t} = \frac{1}{2}\left[\left(\mathbf{p}_{\mathcal{Q}}^1 - \mathbf{s} \times \mathbf{R}_{\alpha}\mathbf{p}_{\mathcal{T}}^1\right) + \left(\mathbf{p}_{\mathcal{Q}}^2 - \mathbf{s} \times \mathbf{R}_{\alpha}\mathbf{p}_{\mathcal{T}}^2\right)\right] \, . \tag{12}$$

4. Additional results

4.1. Using 3D models predicted by Wonder3D

As discussed in Section 4 of the main paper, due to the sensitivity of Wonder3D to the quality of input images, we selected reference images from the test set of LM [4] based on three criteria: (i) not present in the test set of LM-O [1], (ii) the target object is fully visible and (iii) well segmented by Segment Anything [6]. Despite these careful selections, we observe that Wonder3D can still fail due to the low resolution or the lack of "perspective" information in the input image. Figure 4 illustrates these common failure cases of Wonder3D where objects appear "flat" when viewed from novel angles.

We therefore select reference images for each object, sourced from "scene_id/im_id" images as follows (sorted by "object_id"): "000001/000693", "000005/000775", "000006/000949", "000008/000994", "000009/001228", "000010/000289", "000011/001069", and "000012/000647".

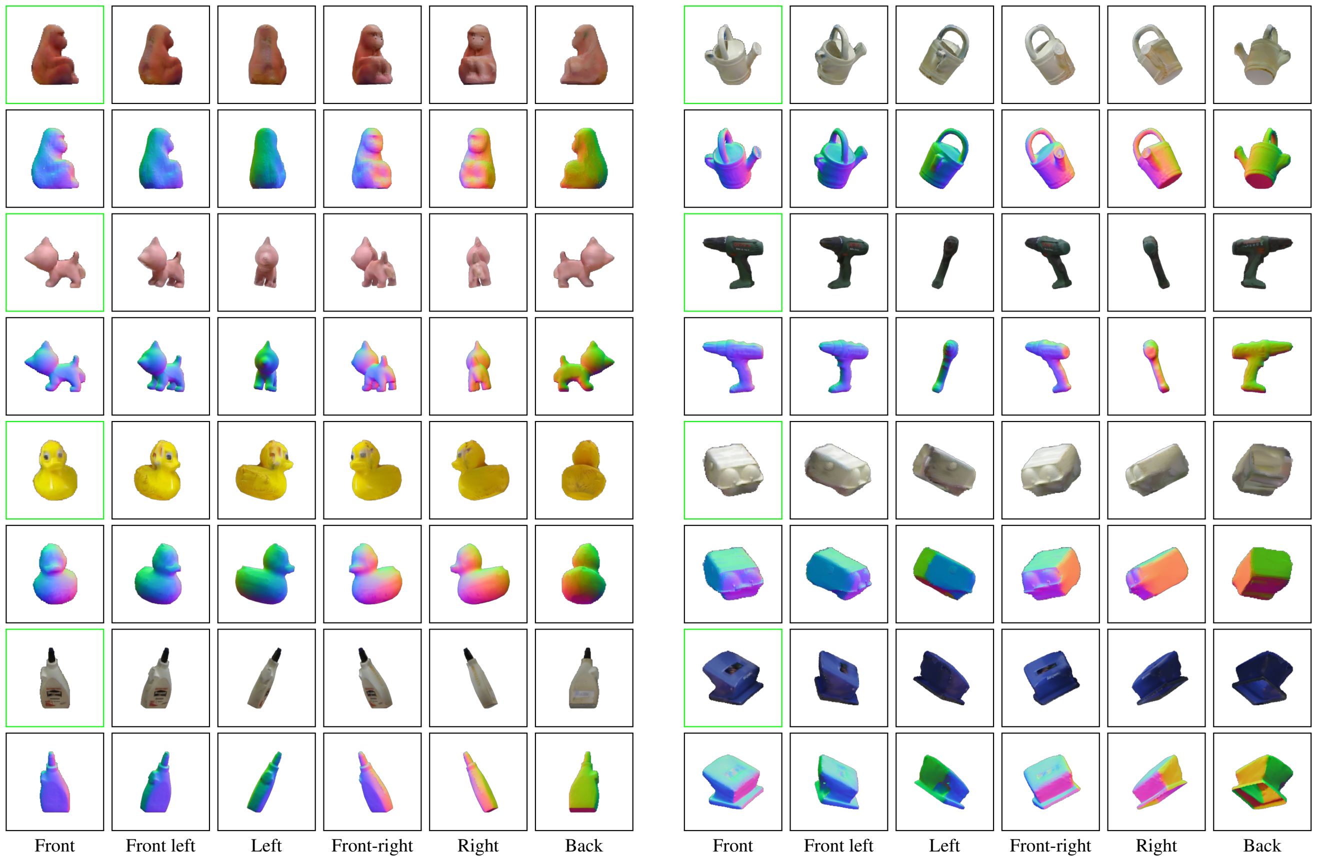

Figure 3. **3D reconstruction by Wonder3D [8]**. For each sample (2×6 images), we show the input image outlined in green (top left), the predicted rgb (second to last column of the first row), and the second row shows the corresponding predicted normals.

