



# CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation

Yann Labbé<sup>1,2(✉)</sup>, Justin Carpentier<sup>1,2</sup>, Mathieu Aubry<sup>3</sup>, and Josef Sivic<sup>1,2,4</sup>

<sup>1</sup> École normale supérieure, CNRS, PSL Research University, Paris, France

<sup>2</sup> INRIA, Paris, France

yann.labbe@inria.fr

<sup>3</sup> LIGM (UMR 8049), École des Ponts, UPE, Marne-la-Vallée, France

<sup>4</sup> Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical  
University in Prague, Prague, Czech Republic

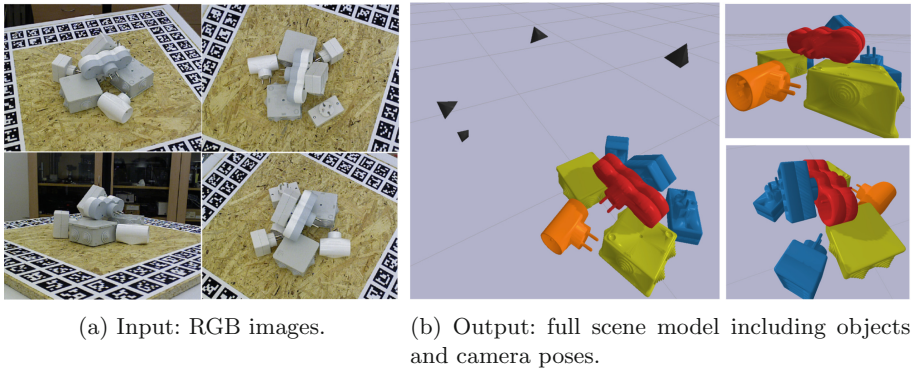
**Abstract.** We introduce an approach for recovering the 6D pose of multiple known objects in a scene captured by a set of input images with unknown camera viewpoints. First, we present a single-view single-object 6D pose estimation method, which we use to generate 6D object pose hypotheses. Second, we develop a robust method for matching individual 6D object pose hypotheses across different input images in order to jointly estimate camera viewpoints and 6D poses of all objects in a *single consistent scene*. Our approach explicitly handles object symmetries, does not require depth measurements, is robust to missing or incorrect object hypotheses, and automatically recovers the number of objects in the scene. Third, we develop a method for global scene refinement given multiple object hypotheses and their correspondences across views. This is achieved by solving an *object-level bundle adjustment* problem that refines the poses of cameras and objects to minimize the reprojection error in all views. We demonstrate that the proposed method, dubbed CosyPose, outperforms current state-of-the-art results for single-view and multi-view 6D object pose estimation by a large margin on two challenging benchmarks: the YCB-Video and T-LESS datasets. Code and pre-trained models are available on the project webpage. ([https://www.di.ens.fr/willow/research/cosypose/.](https://www.di.ens.fr/willow/research/cosypose/))

## 1 Introduction

The goal of this work is to estimate accurate 6D poses of multiple known objects in a 3D scene captured by multiple cameras with unknown positions, as illustrated in Fig. 1. This is a challenging problem because of the texture-less nature of many objects, the presence of multiple similar objects, the unknown number and type of objects in the scene, and the unknown positions of cameras. Solving

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-58520-4\\_34](https://doi.org/10.1007/978-3-030-58520-4_34)) contains supplementary material, which is available to authorized users.



**Fig. 1. CosyPose: 6D object pose estimation optimizing multi-view CONsistency.** Given (a) a set of RGB images depicting a scene with known objects taken from unknown viewpoints, our method accurately reconstructs the scene, (b) recovering all objects in the scene, their 6D pose and the camera viewpoints. Objects are enlarged for the purpose of visualization.

this problem would have, however, important applications in robotics where the knowledge of accurate position and orientation of objects within the scene would allow the robot to plan, navigate and interact with the environment.

Object pose estimation is one of the oldest computer vision problems [1–3], yet it remains an active area of research [4–11]. The best performing methods that operate on RGB (no depth) images [7, 8, 10–12] are based on trainable convolutional neural networks and are able to deal with symmetric or textureless objects, which were challenging for earlier methods relying on local [3, 13–16] or global [17] gradient-based image features. However, most of these works consider objects independently and estimate their poses using a single input (RGB) image. Yet, in practice, scenes are composed of many objects and multiple images of the scene are often available, e.g. obtained by a single moving camera, or in a multi-camera set-up. In this work, we address these limitations and develop an approach that combines information from *multiple views* and estimates jointly the pose of *multiple objects* to obtain a single consistent scene interpretation.

While the idea of jointly estimating poses of multiple objects from multiple views may seem simple, the following challenges need to be addressed. First, object pose hypotheses made in individual images cannot easily be expressed in a common reference frame when the relative transformations between the cameras are unknown. This is often the case in practical scenarios where camera calibration cannot easily be recovered using local feature registration because the scene lacks texture or the baselines are large. Second, the single-view 6D object pose hypotheses have gross errors in the form of false positive and missed detections. Third, the candidate 6D object poses estimated from input images are noisy as they suffer from depth ambiguities inherent to single view methods.

