EPOS: Estimating 6D Pose of Objects with Symmetries

Tomáš Hodaň[1] Dániel Baráth[1,2] Jiří Matas[1]

[1]Visual Recognition Group, Czech Technical University in Prague
[2]Machine Perception Research Laboratory, MTA SZTAKI, Budapest

Abstract

We present a new method for estimating the 6D pose of rigid objects with available 3D models from a single RGB input image. The method is applicable to a broad range of objects, including challenging ones with global or partial symmetries. An object is represented by compact surface fragments which allow handling symmetries in a systematic manner. Correspondences between densely sampled pixels and the fragments are predicted using an encoder-decoder network. At each pixel, the network predicts: (i) the probability of each object's presence, (ii) the probability of the fragments given the object's presence, and (iii) the precise 3D location on each fragment. A data-dependent number of corresponding 3D locations is selected per pixel, and poses of possibly multiple object instances are estimated using a robust and efficient variant of the PnP-RANSAC algorithm. In the BOP Challenge 2019, the method outperforms all RGB and most RGB-D and D methods on the T-LESS and LM-O datasets. On the YCB-V dataset, it is superior to all competitors, with a large margin over the second-best RGB method. Source code is at: `cmp.felk.cvut.cz/epos`.

1. Introduction

Model-based estimation of 6D pose, *i.e*. the 3D translation and 3D rotation, of rigid objects from a single image is a classical computer vision problem, with the first methods dating back to the work of Roberts from 1963 [54]. A common approach to the problem is to establish a set of 2D-3D correspondences between the input image and the object model and robustly estimate the pose by the P*n*P-RANSAC algorithm [14, 36]. Traditional methods [9] establish the correspondences using local image features, such as SIFT [41], and have demonstrated robustness against occlusion and clutter in the case of objects with distinct and non-repeatable shape or texture. Recent methods, which are mostly based on convolutional neural networks, produce dense correspondences [4, 48, 69] or predict 2D image locations of pre-selected 3D keypoints [52, 61, 50].

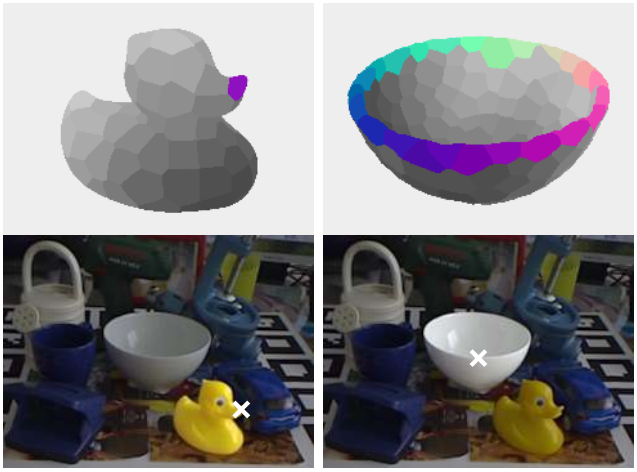

Figure 1. A 2D image location corresponds to a *single* 3D location on the object model in the case of distinct object parts (left), but to *multiple* 3D locations in the case of global or partial object symmetries (right). Representing an object by surface fragments allows predicting *possibly multiple* correspondences per pixel.

Establishing 2D-3D correspondences is challenging for objects with global or partial symmetries [44] in shape or texture. The visible part of such objects, which is determined by self-occlusions and occlusions by other objects, may have multiple fits to the object model. Consequently, the corresponding 2D and 3D locations form a *many-to-many* relationship, *i.e*. a 2D image location may correspond to multiple 3D locations on the model surface (Fig. 1), and vice versa. This degrades the performance of methods assuming a one-to-one relationship. Additionally, methods relying on local image features have a poor performance on texture-less objects, because the feature detectors often fail to provide a sufficient number of reliable locations and the descriptors are no longer discriminative enough [62, 28].

This work proposes a method for estimating 6D pose of possibly multiple instances of possibly multiple rigid objects with available 3D models from a single RGB input image. The method is applicable to a broad range of objects – besides those with distinct and non-repeatable shape or

texture (a shoe, box of corn flakes, *etc*. [41, 9]), the method handles texture-less objects and objects with global or partial symmetries (a bowl, cup, *etc*. [24, 12, 20]).

The key idea is to represent an object by a controllable number of compact surface fragments. This representation allows handling symmetries in a systematic manner and ensures a consistent number and uniform coverage of candidate 3D locations on objects of any type. Correspondences between densely sampled pixels and the surface fragments are predicted using an encoder-decoder convolutional neural network. At each pixel, the network predicts (i) the probability of each object's presence, (ii) the probability of the fragments given the object's presence, and (iii) the precise 3D location on each fragment (Fig. 2). By modeling the probability of fragments conditionally, the uncertainty due to object symmetries is decoupled from the uncertainty of the object's presence and is used to guide the selection of a data-dependent number of 3D locations at each pixel.

Poses of possibly multiple object instances are estimated from the predicted many-to-many 2D-3D correspondences by a robust and efficient variant of the P*n*P-RANSAC algorithm [36] integrated in the Progressive-X scheme [3]. Pose hypotheses are proposed by GC-RANSAC [2] which utilizes the spatial coherence of correspondences – close correspondences (in 2D and 3D) likely belong to the same pose. Efficiency is achieved by the PROSAC sampler [8] that prioritizes correspondences with a high predicted probability.

The proposed method is compared with the participants of the BOP Challenge 2019 [23, 26]. The method outperforms all RGB methods and most RGB-D and D methods on the T-LESS [24] and LM-O [4] datasets, which include texture-less and symmetric objects captured in cluttered scenes under various levels of occlusion. On the YCB-V [68] dataset, which includes textured and texture-less objects, the method is superior to all competitors, with a significant 27% absolute improvement over the second-best RGB method. These results are achieved without any post-refinement of the estimated poses, such as [43, 38, 69, 52].

