PrimitivePose: 3D Bounding Box Prediction of Unseen Objects via Synthetic Geometric Primitives

Andreas Kriegler
Vision, Automation and Control
AIT Austrian Institute of Technology
Vienna, Austria
0000-0002-5653-5181

Csaba Beleznai
Vision, Automation and Control
AIT Austrian Institute of Technology
Vienna, Austria
Csaba.Beleznai@ait.ac.at

Markus Murschitz
Vision, Automation and Control
AIT Austrian Institute of Technology
Vienna, Austria
Markus.Murschitz@ait.ac.at

Kai Göbel
Vision, Automation and Control
AIT Austrian Institute of Technology
Vienna, Austria
Kai.Goebel.fl@ait.ac.at

Margrit Gelautz
Visual Computing and Human-Centered Technology
TU Wien
Vienna, Austria
margrit.gelautz@tuwien.ac.at

***Abstract*—This paper studies the challenging problem of 3D pose and size estimation for multi-object scene configurations from stereo views. Most existing methods rely on CAD models and are therefore limited to a predefined set of known object categories. This closed-set constraint limits the range of applications for robots interacting in dynamic environments where previously unseen objects may appear. To address this problem we propose an oriented 3D bounding box detection method that does not require 3D models or semantic information of the objects and is learned entirely from the category-specific domain, relying on purely geometric cues. These geometric cues are objectness and compactness, as represented in the synthetic domain by generating a diverse set of stereo image pairs featuring pose annotated geometric primitives. We then use stereo matching and derive three representations for 3D image content: disparity maps, surface normal images and a novel representation of disparity-scaled surface normal images. The proposed model, PrimitivePose, is trained as a single-stage multi-task neural network using any one of those representations as input and 3D oriented bounding boxes, object centroids and object sizes as output. We evaluate PrimitivePose for 3D bounding box prediction on difficult unseen objects in a tabletop environment and compare it to the popular PoseCNN model – a video showcasing our results can be found at: https://preview.tinyurl.com/2pccumvt.**

***Index Terms*—3D bounding box prediction, unseen objects, synthetic data, geometric primitives, object pose annotation**

I. Introduction

For robots to operate successfully in diverse real-world environments, they must be able to perceive previously unseen objects. To acquire the information necessary for interaction with these objects, approaches from 3D object recognition and object pose estimation are used. Nevertheless, many of the existing works require 3D models or other annotated data and as such are limited to a closed-set of object categories [1]–[6], i.e. the models are only capable of recognizing objects from a small, specific set of classes and fail at detecting novel objects from previously unseen classes. Some recent works have tried to lift the closed-set constraint for 2D [7] or 3D object detection [8] as well as object segmentation [9], detecting everything that looks like an object similar to the perceptual grouping principles [10]. *Objectness*, or a measure for the probability that an object exists in a given region of interest, is an important concept in this regard. 2D objectness often relies on learned representations of spatial groupings (bounding box or segmentation estimates) [11]. 3D objectness requires more geometry-oriented reasoning, where dimension, orientation and distance from the observer are key parameters to estimate. Therefore, for 3D objectness, typically more complex data representations are required, e.g. RGB-D or LiDAR point clouds [12]–[14], segmentation masks [15] or entire CAD models [1], [16]–[18]. In this work, we want to learn 3D objectness and estimate oriented 3D bounding boxes of novel objects without category constraints or CAD models.

For learning-based models to perform well, large-scale data sets are required, which are very labour-intensive and challenging to obtain, especially for object centric tasks. Using synthetic data to this end is becoming more popular [6], [22]–[24], but it has its own set of challenges. Most notably the data quality gap emerges when compared to real images, since image formation via real RGB cameras is governed by complex physical phenomena and it is difficult to recreate the variation and noise in RGB space of real cameras [25], requiring advanced synthesis/rendering solutions. Depth maps from stereo matching also hold the cues required for learning 3D objectness and can be fully synthesized, but a similar data quality gap to real depth becomes apparent. On the other hand, we found in previous work [26] that simulating stereo RGB images but then performing stereo matching for disparity estimation on the synthetic images delivers semi-synthetic depth maps similar to those estimated from real stereo images, reducing the sim-to-real gap. We take a similar approach in this work, leveraging a rendering engine to generate vast amounts of synthetic stereo image pairs, for a subsequent disparity computation.

