# 3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings

Giorgia Pitteri, Aurélie Bugeau, Slobodan Ilic, Vincent Lepetit

# 3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings

Giorgia Pitteri[1], Aurélie Bugeau[1], Slobodan Ilic[3], and Vincent Lepetit[2]

[1] Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France {giorgia.pitteri,aurelie.bugeau}@u-bordeaux.fr
[2] LIGM, IMAGINE, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallee, France
vincent.lepetit@enpc.fr
[3] Siemens Corporate Technology, Munich, Germany
sobodan.ilic@siemens.com

**Abstract.** We present an approach for detecting and estimating the 3D poses of objects in images that requires only an untextured CAD model and no training phase for new objects. Our approach combines Deep Learning and 3D geometry: It relies on an embedding of local 3D geometry to match the CAD models to the input images. For points at the surface of objects, this embedding can be computed directly from the CAD model; for image locations, we learn to predict it from the image itself. This establishes correspondences between 3D points on the CAD model and 2D locations of the input images. However, many of these correspondences are ambiguous as many points may have similar local geometries. We show that we can use Mask-RCNN in a class-agnostic way to detect the new objects without retraining and thus drastically limit the number of possible correspondences. We can then robustly estimate a 3D pose from these discriminative correspondences using a RANSAC-like algorithm. We demonstrate the performance of this approach on the T-LESS dataset, by using a small number of objects to learn the embedding and testing it on the other objects. Our experiments show that our method is on par or better than previous methods.

## 1 Introduction

Deep Learning (DL) provides powerful techniques to estimate the 6D pose of an object from color images, and impressive results have been achieved over the last years [1,2,3,4,5,6], including in the presence of occlusions [7,8,9], in the absence of texture, and for objects with symmetries (which create pose ambiguities) [10,11]. However, most of recent works focus on supervised approaches, which require that for each new object, these methods have to be retrained on many different registered images of this object. Even if domain transfer methods allow for training such methods on synthetic images instead of real ones at least to some extent, such training sessions take time, and avoiding them is highly appealing.

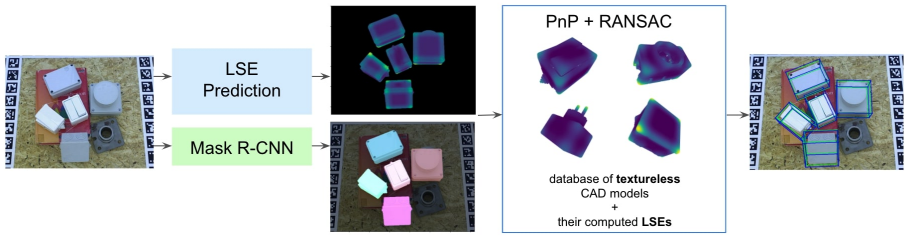



**Fig. 1.** Overview of our method. We detect and estimate the 3D poses of objects, given only an untextured CAD model, without having to retrain a deep model for these objects. Given an input RGB image, we predict local surface embeddings (LSEs) for each pixel that we match with the LSEs of 3D points on the CAD models. We then use a P$n$P algorithm and RANSAC to estimate the 3D poses from these correspondences. We use the predicted masks to constrain the correspondences in a RANSAC sample to lie on the same object, in order to control the complexity. The LSE prediction network is trained on known objects but generalizes well to new objects. Similarly, we train Mask-RCNN on known objects, and use mask R-CNN to segment the objects in the image. Because we train Mask-RCNN in a *class-agnostic* way, it also generalizes to new objects without retraining. Note that we use masks of different colors for visualization only.

Recently, a few methods were proposed for 3D pose estimation for objects that were not seen during training, only exploiting an untextured CAD model provided for the new objects. This is an important problem in industrial contexts, but also very challenging as aligning an untextured CAD model to a color image remains very difficult, especially without pose prior. In DeepIM [12], the authors propose a pose refiner able to perform such alignment given some initial pose, however it has been demonstrated on very simple synthetic images with constant lighting. [13] proposes to learn to detect corners by using training images of a small set of objects, and estimates the object pose by robustly matching the corners of the CAD model with the corners detected in the input image. However, this requires the object to have specific corners and a skilled user to select the corners on the CAD model. Very recently, [14] proposes a single-encoder-multi-decoder network to predict the 6D pose of even unseen objects. Their encoder can learn an interleaved encoding where general features can be shared across multiple instances. This leads to encodings that can represent object orientations from novel, untrained instances, even when they belong to untrained categories. Even if the idea is promising, to achieve competitive results, they need to use depth information and refine the pose with an ICP algorithm.

In this paper, we investigate 6D object pose estimation in an industrial scenario with the challenges this implies: We want to handle symmetrical, textureless, ambiguous, and unseen objects, given only their CAD models. By contrast with some previous works, we also do not assume that the ground truth 2D bounding boxes for the objects are available. As shown in Figure 1, our approach

combines machine learning and 3D geometry: Like previous works [15,6,11,16], we establish dense correspondences between the image locations and 3D points on the CAD model, as they showed that this yields to accurate poses. However, there is a fundamental difference between these works and ours: They can train a machine learning model in advance to predict the 3D coordinates of the pixels in a given image. In our case, we want to avoid any training phase for new objects. We therefore rely on a different strategy: We introduce an embedding capturing the local geometry of the 3D points lying on the object surface. Given a training set for a small number of objects, we learn to predict these embeddings per pixel for images of new objects. By matching these embeddings with the embeddings computed for 3D points on the object surface, we get 2D-3D correspondences from which we estimate the object's 6D pose using RANSAC and a P$n$P solver.