This work makes the following contributions:

1. *A 6D object pose estimation method* applicable to a broad range of objects, including objects with symmetries, achieving the state-of-the-art RGB-only results on the standard T-LESS, YCB-V and LM-O datasets.
2. *Object representation by surface fragments* allowing to handle symmetries in a systematic manner and ensuring a consistent number and uniform coverage of candidate 3D locations on any object.
3. *Many-to-many 2D-3D correspondences* established by predicting a data-dependent number of precise 3D locations at each pixel.
4. *A robust and efficient estimator* for recovering poses of multiple object instances, with a demonstrated benefit over standard P*n*P-RANSAC variants.

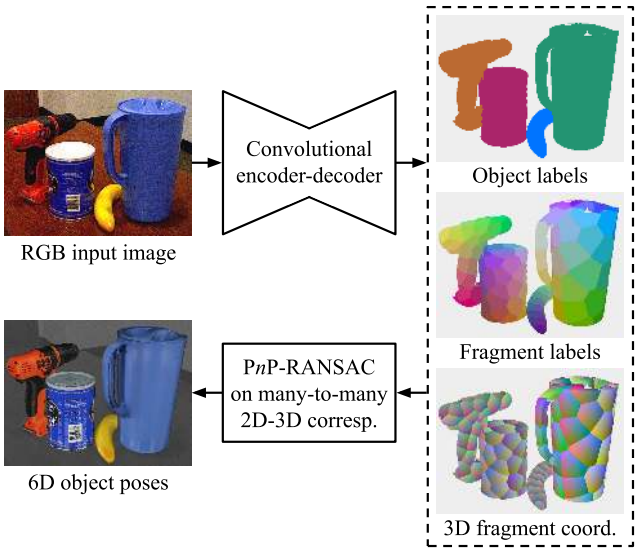

Figure 2. **EPOS pipeline.** During training, an encoder-decoder network is provided a per-pixel annotation in the form of an object label, a fragment label, and 3D fragment coordinates. During inference, 3D locations on *possibly multiple* fragments are predicted at each pixel, which allows to capture object symmetries. Many-to-many 2D-3D correspondences are established by linking pixels with the predicted 3D locations, and a robust and efficient variant of the P*n*P-RANSAC algorithm is used to estimate the 6D poses.

2. Related Work

Classical Methods. In the early attempt, Roberts [54] assumed that objects can be constructed from transformations of known simple 3D models which were fit to edges extracted from a grayscale input image. The first practical approaches were relying on local image features [41, 9] or template matching [5], and assumed a grayscale or RGB input image. Later, with the introduction of the consumer-grade Kinect-like sensors, the attention of the research field was steered towards estimating the object pose from RGB-D images. Methods based on RGB-D template matching [20, 28], point-pair features [13, 21, 66], 3D local features [17], and learning-based methods [4, 60, 35] demonstrated a superior performance over RGB-only counterparts.

CNN-Based Methods. Recent methods are based on convolutional neural networks (CNN's) and focus primarily on estimating the object pose from RGB images. A popular approach is to establish 2D-3D correspondences by predicting the 2D projections of a fixed set of 3D keypoints, which are pre-selected for each object model, and solve for the object pose using P*n*P-RANSAC [52, 49, 47, 61, 65, 15, 29, 50]. Methods establishing the correspondences in the opposite direction, *i.e*. by predicting the 3D object coordinates [4] for a densely sampled set of pixels, have been also proposed [32, 46, 69, 48, 39]. As discussed below, none of the

existing correspondence-based methods can reliably handle pose ambiguity due to object symmetries.

Another approach is to localize the objects with 2D bounding boxes, and for each box predict the pose by regression [68, 37, 42] or by classification into discrete viewpoints [33, 10, 59]. However, in the case of occlusion, estimating accurate 2D bounding boxes covering the whole object (including the invisible parts) is problematic [33].

Despite promising results, the recent CNN-based RGB methods are inferior to the classical RGB-D and D methods, as reported in [23, 26]. Using the depth image as an additional input of the CNN is a promising research direction [37, 58, 67], but with a limited range of applications.

Handling Object Symmetries. The many-to-many relationship of corresponding 2D and 3D locations, which arises in the case of object symmetries (Sec. 1), degrades the performance of correspondence-based methods which assume a one-to-one relationship. In particular, classification-based methods predict for each pixel up to one corresponding 3D location [4, 46], or for each 3D keypoint up to one 2D location which is typically given by the maximum response in a predicted heatmap [49, 47, 15]. This may result in a set of correspondences which carries only a limited support for each of the possible poses. On the other hand, regression-based methods [61, 69, 50] need to compromise among the possible corresponding locations and tend to return the average, which is often not a valid solution. For example, the average of all points on a sphere is the center of the sphere, which is not a valid surface location.

The problem of pose ambiguity due to object symmetries has been approached by several methods. Rad and Lepetit [52] assume that the global object symmetries are known and propose a pose normalization applicable to the case when the projection of the axis of symmetry is close to vertical. Pitteri *et al*. [51] introduce a pose normalization that is not limited to this special case. Kehl *et al*. [33] train a classifier for only a subset of viewpoints defined by global object symmetries. Corona *et al*. [10] show that predicting the order of rotational symmetry can improve the accuracy of pose estimation. Xiang *et al*. [68] optimize a loss function that is invariant to global object symmetries. Park *et al*. [48] guide pose regression by calculating the loss w.r.t. to the closest symmetric pose. However, all of these approaches cover only pose ambiguities due to global object symmetries. Ambiguities due to partial object symmetries (*i.e*. when the visible object part has multiple possible fits to the entire object surface) are not covered.

As EPOS, the methods by Manhardt *et al*. [42] and Li *et al*. [37] can handle pose ambiguities due to both global and partial object symmetries without requiring any a priori information about the symmetries. The first [42] predicts multiple poses for each object instance to estimate the distribution of possible poses induced by symmetries. The second [37] deals with the possibly non-unimodal pose distribution by a classification and regression scheme applied to the rotation and translation space. Nevertheless, both methods rely on estimating accurate 2D bounding boxes which is problematic when the objects are occluded [33].

Object Representation. To increase the robustness of 6D object pose tracking against occlusion, Crivellaro *et al*. [11] represent an object by a set of parts and estimate the 6D pose of each part by predicting the 2D projections of preselected 3D keypoints. Brachmann *et al*. [4] and Nigam *et al*. [46] split the 3D bounding box of the object model into uniform bins and predict up to one corresponding bin per pixel. They represent each bin with its center which yields correspondences with limited precision.

