6 DoF Pose Estimation of Textureless Objects from Multiple RGB Frames

Roman Kaskman[1,2], Ivan Shugurov[1,2], Sergey Zakharov[1,2], and Slobodan Ilic[1,2]

[1] Technical University of Munich
{roman.kaskman, ivan.shugurov, sergey.zakharov}@tum.de
[2] Siemens Corporate Technology
slobodan.ilic@siemens.com

Abstract. This paper addresses the problems of object detection and 6 DoF pose estimation from a sequence of RGB images. Our deep learning-based approach uses only synthetic non-textured 3D CAD models for training and has no access to the images from the target domain. The image sequence is used to obtain a sparse 3D reconstruction of the scene via Structure from Motion. The domain gap is closed by relying on the intuition that geometric edges are the only prominent features that can be extracted from both the 3D models and the sparse reconstructions. Based on this assumption, we have developed a domain-invariant data preparation scheme and 3DKeypointNet, which is a neural network for detecting of the 3D keypoints in sparse and noisy point clouds. The final pose is estimated with RANSAC and a scale-aware point cloud alignment method. The proposed method has been tested on the T-LESS dataset and compared to methods also trained on synthetic data. The results indicate the potential of our method despite the fact that the entire pipeline is solely trained on synthetic data.

Keywords: Pose Estimation, Object Detection, Sparse Point Clouds

1 Introduction

The problem of object detection and pose estimation of 3D objects has been addressed for an infinite number of times in the field of computer vision. As demonstrated in the BOP Challenge [24] for 6 DoF object pose estimation, deep learning dominates if only monocular RGB images are used, while traditional geometry-based methods [14] are still not giving up in depth and RGBD images. On the other hand, deep learning methods operating on unorganized point clouds are massively used in autonomous driving, with one of the task being car detection and pose estimation [45, 32, 9, 29, 61, 65]. This problem is better posed than generic object pose estimation due to physical constraints. For instance, cars cannot be in arbitrary poses; sizes and scales do not vary much, while in general cases, an object can be placed completely arbitrarily. Moreover, obtaining labels for 6 DoF pose estimation is extremely difficult. Therefore, training detectors from synthetic renderings of 3D models is desirable. Unfortunately,

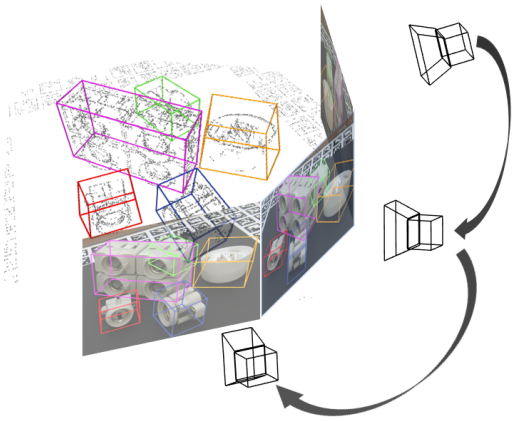


Fig. 1: **Illustration of the proposed method.** The proposed method uses SfM to obtain a sparse point cloud reconstruction from a frame sequence. Then, object poses are estimated in 3D, after which per-frame poses are computed.

they still have lower performance than detectors trained on real data. However, [26] showed that detectors trained on synthetic data do not overfit to particular datasets and seem to be more generic when real data from the target domain (labeled and unlabeled) are not available.

We propose herein a method for 6 DoF pose detection in RGB images, which is trained solely on synthetic data. We address this problem for textureless 3D objects (i.e. objects from the T-LESS dataset [21] as in Figure 1, which do not have distinctive textures) whose 3D CAD models are available. The CAD models come without any color or texture information associated with them, which is typical in industrial scenarios where non-textured CAD models are available. Renderings of such models appear as grayscale models on arbitrary backgrounds, making training on them more challenging because of the domain gap. As shown in [43, 62], classical GAN approaches [5, 17, 56, 36, 6, 30] depending on the data from the target domain fail in such scenarios. Unlike other related approaches that use a single RGB image for detection, we use an entire sequence of RGB images as illustrated in Figure 1. Our intuition was to rely on SfM and perform a sparse reconstruction of the scenes. This point cloud reconstruction is then approximated with an edge-like reconstruction by fitting edge segments to the geometric discontinuities in the scene. We designed a neural network that detects the pre-selected keypoints for each object in the sparse reconstruction. Knowing correspondences between these 3D keypoints and their locations on the CAD models, the pose is determined using Umeyama algorithm [57]. We developed an exhaustive data generation scheme that remains easy to use and requires no extra artistic modeling effort, to close the domain gap between Structure from Motion (SfM) reconstructions and CAD models. It starts by generating a large number of 3D scenes using physics-based simulation. Since geometric edges are the only features robust to light changes and present in non-textured CAD models, we represent each CAD model with an edge-like representation

obtained by the method of Hofer *et al.* [25]. This representation is then used to replace 3D models in simulated scenes with the corresponding edge-based models. The same method has been applied to SfM reconstructions to obtain "edgy" scene representations. In this way, the synthetic and real domains have a smaller discrepancy, which allows our keypoint detection network to detect object keypoints, which are fed to RANSAC and the Umeyama algorithm to output the object poses. Due to scale ambiguity of SfM reconstructions, we perform pose refinement both in 3D with scale-aware ICP (directly on sparse SfM reconstructions, no real depth information is used) and in the image using multi-view edge-based alignment. Our main contributions are summarized as follows:

1. a pipeline for recovering full 6 DoF object poses from sequences of RGB images;
2. domain-invariant data preparation allowing for training from non-textured CAD models; and
3. a 3D keypoint prediction network allowing for 6 DoF object detection and pose estimation in point clouds.

