COPE: End-to-end trainable Constant Runtime Object Pose Estimation

Stefan Thalhammer
Automation and Control Institute, TU Vienna
Gusshausstrasse 27-29
`thalhammer@acin.tuwien.ac.at`

Timothy Patten
Robotics Institute, UTS
81 Broadway, Building 11
`timothy.patten@uts.edu.au`

Markus Vincze
Automation and Control Institute, TU Vienna
Gusshausstrasse 27-29
`vincze@acin.tuwien.ac.at`

Abstract

State-of-the-art object pose estimation handles multiple instances in a test image by using multi-model formulations: detection as a first stage and then separately trained networks per object for 2D-3D geometric correspondence prediction as a second stage. Poses are subsequently estimated using the Perspective-n-Points algorithm at runtime. Unfortunately, multi-model formulations are slow and do not scale well with the number of object instances involved. Recent approaches show that direct 6D object pose estimation is feasible when derived from the aforementioned geometric correspondences. We present an approach that learns an intermediate geometric representation of multiple objects to directly regress 6D poses of all instances in a test image. The inherent end-to-end trainability overcomes the requirement of separately processing individual object instances. By calculating the mutual Intersection-over-Unions, pose hypotheses are clustered into distinct instances, which achieves negligible runtime overhead with respect to the number of object instances. Results on multiple challenging standard datasets show that the pose estimation performance is superior to single-model state-of-the-art approaches despite being more than $\sim$35 times faster. We additionally provide an analysis showing real-time applicability ($>$ 24 fps) for images where more than 90 object instances are present. Further results show the advantage of supervising geometric correspondence-based object pose estimation with the 6D pose.

1. Introduction

Object pose estimation is a challenging problem for monocular computer vision despite being essential for many tasks such as augmented reality, object manipulation, scene
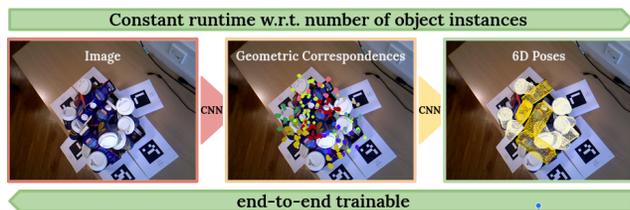

Figure 1. **COPE: Multi Object Instance 6D Pose Estimation with Constant Runtime.** Our end-to-end trainable pose estimator uses intermediate geometric correspondences to directly estimate 6D object poses from a single RGB image, thus processing multiple instances in parallel.

understanding, autonomous driving and industrial inspection [18, 35, 39]. Learning-based object pose estimation research focuses on maximizing the performance under challenging conditions like domain shift, object occlusion and object symmetries by tendentiously separating the detection from the pose correspondence estimation stage [11, 24, 27, 38, 45] then deriving the 6D pose with the Perspective-n-Points (PnP) algorithm [9] using the estimated geometric correspondences. This leads to shortcomings because a) adopting surrogate training targets decouples pose estimation from the training process and thus limits its learning [6] and b) running inference for multi-instance scenarios leads to a computational complexity of at least $\mathcal{O}(n)$ with respect to the number of objects (n) for the pose estimation stage. Thus, this type of approach has severely diminishing applicability for realistic scenarios.

Recent object pose estimation research trends recognize those shortcomings and partially alleviate them by directly regressing the 6D pose from the intermediate pose correspondences to achieve tremendous results [4, 6, 20, 50]. In [50] and [6], detection is separated from the pose estimation stage, which makes them not end-to-end trainable because they require an object detector. The work in [20]

is end-to-end trainable but separate networks need to be trained for each object and pooling geometric correspondences means multiple objects and instances cannot be handled simultaneously. We improve over these findings by proposing a natural extension to efficiently handle multi-object multi-instance scenarios.

In this work we propose a solution to the aforementioned shortcomings by sharing the latent representation as well as the direct pose regressor over objects and their instances; see Figure 1 for a high-level overview. We classify image locations in the feature maps, regress bounding box and view-dependent object geometry correspondences and regress the direct 6D pose. While the first three of these tasks are intermediate representations, the direct 6D pose head is an up-stream task shared over those intermediate outputs of the network. Consequently, the loss related to the 6D pose is also backpropagated to the down-stream task of geometric correspondence estimation. This design also allows further guidance of the learning process by enforcing consistency between these consecutive tasks, which additionally improves each of them. We propose a concurrent solution to anchors [40] for true location sampling during training that does not require manually choosing hyperparameters based on the expected test data distribution. True locations are sampled and regression targets are standardized using a scalar shape prior derived from the respective object mesh and the backpropagated loss is normalized for each object class. Thus, training is not biased towards larger objects and no prior assumptions need to be made in contrast to the case for anchors.

In summary, our contributions are:

- A simple and efficient solution for multi-object multi-instance object pose estimation that improves over the state of the art.

- A training target sampling scheme that requires no assumptions about the test data distribution.

Efficiently sharing internal representations over objects and instances enables end-to-end trainability that requires only one forward pass through the network to process all object instances in a single input image. We show that processing more than 90 object instances in a single image with more than $24fps$ on a modern consumer GPU, our method's performance is competitive to similar state-of-the-art approaches but up to 35 times faster.

