Reconstruct Locally, Localize Globally: A Model Free Method for Object Pose Estimation

Ming Cai, Ian Reid
The University of Adelaide
{ming.cai, ian.reid}@adelaide.edu.au

Abstract

Six degree-of-freedom pose estimation of a known object in a single image is a long-standing computer vision objective. It is classically posed as a correspondence problem between a known geometric model, such as a CAD model, and image locations. If a CAD model is not available, it is possible to use multi-view visual reconstruction methods to create a geometric model, and use this in the same manner. Instead, we propose a learning-based method whose input is a collection of images of a target object, and whose output is the pose of the object in a novel view. At inference time, our method maps from the RoI features of the input image to a dense collection of object-centric 3D coordinates, one per pixel. This dense 2D-3D mapping is then used to determine 6dof pose using standard PnP plus RANSAC. The model that maps 2D to object 3D coordinates is established at training time by automatically discovering and matching image landmarks that are consistent across multiple views. We show that this method eliminates the requirement for a 3D CAD model (needed by classical geometry-based methods and state-of-the-art learning based methods alike) but still achieves performance on a par with the prior art.

1. Introduction

In computer vision, the pose of an object describes the geometric relation of the object instance with respect to the capturing camera. It is mathematically encoded by the Euclidean transformation between the representations of the object structure in two coordinate spaces: object-centric and camera-centric frame. The task we are interested in is to estimate the accurate six-degree-of-freedom (6dof) pose of a previously-seen rigid object instance from an RGB image.

Standard methods to solve this problem make use of a CAD model of the object. This predefined structural information contributes variously to the classical geometry methods[10, 57, 31, 25, 39] and recent machine learning based methods[41, 22, 48, 54, 27, 47, 20, 40, 52, 55, 7]. For instance, in the classic family, such roles might be the reference for registration[57], base for templates generation[18] and provider of texture for feature extraction. As for the CNN-based approaches, this model acts such as the supervision for network learning[2, 38, 4, 22], a source for synthetic image generation[40, 27, 22, 9] and/or an agent for post-process refinement[27, 41, 22] *etc*. However, fine-grained and well-textured 3D structure does not exist for every object in the wild. This limits the generalization of these approaches. In this paper, we are therefore devoted to answer this question: Is it possible to accomplish the object pose estimation task, *without using the 3D CAD model of the object?*

Reconstruction-based methods [51, 37, 36] have shown the feasibility of this proposal. They firstly reconstruct the 3D object from the multi-view RGB images to substitute missing CAD model, using Structure from Motion (SfM[1]). Object pose is then solved using the Perspective-

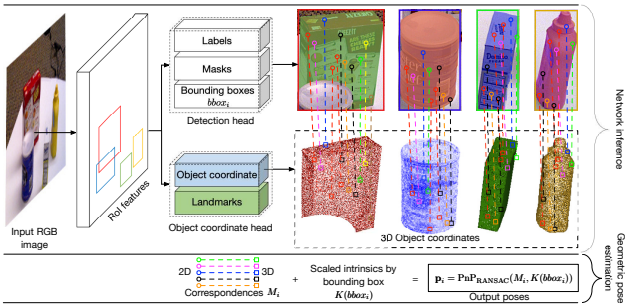


Figure 1. **The inference of our approach.** Although our model is trained with a pair of two-view images, it requires only a single image to infer the object pose. For a novel image, the detection head predicts a box and a mask for the object. Meanwhile the object coordinate head outputs a 3D object point map for the box. The object points on the background are removed according to the mask. The remaining points are used for establishing the 2D-3D correspondences within the box. The 6dof pose is then solved via PnP plus RANSAC based on these correspondences along with the scaled projection matrix derived from the box position. The pose estimate is subsequently refined using the predicted object points. The landmark head is turned off at inference time.

n-Point (PnP) algorithm, based on the correspondence of the 2D visual cues of a new image and those affiliated to the 3D reconstruction. Although the handcrafted feature descriptors perform efficiently in detection and matching, they cause the main limitations in the pipeline: (i) Their main purpose is to generally detect the salient keypoints with rich texture, rather than to describe the structure of the object; (ii) For largely texture-less objects, a paucity of interest points can often lead to a poor or unreliable interpolated reconstruction.

Camera relocalization is a very closely related problem (because its objective is also to find a 6dof pose), and this has recently been tackled from the perspective of CNN regression[24]. However direct regression has not proven as accurate as standard geometric methods. More promising are the methods of [3, 5] which use the power of CNNs to establish high quality dense correspondences and the subsequent accuracy of geometric methods. Nevertheless there are aspects of the camera relocalization problem that are not directly analogous to object pose estimation. The main difference that prevents direct adoption of these methods for object pose is that the object is only visible in part of the scene, necessitating a need to distinguish the object from the rest of the scene.

Hence, the problem we seek to solve is: given as input a collection of images and their poses, learn a system that can then detect and localize the object in any subsequent view. Inspired by the success of the hybrid approach [2, 5, 6, 8], we introduce: Reconstruct Locally, Localise Globally (RLLG), a learning and reconstruction-based method to object pose estimation. Our solution differs from SfM in that there is no explicit 3D model of the target created. We implicitly encode the process of reconstruction within the weights of a neural network during training. At inference stage, this network serves as a 2D-3D correspondences establisher for the test image. Our method then estimates the accurate 6dof pose of the object from the these correspondences using PnP plus RANSAC[12].