The type of representation used for the 3D information

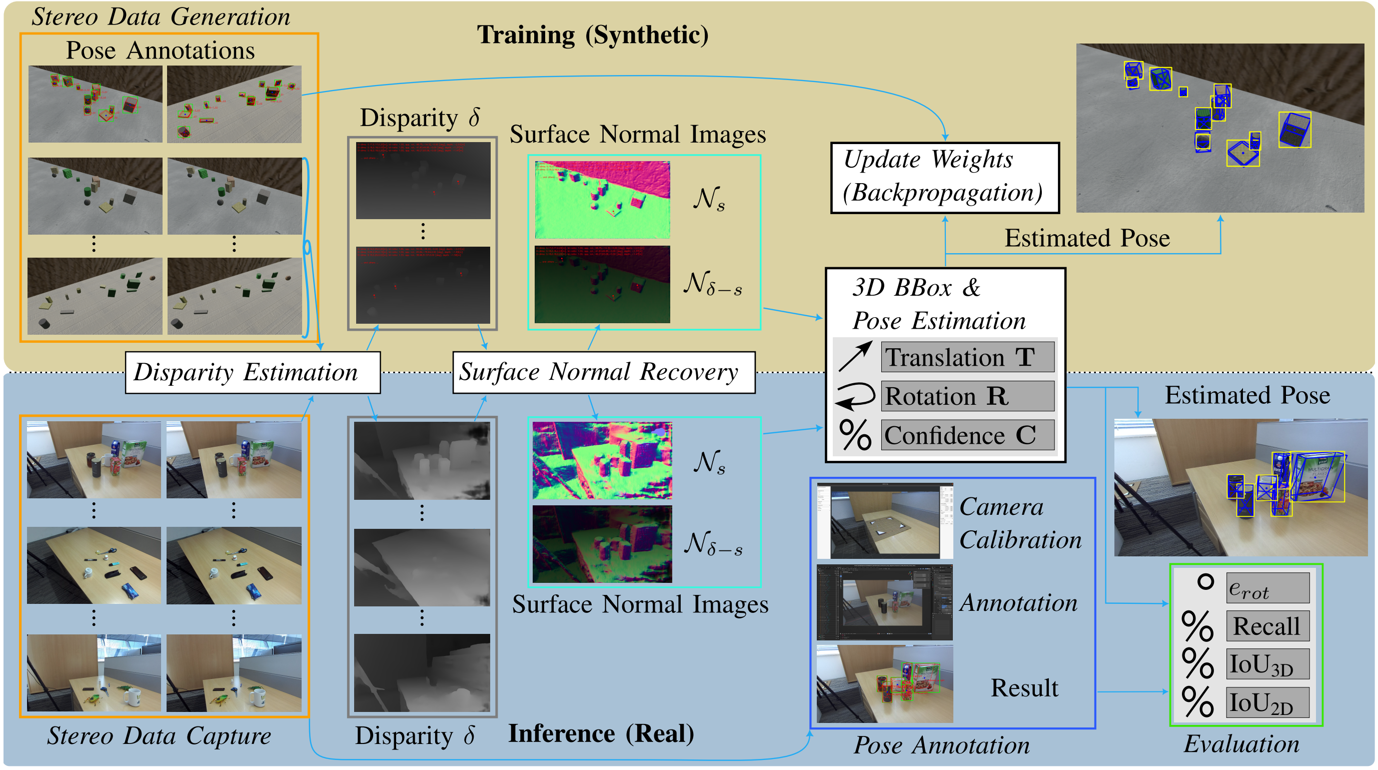

Fig. 1: The overall pipeline of our approach during training (top) and inference (bottom). **Training:** The Blender rendering engine [19] is used to create a vast data set of synthetic stereo images featuring 3D geometric primitives in various configurations and view-point variations. Ground-truth location and pose annotations of all objects are exported. A stereo-matcher [20] is used to obtain disparity maps. We modify a keypoint based 3D detection model [21] and use the disparity maps δ, recovered SN images $\mathcal{N}_s$ or disparity-scaled SNs $\mathcal{N}_{\delta-s}$ for training. **Inference:** A data set of tabletop stereo-images featuring unseen and difficult objects was captured for evaluation. We use a novel annotation tool to obtain ground-truth 6DoF pose, location and size for all objects and evaluate the trained model for 3D bounding box prediction. Best viewed in colour and zoomed in.

is an important component to facilitate the learning process. The surface normal (SN) space can be used to this end [27], where SN images are a 2D 3-channel representation of surface curvature in 3D space. They are highly scale/distance invariant, making the learning task easier, but unfortunately any depth or metric scale information is lost when transitioning from disparity to surface normals. Although surface normals alone enable pose estimation of objects, learning the object dimensions is hindered by this loss of global depth information and sense of scale. Therefore, as an extension to conventional SN computation, we propose to combine the SN images with disparities, scaling each normal vector by the disparity value at that pixel which leads to a map well representing shapes (surface curvature) and also conveys information on the metric scale (distance).

This work presents a stereo-based method for estimation of oriented 3D bounding boxes for unknown objects that does not require CAD models. Our method, which is based on the CenterNet [21] architecture and trained without any real images, can estimate the pose of entirely unknown real objects on generic horizontal surfaces in an end-to-end manner (see Figure 1). We use PrimitivePose for 3D bounding box prediction on two data sets: exclusively tabletops and more general environments. For the tabletop images, we collected stereo views of difficult objects from multiple cameras and viewpoints under varying lighting conditions and manually annotated 6DoF object poses. For eight other environments, we employ PrimitivePose on the STIOS data set [28]. In both cases we compare against the popular PoseCNN method [5]. Although our model has never seen any of the objects in either experiments, it achieves satisfying results. In summary, we make the following three contributions:

- Propose a model for 3D object detection of unseen objects trained without any real images and evaluate our method against state-of-the-art on a novel data set of tabletop images – an evaluation video can be found at: https://preview.tinyurl.com/2pccumvt.
- Introduce the notion of disparity-scaled surface normal images and investigate the efficacy of three different intermediate data representations for 3D object recognition.
- Make our toolkit for 6DoF pose annotation of objects publicly available at: https://preview.tinyurl.com/3ycn8v5k.

II. Related Work

We split related works into three parts. First, we present works which use synthetic data for object-centric learning tasks. We then discuss different approaches to represent object geometries and surfaces via surface normals. We conclude by comparing with existing methods for class-agnostic object detection and pose estimation.

A. Synthetic Data for Object Recognition

Object detection and segmentation in the robotic context can benefit from synthetic data generation. A synthetic camera and randomly dropped object models were used in [22] for fast generation of synthetic depth training data for 3D object segmentation. Similarly, in [24], a vast synthetic data set was successfully used for unseen object segmentation in tabletop environments. Object meshes and a simulated stereo sensor were used in [23] for generating synthetic stereo images to learn parallel-jaw grasping models. [6] relies fully on synthetic data for training a deep pose estimation network and shows that the synth-to-real gap can be bridged, but they only evaluate their method on known objects. The authors of [29] used BlenderProc4BOP, a derivate of the Blender [19] rendering engine, for generating training images for the BOP 2020 challenge. In this work, we want to combine synthetic input generated using Blender with stereo matching - a process that yields depth data comparable to real-image based stereo depth [26]. SimNet [30] similarly exploits synthetic training data and stereo information but only uses a low-resolution disparity image as representation while we use full-resolution disparity as well as SN and disparity-scaled SN images.