The remainder of the paper discusses related work in Section 2, followed by a description of our proposed approach in Section 3 and evaluations in Section 4. Lastly, Section 5 concludes the paper.

2. Related Work

In this section we present the state of the art for monocular 6D object pose estimation with a focus on work that directly regress the object pose. This is followed by a review of training target sampling for object detection.

Object Pose Estimation Since direct pose regression from feature space leads to inferior performance, the dominant monocular object pose estimation approaches leverage geometric correspondences as regression targets [15, 27, 36, 38, 37, 21, 51]. Poses are derived for each estimated set of object correspondences using variants of PnP [9, 26, 46]. Recent trends replace the classical solver with trainable versions [4, 6, 20, 43, 50] to infer the 6D pose directly from the intermediate geometric correspondences. This enables end-to-end trainable object pose estimation as it provides the additional supervision for the down-stream network parts with the 6D pose. Their findings indicate that direct 6D pose estimation also results in state-of-the-art performance by sharing the pose regressor over objects [6, 50]. However, efficient and simultaneous single-stage multi-object instances handling is a problem that still remains [15].

Considering the top performing approaches in the BOP challenge [16], a benchmark that aims to provide a standardized protocol for an unbiased comparison of object pose estimation, a frequently used technique to handle multiple object instances in an image is to separate object detection from pose estimation [27, 24, 15, 32, 38, 11]. In the first stage, 2D location hypotheses are provided using common object detectors such as Faster-RCNN [40], RetinaNet [29] or FCOS [48]. In the second stage, object crops are passed to the pose estimator but this leads to considerable temporal and computational cost. An exception is EPOS [15] where multi-instance handling is facilitated by using Graph-Cut RANSAC [1] to cluster the predicted geometric correspondences to individual instances. Despite providing a sophisticated approach for addressing object symmetries and multiple object instances with one forward pass through a network, their multi-instance fitting of poses using [1] is computationally very demanding. In our work, we alleviate this issue by adopting ideas from object detectors and incorporate direct pose regression into the detection stage.

Detecting Objects in Images Single-stage object detectors offer efficient solutions for multi-object and multi-instance object localization in 2D [2, 8, 25, 33, 29, 48]. Anchors [29, 33, 40] are used to sample bounding box priors with different sizes and aspect ratios in the multi-scale feature map of feature pyramids [28]. For training, foreground image locations are chosen based on the Intersection-over-Union (IoU) between ground truth bounding boxes and the anchor boxes. As such, training locations are correlated with the projected object shape in the image space. This leads to effective handling of objects with different scales.

Using anchors has two downsides. Firstly, it requires the manual specification of 16 hyperparameters that reflect the expected training and test data statistics. Secondly, the size of the output space depends on the number of anchors sam-

pled for each image location. Recent approaches propose alternative formulations to circumvent these shortcomings while retaining the advantages of anchors [8, 25, 48, 53]. The authors of [48] choose the respective feature map resolution for training explicitly by using the bounding box size to overcome the necessity of sampling anchor boxes. True image locations of the feature map, for loss backpropagation, are assigned based on the respective pixel's centerness with respect to the ground truth bounding box. Alternatively, [25] models objects as paired-keypoints: the top-left and the bottom-right corner of the bounding box. Similarly, [8] addresses the inefficiency of anchor-based object detection by modeling objects as their center points and estimating the bounding box relative to it. The authors of [53] overcome the requirement of hyperparameters for assigning objects to anchors by designing a flexible maximum likelihood estimation assignment for network training.

We propose to sample training locations based on the visible object mask and the 3D object dimensions. We also use the 3D object dimension to effectively replace the anchor-based target annotation standardization. As such, we encode objects of different sizes and eccentricities more effectively, while also reducing the size of the output space and the number of required hyperparameters.

3. Constant Runtime Object Pose Estimation

This section describes our direct 6D **C**onstant Runtime **O**bject **P**ose **E**stimation approach, abbreviated as COPE. We start with a high-level overview of the method. Afterwards, we detail the approaches for deriving anchor-free training targets, true image location sampling and geometry correspondence standardization during training. This is followed by an explanation of the direct 6D pose parameterization and symmetry handling. We conclude with a description of multi-instance clustering and hypotheses filtering during testing.

3.1. Constant Runtime via Direct-pose regression

Our aim is to classify and estimate the poses of all object instances in a single RGB input image. The 6D pose is defined as $\hat{P} \in SE(3)$, which represents the object's rotation $R \in \mathbb{R}^3$ and translation $t \in \mathbb{R}^3$ with respect to the camera's coordinate frame. Object meshes are known in advance but no additional information regarding the test scene is required. We define the corner points of the smallest cuboid enclosing the respective object mesh in its coordinate frame as the geometric correspondences (G_{3D}). COPE, outlined in Figure 2, outputs the set of object instances visible in the image, parameterized by object type and 6D pose.