In this work, we describe an approach that addresses these challenges. We start from 6D object pose hypotheses that we estimate from each view using a new render-and-compare approach inspired by DeepIM [10]. First, we match individual object pose hypotheses across different views and use the resulting *object-level* correspondences to recover the relative positions between the cameras. Second, gross errors in object detection are addressed using a robust object-level matching procedure based on RANSAC, optimizing the overall scene consistency. Third, noisy single-view object poses are significantly improved using a global *refinement procedure* based on object-level bundle adjustment. The outcome of our approach that optimizes multi-view COnSistencY, hence dubbed CosyPose, is a single consistent reconstruction of the input scene. Our single-view single-object pose estimation method obtains state-of-the-art results on the YCB-Video [18] and T-LESS [19] datasets, achieving a significant 33.8% absolute improvement over the state-of-the-art [7] on T-LESS. Our multi-view framework clearly outperforms [20] on YCB-Video while not requiring known camera poses and not being limited to a single object of each class per scene. On both datasets, we show that our multi-view solution significantly improves pose estimation and 6D detection accuracy over our single-view baseline.

## 2 Related Work

Our work builds on results in single-view and multi-view object 6D pose estimation from RGB images and object-level SLAM.

*Single-view Single-Object 6D Pose Estimation.* The object pose estimation problem [15, 16] has been approached either by estimating the pose from 2D-3D correspondences using local invariant features [3, 13], or directly by estimating the object pose using template-matching [14]. However, local features do not work well for texture-less objects and global templates often fail to detect partially occluded objects. Both of these approaches (feature-based and template matching) have been revisited using deep neural networks. A convolutional neural network (CNN) can be used to detect object features in 2D [4, 6, 18, 21, 22] or to directly find 2D-to-3D correspondences [5, 7, 8, 23]. Deep approaches have also been used to match implicit pose features, which can be learned without requiring ground truth pose annotations [12]. The estimated 6D pose of the objects can be further refined [4, 10] using an iterative procedure that effectively moves the camera around the object so that the rendered image of the object best matches the input image. Such a refinement step provides important performance improvements and is becoming common practice [8, 11] as a final stage of the estimation process. Our single-view single-object pose estimation described in Sect. 3.2 builds on DeepIM [10]. The performance of 6D pose estimation can be further improved using depth sensors [10, 11, 18], but in this work we focus on the most challenging scenario where only RGB images are available.

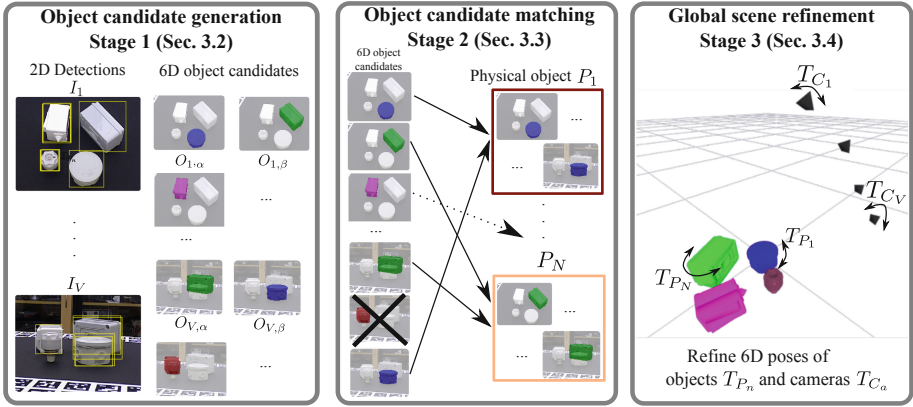
*Multi-view Single-Object 6D Pose Estimation.* Multiple views of an object can be used to resolve depth ambiguities and gain robustness with respect to occlusions. Prior work using local invariant features includes [15, 16, 24, 25] and involves some form of feature matching to establish correspondences across views to aggregate information from multiple viewpoints. More recently, the multi-view single-object pose estimation problem has been revisited with a deep neural network that predicts an object pose candidate in each view [20] and aggregates information from multiple views assuming known camera poses. In contrast, our work does not assume the camera poses to be known. We experimentally demonstrate that our approach outperforms [20] despite requiring less information.

*Multi-view Multi-object 6D Pose Estimation.* Other works consider all objects in a scene together in order to jointly estimate the state of the scene in the form of a compact representation of the object and camera poses in a common coordinate system. This problem is known as object-level SLAM [26] where a depth-based object pose estimation method [27] is used to recognize objects from a database in individual images and estimate their poses. The individual objects are tracked across frames using depth measurements, assuming the motion of the sensor is continuous. Consecutive depth measurements also enable to produce hypotheses for camera poses using ICP [28] and the poses of objects and cameras are finally refined in a joint optimization procedure. Another approach [29] uses local RGBD patches to generate object hypotheses and find the best view of a scene. All of these methods, however, strongly rely on depth sensors to estimate the 3D structure of the scene while our method only exploits RGB images. In addition, they assume temporal continuity between the views, which is also not required by our approach.

Other works have considered monocular RGB only object-level SLAM [30–32]. Related is also [33] where semantic 2D keypoint correspondences across multiple views and local features are used to jointly estimate the pose of a single human and the positions of the observing cameras. All of these works rely on local images features to estimate camera poses. In contrast, our work exploits 6D pose hypotheses generated by a neural network which allows to recover camera poses in situations where feature-based registration fails, as is the case for example for the complex texture-less images of the T-LESS dataset. In addition, [31, 32] do not consider full 6D pose of objects, and [20, 33] only consider scenes with a single instance of each object. In contrast, our method is able to handle scenes with multiple instances of the same object.

### 3 Multi-view Multi-object 6D Object Pose Estimation

In this section, we present our framework for multi-view multi-object pose estimation. We begin with an overview of the approach (Sect. 3.1 and Fig. 2), and then detail the three main steps of the approach in the remaining sections.