In order to identify, detect and isolate the objects from the background, and concurrently perform reconstruction, we seamlessly build our model upon a region proposal network, Mask R-CNN[16]. This framework comprises a backbone network along with three special-purpose heads: bounding box head, classification head and segmentation head. We contribute a new head – the *object coordinate head* – to the same backbone, whose output is the dense 3D coordinates of the object in object-centric frame. It efficiently establishes dense correspondences between 2D positions and 3D points in inference, therefore provides plentiful constrained samples for absolute object pose estimation.

Since the goal of RLLG is to disengage the ground truth 3D model from the pose estimation pipeline, how to learn the object coordinate head without manual annotation is a key issue. We propose to provide an alternative supervisory signal derived from multi-view geometry. We design the head as a two-branched fully convolutional network (FCN)[29]. One of the branches automatically recognizes the viewpoint-independent 2D object landmarks, and the other positions them in the 3D object-centric frame using multi-view constraints. Since landmarks and object coordinates are both intrinsic properties, they are invariant to the change of the external factors (such as pose and illumination). The learning therefore aligns them in pairs of images related by a warp and expects the detector and regressor to be equivariant with the image deformations.

For 2D landmark learning, the warp is created for an image-pair by applying in-plane transformations (*e.g.* in-plane rotation, scaling and crop) to an image. Whereas for 3D object coordinate learning, we propose to explicitly build the constraints based on images from two viewpoints arise from an out-of-plane movement. The reason we do not use single-view deformation to constrain object coordinates is twofold: (i) from a geometric perspective, the pixel-wise correspondences between an image-pair induced by the in-plane operations do *not* constrain the location of the object point in 3D space; (ii) the pose-invariance of the object coordinates is insufficiently guaranteed by feeding multiple single-view images from different viewpoints to the network during iterative training.

We create a dataset to showcase the effectiveness our object coordinate regression and subsequent pose estimation. Our 3D model free pose estimation method is also tested on the LINEMOD [18] and Occlusion LINEMOD [18] dataset to prove its generalization and robustness to real world scenarios. It achieves the on-par performance with the state-of-the-art methods that require the 3D object in different ways.

2. Related work

Feature-based Methods and Template-based Methods: It is necessary to review how the geometry-based methods solve the 6dof pose estimation, since our method is essentially a combination of learning and geometry. Traditionally, these methods [13, 30, 32, 21] consist of two key components: feature detection plus matching, geometric pose solving plus refinement. The features, such as ORB [44, 33], SIFT [30] and FAST [43], are descriptors of the local appearance around the key-points. They are manually handcrafted to achieve invariance over viewpoint transformation and descriptiveness for matching. From these matched 2D-3D correspondences, the transformation between the camera and the object can be estimated by geometric algorithms such as [17, 56, 26, 53]. Robust fitting like [12] is applied to find the optimal pose.

Some authors [51, 37, 36] target the case when the 3D model is missing. The solution is to build an alternative

model using reconstruction approaches such as [1] from the matched 2D feature points. Given a query object image, the same family of features are found and then matched with the 3D database to solve the pose. Despite the efficiency of the descriptors in detection and matching, they are not handcrafted to encode the geometric structure of the object instance. The sparsity of them also potentially cause unreliability in 3D reconstruction for texture-less object.

Template-based methods aim to estimate the pose of the object without using the sparse features. [18, 14, 19] defines templates for the whole object depending on the gradients and features from the RGB images. They are matched for the query image to infer the pose.

Learning-based Methods with CAD Model: Like detection, segmentation and other recognition tasks, object pose estimation also benefits from the recent development of deep learning. Most of the learning-based methods integrate the 3D object model in the process of learning and/or inference. BB8[41], Oberweger[35], Tekin[48] and [20] create a 3D bounding box around the object model, and define the 8 (or 9 with center point in[48]) corners as the 3D key-points on the object. They then annotate their 2D projections and train various networks to perform keypoint detection on image, establishing a sparse 2D-3D correspondences for pose estimation. PVNet[40] proposes a method that automatically discovers a set of keypoints on the 3D object surface based on the physical structure, to ensure that their 2D projection are all within the silhouette.

The CAD model is also very handy when generating new data for the training. [40, 42, 35] use the textured object model and random poses to generate a large amount of synthetic images to augment (or replace) the limited training images, preventing the network from overfitting. The 3D object model could also serve as the base for loss evaluation. [52, 27, 54] compares the offsets between the object model transformed by the predicted pose and the ground truth pose. This error is used for back-propagation to train the network, and successfully avoids the imbalanced weighting between translation and rotation when a model builds the losses using distances in the translational and rotational spaces separately (such as [24] and [23]). Moreover, in [55, 27, 41, 22], the 3D model is used for post-refinement to improve the quality of the pose estimates. Having the output pose from the network as the initialization, a iterative optimization is designed to produce the optimal pose solution by minimizing an objective related to the 3D model. Such objective can be the consistence between the rendered color image from the textured model and the input image[41], or the distance between the transformed object points in camera frame and those recovered from depth[22].

Similar to our work, Pix2Pose[38] and [2] also use object coordinates as an intermediate representation to find the object pose. However, in these methods, a 3D model of the object provides the direct supervision for the model (such as a random forest[2] or a neural work[38]) learning. In contrast, we aim to learn the coordinates without the 3D model in a self-supervised way (by self-supervised, we mean that the supervision that governs the learning of the object coordinate does not come from the ground truth directly).