This approach is conceptually simple, robust to occlusions, and provides an accurate 6D pose. However, to be successful, some special care is needed. First, the embeddings need to be rotation invariant. Second, because of the symmetries and this rotation invariance, many correspondences between pixels and 3D points are possible *a priori* and the complexity of finding a set of correct correspondences can become exponential. We control this complexity in two ways. We focus on image locations with the most discriminative embeddings as they have less potential correspondences. We also observe that Mask R-CNN [17] is able to predict the masks of new objects when trained without any class information, and thus can segment new objects without re-training. We use this to constrain the sets of correspondences in RANSAC to lie on the same mask, and thus drastically decrease the number of samples to consider in RANSAC.

In the remainder of the paper, we review the state-of-the-art on 3D object pose estimation from images, describe our method, and evaluate it on the T-LESS dataset [18], which is made of very challenging objects and sequences.

## 2 Related Work

In this section, we first review recent works on 3D object detection and pose estimation from color images. We also review methods for using synthetic images for training as it is a popular solution for 3D pose estimation. Finally, we review the few works that consider the same problem as us.

### 2.1 3D Object Detection and Pose Estimation from Color Images

The use of Deep Learning has recently significantly improved the performance of 6D pose estimation algorithms. Different general approaches have been proposed. One approach is to first estimate 2D bounding boxes for the visible objects, and predict the 6D pose of each object directly from the image region in the bounding box [1,2,3,5,10]. The pose can be predicted directly using quaternions for the rotation example, or via 3D points or the reprojections of 3D points related to the object, or by learning a code book using AutoEncoders. This last method has the advantage to work well with symmetrical objects, which are common in

industrial contexts. Another approach, aiming to be more robust to occlusions, is to predict for each pixel offsets to the reprojections of 3D points related to the object [7,19,20].

Closer to our own approach, several works first predict for each pixel its 3D coordinates in the object's coordinate frame [21,15,4,22,6,11,23,16,24]. This yields 2D-3D correspondences from which the object's 3D pose can be estimated using a P$n$P algorithm, possibly together with RANSAC for more robustness. In our case, we cannot directly predict the 3D coordinates of pixels as it can be done only for the objects or categories used for training. Instead, we learn to predict an embedding for the 3D local geometry corresponding to each pixel, and we rely on this embedding to match the pixel to its corresponding 3D point on the CAD models of new objects.

### 2.2 Training on synthetic images for 6D pose estimation

One popular approach to 6D pose estimation given only a CAD model and no, or few, real training images is to exploit synthetic images. There is however a domain gap between real and synthetic images, which has to be considered to make sure the method generalizes well to real images.

A very simple approach is to train a convolutional network for some problem such as 2D detection on real images and use the first part of the network for extracting image features [25,1]. Then, a network taking these features as input can be trained on synthetic images. This is easy to do, but it is not clear how many layers should be used exactly. Generative Adversarial Networks (GANs) [26] and Domain Transfer have been used to make synthetic images more realistic [27,28,29,30,31,32,33,34,35]. Another interesting approach is domain randomization [36], which generates synthetic training images with random appearance by applying drastic variations to the object textures and the rendering parameters to improve generalization.

These works can exploit CAD models for learning to detect new objects, however they also require a training phase for new objects. In this work, we do not need such phase.

### 2.3 6D pose estimation without retraining

Very few recent works already tackled 6D pose estimation without retraining for new objects. One early approach targeting texture-less objects is to rely on templates [37]. Deep Learning has also been applied to such problem, by learning to compute a descriptor from pairs or triplets of object images [38,39,40,41]. Like ours, these approaches do not require re-training, as it only requires to compute the descriptors for images of the new objects. However, it requires many images from points of view sampled around the object. It may be possible to use synthetic images, but then, some domain transfer has to be performed. But the main drawback of this approach is the lack of robustness to partial occlusions, as the descriptor is computed for whole images of objects. It is also not clear how it would handle ambiguities, as it is based on metric learning on

images. In fact, such approach has been demonstrated on the LineMod, which is made of relatively simple objects, and never on the T-LESS dataset, which is much more challenging.

More recently, DeepIM proposed a pose refiner able to refine a given initial pose. In [12], this refiner was applied to new objects, but only on very simple synthetic images with constant lighting. [13] proposes to learn to detect corners by using training images of a small set of objects and estimates the object pose by robustly matching the corners of the CAD model with the corners detected in the input image. This method requires objects to have specific corners and to offline select corners on the CAD model. Even more recently, [14] proposes an extension of [10] able to generalize to new objects. Thanks to the single-encoder-multi-decoder architecture, they are able to learn an interleaved encoding where general features can be shared across multiple instances of novel categories. To achieve competitive results, they need to use depth information and refine the pose with an ICP algorithm.

Our approach is related to [13], but considers any 3D location on the objects to get matches, not only corners. We compare against [13] and [14] in the experimental section.

## 3 Method

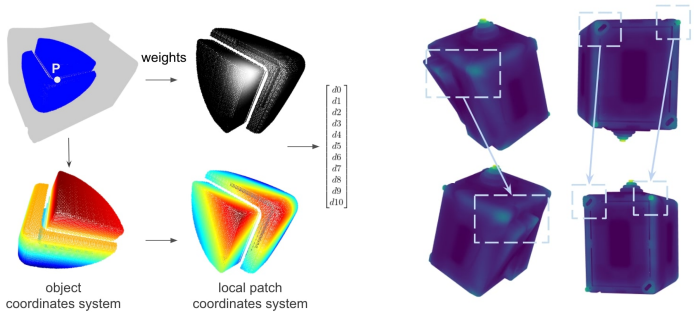
We describe our approach in this section. We first explain how we compute the local surface embeddings and how we obtain correspondences between the CAD models and the images. We then describe our pose estimation algorithm.