For human pose estimation, Güler *et al*. [1] segment the 3D surface of the human body into semantically-defined parts. At each pixel, they predict a label of the corresponding part and the UV texture coordinates defining the precise location on the part. In contrast, to effectively capture the partial object symmetries, we represent an object by a set of compact surface fragments of near-uniform size and predict possibly multiple labels of the corresponding fragments per pixel. Besides, we regress the precise location in local 3D coordinates of the fragment instead of the UV coordinates. Using the UV coordinates requires a well-defined topology of the mesh model, which may need manual intervention, and is problematic for objects with a complicated surface such as a coil or an engine [12].

Model Fitting. Many of the recent correspondence-based methods, *e.g*. [48, 69, 52, 61], estimate the pose using the vanilla P*n*P-RANSAC algorithm [14, 36] implemented in the OpenCV function `solvePnPRansac`. We show that a noticeable improvement can be achieved by replacing the vanilla with a modern robust estimator.

3. EPOS: The Proposed Method

This section provides a detailed description of the proposed model-based method for 6D object pose estimation. The 3D object models are the only necessary training input of the method. Besides a synthesis of automatically annotated training images [27], the models are useful for applications such as robotic grasping or augmented reality.

3.1. Surface Fragments

A mesh model defined by a set of 3D vertices, V_i, and a set of triangular faces, T_i, is assumed available for each object with index $i \in I = \{1, \ldots, m\}$. The set of all 3D points on the model surface, S_i, is split into n fragments with indices $J = \{1, \ldots, n\}$. Surface fragment j of object i is defined as $S_{ij} = \{\boldsymbol{x} \,|\, \boldsymbol{x} \in S_i \wedge d(\boldsymbol{x}, \boldsymbol{g}_{ij}) < d(\boldsymbol{x}, \boldsymbol{g}_{ik})\}$, $\forall k \in J, k \neq j$, where $d(.)$ is the Euclidean distance of two 3D points and $\{\boldsymbol{g}_{ij}\}_{j=1}^{n}$ are pre-selected fragment centers.

The fragment centers are found by the furthest point sampling algorithm which iteratively selects the vertex from V_i that is furthest from the already selected vertices. The algorithm starts with the centroid of the object model which is then discarded from the final set of centers.

3.2. Prediction of 2D-3D Correspondences

Decoupling Uncertainty Due to Symmetries. The probability of surface fragment j of object i being visible at pixel $\boldsymbol{u} = (u, v)$ is modeled as:

$$\Pr(f{=}j, o{=}i \,|\, \boldsymbol{u}) = \Pr(f{=}j \,|\, o{=}i, \boldsymbol{u}) \Pr(o{=}i \,|\, \boldsymbol{u}),$$

where o and f are random variables representing the object and fragment respectively. The probability can be low because (1) object i is not visible at pixel $\boldsymbol{u}$, or (2) $\boldsymbol{u}$ corresponds to multiple fragments due to global or partial symmetries of object i. To disentangle the two cases, we predict $a_i(\boldsymbol{u}) = \Pr(o{=}i \,|\, \boldsymbol{u})$ and $b_{ij}(\boldsymbol{u}) = \Pr(f{=}j \,|\, o{=}i, \boldsymbol{u})$ separately, instead of directly predicting $\Pr(f{=}j, o{=}i \,|\, \boldsymbol{u})$.

Regressing Precise 3D Locations. Surface fragment j of object i is associated with a regressor, $\boldsymbol{r}_{ij} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, which at pixel $\boldsymbol{u}$ predicts the corresponding 3D location: $\boldsymbol{r}_{ij}(\boldsymbol{u}) = (\boldsymbol{x} - \boldsymbol{g}_{ij})/h_{ij}$. The predicted location is expressed in *3D fragment coordinates*, *i.e.* in a 3D coordinate frame with the origin at the fragment center $\boldsymbol{g}_{ij}$. Scalar h_{ij} normalizes the regression range and is defined as the length of the longest side of the 3D bounding box of the fragment.

Dense Prediction. A single deep convolutional neural network with an encoder-decoder structure, DeepLabv3+ [6], is adopted to densely predict $a_k(\boldsymbol{u})$, $b_{ij}(\boldsymbol{u})$ and $\boldsymbol{r}_{ij}(\boldsymbol{u})$, $\forall i \in I$, $\forall j \in J$, $\forall k \in I \cup \{0\}$, where 0 is reserved for the background class. For m objects, each represented by n surface fragments, the network has $4mn{+}m{+}1$ output channels ($m{+}1$ for probabilities of the objects and the background, mn for probabilities of the surface fragments, and $3mn$ for the 3D fragment coordinates).

Network Training. The network is trained by minimizing the following loss averaged over all pixels $\boldsymbol{u}$:

$$\begin{aligned} L(\boldsymbol{u}) = {} & E\big(\bar{\boldsymbol{a}}(\boldsymbol{u}), \boldsymbol{a}(\boldsymbol{u})\big) + \\ & \sum\nolimits_{i \in I} \bar{a}_i(\boldsymbol{u}) \Big[\lambda_1 E\big(\bar{\boldsymbol{b}}_i(\boldsymbol{u}), \boldsymbol{b}_i(\boldsymbol{u})\big) + \\ & \sum\nolimits_{j \in J} \bar{b}_{ij}(\boldsymbol{u}) \lambda_2 H\big(\bar{\boldsymbol{r}}_{ij}(\boldsymbol{u}), \boldsymbol{r}_{ij}(\boldsymbol{u})\big) \Big], \end{aligned}$$

where E is the softmax cross entropy loss and H is the Huber loss [30]. Vector $\boldsymbol{a}(\boldsymbol{u})$ consists of all predicted probabilities $a_i(\boldsymbol{u})$, and vector $\boldsymbol{b}_i(\boldsymbol{u})$ of all predicted probabilities $b_{ij}(\boldsymbol{u})$ for object i. The ground-truth one-hot vectors $\bar{\boldsymbol{a}}(\boldsymbol{u})$ and $\bar{\boldsymbol{b}}_i(\boldsymbol{u})$ indicate which object (or the background) and which fragment is visible at $\boldsymbol{u}$. Elements of these ground-truth vectors are denoted as $\bar{a}_i(\boldsymbol{u})$ and $\bar{b}_{ij}(\boldsymbol{u})$. Vector $\bar{\boldsymbol{b}}_i(\boldsymbol{u})$ is defined only if object i is present at $\boldsymbol{u}$. The ground-truth 3D fragment coordinates are denoted as $\bar{\boldsymbol{r}}_{ij}(\boldsymbol{u})$. Weights λ_1 and λ_2 are used to balance the loss terms.