B. Object Representation and Surface Normals

The approximation of object structure via 3D geometric primitives is a popular idea in computer graphics literature. Fitting primitives to CAD objects in LiDAR point clouds is a common task with its own benchmark [31]. In [32], man-made objects are modelled with geometric primitives at different abstraction levels. In [33], the authors use a superquadric object parameterization for pose estimation of primitive-shaped objects. Poses of strictly cylindrical objects were estimated in [26]. For approximation of object geometries, an adequate representation of structure is important. To this end, a SN description is one possibility, used already in classical approaches [34]. In end-to-end SN estimation methods [35], [36], the normals themselves are seen as the final model output. Other works use surface normals as intermediate representation for object pose estimation [27] [37], hand pose estimation [38], pose estimation of transparent objects via RGB-D data [39] or 3D model retrieval from a CAD model library [40]. Yet none of these methods are class-agnostic or attempt to include an open set of models.

C. Class-Agnostic Object Pose Estimation

A number of class-agnostic pose estimation methods have recently been proposed to circumvent the problem of limited object classes [15]–[17], [41]–[44]. [42] requires multiple views of the same object on a turntable, [45] also needs a number of views of the object in various orientations and [46] requires a RGB video scan for object pose estimation. In [15], a detection and tracking framework of novel objects is proposed requiring an expensive segmentation step limiting real-time capabilities. [43] and [16] predict generic 3D corner points of the 3D bounding box for class-agnostic pose estimation but [43] only reports performance on seen classes and [16] requires CAD models of new objects at test time. Similarly, [17], [41], [44] all use 3D models to adapt to objects unseen during training. The authors of CenterSnap [47] propose a single-shot pipeline for 3D reconstruction and 6D pose estimation, treating object instances as spatial centers inspired by CenterNet [21]. To learn shape-codes for each object, [47] uses an auto-encoder trained on 3D shapes from a set of CAD models. In contrast to above works, we do not need 3D shape information during training or testing. We introduce an "objectness" prior with a diverse set of 3D geometric primitives and exploit synthetic data to generate a vast data set of stereo image pairs of geometric primitives with annotation labels. Our method is not limited to any set of classes, requires a stereo depth image but no LiDAR point clouds, object meshes, CAD models or multiple viewpoints. A similar sim-to-real transfer approach was taken by the authors of [30], although their simulated data was generated showing specific domains and objects while we take the most basic approach of generating primitives on a simple ground plane.

III. Methods

We first give a description of the synthetic data generation pipeline, including an explanation of the pose annotation tool. Then, we formally describe the calculation of surface normals given a disparity map and our proposed change of scaling these vectors with disparity. Finally, we describe the learning scheme used for 3D bounding box estimation.

A. Generating Synthetic Data for Object Recognition

We use the Blender [19] software at two stages of our pipeline: for generating synthetic stereo image pairs and for pose annotation of objects in real images.

Synthetic data generation: For generating synthetic training images, we arbitrarily spawn a collection of 3D compact mesh objects including cuboids, cylinders and spheres. The number of objects, their size, location and rotation are randomized uniformly for every configuration. A virtual stereo camera rig, parameterized as a real camera (ZED 2[1]), looks at the scene. The camera's distance to the scene, view target and elevation angle are randomized. We use the EEVEE engine from Blender to render pairs of stereo images, while extracting ground truth object parameters, such as pose and size as well as object occlusion calculated via ray-tracing. The whole process takes around 0.5 seconds per image pair, depending on renderer settings and scene complexity. The entire training set includes around 350k raw disparity maps, surface normal

[1] https://www.stereolabs.com/zed-2/

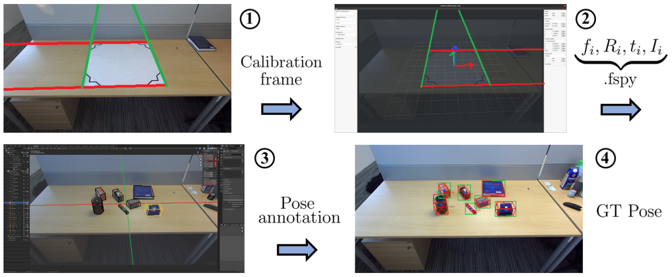

Fig. 2: 6DoF pose annotation for arbitrary objects with known size is possible for uncalibrated monocular cameras using our toolkit. With a calibration frame ① a virtual camera model is obtained ② and computer graphics software used for annotation ③ and export ④ of object poses. Best viewed zoomed in.

images and disparity-scaled surface normal images, all derived from the generated stereo image pairs.

Object pose annotation: The pose annotation tool works for uncalibrated monocular cameras assuming the size of objects is known and the process takes place in four steps (see Figure 2). First a calibration frame for a given camera pose is captured showing simply a square of known size or two other sets of parallel, mutually orthogonal, lines with a known offset ①. A virtual camera model is obtained using the algorithm of [48] – if the intrinsic parameters of the camera are known, only the extrinsics need to be estimated ②. Then, using Blender [19], the camera model is loaded and all objects present in the real scene are recreated using 3D primitives. 3D meshes of objects are aligned with their 2D counterparts from the real background images for annotation of object poses ③. The relative camera-object orientation as well as 3D location of objects, object size and occlusion metrics can then be exported and visualized ④. For the calculation of object occlusion, a ray-tracing algorithm sends rays from the camera focal point through every pixel of the virtual image plane and stores all intersections of the rays with object meshes. Annotation takes around 3-5 minutes per image, depending on the annotator's familiarity with the software – the toolkit is therefore best suited for annotation of test data sets with multiple hundred images. The tool also allows to have several cameras observing the same scene from different viewpoints – this helps create more accurate pose annotations, since 2D-3D alignments can be validated from multiple views.