### 3.1 Local Surface Embeddings

To match new images with CAD models, we rely on embeddings of the local surfaces of the objects. To be able to match these embeddings under unknown poses, they need to be translation invariant and rotation invariant. Achieving translation invariance is straightforward, since we consider the local geometry centered on 3D points. Achieving rotation invariance is more subtle, especially because of ambiguities arising in practice with symmetrical objects. This is illustrated in Figure 2(b): We need to compute the same embeddings for local geometries that are similar up to a 3D rotation.

More exactly, given a 3D point $\mathbf{P}$ on the surface of an object, we define the local geometry as the set of 3D points $\mathbf{M}_n$ in a spherical neighborhood centered on $\mathbf{P}$ and of radius $r$. In practice, on T-LESS, we use $r = 3cm$. To compute a rotation-invariant embedding, we transform these points from the object coordinate system to a local patch coordinate system using a rotation matrix computed from the decomposition of the covariance matrix of the 3D points $\mathbf{M}_n$ after centering on $\mathbf{P}$ [42]:

$$C = \sum_n \mathbf{v}_n \cdot \mathbf{v}_n^\top \,, \tag{1}$$




(a) (b)

**Fig. 2.** (a): Computation of the LSEs for a given point **P** on a CAD model. We transform the 3D points in the neighborhood of **P** into a rotation-invariant local system and weight them before computing their moments. (b): Visualization of the rotation-invariance property on different parts of the same object. Similar local geometries yield similar LSEs. Through this paper, we represent the LSEs using only their 3 first values mapped to the red, green, blue channels except for Figure 3 that shows all the values.

where $\mathbf{v}_n = (\mathbf{M}_n - \mathbf{P})$ using a Singular Value Decomposition (SVD):

$$C = L^\top \Sigma R\,. \tag{2}$$

$R$ is an orthogonal matrix, but not necessarily a rotation matrix, and small differences in the local geometry can result in very different values for $R$. We therefore apply a transformation to $R$ to obtain a new matrix $\bar{R}$ so that $\bar{R}$ is a suitable rotation matrix. It can be checked that applying $\bar{R}$ to the $\mathbf{v}_i$ vectors will achieve rotation invariance for the local surface embeddings.

Let's denote by $r_1$, $r_2$, and $r_3$ the rows of $R$, and by $\bar{r}_1$, $\bar{r}_2$, and $\bar{r}_3$ the rows of $\bar{R}$. Applying $R$ to the normal $n$ of the object's surface at $\mathbf{P}$ yields a 3-vector $R\cdot n$ close to either $[0,0,1]^\top$ or $[0,0,-1]^\top$, depending on the orientation of $R$ selected for the SVD. For normalisation, we choose that $\bar{R}\cdot n$ should always be closer to $[0,0,1]^\top$. We therefore compute $o = r_3^\top \cdot n$. If $o$ is positive, we take $\bar{r}_3 = r_3$, otherwise we take $\bar{r}_3 = -r_3$. As a result, $\bar{R}\cdot n$ is always closer to $[0,0,1]^\top$ that to $[0,0,-1]^\top$. Finally, we take $\bar{r}_1 = r_1$ and $\bar{r}_2 = -\bar{r}_1 \wedge \bar{r}_3$, where $\wedge$ denotes the cross-product, which ensures that $\bar{R}$ is a rotation matrix.

We explain now how we define the local surface embeddings. For our experiments, we use the local moments of the local 3D points for simplicity but any other embeddings such as [43] could also work. Let us denote by $[x_n, y_n, z_n]$ the vectors $\bar{R}\mathbf{v}_n$, then local surface embeddings can be computed as:

$$\mathrm{LSE}_{i,j,k}(\mathbf{P}) = \sum_n w_n x_n^i y_n^j z_n^k\,, \tag{3}$$

where $w_n = \exp(-\|\mathbf{v}_n\|^2/\sigma^2)$ is a weight associated to each point based on its distance from $\mathbf{P}$ (we use $\sigma = 5$ in practice) and $i, j, k$ are exponents in the range

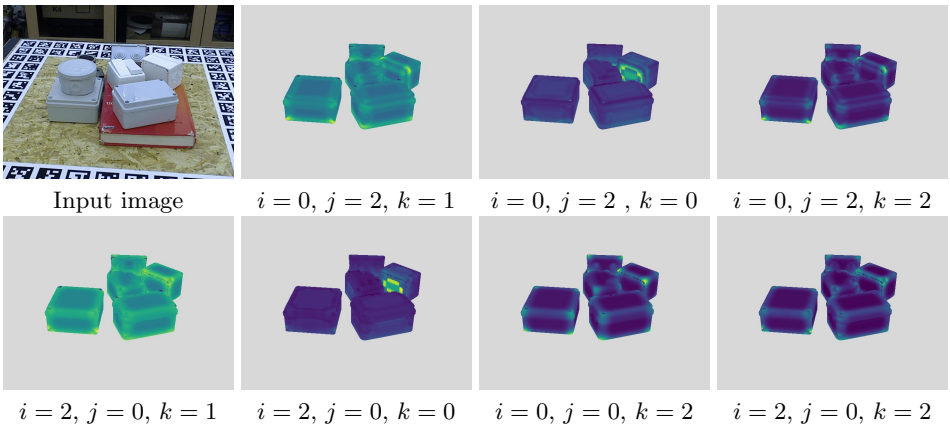



**Fig. 3.** Visualization of some LSEs coordinates for an example image.

$[0, 1, 2]$. Theoretically it is possible to take all the combinations of exponents but we empirically found that the most discriminative values are computed using: $i \in \{0, 2\}$, $j \in \{0, 2\}$, $k \in \{0, 1, 2\}$, which gives 11 values for the full vector LSE($\mathbf{P}$) as taking $i = j = k = 0$ gives a constant value and is not useful. Finally, we normalize the values of LSE($\mathbf{P}$) to zero mean and unit variance so they have similar ranges. Figure 3 displays the embeddings for an example image.