**Fig. 2. Multi-view multi-object 6D pose estimation.** In the first stage, we obtain initial object candidates in each view separately. In the second stage, we match these object candidates across views to recover a single consistent scene. In the third stage, we globally refine all object and camera poses to minimize multi-view reprojection error.

### 3.1 Approach Overview

Our goal is to reconstruct a scene composed of multiple objects given a set of RGB images. We assume that we know the 3D models of objects of interest. However, there can be multiple objects of the same type in the scene and no information on the number or type of objects in the scene is available. Furthermore, objects may not be visible in some views, and the relative poses between the cameras are unknown. Our output is a scene model, which includes the number of objects of each type, their 6D poses and the relative poses of the cameras. Our approach is composed of three main stages, summarized in Fig. 2.

In the first stage, we build on the success of recent methods for single-view RGB object detection and 6D pose estimation. Given a set of objects with known 3D models and a single image of a scene, we output a set of candidate detections for each object and for each detection the 6D pose of the object with respect to the camera associated to the image. Note that some of these detections and poses are wrong, and some are missing. We thus consider the poses obtained in this stage as a set of initial *object candidates*, i.e. objects that may be seen in the given view together with an estimate of their pose with respect to this view. This *object candidate generation* process is described in Sect. 3.2.

In the second stage, called *object candidate matching* and described in detail in Sect. 3.3, we match objects visible in multiple views to obtain a single consistent scene. This is a difficult problem since object candidates from the first stage typically include many errors due to (i) heavily occluded objects that might be mis-identified or for which the pose estimate might be completely wrong; (ii) confusion between similar objects; and (iii) unusual poses that do not appear in the training set and are not detected correctly. To tackle these challenges,

we take inspiration from robust patch matching strategies that have been used in the structure from motion (SfM) literature [34, 35]. In particular, we design a matching strategy similar in spirit to [36] but where we match entire 3D objects across views to obtain a single consistent 3D scene, rather than matching local 2D patches on a single 3D object [36].

The final stage of our approach, described in Sect. 3.4, is a global *scene refinement*. We draw inspiration from bundle adjustment [37], but the optimization is performed at the level of objects: the 6D poses of all objects and cameras are refined to minimize a global reprojection error.

### 3.2 Stage 1: Object Candidate Generation

Our system takes as input multiple photographs of a scene  $\{I_a\}$  and a set of 3D models, each associated to an object label  $l$ . We assume the intrinsic parameters of camera  $C_a$  associated to image  $I_a$  are known as is usually the case in single-view pose estimation methods. In each view  $I_a$ , we obtain a set of object detections using an object detector (e.g. FasterRCNN [38], RetinaNet [39]), and a set of candidate pose estimates using a single-view single-object pose estimator (e.g. PoseCNN [18], DPOD [8], DeepIM [10]). While our approach is agnostic to the particular method used, we develop our own single-view single-object pose estimator, inspired by DeepIM [10], which improves significantly over state of the art and which we describe in the next paragraph. Each 2D candidate detection in view  $I_a$  is identified by an index  $\alpha$  and corresponds to an *object candidate*  $O_{a,\alpha}$ , associated with a predicted object label  $l_{a,\alpha}$  and a 6D pose estimate  $T_{C_a O_{a,\alpha}}$  with respect to camera  $C_a$ . We model a 6D pose  $T \in \text{SE}(3)$  as a  $4 \times 4$  homogeneous matrix composed of a 3D rotation matrix and a 3D translation vector.

*Single-View 6D Pose Estimation.* We introduce a method for single-view 6D object pose estimation building on the idea of DeepIM [10] with some simplifications and technical improvements. First, we use a more recent neural-network architecture based on EfficientNet-B3 [40] and do not include auxiliary signals while training. Second, we exploit the rotation parametrization recently introduced in [41], which has been shown to lead to more stable CNN training than quaternions. Third, we disentangle depth and translation prediction in the loss following [42] and handle symmetries explicitly as in [9] instead of using the point-matching loss. Fourth, instead of fixing focal lengths to 1 during training as in [10], we use focal lengths of the camera equivalent to the cropped images. Fifth, in addition to the real training images supplied with both dataset, we also render a million images for each dataset using the provided CAD models for T-LESS and the reconstructed models for YCB-Video. The CNNs are first pretrained using synthetic data only, then fine-tuned on both real and synthetic images. Finally, we use data augmentation on the RGB images while training our models, which has been demonstrated to be crucial to obtain good performance on T-LESS [12]. We also note that this approach can be used for coarse estimation simply by providing a canonical pose as the input pose estimate during both

training and testing. We rendered objects at a distance of 1 m from the camera and used this approach to perform coarse estimate on T-LESS. Additional details are provided in the supplementary material.

*Object Symmetries.* Handling object symmetries is a major challenge for object pose estimation since the object pose can only be estimated up to a symmetry. This is in particular true for our object candidates pose estimates. We thus need to consider symmetries explicitly together with the pose estimates. Each 3D model  $l$  is associated to a set of symmetries  $S(l)$ . Following the framework introduced in [43], we define the set of symmetries  $S(l)$  as the set of transformations  $S$  that leave the appearance of object  $l$  unchanged:

$$S(l) = \{S \in \text{SE}(3) \text{ s.t. } \forall T \in \text{SE}(3), \mathcal{R}(l, T) = \mathcal{R}(l, TS)\}, \quad (1)$$

where  $\mathcal{R}(l, X)$  is the rendered image of object  $l$  captured in pose  $X$  and  $S$  is the rigid motion associated to the symmetry. Note that  $S(l)$  is infinite for objects that have axes of symmetry (e.g. bowls).