COPE builds upon the success of recent efficient object detection approaches [29, 48, 52]. The RGB input image is first processed with a CNN backbone and then multi-scale features are computed using a feature pyramid to esti-

mate the intermediate object representation. Three modules shared over feature maps of sizes [$s/8$, $s/16$, $s/32$], with s being the input image resolution, generate the intermediate outputs $\hat{O}$, $\hat{B}$ and $\hat{G}$. The first module predicts the set of object class probabilities $\hat{O} := \{ \hat{o}_0, ..., \hat{o}_k \}$, where k is the number of image locations in the multi-scale feature map and $\hat{o}_k \in \mathbb{R}^a$ is the Bernoulli distributed object class prediction. We denote the number of object classes in the dataset with a. The second module predicts the amodal bounding boxes $\hat{B} := \{ \hat{b}_0, ..., \hat{b}_k \}$, where $\hat{b}_k \in \mathbb{R}^4$. The third module predicts the projection of G_{3D} in the image space $\hat{G} := \{ \hat{g}_0, ..., \hat{g}_k \}$, where $\hat{g}_k \in \mathbb{R}^{16}$ represents the 2D coordinates of the 8 corner points of the smallest cuboid enclosing the respective object mesh. A shared direct pose module slides over the set $\hat{G}$, directly estimating pose hypotheses $\hat{P} := \{ \hat{p}_0, ..., \hat{p}_k \}$, where $\hat{p}_k \in \mathbb{R}^9$. The pose output is parameterized by 3 values for translation in $\mathbb{R}^3$ and 6 values for rotation, the first two basis vectors of the rotation matrix in $\mathbb{R}^3$ [54]. A set $\hat{C} := \{ \hat{c}_0, ..., \hat{c}_k \}$ is computed to quantify the consistency $\hat{c}_k \in \mathbb{R}^1$ between $\hat{g}_k$ and $proj_{3D \to 2D}(G_{3D} \cdot \hat{p}_k)$ for each image location separately.

During inference, the network predicts $H = \{\hat{O}, \hat{B}, \hat{G}, \hat{P}, \hat{C}\}$ with constant runtime for a query image. Corresponding elements of H with an image location k of maximum class probability $\hat{o}_k$ below the detection threshold are discarded. The resulting subsets are clustered into object instances using the IoU between elements of $\hat{B}$. The hyperparameter n represents the highest number of consistencies in $\hat{C}$ and is set to 10 for the presented experiments; see the ablation in Table 3. Finally, the detected object classes and the mean of the n poses with the highest consistencies per instance are returned with negligible increase in runtime with respect to the number of object instances. Through this procedure, our method estimates the poses of a large number of object instances in a single test image in real-time ($> 24 fps$) on an Nvidia Geforce 3090 GPU.

3.2. Training Target Sampling

Effective assignment of true image locations for updating network weights during training is an ongoing research problem [8, 25, 29, 48, 52, 53]. These true image locations are often sampled in the output feature maps of feature pyramids [28], which is a great tool to efficiently encode scale information in the feature space. Anchors [40] are the standard representation for providing bounding box priors to sample true image location based on the IoU with the ground truth during training [40, 33, 29]. For each image location in the multi-scale feature map, 9 differently shaped and sized bounding box priors are sampled. This requires 16 hyperparameters: 5 each for *base sizes* and *strides* and 3 each for *ratios* and *scales* [29]. This results in two convenient traits since anchor locations used for updating the network's weights are chosen based on a threshold parame-
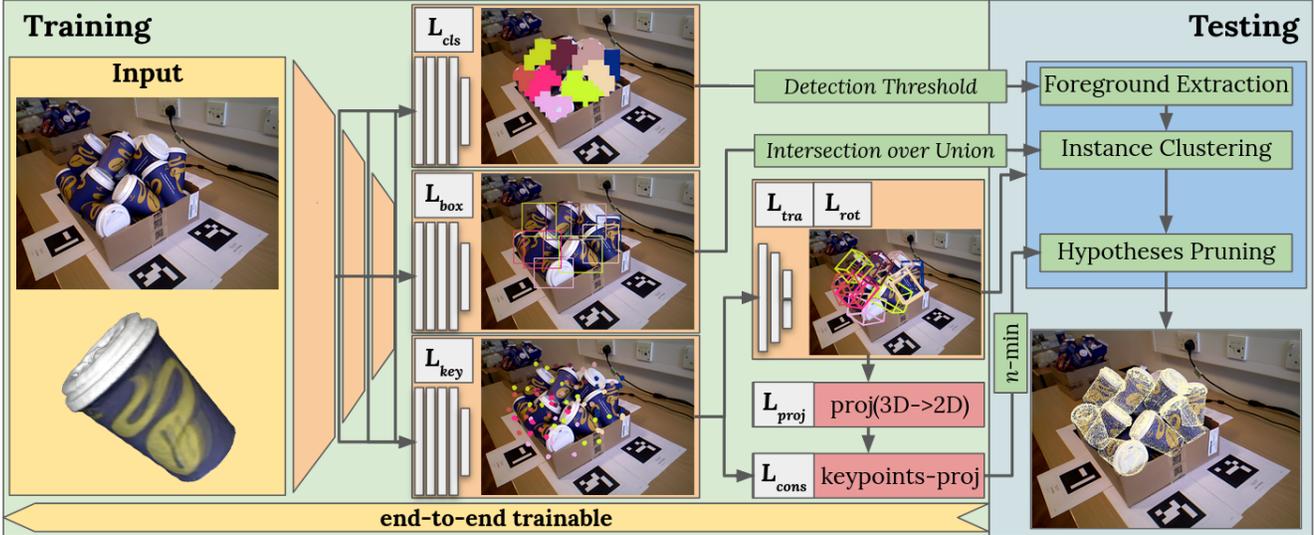
Figure 2. **Constant Runtime Object Pose Estimation.** Given an input image and a 3D model, image locations are classified while bounding boxes and geometric correspondences are regressed. A direct pose regression module slides over the image locations and regresses the 6D pose from the geometric correspondences. Training is supervised with losses for each module (L_{cls}, L_{box}, L_{key}, L_{tra} and L_{rot}) as well as auxiliary losses (L_{proj} and L_{cons}) to enforce consistency between estimated correspondences and direct poses. During testing, instances are efficiently clustered using their 2D IoU then the n hypotheses with the highest consistency generate the 6D output.