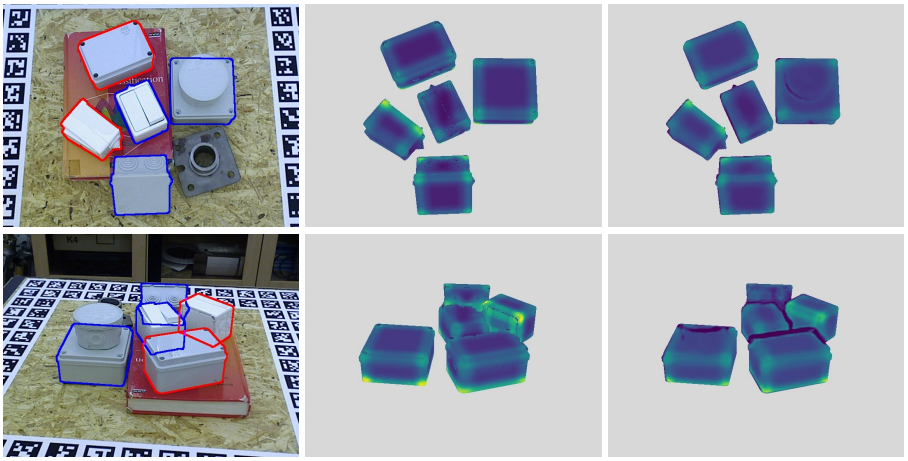
### 3.2 Predicting the local surface embeddings for new images

Given a new CAD model, it is trivial to compute the local surface embeddings on points on its surface. Given a new input image, we would like to also compute the embeddings for the object points visible in this image. We use a deep network to perform this task. To do so, we create a training set by generating many synthetic images of known objects under various poses. We also compute the LSEs for all the pixels corresponding to a 3D point of one of the objects. We then train a U-Net-like architecture [44] to predict the LSEs given a color image. More details on the architecture and its training are provided in the experimental section.

This training is done once, on known objects, but because the embeddings depend only on the local geometry, the network generalizes well to new objects, as shown in Figure 4.

### 3.3 Pose Estimation Algorithm

The pseudocode for our detection and pose estimation algorithm is given as Algorithm 1. Given a new image, we compute the LSEs for each of its pixels using the network described in Section 3.2 and establish correspondences between image pixels and object 3D points. However, the number of possible correspondences can quickly become very large, which would yield a combinatorial explosion in



(a) (b) (c)

**Fig. 4.** Generalization of the LSE prediction network to new objects. (a) Input RGB image with objects seen during the training of the network (blue boundaries) and new objects (red boundaries). The LSE predictions (c) are close to the LSE Ground truth (b) for both the known and new objects.

the number of set of correspondences needed in RANSAC. We control the complexity in two different ways.

First, we focus on the most discriminative embeddings. Points on planar regions are very common and would generate many correspondences. We discard them by thresholding the embedding values: Points with very low absolute embedding values for the LSEs are removed. Figure 5 (a) shows how pixels are selected.

Second, we force the correspondences in each sample considered by RANSAC to belong to the same object. Even when objects are not known in advance, it is possible to segment them: To do so, we use Mask-RCNN [17] to predict the masks of the objects. We fine-tuned it on our synthetic images already used for training the LSE predictor, as described in Section 3.2 in a class-agnostic way since we want to generalize to new objects. We found out that it works very well with new objects even for cluttered backgrounds, as shown in Figure 6. This also allows us to easily discard pixels on the background from the possible correspondences.

We match the embeddings predicted for the pixels of the input image against the embeddings computed for the 3D points on the CAD model based on their Euclidean distances. In our implementation, we use the FLANN library [45] to efficiently get the $k$ nearest neighbors of a query embedding. In practice, we use $k = 100$. This usually returns points in several clusters, as close points tend to have similar embeddings. We therefore go through the list of nearest neighbors sorted by increasing distances. We keep the first 3D point and remove from the

**Algorithm 1** Pose estimation algorithm.

$\mathcal{C} \leftarrow$ CAD models for the new objects
$\mathcal{E}(C) \leftarrow \text{LSE}_{\text{CAD}}(C)$, the LSEs of 3D points for each CAD model $C$
$I \leftarrow$ input image
$\mathcal{F} \leftarrow \text{LSE}_{\text{pred}}(I)$, the predicted LSEs for the input image
$\mathcal{O} \leftarrow$ Mask-RCNN$(I)$, the masks predicted by Mask-RCNN
$\mathcal{M} \leftarrow \{m_i\}_i$, the set of 2D-3D matches based on $\mathcal{E}(C)$ and $\mathcal{F}$. Each match $m_i$ is made of an image location $\mathbf{p}$ and 3D points on the CAD models: $(\mathbf{p}, [\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_{m_i}])$