We evaluated our method on the challenging T-LESS [21] dataset and compared it with the best-performing synthetic methods on the recent BOP 6 DoF pose recovery challenge [23].

2 Related Work

As two of the most classical problems in computer vision with a wide range of applications in different domains, object detection and pose estimation have been extensively studied in the past. In this section, we present a summary of the relevant past works on this topic, ranging from classical methods to modern deep learning approaches. Our method spans between RGB approaches and point cloud based methods; hence, we are going to discuss related methods from both categories.

2.1 DL methods for RGB images

Deep learning and convolutional neural networks allow for feature extraction even from surfaces that do not exhibit pronounced textures, thereby making it possible to estimate a pose directly from RGB images.

Training on synthetic data. SSD6D [27] paved the way for CNN-based pose estimation approaches that rely solely on synthetic training data. The detector extends the standard SSD detector [34] to predict a discrete approximation of the rotational component of the transformation. The drawback of the approach is that it is slow, and poses estimated without ICP refinement are extremely rough. Moreover, a confidence threshold must be chosen separately for each object to achieve a good recall. The idea of discrete viewpoint approximation was further revisited by the Augmented Autoencoder (AAE) [52, 53].

The AAE approach consists of two disconnected stages. The first stage is a 2D object detector trained on real data. At the second stage, each detected object is transformed with a neural network to output a descriptor, which is matched with a dictionary of pre-computed descriptors of all discrete viewpoints. The pre-comuted descriptors are obtained using synthetic renderings of available 3D models possessing texture information. The AAE achieves state of the art results on the T-LESS dataset [21]. DPOD [64] estimates dense correspondences between the 3D model and its instance in the scene. This method can be trained both on real and synthetic data, and achieves good results in both cases. When training on synthetic data, we would like to mention methods using GAN networks [5, 54, 51, 1]. These approaches typically start from source domain images (e.g., synthetic renderings of textured objects) and then alter them to resemble the target domain images (e.g., coming from the real camera). For this, the unlabeled images from the target domain must be available. On the contrary, the approaches presented in [43, 63] rely on the exhaustive domain randomization applied to synthetic images, which are then fed to a GAN network. The GAN denoises them and makes them look similar to the synthetic images used for training the target classifier and the pose estimation network. DeceptionNet [62] elevates the problem of blind data randomization and performs network-driven domain randomization that generalized well to new domains. However, the domain adaptation methods need object detection to be performed separately as a preprocessing step. Moreover, they are applied separately to the object patches, which is a simpler task than simultaneous detection and pose estimation.

Training on real data. The following methods utilize the prediction of 2D-3D correspondences, that are used with PnP [31] to directly compute an object's pose. BB8 [48] proposed the prediction of the location of the projections of the object's 3D bounding box corners. The idea was extended further in YOLO6D [55]. The multi-stage procedure was replaced with a direct one-shot detection with the YOLO detector [49]. YOLO was additionally augmented to simultaneously regress the projection of the 3D bounding box. As a result, YOLO6D has superior performance than BB8 both in terms of runtime and pose quality. Alternatively, CorNet [41] learns to predict generic corners which are then matched to the actual model corners.

Another group of methods relies on semantic segmentation rather than on detection. All methods in this group perform pixel-wise segmentation in order to increase the quantity of 2D-3D correspondences and improve robustness to occlusions. In Pixel-wise Voting Network (PVNet) [40]. each foreground pixel votes for the location of a predefined set of keypoints. This is similar to our approach, but ours is performed on sparse reconstructions represented with a point cloud. Two segmentation-based approaches, that are very similar to each other, are Pix2Pose [38] and DPOD [64]. In contrast to Hough voting for a fixed number of key points in PVNet, both Pix2Pose and DPOD treat each foreground pixel as a key point and predict a 2D-3D correspondence pair for each pixel. CDPN [33] combines a segmentation-based PnP method and direct pose regression. In [59], it is proposed using a two-stage approach. In the first stage,

a standard semantic segmentation network is used to localize pixels belonging to the objects of interest. At the second stage, a Pointnet [46, 47] is applied to the detected objects to estimate per-pixel pose hypothesis via direct pose regression. An alternative line of research focuses on how to properly deal with the ambiguities caused by symmetries [42, 37].

Even though deep learning RGB methods have been extensively studied in the past, as described earlier, they all target pose estimation from a single image. Therefore, they are not directly applicable to the described multi-view scenario.

2.2 Methods for depth images and point clouds

Hand-crafted features still perform best when it comes to generic pose estimation in depth images. The leading method based on Point Pair Features (PPF) [14] and its numerous extensions [20, 3, 58] are still among the best-performing methods. However, despite its effectiveness, PPF is not appropriate for our tasks because no reliable normals can be estimated from a sparse point cloud. Another problem is that SfM reconstructions are defined up to a scale, and using them directly will be difficult because PPF are not scale-invariant.

