

Appendix A Pseudo-code of Per-image Ground Truth Pose Annotation

In this section, we give the key elements of our annotation method with the following pseudo-codes.

```

1 # Input: CAD model, candidate transformation set, threshold
2 # Output: per point epsilon sym set
3 def EpsSym(ModelCAD, TransformSet,  $\epsilon$ ,  $\zeta$ , resolution):
4     SampledModel = uniform_resampling(ModelCAD, resolution)
5     for point in SampledModel:
6         for T in TransformSet:
7             # Texture-less case
8             if testColor == False:
9                 if knnSearch(T*point, SampledModel,  $\epsilon$ ) == True:
10                     EpsSym[point].append(T)
11             # Textured case
12             else:
13                 if knnSearch(T*point, SampledModel,  $\epsilon$ ) == True:
14                     if  $d_{Color}(\text{point}, T*\text{point}) < \zeta$ :
15                         EpsSym[point].append(T)
16
17 return EpsSym, SampledModel

```

Listing 1 Offline ϵ -sym pre-computation with all vertices visible, pseudo-code of Equation 3 from Section 3.1.

The geometric (d_{Geom}) and colorimetric (d_{Color}) distances are implemented as follows:
 $d_{Geom}(x, m) = \|x - m\|_2$ and

$$\begin{aligned}
 d_{Color}(x, m) < \zeta &\Leftrightarrow \min \left(|h(x) - h(m)|, \right. \\
 &\quad \left. |h(x) - h(m) - 360|, |h(x) - h(m) + 360| \right) < \zeta_h, \\
 &\quad \text{AND } |s(x) - s(m)| < \zeta_s, \\
 &\quad \text{AND } |v(x) - v(m)| < \zeta_v.
 \end{aligned}$$

with $h(\cdot), s(\cdot), v(\cdot)$ being hue, saturation and value of the point, and $\zeta_h, \zeta_s, \zeta_v$, the respective hue, saturation and value thresholds.

```

1 # Input: SampledModel, EpsSym, PoseGT, MaskVisib,  $\tau$ 
2 # Output: EpsSymImage
3 def SoftIntersection(SampledModel, EpsSym, PoseGT, MaskVisib,  $\tau$ ):
4     # Count the vertices that vote for a transform
5     for point in SampledModel:
6         if K*[ $R_{gt}$ ,  $T_{gt}$ ]*point in mask_visib:
7             for T in EpsSym[point]:
8                 H[T]++
9     Sort(H)
10    # Count the vertices that vote for a transform
11    for i in size(H):
12        if H[0]-H[i] <  $\tau$ :
13            EpsSymImage.append(i)
14
15 return EpsSymImage

```

Listing 2 Image annotation, pseudo-code of Equation 4 from Section 3.2.

```

1 # Input: ModelCAD, EpsSymImage, PoseGT, SensorDepth,  $\delta$ 
2 # Output: EpsSymImageGlobal
3 def PostProcessAnnotateImage(ModelCAD, EpsSymImage, PoseGT, SensorDepth,  $\delta$ ,
4     threshold):
5     depthGT = render(ModelCAD, PoseGT)

```

```

5  distGT = depthToDistImage(depthGT)
6  visibleMaskGT = generateMask(depthGT, SensorDepth)
7  for Ti in EpsSymImage:
8      depthEst = render(ModelCAD, Ti)
9      distEst = depthToDistImage(depthEst)
10     visibleMaskEst = generateMask(depthEst, SensorDepth)
11     maskIntersection = intersection(visibleMaskGT, visibleMaskEst)
12     maskUnion = union(visibleMaskGT, visibleMaskEst)
13     nbPixelsOutlier += maskUnion - maskIntersection
14     nbPixelsOutlier += abs(distGT - distEst)[maskIntersection] >=  $\delta$ 
15     if nbPixelsOutlier < threshold:
16         EpsSymImageGlobal.append(Ti)
17
18 return EpsSymImageGlobal

```

Listing 3 Image annotation depth post-processing, pseudo-code of Equation 4 from Section 4.5, based on VSD implementation from BOP toolkit.

Section 4.5 does not use the VSD metric, as VSD is normalized by the number of pixels in the union of the visible masks. In our case, we need an absolute score and not a relative one, so that the counting of outliers has a metric meaning and can represent the size of a minimal disambiguating element.

Appendix B Analysis of our BOP-Distrib Annotations

B.1 Additionnal BOP-Distrib Ground Truths Visualizations

We first provide more visualizations for qualitative appreciation of the new ground truth annotations accuracy in Figure B5.

These images are taken from a video compilation of all ground truth annotations sorted by object identifier, also provided as supplementary material ([BOP_Distrib_id8513.supp_newGT_visualizations.mp4](#)), to convince the reader of their quality. We invite the reader to stop on some frames and check that the distribution recovered by our method does correspond to the ambiguities in the image for the object in the bounding box.

B.2 T-LESS [14] Annotation Details

In our experiments, we sample surface points from the CAD models with a resolution of 0.5mm. For the pre-computation of elementary symmetries patterns for a given object, we use the per-object symmetries pattern given by the BOP challenge [1] as the initial symmetry candidates, with a tolerance factor ϵ -sym set to 1mm.

The object surface visibility $\mathcal{V}(M)$ is computed by Z-buffering using ground truth pose P_{GT} and the 3D model of the object. For the robust symmetry pattern intersection, the soft intersection tolerance factor τ was experimentally adjusted to 28 3D points, resulting in a minimal disambiguating element size of roughly $2.5 \times 2.5 \text{mm}^2$, $\delta = 5 \text{mm}$ and $\tau_{pix} = 30 \text{pix}$.

B.3 YCB-V [15] Annotation Details

In our experiments, we sample surface points from the CAD models with a resolution of 1mm. For the pre-computation of elementary symmetries patterns for a given object,

we use the per-object symmetries pattern given by the BOP challenge [1] as the initial symmetry candidates, with a tolerance factor ϵ -sym set to 2mm. The color tolerance is set in the HSV color space to have the chrominance on a single channel (hue) and only luminance on the other two (saturation and value). It is empirically set to 4° in hue and 0.1 in saturation and value.