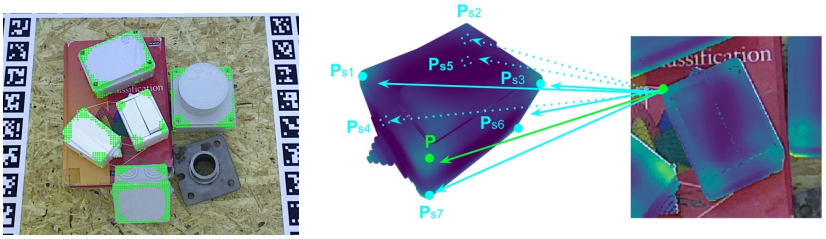
**procedure** POSE_ESTIMATION_O_C($O$, $C$)
    $s_{\text{best}} \leftarrow 0$
    **for** iter $\in [0; N_{\text{iter}}]$ **do**
        $n \leftarrow$ random integer in $[6; 10]$
        $M \leftarrow n$ random correspondences $(\mathbf{p}, \mathbf{P})$,
            where $\mathbf{p} \in O$ and $\mathbf{P}$ is matched to $\mathbf{p}$ in $\mathcal{M}$
        **pose** $\leftarrow$ PNP$(M)$
        $s \leftarrow$ SCORE(**pose**$, C, \mathcal{E}(C), \mathcal{F}, O)$
        **if** $s > s_{\text{best}}$ **then**
            $\textbf{pose}_{\text{best}} \leftarrow$ **pose**
            $s_{\text{best}} \leftarrow s$
    Refine $\textbf{pose}_{\text{best}}$
    **return** $\textbf{pose}_{\text{best}}$, SCORE($\textbf{pose}_{\text{best}}, \mathcal{E}(C), \mathcal{F}, O, C$)

**procedure** POSE_ESTIMATION
    **for** each mask $O \in \mathcal{O}$ **do**
        $\triangleright$ $s_{\min}$ is the minimum score for a match with a CAD model:
        $s_{\text{best}}(O) \leftarrow s_{\min}$
        **for** each CAD model $C$ **do**
            **pose**$, s \leftarrow$ POSE_ESTIMATION_O_C($O$, $C$)
            **if** $s > s_{\text{best}}$ **then**
                $s_{\text{best}} \leftarrow s$
                $\textbf{pose}_{\text{best}}(O) \leftarrow$ **pose**
                $C_{\text{best}}(O) \leftarrow C$

list the other points that are also close to this point, and we iterate. This provides for each pixel a list of potential corresponding 3D points separated from each other.

When working on industrial objects like the ones in T-LESS, some pixels can be matched with several 3D points, as shown in Figure 5(b), because of the rotation invariance property of the local LSEs and the similarities between local parts of different objects.

We finally use LO-RANSAC (Locally Optimized RANSAC) [46] with a P$n$P algorithm (we use [47] followed by a Levenberg-Marquardt optimization) to compute the poses of the visible objects. We take random $n \in [6; 10]$ for each RANSAC sample. At each iteration, we compute a score for the predicted pose as a weighted sum of the Intersection-over-Union between the mask from Mask-




(a) (b)

**Fig. 5.** Focusing on the most discriminative pixels. (a): In green, pixels with discriminative LSEs. We only consider them for correspondences with the CAD models. (b): A pixel can be matched with multiple 3D points on symmetrical objects because of the rotation invariance property of the LSEs.

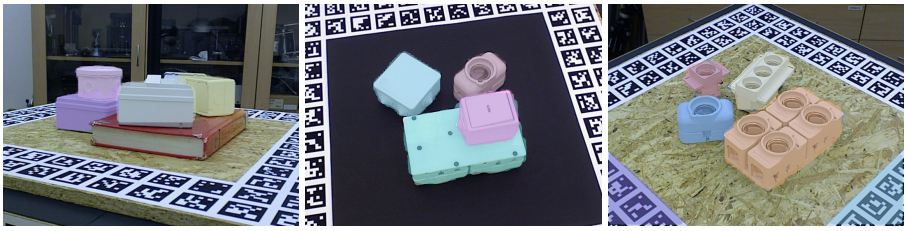


**Fig. 6.** Generalization of Mask-RCNN to unknown objects. We train Mask-RCNN in a class-agnostic way on a set of known objects. It generalizes well to new objects, and we use these masks to constrain the pose estimation. Note that we use masks of different colors for visualization only. Mask-RCNN cannot identify the new objects individually as it was not trained on them, it can only detect objects in a class-agnostic way.

RCNN and the mask obtained by rendering the model under the estimated pose, and the Euclidean distances between the predicted LSEs and the LSEs for the CAD model after reprojection. We keep the pose with the largest score and refine it using all the inlier correspondences to obtain the final 6D pose.

## 4 Evaluation

In this section, we present and discuss the results of our pose estimation algorithm on the challenging T-LESS dataset [18], made of texture-less, ambiguous and symmetrical objects with no disciminative parts. It is well representative of the problems encountered in industrial context.

### 4.1 Dataset

To train our LSE prediction network, we generate synthetic images using the CAD models provided with T-LESS for a subset of objects in this dataset. The

exact subset depends on the experiment, and we will detail them below. Similar to the *BlenderProc4BOP* [48] introduced in the BOP challenge [49], these images are created with Cycles, a photorealistic rendering engine of the open source software Blender by randomly placing the training objects in a simple scene made of a plane randomly textured and randomly lighted. We used both these synthetics and real images for training the network combined with data augmentation to take care of the domain gap between our synthetic images and the real test images. More specifically, we use $15K$ synthetic images and $\sim 7K$ real images—all the training images provided by T-LESS for the objects that are used for training the LSE prediction and Mask-RCNN. To create the ground truth embeddings, for each training image, we backproject the pixels lying on the objects to obtain their corresponding 3D points and their LSEs. Neither the embedding prediction network nor Mask-RCNN see the test objects during training.

### 4.2 LSE prediction network architecture and training

The architecture of the network predicting the LSEs for a given input image is a standard U-Net-like [44] encoder-decoder convolutional neural network taking a $720 \times 540$ RGB image as input. The encoder part is a 12-layer ResNet-like [50] architecture; the decoder upsample the feature maps up to the original size using bilinear interpolations followed by convolutional layers. We train the network with the Adam optimizer and a learning rate set to $10^{-4}$. We also use batch normalization to ensure good convergence of the model. Finally, the batch size is set to 8 and we train the network for 150 epochs.

