# Out-of-Region Keypoint Localization for 6D Pose Estimation

Xin Zhang, Zhiguo Jiang, and Haopeng Zhang*

*Image Processing Center, School of Astronautics, Beihang University, Beijing, 102206, China*

*Beijing Key Laboratory of Digital Media, Beijing, 102206, China*

*Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies, Ministry of Education, Beijing, 102206, China*

## Abstract

This paper addresses the problem of instance level 6D pose estimation from a single RGB image. Our approach simultaneously detects objects and recovers poses by predicting the 2D image locations of the object's 3D bounding box vertices. Specifically, we focus on the challenge of locating virtual keypoints outside the object region proposals, and propose a boundary-based keypoint representation which incorporates classification and regression schemes to reduce output space. Moreover, our method predicts localization confidences and alleviates the influence of difficult keypoints by a voting process. We implement proposed method based on 2D detection pipeline, meanwhile bridge the feature gap between detection and pose estimation. Our network has real-time processing capability, which runs 30 fps on a GTX 1080Ti GPU. For single object and multiple objects pose estimation on two benchmark datasets, our approach achieves competitive or superior performance compared with state-of-the-art RGB based pose estimation methods.

*Keywords:* 6D pose estimation, keypoint representation, localization confidence, real-time processing

## 1. Introduction

6D relative pose estimation between object and camera is a classical problem in computer vision, but has recently attracted intensive attention. Effective acquisition of an object's position and orientation is critical for accomplishing

☆THIS IS A PREPRINT VERSION FOR PERSONAL USE ONLY. Published version can be accessed via ScienceDirect at https://doi.org/10.1016/j.imavis.2019.103854.

*Corresponding author

*Email address:* zhang_xin_by@buaa.edu.cn, jiangzg@buaa.edu.cn, and zhanghaopeng@buaa.edu.cn (Xin Zhang, Zhiguo Jiang, and Haopeng Zhang)

various higher level vision tasks such as augmented reality, autonomous driving and robotics. Although methods [1, 2, 3, 4] using RGB-D data have achieved high pose accuracy, a large number of researchers are engaged in RGB-based 6D pose estimation for better efficiency and usability. According to the scope of objects of interest, prevailing pose estimation methods can be classified into category level methods [5, 6, 7, 8, 9] and instance level methods [10, 11, 12, 13, 14, 15]. The former class concentrates on handling intra-category variation to coarsely determine relative orientations for an entire category. The later class aims at achieving high-accuracy pose estimation for several particular objects, which is what we do in this paper.

Traditional pose estimation methods [16, 17, 18] are typically limited to objects with rich texture, while recent deep learning based methods [1, 2, 10, 11, 12, 14, 15, 19] have boosted the performance on poorly textured objects. Among the various strategies proposed to employ convolutional neural networks (CNN) to estimate 6D poses, one popular way is to establish 2D-3D correspondences by predicting 2D projection locations of objects' 3D bounding box corners. This kind of approaches [10, 12, 13, 15, 19] train neural networks to detect keypoints instead of depending on hand-crafted features. As an instance level task, pose estimation used to heavily rely on object regions provided by 2D detectors. [11, 12, 13] validate that pose estimation can be effectively integrated into 2D detection frameworks in a multi-task learning manner. Recent state-of-the-art methods [14, 15] leverage segmentation supervision to locate objects, and yield pixel-wise dense predictions for pose hypotheses.

Our approach follows the paradigm of keypoint-based methods, and perform 2D detection and pose estimation simultaneously. Specifically, we focus on the challenge of accurately locating keypoints outside the object region proposals. Many keypoint-related tasks such as human pose estimation may encounter this problem due to inaccurate detection results. For purpose of 6D pose estimation, we need to locate virtual control points instead of appearance feature points, which are more likely to lie outside the 2D bounding boxes as shown in Fig.1. Besides, some virtual vertices are naturally more difficult than others. They are typically occluded or lie on the background, thus lacking discriminative local features. It is challenging to locate these difficult vertices confidently from RGB images only, and may consequently hinder the pose accuracy. However, as far as we know these two issues have hardly been explicitly considered in keypoint-based pose estimation methods.

To address the above problems, we propose a novel keypoint representation based on region boundaries and integrate classification and regression schemes. As illustrated in Fig. 1b, we assign a keypoint to one of four subspaces according to the distances from the corners, and regress the offsets from the nearest two boundaries. In contrast to heatmap based representation, our approach can apply to keypoints both within and outside the region proposals. Compared with similar keypoint-based methods [12, 13], our proposed representation has a smaller output space by reducing the length and variance of regression targets, which is conductive to stable training and achieving robust localization. Our representation can also predict localization confidences, which are used to refine the keypoints during non-maximum suppression (NMS) for pose estimation. Whereas PVNet [14] acquire confidence scores through RANSAC [20] based voting scheme, which takes up a dominant portion of the entire runtime. To implement our method, we develop an end-to-end trainable network extending 2D detection pipeline with elaborate design for pose estimation. We introduce feature transition module and feature fusion module to bridge the gap between detection and pose estimation, meanwhile maintain high resolution representations. Then we extract refined regional features for locating keypoints, and adopt EPnP algorithm [21] to calculate poses. Since [14] and [15] both perform class-based segmentation to locate objects, they may have trouble handling multi-instance clustered scenes. In contrast, we use less supervision and achieve comparable pose accuracy. Our approach offers an inference speed of about 30 fps on a GTX 1080Ti GPU, which is faster than [14] and [15].

We conduct comprehensive experiments on LINEMOD dataset [22] and OCCLUSION dataset [23]. The results verify that our approach achieves the best pose accuracy among the compared methods that do not use segmentation supervision, and even competes with state-of-the-art methods [14, 15].

In summary, the main contributions of our work are three-fold:

- We propose a novel region boundary based keypoint representation for locating 3D vertices in the context of 6D pose estimation, which reduces output space by integrating classification and regression schemes. The proposed representation not only applies to out-of-region keypoints but also predicts localization confidences.

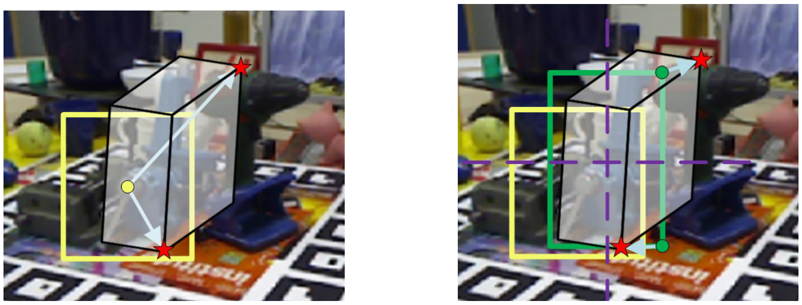
- We develop an efficient two-stage detection-driven architecture for 6D pose estimation, which introduces feature transition and fusion module to close the gap between 2D detection and pose estimation.