B. From Disparity to Surface Normals

We use an off-the-shelf learning-based stereo matching model, AANet [39], to obtain disparity estimates from synthetic, generated image pairs. Although recent monocular approaches [49], [50] have delivered good results, stereo vision is still preferred for accurate depth estimation [51], since monocular depth estimation suffers from global consistency and can introduce geometric distortions [26]. We transform the disparity maps into a surface-normal representation as follows. Let δ_i be the i-th disparity image and d_i the resulting depth image:

$$d_i = \frac{f \cdot b}{\delta_i + \epsilon} \tag{1}$$

where f is the horizontal focal length in pixels, b is the baseline in millimeters and ϵ a very small constant. We compute the two-dimensional gradient images $\nabla_x, \nabla_y \in \mathbb{R}^{w \times h \times 1}$ in x and y direction via convolution of the image with Sobel kernels of size $k = 3$, using the OpenCV implementation[2]. $w = 896$ and $h = 512$ are the width and height of the image:

$$\nabla_x = Sobel(d_i, 1, 0, k) \qquad \nabla_y = Sobel(d_i, 0, 1, k). \tag{2}$$

The vector field of all surface normals $\mathcal{N}_i \in \mathbb{R}^{w \times h \times 3}$ to all supporting gradient planes can then be constructed and normalized to unit length via the Frobenius matrix norm:

$$\mathcal{N}_i = \frac{[\nabla_x, \nabla_y, \nabla_z]}{||[\nabla_x, \nabla_y, \nabla_z]||_F} \tag{3}$$

Here the three two-dimensional gradient images are stacked channel-wise. For estimates of the gradient map ∇_z see [36] – we have found in practice that making the simplifying assumption $\nabla_z = 1$ works similarly well, i.e. the model trained on the surface normals shows comparable convergence behaviour. For every pixel of $\mathcal{N}_i$ the three channels express the surface normal direction at that pixel. The normals are then scaled from $[-1, 1]$ to $[0, 1]$:

$$\mathcal{N}_i = 0.5 \cdot (\mathcal{N}_i + 1). \tag{4}$$

A standard surface normal image $\mathcal{N}_{s,i}$ is then obtained by scaling each of the three image channels, each encoding one dimension of the 3D surface normal vector, to $[0, 255]$.

$$\mathcal{N}_{s,i} = \mathcal{N}_{s,i} \cdot 255 \tag{5}$$

For the disparity-scaled surface normals $\mathcal{N}_{\delta-s,i}$, first a scaling factor s is obtained for $[0, 255]$ range via

$$s = \frac{255.0}{max\_disp} \tag{6}$$

where the maximum disparity value max_disp is 192 for AANet. All three channels of the original disparity maps are then scaled with this factor

$$\delta_i = [\delta_{i,1} \cdot s, \delta_{i,2} \cdot s, \delta_{i,3} \cdot s] \tag{7}$$

and stacked for a three-channel disparity vector field. To obtain the final disparity-scaled surface normals $\mathcal{N}_{\delta-s,i}$, the original unit normals $\mathcal{N}_i$ are scaled with the vector field δ_i:

$$\mathcal{N}_{\delta-s,i} = \mathcal{N}_i \cdot \delta_i. \tag{8}$$

This representation encodes the orientation of the 3D surface normal vector at every pixel (proportion of RGB values), while the length of the vector (magnitude of RGB values) encodes the disparity information. In practice, this results in brightness changes of the (colour-encoded) surface normal image which are inverse proportional to the depth at each pixel (see middle part of Figure 1).

[2] https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html

TABLE I: Pose prediction recall and 3D IoU when training our model on raw disparity δ, surface normals $\mathcal{N}_s$ or disparity-scaled surface normals $\mathcal{N}_{\delta-s}$ and applying the model on our tabletop testset. The first row additionally gives PoseCNN results. We split our test set equally according to the object-sizes (volume of the 3D bounding box) into images showing only small objects, only large objects or a mixture of the two.

Method	Large	Mixed	Small	Large	Mixed	Small	Large	Mixed	Small	Large	Mixed	Small	Large	Mixed	Small	Large	Mixed	Small
	Recall ($e_{rot} < 2°$)			Recall ($e_{rot} < 5°$)			Recall ($e_{rot} < 10°$)			Recall ($e_{rot} < 15°$)			Recall ($e_{rot} < 25°$)			Recall ($e_{rot} < 40°$)		
PoseCNN [5]	0.0	0.0	0.0	0.0	1.2	0.0	2.5	1.6	0.8	3.7	2.3	1.1	4.4	3.7	4.1	7.4	9.3	10.6
δ	**27.5**	**22.5**	**47.7**	**38.6**	32.8	**55.6**	**45.5**	39.2	**60.3**	49.9	43.6	**62.7**	53.8	47.1	**65.0**	57.1	49.9	**67.4**
$\mathcal{N}_s$	19.1	14.6	29.1	33.6	26.4	40.0	42.3	35.7	45.0	47.5	40.8	48.0	51.7	44.9	51.2	55.5	48.2	54.0
$\mathcal{N}_{\delta-s}$	22.0	19.3	29.7	37.2	**34.5**	43.4	45.3	**42.1**	50.4	**50.7**	**46.9**	54.6	**55.8**	**50.8**	57.6	**59.8**	**54.2**	60.5
	IoU$_{3D}$ ($d_{tol} = 0\%$)			IoU$_{3D}$ ($d_{tol} = 4\%$)			IoU$_{3D}$ ($d_{tol} = 8\%$)			IoU$_{3D}$ ($d_{tol} = 12\%$)			IoU$_{3D}$ ($d_{tol} = 16\%$)			IoU$_{3D}$ ($d_{tol} = 20\%$)		
δ	**15.3**	**12.2**	**5.0**	**20.0**	**16.7**	**8.6**	**24.2**	**20.5**	**12.4**	**27.3**	**23.0**	**15.5**	**29.5**	**24.7**	**17.6**	**31.0**	**25.9**	**19.0**
$\mathcal{N}_s$	6.8	1.8	0.2	9.1	3.0	0.6	11.7	4.7	1.5	14.4	6.6	3.0	16.8	8.5	4.7	18.9	10.2	6.1
$\mathcal{N}_{\delta-s}$	12.4	7.0	2.4	15.9	10.2	4.0	19.2	13.4	6.4	21.9	16.3	8.7	24.0	18.6	10.6	25.7	20.3	12.3