3. Reconstruct locally, Localize globally

Denote by $I_i, i \in \{1...n\}$ an image of object O_l, where $l \in \{1...L\}$ is object label, and by $\mathbf{P}_{i,l}$ the visible 3D object points in I_i. Their coordinates in object-centric frame O and camera-centric frame C are $\mathbf{P}^O_{i,l}$ and $\mathbf{P}^C_{i,l}$ respectively. The pose of this object $\mathbf{T}_{i,l}$ consists of two parts: rotation $\mathbf{R}_{i,l} \in \mathbb{R}^{3\times3}$ and translation vector $\mathbf{t}_{i,l} \in \mathbb{R}^3$. It is essentially the transformation between two Euclidean spaces:

$$\mathbf{P}^C_{i,l} = \mathbf{R}_{i,l}\mathbf{P}^O_{i,l} + \mathbf{t}_{i,l}. \tag{1}$$

Camera intrinsics $\mathbf{K}$ projects $\mathbf{P}^C_{i,l}$ onto image and obtains the 2D coordinates of the projections $\mathbf{p}_{i,l} = [\mathbf{u}, \mathbf{v}]$, where

$$s\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\\ 1\end{bmatrix} = \mathbf{K}\mathbf{P}^C_{i,l} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix}f_x & 0 & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{bmatrix}. \tag{2}$$

s is a scale factor, f_x and f_y are the focal lengths and (c_x, c_y) is the camera center.

The correspondences between 2D points $\mathbf{p}_{i,l} = [\mathbf{u}, \mathbf{v}]$ and 3D points $\mathbf{P}^O_{i,l}$ preserve the geometric transformation of the object to the camera, and therefore are used to estimate the pose at inference time. We aim to build a network to densely build these correspondences, by mapping from the RGB image pixels to 3D coordinates in the object space.

Mask R-CNN: We start by recapping the Mask R-CNN detector and segmenter [16] in brief. There are two stages in Mask R-CNN. The first is carried out by a Region Proposal Network (RPN), which proposes candidate object bounding boxes (Regions of Interest, RoIs). The second stage then extracts features using RoIAlign from each RoI, and subsequently performs classification, bounding-box regression, and instance segmentation. During training, the multi-task loss on each sampled RoI is $L = L_{cls} + L_{box} + L_{mask}$. Please refer to [16] for loss definitions. RoIAlign layer performs bilinear interpolation over the feature from the RPN, and pools out a fixed-size RoI feature. In analogy to the mask head, our proposed *object coordinate head* learns to transfer from the RoI features to a coordinate map.

3.1. Object Coordinate Head

Fig. 2 shows the training of the proposed object coordinate head. As mentioned in Section 1, this new head consists of two branches: object coordinate branch and landmark branch. The object coordinate branch is introduced

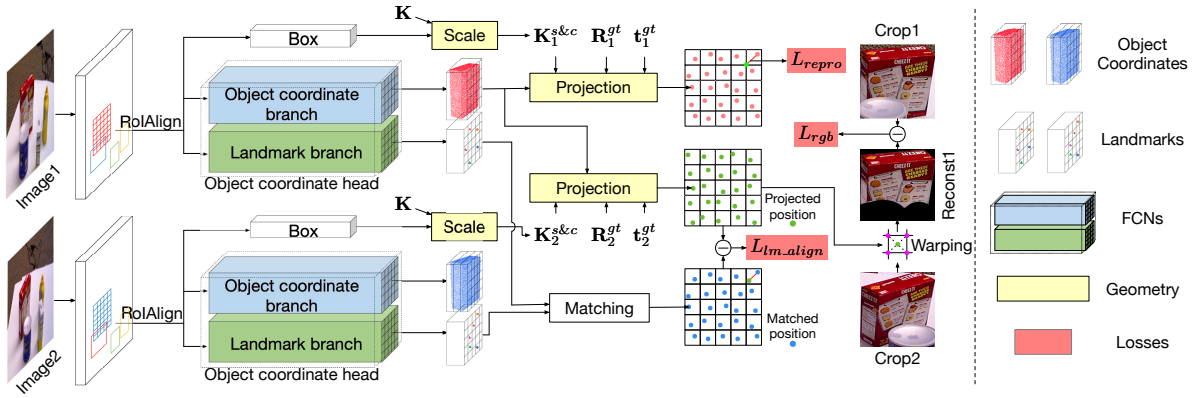


Figure 2. **The training of object coordinate branch.** The losses for detection heads and landmark head are omitted for simplicity.

first since it is directly related to the pose estimator. We then show the reason why the landmark branch is needed and how it benefits the learning of object coordinates.

Object Coordinate Branch: The spatial map of the object coordinate relates to the 2D layout of the object in the image. Therefore by nature we use convolutions to provide the pixel-to-pixel correspondences between image and object coordinates. We apply a FCN Φ_{obj} on each RoI features. The output of Φ_{obj} is a $m \times m \times 3$ vector map $\mathbf{P}^{O}_{i,l,(h,w)} = \Phi_{obj}(I_i),\ h \in \{1 \dots m\}, w \in \{1 \dots m\}$, where each pixel is a 3D vector that represents a location on the imaginary 3D model of the target object.

The training of Φ_{obj} is straightforward if the 3D object model is accessible, which makes the learning fully supervised. Instead, we aim to present a model-free method and therefore propose to explore alternative supervisions.

Due to the graceful alignment provided by FCN, the predicted object coordinate map maintains explicit per-pixel spatial correspondence with RoIs. We first explore supervision according to these correspondences via projection.