- Our approach outperforms the compared detection-driven RGB based 6D pose estimation methods on two common benchmarks, and competes with SOTA segmentation-driven methods [14, 15].

The rest of this paper is organized as follows. We review related works in Section 2, and then detail each component of our method in Section 3. We present ablation experiments and comparison with the state-of-the-art methods in Section 4. Finally conclusions are summarized in Section 5.

## 2. Related Work

The vast majority of instance level 6D pose estimation methods assume calibrated cameras and available 3D models. The main difference in the input data is whether or not depth data is included. It has been validated that depth information is critical for both pose estimation and pose refinement. However, acquiring depth data by active sensors consumes extensive energy, and the depth data may need complex post-processing such as filling holes. Despite the leading



(a) Tekin et al. [12]

(b) Proposed boundary-based representation

Figure 1: Illustration of proposed boundary-based representation for out-of-region keypoints. The yellow rectangles represent anchor boxes in 2D detectors. The green rectangle represents detection result. The gray cuboids are 3D bounding boxes of an object. In (a), Tekin et al. [12] regress all the image coordinates of 3D bounding box vertices w.r.t. the anchor box center. Whereas we adopt a two-step procedure to locate keypoints based on refined regional features, and propose a boundary-based keypoint representation to significantly reduce output space.

pose accuracy, depth-based methods [22, 23, 2, 3, 4] usually have large computational cost due to sampling and voting schemes. Therefore, in this work we mainly focus on RGB based 6D pose estimation methods for better efficiency and usability.

### *2.1. RGB-based 6D pose estimation.*

The studies of 6D pose estimation originated from the Perspective-n-Point (PnP) solutions [21, 24], which calculate relative transformations given some pairs of 2D-3D correspondence. Traditional pose estimation approaches can be categorized into keypoint-based methods and appearance-based methods. Keypoint-based methods [16, 17, 18] rely on hand-crafted features to establish 2D-3D correspondences, and then use a PnP algorithm to calculate 6D poses. Despite the high precision, they are slow and unable to tackle textureless objects. Appearance-based methods mainly adopt template matching techniques [25, 26, 27] to directly estimate poses. Methods of this kind are typically sensitive to partial occlusion and appearance variation. Recent works [12, 28, 29, 10, 11, 19, 30, 14, 15] mostly utilize convolutional neural network (CNN) to boost the performance of 6D pose estimation. Commonly used pose representations in CNN-based approaches include continuous pose parameters, discretized viewpoints, and semantic or virtual keypoints. For example, PoseCNN [29] is designed to detect and segment objects in input images, meanwhile regress convolutional features of the objects to 6D pose parameters. SSD-6D [11] discretizes the pose space in the form of viewpoint and in-plane rotation, and then extends SSD [31] with a pose classification branch. These two pose representations attempt to estimate poses in a single shot, and usually need post-refinement to improve pose accuracy. Modern keypoint-based methods [10, 12, 19, 14, 15] learn to predict 2D-3D correspondences between RGB images and 3D models. In terms of keypoint selection, the 3D bounding box corners of the object are most commonly used since they do not require a detailed 3D model. With the help of deep learning, these approaches are able to establish 2D-3D correspondences under challenging conditions where hand-crafted features fail, followed by a PnP solution to achieve accurate pose estimation on multiple 6D pose benchmark datasets.

### *2.2. Keypoint Localization*

One popular application of keypoint localization is human pose estimation. In terms of keypoint representation, heatmap is widely used by many state-of-the-art human pose estimation approaches. For 6D pose estimation, [7] employ stacked hourglass network [32] to predict heatmaps for semantic keypoints. Although the success in locating appearance keypoints, heatmap representation has trouble in handling keypoints outside the object regions, and a common practice during training is to ignore these keypoints directly. However, this strategy is invalid for keypoint-based 6D pose estimation, since 3D bounding box corners are usually outside the object region proposals. In order to break this constrain, [19] samples numerous image patches in a sliding window fashion, and then aggregates all the heatmaps to predict virtual keypoints. More efficiently, [33] applies an extended region mapping approach to enable out-of-view feature point prediction. In addition to heatmap representation, [14] and [15] both predict pixel-wise directions to the keypoints based on segmentation frameworks. Tekin et al. [12] regress offsets of keypoints with respect to the anchor box centers of a single-shot detector. Since they predict on a low resolution feature map ($13 \times 13$), the anchor boxes are usually not well fitted to the objects, thus leading to a large output space of coordinate regression. In contrast, we regress offsets with respect to the closest boundaries by integrating classification and regression schemes to reduce the output space.

### *2.3. 2D Object Detection*

CNN-based object detection methods are commonly categorized into single-shot detectors and two-stage detectors. Single-shot detectors are tuned for speed to directly make predictions for densely-sampled anchor boxes in fully convolutional forms. Whereas two-stage detectors adopt resampling (e.g. ROI-Align [34]) to extract refined regional features of proposals for better accuracy. 6D pose estimation methods used to perform on image patches located by off-the-shelf 2D detectors. Currently the trend is to achieve pose estimation within the detection pipeline by integrating multi-task supervision. To name a few, Tekin et al. [12] and SSD-6D [11] extend single-shot detectors, while PoseCNN [29] selects two-stage detectors. Several works [10, 29, 14, 15] also involve segmentation supervision to better detect objects and predict poses. In this work, we combine the advantages of both classes. We first utilize a single-shot detector to classify and locate objects in real time, then feed the detection results into pose module as proposals. In pose module we extract regional features for keypoint localization using proposed representation.

## 3. Approach

Given an input image, our approach aims to detect all objects of interest and estimate their 6D poses. Motivated by [12], we extend 2D detection pipeline to predict 3D bounding box corners of each object instance in the image. Then we can calculate 6D poses with an efficient PnP algorithm [21] given these 2D-3D correspondences. The schematic overview of proposed network is shown in Fig. 2. Firstly, the input RGB image is resized to 512 × 512 and fed into a single shot 2D detector. We introduce transition and fusion module to bridge feature gap and maintain high spatial resolution. Secondly, we extract refined regional features to locate virtual keypoints and estimate localization confidences using proposed boundary-based keypoint representation. Finally, we integrate confidence-based voting strategy into non-maximum suppression to refine keypoint locations. We now describe each part of our approach in more detail.