The object surface visibility $\mathcal{V}(M)$ is computed by Z-buffering using ground truth pose P_{GT} and the 3D model of the object. For the robust symmetry pattern intersection, the soft intersection tolerance factor τ was experimentally adjusted to 28 3D points, resulting in a minimal disambiguating element size of roughly $5.3 \times 5.3 \text{mm}^2$. The pruning stage was not necessary for YCB-V, as objects are simpler, with much less occlusions.

B.4 Differences between Original Object-wise BOP Annotations and Our Image-wise Annotations

Figure B1 for T-LESS [14] and Figure B2 for YCB-V [15] highlight the differences between the poses that are accepted by BOP and the ones accepted when using our annotations. For this purpose, for each object, we provide a bar plot. This bar plot shows, for each image where the object appears, the percentage of poses considered correct by BOP that are also considered correct with our annotations. The bar plots show the percentages after sorting them, i.e., the bar on the left corresponds to the image with the smallest difference.

Concerning T-LESS, the annotations for some objects remain mostly unchanged. But because our method analyses more finely the possible object symmetries, many poses are actually not accepted with our annotations for the other objects (mainly the 'circular' ones). T-LESS is a very good dataset for our per-image pose distribution annotation method as it features objects with complex symmetries as well as a lot inter-object occlusions.

Concerning YCB-V, some objects such as the mug (object 14 on Figure B2) could yield very interesting symmetries with occlusions [2]. However, the disambiguating handle of the mug is never occluded. Hence the BOP symmetries that tag the mug as unambiguous. Overall, YCB-V has less potential for displaying visual ambiguities. Most objects are disambiguated by texture and the few ambiguous YCB-V objects do not face sufficient occlusions to become ambiguous (objects 11, 14 and 18 mainly), as depicted by the visualization of the scenes in Figure B3. We show here that our per-image annotation method is able to retrieve finer symmetries patterns, which have an effect when evaluating Single Pose Estimation methods as in Table 3.

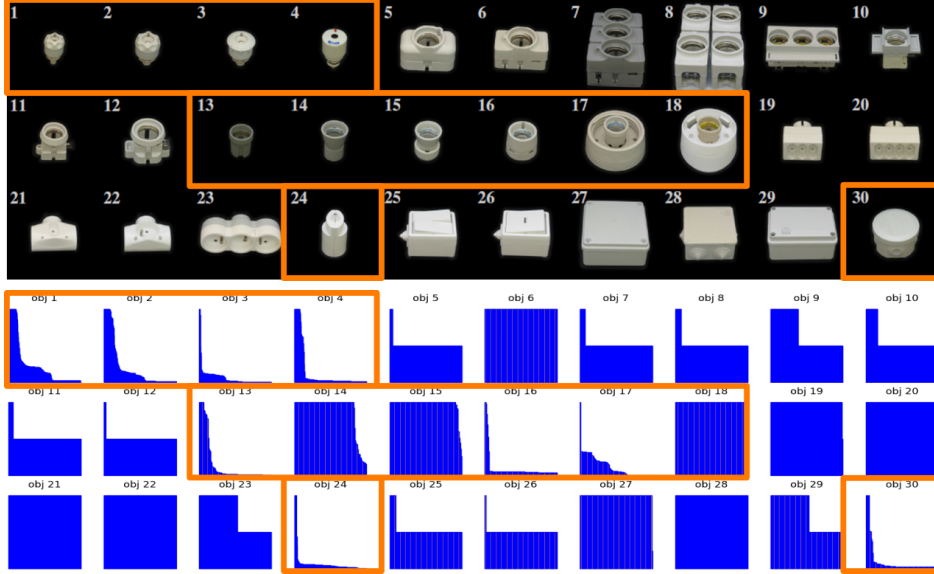


Fig. B1 Visualization of annotation changes compared to the T-LESS original annotations. Top: T-LESS objects with their identifiers. **Bottom:** For each object, we plot the percentages of poses kept from the original annotations by our method over the images (sorted by percentages). T-LESS assumes full rotational symmetries, while our annotation method captures more complex symmetry patterns. Only Object 18 is perfectly symmetrical and our method retrieves the same poses as the original annotations. For the other objects, in particular the objects with complex symmetry patterns like the first 4 objects, our annotations significantly change the original annotations. These changes in the annotations explain the score changes for T-LESS in Table 3. The objects annotated as ‘circular’ in BOP are highlighted in orange.

Appendix C Computing the Pose Estimation Results

C.1 T-LESS Pose Estimation and Unseen Objects Pose Estimation

Recently, the BOP challenge [5] made public the pose estimation results of the methods evaluated in the leaderboard ⁴. Based on these results and the BOP toolkit ⁵ in which we implemented the variations of **MSSD** and **MSPD**, that use our per-image symmetries patterns instead of BOP global object symmetries, we were able to reprocess these pose estimates against our new ground truths. Our results on T-LESS have been presented in Table 3. Section D of supplementary material illustrates some of the failure cases with the new and more accurate ground truth.

⁴<https://bop.felk.cvut.cz/leaderboards/>

⁵https://github.com/thodan/bop_toolkit/tree/master

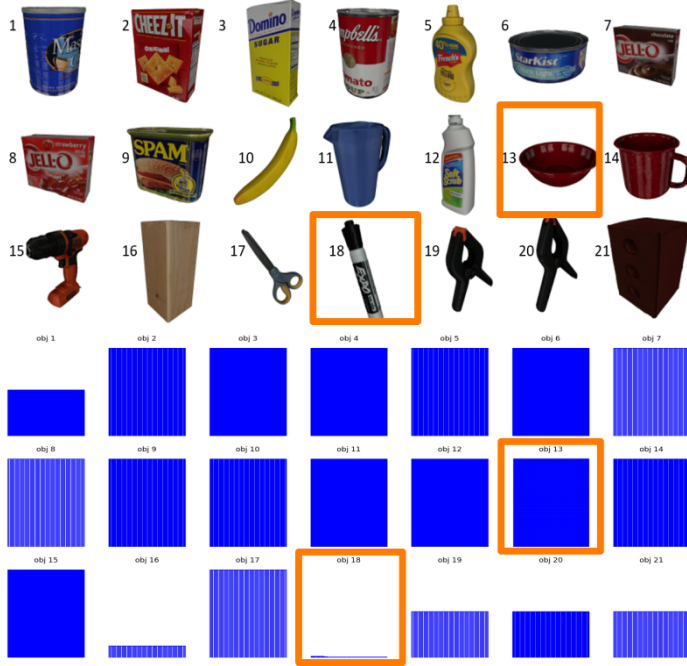


Fig. B2 Visualization of annotation changes compared to the YCB-V original annotations. Top: YCB-V objects with their identifiers. **Bottom:** For each object, we plot the percentages of poses kept from the original annotations by our method over the images (sorted by percentages). YCB-V assumes full rotational symmetries, while our annotation method captures more complex symmetry patterns. Only Object 13 is perfectly symmetrical, both in terms of geometry and texture, and our method retrieves the same poses as the original annotations. Object 18 is given as completely symmetrical but our method tags it as non-ambiguous due to its texture. Similarly, objects 1, 16, 19, 20 and 21 have few symmetrical poses for BOP annotations where our method keep always only one pose, as the texture disambiguate them. These changes in the annotations explain the score changes for YCB-V in Table 3. The objects annotated as 'circular' in BOP are highlighted in orange.