Given a set of symmetries  $S(l)$  for the 3D object  $l$ , we define the symmetric distance  $D_l$  which measures the distance between two 6D poses represented by transformations  $T_1$  and  $T_2$ . Given an object  $l$  associated to a set  $\mathcal{X}_l$  of  $|\mathcal{X}_l|$  3D points  $\mathbf{x} \in \mathcal{X}_l$ , we define:

$$D_l(T_1, T_2) = \min_{S \in S(l)} \frac{1}{|\mathcal{X}_l|} \sum_{\mathbf{x} \in \mathcal{X}_l} \|T_1 S \mathbf{x} - T_2 \mathbf{x}\|_2. \quad (2)$$

$D_l(T_1, T_2)$  measures the average error between the points transformed with  $T_1$  and  $T_2$  for the symmetry  $S$  that best aligns the (transformed) points. In practice, to compute this distance for objects with axes of symmetries, we discretize  $S(l)$  using 64 rotation angles around each symmetry axis, similar to [9].

### 3.3 Stage 2: Object Candidate Matching

As illustrated in Fig. 2, given the object candidates for all views  $\{O_{a,\alpha}\}$ , our matching module aims at (i) removing the object candidates that are not consistent across views and (ii) matching object candidates that correspond to the same physical object. We solve this problem in two steps detailed below: (A) selection of candidate pairs of objects in all pairs of views, and (B) scene-level matching.

*A. 2-View Candidate Pair Selection.* We first focus on a single pair of views  $(I_a, I_b)$  of the scene and find all pairs of object candidates  $(O_{a,\alpha}, O_{b,\beta})$ , one in each view, which correspond to the same physical object in these two views. To do so, we use a RANSAC procedure where we hypothesize a relative pose between the two cameras and count the number of inliers, i.e. the number of consistent pairs of object candidates in the two views. We then select the solution with the most inliers which gives associations between the object candidates in the two views. In the rest of the section, we describe in more detail how we sample relative camera poses and how we define inlier candidate pairs.



*Sampling of Relative Camera Poses.* Sampling meaningful camera poses is one of the main challenges for our approach. Indeed, directly sampling at random the space of possible camera poses would be inefficient. Instead, as usual in RANSAC, we sample pairs of object candidates (associated to the same object label) in the two views, hypothesize that they correspond to the same physical object and use them to infer a relative camera pose hypothesis. However, since objects can have symmetries, a single pair of candidates is not enough to obtain a relative pose hypothesis without ambiguities and we thus sample two pairs of object candidates, which in most cases is sufficient to disambiguate symmetries.

In detail, we sample two tentative object candidate pairs with pair-wise consistent labels  $(O_{a,\alpha}, O_{b,\beta})$  and  $(O_{a,\gamma}, O_{b,\delta})$  and use them to build a relative camera pose hypothesis,  $T_{C_a C_b}$ . We obtain the relative camera pose hypothesis by (i) assuming that  $(O_{a,\alpha}, O_{b,\beta})$  correspond to the same physical object and (ii) disambiguating symmetries by assuming that  $(O_{a,\gamma}, O_{b,\delta})$  also correspond to the same physical object, and thus selecting the symmetry that minimize their symmetric distance

$$T_{C_a C_b} = T_{C_a O_{a,\alpha}} S^* T_{C_b O_{b,\beta}}^{-1} \quad (3)$$

$$\text{with } S^* = \underset{S \in S(l)}{\operatorname{argmin}} D_l(T_{C_a O_{a,\gamma}}, (T_{C_a O_{a,\alpha}} S T_{C_b O_{b,\beta}}^{-1}) T_{C_b O_{b,\delta}}), \quad (4)$$

where  $l = l_{a,\alpha} = l_{b,\beta}$  is the object label associated to the first pair, and  $S^*$  is the object symmetry which best aligns the point clouds associated to the second pair of objects  $(O_{a,\gamma}$  and  $O_{b,\delta})$ . If the union of the two physical objects is symmetric, e.g. two spheres, the pose computed may be incorrect but it would not be verified by a third pair of objects, and the hypothesis would be discarded.

*Counting Pairs of Inlier Candidates.* Let's assume we are given a relative pose hypothesis between the cameras  $T_{C_a C_b}$ . For each object candidate  $O_{a,\alpha}$  in the first view, we find the object candidate in the second view  $O_{b,\beta}$  with the same label  $l = l_{a,\alpha} = l_{b,\beta}$  that minimizes the symmetric distance  $D_l(T_{C_a O_{a,\alpha}}, T_{C_a C_b} T_{C_b O_{b,\beta}})$ . In other words,  $O_{b,\beta}$  is the object candidate in the second view closest to  $O_{a,\alpha}$  under the hypothesized relative pose between the cameras. This pair  $(O_{a,\alpha}, O_{b,\beta})$  is considered an inlier if the associated symmetric distance is smaller than a given threshold  $C$ . The total number of inliers is used to score the relative camera pose  $T_{C_a C_b}$ . Note that we discard the hypothesis which have fewer than three inliers.