### *3.1. Keypoint Representation*

In order to locate out-of-region keypoints, we propose a boundary-based representation by integrating classification and regression schemes. Given a keypoint, we first determine the two perpendicular boundaries that are closest to it, which is equivalent to finding the nearest corner of the region proposal. We achieve this goal by classifying a keypoint into one of four subspaces divided by symmetry axes as shown in Fig. 1b. And then we regress the coordinate offsets of the keypoints with reference to the assigned corners. For subspace classification, we assign ground truth labels from a probabilistic perspective. Instead of one-hot labels, we calculate classification probabilities as follows:

$$p_i = \frac{e^{d_i}}{\sum\limits_{j=1}^{4} e^{d_j}}, \ i = 1, 2, 3, 4 \tag{1}$$

where $d_i$ indicates the distance between a keypoint and one of four corners. In this way we attempt to encode keypoint distribution information into the labels. For example, the classification score of a keypoint close to a symmetry axis should be remarkably lower than those of keypoints near the corners. Meanwhile, keypoints close to symmetry axes are empirically considered to be difficult for subspace classification. Therefore, with this distribution prior, we are able to exploit hard examples. We adopt a re-weighted cross entropy loss during training:

$$L_{cls} = -\alpha \sum_{i=1}^{N} [(1 - \max p_i) \sum_{j=1}^{4} p_{ij} \log \hat{p}_{ij}] \tag{2}$$

where $\hat{p}_{ij}$ is predicted probability, $N$ is the number of keypoints in a mini-batch. $\alpha$ is a normalization factor which is calculated online:

$$\alpha = \frac{N}{\sum\limits_{i=1}^{N}(1 - \max p_i)} \tag{3}$$

For coordinate offsets, the regression targets are calculated as follows:

$$t_x = \frac{gt_x - b_x}{width},\ t_y = \frac{gt_y - b_y}{height} \tag{4}$$

where $b_x$ and $b_y$ indicate coordinates of selected boundaries. *width* and *height* are from the region proposal. We employ smooth KL divergence loss [35] for coordinate regression along with confidence prediction.

$$L_{reg\&conf} = \begin{cases} \frac{e^{-var}}{2}(target_{reg} - pred_{reg})^2 + \frac{var}{2}, & |target_{reg} - pred_{reg}| \leq 1 \\ e^{-var}(|target_{reg} - pred_{reg}| - \frac{1}{2}) + \frac{var}{2}, & |target_{reg} - pred_{reg}| > 1 \end{cases} \tag{5}$$

KL loss formulates predicted coordinates as Gaussian distribution, and *var* represents the learned variances. Ideally, a highly confident prediction should have low variance. Therefore, we calculate localization confidence as follows:

$$conf = e^{-var} \tag{6}$$

Instead of Online Hard Keypoints Mining (OHKM) [36], we adopt Online Hard Coordinates Mining (OHCM) based on the observation that errors of x and y coordinates are usually irrelevant. We process each coordinate separately, and only punish the top K coordinate losses out of $8 \times 2$ in our implementation. The overall loss function for pose estimation consists of the above two parts:

$$L_{pose} = L_{cls} + L_{reg\&conf} \tag{7}$$

### *3.2. Network Architecture*

For object detection, we generally follow a SSD-style [31] structure. The input RGB image is resized to $512 \times 512$ and fed into the backbone network, a FPN-ResNet [37] architecture. Top-down and lateral connections are attached after ResNet Stage 2 through Stage 5 to extract multi-scale features for detection. P6 is down-sampled from Stage 5 output by a $3 \times 3$ stride-2 max pooling layer only to cover large objects. The dimensions of the feature map at scale $s$ is denoted by $(w_s, h_s, c_s)$, where $c_s$ is set to 256 for all feature levels P2 through P6. We create 4 anchor boxes at each location of the feature maps at three aspect ratios $\{1:2,\ 1:1,\ 2:1\}$, with sizes of $25^2$ to $256^2$ on pyramid levels P2 to P6, respectively. All the feature maps are convolved with a set of $3 \times 3 \times c_s$ kernels that share parameters across levels to jointly classify the objects and refine the 2D bounding boxes. The output of detection module at scale $s$ is a 3D tensor of size $(w_s, h_s, 4 \times (4 + C + 1))$, where $C$ denotes the number of object classes excluding the background. Then we merge predictions from all levels and feed the detection results into pose module as region proposals.

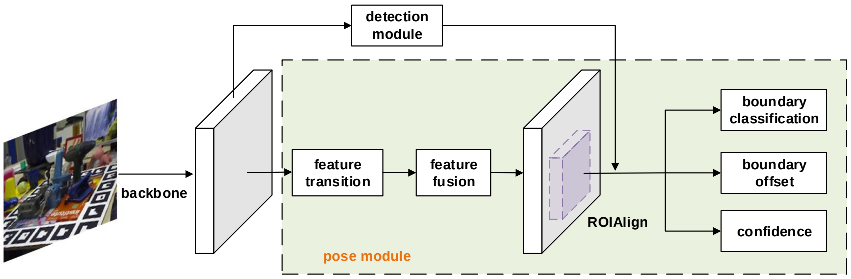


Figure 2: The schematic overview of proposed network. We extend 2D detection pipeline to predict the image coordinates of 3D bounding box vertices for each object instance in the image. In pose module, we introduce feature transition module and fusion module to bridge the gap between detection and pose estimation.

We introduce feature transition module and fusion module to bridge the gap between detection and pose estimation. We argue that pose estimation focuses on appearance variations from different perspectives, whereas object detection aims to achieve invariance against these changes. Therefore, features for detection may not well suited for pose estimation. We illustrate the transition module and fusion module in Fig. 3. Some structure design ideas are inspired by human pose estimation methods [36, 38]. In transition module, we stack more bottleneck residual units into deeper features to compensate for the lack of spatial information. Each residual unit keeps the input and output dimensions the same. In fusion module, each level of features incorporates features from other levels. We adopt consecutive stride-2 $3 \times 3$ convolution for down-sampling, and nearest neighbour interpolation for up-sampling. Then the down-sampled and up-sampled feature maps are element-wise added to the origin feature map. Finally, we concatenate all the fused layers and attach a residual unit to achieve features for pose estimation with 256 channels and stride of 4 w.r.t. input image.

We adopt ROI-Align to extract regional features for pose estimation according to detection results. Each feature vector is fed into two consecutive 1024-way fully connected layers followed by three output branches. The first one has $8 \times 4$ output neurons, where each set of 4 neurons produces softmax probability estimates for subspace classification of a keypoint. The second and third branch has $8 \times 4 \times 2$ output neurons. They encodes positions and variances for keypoint coordinates in a subspace-specific manner, respectively.

### *3.3. Training Procedure*

Inspired by [39], we construct synthetic training sets with two complementary strategies to handle the problem of insufficient annotated real-world data. Firstly, we use 3D models to render images for each object with uniformly sampled poses and scales. Secondly, we employ the ”cut and paste” strategy used

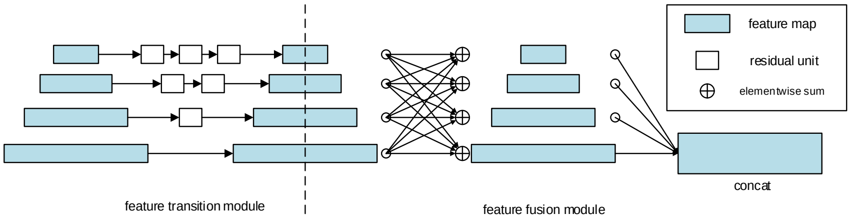


Figure 3: The structure of feature transition module and feature fusion module.

in [12, 10]. The segmented target objects are scaled by a factor of $s \in [0.8, 1.2]$ and randomly placed onto the background. For both strategies the background images are selected from MS COCO dataset [40]. The combined synthetic training set can densely cover 6D pose space while focusing on real-world appearance of the objects. In order to introduce more occlusion patterns, we put multiple object instances into each training image. we also apply random cropping and color jittering during training.