C.2 YCB-V Pose Estimation

We conducted a similar experience of reprocessing BOP competitors on our re-annotation of the YCB-V. Our results have been presented in Table 3.

C.3 Computation of the SpyroPose [3] Distribution Results

Beyond finer evaluation of Single Pose Estimation methods, our new per-image annotations allow us to propose the first evaluation on real data of Pose Distribution Estimation methods. As no pose distribution evaluation existed, the authors of [3] reported only per-object averaged log-likelihood against BOP original ground truth.

The implementation provided by the authors of SpyroPose allows to train a network for one object of T-LESS. We batched the training stage for all objects, and then batched the inference. For one object, SpyroPose produces more than 100 000 estimates, sorted by their probabilities, as it samples $SO(3) \times \mathbb{R}^3$ (using a slice of $\mathbb{R}^3$).



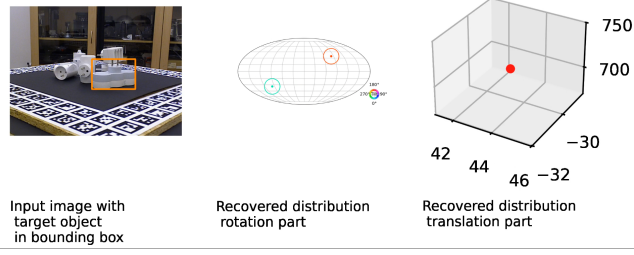
Fig. B3 Visualization of YCB-V scenes. The scenes present at most 6 different objects, with one instance of each. The objects with potential ambiguities (mainly objects 14 and 18) have limited occlusions. YCB-V is not really suited for pose distribution evaluation as no object becomes ambiguous, apart from object 13.



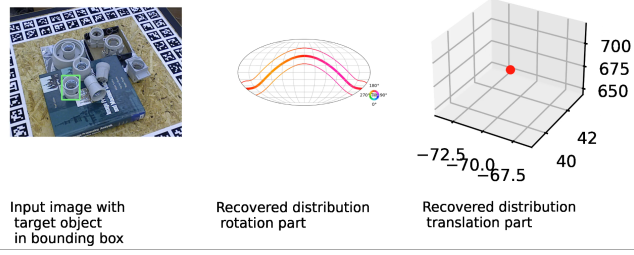
Fig. B4 Visualization of T-LESS scenes. The scenes present up to 18 instances of different objects, with a lot of occlusions and with ambiguous objects. T-LESS is really suited for pose distribution evaluation. (illustration borrowed from [14])

As the probabilities of these estimates quickly tend to zero, we reduce their distributions to the 400 best estimates. These 400 estimates are used for the evaluations of Section 6.4 of the article.

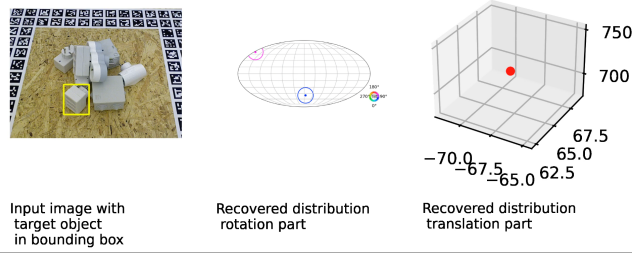
Object 23, Scene 8, Image 441



Object 15, Scene 16, Image 194



Object 19, Scene 13, Image 144



Object 7, Scene 17, Image 121

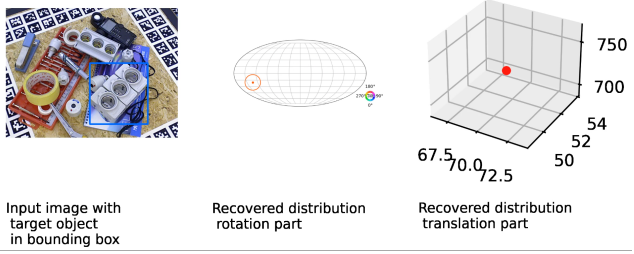


Fig. B5 Visualizing some BOP-Distrib ground truths as computed by our method. Each example features an object of interest, in the bounding box in the left image, and its BOP-Distrib pose distribution, split between the rotation part (center) and translation part (right). As no object present symmetries in translation here, the translation part of the distribution is always on the same point of $\mathbb{R}^3$. We provide much more examples in the accompanying video.

C.4 Computation of the Lie-Pose Diffusion [4] Distribution Results

Similarly, as no pose distribution evaluation existed, the authors of [4] reported only few qualitative illustrations of pose distribution results on T-LESS. They evaluated their method as a Single Pose Estimation method, with a single run of the method.

The implementation provided by the authors of Lie-Pose allows to train a network for all objects of T-LESS. The inference phase produces one pose estimate per image crop. We batched the inference phase, with varying seeds. We then merged all these estimates into a single set of poses per image crop. The authors report 1000 runs to produce a distribution. For our evaluation, due to important computation time, we ran the code 100 times with different input noises to produce Lie-Pose distribution results.

Appendix D Illustrations of Single Pose Re-evaluation

We provide here more cues about the changes in the Single Pose ranking, based on the **MSSD** and **MSPD** metrics using our new more accurate ground truth.

To do so, we look at the pose estimates of the method `gdrnpp-pbrreal-rgbd-mmodelv1.3`. When we evaluated `gdrnpp-pbrreal-rgbd-mmodelv1.3` estimates against our new ground truths, it changed the method ranking from rank 7 to rank 10 in Table 3. Even more interestingly, the metrics went down from an **MSSD** of 88.4 and an **MSPD** of 90.9 to an **MSSD** of 60.6 and an **MSPD** of 63.7.