### 4.3 Metrics

We evaluate our method using several metrics from the literature. Analogously to other related papers [6,2,13,3], we consider the percentage of correctly predicted poses for each sequence and each object of interest, where a pose is considered correct based on the ADD metric (or the ADI metric for symmetrical objects) [37]. This metric is based on the average distance in 3D between the model points after applying the ground truth pose and the estimated one. A pose is considered correct if the distance is less than 10% of the object's diameter.

Following the BOP benchmark [49], we also report the *Visible Surface Discrepancy* (VSD) metric. The VSD metric evaluates the pose error in a way that is invariant to the pose ambiguities due to object symmetries. It is computed from the distance between the estimated and ground truth visible object surfaces in the following way:

$$err_{\text{VSD}}(\hat{S}, \bar{S}, S_I, \hat{V}, \bar{V}, \tau) = \underset{p \in \hat{V} \bigcup \bar{V}}{\text{Mean}} \begin{cases} 0, & \text{if p} \in \hat{V} \bigcap \bar{V} \wedge |\hat{S}(p) - \bar{S}(p)| < \tau \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where $\hat{S}$ and $\bar{S}$ are distance maps obtained by rendering the object model in the estimated and ground-truth poses respectively. The distance maps are compared

| Scene | 02 | 03 | 04 | 04 | 06 | 08 | 10 | 11 | 11 | 12 | 12 | 13 | 15 | 15 | 14 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Obj | 7 | 8 | 26 | 8 | 7 | 20 | 20 | 8 | 9 | 7 | 9 | 20 | 29 | 26 | 20 | Avg |
| [13] | **68.3** | **57.9** | 28.1 | 21.2 | 36.8 | 10.0 | 27.8 | **58.8** | - | 23.1 | - | 26.6 | 48.0 | - | 10.0 | 34.7(±18.5) |
| **Ours** | 61.0 | 44.1 | **55.6** | **39.1** | **44.8** | **38.2** | **38.3** | 40.8 | **46.1** | **41.2** | **45.8** | **39.5** | **77.0** | **63.6** | **24.9** | **46.7** (±12.0) |

**Table 1.** Our quantitative results on T-LESS test Scenes #02, #03, #04, #06, #08, #10, #11, #12, #13, #14, #15 as used in [13]. We report results also for Objects #9 in Scenes #11 and #12 and for Object #26 in Scene #15 even though [13] does not. See text for details.

with the distance map $S_I$ of the test image $I$ to obtain the visibility masks $\hat{V}$ and $\bar{V}$, *i.e.* the sets of pixels where the object model is visible in image $I$. We report the mean VSD recall of 6D object poses at $err_{\text{VSD}} < 0.3$ with tolerance $\tau = 20mm$ and $> 10\%$ object visibility.

### 4.4 Results

The complexity of the test scenes in T-LESS varies from several isolated objects on a clean background to very challenging ones with multiple instances of several objects with a high amount of occlusions and clutters. We compare our method against the two works that already consider 6D object detection and pose estimation for unknown objects on T-LESS, CorNet [13] and the MP-Encoder method of [14]. As the codes for these two methods are not available at the time of writing, we use the same protocols as in these works and report the results from the papers.

*Comparison with CorNet [13].* We use here the same protocol as in [13]: We split the objects from T-LESS into two sets: One set of known objects (#6, #19, #25, #27, and #28) and one set of unknown objects (#7, #8, #20, #26, and #29), and we compare the 3D detection and pose estimation performance of our method and CorNet for the unknown objects in T-LESS test scenes #02, #03, #04, #06, #08, #10, #11, #12, #13, #14, and #15. We use synthetic images of the known objects for training the LSE prediction network. Results are reported in Table 1. We outperform CorNet on most of the objects, except on objects #7 and #8. This is because these objects have some 3D points with local geometry very different from the training objects (at the connections of the different parts). As a result, the predicted LSEs for these parts are not very accurate, generating wrong matches. Figured 8 shows some qualitative results for the unknown objects in the test images.

*Comparison with MP-Encoder [14].* We use here the same protocol as in [14]: The objects from T-LESS are split into a set of known objects (#1-#18) and one set of unknown objects (#19-#30), and we compare the 3D detection and pose estimation performance of our method and MP-Encoder for the unknown objects in T-LESS test scenes following the BOP benchmark [49]. We use synthetic images of the known objects for training the LSE prediction network. Note that

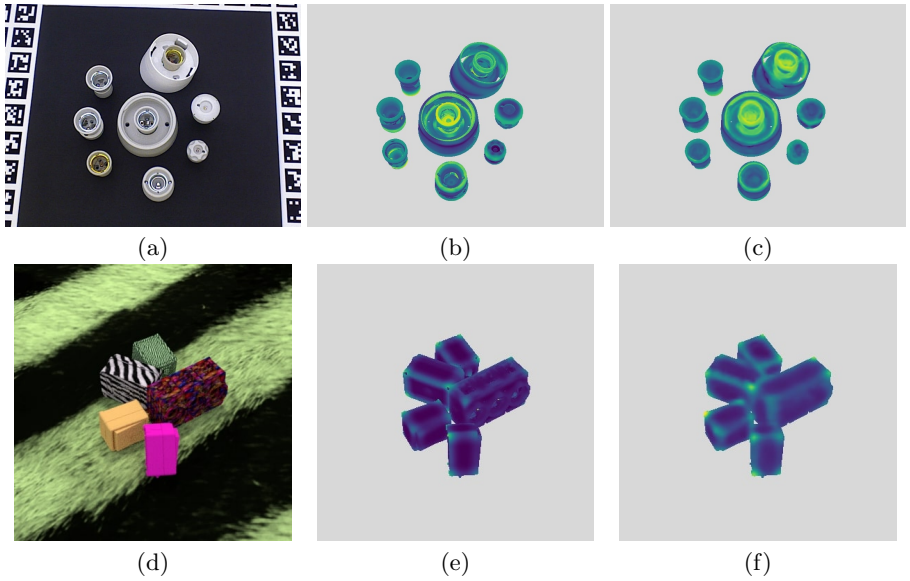



**Fig. 7.** Top row: Image with objects with rounded shapes (a). Ground truth and predicted LSEs (b) and (c). Bottom row: Image with random textures applied to some T-LESS objects (d). Ground truth and predicted LSEs (e) and (f).