For object detection, positive and negative anchor boxes are decided by the overlaps with ground truth 2D bounding boxes. An anchor box is considered to be positive if it has IoU with a ground-truth bounding box of at least 0.5 and negative otherwise. We select hard negatives anchor boxes so that the positives-negatives ratio is 1:3, to achieve fast convergence and stable training. The detection loss $L_{det}$ is identical as the MultiBox loss in SSD [31]. During training, we select 2000 top-scoring proposals per image from merged multi-scale detection results, followed by non-maximum suppression with a threshold of 0.7. The pose loss $L_{pose}$ is defined only on positive proposals which has IoU with a ground-truth bounding box of at least 0.5. We select at most 256 positive proposals per image. We minimize the overall multi-task loss function:

$$L = L_{det} + \beta L_{pose} \tag{8}$$

The weight term $\beta$ is set to 10 in our implementation. With pretrained ResNet-50 backbone, we train the network using stochastic gradient descent with 0.9 momentum, 0.0005 weight decay, and batch size 8. The initial learning rate is set to 0.001 and divided by 10 at 60k and 80k iterations. All models in our experiments are trained for 90k iterations.

### *3.4. Inference*

When testing, we simultaneously detect objects and estimate 6D poses for all instances by conducting a forward pass of our network. The detection module outputs object identities with scores and 2D bounding boxes. We only select at most 100 top-scoring predictions per image after thresholding score at 0.01. The pose module takes these proposals as input, and yield keypoint positions along with confidences. Motivated by [35], we vote the location of selected keypoints using neighboring instances according to learned confidences within the loop of NMS.

**Algorithm 1** Confidence Voting Keypoint NMS

```
$K$ is $N \times 16$ matrix of initial keypoint positions. $C$ is $N \times 16$ matrix of corresponding confidences. $S$ is corresponding pose scores. $N$ represents the number of proposals. $D$ is the final set of refined keypoints, and $E$ is the final set of refined confidences. $N_t$ is the NMS threshold.
  $K = \{k_1, ..., k_N\}, S = \{s_1, ..., s_N\}, C = \{c_1, ..., c_N\}$
  $D \leftarrow \{\}, E \leftarrow \{\}$
  while $K \neq empty$ do
    $m \leftarrow \arg\max S$
    $idx \leftarrow kpt\_IoU(k_m, K) > N_t$
    $K \leftarrow K - K[idx]$, $S \leftarrow S - S[idx]$
    $p \leftarrow \exp(-(1 - kpt\_IoU(k_m, K[idx]))^2/\sigma_t)$
    $k_m \leftarrow \sum_{i \in idx} p_i c_i k_i \Big/ \sum_{i \in idx} p_i c_i$
    $c_m \leftarrow \sum_{i \in idx} p_i c_i \Big/ \sum_{i \in idx} p_i$
    $D \leftarrow D \cup k_m$, $E \leftarrow E \cup c_m$
  end while
  return $D, E$
```

Similar to [36], we use rescoring strategy in Algorithm 1 to calculated pose scores:

$$s_i = \frac{det\_score_i}{8} \sum_{j=1}^{8} subspace\_score_j \min(conf_{jx}, conf_{jy}),\ i = 1, ..., N \tag{9}$$

where the product of detection score and the average score of all keypoints weighted by confidences is considered as pose score of an object instance. The *kpt_IoU* is adapted from Object Keypoint Similarity (OKS):

$$kpt\_IoU(k_1, k_2) = \frac{1}{8} \sum_{i=1}^{8} \exp\{-\frac{(k_{1ix} - k_{2ix})^2 + (k_{1iy} - k_{2iy})^2}{(w_1 h_1 + w_2 h_2)/2}\} \tag{10}$$

We find our method is robust to the tunable parameter $\sigma_t$, which is set to 0.001 in our experiments. As in [12, 19], we employ efficient PnP algorithm [21] to achieve an estimate of the 6D transformation of the object coordinate frame with respect to the camera coordinate frame.

## 4. Experiments

Our method is implemented using MXNet [41] and ran on an Intel Core i7-6800K@3.40GHz desktop with a GeForce 1080Ti GPU. We present our results on the LINEMOD [22] and OCCLUSION [23] datasets and compare with the state-of-the-art pose estimation methods. LINEMOD dataset consists of 15 sequences of indoor scenes, in which one textureless central object is annotated with identity, 2D bounding box and 6D pose. OCCLUSION is an additionally annotated version of a sequence in the LINEMOD dataset where each frame contains multiple heavily occluded objects in most cases.

### *4.1. Evaluation Metrics*

We use three common metrics to evaluate 6D pose accuracy, including reprojection error, $5cm$ $5^\circ$ metric, and average distance of model points (referred to as ADD metric) as in [12, 11, 10].[1] We report results as the percentage of correctly estimated poses within certain error thresholds. Reprojection error measures pose accuracy in 2D. We project the object's model vertices into the image plane using the estimated poses and the ground truth poses. Estimated pose is considered to be correct when the mean distance between the 2D projections is less than 5 pixels. This metric is suitable for applications such as augmented reality. To measure pose errors in 3D, ADD metric [22] calculates average distance between transformed vertices of object model $M$ by ground truth pose $\mathbf{P}$ and estimated pose $\hat{\mathbf{P}}$.

$$e_{ADD}(\mathbf{P}, \hat{\mathbf{P}}; M) = \underset{\mathbf{x} \in M}{avg} \, ||\mathbf{P}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}||_2 \tag{11}$$

For symmetric objects with ambiguous poses such as *EggBox* and *Glue* in the LINEMOD dataset, the indistinguishable version of the ADD metric is used as in [12, 10]. The threshold is set to 10% of the object's diameter.

$$e_{ADI}(\mathbf{P}, \hat{\mathbf{P}}; M) = \underset{\mathbf{x}_1 \in M}{avg} \, \min_{\mathbf{x}_2 \in M} ||\mathbf{P}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2||_2 \tag{12}$$

We also compare the absolute error of 6D poses using the $5cm$ $5^\circ$ metric. With this metric, the estimated pose is accepted if the translation and rotation errors are below 5cm and 5°, respectively.

### *4.2. Ablation Study*

In this subsection, we analyse the contribution of proposed keypoint representation, network architecture and confidence-voting keypoint NMS to 6D pose estimation accuracy. Ablation experiments are conducted on the OCCLUSION dataset, and average results over 8 objects (see Sec. 4.3.2) are presented.