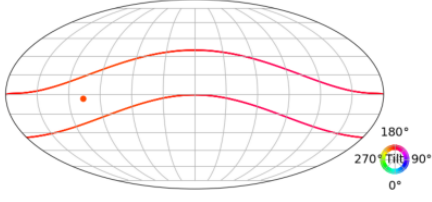
Figure D6 illustrates why **MSSD** and **MSPD** changed for this method. It appears that, although `gdrnpp-pbrreal-rgbd-mmodelv1.3` estimates were close to the ground truth poses, its rotations were not precise. When they are evaluated against a symmetries pattern that is not precise, the evaluation appears correct. Our new ground truth shows that `gdrnpp-pbrreal-rgbd-mmodelv1.3` tends not to align correctly some of the objects. Hence the drop in performances.

Appendix E Visualizing results by SpyroPose [3]

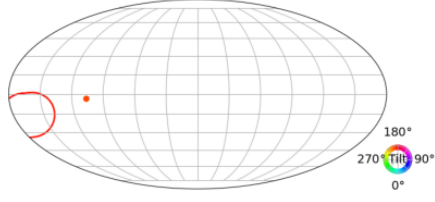
We display some of SpyroPose distribution estimates against our ground truth in Figure E7 on T-LESS. For the case of the three instances of Object 1, SpyroPose correctly retrieves the single mode for Instance 1. The continuous symmetry of Instance 2 is partially retrieved.

Appendix F Visualizing results by LiePose diffusion [4]

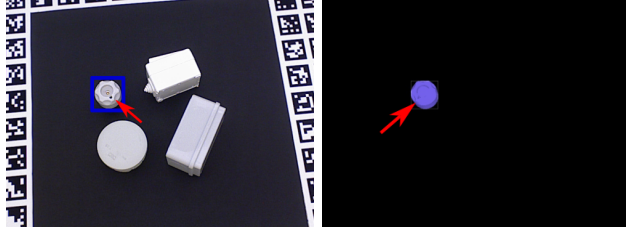
We show some of LiePose [4] distribution estimates against our ground truth in Figure F8 on T-LESS. Similarly to SpyroPose [3], LiePose [4] is able to retrieve the single mode of Instance 1, but gets better results when estimating continuous distributions.



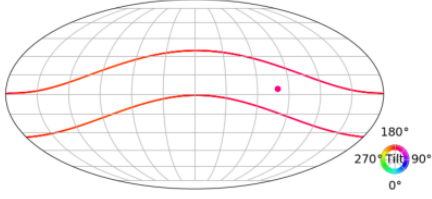
(a) BOP continuous symmetry pattern of Object 2 (envelop) and gdrnpp-pbrreal-rgb-d-mmodelv1.3 estimate (plain circle). The circle belongs to the envelop, yielding a low **MSSD** error of 4.46.



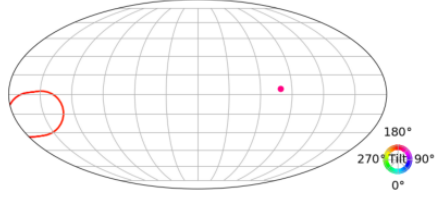
(b) Our visual symmetry pattern (the much smaller envelop) and GDRNPP estimate (plain circle). The circle does not belong to the envelop anymore, the **MSSD** error becomes 21.62.



(c) Corresponding image for (a) and (b): T-LESS Scene 1, Image 17, Object 2 (in bounding box) and rendering of the pose (red arrow highlights disambiguating element).

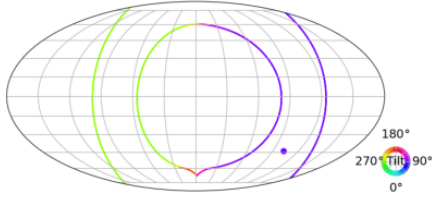


(d) BOP continuous symmetry pattern for Object 1 (envelop) and gdrnpp-pbrreal-rgb-d-mmodelv1.3 estimate (plain circle). The circle belongs to the envelop, yielding a low **MSSD** error of 9.43.

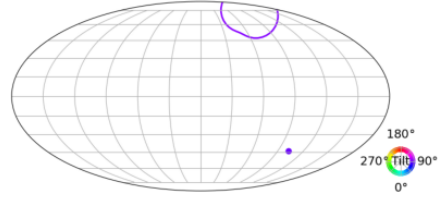


(e) Our visual symmetry pattern (much smaller envelop) and GDRNPP estimate (plain circle). The circle does not belong to the envelop anymore, the **MSSD** error becomes 32.87.

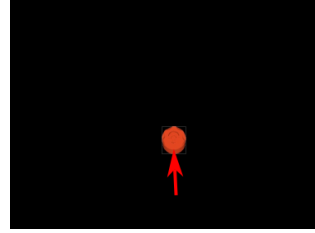
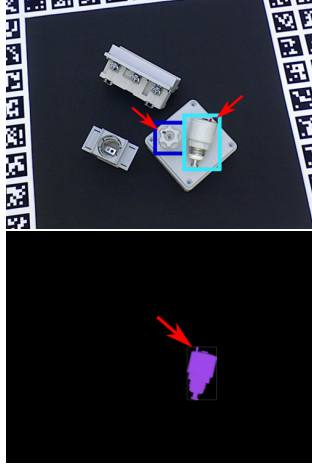
Figures F9, F10, F11 and F12 compare SpyroPose [3] and LiePose [4] results on objects with discrete and continuous symmetries. SpyroPose [3] rotations tends to be more precise than LiePose [4], but misses some of the modes. LiePose [4] tends to estimate continuous symmetries when the image produces discrete ones. These images are taken from a video compilation of SpyroPose [3] and LiePose [4] results also provided as supplementary material ([BOP_Distrib_id8513_supp-distribution-comparison-SpyroPose-LiePose.mp4](#)). Scenes with single instance of objects have been chosen, to



BOP continuous symmetry pattern of Object 4 (envelop) and gdrnpp-pbrreal-rgbd-mmodelv1.3 estimate (plain circle). The circle belongs to the envelop, yielding a low **MSSD** error of 3.33.