C. 3D Object Pose Parameterization and Estimation

Lastly we require an adequate and learnable representation of object pose and size, with respect to the camera position and orientation. In this work we use simple Euler angles for representing the pose of objects relative to the camera pose. Using a center-point detection architecture we optimize multiple learning tasks to directly regress bounding box parameters around any objects in an image. Our model, a modified version of CenterNet [21], learns to estimate 3D oriented bounding boxes using 17 parameters in total grouped according to their loss functions: $\mathcal{L}_{hm}$ - 3 parameters for the heatmap-based 2D object center, $\mathcal{L}_{ang}$ - 6 parameters for the relative object-camera rotations ($sin(\cdot)$ and $cos(\cdot)$ of all three angles), $\mathcal{L}_{dim}$ - 3 parameters for object dimensions, $\mathcal{L}_{dep}$ - 1 parameter for the depth, i.e. the distance between object and camera center along the z-axis, and $\mathcal{L}_{box}$ - 4 parameters for the 2D bounding box to enable 2D IoU correspondence matching, although this is not a necessity for 3D object detection itself. We modify CenterNet [21] by adding additional heads to the network architecture, enabling us to learn 3D pose, location and size parameters in addition to the standard 2D objectives. Furthermore, we use standard L1 loss, weighting each term equally. To alleviate problems of ambiguity from object symmetries, we define canonical object poses setting object orientation axes aligned with the tabletop plane while also clamping certain rotation angles where necessary. We train all models for 23 epochs, starting with a learning rate of 0.5e-3 and decreasing it by a factor of 10 with epoch 20. We observe that the models in general converge quickly, regardless of the intermediate data representation: after 5 epochs predictions from the model are already fairly accurate. We use a detection threshold of 0.2 in all experiments.

IV. Experiments

In this section, we aim to answer the following questions with experiments: 1) How well does PrimitivePose estimate oriented 3D bounding boxes around objects on tabletops from a stereo observation? 2) Which intermediate data representation (disparity, surface normals, disparity-scaled surface normals) is best suited for this task? 3) Does PrimitivePose transfer to more complex environments and objects?

To answer the first two questions we report results on our captured data set of tabletop images – for the last question we use the STIOS [28] data set.

A. Oriented 3D Bounding Box Detection on Tabletops

Dataset: The tabletop test set consists of roughly 200 images from multiple camera viewpoints showing a generic wooden table with various objects scattered on it. These objects include compact cuboid-like objects such as books, objects with an axis of rotational symmetry such as markers and cans, objects that are cylindrical but have an additional identifying feature such as handles on cups, as well as challenging non-compact objects like a pair of scissors and plastic toy bugs with outreaching appendages.

Metrics: Since we do not use any 3D models at any stage, we cannot adopt the popularly used ADD metric [52] that compares model points. Instead, an object pose, or oriented 3D bounding box, is considered correctly predicted if the rotational error $e_{rot} \in [0, 180]$ [53]

$$e_{rot} = \arccos\left((Trace(\hat{\mathbf{R}}\bar{\mathbf{R}}^{-1}) - 1)/2\right) \times \frac{180}{\pi} \qquad (9)$$

is below a certain threshold $\theta \in \{2, 5, 10, 15, 25, 40\}°$. Ground truth and predicted rotation matrices $\bar{\mathbf{R}}$ and $\hat{\mathbf{R}}$ are created using the general extrinsic rotation matrices from the three ground-truth and predicted Euler angles.

Correspondence matching: After obtaining bounding box predictions from inference, we still have to solve the correspondence problem: matching ground truth and predicted bounding boxes. It should be noted at this point that, due to imperfect camera calibration, the ground truth distance from camera to object center following manual annotation can be off by a few centimeters, which could have significant impact on calculated 3D IoU considering many of our objects are flat or small. Therefore, to report 3D detection results, we use 2D IoU or the Jaccard index [54] for a more stable object matching. To report 3D IoU, we use the 3D IoU itself to find matches, but allow a tolerance of $d_{tol} \in \{0, 4, 8, 12, 16, 20\}\%$ in depth. To do so, we uniformly sample 3D bounding box candidates along the vector connecting camera and object centers. Finally, we use the occlusion measure calculated automatically during