PointNet [46, 47] paved the way for feature extraction from raw point clouds for numerous object detection approaches in point clouds [45, 32, 44]. Even though these approaches are aimed at solving a related task, they are not directly applicable because they are targeted toward the type of challenges found in the KITTI dataset [18]. First, the KITTI dataset has a very limited number of unique object classes. Second, all objects are in the upright position and exhibit rotations only around a single axis. Lastly, all these approaches use real data for training. The most relevant paper to our approach is VoteNet [44]. In [44], it was proposed using semantic segmentation of point clouds and Hough voting for object detection and pose estimation. However, VoteNet does not use Hough votes to directly estimate the pose and does not use geometric constraints. Votes are only used for point cloud clustering and region proposal generation. Our approach differs from [44] in several ways. Our detector votes for multiple key points rather than only for the object's center. Each keypoint has a fixed location on the CAD model. The object pose can then be directly estimated from the votes and the known correspondences using the Umeyama algorithm [57].

3 Method

This section provides an in-depth description of all the components of the proposed method. The section starts with an overview of the pipeline: 1) synthetic data preparation; 2) the proposed network architecture, including the loss functions; 3) the inference stage of the proposed detector; and 4) implementation and training details.

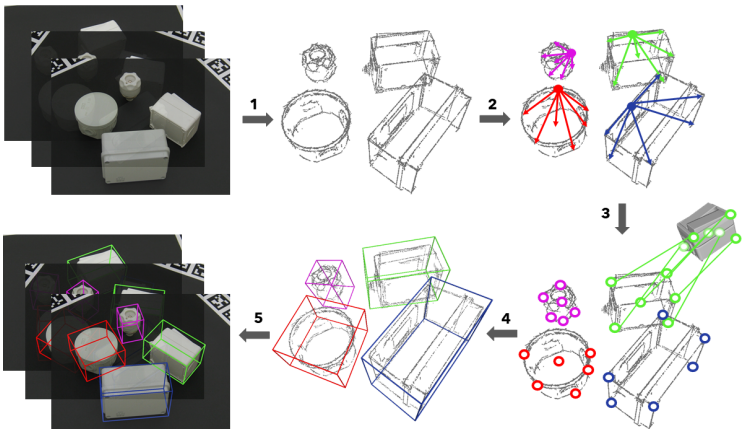


Fig. 2: **Illustration of the inference steps of our method**. 1) sparse reconstructions are obtained from a sequence of RGB images using COLMAP [50] and Lined3Dpp [25]; 2) seed points belonging to different objects vote for the keypoints. 3) the keypoints are localized, and the correspondences between a CAD model and an object in the scene are established; 4) poses of the objects in the sparse scene are estimated based on model-to-scene correspondences; and 5) poses of the objects in the sparse scene and camera poses from the COLMAP are used to recalculate the object poses in each frame.

3.1 General Pipeline

We deal herein with the problems of object detection and pose estimation of 3D objects, whose non-textured CAD models are known, in RGB images. In contrast to the other RGB-based deep learning approaches described in Section 2, which operate on individual frames, the proposed approach takes advantage of using multiple consecutive frames and jointly estimates poses for all of them. Figure 2 illustrates the overall idea of the presented approach comprising five key steps. 1) A sequence of RGB images (no depth information is used) is fed to the COLMAP SfM algorithm [50]. In this way, a sparse reconstruction of the scene is obtained, which is then approximated with the lines [25]. 2) 3DKeypointNet, which is based on PointNet++, is used to detect the 3D keypoints belonging to the objects. Each point belonging to the object in the scene votes for the locations of a predefined set of 3D keypoints of this particular object. 3) With votes at hand, the 3D locations of keypoints are estimated separately for each object with the RANSAC scheme [16]. 4) Object poses can be computed in closed form with the Umeyama algorithm [57] given the known 3D-3D correspondences between the detected keypoints and their location on the CAD model. At this stage, the object poses in the scene coordinate system are available. 5) Per-frame poses are recomputed and refined to better account for the pose errors stemming from SfM reconstructions defined up to a scale.

3.2 Domain-Invariant Data Preparation

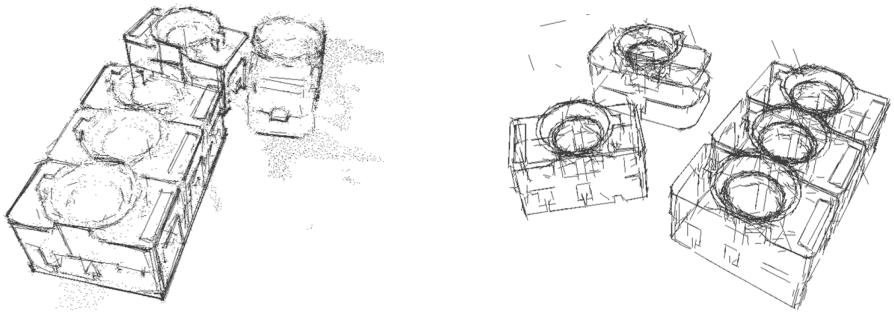
Our goal is to train the detector solely on sparse synthetic data. First, synthetic data are simpler and cheaper to obtain than real data. Second, given a CAD model, it is possible to generate infinitely many images or simulations in contrast to limited sizes of manually labeled real datasets. In spite of the conceptual advantages of synthetic data, special care must be taken to bridge the domain gap. We solve this problem by assuming that 3D geometric edges are invariant to light changes; therefore, they can be reliably extracted from synthetic train and real test data. In both cases, we use the method of Hofer et al. [25] to replace a sparse 3D scene with its line representation.