Projection within RoIs: To perform projection inside the RoIs, we need to adapt the projection matrix $\mathbf{K}$ to a proposal box. For each proposal, the RPN estimates a 4D vector $(x_{min}, y_{min}, x_{max}, y_{max})$ that parameterizes a box around the target pixel. In term of spatial dimension, with this box, the RoIAlign layer gathers and pools the RoI features from the backbone and then up/down-samples to $m \times m$ via the FCN Φ_{obj}. Two operations change the spatial dimension of our interest region and consequently reform the projection model: crop (by the RoIAlign) and resize (by up/down-sampling). We therefore assume the $m \times m$ object point map fully corresponds to a *new* $m \times m$ image $I_{i,s\&c}$, which is a resized crop of I_i. The intrinsics hence scales to

$$\mathbf{K}_{c\&s} = \begin{bmatrix} s_w f_x & 0 & s_w(c_x - x_{min}) \\ 0 & s_h f_y & s_h(c_y - y_{min}) \\ 0 & 0 & 1 \end{bmatrix}, \qquad (3)$$

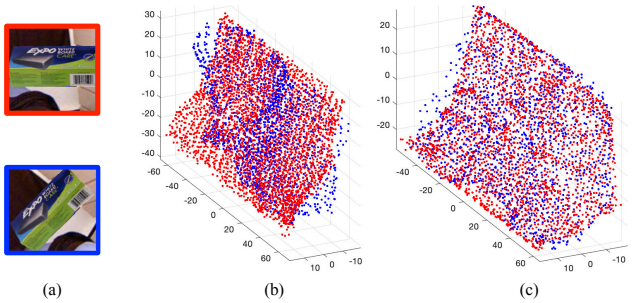
where $s_w = m/(x_{max} - x_{min})$ and $s_h = m/(y_{max} - y_{min})$. As a result, the predicted re-projection on $I_{i,s\&c}$ from ground truth object pose is

$$\mathbf{p}^{pred}_{i,l,(h,w)} = \frac{1}{s}\mathbf{K}_{c\&s}(\mathbf{R}^{gt}_{i,l}\mathbf{P}^{O}_{i,l,(h,w)} + \mathbf{t}^{gt}_{i,l}). \qquad (4)$$

The expected projection of an object coordinate simply is the 2D pixel position where it lies in the output map, which means $\mathbf{p}^{gt}_{i,l,(h,w)} = [h, w], h \in \{1 \dots m\}, w \in \{1 \dots m\}$. The learning objective is to minimize the reprojection error triggered by any difference that we assume arises from an error in the predicted object coordinates. We therefore define the single-view reprojection loss as

$$L_{repro} = \frac{1}{m \times m}\sum_{h,w}\left\|\mathbf{p}^{pred}_{i,l,(h,w)} - \mathbf{p}^{gt}_{i,l,(h,w)}\right\|_2. \qquad (5)$$

Since loss (5) is evaluated for a single image, it potentially has limitations. From a geometric perspective, loss (5) settles to optimal for any point on the line that connects the camera origin and the real 3D object point. Hence, theoretically, minimizing loss (5) does not guarantee the network to regress to the correct coordinates. The training however happens iteratively in practice, which means that the network sees images of the object in different viewpoints from batch to batch. It is expected that the network learns to recognize the same object point with various visual appearance (caused by viewpoint change) in different images and *consistently* regress to a same coordinate. Such behavior would be an implicit multi-view constraint for the learning and contributes to discover the true geometry of the object. In order to experimentally validate this hypothesis, we create a synthetic dataset (the details is given in Section 4) and train the object coordinate head with loss term (5). The trained model is tested with an object image and its rotated variant (Fig. 3(a)). The predicted object points are shown in Fig. 3(b) in red and blue respectively. The obvious incompatibility in two reconstructions suggests that single-view loss-based training does not produce a consistent 3D coordinate for the same object point in different views.



(a) (b) (c)

Figure 3. **Comparison between the 3D object points for an image and its variant.** (a): an object image and its rotated version. (b): Reconstruction from single-view reprojection loss. (c): Reconstruction from multi-view consistence loss.

To overcome this limitation, we propose to make the multi-view constraints explicit and provide strong geometric supervision for object coordinates learning. Based on [15], images from multiple viewpoints can be used to constrain the coordinate for a 3D point using triangulation. Such geometry is built upon the 2D-2D correspondences between the objects in different images. To that end, we propose to include an additional *landmark branch*, which discovers characteristic points on the object. The learned landmarks for multiple images are then matched during the learning of object coordinates, establishing a dense collection of 2D-2D correspondences. The multi-view constraints is explicitly built accordingly.

Landmark Branch: Landmark is defined as the characteristic keypoint on the object that can be recognized and correlated from different viewpoints. Its representation is a d dimensional feature vector explored automatically by the network for uniqueness and rich descriptiveness. It is intrinsic to the object, which means the change of viewpoint or deformation should not cause any difference to the representation of a unique landmark on the object. Such behavior is defined as *equivariance constraint* [50]. We therefore exploit this property as the supervision for landmark learning due to the lack of manual annotation.

The landmark branch is also a FCN due to the one-to-one mapping from pixels to landmarks. It is learned in a Siamese setting with two images – I_i and $r(I_i)$ – correlated by a known deformation r. Such deformation transforms the point (h, w) of the source to (h_r, w_r) on the target. Denote by Φ_{lm} the landmark branch. It takes I_i and $r(I_i)$ as input at the same time and outputs two $m \times m \times d$ landmark maps $L = \Phi_{lm}(I_i)$ and $L^r = \Phi_{lm}(r(I_i))$ for each RoI. The equivariance constraint is defined as $L_{(h,w)} = L^r_{(h_r,w_r)}$ where $h, w \in 1 \dots m$. In order to prevent this constraint from falling into a degenerated case, when all the pixels are mapped to a singular object landmark, we follow [49] to reformulate it to a distance-aware softmax loss.