#### *4.2.1. Keypoint Representation.*

We first investigate the effect of subspace classification targets. We use one-hot labels and standard cross entropy loss as baseline. The result is listed in the row ”hard-cls” in Table 1. Simply using probabilistic labels calculated by

[1] We use the public code in https://github.com/thodan/obj_pose_eval.

Table 1: Ablation studies about subspace classification.

| cls strategy | ADD | $5cm$ 5° | Reproj. 5px |
|---|---|---|---|
| reweighted soft-cls | 29.9 | 27.5 | 61.2 |
| soft-cls | 28.1 | 25.7 | 60.1 |
| hard-cls | 27.2 | 25.6 | 59.8 |

Table 2: Ablation studies about offset regression using Online Hard Coordinate Mining (OHCM).

| topK | ADD | $5cm$ 5° | Reproj. 5px |
|---|---|---|---|
| 4 | 31.2 | 31.0 | 62.5 |
| 8 | 30.6 | 29.4 | 62.4 |
| 12 | 29.9 | 27.8 | 61.4 |
| 16 | 29.9 | 27.5 | 61.2 |

equation 1 provides slight improvement. Combined with proposed reweighted cross entropy loss in equation 2, we can increase ADD metric by 2.7 points over the baseline. We add this empirical hard sample mining strategy after 45k iterations during training.

Our next attempt to improve learning involves using Online Hard Coordinate Mining (OHCM) strategy. We process each coordinate independently, whereas OHKM [36] cannot since they use heatmap representation. We select hard coordinates according to absolute error of offset regression instead of the whole KL loss, whose value can be greatly affected by variance prediction as shown in equation 5. We add OHCM strategy after 45k iterations during training. For each proposal, only the top K coordinate losses out of 16 are punished, and the influence of K is shown in Table 2. Setting $K = 4$ gives a gain of 1.3 points in ADD metric.

We compare proposed boundary-based keypoint representation with heatmap in Table 3. We adapt our network for heatmap representation by replacing the pose ROI head with the mask branch for keypoint localization in Mask R-CNN [34]. The mask branch is trained to locate 8 bounding box corners for each object instance. Out-of-region keypoints are directly ignored during training. As expected, the performance of plain heatmap representation are limited due to the missing of massive supervision information. Using the same network structure, the extended heatmapping approach [33] re-defines the representation area of output heatmap to utilize out-of-region keypoint training samples, thus improves pose results. Nonetheless, proposed boundary-based keypoint representation still significantly outperforms it by 11.2 points in ADD metric.

*4.2.2. Network Architecture.*

We validate the importance of feature transition module and feature fusion module in Table 4. The ablative studies are conducted by removing the modules from our network pipeline. The plain structure without transition and fusion modules directly extract regional features for pose estimation from de-

Table 3: Comparison of proposed boundary-based keypoint representation with heatmap representation.

| Method | ADD | $5cm$ 5° | Reproj. 5px |
|---|---|---|---|
| heatmap | 5.6 | 10.4 | 17.9 |
| extended heatmapping | 20.0 | 23.4 | 52.4 |
| proposed | 31.2 | 31.0 | 62.5 |

Table 4: Ablation studies about feature transition module and feature fusion module.

| transition | fusion | ADD | $5cm$ 5° | Reproj. 5px | speed |
|---|---|---|---|---|---|
|  |  | 28.1 | 26.1 | 59.2 | 35 fps |
| ✓ |  | 29.9 | 27.8 | 60.9 | 32 fps |
|  | ✓ | 29.5 | 27.9 | 60.5 | 34 fps |
| ✓ | ✓ | 31.2 | 31.0 | 62.5 | 30 fps |

tection features. We can observe performance degradation in all three ablative experiments, thus proving the validity of both modules. The combined transition module and fusion module can improve 3.1 points in ADD metric, while bringing in little computational cost.

### *4.2.3. Confidence Voting Keypoint NMS.*

As shown in Table 5, we compare the performance of different NMS strategies and proposed confidence-voting keypoint NMS under various thresholds. The first row presents pose results using standard bounding box NMS. Keypoint NMS adds rescoring strategy (equation 9) and uses keypoint IoU (eqaution 10). These two NMS strategies do not change keypoint positions of top-scoring proposals. Whereas proposed confidence-voting keypoint NMS refines keypoint positions of top-scoring proposals according to localization confidences. Raising the NMS threshold can filter out inaccurate predictions in confidence voting, but it may also increase false positive detections. We set confidence voting keypoint NMS threshold to 0.55 to balance the impact of these two aspects.

## *4.3. Comparison with the State-of-the-art Methods*

We evaluate the performance of our algorithm on simultaneous detection and pose estimation for a single object and multiple objects. Comparison with the

Table 5: Comparison between different NMS strategies.

| Method | Threshold | ADD | $5cm$ 5° | Reproj. 5px |
|---|---|---|---|---|
| bounding box nms | 0.45 | 31.2 | 31.0 | 62.5 |
| keypoint nms | 0.45 | 31.3 | 31.3 | 62.7 |
| conf-voting kpt nms | 0.45 | 31.6 | 31.5 | 62.9 |
|  | 0.55 | 31.8 | 31.6 | 63.2 |
|  | 0.65 | 31.7 | 31.5 | 63.0 |
|  | 0.75 | 31.4 | 31.4 | 63.0 |

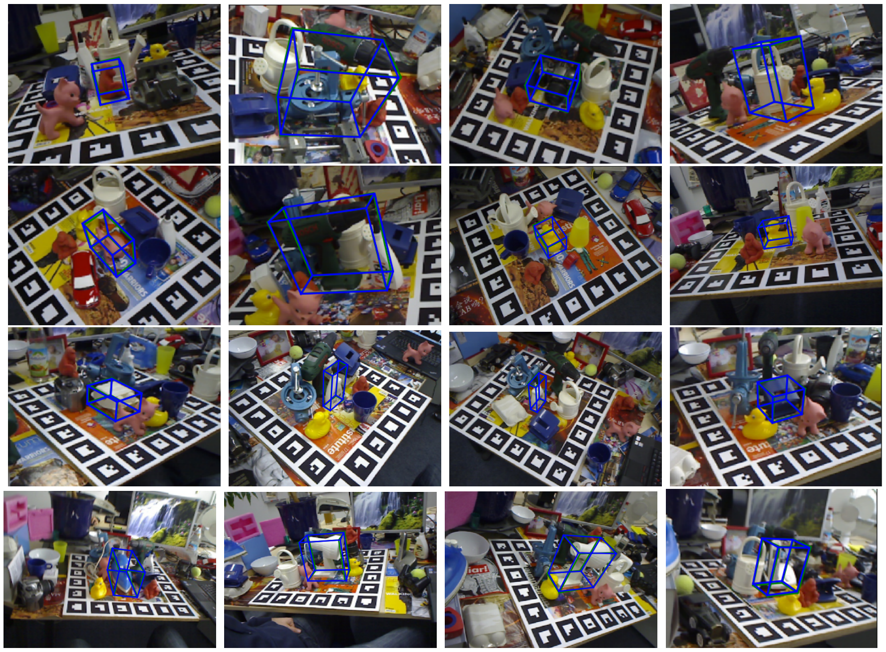


Figure 4: We present qualitative 6D pose estimation results on the LINEMOD dataset. The green and blue 3D bounding boxes are rendered using ground truth poses and predicted poses, respectively.

state-of-the-art RGB based pose estimation methods in terms of various pose metrics is performed on the LINEMOD and OCCLUSION dataset.