The data preparation consists of three stages: 1) model preparation to represent the given CAD models with lines; 2) randomized simulation of synthetic 3D scenes and their post-processing; 3) approximation of the generated scenes with lines.

Model preparation. The first step of the training data preparation is the pre-processing of the provided CAD models. Each CAD model is rendered in 1296 poses sampled from a sphere around the object. They are then used to reconstruct prominent 3D edges of the model, which are visible in RGB images, using [25]. The obtained reconstruction is essentially a sparse representation of a model, with geometric features that can be observed in both the captured RGB images of the object and in its synthetic renderings (see Figure 3). Subsequently, a set of K keypoints is computed for each sparse model. The first selected keypoint is the object's center. The other points are obtained via the furthest point sampling of the model vertices.

Simulation of synthetic 3D scenes. The sparse nature of the test data also affects the training data generation pipeline, which consists of three steps. In the first step, synthetic training sequences are randomly simulated with the Bullet Physics Simulation engine [10]. The library generates scenes by randomly dropping objects on planar surfaces, which are sampled from the ScanNet dataset [11] to introduce more variability and make the scenes more realistic. The original dense CAD models are used in this stage. Each generated scene is saved together with poses of all the objects in the scene coordinate system. Figure 4 (a) depicts an example of such a dense scene. These scenes are not yet suitable for training because they are not sparse and contain all the object points, even the invisible ones. The second step of the data preparation pipeline addresses these issues. The view-based model sampling method of Birdal et al. [4] is applied to make the simulated scenes resemble the reconstructions from a frame sequence. A random number of camera poses (i.e., between 2 and 8) located nearby on the upper hemisphere encapsulating the synthetic scene mesh is selected, followed by the removal of the invisible faces of the mesh for each pose. Consequently, this results in a mesh seen from a set of sampled viewpoints with removed invisible faces and respective vertices. However, the mesh is still dense.

Approximation of synthetic scenes with lines. In the third and last step of data preparation, a sparse representation of the scene is computed using the poses obtained during the physical simulation and the sparse CAD models



(a) (b)

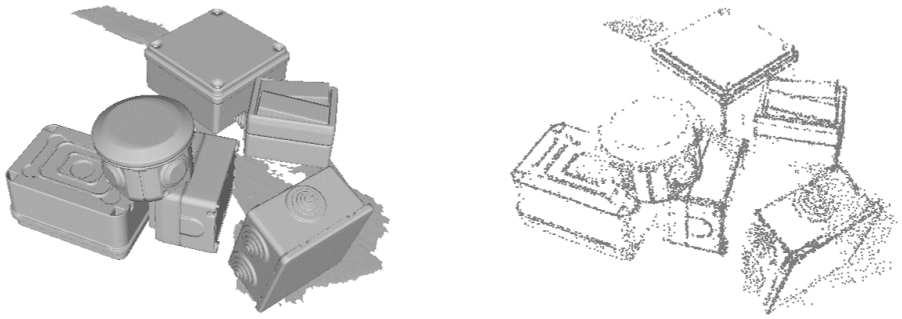
Fig. 3: **Comparison of the synthetic and real edge reconstructions:** (a) synthetic scene composed of edges and (b) real scene composed of edges.

represented with 3D edges. The effect of the view-based sampling is propagated onto the sparse scene. A lookup table is created for each pair of dense and sparse CAD models. It associates the sparse model edges with the nearest vertices on the dense model. The vertices, which were removed during the view-based sampling, are subsequently used to remove the corresponding model edges in the sparse scenes. Finally, the sparse scenes are stored as a set of vertices and 3D edges, with a respective class label assigned for each of the vertices. Figure 4(b) depicts an example of a sparse scene is presented. The keypoints of each model, transformed with their poses in the scene coordinate system, are saved.

3.3 3DKeypointNet - Keypoint Localization Network

While there are volumetric approaches for object detection and pose estimation [2], it was more natural for the given task to opt for the point cloud-based methods [46, 47] because the data are already represented as sparse points and edges between them. This data representation can easily be converted into the point cloud format by sampling points from the existing edges.

The network architecture is based on the PointNet++ [47]. Its backbone architecture has several set abstraction levels followed by upsampling layers utilizing skip connections, which facilitates learning local features with increasing contextual scales. For a set of points $\{\mathbf{p}_i\}_{i=1}^{N}$, $\mathbf{p}_i \in \mathbb{R}^3$, the backbone module outputs M seed points with corresponding features of dimension D, namely $\{\mathbf{s}_i\}_{i=1}^{M}$, where $\mathbf{s}_i = [\mathbf{p}_i; \mathbf{f}_i]$, $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{f}_i \in \mathbb{R}^D$. The seed features $\mathbf{f}_i$ are then passed to the voting module composed of a shared multilayer perceptron with the ReLU activation function and batch normalization. For each seed point, the network outputs class confidences $\{\mathbf{c}_i\}_{i=1}^{M}$, where $\mathbf{c}_i \in \mathbb{R}^C$ and C is a number of classes, and estimated keypoint directions $\{\mathbf{D_i}\}_{i=1}^{M}$, where $\mathbf{D}_i \in \mathbb{R}^{K\times 3}$ is composed of row-wise normalized vectors $\mathbf{d}_j$ estimating the directions toward keypoints $\mathbf{k}_j \in \mathbb{R}^3$ on an object a seed point belongs to. The approach is aimed at making the network learn to estimate a relative position of a seed point in



(a) (b)

Fig. 4: **Illustration of synthetic training data:** (a) synthetic scene composed of dense meshes and (b) resulting sparse scene after view-based sampling

a more global context based on the information extracted from local neighborhoods of different sizes.