| | VSD recall |
|---|---|
| MP-Encoder [14] | 20.53 |
| **Ours** | **23.27** |

**Table 2.** Mean *Visible Surface Discrepancy* (VSD) recall using the protocol of [14]. This metric evaluates the pose error in a way that is invariant to the pose ambiguities due to object symmetries. It is computed from the distance between the estimated and ground truth visible object surfaces.

we report here the number of Table 3 from the [14] paper as the other reported results assume that the ground truth bounding boxes, the ground truth masks, or depth information are provided. Results are reported in Table 2. While our method performs slightly better, the performances are close and tells us that both methods are promising. The main difference is that the MP-Encoder relies on an embedding completely learnt by a network while our method incorporates some geometrical meaning that makes our approach more appealing for industrial purposes. Note that, as shown in Fig. 7, our LSE network can handle objects with rounded shape without being limited to objects with prominent corners as [13].

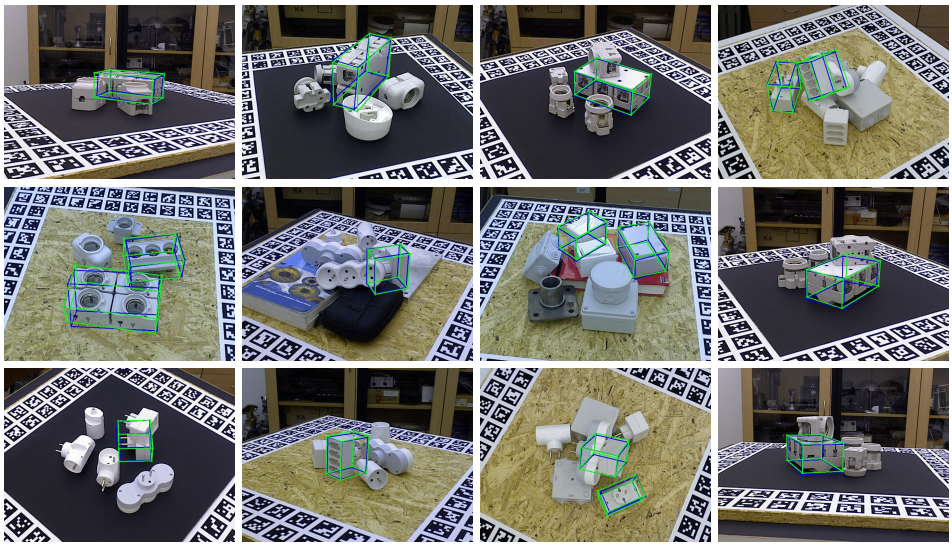


**Fig. 8.** Qualitative results on the unknown objects of the test scenes from T-LESS. Green bounding boxes denote ground truth poses, blue bounding boxes correspond to our predicted poses.

### 4.5 Robustness to Texture

Our focus is on untextured objects, as industrial objects typically do not exhibit textures like the T-LESS objects. However, our LSEs can be predicted for textured objects as well. To show this, we retrained our LSE prediction network on synthetic images of the T-LESS objects rendered with random textures. The LSEs prediction for some test images are shown in the bottom row of Fig. 7.

## 5 Conclusion

We introduced a novel approach for the detection and the 3D pose estimation of industrial objects in color images. It only requires the CAD models of the objects and no retraining is needed for new objects. We introduce a new type of embedding capturing the local geometry of the 3D points lying on the object surface and we train a network to predict these embeddings per pixel for images of new objects. From these local surface embeddings, we establish correspondences and obtain the pose with a P$n$P+RANSAC algorithm. Describing the local geometries of the objects allows to generalize to new categories and the rotation invariance of our embeddings makes the method able to solve typical ambiguities that raise with industrial and symmetrical objects. We believe that using local and rotation invariance descriptors is the key to solve the 6D pose of new textureless objects from color images.