*B. Scene-Level Matching.* We use the result of the 2-view candidate pair selection applied to each image pair to define a graph between all candidate objects. Each vertex corresponds to an object candidate in one view and edges correspond to pairs selected from 2-view candidate pair selection, i.e. pairs that had sufficient inlier support. We first remove isolated vertices, which correspond to object candidates that have not been validated by other views. Then, we associate to each connected component in the graph a unique physical object, which corresponds to a set of initial object candidates originating from different views. We call



these physical objects  $P_1, \dots, P_N$  with  $N$  the total number of physical objects, i.e. the number of connected components in the graph. We write  $(a, \alpha) \in P_n$  to denote the fact that an object candidate  $O_{a,\alpha}$  is in the connected component of object  $P_n$ . Since all the objects in a connected component share the same object label (they could not have been connected otherwise), we can associate without ambiguity an object label  $l_n$  to each physical object  $P_n$ .

### 3.4 Stage 3: Scene Refinement

After the previous stage, the correspondences between object candidates in the individual images are known, and the non-coherent object candidates have been removed. The final stage aims at recovering a unique and consistent scene model by performing global joint refinement of objects and camera poses.

In detail, the goal of this stage is to estimate poses of physical objects  $P_n$ , represented by transformations  $T_{P_1}, \dots, T_{P_N}$ , and cameras  $C_v$ , represented by transformations  $T_{C_1}, \dots, T_{C_V}$ , in a common world coordinate frame. This is similar to the standard bundle adjustment problem where the goal is to recover the 3D points of a scene together with the camera poses. This is typically addressed by minimizing a reconstruction loss that measures the 2D discrepancies between the projection of the 3D points and their measurements in the cameras. In our case, instead of working at the level of points as done in the bundle adjustment setting, we introduce a reconstruction loss that operates at the level of objects.

More formally, for each object present in the scene, we introduce an object-candidate reprojection loss accounting for symmetries. We define the loss for a candidate object  $O_{a,\alpha}$  associated to a physical object  $P_n$  (i.e.  $(a, \alpha) \in P_n$ ) and the estimated candidate object pose  $T_{C_a O_{a,\alpha}}$  with respect to  $C_a$  as:

$$L(T_{P_n}, T_{C_a} | T_{C_a O_{a,\alpha}}) = \min_{S \in S(l)} \frac{1}{|\mathcal{X}_l|} \sum_{x \in \mathcal{X}_l} \|\pi_a(T_{C_a O_{a,\alpha}} Sx) - \pi_a(T_{C_a}^{-1} T_{P_n} x)\|, \quad (5)$$

where  $\|\cdot\|$  is a truncated L2 loss,  $l = l_n$  is the label of the physical object  $P_n$ ,  $T_{P_n}$  the 6D pose of object  $P_n$  in the world coordinate frame,  $T_{C_a}$  the pose of camera  $C_a$  in the world coordinate frame,  $\mathcal{X}_l$  the set of 3D points associated to the 3D model of object  $l$ ,  $S(l)$  the symmetries of the object model  $l$ , and the operator  $\pi_a$  corresponds to the 2D projection of 3D points expressed in the camera frame  $C_a$  by the intrinsic calibration matrix of camera  $C_a$ . The inner sum in Eq. (5) is the error between (i) the 3D points  $x$  of the object model  $l$  projected to the image with the single view estimate of the transformation  $T_{C_a O_{a,\alpha}}$  that is associated with the physical object (i.e.  $(a, \alpha) \in P_n$ ) (first term, the image measurement) and (ii) the 3D points  $T_{P_n} x$  on the object  $P_n$  projected to the image by the global estimate of camera  $C_a$  (second term, global estimates).

Recovering the state of the unique scene which best explains the measurements consists in solving the following consensus optimization problem:

$$\min_{T_{P_1}, \dots, T_{P_N}, T_{C_1}, \dots, T_{C_V}} \sum_{n=1}^N \sum_{(a,\alpha) \in P_n} L(T_{P_n}, T_{C_a} | T_{C_a O_{a,\alpha}}), \quad (6)$$

**Table 1. Single-view 6D pose estimation.** Comparisons with state-of-the-art methods on the YCB-Video (a) and T-LESS datasets (b).

	AUC of ADD-S	AUC of ADD(-S)		$e_{\text{vsd}} < 0.3$
PoseCNN [18]	–	61.3	Implicit [12]	26.8
MCN [21]	75.1	-	Pix2pose [7]	29.5
PVNet [5]	–	73.4	Ours	<b>63.3</b>
DeepIM [10]	88.1	81.9	w/o loss	59.5
Ours	<b>89.8</b>	<b>84.5</b>	w/o network	58.9
			w/o rot.	60.5
			w/o data augm.	35.5

(a) YCB-Video

(b) T-LESS SiSo task

where the first sum is over all the physical objects  $P_n$  and the second one over all object candidates  $O_{a,\alpha}$  corresponding to the physical object  $P_n$ . In other words, we wish to find global estimates of object poses  $T_{P_n}$  and camera poses  $T_{C_a}$  to match the (inlier) object candidate poses  $T_{C_a O_{a,\alpha}}$  obtained in the individual views. The optimization problem is solved using the Levenberg-Marquart algorithm. We provide more details in the supplementary.

## 4 Results

In this section, we experimentally evaluate our method on the YCB-Video [18] and T-LESS [19] datasets, which both provide multiple views and ground truth 6D object poses for cluttered scenes with multiple objects. In Sect. 4.1, we first validate and analyze our single-view single-object 6D pose estimator. We notably show that our single-view single-object 6D pose estimation method already improves state-of-the-art results on both datasets. In Sect. 4.2, we validate our multi-view multi-object framework by demonstrating consistent improvements over the single-view baseline.