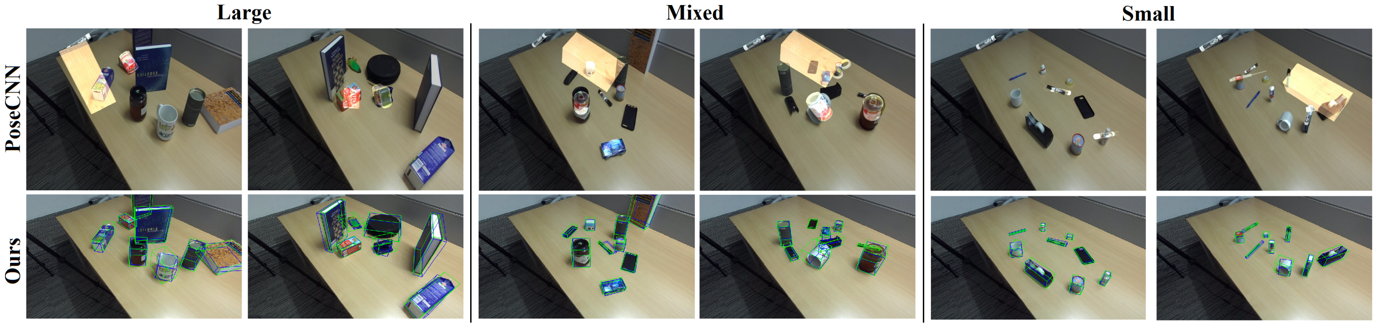

Fig. 3: Comparing predictions from PoseCNN [5] (top row) and our approach (bottom row) on a set of tabletop images, splitting objects into three bins according to their size (3D bounding box volume). Ground truth bounding boxes are green, predictions blue.

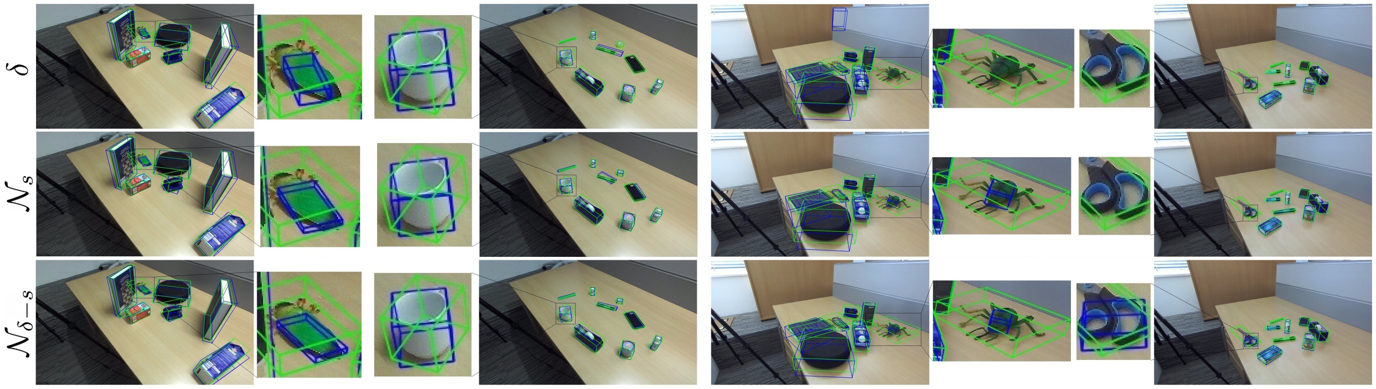

Fig. 4: Prediction results for all three intermediate representation types (top: disparity δ, middle: surface normals $\mathcal{N}_s$, bottom: disparity-scaled surface normals $\mathcal{N}_{\delta-s}$) – failure cases seen zoomed in. Errors happen most commonly for non-compact objects (toy-bugs, scissors) or due to rotational symmetry (mug). Ground truth bounding boxes are green, predictions blue.

annotation to filter all objects with occlusion greater than 90% for the evaluation.

PoseCNN comparison: We compare pose predictions from PrimitivePose against PoseCNN [5]. PoseCNN was originally trained on 21 YCB objects [55] which look, at least geometrically, similar to ours which justifies our comparison against the method. We apply PoseCNN on the left stereo image resized to $640x480$, using the default detection threshold of 0.2 and changing the intrinsic camera matrix for model reconstruction accordingly. PoseCNN predicts 3D pose in the form of oriented 3D YCB object models which enables a comparison in terms of pose prediction recall. Unfortunately, since the actual size of the YCB objects or the 3D bounding box encompassing them, is not publicly reported, we cannot make the transition from 3D object models to bounding boxes for evaluating 3D IoU. Table I lists obtained results.

Evaluation results: For pose prediction following the rotational error shown in the upper half of the table, both raw disparity maps δ and disparity-scaled surface normals $\mathcal{N}_{\delta-s}$ produce good results. Perhaps surprising to see is that predictions are consistently more accurate for smaller than for larger objects. One possible explanation is that the larger objects often have 2 sides of roughly equal length, leading to a possible 90° flipping ambiguity. PoseCNN gives poor results, most likely due to appearance changes of the objects compared to YCB, highlighting the importance of geometric cues. For 3D IoU seen in the lower half of the table, the loss of depth information with the standard surface normals $\mathcal{N}_s$ leads to expected inferior results. It is still interesting to see that raw disparity alone seems to be the best suited data representation, across object sizes and thresholds. A possible explanation for this might be that given enough data, the neural network model is capable of approximating the desired input-output transfer function anyway, without the need of explicitly transforming inputs into a surface normal representation. Regardless, the importance of preserving depth information and not only surface orientation is clear.

Visual results: Observing qualitative results in Figure 3, we note particular problems with the plastic toy bugs, where predicted bounding boxes are consistently too small, and the coffee/tea cups, where the rotation indicated by the handle is rarely inferred correctly - as shown in more detail in Figure 4. Neither of these failure cases are surprising, since non-compact objects or objects with cavities were not among the objects rendered for the training data in the former case and the model was trained to predict bounding boxes around