The following loss function is optimized during training:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{dir}, \tag{1}$$

where $\mathcal{L}_{cls}$ is a cross-entropy classification loss and $\mathcal{L}_{dir}$ is a direction regression loss defined as

$$\mathcal{L}_{dir} = \sum_{i=1}^{M} \sum_{j=1}^{K} smooth_{L_1}(\mathbf{d}_{ij}, \hat{\mathbf{d}}_{ij}) \mathbb{1}[\mathbf{p}_i \text{ is on object}], \tag{2}$$

where $\hat{\mathbf{d}}_{ij} \in \mathbb{R}^3$ is a ground truth normalized direction vector from a seed point $\mathbf{p}_i$ to a keypoint $\mathbf{k}_j$ of an object, namely $\hat{\mathbf{d}}_{ij} = \frac{\mathbf{p}_i - \mathbf{k}_j}{\|\mathbf{p}_i - \mathbf{k}_j\|_2}$, and $\lambda \in \mathbb{R}$ is a weighting scalar. The smooth L_1 loss is defined as in [34]. The indicator function $\mathbb{1}[\mathbf{p}_i \text{ is on object}]$ is used to eliminate objects on the background.

3.4 Inference

A sequence of RGB images $\{I_1, ..., I_n\}$ is taken as the initial input for inference. Accordingly, sequentially captured images were used in our experiment. The sequence is then processed with the COLMAP SfM method [50] that jointly estimates the camera poses $\mathbf{\Xi}_i \in \mathrm{SE}(3)$ for each image and sparse 3D points of the scene seen in the image sequence. As in the case of the training data, the method of Hofer *et al.* [25] is used to obtain a more detailed reconstruction of the edges in 3D.

It needs to be taken into account at the inference time that the reconstruction and the camera poses obtained through the SfM are defined up to a scale because no ground truth depth information or known camera locations are used in our method. The input 3D edges are scaled by a scalar $s = \frac{\mu_{train}}{d}$, where μ_{train} is

the mean diameter of the training samples, and d is the diameter of the current test scene. Moreover, PCA whitening is applied to the reconstructed scenes to center and axis-align them, which increases their resemblance to the synthetic scenes used during training. The points are then sampled on the 3D edges with the average density used during the training time. The resulting point cloud is fed as an input to the detector. The network outputs per-seed point classification labels as well as estimation of the keypoint direction vectors, which are further used for the RANSAC-based voting for the object keypoint locations, similar to PVNet [40]. For each object class $c \in \{1, ..., C\}$ and for each object keypoint $\mathbf{k}_j$, $j \in \{1, ..., K\}$, a set of H keypoint location proposals is created by randomly sampling tuples of three seed points with the direction vectors. These seed points and directions define lines in 3D. Accordingly, a potential keypoint candidate $\mathbf{k}^*_{hcj}$ is defined as an approximate intersection of these three lines, which we define herein as an optimization problem that looks for the closest point to all those lines in terms of Euclidean distance

$$\mathbf{k}^*_{hcj} = \arg\min_{\mathbf{k}_{hcj}} \sum_{i \in \mathcal{I}} \|\mathbf{p}_{ij} + \lambda_{ij}\mathbf{d}_{ij} - \mathbf{k}_{hcj}\|_2^2 , \tag{3}$$

where $\mathcal{I}$ is an index set of the selected seed points. The solutions to Equation 3 are computed in closed form. All the other seed points belonging to the object of a particular class then vote for the computed keypoint candidates. If the dot product between the estimated vote vector $\mathbf{d}_j(\mathbf{p})$ of a seed point and the computed direction vector from a seed point $\mathbf{p}$ to a candidate $\mathbf{k}^*_{hcj}$ is above the threshold of θ, then the seed point is counted as an inlier:

$$\sum_{\mathbf{p} \in O_c} \mathbb{1}\left[\frac{(\mathbf{k}^*_{hcj} - \mathbf{p})^T}{\left\| \mathbf{k}^*_{hcj} - \mathbf{p} \right\|_2} \mathbf{d}_j(\mathbf{p}) \geq \theta \right] \tag{4}$$

Finally, all the inlier seed points are used to estimate the keypoints $\mathbf{k}^*_{hcj}$ using Equation 3. The procedure can be executed several times to generate multiple hypotheses for each keypoint. In practice, the predicted keypoints are relatively unreliable because of data sparsity and the seed points sampling done internally in PointNet. The problem has been circumvented by running PointNet multiple times with random re-sampling, as was also done in [35], to produce more keypoint hypotheses.