### 4.1 Single-View Single-Object Experiments

*Evaluation on YCB-Video.* Following [5, 10, 18], we evaluate on a subset of 2949 keyframes from videos of the 12 testing scenes. We use the standard ADD-S and ADD(-S) metrics and their area-under-the-curves [18] (please see supplementary material for details on the metrics). We evaluate our refinement method using the same detections and coarse estimates as DeepIM [10], provided by PoseCNN [18]. We ran two iterations of pose refinement network. Results are shown in Table 1a. Our method improves over the current-state-of-the-art DeepIM [10], by approximately 2 points on the AUC of ADD-S and ADD(-S) metrics.

*Evaluation on T-LESS.* As explained in Sect. 3.2, we use our single-view approach both for coarse pose estimation and refinement. We compare our method against the two recent RGB-only methods Pix2Pose [7] and Implicit [12]. For a fair comparison, we use the detections from the same RetinaNet model as in [7]. We report results on the SiSo task [44] and use the standard visual surface discrepancy (vsd) recall metric with the same parameters as in [7, 12]. Results are presented in Table 1b. On the  $e_{\text{vsd}} < 0.3$  metric, our {coarse + refinement} solution achieves a significant 33.8% absolute improvement compared to existing state-of-the-art methods. Note that [10] did not report results on T-LESS. We also evaluate on this dataset the benefits of the key components of our single view approach compared to the components used in DeepIM [10]. More precisely, we evaluate the importance of the base network (our EfficientNet vs FlowNet pre-trained), loss (our symmetric and disentangled vs. point-matching loss with  $L_1$  norm), rotation parametrization (our using [41] vs. quaternions) and data augmentation (our color augmentation, similar to [12] vs. none). Loss, network and rotation parametrization bring a small but clear improvement. Using data augmentation is crucial on the T-LESS dataset where training is performed only on synthetic data and real images of the objects on dark background.

## 4.2 Multi-view Experiments

As shown above, our single-view method achieves state-of-the-art results on both datasets. We now evaluate the performance of our multi-view approach to estimate 6D poses in scenes with multiple objects and multiples views.

*Implementation Details.* On both datasets, we use the same hyper-parameters. In stage 1, we only consider object detections with a score superior to 0.3 to limit the number of detections. In stage 2, we use a RANSAC 3D inlier threshold of  $C = 2$  cm. This low threshold ensures that no outliers are considered while associating object candidates. We use a maximum number of 2000 RANSAC iterations for each pair of views, but this limit is only reached for the most complex scenes of the T-LESS dataset containing tens of detections. For instance, in the context of two views with six different 6D object candidates in each view, only 15 RANSAC iterations are enough to explore all relative camera pose hypotheses. For the scene refinement (stage 3), we use 100 iterations of Levenberg-Marquart (the optimization typically converges in less than 10 iterations).

*Evaluation Details.* In the single-view evaluation, the poses of the objects are expressed with respect to the camera frame. To fairly compare with the single-view baseline, we also evaluate the object poses in the camera frames, that we compute using the absolute object poses and camera placements estimated by our global scene refinement method. Standard metrics for 6D pose estimation strongly penalize methods with low detection recall. To avoid being penalized for removing objects that cannot be verified across several views, we thus add the initial object candidates to the set of predictions but with confidence scores strictly lower than the predictions from our full scene reconstruction.

**Table 2. Multi-view multi-object results.** (a) Our approach significantly outperforms [20] on the YCB-Video dataset in both the single view and multi-view scenarios while not requiring known camera poses. (b) Results on the T-LESS dataset. Using multiple views clearly improves our results.

	1 view	5 views		1 view	4 views	8 views
			AUC of ADD-S	72.1	76.0	<b>78.9</b>
[21]	75.1	80.2	ADD-S < 0.1d	62.7	66.6	<b>70.9</b>
Ours	89.8	<b>93.4</b>	$e_{\text{vsd}} < 0.3$	57.7	61.8	<b>65.6</b>
(a) YCB-Video (AUC of ADD-S)			mAP@ADD-S<0.1d	55.0	61.6	<b>69.0</b>
			(b) T-LESS ViVo task (ours, 1000 images)			

**Table 3.** Benefits of the scene refinement stage. We report pose ADD-S errors (in mm) for the inlier object candidates before and after global scene refinement. Scene-refinement improves 6D pose estimation accuracy.

	YCB dataset	T-LESS dataset
Before refinement	6.40	4.43
After refinement	<b>5.05</b>	<b>3.19</b>

*Multi-view Multi-object Quantitative Results.* The problem that we consider, recovering the 6D object poses of multiple known objects in a scene captured by several RGB images taken from unknown viewpoints has not, to the best of our knowledge, been addressed by prior work reporting results on the YCB-Video and T-LESS datasets. The closest work is [20], which considers multi-view scenarios on YCB-Video and uses ground truth camera poses to align the viewpoints. In [20], results are provided for prediction using 5 views. We use our approach with the same number of input images but without using ground truth calibration and report results in Table 2a. Our method significantly outperforms [20] in both single-view and multi-view scenarios.

We also perform multi-view experiments on T-LESS with a variable number of views. We follow the multi-instance BOP [44] protocol for ADD-S<0.1d and  $e_{\text{vsd}} < 0.3$ . We also analyze precision-recall tradeoff similar to the standard practice in object detection. We consider positive predictions that satisfy ADD-S<0.1d and report mAP@ADD-S<0.1d. Results are shown in Table 2b for the ViVo task on 1000 images. To the best of our knowledge, no other method has reported results on this task. As expected, our multi-view approach brings significant improvements compared to only single-view baseline.