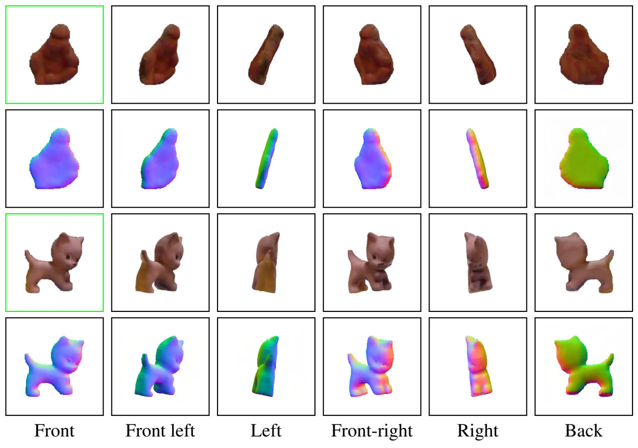

Figure 4. **Failure cases of Wonder3D [8]**. We present common failure cases of the Wonder3D with the "ape" and "cat" objects from the LM-O dataset [1]. For each sample (2×6 images), we show the input image outlined in green (top left), the predicted rgb (second to last column of the first row), and the second row shows the corresponding predicted normals. These objects appear "flat" when viewed from novel angles.

Input	Detection	Segmentation
GT 3D model	43.3	39.7
Predicted 3D model	38.3	36.3

Table 1. **CNOS [9]'s performance on LM-O dataset [1].** For this evaluation, we use the standard protocol designed for detection and segmentation tasks, used in Tasks 5 and 6 of the BOP Challenge of BOP challenge [10].

Figure 3 presents additional visualizations of 3D reconstruction from a single image by Wonder3D [8], using these images. It is important to note that the final 3D models are reconstructed from these six views using the instant-NGP based SDF reconstruction method [3]. The reference is defined as the front view, while the five predicted views are the front-left, left, front-right, right, and back.

We show in Table 1 CNOS's results for both detection and segmentation tasks when using 3D models predicted from a single image by Wonder3D [8].

4.2. Using ground-truth 3D models

We show in Figure 5, and 6 qualitative results on challenging conditions of LM-O [1] and YCB-V [11] datasets.