The network is trained on images annotated with ground-truth 6D object poses. Vectors $\bar{\boldsymbol{a}}(\boldsymbol{u})$, $\bar{\boldsymbol{b}}_i(\boldsymbol{u})$, and $\bar{\boldsymbol{r}}_{ij}(\boldsymbol{u})$ are obtained by rendering the 3D object models in the ground-truth poses with a custom OpenGL shader. Pixels outside the visibility masks of the objects are considered to be the background. The masks are calculated as in [25].

Learning Object Symmetries. Identifying all possible correspondences for training the network is not trivial. One would need to identify the visible object parts in each training image and find their fits to the object models. Instead, we provide the network with only a single corresponding fragment per pixel during training and let the network learn the object symmetries implicitly. Minimizing the softmax cross entropy loss $E\left(\bar{\boldsymbol{b}}_i(\boldsymbol{u}), \boldsymbol{b}_i(\boldsymbol{u})\right)$ corresponds exactly to minimizing the Kullback-Leibler divergence of distributions $\bar{\boldsymbol{b}}_i(\boldsymbol{u})$ and $\boldsymbol{b}_i(\boldsymbol{u})$ [16]. Hence, if the ground-truth one-hot distribution $\bar{\boldsymbol{b}}_i(\boldsymbol{u})$ indicates a different fragment at pixels with similar appearance, the network is expected to learn at such pixels the same probability $b_{ij}(\boldsymbol{u})$ for all the indicated fragments. This assumes that the object poses are distributed uniformly in the training images, which is easy to ensure with synthetic training images.

Establishing Correspondences. Pixel $\boldsymbol{u}$ is linked with a 3D location, $\boldsymbol{x}_{ij}(\boldsymbol{u}) = h_{ij}\boldsymbol{r}_{ij}(\boldsymbol{u}) + \boldsymbol{g}_{ij}$, on every fragment for which $a_i(\boldsymbol{u}) > \tau_a$ and $b_{ij}(\boldsymbol{u}) / \max_{k=1}^{n}(b_{ik}(\boldsymbol{u})) > \tau_b$. Threshold τ_b is relative to the maximum to collect locations from all indistinguishable fragments that are expected to have similarly high probability $b_{ij}(\boldsymbol{u})$. For example, the probability distribution on a sphere is expected to be uniform, *i.e.* $b_{ij}(\boldsymbol{u}) = 1/n, \forall j \in J$. On a bowl, the probability is expected to be constant around the axis of symmetry.

The set of correspondences established for instances of object i is denoted as $C_i = \{(\boldsymbol{u}, \boldsymbol{x}_{ij}(\boldsymbol{u}), s_{ij}(\boldsymbol{u}))\}$, where $s_{ij}(\boldsymbol{u}) = a_i(\boldsymbol{u}) b_{ij}(\boldsymbol{u})$ is the confidence of a correspondence. The set forms a many-to-many relationship between the 2D image locations and the predicted 3D locations.

3.3. Robust and Efficient 6D Pose Fitting

Sources of Outliers. With respect to a single object pose hypothesis, set C_i of the many-to-many 2D-3D correspondences includes three types of outliers. First, it includes outliers due to erroneous prediction of the 3D locations. Second, for each 2D/3D location there is up to one correspondence which is compatible with the pose hypothesis; the other correspondences act as outliers. Third, correspondences originating from different instances of object i are also incompatible with the pose hypothesis. Set C_i may be therefore contaminated with a high proportion of outliers and a robust estimator is needed to achieve stable results.

Multi-Instance Fitting. To estimate poses of possibly multiple instances of object i from correspondences C_i, we use a robust and efficient variant of the PnP-RANSAC algorithm [14, 36] integrated in the Progressive-X scheme [3][1]. In this scheme, pose hypotheses are proposed sequentially and added to a set of maintained hypotheses by the PEARL optimization [31], which minimizes the energy calculated over all hypotheses and correspondences. PEARL utilizes the spatial coherence of correspondences – the closer they are (in 2D and 3D), the more likely they belong to the same pose of the same object instance. To reason about the spatial coherence, a neighborhood graph is constructed by describing each correspondence by a 5D vector consisting of the 2D and 3D coordinates (in pixels and centimeters), and linking two 5D descriptors if their Euclidean distance is below threshold τ_d. The inlier-outlier threshold, denoted as τ_r, is set manually and defined on the re-projection error [36].

Hypothesis Proposal. The pose hypotheses are proposed by GC-RANSAC [2][2], a locally optimized RANSAC which selects the inliers by the s-t graph-cut optimization. GC-RANSAC utilizes the spatial coherence via the same neighborhood graph as PEARL. The pose is estimated from a sampled triplet of correspondences by the P3P solver [34], and refined from all inliers by the EPnP solver [36] followed by the Levenberg-Marquardt optimization [45]. The triplets are sampled by PROSAC [8], which first focuses on correspondences with high confidence s_{ij} (Sec. 3.2) and progressively blends to a uniform sampling.

Hypothesis Verification. Inside GC-RANSAC, the quality of a pose hypothesis, denoted as $\hat{\mathbf{P}}$, is calculated as:

$$q = 1/|U_i| \sum\nolimits_{\boldsymbol{u} \in U_i} \max\nolimits_{\boldsymbol{c} \in C_{i\boldsymbol{u}}} \max\Big(0,\, 1 - e^2\big(\hat{\mathbf{P}}, \boldsymbol{c}\big)/\tau_r^2\Big),$$

where U_i is a set of pixels at which correspondences C_i are established, $C_{i\boldsymbol{u}} \subset C_i$ is a subset established at pixel $\boldsymbol{u}$, $e\big(\hat{\mathbf{P}}, \boldsymbol{c}\big)$ is the re-projection error [36], and τ_r is the inlier-outlier threshold. At each pixel, quality q considers only the most accurate correspondence as only up to one correspondence may be compatible with the hypothesis; the others provide alternative explanations and should not influence the quality. GC-RANSAC runs for up to τ_i iterations until quality q of an hypothesis reaches threshold τ_q. The hypothesis with the highest q is the outcome of each proposal stage and is integrated into the set of maintained hypotheses.