### *4.3.1. Results on the LINEMOD Dataset*

The LINEMOD [22] dataset contains 15 sequences of indoor images, among which two sequences, *Cup* and *Bowl*, are commonly ignored since the 3D models are incomplete. In each image, only a central object is annotated with ground truth pose. We use the same train/test split as in [12, 10] and augment the training sets as described in Sec. 3.3. We report quantitative results of our method in terms of 2D reprojection metric and ADD(-I) metric. Qualitative examples of pose predictions are also presented in Figure 4.

In Table 6, we compare our results with those of the state-of-the-art methods under 2D reprojection metric. Detection-driven methods only utilize 2D bounding box and pose annotation, whereas segmentation-driven methods involve additional segmentation supervision. Different from Tekin et al. [12] and Zhang et al. [13], we propose a novel representation to locate keypoints based on detected proposals meanwhile estimate confidences. As can be seen, our approach achieves best accuracy among the compared detection-driven methods. BB8 [10] relies on post-refinement to boost its pose accuracy, but we still outperform it by 7.3%. Our results even competes with recent state-of-the-art method, PVNet [14], which generate keypoint hypotheses by RANSAC-based

Table 6: Comparison of our approach with state-of-the-art algorithms on the **LINEMOD** dataset in terms of **2D reprojection** metric. We report percentages of correctly estimated poses. **Bold face** numbers denote the best overall methods, and blue numbers denote the best detection-driven methods.

| Method | Detection-driven | | | Segmentation-driven | |
|---|---|---|---|---|---|
| Object | Tekin [12] | Zhang [13] | OURS | BB8 [10] w/ ref. | PVNet [14] |
| Ape | 92.10 | 98.0 | 98.8 | 96.6 | **99.23** |
| Benchvise | 95.06 | 93.6 | 94.6 | 90.1 | **99.81** |
| Cam | 93.24 | 98.4 | 98.1 | 86.0 | **99.21** |
| Can | 97.44 | 96.5 | 97.3 | 91.2 | **99.90** |
| Cat | 97.41 | 98.9 | 99.2 | 98.8 | **99.30** |
| Driller | 79.41 | 87.2 | 91.9 | 80.9 | **96.92** |
| Duck | 94.65 | **98.2** | **98.2** | 92.2 | 98.02 |
| Eggbox | 90.33 | 96.8 | 97.9 | 91.0 | **99.34** |
| Glue | 96.53 | 95.3 | 97.3 | 92.3 | **98.45** |
| Holepuncher | 92.86 | 98.2 | 99.0 | 95.3 | **100.0** |
| Iron | 82.94 | 89.7 | 92.7 | 84.8 | **99.18** |
| Lamp | 76.87 | 86.2 | 94.1 | 75.8 | **98.27** |
| Phone | 86.07 | 93.8 | 96.3 | 85.3 | **99.42** |
| Average | 90.37 | 94.7 | 96.6 | 89.3 | **99.00** |

voting.

In Table 7, we report pose accuracy in terms of the ADD(-I) metric described in Section 4.1. *EggBox* and *Glue* are considered as symmetric objects, and the corresponding results are measured using the ADI metric as suggested in [12, 10]. We have achieved substantial improvement compared with Tekin [12] and Zhang [13], thanks to proposed keypoint representation and network architecture. Taking advantage of detailed 3D CAD models, BB8 [10] and SSD-6D [11] significantly boost their pose accuracy by rendering and aligning, which are computationally intensive. However, our results are still better than SSD-6D after refinement by 3.4%. Our results are second only to PVNet [14] which uses segmentation supervision.

The inference speed of our approach for single object is reported in Table 8. With no need of refinement, We can perform simultaneous detection and pose estimation with real-time processing capability. To process a 480 × 640 image, our implementation takes 10 ms for data loading, 16.8 ms for network forward propagation, 3.4 ms for confidence voting keypoint NMS, and 0.1 ms for PnP calculation. Our approach strikes a good balance between speed and pose accuracy.

### *4.3.2. Results on the OCCLUSION Dataset*

In this section, we compare with state-of-the-art methods for multi-object detection and 6D pose estimation on the challenging OCCLUSION dataset. As described in Sec. 3.3, we construct a synthetic training set of 20,000 images by rendering and extracting object patches from corresponding sequences in the LINEMOD dataset. We only use the OCCLUSION dataset as test set to

Table 7: Comparison of our approach with state-of-the-art algorithms on the **LINEMOD** dataset in terms of **ADD(-I)** metric. We report percentages of correctly estimated poses. **Bold face** numbers denote the best overall methods, and blue numbers denote the best detection-driven methods.

| Method | Detection-driven | | | | Segmentation-driven | |
|---|---|---|---|---|---|---|
| Object | SSD-6D[11] w/ ref. | Tekin[12] | Zhang [13] | OURS | BB8[10] w/ ref. | PVNet[14] |
| Ape | **65** | 21.62 | 41.48 | 55.8 | 40.4 | 43.62 |
| Bvise | 80 | 81.80 | 85.38 | 92.8 | 91.8 | **99.90** |
| Cam | 78 | 36.57 | 67.19 | 82.1 | 55.7 | **86.86** |
| Can | 86 | 68.80 | 80.47 | 89.5 | 64.1 | **95.47** |
| Cat | 70 | 41.82 | 60.32 | 72.3 | 62.6 | **79.34** |
| Driller | 73 | 63.51 | 79.79 | 91.0 | 74.4 | **96.43** |
| Duck | **66** | 27.23 | 44.78 | 61.3 | 44.3 | 52.58 |
| Eggbox | **100** | 69.58 | 96.08 | 96.8 | 57.8 | 99.15 |
| Glue | **100** | 80.02 | 87.69 | 92.0 | 41.2 | 95.66 |
| Holep | 49 | 42.63 | 55.59 | 72.4 | 67.2 | **81.92** |
| Iron | 78 | 74.97 | 81.75 | 87.9 | 84.7 | **98.88** |
| Lamp | 73 | 71.11 | 86.08 | 93.4 | 76.5 | **99.33** |
| Phone | 79 | 47.74 | 65.49 | 84.5 | 54.0 | **92.41** |
| Average | 79 | 55.95 | 71.70 | 82.4 | 62.7 | **86.27** |

Table 8: Comparison of our approach with state-of-the-art algorithms in terms of inference speed.