ter for the IoU with the ground truth:

- Sampling anchors leads to a uniform scale space for the expected bounding boxes. As such, a similar amount of training locations are sampled per object, independent of the object's size in the image space.

- Regression targets are standardized using the respective anchor's center, width and height. This means that the regression target space has similar statistics for differently sized objects.

Despite these convenient traits, training target sampling is cumbersome since anchors a) require choosing 16 hyperparameters depending on the expected object scales in image space and b) slow down convergence due to the large output space of 9 anchors per feature map location. We overcome these shortcomings by using a regression target standardization scheme that reflects the object's geometry and scale.

3.2.1 True Location Sampling

Object masks are used for true training location sampling as in [21]. However, instead of predicting object masks and correspondences from a single feature map resolution, our work adopts the divide-and-conquer strategy of feature pyramids to make predictions from multiple feature map resolutions. To overcome the necessity of requiring hyperparameters [40, 48] for choosing the best suited feature map resolution for locating an object, we propose a geometry-based approach to assign true training locations. We sup-

plement true location sampling with a scalar shape prior:

$$\delta_o = max||(m_i - m_j)||_2 \ \forall \ m_i, m_j \in M, i \neq j \quad (1)$$

where M is the set of object model vertices. Since the spatial downsampling of the input image through the backbone follows an exponential function, it is intuitive to explicitly choose pyramid levels using a logarithmic function. As such, we choose the respective feature pyramid level with:

$$level = f + log_d(\delta_o/t_z), \quad (2)$$

where f depends on the number of pyramid levels used, t_z corresponds to the object's distance from the camera and d is the only remaining hyperparameter. Since we use three pyramid levels, as in [47], this requires choosing only 6 hyperparameters for FCOS and 12 when using anchors. An additional advantage is that δ_o better reflects the object shape in all three spatial dimensions and thus also the visible object surface in the image space compared to using the bounding boxes for the assignment of true training locations. As a consequence, elongated objects are tendentiously sampled in higher resolved feature pyramid levels than boxy shaped objects. Despite needing fewer hyperparameters, we retain a similar amount of true locations used for training. Classifying true image locations (L_{cls}) is supervised using the focal loss [29].

3.2.2 Geometric Correspondence Standardization

Instead of standardizing the projected object correspondences G using anchor priors or with a scalar value agnostic

to object shape [48], we directly incorporate δ_o to scale regression targets of different objects to a similar magnitude:

$$y_G = (c - G)/\delta_o, \qquad (3)$$

where c is the center of the respective feature map location, G are the image locations of the geometric correspondences and y_G are the standardized regression targets. As such, regression targets are encoded similarly as with anchors (with similar σ for G for all objects independent of their scale or shape eccentricity). Thus, the computed error is independent of the object's scale in the image space and the training process is not biased for larger objects. Our approach needs no hyperparameters for standardization and convergence is improved since 9 times fewer network output parameters per feature map location are required compared to anchors.

3.2.3   Imbalance Problem of Target Locations

Choosing training target locations based on the object mask leads to a training process that is biased towards objects with a larger projected image surface. For classification this is commonly circumvented using the focal loss [40]. Using anchors as location priors alleviates the issue since anchors are sampled uniformly over the expected object scale space. We define a concurrent solution by normalizing over the number of true training locations l and accumulating the gradient afterwards. The regression loss is:

$$L_{reg}(\hat{y}, y) = \frac{1}{a} \cdot \sum_{i=0}^{a} \frac{1}{l_i} \cdot \sum_{j=0}^{l_i} huber(\hat{y}_j, y_j), \qquad (4)$$

where $huber$ is the augmented l_1 loss used in RetinaNet [10, 29] and y and $\hat{y}$ are the ground truth and estimate, respectively. This procedure requires no additional trainable parameters and only leads to minor computational overhead during training time and to none during test time despite improving multi-object handling.