Degeneracy Testing. Sampled triplets which form 2D triangles with the area below τ_t or have collinear 3D locations are rejected. Moreover, pose hypotheses behind the camera or with the determinant of the rotation matrix equal to -1 (*i.e.* an improper rotation matrix [18]) are discarded.

[1] https://github.com/danini/progressive-x
[2] https://github.com/danini/graph-cut-ransac

Figure 3. **Example EPOS results** on T-LESS (top), YCB-V (middle) and LM-O (bottom). On the right are renderings of the 3D object models in poses estimated from the RGB images on the left. All eight LM-O objects, including two truncated ones, are detected in the bottom example. More examples are on the project website.

4. Experiments

This section compares the performance of EPOS with other model-based methods for 6D object pose estimation and presents ablation experiments.

4.1. Experimental Setup

Evaluation Protocol. We follow the evaluation protocol of the BOP Challenge 2019 [23, 26] (BOP19 for short). The task is to estimate the 6D poses of a varying number of instances of a varying number of objects in a single image, with the number of instances provided with each image.

The error of an estimated pose $\hat{\mathbf{P}}$ w.r.t. the ground-truth pose $\bar{\mathbf{P}}$ is calculated by three pose-error functions. The first, Visible Surface Discrepancy, treats indistinguishable poses as equivalent by considering only the visible object part:

$$e_{\text{VSD}} = \operatorname*{avg}_{p \in \hat{V} \cup \bar{V}} \begin{cases} 0 & \text{if } p \in \hat{V} \cap \bar{V} \;\wedge\; |\hat{D}(p) - \bar{D}(p)| < \tau \\ 1 & \text{otherwise,} \end{cases}$$

where $\hat{D}$ and $\bar{D}$ are distance maps obtained by rendering the object model in the estimated and the ground-truth pose respectively. The distance maps are compared with distance map D_I of test image I in order to obtain visibility masks

$\hat{V}$ and $\bar{V}$, *i.e.* sets of pixels where the object model is visible in image I. The distance map D_I is available for all images included in BOP. Parameter τ is a misalignment tolerance.

The second pose-error function, Maximum Symmetry-Aware Surface Distance, measures the surface deviation in 3D and is therefore relevant for robotic applications:

$$e_{\text{MSSD}} = \min_{\mathbf{T} \in T_i} \max_{\mathbf{x} \in V_i} \|\hat{\mathbf{P}}\mathbf{x} - \bar{\mathbf{P}}\mathbf{T}\mathbf{x}\|_2,$$

where T_i is a set of symmetry transformations of object i (provided in BOP19), and V_i is a set of model vertices.

The third pose-error function, Maximum Symmetry-Aware Projection Distance, measures the perceivable deviation. It is relevant for augmented reality applications and suitable for the evaluation of RGB methods, for which estimating the Z translational component is more challenging:

$$e_{\text{MSPD}} = \min_{\mathbf{T} \in T_i} \max_{\mathbf{x} \in V_i} \|\text{proj}(\hat{\mathbf{P}}\mathbf{x}) - \text{proj}(\bar{\mathbf{P}}\mathbf{T}\mathbf{x})\|_2,$$

where proj(.) denotes the 2D projection operation and the meaning of the other symbols is as in e_{MSSD}.

An estimated pose is considered correct w.r.t. pose-error function e if $e < \theta_e$, where $e \in \{e_{\text{VSD}}, e_{\text{MSSD}}, e_{\text{MSPD}}\}$ and θ_e is the threshold of correctness. The fraction of annotated object instances, for which a correct pose is estimated, is referred to as recall. The Average Recall w.r.t. function e (AR_e) is defined as the average of the recall rates calculated for multiple settings of threshold θ_e, and also for multiple settings of the misalignment tolerance τ in the case of e_{VSD}. The overall performance of a method is measured by the Average Recall: $\text{AR} = (\text{AR}_{\text{VSD}} + \text{AR}_{\text{MSSD}} + \text{AR}_{\text{MSPD}}) / 3$. As EPOS uses only RGB, besides AR we report AR_{MSPD}.

Datasets. The experiments are conducted on three datasets: T-LESS [24], YCB-V [68], LM-O [4]. The datasets include color 3D object models and RGB-D images of VGA resolution with ground-truth 6D object poses (EPOS uses only the RGB channels). The same subsets of test images as in BOP19 were used. LM-O contains 200 test images with the ground truth for eight, mostly texture-less objects from LM [20] captured in a clutered scene under various levels of occlusion. YCB-V includes 21 objects, which are both textured and texture-less, and 900 test images showing the objects with occasional occlusions and limited clutter. T-LESS contains 30 objects with no significant texture or discriminative color, and with symmetries and mutual similarities in shape and/or size. It includes 1000 test images from 20 scenes with varying complexity, including challenging scenes with multiple instances of several objects and with a high amount of clutter and occlusion.

Training Images. The network is trained on several types of synthetic images. For T-LESS, we use 30K physically-based rendered (PBR) images from SyntheT-LESS [51], 50K images of objects rendered with OpenGL on random photographs from NYU Depth V2 [57] (similarly to [22]), and 38K real images from [24] showing objects on black background, where we replaced the background with random photographs. For YCB-V, we use the provided 113K real and 80K synthetic images. For LM-O, we use 67K PBR images from [27] (scenes 1 and 2), and 50K images of objects rendered with OpenGL on random photographs. No real images of the objects are used for training on LM-O.

Optimization. We use the DeepLabv3+ encoder-decoder network [6] with Xception-65 [7] as the backbone. The network is pre-trained on Microsoft COCO [40] and fine-tuned on the training images described above for 2M iterations. The batch size is set to 1, initial learning rate to 0.0001, parameters of batch normalization are not fine-tuned and other hyper-parameters are set as in [6].