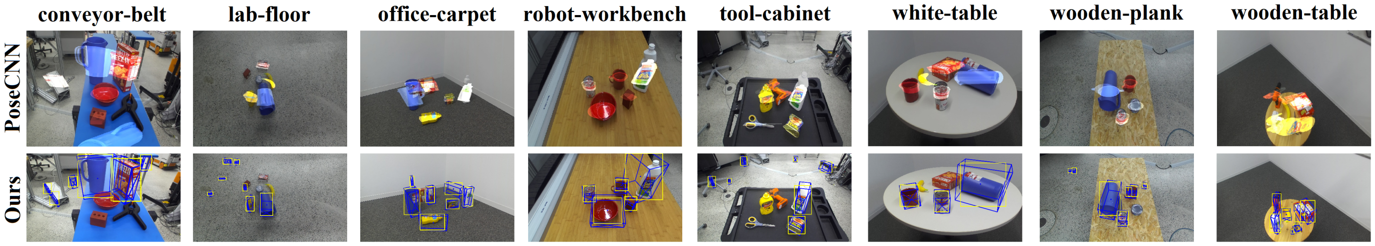

Fig. 5: 3D bounding box predictions on the STIOS [28] data set. Our model correctly detects many objects and gives reasonable orientation and size estimates even though it has never seen any of the objects. Blue boxes: 3D pose estimates, yellow rectangles: 2D object bounding boxes.

cylindrical objects to face the camera directly for the latter case. PoseCNN struggles to recognize any of the objects, most likely due to changes in appearance (colour) from the YCB objects.

B. Application on the STIOS data set

The STIOS [28] data set was chosen for evaluation because: 1) it provides stereo images; 2) all objects in it are YCB objects and were seen by PoseCNN; 3) many objects can be approximated with a 3D primitive enabling application of our model. STIOS contains recordings from a stereo ZEDCam in eight different environments. Roughly 25 images exist for each of the eight environments – images taken from different camera poses and/or showing mixed configurations of objects. Ground truth segmentation masks, 2D bounding boxes derived from the masks and object class labels are provided but no pose annoations, which is why we only provide qualitative prediction results. Many images in STIOS are also fairly close to the camera, which resulted in the stereo matching algorithm reaching maximum disparity (which cannot be increased by the user) and the disparity maps becoming corrupted. Even so, many objects were recognized and reasonable 3D bounding boxes predicted, as can be seen in Figure 5.

V. Conclusion

In this paper we presented an approach to predict generic oriented 3D object bounding boxes for previously unseen objects from stereo data. Our models were trained on representations derived from stereo depth computed exclusively from synthetic stereo image pairs and yield a good generalization on real images. We analyzed three intermediate representations - raw disparity maps, surface normal images and a novel representation via disparity-scaled surface normal images - and showed the importance of preserving depth information. We evaluated our models on a set of real images showing difficult tabletop scenes with arbitrary, unseen objects and compared against a state-of-the-art model which our models were able to outperform. We further applied PrimitivePose on a second data set and show promising 3D detection results even though our model has never seen any of the objects and only relies on generic 3D geometric cues. We believe our approach is useful towards an increasingly open-ended object recognition task in a robotic context. Alternatively it could also be used for object-centric event detection in video streams, in combinations with frameworks such as [56].

For future work, using object proposals as a part-vocabulary to model more complex geometries could further extend the set of recognizable objects. We also aim to improve the synthetic part of our pipeline, for more advanced data rendering including more primitive objects and better annotation of real images. In this way the suitability of our approach for more complex scenes including for example overlapping and rotated objects will be shown.

References

[1] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2637–2646.

[2] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D Region Proposal Network for Object Detection," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 9286–9295.

[3] H. Bradler, A. Kretz, and R. Mester, "Urban Traffic Surveillance (UTS): A fully probabilistic 3D tracking approach based on 2D detections," in *Intelligent Vehicles Symposium (IV)*, 2021, pp. 1198–1205.

[4] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A General Framework for Monocular 3D Object Detection," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 5170–5184, 2021.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *Robotics: Science and Systems XIV (RSS)*, 2018.

[6] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," in *Conference on Robot Learning (CoRL)*, 2018, pp. 306–316.

[7] A. Jaiswal, Y. Wu, P. Natarajan, and P. Natarajan, "Class-agnostic Object Detection," in *Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 919–928.

[8] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3D Object Proposals for Accurate Object Class Detection," in *Neural Information Processing Systems (NIPS)*, 2015, pp. 424–432.

[9] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z. C. Marton, and R. Triebel, "Unknown Object Segmentation from Stereo Images," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4823–4830.

[10] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, "A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization," *Psychological Bulletin*, vol. 138, no. 6, pp. 1172–1217, 2012.

[11] B. Xiong, J. S. Dutt, and G. Kristen, "Pixel Objectness: Learning to Segment Generic Objects Automatically in Images and Videos," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2677–2692, 2019.

[12] Y. Shi, J. Huang, X. Xu, Y. Zhang, and K. Xu, "StablePose: Learning 6D Object Poses from Geometrically Stable Patches," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 217–15 226.

[13] D. M. de Oliveira, C. C. B. Viturino, and A. G. S. Conceicao, "6D Grasping Based On Lateral Curvatures and Geometric Primitives," in *Latin American Robotics Symposium (LARS)*, 2021, pp. 138–143.

[14] K. Yang and X. Chen, "Unsupervised Learning for Cuboid Shape Abstraction via Joint Segmentation from Point Clouds," *Transactions on Graphics*, vol. 40, no. 4, pp. 1–11, 2021.

[15] B. Wen and K. Bekris, "BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models," in *Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8067–8074.

[16] G. Pitteri, S. Ilic, and V. Lepetit, "CorNet: Generic 3D Corners for 6D Pose Estimation of New Objects without Retraining," in *International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2807–2815.

[17] G. Pitteri, A. Bugeau, S. Ilic, and V. Lepetit, "3D Object Detection and Pose Estimation of Unseen Objects in Color Images with Local Surface Embeddings," in *Asian Conference on Computer Vision (ACCV)*, 2020, pp. 38–54.

[18] A. Kundu, Y. Li, and J. M. Rehg, "3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3559–3568, 2018.