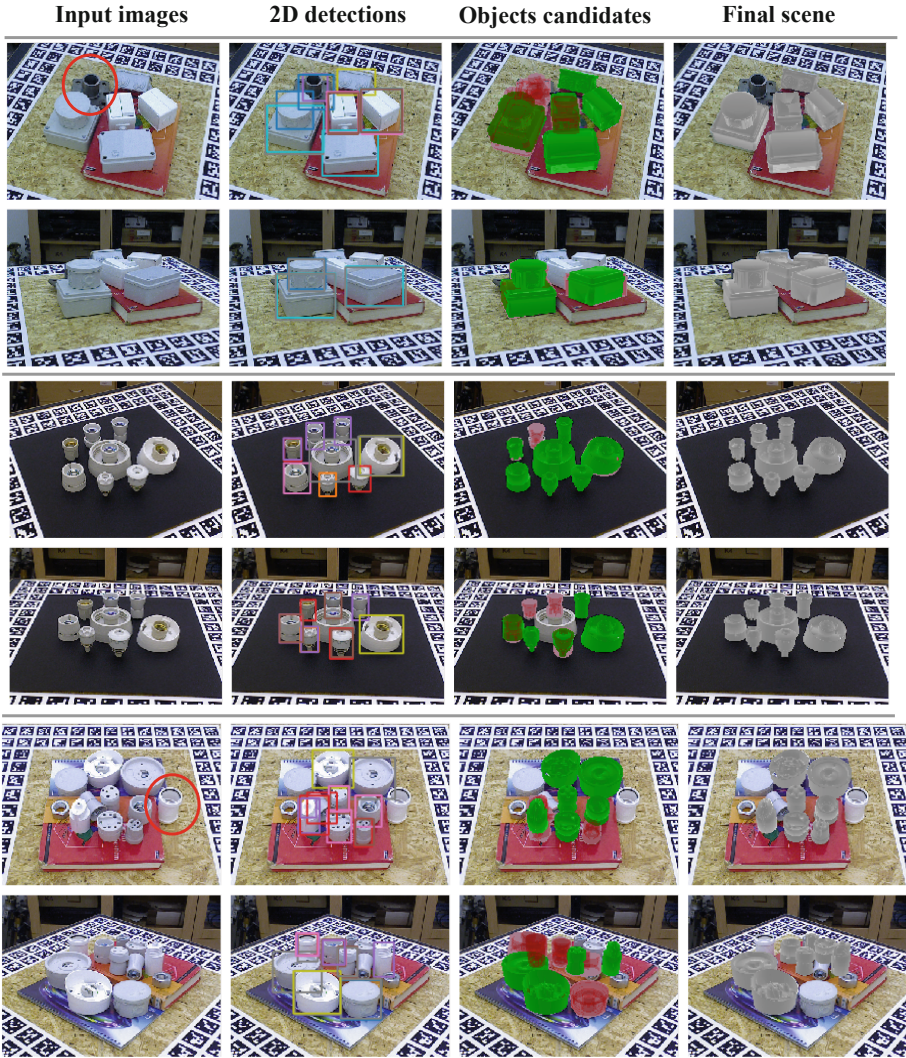
*Benefits of Scene Refinement.* To demonstrate the benefits of global scene refinement (stage 3), we report in Table 3 the average ADD-S errors of the inlier candidates before and after solving the optimization problem of Eq.(6). We note a clear relative improvement, around 20% on both datasets..

*Relative Camera Pose Estimation.* A key feature of our method is that it does not require camera position to be known and instead robustly estimates it from the 6D object candidates. We investigated alternatives to our joint camera pose estimation. First, we used COLMAP [45,46], a popular feature-based SfM software, to recover camera poses. On randomly sampled groups of 5 views from the YCB-Video dataset COLMAP outputs camera poses in only 67% of cases compared to 95% for our method. On groups of 8 views from the more difficult T-LESS dataset, COLMAP outputs camera poses only in 4% of cases, compared to 74% for our method. Our method therefore demonstrates a significant interest compared to COLMAP that uses features to recover camera poses, especially for complex textureless scenes like in the T-LESS dataset. Second, instead of estimating camera poses using our approach, we investigated using ground truth camera poses available for the two datasets. We found that the improvements using ground truth camera poses over the camera poses recovered automatically by our method were only minor: within 1% for T-LESS (4 views) and YCB-Video (5 views), and within 3% for T-LESS (8 views). This demonstrates that our approach recovers accurate camera poses even for scenes containing only symmetric objects as in the T-LESS dataset.

*Qualitative Results.* We provide examples of recovered 6D object poses in Fig. 3 where we show both object candidates and the final estimated scenes. **Please see the supplementary material for additional results**, including detailed discussion of failure modes and examples on the YCB-Video dataset.

*Computational Cost.* For a common case with 4 views and 6 2D detections per view, our approach takes approximately 320 ms to predict the state of the scene. This timing includes: 190 ms for estimating the 6D poses of all candidates (stage 1, 1 iteration of the coarse and refinement networks), 40 ms for the object candidate association (stage 2) and 90 ms for the scene refinement (stage 3). Further speed-ups towards real-time performance could be achieved, for example, by exploiting temporal continuity in a video sequence.