| Method | Overall speed for 1 object | Refinement runtime |
|---|---|---|
| BB8 [10] | 4 fps (Titan X) | 21 ms/object |
| SSD-6D [11] | 10 fps (GTX 1080) | 24 ms/object |
| PoseCNN [29] | 2 fps (GTX 1080) | 24 ms/object |
| Tekin [12] | 50 fps (Titan X) | - |
| Zhang [13] | 25 fps (GTX 1080Ti) | - |
| DeepHMap [19] | < 4 fps (GTX 980Ti) | - |
| PVNet [14] | 25 fps (GTX 1080Ti) | - |
| [15] | 22 fps (Modern GPU) | - |
| OURS | 33.0 fps (GTX 1080Ti) | - |

Table 9: Results on the OCCLUSION dataset. We report percentages of correctly estimated poses. **Bold face** numbers denote the best overall methods.

| metric | ADD(-I) | | | | Reproj. 5px | | | | |
|---|---|---|---|---|---|---|---|---|---|
| method | [15] | PVNet [14] | DeepHMap w/ FM [19] | ours | [15] | Tekin [12] | PVNet [14] | DeepHMap w/ FM [19] | ours |
| Ape | 12.1 | 6.50 | **17.6** | 14.3 | 59.1 | 7.0 | 69.14 | **69.6** | 68.0 |
| Can | 39.9 | **65.04** | 53.9 | 59.7 | 59.8 | 11.2 | 86.09 | 82.6 | **88.4** |
| Cat | 8.2 | 15.00 | 3.3 | **19.7** | 46.9 | 3.6 | **65.12** | 65.1 | 64.2 |
| Driller | 45.2 | 55.60 | **62.4** | 49.3 | 59.0 | 5.1 | 73.06 | **73.8** | 69.5 |
| Duck | 17.2 | 15.95 | 19.2 | **28.6** | 42.6 | 1.4 | 61.44 | 61.4 | **75.7** |
| Eggbox | 22.1 | **35.23** | 25.9 | 17.9 | 11.9 | - | 8.43 | **13.1** | 8.9 |
| Glue | 35.8 | 42.64 | 39.6 | **44.2** | 16.5 | 4.7 | **55.37** | 54.9 | 54.9 |
| Holep. | **36.0** | 35.06 | 21.3 | 20.8 | 63.6 | 8.3 | 69.84 | 66.4 | **75.9** |
| Average | 27.0 | **33.88** | 30.4 | 31.8 | 44.9 | 6.2 | 61.06 | 60.9 | **63.2** |

avoid seeing the occlusion patterns in advance. Other training settings are the same as in Sec. 3.3. Pose estimation results are presented in Table 9. As can be seen, we achieve the best pose accuracy in terms of 2D reprojection metric and outperform PVNet [14] by 2.1 points. Compared with Tekin et al. [12], our method is significantly more robust to partial occlusions since proposed keypoint representation explicitly measures localization confidences. At expense of great computational cost, [19] utilizes a sampling and accumulating scheme to handle occlusions. They also introduce Feature Mapping (FM) [42] procedure to boost pose accuracy by bridging the domain gap between synthetic training data and real-world test images. Whereas we seek to achieve the same goal by elaborately constructing training set. Our approach is much more efficient than [19], meanwhile achieves better pose accuracy as shown in Table 8 and Table 9. Note that PVNet [14] elaborately select surface keypoints instead of the 3D bounding box corners. For a fair comparison, we present the results of all methods including PVNet using the 3D bounding box corners as keypoints. In terms of object detection, our method achieves a mean Average Precision (mAP) of 0.80 at IoU threshold 0.5 over the 8 objects. We present qualitative results on the OCCLUSION dataset in Figure 5.

On the OCCLUSION dataset, our implementation takes 10 ms for data loading, 19.5 ms for network forward propagation, 3.6 ms for confidence voting keypoint NMS, and about 1 ms for PnP calculation. Our approach runs at 30 fps for multi-object pose estimation.

## 5. Conclusion

In summary, we have developed an effective CNN framework for RGB based 6D pose estimation by locating 3D bounding box vertices. For that goal, we propose a boundary-based keypoint representation to better locate out-of-region keypoints. Proposed representation also estimates localization confidences to

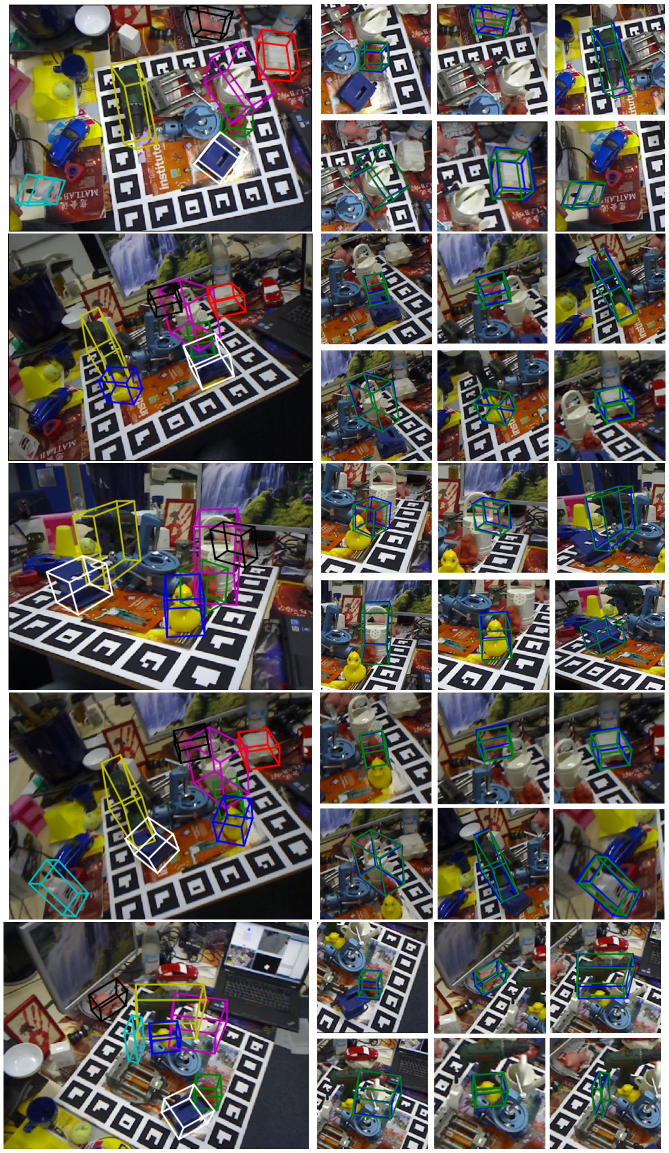


Figure 5: We present qualitative 6D pose estimation results on the OCCLUSION dataset. In left column we only draw the 3D bounding boxes rendered by predicted poses. In right column we render the green and blue 3D bounding boxes using ground truth poses and predicted poses, respectively.

alleviate the influence of diffcult keypoints by confidence voting keypoint NMS. As an extension of 2D detection pipeline, proposed network runs fast and can be trained in end-to-end manner. Our approach applies to textureless objects and is robust to partial occlusions. Experimental results on two common benchmarks validate that proposed detection-driven method achieves state-of-the-art pose accuracy.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61501009 and 61771031), the National Key Research and Development Program of China (Grant Nos. 2016YFB0501300 and 2016YFB0501302), and the Fundamental Research Funds for the Central Universities.