3.2.4   Direct Pose Regression

The direct pose is regressed using the output $\hat{y}$ of the module that estimates intermediate geometric correspondences as in [6, 20, 43, 50]. The 6D pose is parameterized as $P \in SE(3)$, with $t \in \mathbb{R}^3$ being the 3D translation vector and $R \in \mathbb{R}^6$ the first two base vectors of the $SO(3)$ rotation matrix as in [6, 23, 50]. These methods perform pose estimation on zoomed crops of the detected objects of interest. They predict the rotation of the camera in the object coordinate frame, i.e. the egocentric rotation, since predicting the rotation in the camera coordinate system, i.e. the allocentric rotation, results in ambiguities due to the cropping [23].

In contrast, we learn to predict geometric correspondences directly in the image space. These correspondences

are destandardized with the inversion of Equation (3) and fed to the direct pose estimation module. As such, our approach correlates object rotation with its image location. This means that we are able to directly regress the allocentric rotation since we require no zooming or cropping. Additionally, we can directly regress the 3D translation without requiring a scale-invariant translation representation as used in [6, 27, 50]. The network training is supervised using the image locations sampled with Equation (2).

3.3. Symmetry-aware Loss

Objects exhibiting discrete or continuous symmetries, i.e. similar views that correspond to different ground truth poses P, are detrimental to the convergence of the network training [34, 38, 42]. We adopt the transformer loss of [38] since symmetries are efficiently handled during loss computation and require no additional trainable weights. We define our keypoint estimation loss for supervising the training of the geometric correspondence learning with:

$$L_{key} = \min_{s \in S_i} L_{reg}(\hat{y}, hy), \qquad (5)$$

where S_i is a set of symmetry transformations that depend on the visual ambiguities of the object. We observed that separately choosing hypotheses with L_{key} as well as substituting the direct pose losses for L_{rot} and L_{tra} with Equation (5) introduces ambiguities since the 6D pose is directly derived from the estimated intermediate geometric correspondences. To alleviate this issue we define an indicator function $\mathbb{I}$, indicating the symmetry that minimizes L_{key}. As such, we supervise the direct pose regression with:

$$L_{rot/tra} = L_{reg}(\hat{y}, \mathbb{I}(S)y). \qquad (6)$$

Since only one set of $\hat{G}$ is predicted per image location and Equations (5) and (6) sufficiently account for object symmetries, L_{proj} and L_{cons} can be directly computed with L_{key}. The projection and the consistency loss are thus defined as:

$$L_{proj} = L_{reg}(G_{3D}\hat{P}, G), \qquad (7)$$

$$L_{cons} = L_{reg}(G_{3D}\hat{P}, \hat{G}). \qquad (8)$$

The overall loss is:

$$L = \alpha \cdot L_{cls} + \beta \cdot L_{box} + \gamma \cdot L_{key} + \delta \cdot L_{rot}$$
$$+ \epsilon \cdot L_{tra} + \zeta \cdot L_{proj} + \eta \cdot L_{cons}, \quad (9)$$

where α, β, γ, δ, ϵ, ζ and η are loss weights. The bounding box estimation, L_{box}, is supervised using Equation (4).

3.4. Multi-instance Handling

Commonly, multiple instances of the same object in a single image are handled before correspondence estimation by non-maximum suppression of the detection

stage [27, 49, 24, 38] or by clustering correspondences afterwards [15]. The first family of methods individually process each instance's image crop to estimate the 6D pose. The second family of methods is more advantageous because the network is shared over all objects of interest. Unfortunately, since [15] predicts dense geometric correspondences the method has a high runtime. This is due to the clustering of correspondences to object instances using [1], which is computationally demanding.

In our work, H is first filtered by discarding the non-maximally scoring object classes for each image location k. Subsequently, image locations with a detection score below the detection threshold are pruned. The remaining hypotheses correspond to detected objects. The 2D bounding boxes, $\hat{B}$, are used to cluster object instances based on the respective IoU between the outputs of different image locations. Ultimately, using the computed consistency $\hat{C}$, the pose is averaged over the n hypotheses of $\hat{P}$ with the highest consistency for each object instance.

4. Experiments

This section provides quantitative and qualitative evaluations of COPE on several datasets. After introducing the experimental setup, we proceed with comparisons to the state of the art on two challenging datasets using the BOP protocol [16]. In addition, ablation studies are presented to quantify the influence of direct pose supervision on an additional dataset. To further validate and thoroughly test the capabilities of our method, we present results on a synthetic dataset with up to 100 object instances per image.

4.1. Datasets

Evaluation is provided on three standard datasets: LM [13], LM-O [3] and IC-BIN [7]. For evaluation, we use the subsets provided with the BOP challenge. LM provides 200 test images for each of the 13 objects that come with watertight object models. LM constitutes a common benchmark for object pose estimation in cluttered environments. LM-O consists of 200 test images of LM's second test sequence with all eight objects annotated in each. LM-O provides test images with challenging object occlusions. IC-BIN presents 150 test images of up to 21 instances of two objects with heavy occlusion. COPE is trained on 50k images generated with physically based rendering [5, 17] Results on LM and LM-O are provided with the same models trained on all 13 objects of LM.