To overcome the domain gap between the synthetic training and real test images, we apply the simple technique from [22] and freeze the "early flow" part of Xception-65. For LM-O, we additionally freeze the "middle flow" since there are no real training images in this dataset. The training images are augmented by randomly adjusting brightness, contrast, hue, and saturation, and by applying random Gaussian noise and blur, similarly to [22].

Method Parameters. The rates of atrous spatial pyramid pooling in the DeepLabv3+ network are set to 12, 24, and 36, and the output stride to 8 px. The spatial resolution of the output channels is doubled by the bilinear interpolation, *i.e.* locations $\boldsymbol{u}$ for which the predictions are made are at the centers of 4×4 px regions in the input image. A single network per dataset is trained, each object is represented by $n = 64$ fragments (unless stated otherwise), and the other parameters are set as follows: $\lambda_1 = 1$, $\lambda_2 = 100$, $\tau_a = 0.1$, $\tau_b = 0.5$, $\tau_d = 20$, $\tau_r = 4$ px, $\tau_i = 400$, $\tau_q = 0.5$, $\tau_t = 100$ px.

4.2. Main Results

Accuracy. Tab. 1 compares the performance of EPOS with the participants of the BOP Challenge 2019 [23, 26]. EPOS outperforms all RGB methods on all three datasets by a large margin in both AR and AR_{MSPD} scores. On the YCB-V dataset, it achieves 27% absolute improvement in both scores over the second-best RGB method and also outperforms all RGB-D and D methods. On the T-LESS and LM-O datasets, which include symmetric and texture-less objects, EPOS achieves the overall best AR_{MSPD} score.

As the BOP rules require the method parameters to be fixed across datasets, Tab. 1 reports scores achieved with objects from all datasets represented by 64 fragments. As reported in Tab. 2, increasing the number of fragments from 64 to 256 yields in some cases additional improvements but around double image processing time. Note that we do not perform any post-refinement of the estimated poses, such as [43, 38, 69, 52], which could improve the accuracy further.

6D object pose estimation method	Image	T-LESS [24]		YCB-V [68]		LM-O [4]		Time
		AR	AR_{MSPD}	AR	AR_{MSPD}	AR	AR_{MSPD}	
EPOS	RGB	**47.6**	**63.5**	**69.6**	**78.3**	**44.3**	**65.9**	0.75
Zhigang-CDPN-ICCV19 [39]	RGB	12.4	17.0	42.2	51.2	37.4	55.8	0.67
Sundermeyer-IJCV19 [59]	RGB	30.4	50.4	37.7	41.0	14.6	25.4	0.19
Pix2Pose-BOP-ICCV19 [48]	RGB	27.5	40.3	29.0	40.7	7.7	16.5	0.81
DPOD-ICCV19 (synthetic) [69]	RGB	8.1	13.9	22.2	25.6	16.9	27.8	0.24
Pix2Pose-BOP_w/ICP-ICCV19 [48]	RGB-D	–	–	**67.5**	**63.0**	–	–	–
Drost-CVPR10-Edges [13]	RGB-D	**50.0**	**51.8**	37.5	27.5	**51.5**	**56.9**	144.10
Félix&Neves-ICRA17-IET19 [55, 53]	RGB-D	21.2	21.3	51.0	38.4	39.4	43.0	52.97
Sundermeyer-IJCV19+ICP [59]	RGB-D	48.7	51.4	50.5	47.5	23.7	28.5	1.10
Vidal-Sensors18 [66]	D	**53.8**	**57.4**	**45.0**	**34.7**	**58.2**	**64.7**	4.93
Drost-CVPR10-3D-Only [13]	D	44.4	48.0	34.4	26.3	52.7	58.1	10.47
Drost-CVPR10-3D-Only-Faster [13]	D	40.5	43.6	33.0	24.4	49.2	54.2	2.20

Table 1. **BOP Challenge 2019** [23, 26] results on datasets T-LESS, YCB-V and LM-O, with objects represented by 64 surface fragments. Top scores for image types are **bold**, the best overall are **blue**. The time [s] is the average image processing time averaged over the datasets.

Speed. With an unoptimized implementation, EPOS takes 0.75 s per image on average (with a 6-core Intel i7-8700K CPU, 64GB RAM, and Nvidia P100 GPU). As the other RGB methods, which are all based on convolutional neural networks, EPOS is noticeably faster than the RGB-D and D methods (Tab. 1), which are slower typically due to an ICP post-processing step [56]. The RGB methods of [59, 69] are 3–4 times faster but significantly less accurate than EPOS. Depending on the application requirements, the trade-off between the accuracy and speed of EPOS can be controlled by, *e.g.*, the number of surface fragments, the network size, the image resolution, the density of pixels at which the correspondences are predicted, or the maximum allowed number of GC-RANSAC iterations.

4.3. Ablation Experiments

Surface Fragments. The performance scores of EPOS for different numbers of surface fragments are shown in the upper half of Tab. 2. With a single fragment, the method performs direct regression of the so-called 3D object coordinates [4], similarly to [32, 48, 39]. The accuracy increases with the number of fragments and reaches the peak at 64 or 256 fragments. On all three datasets, the peaks of both AR and AR_{MSPD} scores are 18–33% higher than the scores achieved with the direct regression of the 3D object coordinates. This significant improvement demonstrates the effectiveness of fragments on various types of objects, including textured, texture-less, and symmetric objects.

On T-LESS, the accuracy drops when the number of fragments is increased from 64 to 256. We suspect this is because the fragments become too small (T-LESS includes smaller objects) and training of the network becomes challenging due to a lower number of examples per fragment.

The average number of correspondences increases with the number of fragments, *i.e.* each pixel gets linked with more fragments (columns Corr. in Tab. 2). At the same time, the average number of fitting iterations tends to decrease (columns Iter.). This shows that the pose fitting method can benefit from knowing more possible correspondences per pixel – GC-RANSAC finds a pose hypothesis with quality q (Sec. 3.3) reaching threshold τ_q in less iterations. However, although the average number of iterations decreases, the average image processing time tends to increase (at higher numbers of fragments) due to a higher computational cost of the network inference and of each fitting iteration. Setting the number of fragments to 64 provides a practical trade-off between the speed and accuracy.