Our visual symmetry pattern (much smaller envelop) and GDRNPP estimate (plain circle). The circle does not belong to the envelop anymore, the **MSSD** error becomes 35.38.



(h) Corresponding image for T-LESS Scene 5, Image 70, Objects 1 (d-e) and and 4 (f-g), in bounding boxes) and renderings of the poses (red arrow highlights disambiguating elements).

Fig. D6 Impact of our annotations on Single Pose evaluation. We show here cases where the estimates by a state-of-the-art method (gdrnpp-pbrreal-rgbd-mmodelv1.3) produces fairly good **MSSD** errors when considering ground truth provided by BOP. For these cases, our more accurate ground truth yields worse **MSSD** errors, as it appears that the estimate belongs to the global symmetry pattern, but does not explain what is visible in the image.

facilitate visualization. We invite the reader to stop on some frames and check the differences in the estimates. Our ground truth distribution is displayed as the envelop for the rotation part and as red stars for the translation part.

Appendix G Metrics for evaluation: extended discussion

Rotation and translation errors are model-independent. [16], which led to BOP, states that *"fitness of object surface alignment is the main indicator of object pose quality,*

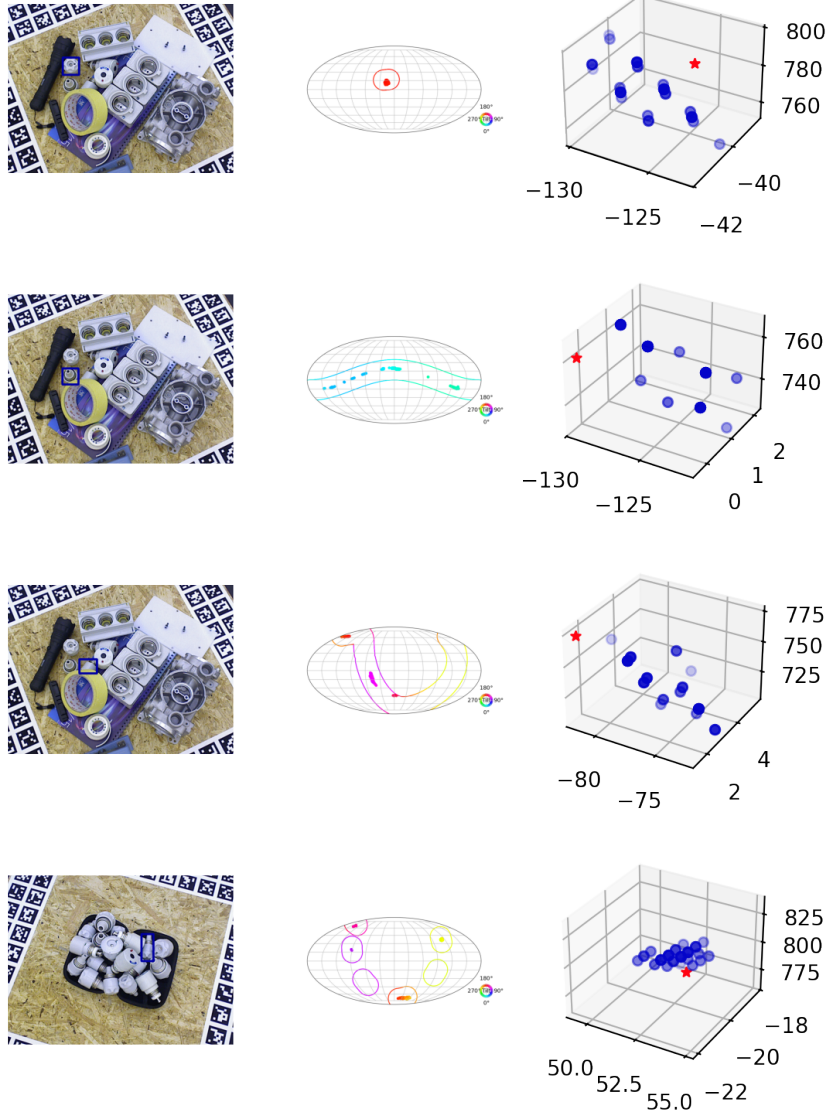


Fig. E7 Illustration of SpyroPose [3] results on T-LESS. We show the distribution estimates for the 3 instances of Object 1 (in the bounding box). The ground truth distribution is displayed as an envelop for the rotation part and as red star for the translation part.

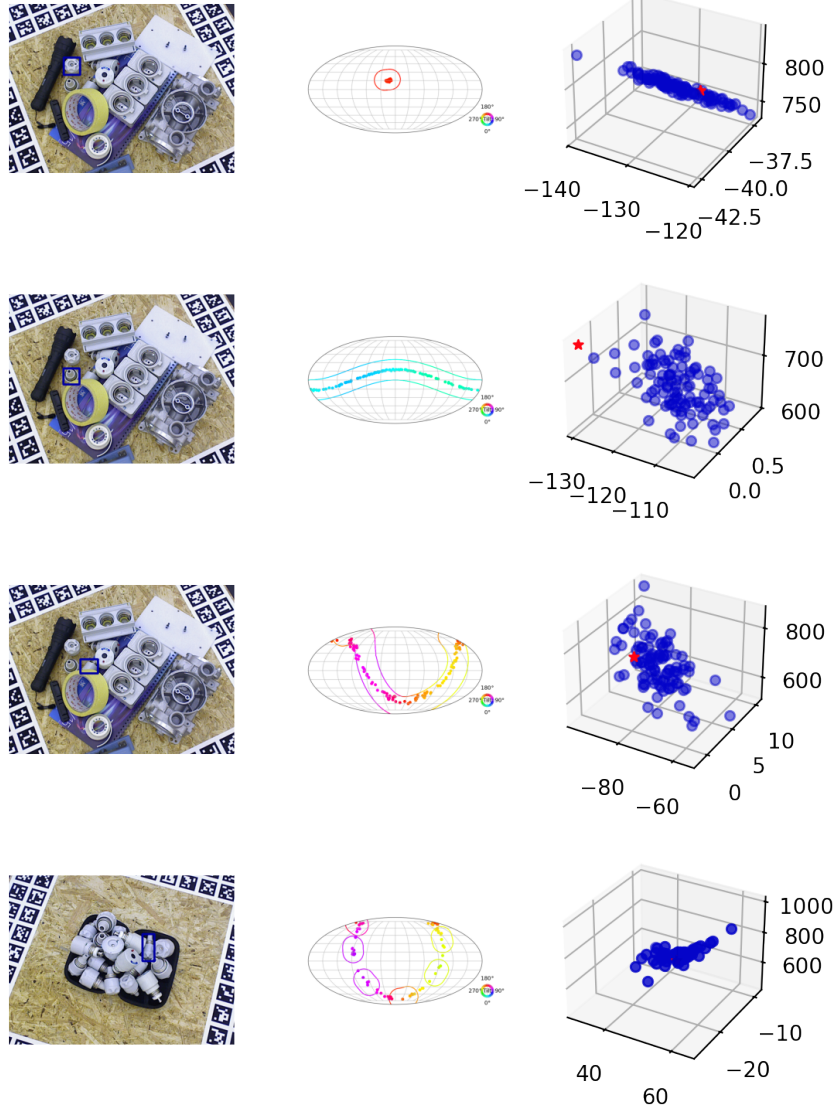
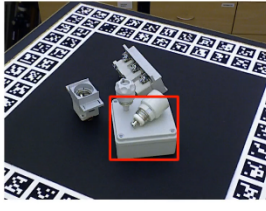
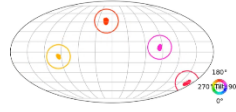