4.2. Evaluation Metrics

Comparison to the state of the art is provided using the performance score of the BOP challenge [16]. Results for pose estimation are reported using the Average Recall: $AR = (AR_{VSD} + AR_{MSSD} + AR_{MSPD})/3$. Ablations are evaluated using the ADD recall, or ADD-S recall for

objects exhibiting symmetries [13]. We report the fraction of poses below the commonly used error threshold of 10% of the object diameter. Results for object detection are reported using the the mean Average Precision (mAP) of the Microsoft COCO object detection challenge [30]. The results are those for the IoU values from 0.5 to 0.95 in 0.05 steps. Please refer to the supplementary material for details.

4.3. Implementation Details

The weights of the backbone are pre-trained on ImageNet [41] and fine-tuned for 120 epochs using the Adam [22] optimizer with a learning rate of 10^{-5} and a batch size of 8. Previous work suggests overcoming the domain gap between training on synthetic and testing on real images by not updating certain network weights during optimization [14, 51]. Similarly, we do not update parameters of batch normalizations and the convolutions of the first two stages of the backbone during fine-tuning. We also apply image augmentations as described in [47]. The parameter d in Equation (2) is set to 3 for all experiments.

4.4. Comparison to the State of the Art

Object Pose Estimation We compare the performance of COPE to the state of the art on IC-BIN and LM-O. Results using the BOP setting reporting the *AR* are provided in Table 1. The bottom section compares single-model methods, i.e approaches that produce estimates for all object classes and their instances in a single forward pass. Both DPOD [51] and EPOS [15] require PnP for deriving the 6D pose from the predicted geometric correspondences, while COPE directly outputs the 6D pose. COPE improves over both methods in *AR* on average. Compared to the previous single-model state of the art, EPOS, COPE achieves similar *AR* on LM-O 0.543 as compared to 0.547 but improves to 0.440 in comparison to 0.363 on IC-BIN. More remarkable, however, is that the runtime of COPE is 37 times faster using the inference speed calculated by the BOP toolkit[1].

The top section of Table 1 presents the results of multi-model methods. These multi-model methods use an object detector to sample sparse location priors then separately trained networks per object class for correspondence prediction and pose estimation. For the methods [6, 19, 31, 44] no results on IC-BIN are available. Compared to the best individually performing methods on both datasets, Cosy-Pose on IC-BIN and ZebraPose on LM-O, COPE results in $\sim 24\%$ relative performance decrease. This is in the expected range due to the known performance decrease for single-staged approaches [29].

Runtime Figure 3 presents the average runtime and standard deviation on IC-BIN for five test runs of COPE, CDPNv2 and CosyPose on an Intel CPU with 3.6GHz and an Nvidia Geforce 3090 GPU. The times reported for
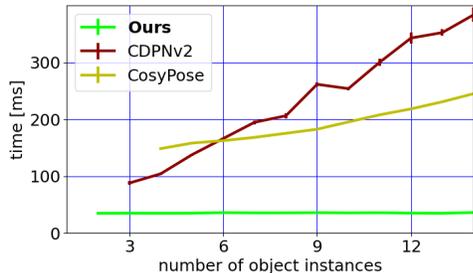
---

[1]https://github.com/thodan/bop_toolkit

Figure 3. **Runtime Comparison to the State of the Art on IC-BIN.** Provided are the times it takes to estimate poses for all object instances in a single image.

Table 1. **Comparison to the State of the Art for Pose Estimation.** Presented are the *Average Recall* on IC-BIN and LM-O, the average over both and the inference speed using the BOP toolkit.

Method	IC-BIN	LM-O	Avg.	Time
Multi-model				
AAE [45]	0.217	0.146	0.182	**0.199**
Pix2Pose [38]	0.226	0.363	0.295	1.230
2Dto3D [32]	0.342	0.525	0.434	0.546
CDPNv2 [27]	0.473	0.624	0.549	1.010
SurfEmb [11]	0.550	0.623	0.587	6.296
CosyPose [24]	**0.574**	0.618	**0.596**	0.227
SO-Pose [6]	-	0.613	-	-
CIR [31]	-	0.655	-	-
PFA [19]	-	0.683	-	-
ZebraPose [44]	-	**0.718**	-	0.250
Single-model				
DPOD [51]	0.169	0.130	0.150	0.211
EPOS [15]	0.363	**0.547**	0.455	2.804
Ours	**0.440**	0.543	**0.492**	**0.075**

CDPNv2 exclude the time required for detecting objects. Despite omitting the runtime of CDPNv2's first stage, our method is more than 12 times faster and 7 times faster than CosyPose when processing 15 object instances. Most notably, in contrast to multi-model approaches, COPE is capable of directly providing 6D poses for multi-object multi-instance cases at almost constant runtime, which makes it highly suitable for real-time scenarios.