Having a set of correspondence hypotheses between the estimated keypoints $\{\mathbf{k}^*_{hcj}\}_{j=1}^K$ and the keypoints on the canonical model of an object $\{\mathbf{k}_{hcj}\}_{j=1}^K$ for an object of class c it becomes possible to estimate a similarity transformation $\mathbf{S} \in \mathrm{Sim}(3)$,

$$\mathbf{S} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \text{ with } \mathbf{R} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3 \text{ and } s \in \mathbb{R}^+, \tag{5}$$

which transforms and scales from the object's coordinate system to the scene point cloud coordinate system. This transformation can be estimated from the point correspondences in closed form using the Umeyama algorithm [57]. A

RANSAC-based scheme is employed to choose from the hypotheses and deal with the outliers. Given an object, object pose and a set $\mathcal{PC}_c$ of points classified to belong to this object class, the pose quality is measured as a number of points from $\mathcal{PC}_c$ which lie within a certain threshold from the object's surface. The global scale of the reconstructed sparse scene is also estimated in this step.

However, even a minor scale estimation error may result in a considerable camera translation error in the Z-direction once the poses are reformulated using SE(3) transformations only. Therefore, further pose refinement is necessary. First, a variant of scale-aware point-to-point ICP is used on the reconstructed point cloud, followed by a multi-view edge-based refinement on RGB images inspired by the work of Drummond *et al.* [15]. The camera poses $\mathbf{\Xi}_i$ obtained from COLMAP remain fixed and only the similarity transformation from the model to the scene point cloud coordinate system is optimized. Again, no depth information is used. Multi-view consistency enforces the alignment of the object contours in every image, leading to a better solution, as opposed to the refinement in each RGB image separately, which may suffer from projective ambiguities.

3.5 Training and Implementation Details

We trained our detector on synthetic data generated from non-textured CAD models of the T-LESS dataset [21]. In our experiments, we focused on per-scene training. The detector is trained simultaneously for all the objects present in a particular test scene, as opposed to training one network per object. Training a separate network per each object is commonly done in other object detection and pose estimation pipelines because it tends to improve the results. However, this approach is not scalable and hard to apply in practice. For the training herein, a separate synthetic dataset for each of the scenes was generated. Each dataset contained 15K training samples. The data were generated completely automatically at random without any artistic modeling effort. With this setup, it took approximately 12 hours (with 16 Intel Core i9-9900K) to generate a train set for a particular scene. The points on the edges of the synthetic data samples are randomly sampled online during training. Sampling was done uniformly along the edge. Moreover, a number of samples was determined by $\frac{\|\mathbf{e}\|_2}{\nu}$, where $\|\mathbf{e}\|_2$ is the length of an edge and $\nu \sim U[3; 10]$. The purpose of the sampling was twofold. It prevented the network from overfitting to exact point locations. Additionally, it enforced the detector to learn how to recognize more global geometric structures. The resulting point cloud was either subsampled or padded to contain 16384 points.

Various augmentations were applied to the input point clouds, including random rotations around the Z-axis and around X and Y axes between $[-20°, 20°]$, scene flipping around the Z-axis. Random scaling by a factor between $[0.5, 2.0]$ is used to make the network more robust to scale changes. In addition, random dropout of points and Gaussian noise are used.

The network has been implemented using PyTorch deep learning framework [39]. Training was done using the Adam optimizer [28] with a learning rate of 0.01, a learning rate decay of 0.5, a and decay step of $2 \cdot 10^5$. On average,

convergence was observed within 200 training epochs. The weighting coefficient λ from the loss function in Equation 1 was set to 5.0 in all the experiments.

4 Experiments

4.1 Dataset and Evaluation Metrics

We evaluate our method on the challenging T-LESS [21] dataset consisting of 30 textureless industrial objects. Two types of 3D models are supplied with the dataset: untextured CAD models and low-quality textured reconstructions. T-LESS features 20 test sequences of various complexities, each of which contains 501 images captured at different camera elevations. We chose the T-LESS dataset because it is still challenging not only for pure RGB detectors but also for RGBD detectors. Several factors contribute to the complexity of the dataset. First, all objects are textureless in a sense that they do not have distinctive colors. All of them are colored in more or less the same shade of gray, except for certain structural parts. As a result, the performance of RGB detectors is automatically hindered because they cannot rely on color information. Second, the T-LESS objects exhibit symmetries leading to pose ambiguity. Mutual and self-occlusions of the objects are extensively present in the test sequences. The presence of similar-looking objects also makes it difficult to distinguish between them in the monocular case. For RGB-based methods, detection becomes even more complicated when only the untextured CAD models with no additional real images are used as an input for training. We aim to demonstrate that by using sequences of RGB images instead of separate frames it is possible to leverage the performance of detection and pose estimation, even when only synthetic CAD models are used as an input for training. Thus, the TLESS dataset is an ideal test set for our approach, since significant performance boosts can be expected for the approaches utilizing multi-frame consistency to overcome the those challenges.

While other popular datasets for pose estimation can be used, such as YCB-Video [60], LINEMOD[19] and OCCLUSION [7], they are not fully suitable for our approach. Evaluation on them is hindered by the fact that they feature RGB images captured with low resolution of 640×480 pixels, which does not allow for reliable 3D reconstruction and approximation with lines. In addition, scenes in LINEMOD and OCCLUSION change too fast. Thus, the existing subsequences of images with a constant scene consist of too few frames with little camera motion, which significantly reduces the amount of extra information that the multi-view setup brings. This is the reason why we have not evaluated our approach on these datasets and fully focused on the T-LESS dataset, which also tends to be more complicated than the aforementioned datasets.