Fig. F8 Illustration of LiePose [4] results on T-LESS. We show the distribution estimates for the 3 instances of Object 1 (in the bounding box). The ground truth distribution is displayed as an envelop for the rotation part and as red star for the translation part.

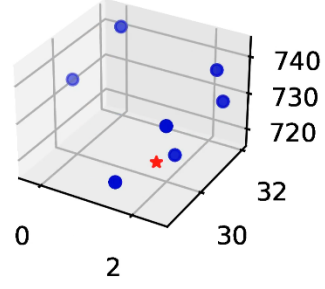
Object 27, Scene 5, Image 221



Input image with
target object
in bounding box

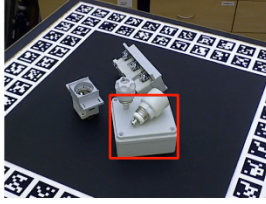


SpyroPose[7] distribution
rotation part

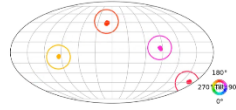


SpyroPose[7] distribution
translation part

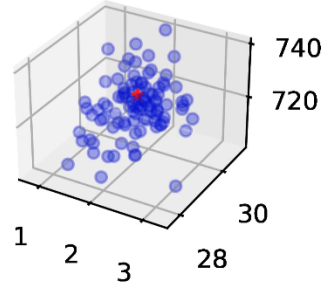
Object 27, Scene 5, Image 221



Input image with
target object
in bounding box



LiePose[17] distribution
rotation part



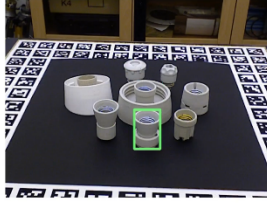
LiePose[17] distribution
translation part

Fig. F9 Visualizing SpyroPose [3] (top row) and LiePose [4] (bottom row) distribution results for object 27 (four rotation modes). Each example features an object of interest, in the bounding box in the left image, and the methods distribution estimation, split between the rotation part (center) and translation part (right). Both methods are able to retrieve the four rotation modes of the object. The envelop in the rotation part represents our BOP-Distrib annotation. We provide much more examples in the accompanying video.

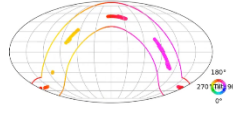
model-dependent pose error functions should be therefore preferred.” However, translation and rotation errors, as defined by [16]) with Equation G1 & Equation G2 can be exploited with our definitions of precision and recall over distributions (Equation 15 & Equation 16).

$$\mathbf{d}_{TE}(\hat{\mathbf{t}}, \bar{\mathbf{t}}) = \|\bar{\mathbf{t}} - \hat{\mathbf{t}}\|_2, \quad (\text{G1})$$

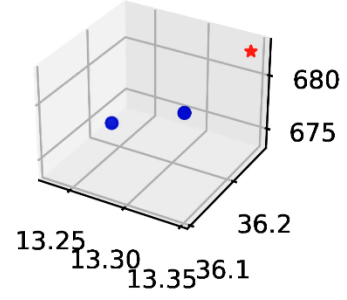
Object 15, Scene 7, Image 342



Input image with
target object
in bounding box

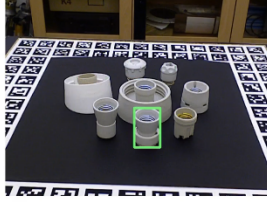


SpyroPose[7] distribution
rotation part

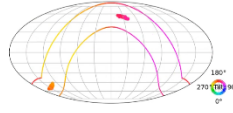


SpyroPose[7] distribution
translation part

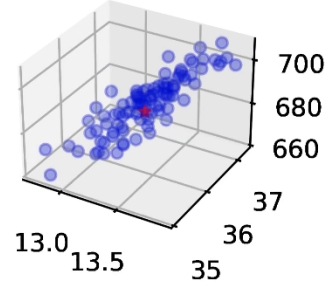
Object 15, Scene 7, Image 342



Input image with
target object
in bounding box



LiePose[17] distribution
rotation part



LiePose[17] distribution
translation part

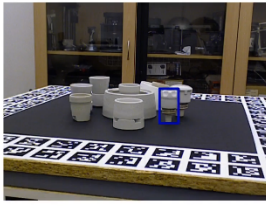
Fig. F10 Visualizing SpyroPose [3] (top row) and LiePose [4] (bottom row) distribution results for object 15 (continuous rotation). Each example features an object of interest, in the bounding box in the left image, and the methods distribution estimation, split between the rotation part (center) and translation part (right). Both methods fail to generate the target continuous rotation, although SpyroPose produces more correct rotations. The envelop in the rotation part represents our BOP-Distrib annotation. We provide much more examples in the accompanying video.

$$d_{RE}(\hat{\mathbf{R}}, \bar{\mathbf{R}}) = \arccos \left(\text{Tr}(\hat{\mathbf{R}}\bar{\mathbf{R}} - 1)/2 \right), \quad (\text{G2})$$

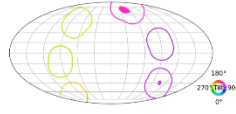
where $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ are respectively the ground truth rotation and translation, and where $\bar{\mathbf{R}}$ and $\bar{\mathbf{t}}$ are respectively the estimated rotation and translation.