Object Detection Table 2 compares the object detection accuracy (*mAP* [30]) of COPE to the state of the art on IC-BIN and LM-O using the same training data. On average, COPE outperforms both MaskRCNN [12] and FCOS [48], achieving the highest average *mAP* over both datasets. On LM-O, COPE is superior to MaskRCNN, achieving 0.532 as compared to 0.375 but slightly inferior to FCOS that reaches 0.622. This is partly due to the lean network design of COPE. Please refer to the supplementary material for more details. COPE's detected bounding boxes are more precise than both standard detectors used by many multi-

Table 2. **Comparison to the State of the Art for Object Detection.** Presented are the *Average Recall* on IC-BIN and LM-O and the average using the BOP toolkit.

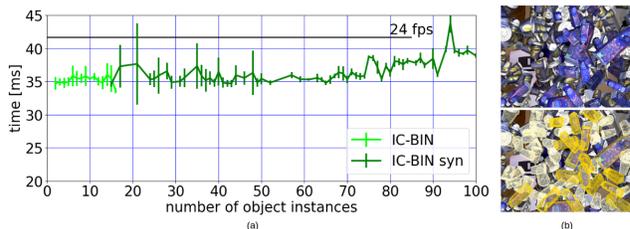| Method	IC-BIN	LM-O	Avg.
MaskRCNN [12, 24, 11]	0.316	0.375	0.346
FCOS [48, 27, 44]	0.323	**0.622**	0.473
Ours	**0.431**	0.532	**0.482**

Figure 4. **Runtime Evaluation of COPE on IC-BIN and IC-BIN syn.** (a) COPE provides negligible runtime increase up to 70 object instances per image. The black line indicates the threshold for real-time processing. (b) An example of a test image of *IC-BIN syn* and a visualization of the estimated poses.

model methods on IC-BIN, achieving 0.431 as compared to 0.323 and 0.316. As such our method provides excellent location priors for pose refinement.

4.5. Ablation Studies

Runtime Evaluation In order to exhaustively test the runtime and scalability of COPE, we create a synthetic test dataset using the IC-BIN objects and OpenGL[2] rendering, named *IC-BIN syn*. The number of object instances rendered per image is sampled from a uniform distribution with a lower bound of 10 and upper bound of 100. We render the sampled number of object instances randomly from the IC-BIN objects onto the test images of IC-BIN. Results are again provided for processing one test image on an Intel CPU with 3.6GHz and an Nvidia Geforce 3090 GPU.

Figure 4 (a) presents the runtime of our method for detecting and estimating the poses of up to 100 object instances in a single image. We report the average runtime and standard deviation for five test runs. The runtime increases negligibly up to 70 detected object instances. For more than 90 instances per image, our method exceeds real-time processing. As such, it provides quantitative proof of the tremendous scalability of the presented approach and the constancy of the runtime with respect to the number of object instances in a single test image. Figure 4 (b) provides a rendered synthetic test image (top) and projected object models based on the estimated poses (bottom).

Direct-pose Regression Table 3 displays the influence of direct pose regression on the end-to-end architecture on the LM [13] and LM-O [3] datasets using the *ADD/(-S)* re-

---

[2]https://github.com/thodan/bop_renderer

Table 3. **Ablation Study for Pose Supervision.** Provided is the average *ADD/(-S)* recall. The objects *Eggbox* and *Glue* are considered as symmetric objects.

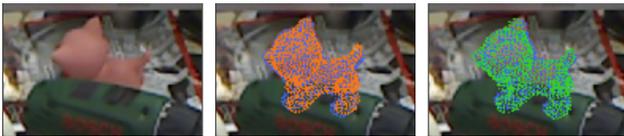| Supervision	Voting		LM	LM-O
IM	2D	PnP	0.654	0.280
DR	2D	PnP	0.712	0.330
	6D	all	0.715	0.342
DR-P	2D	PnP	0.672	0.341
	6D	all	0.672	0.345
DR-PC	2D	PnP	0.712	0.348
	6D	n=1	0.722	0.338
	6D	n=5	0.724	0.346
	6D	n=10	0.732	**0.350**
	6D	all	**0.738**	0.349

Figure 5. **Pose Supervision Comparison on LM-O's *Cat*.** From left to right: raw image, pose obtained from geometric correspondences and RANSAC-EP*n*P, and direct pose regression. Blue, red and green meshes indicate ground truth, false positive and true positive pose (as measured by *ADD*).
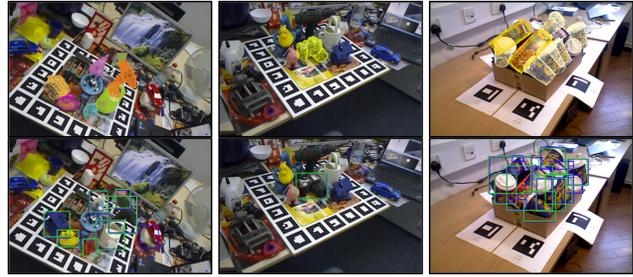

Figure 6. **Qualitative results on LM-O, LM and IC-BIN.** Top row shows reprojected object meshes based on the estimated poses, bottom row shows bounding box estimates. Blue boxes indicate the ground truth while green boxes indicate estimates.

call. The column *Voting* indicates the pose voting procedure using RANSAC-EP*n*P, an average of all direct pose hypotheses, or an average of the direct pose estimates with the best *n* hypotheses in terms of $\hat{C}$.