Regression of 3D Fragment Coordinates. The upper half of Tab. 2 shows scores achieved with regressing the precise 3D locations, while the lower half shows scores achieved with the same network models but using the fragment centers (Sec. 3.1) instead of the regressed locations. Without the regression, the scores increase with the number of fragments as the deviation of the fragment centers from the true corresponding 3D locations decreases. However, the accuracy is often noticeably lower than with the regression. With a single fragment and without the regression, all pixels are linked to the same fragment center and all samples of three correspondences are immediately rejected because they fail the non-collinearity test, hence the low processing time.

Even though the regressed 3D locations are not guaranteed to lie on the model surface, their average distance from the surface is less than 1 mm (with 64 and 256 fragments), which is negligible compared to the object sizes. No improvement was observed when the regressed locations were replaced by the closest points on the object model.

n	T-LESS [24]					YCB-V [68]					LM-O [4]				
	AR	AR_{MSPD}	Corr.	Iter.	Time	AR	AR_{MSPD}	Corr.	Iter.	Time	AR	AR_{MSPD}	Corr.	Iter.	Time
With regression of 3D fragment coordinates															
1	17.2	30.7	911	347	0.97	41.7	52.6	1079	183	0.56	26.8	47.5	237	111	0.53
4	39.5	57.1	1196	273	0.95	54.4	66.1	1129	110	0.52	33.5	56.0	267	58	0.51
16	45.4	62.7	1301	246	0.96	63.2	72.7	1174	71	0.51	39.3	61.3	275	54	0.50
64	**47.6**	**63.5**	1612	236	1.18	69.6	78.3	1266	56	0.57	44.3	**65.9**	330	53	0.49
256	45.6	59.7	3382	230	2.99	**71.4**	**79.8**	1497	56	0.94	**46.0**	65.4	457	70	0.60
Without regression of 3D fragment coordinates															
1	0.0	0.0	911	400	0.23	0.0	0.0	1079	400	0.17	0.0	0.0	237	400	0.24
4	3.2	8.8	1196	399	0.89	3.0	7.4	1129	400	0.53	5.2	15.2	267	390	0.50
16	13.9	37.5	1301	396	1.02	16.1	36.4	1174	400	0.61	17.1	47.7	275	359	0.55
64	29.4	55.0	1612	380	1.35	41.5	66.6	1266	383	0.73	31.0	62.3	330	171	0.55
256	43.0	58.2	3382	299	2.95	64.5	77.7	1497	206	0.88	43.2	64.9	457	72	0.58

Table 2. **Number of fragments and regression.** Performance scores for different numbers of surface fragments (n) with and without regression of the 3D fragment coordinates (the fragment centers are used in the case of no regression). The table also reports the average number of correspondences established per object model in an image, the average number of GC-RANSAC iterations to fit a single pose (both are rounded to integers), and the average image processing time [s].

RANSAC variant	Non-minimal solver	T-LESS [24]		YCB-V [68]		LM-O [4]		Time
		AR	AR_{MSPD}	AR	AR_{MSPD}	AR	AR_{MSPD}	
OpenCV RANSAC	EP*n*P [36]	35.5	47.9	67.2	76.6	41.2	63.5	0.16
MSAC [64]	EP*n*P [36] + LM [45]	44.3	61.0	63.8	73.7	39.7	61.7	0.49
GC-RANSAC [2]	DLS-P*n*P [19]	44.3	59.5	67.5	76.1	35.6	53.9	0.53
GC-RANSAC [2]	EP*n*P [36]	46.9	62.6	69.2	77.9	42.6	63.6	0.39
GC-RANSAC [2]	EP*n*P [36] + LM [45]	**47.6**	**63.5**	**69.6**	**78.3**	**44.3**	**65.9**	0.52

Table 3. **RANSAC variants and non-minimal solvers.** The P3P solver [34] is used to estimate the pose from a minimal sample of 2D-3D correspondences. The non-minimal solvers are applied when estimating the pose from a larger-than-minimal sample. The reported time [s] is the average time to fit poses of all object instances in an image averaged over the datasets.

Robust Pose Fitting. Tab. 3 evaluates several methods for robust pose estimation from the 2D-3D correspondences: RANSAC [14] from OpenCV, MSAC [63], and GC-RANSAC [2]. The methods were evaluated within the Progressive-X scheme (Sec. 3.3), with the P3P solver [34] to estimate the pose from a minimal sample, *i.e*. three correspondences, and with several solvers to estimate the pose from a non-minimal sample. In OpenCV RANSAC and MSAC, the non-minimal solver refines the pose from all inliers. In GC-RANSAC, it is additionally used in the graph-cut-based local optimization which is applied when a new so-far-the-best pose is found. We tested OpenCV RANSAC with all available non-minimal solvers and achieved the best scores with EP*n*P [36]. The top-performing estimation method on all datasets is GC-RANSAC with EP*n*P followed by the Levenberg-Marquardt optimization [45] as the non-minimal solver. Note the gap in accuracy, especially on T-LESS, between this method and OpenCV RANSAC.

5. Conclusion

We have proposed a new model-based method for 6D object pose estimation from a single RGB image. The key idea is to represent an object by compact surface fragments, predict possibly multiple corresponding 3D locations at each pixel, and solve for the pose using a robust and efficient variant of the P*n*P-RANSAC algorithm. The experimental evaluation has demonstrated the method to be applicable to a broad range of objects, including challenging objects with symmetries. A study of object-specific numbers of fragments, which may depend on factors such as the physical object size, shape or the range of distances of the object from the camera, is left for future work. The project website with source code is at: `cmp.felk.cvut.cz/epos`.

This research was supported by Research Center for Informatics (project CZ.02.1.01/0.0/0.0/16_019/0000765 funded by OP VVV), CTU student grant (SGS OHK3-019/20), and grant "Exploring the Mathematical Foundations of Artificial Intelligence" (2018-1.2.1-NKP-00008).