## References

[1] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, S. Savarese, Densefusion: 6d object pose estimation by iterative dense fusion, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[2] W. Kehl, F. Milletari, F. Tombari, S. Ilic, N. Navab, Deep learning of local rgb-d patches for 3dobject detection and 6d pose estimation, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 205–220.

[3] S. Hinterstoisser, V. Lepetit, N. Rajkumar, K. Konolige, Going further with point pair features, in: European Conference on Computer Vision, Springer, 2016, pp. 834–848.

[4] J. Vidal, C.-Y. Lin, R. Martí, 6d pose estimation using an improved method based on point pair features, in: 2018 4th International Conference on Control, Automation and Robotics (ICCAR), IEEE, 2018, pp. 405–409.

[5] H. Su, C. R. Qi, Y. Li, L. J. Guibas, Render for CNN: Viewpoint estimation in images using cnns trained with rendered 3d model views, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2686–2694. doi:10.1109/ICCV.2015.308.

[6] S. Tulsiani, J. Malik, Viewpoints and keypoints, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1510–1519. doi:10.1109/CVPR.2015.7298758.

[7] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, K. Daniilidis, 6-dof object pose from semantic keypoints, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2011–2018. doi:10.1109/ICRA.2017.7989233.

[8] Y. Kao, W. Li, Z. Wang, D. Zou, R. He, Q. Wang, M. Ahn, S. Hong, et al., An appearance-and-structure fusion network for object viewpoint estimation., in: IJCAI, 2018, pp. 4929–4935.

[9] Z. Wang, W. Li, Y. Kao, D. Zou, Q. Wang, M. Ahn, S. Hong, Hcr-net: A hybrid of classification and regression network for object pose estimation., in: IJCAI, 2018, pp. 1014–1020.

[10] M. Rad, V. Lepetit, BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3848–3856. doi:10.1109/ICCV.2017.413.

[11] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, N. Navab, SSD-6D: Making rgb-based 3d detection and 6d pose estimation great again, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1530–1538. doi:10.1109/ICCV.2017.169.

[12] B. Tekin, S. N. Sinha, P. Fua, Real-time seamless single shot 6d object pose prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 292–301.

[13] X. Zhang, Z. Jiang, H. Zhang, Real-time 6d pose estimation from a single rgb image, Image and Vision Computing 89 (2019) 1–11.

[14] S. Peng, Y. Liu, Q. Huang, X. Zhou, H. Bao, Pvnet: Pixel-wise voting network for 6dof pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4561–4570.

[15] Y. Hu, J. Hugonot, P. Fua, M. Salzmann, Segmentation-driven 6d object pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3385–3394.

[16] A. Rubio, M. Villamizar, L. Ferraz, A. Penate-Sanchez, A. Ramisa, E. Simo-Serra, A. Sanfeliu, F. Moreno-Noguer, Efficient monocular pose estimation for complex 3d models, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1397–1402. doi:10.1109/ICRA.2015.7139372.

[17] L. Svrm, O. Enqvist, M. Oskarsson, F. Kahl, Accurate localization and pose estimation for large 3d models, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 532–539. doi:10.1109/CVPR.2014.75.

[18] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints, International Journal of Computer Vision 66 (3) (2006) 231–259. doi:10.1007/s11263-005-3674-1.
URL https://doi.org/10.1007/s11263-005-3674-1

[19] M. Oberweger, M. Rad, V. Lepetit, Making deep heatmaps robust to partial occlusions for 3d object pose estimation, European Conference on Computer Vision.

[20] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[21] V. Lepetit, F. Moreno-Noguer, P. Fua, EPnP: An accurate o(n) solution to the pnp problem, International Journal of Computer Vision 81 (2) (2008) 155. doi:10.1007/s11263-008-0152-6.
URL https://doi.org/10.1007/s11263-008-0152-6

[22] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model based training, detection and pose estimation of textureless 3d objects in heavily cluttered scenes, in: K. M. Lee, Y. Matsushita, J. M. Rehg, Z. Hu (Eds.), Computer Vision – ACCV 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 548–562.

[23] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, C. Rother, Learning 6d object pose estimation using 3d object coordinates, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 536–551.

[24] C. P. Lu, G. D. Hager, E. Mjolsness, Fast and globally convergent pose estimation from video images, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (6) (2000) 610–622. doi:10.1109/34.862199.

[25] Z. Cao, Y. Sheikh, N. K. Banerjee, Real-time scalable 6dof pose estimation for textureless objects, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 2441–2448. doi:10.1109/ICRA.2016.7487396.

[26] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, V. Lepetit, Gradient response maps for real-time detection of textureless objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (5) (2012) 876–888. doi:10.1109/TPAMI.2011.206.

[27] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, K. Daniilidis, Single image 3d object detection and pose estimation for grasping, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 3936–3943. doi:10.1109/ICRA.2014.6907430.

[28] A. Kendall, M. Grimes, R. Cipolla, PoseNet: A convolutional network for real-time 6-dof camera relocalization, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2938–2946. doi:10.1109/ICCV.2015.336.

[29] Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes, in: Robotics: Science and Systems (RSS), 2018.

[30] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3d orientation learning for 6d object detection from rgb images, in: European Conference on Computer Vision, Springer, 2018, pp. 712–729.

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in: ECCV, 2016.

[32] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 483–499.

[33] O. Moolan-Feroze, A. Calway, Predicting out-of-view feature points for model-based camera pose estimation, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 82–88. `doi:10.1109/IROS.2018.8594297`.

[34] K. He, G. Gkioxari, P. Dollr, R. Girshick, Mask r-cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. `doi:10.1109/ICCV.2017.322`.

[35] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[36] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded Pyramid Network for Multi-Person Pose Estimation.

[37] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, S. J. Belongie, Feature pyramid networks for object detection., in: CVPR, Vol. 1, 2017, p. 4.

[38] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: CVPR, 2019.

[39] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, S. Birchfield, Deep object pose estimation for semantic robotic grasping of household objects, in: Conference on Robot Learning (CoRL), 2018.
URL `https://arxiv.org/abs/1809.10790`

[40] G. Lin, C. Shen, Q. Shi, A. van den Hengel, D. Suter, Fast supervised hashing with decision trees for high-dimensional data, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1971–1978. `doi:10.1109/CVPR.2014.253`.

[41] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, Z. Zhang, MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems, in: Neural Information Processing Systems, Workshop on Machine Learning Systems, 2015.

[42] M. Rad, M. Oberweger, V. Lepetit, Feature mapping for learning fast and accurate 3d pose inference from synthetic images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4663–4672.