The results show that supervising the training process with direct pose regression (**DR**) improves the quality of the intermediate representation (**IM**) tremendously. The improvement is from 0.654 to 0.715 on LM and from 0.280 to 0.342 on LM-O. Using **DR** direct pose estimates is superior to using those estimated by RANSAC-EP*n*P. Providing additional guidance with L_{proj} (**DR-P**) improves for the occluded scenario of LM-O but is detrimental for LM. Ultimately, enforcing consistency between the internal representation and the correspondences projected to 2D using the regressed 6D pose with L_{cons} (**DR-PC**) leads to good results for direct regression and when using the intermediate representation on both datasets. Figure 5 shows an example of LM-O's cat under occlusion. The re-projected model using the ground truth is colored blue and the wrong estimate based on *ADD*, using the intermediate representation and RANSAC-EP*n*P, is colored red (middle image). Direct pose regression recovers from the incorrect intermediate representation, which is displayed in green (right image).

4.6. Qualitative Evaluation

Figure 6 visualizes results on LM-O, LM and IC-BIN. Displayed are projected object meshes based on the estimated pose in the top row and estimated bounding boxes in comparison to the respective ground truth in the bottom row. Green and red bounding boxes portray estimates and ground truth, respectively. The left image pair indicates a common error for LM-O: a false negative detection of the *Eggbox*. The right image pair shows that some of IC-BIN's instances of *Juice* are difficult to detect while detecting *Coffeecup* works well even under heavy occlusion if more than the lid is visible.

5. Conclusion

This paper presented a framework for pose estimation that processes up to 70 object instances in a single image with effectively constant runtime and up to 90 in real-time (> 24 fps). Our end-to-end trainable, single-staged approach achieves up to 35 times faster runtime than state-of-the-art approaches with similar formulation while also generating similar pose estimation accuracy. Directly regressing the 6D pose from sparse intermediate geometric correspondences enables efficient network scaling with respect to the amount of object classes and instances. During test time, multiple instances are handled based on their 2D overlaps, which results in negligible runtime increase with respect to the number of instances. As such, we have developed an object pose estimator that is applicable and useful for a broad variety of real-time tasks. In the future, we plan to improve our work by learning the intermediate representation in a self-supervised manner in order to overcome the necessity of defining geometric correspondences.

References

[1] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6733–6741, 2018.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. In *Proceedings of the European Conference on Computer Vision*, pages 536–551, 2014.

[4] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backprop-agating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8100–8109, 2020.

[5] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad El-badrawy, Ahsan Lodhi, and Harinandan Katam. Blender-proc. *CoRR*, abs/1911.01911, 2019.

[6] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021.

[7] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3583–3592, 2016.

[8] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qing-ming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[11] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6749–6758, 2022.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2961–2969, 2017.

[13] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *Proceedings of the Asian Conference on Computer Vision*, pages 548–562, 2012.

[14] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 0–0, 2018.

[15] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11703–11712, 2020.

[16] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *Proceedings of the European Conference on Computer Vision Workshops*, 2020.

[17] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. In *2019 IEEE international conference on image processing*, pages 66–70. IEEE, 2019.

[18] Tingbo Hou, Adel Ahmadyan, Liangkai Zhang, Jianing Wei, and Matthias Grundmann. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. *arXiv preprint arXiv:2003.03522*, 2020.

[19] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6d object pose estimation. *arXiv preprint arXiv:2203.09836*, 2022.

[20] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2930–2939, 2020.

[21] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[23] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3559–3568, 2018.

[24] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision*, pages 734–750, 2018.

[26] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[27] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time

rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019.

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2980–2988, 2017.

[30] 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. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[31] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6728–6737, 2022.

[32] Jinhui Liu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, Errui Ding, Feng Xu, and Xin Yu. Leaping from 2d detection to efficient 6dof object pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 707–714. Springer, 2020.

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European conference on computer vision*, pages 21–37. Springer, 2016.

[34] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6841–6850, 2019.

[35] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[36] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 119–134, 2018.

[37] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10710–10719, 2020.

[38] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Oct 2019.

[39] Timothy Patten, Kiru Park, and Markus Vincze. Dgcm-net: dense geometrical correspondence matching network for incremental experience-based robotic grasping. *Frontiers in Robotics and AI*, page 120, 2020.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[42] Yifei Shi, Junwen Huang, Xin Xu, Yifan Zhang, and Kai Xu. Stablepose: Learning 6d object poses from geometrically stable patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15222–15231, 2021.

[43] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

[44] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022.

[45] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision*, pages 699–715, 2018.

[46] George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *Proceedings of the European Conference on Computer Vision*, pages 478–494. Springer, 2020.

[47] Stefan Thalhammer, Markus Leitner, Timothy Patten, and Markus Vincze. Pyrapose: Feature pyramids for fast and accurate object pose estimation under domain shift. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021.

[48] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[49] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 108–125. Springer, 2020.

[50] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[51] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of*

the IEEE/CVF international conference on computer vision, pages 1941–1950, 2019.

[52] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[53] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *Neural Information Processing Systems*, 2019.

[54] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.