The relative similarities between landmarks on two RoIs are formulated by a softmax function on the cos similarities. What is expected from the leaning is that the landmarks on two images with short distance have large similarity, and vice versa. Therefore the relative similarities are weighted by the landmark distances in the loss term

$$L_{lm} = \frac{1}{m^4} \sum_{\substack{h_s,w_s \\ h_t,w_t}}^{m} \text{dist}(s,t) \frac{e^{s((h_s,w_s),(h_t,w_t))}}{\sum_{h'_t,w'_t}^{m} e^{s((h_s,w_s),(h'_t,w'_t))}}, \quad (6)$$

where $\text{dist}(s,t) = \|(h_s, w_s) - (h_t, w_t)\|_2$, and

$$s((h_s,w_s),(h_t,w_t)) = \frac{L_{(h_s,w_s)} \cdot L^r_{(h_t,w_t)}}{\left\|L_{(h_s,w_s)}\right\|_2 \left\|L^r_{(h_t,w_t)}\right\|_2} \quad (7)$$

is the cos similarity.

There are various of choices for deformation r to benefit the discovery of landmarks. Nonetheless we consider the in-plane rotation and scaling (to ensure a same dimension with the original image), which preserve the rigidness of the object. Therefore we can re-use the object coordinate branch to predict the 3D points for the transformed image $r(I_i)$, without non-trivial modification to the projection matrix. The in-plane rotation changes the camera matrix to

$$\mathbf{K}_r = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\alpha) & sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

where α is the angle of the in-plane rotation. Using Eq. (8) along with (3), the projection model for the RoIs after image rotation is easily tractable. In such a way, deformation r does not only provide constraint to learn the landmark branch, but also can be considered as a way of augmenting data for the learning of the object coordinate branch.

Thanks to the uniqueness of the learned landmarks, they can be matched from two images. The following paragraphs show our method of incorporating the matched 2D-2D correspondences into a multi-view loss term.

Multi-View Loss: With the motivation of introducing multi-view geometry into learning, we upgrade the object coordinate branch to a Siamese configuration as well. Due to the degenerated in-plane 6dof transformation, deformation r no longer suits for constraining consistence of object coordinate. We hence use two images I_s and target I_t – from different viewpoints caused by an out-of-plane movement – as the inputs for the Siamese network.

The proposed multi-view loss for the object coordinate branch consists of two terms. Firstly, we focus on the cross-projection between two viewpoints. Given I_s and I_t as the inputs for the object coordinate branch *and* landmark branch, four outputs are obtained: object coordinate maps $\Phi_{obj}(I_s), \Phi_{obj}(I_t)$ and landmark maps $\Phi_{lm}(I_s), \Phi_{lm}(I_t)$. Pixel-wise matching is performed on these landmark maps.

Denote by $\mathbf{p}^{lm}_{t,l,(h,w)} = M(\Phi_{lm}(I_s), \Phi_{lm}(I_t))$ the matched position of I_s's pixel on I_t, where the M is a matching operation. Given the ground truth pose of the target image $\mathbf{R}^{gt}_t, \mathbf{t}^{gt}_t$ and the scaled camera matrix $\mathbf{K}_t$, the projection of predicted source object points on the target RoI is

$$\mathbf{p}^{proj}_{t,l,(h,w)} = \frac{1}{s}\mathbf{K}_t(\mathbf{R}_t\mathbf{P}^{O}_{s,l,(h,w)} + \mathbf{t}_t). \tag{9}$$

$\mathbf{p}^{proj}_{t,l,(h,w)}$ and $\mathbf{p}^{lm}_{t,l,(h,w)}$ are the position of a same 3D object point on the target RoI. The difference between them is used for back-propagation to learn a 3D coordinate whose projection agrees with the matched position. Thus the first loss term is defined as the landmark alignment loss:

$$L_{lm\_align} = \frac{1}{m \times m}\sum_{h,w}\left\|\mathbf{p}^{proj}_{t,l,(h,w)} - \mathbf{p}^{lm}_{t,l,(h,w)}\right\|_2. \tag{10}$$

Secondly, we propose to encode the multi-view constraints as a photometric loss. Specifically, the projections $\mathbf{p}^{proj}_{t,l,(h,w)}$ warp a reconstructed image $I_{s\leftarrow t}$ from I_t. Any difference that we assume arises from an error in the predicted object coordinates leads to an error in the normalized RGB space. This behavior encodes a photometric loss:

$$L_{rgb} = \frac{1}{m \times m}\sum_{h,w}\left\|I_{s\leftarrow t} - I_s\right\|. \tag{11}$$

Our multi-view geometry-based loss ultimately is $L_{multi} = L_{lm\_align} + L_{rgb}$. The first loss term Eq. (10) ensures that similar landmarks regress to the similar object points and the second loss Eq. (11) term ensures that an object point has the same visual feature on different images. These strong geometric supervisions improve the consistence for the object coordinate regression. Reconstructed results in Fig. 3(c) shows the improvement, in which two sets of object points are well aligned for images from different views.

Inference: See Fig. 1.

3.2. Implementation Details

The backbone for RPN in our implementation is ResNet-50 with Feature Pyramid Network (FPN) [28]. See the details of the detection and segmentation head in [16]. The architecture of our object coordinate branch is shown in Fig.4. We follow the structure of the *SmallNet* in [50, 49] for the landmark branch. We train all the heads in our model simultaneously in and end-to-end fashion with loss $L = L_{cls} + L_{box} + L_{mask} + L_{repro} + L_{multi} + L_{lm}$. The weights for these loss terms are not highly tuned, and are set equally. The network is trained for 200k iterations on a Nvidia Tesla V100 GPU with batch size 2. The schedule for learning rate decay follows [16]. For RANSAC at the test time, the threshold for inliers is set to 1px, and number of hypotheses is 256. The refinement runs up to 100 times.