For the evaluation, we followed the BOP challenge [23] protocol for Varying number of Instances of a Varying number of Objects (VIVO) task. The BOP pipeline was chosen because it strives to unify the dataset formats and evaluation methodologies to facilitate a comparison of various methods. We computed the overall performance score as the mean of VSD, MSSD and MSPD recalls, each of which is averaged across several thresholds. Visible Surface Discrepancy (VSD)

Table 1: Per-scene quality on BOP Challenge images. First three methods rely solely on RGB images. Other methods use real depth data for pose refinement.

Modality	Method/Scene	1	2	3	4	5	7	8	10	11	15	Average
RGB	DPOD	-	24.22	27.12	31.63	11.53	17.2	7.09	18.78	4.77	20.83	-
	AAE w\o ICP	49.32	64.08	45.2	42.65	38.56	43.81	31.74	26.88	50.57	21.81	41.46
	Ours	72.45	81.45	77.62	75.08	64.23	33.08	53.18	49.46	64.27	49	61.98
Depth	AAE w\ ICP	73.78	95.82	77.66	70.29	68.92	58.76	57.83	48.39	88.1	36.99	67.65
	PPF	64.33	68.72	60.88	52	45.05	53.59	56.12	57.44	67.34	42.7	56.82
	PPF with ref.	69.38	73.72	74.37	71.76	73.55	56.52	49.73	62.58	85.25	47	66.39

Table 2: Per scene quality for our method on selected T-LESS sequences

Modality	Method/Scene	1	2	3	4	5	7	8	10	11	15	Average
RGB	**Ours**	71.44	80.77	78.09	74.71	65.15	33.75	55.9	50.25	66.89	48.5	62.60

[22, 24] was used as a de-facto standard measure for comparing the results on the T-LESS dataset. VSD compares distance maps of the object rendered in the estimated and ground truth poses. Maximum Symmetry-Aware Surface Distance (MSSD) was defined in [13] as the maximal distance between corresponding model vertices in the ground truth similarly to the symmetric ADD [19]. Maximum Symmetry-Aware Projection Distance (MSPD) extends the idea of the projection error of [8] by taking explicit care of the object symmetries.

4.2 Results

We selected a minimal subset of all the scenes such that they cover all objects in the dataset. In the evaluation, subsequences of 72 consecutive frames were used to obtain the sparse reconstructions. Figure 5 demonstrates how the average recall changes depending on the number of frames used for the scene reconstruction. Two type of sequences were used: 1) consecutive frames, 2) frames with interleaving, when only every t-th frame was taken in order to cover as many camera poses around the object as possible. The plot confirms our assumption that a larger number of frames facilitates object detection and pose estimation. Taking frames with interleaving is also more robust when a fewer number of frames is used. The number of keypoints per model K was set to 8, and the inlier threshold θ introduced in Equation 4 was set to 0.995. We experimented with using from 5 to 20 keypoints, but there was no significant change in performance with the increased number of points.

The evaluation was conducted in two directions: 1) to compare with the other methods utilizing synthetic training data; and 2) to report the recall on all the images in the dataset. In all the experiments only 360 first images of each scene were used. The reason for that is the fact that a significant portions of static background become visible on the lower camera elevations, and motion of the turntable becomes explicit. This violates the static scene requirement which is essential for the SfM, making scene reconstruction impossible.

We ran the evaluation procedure on the BOP subset of the T-LESS data to perform a comparison with the other top-performing methods. Table 1 provides average recall for three different methods: AAE (with and without ICP) [53], DPOD [64], PPF (uses real depth and ICP) [14] and PPF with refinement [12]. The results show that our method clearly outperforms all the other approaches if no depth-based ICP is applied. Even result of AAE, which uses real data for training the detector, fall far behind. The only exception is scene 7, which contains numerous small round-shaped objects exhibiting strong inter-object similarity and no strong geometric features. The sparsity of the reconstructions does not allow for reliable representation of their geometry. This resulted in misdetections and inaccurately predicted poses. Moreover, the network also confuses similarly looking objects. Compared to AAE with ICP and the PPF variants, our approach still shows competitive results even though we do not use any real depth for refinement. With a score of 61.98 it outperforms the classical PPF, which achieves 56.82, by a large margin. In comparison with the PPF and AAE refined with ICP, our detector performs only slightly inferior. Table 2 presents the average per-scene recall of our detector on all the first 360 of images of each scene from the T-LESS dataset. A visual inspection of the predicted poses showed that the pose recall is mostly bounded by the object detection performance. If an object is detected, then its estimated pose is almost always nearly perfect.

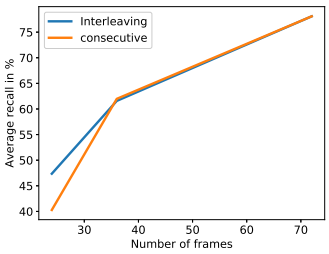


Fig. 5: Ablation study of the necessary number of sequential frames

5 Conclusions

In this work, we have introduced a pipeline for object detection from a sequence of RGB-only images. The presented approach is trained on synthetic simulations and, therefore, does not require any labeled real data. A sequence of images is processed with an SfM algorithm to obtain a sparse reconstruction for the scene to allow for a simultaneous detection for all of the images in one shot. We relied heavily on the intuition that the geometric edges are the only prominent features that can both be extracted from the 3D models as well as from the sequences of real RGB images. The resulting detector was found to work significantly better than DPOD, which was also trained completely on synthetic data, and AAE, which used real data for training the 2D detector and synthetic data for the pose estimation network. Additionally, our detector works better or on par with AAE and the PPF-based approaches even if their poses are refined with depth ICP.