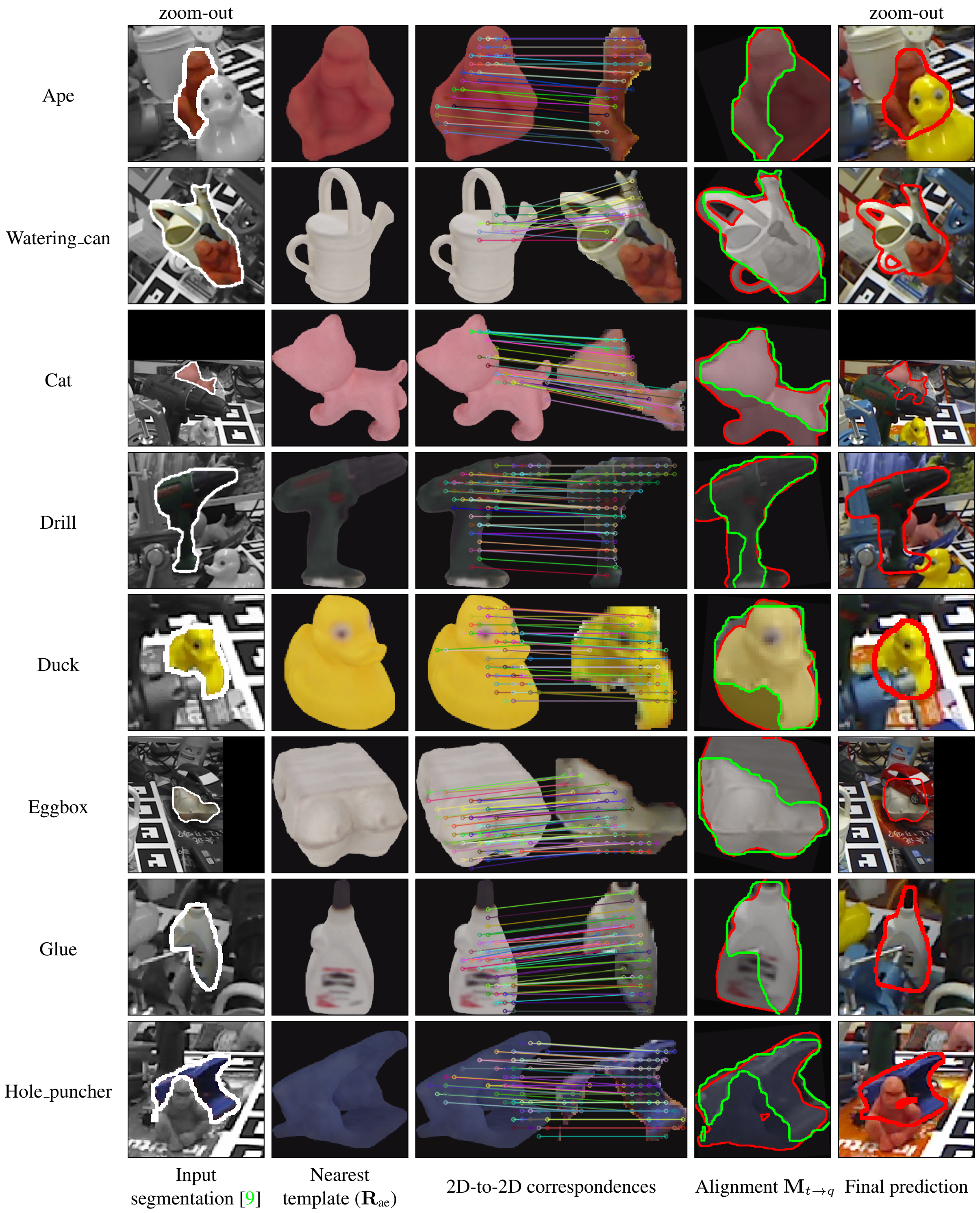

Figure 5. **Qualitative results on LM-O [1]**. The first column shows CNOS [9]'s segmentation. The second and third columns illustrate the outputs of the nearest neighbor search step, which includes the nearest template ($\mathbf{R}_{\text{ae}}$) and the 2D-to-2D correspondences. The fourth column demonstrates the alignment achieved by applying the predicted affine transform $\mathbf{M}_{t\rightarrow q}$ to the template, then overlaying it on the query input: The green contour indicates the noisy segmentation by CNOS [9], while the red contour highlights the boundary of the aligned template. The last column show the final prediction after refinement [7].