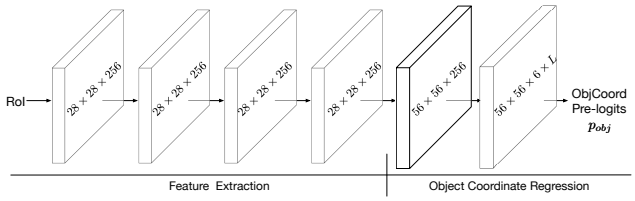


Figure 4. **Object Coordinate Head Architecture.** The feature extractor comprises 4 convolutional layers (conv) with kernel size 3×3 and stride 1. The deconvolutional layer in object coordinate regressor is 2×2 with stride 2. The last conv is 3×3 with stride 1. The final output for object coordinate is $d \times (\text{sigmoid}(p_{obj}) - 0.5)$, where d is the approximated diameter of the object and p_{obj} is the output pre-logits from the last conv.

Figure 5. **The generation of the demo synthetic dataset.** Training and test viewpoints are in red and blue, respectively.

4. Experiments

We first introduce the creation of the dataset we used in previous section. We then conduct ablation studies to investigate the effect of each supervisory signal for the object coordinate head. Thirdly, we compare the reconstruction from our network and the classic reconstruction-based method. At last, we run our methods on the two real world datasets: LINEMOD [18] and Occlusion LINEMOD [18] and compare with the state-of-the-art learning-based methods that require the 3D model in their pipeline.

Expo Dataset: The synthetic dataset contains a square rigid object `expo`. 200 and 2500+ viewpoints are sampled from a sphere for training and test respectively. The locations of the viewpoints are randomized to make sure the object spread over the whole image frame, with various scales. We render the synthetic images using the textured CAD model from these poses. The black background is then replaced with real world images from NYU-Depth V2 [34] dataset. See Fig. 5 for examples.

Metric: The metrics we use to assess the pose estimation performance are ADD-10 and 5cm5deg. ADD is the average 3D distance of model points transformed by the predicted pose and ground truth pose. For symmetric objects, ADD is relaxed to ADD-S, which is the distance between the closest points in two transformed models. If the average (or closest) distance derived by a test pose is less than 10% of the object diameter, the pose estimate is considered correct. As the for 5cm5deg, an estimate is correct when the translation and rotation error is below $(5cm, 5°)$. The numbers we report in Table 1, 2 and 3 are the proportion of frames with correct pose estimates among all test images.

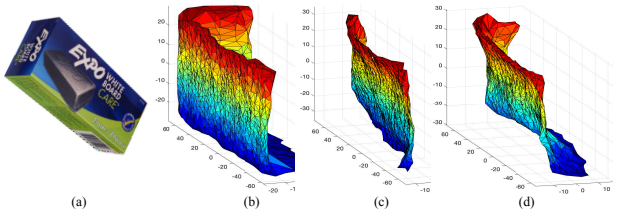


Figure 6. **Visualization of the reconstruction from the object coordinate head.** (a) is a test image. (b) is the true reconstruction from this viewpoint. (c) is the output from the head trained with reprojection loss. (d) is the output from the head trained with multi-view loss in addition to reprojection loss.

Ablations: We train the network using three different supervisions: (i) direct supervision from depths (as a reference); (ii) single-view reprojection loss; (iii) single-view reprojection loss along with multi-view geometry losses. The qualitative meshed visualization of the predicted 3D points from models trained with different losses is shown in Fig. 6. The quantitative results for pose estimation are shown in Table 1.

Fig. 6(b) shows that the true shape of the object from the test viewpoint comprises 3 perpendicular planes. With only the single-view reprojection loss as supervision, the network failed to discover the geometry of the object and predicts a set of points that lies on a plane (See Fig. 6(c)). What is interesting is that these erroneous object coordinates surprisingly result in highly (5cm5deg: 99% and ADD-10: 99.5%) accurate poses for the *training* set. It suggests that optimizing the loss term (5) alone leads the network to a degenerated case that only the ray of the 3D point lies on is decided, rather than a full 3D coordinate. As a result, the trained model produces an arbitrary shape, as long as whose projections from the ground truth pose match the silhouette of the object on the image. Hence the correspondences built by this shape and the 2D positions result in fine pose estimates for training set (the performance on test set is reported later). In contrast, the reconstruction from the model trained with additional multi-view losses shows the corner and the 3-face structure of the object in Fig. 6(d). Quantitatively, the median chamfer distances (two-way, in m, smaller is better) between single-view reconstruction against the groundtruth shape are (0.152, 0.067), and for multi-view reconstruction they are (0.094, 0.048).

The failure caused by using reprojection loss as the only supervision also presents in the quantitative results for the test images. In Table 1(`repro`), the 5cm5deg and ADD-10 accuracy for the model trained with reprojection loss are only 14.3% and 23.6%. This is because that the trained model does not encode the true geometry and therefore generalizes poorly to the unseen images.

In column `repro+lm`, the model is trained with reprojection loss and landmark alignment loss. The accuracy increases to 39.3% (5cm5deg) and 52.5% (ADD-10), which is approximately 2.5 times of `repro`. Combining reprojection loss with photometric loss (column `repro+rgb`) achieves similar results. The best performance comes from the column `rgb+lm+rgb`. It is obtained by training the model with reprojection loss and all multi-view losses ($L_{lm_{align}} + L_{rgb}$). It shows that with additional multi-view constraints provided by the photometric loss, the object coordinate achieves a better pose estimates, which is even comparable with the model from direct supervision, whose accuracy is 61.3% (5cm5deg) and 58.5% (ADD-10).