## References


1. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In: International Conference on Computer Vision. (2017)
2. Rad, M., Lepetit, V.: BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In: International Conference on Computer Vision. (2017)
3. Tekin, B., Sinha, S.N., Fua, P.: Real-Time Seamless Single Shot 6D Object Pose Prediction. In: Conference on Computer Vision and Pattern Recognition. (2018)
4. Jafari, O.H., Mustikovela, S.K., Pertsch, K., Brachmann, E., Rother, C.: IPose: Instance-Aware 6D Pose Estimation of Partly Occluded Objects. CoRR **abs/1712.01924** (2017)
5. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. Robotics: Science and Systems Conference (2018)
6. Zakharov, S., Shugurov, I., Ilic, S.: DPOD: Dense 6D Pose Object Detector and Refiner. In: International Conference on Computer Vision. (2019)
7. Oberweger, M., Rad, M., Lepetit, V.: Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In: European Conference on Computer Vision. (2018)
8. Peng, S., Liu, Y., Huang, Q., Bao, H., Zhou, X.: PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. CoRR **abs/1812.11788** (2018)
9. Hu, Y., Fua, P., Wang, W., Salzmann, M.: Single-Stage 6D Object Pose Estimation. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020)
10. Sundermeyer, M., Marton, Z.C., Durner, M., Triebel, R.: Augmented Autoencoders: Implicit 3D Orientation Learning for 6D Object Detection. International Journal of Computer Vision **128** (2020) 714–729
11. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 7668–7677
12. Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: Deep Iterative Matching for 6D Pose Estimation. In: European Conference on Computer Vision. (2018) 683–698
13. Pitteri, G., Lepetit, V., Ilic, S.: CorNet: Generic 3D Corners for 6D Pose Estimation of New Objects without Retraining. In: International Conference on Computer Vision Workshops. (2019)
14. Sundermeyer, M., Durner, M., Puang, E.Y., Marton, Z.C., Vaskevicius, N., Arras, K.O., Triebel, R.: Multi-path Learning for Object Pose Estimation Across Domains. In: Conference on Computer Vision and Pattern Recognition. (2020)
15. Brachmann, E., Michel, F., Krull, A., Yang, M.M., Gumhold, S., Rother, C.: Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In: Conference on Computer Vision and Pattern Recognition. (2016)
16. Wang, H., Sridhar, S., Valentin, J.H.J., Song, S., Guibas, L.J.: Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In: Conference on Computer Vision and Pattern Recognition. (2019)
17. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: International Conference on Computer Vision. (2017)

18. Hodan, T., Haluza, P., Obdrzalek, S., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects. In: IEEE Winter Conference on Applications of Computer Vision. (2017)
19. Hu, Y., Hugonot, J., Fua, P., Salzmann, M.: Segmentation-Driven 6D Object Pose Estimation. In: Conference on Computer Vision and Pattern Recognition. (2019) 3385–3394
20. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In: Conference on Computer Vision and Pattern Recognition. (2019) 4561–4570
21. Taylor, J., Shotton, J., Sharp, T., Fitzgibbon, A.: The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. In: Conference on Computer Vision and Pattern Recognition. (2012) 103–110
22. Wang, Y., Tan, X., Yang, Y., Liu, X., Ding, E., Zhou, F., Davis, L.S.: 3d pose estimation for fine-grained object categories. In: European Conference on Computer Vision Workshops. (2018)
23. : (cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation)
24. Hodan, T., Barath, D., Matas, J.: EPOS: Estimating 6D Pose of Objects with Symmetries. arXiv preprint arXiv:2004.00605 (2020)
25. Hinterstoisser, S., Lepetit, V., Wohlhart, P., Konolige, K.: On Pre-Trained Image Features and Synthetic Images for Deep Learning. In: arXiv. (2017)
26. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems. (2014)
27. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain Separation Networks. In: Advances in Neural Information Processing Systems. (2016) 343–351
28. Müller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In: Conference on Computer Vision and Pattern Recognition. (2018)
29. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In: Conference on Computer Vision and Pattern Recognition. (2017)
30. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In: International Conference on Computer Vision. (2017)
31. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-Adversarial Training of Neural Networks. Journal of Machine Learning Research (2016)
32. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning Transferable Features with Deep Adaptation Networks. In: International Conference on Machine Learning. (2015)
33. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous Deep Transfer Across Domains and Tasks. In: International Conference on Computer Vision. (2015)
34. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse Image-To-Image Translation via Disentangled Representations. In: European Conference on Computer Vision. (2018)
35. Zakharov, S., Planche, B., Wu, Z., Hutter, A., Kosch, H., Ilic, S.: Keep It Unreal: Bridging the Realism Gap for 2.5D Recognition with Geometry Priors Only. In: International Conference on 3D Vision. (2018)

36. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In: International Conference on Intelligent Robots and Systems. (2017)
37. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient Response Maps for Real-Time Detection of Textureless Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012)
38. Wohlhart, P., Lepetit, V.: Learning Descriptors for Object Recognition and 3D Pose Estimation. In: Conference on Computer Vision and Pattern Recognition. (2015)
39. Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R., Kim, T.K.: Pose Guided RGBD Feature Learning for 3D Object Pose Estimation. In: International Conference on Computer Vision. (2017)
40. Zakharov, S., Kehl, W., Planche, B., Hutter, A., Ilic, S.: 3D Object Instance Recognition and Pose Estimation Using Triplet Loss with Dynamic Margin. In: International Conference on Intelligent Robots and Systems. (2017)
41. Bui, M., Zakharov, S., Albarqouni, S., Ilic, S., Navab, N.: When Regression Meets Manifold Learning for Object Recognition and Pose Estimation. In: International Conference on Robotics and Automation. (2018)
42. Eggert, D., Lorusso, A., Fisher, R.: Estimating 3D Rigid Body Transformations: A Comparison of Four Major Algorithms. Machine Vision and Applications **9** (1997) 272–290
43. Deng, H., Birdal, T., Slobodan, I.: PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. In: European Conference on Computer Vision. (2018)
44. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Conference on Medical Image Computing and Computer Assisted Intervention. (2015)
45. Muja, M., Lowe, D.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: International Conference on Computer Vision. (2009)
46. Chum, O., Matas, J., Kittler, J.: Locally Optimized RANSAC. In: German Conference on Pattern Recognition. (2003)
47. Lepetit, V., Moreno-noguer, F., Fua, P.: EP$n$P: An Accurate $o(n)$ Solution to the P$n$P Problem. International Journal of Computer Vision (2009)
48. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
49. Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al.: BOP: Benchmark for 6D Object Pose Estimation. In: European Conference on Computer Vision. (2018) 19–34
50. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Conference on Computer Vision and Pattern Recognition. (2016)