Table G1 is a reprocessing of pose distribution estimation method results, with precision and recall over distribution, as defined by Equation 15 & Equation 16, with translation and rotation errors from Equation G1 & Equation G2.

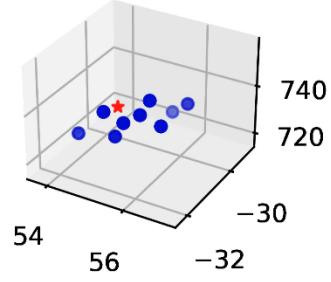
Object 1, Scene 7, Image 435



Input image with
target object
in bounding box



SpyroPose[7] distribution
rotation part

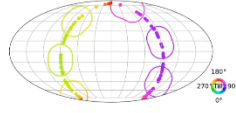


SpyroPose[7] distribution
translation part

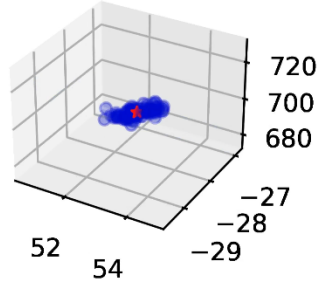
Object 1, Scene 7, Image 435



Input image with
target object
in bounding box



LiePose[17] distribution
rotation part



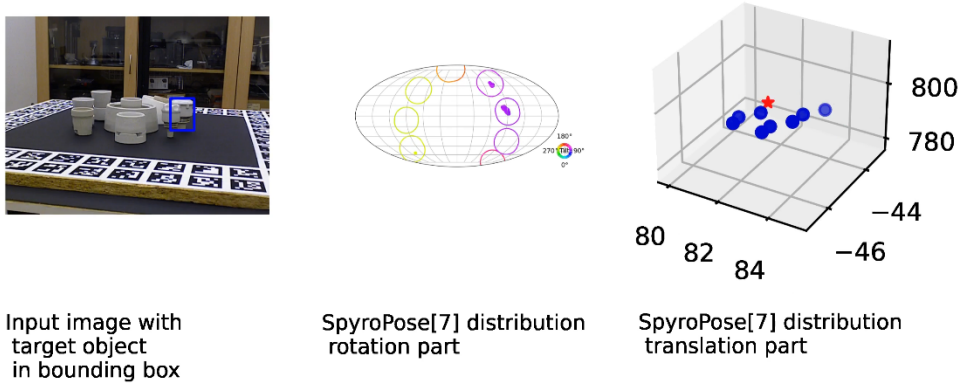
LiePose[17] distribution
translation part

Fig. F11 Visualizing SpyroPose [3] (top row) and LiePose [4] (bottom row) distribution results for object 1 (six rotation modes). Each example features an object of interest, in the bounding box in the left image, and the methods distribution estimation, split between the rotation part (center) and translation part (right). For this case of six rotations modes, SpyroPose is able to retrieve only two of them, whereas LiePose tends to a continuous distribution, thus generating false rotations. The envelop in the rotation part represents our BOP-Distrib annotation. We provide much more examples in the accompanying video.

Methods	P_{RE}	R_{RE}	P_{TE}	R_{TE}
SpyroPose [3]	73.2	69.1	43	66.9
LiePose [4]	68	91.1	46.2	92.9

Table G1 Precision/Recall over distribution for separate rotation and translation errors. We reprocess methods pose distribution estimates with decoupled rotation and translation errors.

Object 3, Scene 7, Image 435



Object 3, Scene 7, Image 435

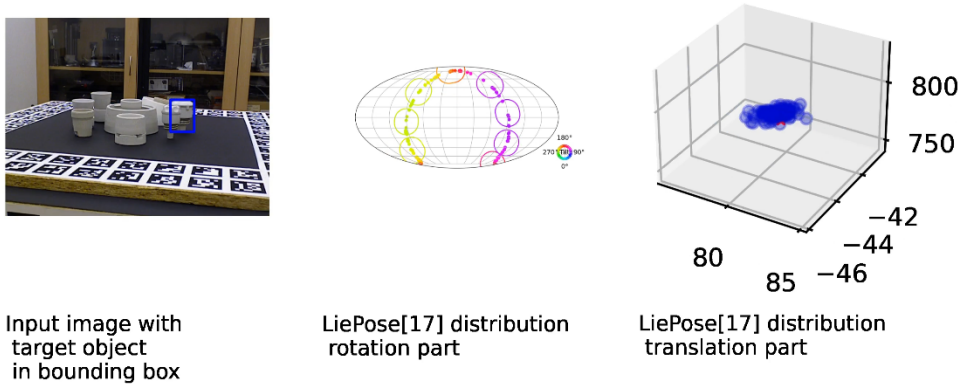


Fig. F12 Visualizing SpyroPose [3] (top row) and LiePose [4] (bottom row) distribution results for object 3 (eight rotation modes). Each example features an object of interest, in the bounding box in the left image, and the methods distribution estimation, split between the rotation part (center) and translation part (right). For this case of eight rotations modes, SpyroPose is able to retrieve only two of them, whereas LiePose tends to a continuous distribution, thus generating false rotations. The envelop in the rotation part represents our BOP-Distrib annotation. We provide much more examples in the accompanying video.