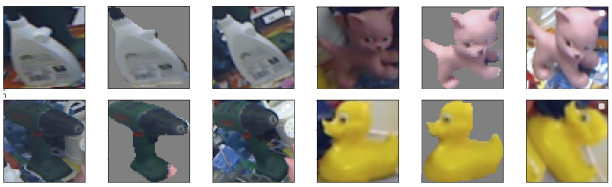


Figure 7. **The reconstruction (middle) of the source (left) by warping the target (right) using matched landmark positions.**

	depth	repro	repro +lm	repro +rgb	repro +lm+rgb
5cm5deg	61.3	14.3	39.3	40.1	53.1
ADD-10	57.1	23.6	52.5	51.3	58.5

Table 1. **The pose estimation performance of different combinations of the loss terms on test set of `expo`.**

Landmark Matching: We show several examples of the dense matching based on the learned object landmarks in two views from LINEMOD in Fig. 7. The positions of the matched landmarks in the source and target images are used to reconstruct the source image. These middle warped images show that the learnt landmark successfully build 2D-2D correspondences in two images which could be used to triangulate the coordinates of the object points in 3D.

Comparison with SfM-based Method: We run SfM using colmap [46, 45] from 200 training images in `expo` datasets to build an explicit reconstruction from the sparse features. Fig. 8 compares the reconstruction from SfM and our object coordinate head. It shows that only five out of the six planes of the object are successfully built by SfM. Apparently it is caused by the lack of textures on the missing plane, where the sparse feature detector struggles to recognize any salient points. In contrast, our model manages to build every surface despite its textureness. Our hypothetical explanation is that the backbone explores both coarse and fine features from multiple scales therefore it is more robust to the density of the visual features on the image. As a trade-off, our method visually practice the accrual of reconstruction error at 3D corners of objects (see Fig. 6(d)), where invariance and equivariance constraints are most "stressed" by out-of-plane motion (also may exhibit self-occlusion).

On LineMOD: Our method is performed on the LINEMOD dataset to verify the generalization to the real world images. LINEMOD contains 13 objects sequences

method	w/ CAD model								w/o CAD model		
	BB8 [41]	BB8 w/ r	SSD-6D w/ r [22]	Tekin [48]	DeepIM w/ r [27]	Dense-Fusion [52]	Pix2-Pose [38]	PVNet w/ r [40]	SSD-6D [22]	LieNet [11]	Ours
ape	27.9	40.4	65	21.62	77.0	92	58.1	43.62	0.00	38.8	52.91
benchwise	62.0	91.8	80	81.80	97.5	93	91.0	99.90	0.18	71.2	96.51
cam	40.1	55.7	78	36.57	93.5	94	60.0	86.86	0.41	52.5	87.84
can	48.1	64.1	86	68.80	96.5	93	84.4	95.47	1.35	86.1	86.81
cat	45.2	62.6	70	41.82	82.1	97	65.0	79.34	0.51	66.2	67.30
driller	58.6	74.4	73	63.51	95.0	87	76.3	96.43	2.58	82.3	88.70
duck	32.8	44.3	66	27.23	77.7	92	43.8	52.58	0.00	32.5	54.74
eggbox*	40.0	57.8	100	69.58	97.1	100	96.8	99.15	8.90	79.4	94.74
glue*	27.0	41.2	100	80.02	99.4	100	79.4	95.66	0.00	63.7	91.98
holepuncher	42.4	67.2	49	42.63	52.8	92	74.8	81.92	0.30	56.4	75.41
iron	67.0	84.7	78	74.97	98.3	97	83.4	98.88	8.86	65.1	94.59
lamp	39.9	76.5	73	71.11	97.5	95	82.0	99.33	8.20	89.4	96.64
phone	35.2	54.0	79	47.74	87.7	93	45.0	92.41	0.18	65.0	89.24
average	43.6	62.7	79	55.95	88.6	94	72.4	86.27	2.42	65.2	82.88

Table 2. **LineMOD: Percentages of correct pose estimates in ADD-10**. * denotes that the object is symmetric and is evaluated in ADD-S. w/r means the pose is refined with 3D model.

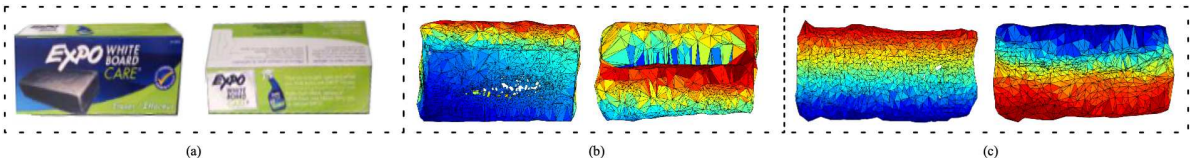


Figure 8. **Comparison between the reconstructions from SfM and our method**. Left: images from two example viewpoints; Middle: meshed reconstructions from SfM; Right: meshed reconstructions from our model.

with annotated bounding box and pose for the interest object. We train our network strictly following the training/test split in [48]. No additional synthetic data is required, as well as the 3D CAD model in our method. We report the performance in Table 2. Our method outperforms more than half of the learning-based methods and achieves comparable result with the state-of-the-art method, which use a large amount of synthetic training images from new viewpoints [40] and/or 3D model for refinement [52, 27].

On Occlusion LINEMOD: We also test our approach on a more challenging dataset: Occlusion LINEMOD, a sequence with annotations for occluded objects. ADD-10 results are shown in Table 3 following the test scheme of [40]. It shows the robustness of our method to occlusion.