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. *CVPR*, 2018. 3

[2] Daniel Barath and Jiri Matas. Graph-Cut RANSAC. *CVPR*. 2, 5, 8

[3] Daniel Barath and Jiri Matas. Progressive-X: Efficient, anytime, multi-model fitting algorithm. *ICCV*. 2, 5

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. *ECCV*, 2014. 1, 2, 3, 6, 7, 8

[5] Roberto Brunelli. Template matching techniques in computer vision: Theory and practice. 2009. 2

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV*, 2018. 4, 6

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CVPR*, 2017. 6

[8] Ondrej Chum and Jiri Matas. Matching with PROSAC – Progressive sample consensus. *CVPR)*, 2005. 2, 5

[9] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *IJRR*, 2011. 1, 2

[10] Enric Corona, Kaustav Kundu, and Sanja Fidler. Pose estimation for objects with rotational symmetry. *IROS*, 2018. 3

[11] Alberto Crivellaro, Mahdi Rad, Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. Robust 3D object tracking from monocular images using stable parts. *TPAMI*, 2017. 3

[12] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing MVTec ITODD – A dataset for 3D object recognition in industry. *ICCVW*, 2017. 2, 3

[13] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. *CVPR*, 2010. 2, 7

[14] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 3, 5, 8

[15] Mingliang Fu and Weijia Zhou. DeepHMap++: Combined projection grouping and correspondence learning for full DoF pose estimation. *Sensors*, 2019. 2, 3

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 4

[17] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3D local feature descriptors. *IJCV*, 2016. 2

[18] H. Haber. Three-dimensional proper and improper rotation matrices. *University of California, Santa Cruz Physics 116A Lecture Notes*, 2011. 5

[19] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (DLS) method for PnP. *ICCV*, 2011. 8

[20] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. *ACCV*, 2012. 2, 6

[21] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. *ECCV*, 2016. 2

[22] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. *ECCVW*, 2018. 6

[23] Tomáš Hodaň, Eric Brachmann, Bertram Drost, Frank Michel, Martin Sundermeyer, Jiří Matas, and Carsten Rother. BOP Challenge 2019. `https://bop.felk.cvut.cz/challenges/bop-challenge-2019/`. 2, 3, 5, 6, 7

[24] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *WACV*, 2017. 2, 6, 7, 8

[25] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6D object pose estimation. *ECCVW*, 2016. 4

[26] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation. *ECCV*, 2018. 2, 3, 5, 6, 7

[27] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. *ICIP*, 2019. 3, 6

[28] Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3D pose estimation of texture-less objects in RGB-D images. *IROS*, 2015. 1, 2

[29] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D object pose estimation. *CVPR*, 2019. 2

[30] Peter J Huber. Robust estimation of a location parameter. 1992. 4

[31] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2012. 5

[32] Omid Hosseini Jafari, Siva Karthik Mustikovela, Karl Pertsch, Eric Brachmann, and Carsten Rother. iPose: Instance-aware 6D pose estimation of partly occluded objects. *ACCV*, 2018. 2, 7

[33] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. *ICCV*, 2017. 3

[34] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. *CVPR*, 2011. 5, 8

[35] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. *ICCV*, 2015. 2

[36] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 1, 2, 3, 5, 8

[37] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. *ECCV*, 2018. 3

[38] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. *ECCV*, 2018. 2, 6

[39] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation. *ICCV*, 2019. 2, 7

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014. 6

[41] David G Lowe et al. Object recognition from local scale-invariant features. *ICCV*, 1999. 1, 2

[42] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6D pose from visual data. *ICCV*, 2019. 3

[43] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6D pose refinement in RGB. *ECCV*, 2018. 2, 6

[44] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics*, 2006. 1

[45] Jorge J Moré. The Levenberg-Marquardt algorithm: Implementation and theory. Springer, 1978. 5, 8

[46] Apurv Nigam, Adrian Penate-Sanchez, and Lourdes Agapito. Detect globally, label locally: Learning accurate 6-DOF object pose estimation by joint segmentation and coordinate regression. *RAL*, 2018. 2, 3

[47] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3D object pose estimation. *ECCV*, 2018. 2, 3

[48] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. *ICCV*, 2019. 1, 2, 3, 7

[49] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-DoF object pose from semantic keypoints. *ICRA*, 2017. 2, 3

[50] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6DoF pose estimation. *CVPR*, 2019. 1, 2, 3

[51] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, and Vincent Lepetit. On object symmetries and 6D pose estimation from images. *3DV*, 2019. 3, 6

[52] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. *ICCV*, 2017. 1, 2, 3, 6

[53] Carolina Raposo and Joao P Barreto. Using 2 point+ normal sets for fast registration of point clouds with small overlap. *ICRA*, 2017. 7

[54] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 1, 2

[55] Pedro Rodrigues, Michel Antunes, Carolina Raposo, Pedro Marques, Fernando Fonseca, and Joao Barreto. Deep segmentation leverages geometric pose estimation in computer-aided total knee arthroplasty. *Healthcare Technology Letters*, 2019. 7

[56] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. *Third International Conference on 3-D Digital Imaging and Modeling*, 2001. 7

[57] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. *ECCV*, 2012. 6

[58] Juil Sock, Kwang In Kim, Caner Sahin, and Tae-Kyun Kim. Multi-task deep networks for depth-based 6D object pose and joint registration in crowd scenarios. *BMVC*, 2018. 3

[59] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. Augmented autoencoders: Implicit 3D orientation learning for 6D object detection. *IJCV*, 2019. 3, 7

[60] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3D object detection and pose estimation. *ECCV*, 2014. 2

[61] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. *CVPR*, 2018. 1, 2, 3

[62] Federico Tombari, Alessandro Franchi, and Luigi Di Stefano. BOLD features to detect texture-less objects. *ICCV*, 2013. 1

[63] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 2000. 8

[64] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 2002. 8

[65] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *CoRL*, 2018. 2

[66] Joel Vidal, Chyi-Yeu Lin, Xavier Lladó, and Robert Martí. A method for 6D pose estimation of free-form rigid objects using point pair features on range data. *Sensors*, 2018. 2, 7

[67] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. *CVPR*, 2019. 3

[68] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *RSS*, 2018. 2, 3, 6, 7, 8

[69] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D pose object detector and refiner. *ICCV*, 2019. 1, 2, 3, 6, 7