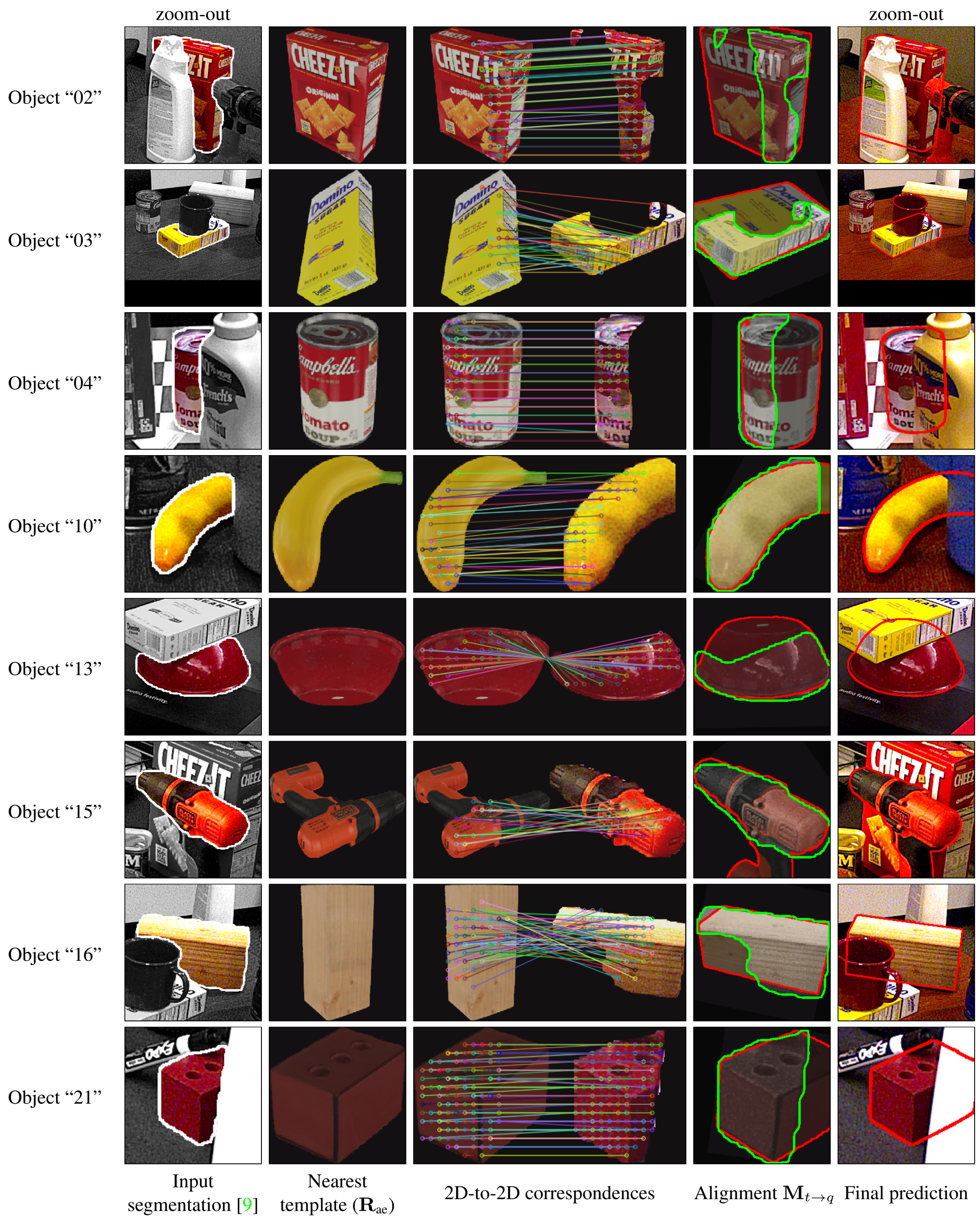

Figure 6. **Qualitative results on YCB-V [11]**. The first column shows CNOS [9]'s segmentation. The second and third columns illustrate the outputs of the nearest neighbor search step, which includes the nearest template ($\mathbf{R}_{\text{ae}}$) and the 2D-to-2D correspondences. The fourth column demonstrates the alignment achieved by applying the predicted affine transform $\mathbf{M}_{t\rightarrow q}$ to the template, then overlaying it on the query input: The green contour indicates the noisy segmentation by CNOS [9], while the red contour highlights the boundary of the aligned template. The last column show the final prediction after refinement [7].

5. Future work

As discussed in Section 4.3 of the main paper, GigaPose fails on challenging conditions where heavy occlusions, low-resolution segmentation, and low-fidelity CAD models are present, as observed in the LM-O dataset. To address this issue, incorporating additional modalities, such as depth images, can be beneficial. Depth images, which capture information about object geometries, can significantly enhance model performance under these conditions. Moreover, although our method shows promising results in a single-reference setting using Wonder3D [8], it requires manual selection to achieve high-quality 3D reconstruction. Therefore, the development of more advanced 3D reconstruction techniques capable of generating high-quality outputs from a single image would be particularly valuable in this context.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *ECCV*, 2014. 2, 3, 4

[2] Alex Clark and Others. The Pillow Imaging Library. https://github.com/python-pillow/Pillow. *IEEE Robot. Autom. Mag.*, 22(3), Sept. 2015. 1

[3] Yuan-Chen Guo. Instant neural surface reconstruction, 2022. https://github.com/bennyguo/instant-nsr-pl. 3

[4] Stefan Hinterstoißer, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *ACCV*, 2012. 2

[5] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32:922–923, 1976. 2

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, and Others. Segment Anything. In *arXiv*, 2023. 2

[7] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*, 2022. 1, 4, 5

[8] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 3, 6

[9] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A Strong Baseline for CAD-based Novel Object Segmentation. In *ICCV Workshops*, 2023. 3, 4, 5

[10] Martin Sundermeyer, Tomáš Hodaň, Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. BOP Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects. In *CVPR Workshops*, 2023. 1, 3

[11] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *RSS*, 2018. 3, 5