	Tekin [48]	Pose-CNN [54]	Ober-weger [35]	PV-Net [40]	Pix2-Pose [38]	Ours
ape	2.48	9.6	17.6	15.8	22.0	7.1
can	17.48	45.2	59.3	63.3	44.7	40.6
cat	0.67	0.93	3.31	16.7	22.7	15.6
driller	7.66	41.4	62.4	25.2	44.7	43.9
duck	1.14	19.6	19.2	65.7	15.0	12.9
ebox*	-	22.0	25.9	50.1	25.2	46.4
glue*	10.08	38.5	39.6	49.6	32.4	51.7
holp.	5.54	22.1	21.3	39.7	49.5	24.5
avg.	6.42	24.9	30.4	40.8	32.0	30.3

Table 3. **Results on Occlusion LINEMOD**. Note that all the methods requires the 3D model in the pipeline except ours.

5. Conclusion

We have proposed an method that performs accurate 6dof object pose estimation from a single RGB image. Our learning-based method implicitly encodes the object reconstruction into a network by regressing object pixel to 3D object coordinate. It then carries out 2D-3D correspondences for geometric pose solving at inference time. The learning of the network explicitly enforces the multi-view geometric constraints for the object coordinates. The additional landmark branch provides consistence for objects across multiple views. We explore self-supervision for learning from image deformation and eliminates the need of 3D model in the system. Our 3D model free method reduced the performance gap between approaches with and without 3D model.

Acknowledgement

We gratefully acknowledge the support of the Australian Research Council through the Centre of Excellence for Robotic Vision CE140100016 and Laureate Fellowship FL130100102 to IR.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, Oct. 2011. 1, 3

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

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. 2

[4] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[5] Eric Brachmann and Carsten Rother. Learning Less Is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018. 2

[6] Mai Bui, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab. Scene coordinate and correspondence learning for image-based localization. *arXiv preprint arXiv:1805.08443*, 2018. 2

[7] Mai Bui, Sergey Zakharov, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab. When regression meets manifold learning for object recognition and pose estimation. 2018. 1

[8] Ming Cai, Huangying Zhan, Chamara Saroj Weerasekera, Kejie Li, and Ian Reid. Camera relocalization by exploiting multi-view constraints for scene coordinates regression. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1

[10] Daniel F. Dementhon and Larry S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1):123–141, Jun 1995. 1

[11] Thanh-Toan Do, Trung Pham, Ming Cai, and Ian D. Reid. Lienet: Real-time monocular object instance 6d pose estimation. In *British Machine Vision Conference 2018, BMVC 2018*, page 2, 2018. 8

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 2

[13] Iryna Gordon and David G Lowe. What and where: 3d object recognition with accurate pose. In *Toward category-level object recognition*, pages 67–82. Springer, 2006. 2

[14] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *European Conference on Computer Vision*, pages 408–421. Springer, 2010. 3

[15] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 3, 6

[17] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390, Nov 2011. 2

[18] 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 *Computer Vision – ACCV 2012*, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 1, 2, 3, 6

[19] Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4421–4428. IEEE, 2015. 3

[20] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *CVPR*, 2019. 1, 3

[21] Jie Tang, S. Miller, A. Singh, and P. Abbeel. A textured object recognition pipeline for color and depth image data. In *2012 IEEE International Conference on Robotics and Automation*, pages 3467–3474, May 2012. 2

[22] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 3, 8

[23] Alex Kendall and Roberto Cipolla. Modelling Uncertainty in Deep Learning for Camera Relocalization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016. 3

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, 2015. 2, 3

[25] Vincent Lepetit, Pascal Fua, et al. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 1(1):1–89, 2005. 1

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

[27] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 3, 8

[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 conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[30] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[31] Eric Marchand, Patrick Bouthemy, François Chaumette, and Valérie Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 1, pages 262–268. IEEE, 1999. 1

[32] Manuel Martinez, Alvaro Collet, and Siddhartha S Srinivasa. Moped: A scalable and low latency object recognition and pose estimation system. In *2010 IEEE International Conference on Robotics and Automation*, pages 2043–2049. IEEE, 2010. 2

[33] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 2

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

[35] 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 (ECCV)*, 2018. 3, 8

[36] Qi Pan, Gerhard Reitmayr, and Tom Drummond. Proforma: Probabilistic feature-based on-line rapid model acquisition. In *BMVC*, volume 2, page 6. Citeseer, 2009. 1, 2

[37] Qi Pan, Gerhard Reitmayr, Edward Rosten, and Tom Drummond. Rapid 3d modelling from live video. In *The 33rd International Convention MIPRO*, pages 252–257. IEEE, 2010. 1, 2

[38] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 1, 3, 8

[39] Karl Pauwels, Leonardo Rubio, Javier Diaz, and Eduardo Ros. Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 1

[40] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 3, 8

[41] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017. 1, 3, 8

[42] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. 3

[43] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006. 2

[44] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011. 2

[45] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[46] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 7

[47] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[48] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3, 8

[49] J. Thewlis, S. Albanie, H. Bilen, and A. Vedaldi. Unsupervised learning of landmarks by exchanging descriptor vectors. In *International Conference on Computer Vision*, 2019. 5, 6

[50] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5916–5925, 2017. 5, 6

[51] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*, pages 125–134. IEEE Computer Society, 2008. 1, 2

[52] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 8

[53] Ping Wang, Guili Xu, Zhengsheng Wang, and Yuehua Cheng. An efficient solution to the perspective-three-point pose problem. *Comput. Vis. Image Underst.*, 166(C):81–87, Jan. 2018. 2

[54] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems (RSS)*, 2018. 1, 3, 8

[55] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3D objects. In *British Machine Vision Conference (BMVC)*, 2019. 1, 3

[56] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, Aug 2003. 2

[57] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2), Oct 1994. 1