Appendix H Downstream Tasks Discussion

Our work implication on downstream task is two-fold. First, as highlighted by the BOP ranking changes in table 2, removing erroneous poses from the BOP ground truth implies a more reliable performance evaluation. Indeed, as illustrated by Fig. 11 of supp. mat., performance variations are related to poses that the current BOP ground-truth unduly classify as valid but are properly classified as invalid by our ground truth. This is due to overly lax ground truth annotation: some images are annotated as

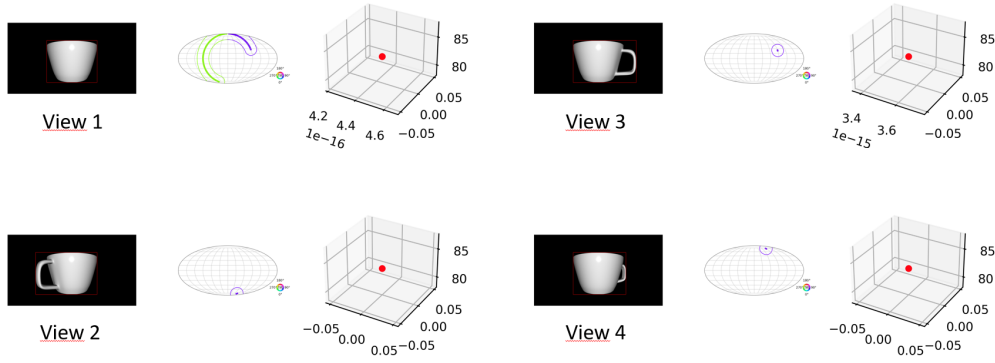


Fig. H13 Additional experiment for mug grasping task. We display multiple views of mug, associated to our pose distribution annotation. We are interested at pose estimation downstream tasks. We take the example of robotic grasping. We want the robot to grasp the mug by the handle. On view 1, the pose distribution is multi-modal, *i.e.* the object pose is ambiguous and the handle position in space cannot be accessed. For view 2, 3 and 4, the pose distribution is uni-modal, the image allows to estimate the pose of the mug without ambiguity. In such case, the downstream task of robotic handle grasping becomes feasible. Now, when it comes to evaluating pose distribution estimation methods against our ground truth, our recall metric evaluates the capacity of the estimation method to retrieve multi-modal distributions. This metric is a key indicator of pose distribution estimation methods for downstream tasks usability.

having multiple pose solutions due to the object symmetries whereas disambiguating elements breaking those symmetries are visible in the image. For downstream tasks, such as grasping or Augmented Reality, reliable ranking permits to choose the real best performer, and thus a higher success rate of the task.

The second implication on downstream tasks is related to Table 3, *i.e.* evaluating the ability of a method to determine the complete distribution of poses that explain the observed image. For applications such as grasping or Augmented Reality, if the object includes disambiguating elements, it implies that only one pose is valid for the task (*e.g.* a robot that should grab a mug at a specific position on the handle). However, if the observed image can be explained by multiple pose, it implies that disambiguating elements are not visible (*e.g.* the handle of the mug is not visible) and the downstream task is impossible to achieve from this unique viewpoint. A method that outputs the full distribution of poses provides the downstream task with the ability to determine if the task can be achieved (case of uni-modal distribution) or not (case of multi-modal solution). In such situation, a method that outputs a maximum of one pose does not permit to determine if the task can be achieved or not. Moreover, in such situation the complete distribution can help to determine the next best viewpoint to make the task feasible. Our distribution recall metric 16 evaluates the capacity of the estimation method to retrieve multi-modal distributions. This metric is a key indicator of pose distribution estimation methods for downstream tasks usability Figure H13 illustrates the robotic mug handle grabbing case.

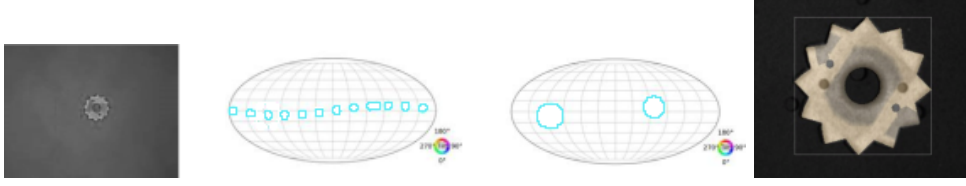


Fig. I14 Illustration of ITODD symmetries on the validation set (with public GT). For the star image (left, first row), BOP symmetries display 12 rotation modes (middle), whereas our annotation method keeps only 2 rotation modes (right), which align the two holes (size was set to one over the number of modes, hence the bigger modes on the right). We also show (second row left) a pose estimate of GPose [56], ranked first at BOP 2023, for the star object overlaid on its image. We observe that the holes are not correctly aligned. **MSPD** and **MSSD** metrics validate this estimate, whereas **MSPD** and **MSSD** metrics with our new annotations would have penalized it.

Appendix I Using Our Per-Image Annotations for Other Pose Estimation Datasets

Among BOP datasets, ITODD [31] and HomeBrewedDB [45] are the two other presenting object symmetries. They could easily be processed by our method, however their ground truth poses, needed as input to our method, are not public.

We give here an illustration of the interest to reprocess their symmetries patterns. We take ITODD’s small validation set, for which the ground truth is public. Figure I14 presents our result for the star object new ground truth, as well as one case of the current best performer for Single Pose estimation (gpose2023 [56]) failing to align the holes. In the current version of BOP evaluation, this pose estimation is validated. With our annotations, it would be penalized. New efforts at proposing challenging pose estimation datasets, such as [46], could be processed by our method. However, texture disambiguate a lot of the 3D models, and the scenes do not present much occlusion between objects.

Appendix J Discussion on Alignist [28]

Alignist [28] proposes to estimate rotation distribution for ambiguous object shapes from images. It was the first method that introduced a solution for supervising the training with a pseudo-ground truth generated rotation distribution. To do so, it resorts to a precomputation of such rotation distribution based on ground truth pose, rotation sampling, and SDF (Signed Distance Function) and Surfemb [23] features comparison. This precomputation is performed on renderings of single objects, and used to train a double MLP network to infer these distributions. The translation part of the pose is not considered. The test of the method on T-LESS [14] is conducted following Gilitschenski [8] protocol: it only processes single isolated objects on black background, and is evaluated with log likelihood.

In contrast, our annotation procedure does not rely on SurfEmb [23] comparisons which results are not guaranteed but it uses geometrical comparisons (see Equation 1). Moreover and unlike Alignist [28], our annotation procedure has a rejection mechanism for false visible points and false occluded points, as illustrated in Section ?? (see

Section 4.5). This point is crucial to be able to generate a proper ground truth annotation. Finally, our approach does not need to retrain SurfEmb [23] to annotate a new dataset.

Alignist [28] and other pose distributions estimation methods would benefit from our more accurate pose distributions for their trainings. Finally, Alignist [28] method would benefit from our evaluation framework (see Section 6.4). For that though, Alignist [28] would need to be tested on the full T-LESS [14] test set (and not just Gilitschenski [8] protocol that excludes external occlusions). We could not conduct such tests as the codes and results are not public.