Acknowledgements. We thank the authors of AAE [52, 53], DPOD [64] and PPF [14, 12] for providing us with the poses estimated with their detectors.

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
2. Avetisyan, A., Dai, A., Nießner, M.: End-to-end cad model retrieval and 9dof alignment in 3d scans. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2551–2560 (2019)
3. Birdal, T., Ilic, S.: Point pair features based object detection and pose estimation revisited. In: 2015 International Conference on 3D Vision. pp. 527–535. IEEE (2015)
4. Birdal, T., Ilic, S.: A point sampling algorithm for 3d matching of irregular geometries. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6871–6878. IEEE (2017)
5. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3722–3731 (2017)
6. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: Advances in neural information processing systems. pp. 343–351 (2016)
7. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: European conference on computer vision. pp. 536–551. Springer (2014)
8. Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., et al.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3364–3372 (2016)
9. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
10. Coumans, E.: Bullet physics simulation. In: ACM SIGGRAPH 2015 Courses. p. 7. ACM (2015)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
12. Drost, B., Ilic, S.: 3d object detection and localization using multimodal point pair features. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission. pp. 9–16. IEEE (2012)
13. Drost, B., Ulrich, M., Bergmann, P., Hartinger, P., Steger, C.: Introducing mvtec itodd-a dataset for 3d object recognition in industry. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2200–2208 (2017)
14. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 998–1005. Ieee (2010)
15. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 932–946 (Jul 2002). https://doi.org/10.1109/TPAMI.2002.1017620, http://dx.doi.org/10.1109/TPAMI.2002.1017620
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)

17. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
18. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
19. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. pp. 548–562. Springer (2012)
20. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K.: Going further with point pair features. In: European conference on computer vision. pp. 834–848. Springer (2016)
21. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. IEEE Winter Conference on Applications of Computer Vision (WACV) (2017)
22. Hodaň, T., Matas, J., Obdržálek, Š.: On evaluation of 6d object pose estimation. In: European Conference on Computer Vision. pp. 606–619. Springer (2016)
23. Hodan, T., Melenovsky, A.: Bop: Benchmark for 6d object pose estimation: https://bop.felk.cvut.cz/home/ (2019)
24. 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: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–34 (2018)
25. Hofer, M., Maurer, M., Bischof, H.: Efficient 3d scene abstraction using line segments. Computer Vision and Image Understanding **157**, 167–178 (2017)
26. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
27. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1521–1529 (2017)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8. IEEE (2018)
30. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 35–51 (2018)
31. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. International journal of computer vision **81**(2), 155 (2009)
32. Li, Y., Bu, R., Sun, M., Chen, B.: Pointcnn. arXiv preprint arXiv:1801.07791 (2018)
33. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
34. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)

35. Liu, X., Qi, C.R., Guibas, L.J.: Flownet3d: Learning scene flow in 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 529–537 (2019)
36. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791 (2015)
37. Manhardt, F., Arroyo, D.M., Rupprecht, C., Busam, B., Birdal, T., Navab, N., Tombari, F.: Explaining the ambiguity of object detection and 6d pose from visual data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6841–6850 (2019)
38. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
40. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4561–4570 (2019)
41. Pitteri, G., Ilic, S., Lepetit, V.: Cornet: Generic 3d corners for 6d pose estimation of new objects without retraining. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
42. Pitteri, G., Ramamonjisoa, M., Ilic, S., Lepetit, V.: On object symmetries and 6d pose estimation from images. In: 2019 International Conference on 3D Vision (3DV). pp. 614–622. IEEE (2019)
43. Planche, B., Zakharov, S., Wu, Z., Hutter, A., Kosch, H., Ilic, S.: Seeing beyond appearance-mapping real images into geometrical domains for unsupervised cad-based recognition. arXiv preprint arXiv:1810.04158 (2018)
44. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. arXiv preprint arXiv:1904.09664 (2019)
45. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 918–927 (2018)
46. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
47. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
48. 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: Proceedings of the IEEE International Conference on Computer Vision. pp. 3828–3836 (2017)
49. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
50. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
51. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: Pro-

ceedings of the IEEE conference on computer vision and pattern recognition. pp. 2107–2116 (2017)
52. Sundermeyer, M., Marton, Z.C., Durner, M., Brucker, M., Triebel, R.: Implicit 3d orientation learning for 6d object detection from rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 699–715 (2018)
53. Sundermeyer, M., Marton, Z.C., Durner, M., Triebel, R.: Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. International Journal of Computer Vision pp. 1–16 (2019)
54. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200 (2016)
55. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 292–301 (2018)
56. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
57. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis & Machine Intelligence (4), 376–380 (1991)
58. Vidal, J., Lin, C.Y., Lladó, X., Martí, R.: A method for 6d pose estimation of free-form rigid objects using point pair features on range data. Sensors **18**(8), 2678 (2018)
59. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3343–3352 (2019)
60. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199 (2017)
61. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018)
62. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 532–541 (2019)
63. 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. 3DV (2018)
64. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1941–1950 (2019)
65. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)