**Fig. 3. Qualitative results.** We present three examples of scene reconstructions. For each scene, two (out of 4) views that were used to reconstruct the scene are shown as two rows. In each row, the first column shows the input RGB image. The second column shows the 2D detections. The third column shows all object candidates with marked inliers (green) and outliers (red). The fourth column shows the final scene reconstruction. Objects marked by red circles are not in the database, but are sometimes incorrectly detected. Notice how our method estimates accurate 6D object poses for many objects in challenging scenes containing texture-less and symmetric objects, severe occlusions, and where many objects are similar to each other. **More examples are in the supplementary material.** (Color figure online)

## 5 Conclusion

We have developed an approach, dubbed CosyPose, for recovering the 6D pose of multiple known objects viewed by several non-calibrated cameras. Our main contribution is to combine learnable 6D pose estimation with robust multi-view matching and global refinement to reconstruct a single consistent scene. Our approach explicitly handles object symmetries, does not require depth measurements, is robust to missing and incorrect object hypothesis, and automatically recovers the camera poses and the number of objects in the scene. These results make a step towards the robustness and accuracy required for visually driven robotic manipulation in unconstrained scenarios with moving cameras, and open up the possibility of including object pose estimation in an active visual perception loop.

**Acknowledgments.** This work was partially supported by the HPC resources from GENCI-IDRIS (Grant 011011181), the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468), Louis Vuitton ENS Chair on Artificial Intelligence, and the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

## References

1. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
2. Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.* **31**(3), 355–395 (1987)
3. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, September 1999
4. Rad, M., Lepetit, V.: BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3836 (2017)
5. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: pixel-wise voting network for 6DoF pose estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570 (2019)
6. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deep object pose estimation for semantic robotic grasping of household objects. In: *Conference on Robot Learning (CoRL)* (2018)
7. Park, K., Patten, T., Vincze, M.: Pix2Pose: pixel-wise coordinate regression of objects for 6D pose estimation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7668–7677 (2019)
8. Zakharov, S., Shugurov, I., Ilic, S.: DPOD: 6D pose object detector and refiner. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1941–1950 (2019)



9. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6D object pose and size estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2642–2651 (2019)
10. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: deep iterative matching for 6D pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 683–698 (2018)
11. Wang, C., et al.: DenseFusion: 6D object pose estimation by iterative dense fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3343–3352 (2019)
12. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3D orientation learning for 6D object detection from RGB images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 699–715 (2018)
13. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). [https://doi.org/10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32)
14. Hinterstoisser, S., et al.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 International Conference on Computer Vision, pp. 858–865, November 2011
15. Collet, A., Srinivasa, S.S.: Efficient multi-view object recognition and full pose estimation. In: 2010 IEEE International Conference on Robotics and Automation, pp. 2050–2055, May 2010
16. Collet, A., Martinez, M., Srinivasa, S.S.: The moped framework: object recognition and pose estimation for manipulation. *Int. J. Rob. Res.* **30**(10), 1284–1306 (2011)
17. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893, June 2005
18. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. In: Robotics: Science and Systems XIV (2018)
19. Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: an RGB-D dataset for 6D pose estimation of Texture-Less objects. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 880–888, March 2017
20. Li, C., Bai, J., Hager, G.D.: A unified framework for multi-view multi-class object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 254–269 (2018)
21. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: making RGB-based 3D detection and 6D pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1521–1529 (2017)
22. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6D object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 292–301 (2018)
23. Pitteri, G., Ilic, S., Lepetit, V.: CorNet: generic 3D corners for 6D pose estimation of new objects without retraining. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)
24. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, pp. 108–115. IEEE (2001)

25. Pless, R.: Using many cameras as one. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003 Proceedings, vol. 2, II-587, June 2003
26. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H.J., Davison, A.J.: SLAM++: simultaneous localisation and mapping at the level of objects. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1352–1359, June 2013
27. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 998–1005, June 2010
28. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **13**(2), 119–152 (1994)
29. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3583–3592 (2016)
30. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: CVPR 2011, pp. 2025–2032. IEEE (2011)
31. Pillai, S., Leonard, J.: Monocular SLAM supported object recognition. In: Robotics: Science and Systems XI, Robotics: Science and Systems Foundation, July 2015
32. Yang, S., Scherer, S.: CubeSLAM: monocular 3-D object slam. *IEEE Trans. Rob.* **35**(4), 925–938 (2019)
33. Bachmann, R., Spörri, J., Fua, P., Rhodin, H.: Motion capture from pan-tilt cameras with unknown orientation. In: 2019 International Conference on 3D Vision (3DV), pp. 308–317. IEEE (2019)
34. Szeliski, R., Kang, S.B.: Recovering 3D shape and motion from image streams using nonlinear least squares. *J. Vis. Commun. Image Represent.* **5**(1), 10–28 (1994)
35. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2003)
36. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local Affine-Invariant image descriptors and multi-view spatial constraints. *Int. J. Comput. Vis.* **66**(3), 231–259 (2006)
37. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment — a modern synthesis. In: Triggs, B., Zisserman, A., Szeliski, R. (eds.) *IWVA 1999*. LNCS, vol. 1883, pp. 298–372. Springer, Heidelberg (2000). [https://doi.org/10.1007/3-540-44480-7\\_21](https://doi.org/10.1007/3-540-44480-7_21)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
39. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
40. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, Proceedings of Machine Learning Research, PMLR, vol. 97, pp. 6105–6114 (2019)
41. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5745–5753 (2019)

42. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kotschieder, P.: Disentangling monocular 3D object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1991–1999 (2019)
43. Pitteri, G., Ramamonjisoa, M., Ilic, S., Lepetit, V.: On object symmetries and 6D pose estimation from images. In: 2019 International Conference on 3D Vision (3DV), pp. 614–622. IEEE (2019)
44. Hodan, T., et al.: Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 19–34 (2018)
45. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
46. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)