[19] BlenderOnlineCommunity, "Blender - a 3D modelling and rendering package," Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: https://www.blender.org

[20] H. Xu and J. Zhang, "AANet: Adaptive Aggregation Network for Efficient Stereo Matching," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1959–1968.

[21] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 6569–6578.

[22] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7283–7290.

[23] J. Drogemuller, C. X. Garcia, E. Gambaro, M. Suppa, J. Steil, and M. A. Roa, "Automatic generation of realistic training data for learning parallel-jaw grasping from synthetic stereo images," in *International Conference on Advanced Robotics (ICAR)*, 2021, pp. 730–737.

[24] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen Object Instance Segmentation for Robotic Environments," *Transactions on Robotics*, vol. 37, no. 5, pp. 1343–1359, 2021.

[25] D. V. Vargas, B. Liao, and T. Kanzaki, "Perceptual Deep Neural Networks: Adversarial Robustness through Input Recreation," *arXiv:2009.01110*, 2020. [Online]. Available: http://arxiv.org/abs/2009.01110

[26] A. Kriegler, C. Beleznai, and M. Gelautz, "Evaluation of Monocular and Stereo Depth Data for Geometry-Assisted Learning of 3D Pose," in *OAGM Workshop*, 2021, pp. 1–7.

[27] O. Kundu and S. Kumar, "A Novel Geometry-based Algorithm for Robust Grasping in Extreme Clutter Environment," in *International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–8.

[28] M. Durner and W. Boerdijk, "Stereo Instances on Surfaces (STIOS)," 2021. [Online]. Available: https://zenodo.org/record/4706907

[29] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP Challenge 2020 on 6D Object Localization," in *European Conference on Computer Vision Workshops (ECCVW)*, 2020, pp. 577–594.

[30] T. Kollar, M. Laskey, K. Stone, B. Thananjeyan, and M. Tjersland, "SimNet: Enabling Robust Unknown Object Manipulation from Pure Synthetic Data via Stereo," in *Conference on Robot Learning (CoRL)*, 2021, pp. 938–948.

[31] C. Romanengo, A. Raffo, Y. Qie, N. Anwer, and B. Falcidieno, "Fit4CAD: A point cloud benchmark for fitting simple geometric primitives in CAD objects," *arXiv:2105.06858*, 2021. [Online]. Available: http://arxiv.org/abs/2105.06858

[32] H. Fang, "Geometric modeling of man-made objects at different level of details," Ph.D. dissertation, Université Côte d'Azur, 2019.

[33] R. Hachiuma and H. Saito, "Pose Estimation of Primitive-Shaped Objects from a Depth Image Using Superquadric Representation," *Applied Sciences*, vol. 10, no. 16:5442, 2020.

[34] R. T. Chin and C. R. Dyer, "Model-Based Recognition in Robot Vision," *Computing Surveys*, vol. 18, no. 1, pp. 67–108, 1986.

[35] J. Zhang, J. J. Cao, H. R. Zhu, D. M. Yan, and X. P. Liu, "Geometry Guided Deep Surface Normal Estimation," *CAD Computer Aided Design*, vol. 142, no. C, 2022.

[36] R. Fan, H. Wang, B. Xue, H. Huang, Y. Wang, M. Liu, and I. Pitas, "Three-Filters-to-Normal: An Accurate and Ultrafast Surface Normal Estimator," *Robotics and Automation Letters*, vol. 6, no. 3, pp. 5405–5412, 2021.

[37] N. Nejatishahidin, P. Fayyazsanavi, and J. Kosecka, "Object Pose Estimation using Mid-level Visual Representations," *arXiv:2203.01449*, 2022. [Online]. Available: http://arxiv.org/abs/2203.01449

[38] C. Wan, A. Yao, and L. Van Gool, "Hand Pose Estimation from LocalSurface Normals," in *European Conference of Computer Vision (ECCV)*, 2016, pp. 554–569.

[39] C. Xu, J. Chen, M. Yao, J. Zhou, L. Zhang, and Y. Liu, "6DoF Pose Estimation of Transparent Object from a Single RGB-D Image," *Sensors*, vol. 20, no. 23, pp. 1–19, 2020.

[40] A. Bansal, B. Russell, and A. Gupta, "Marr Revisited: 2D-3D Alignment via Surface Normal Prediction," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5965–5974.

[41] M. Dani, K. Narain, and R. Hebbalaguppe, "3DPoseLite: A compact 3d pose estimation using node embeddings," in *Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1877–1886.

[42] Y. Ge, J. Zhao, and L. Itti, "Pose Augmentation: Class-agnostic Object Pose Transformation for Object Recognition," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 138–155.

[43] A. Grabner, P. Roth, and V. Lepetit, "GP2C: Geometric projection parameter consensus for joint 3D pose and focal length estimation in the wild," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 2222–2231.

[44] Y. Xiao, X. Qiu, P. A. Langlois, M. Aubry, and R. Marlet, "Pose from Shape: Deep pose estimation for arbitrary 3D objects," in *British Machine Vision Conference (BMVC)*, 2019, pp. 1–18.

[45] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen, "FS6D: Few-Shot 6D Pose Estimation of Novel Objects," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6814–6824.

[46] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou, "OnePose: One-Shot Object Pose Estimation without CAD Models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6825–6834.

[47] M. Z. Irshad, T. Kollar, M. Laskey, K. Stone, and Z. Kira, "CenterSnap: Single-Shot Multi-Object 3D Shape Reconstruction and Categorical 6D Pose and Size Estimation," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 632–10 640.

[48] E. Guillou, D. Méneveaux, E. Maisel, and K. Bouatouch, "Using Vanishing Points for Camera Calibration and Coarse 3D Reconstruction from A Single Image," *The Visual Computer*, vol. 16, no. 7, pp. 396–410, 2000.

[49] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 3828–3838.

[50] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.

[51] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the Importance of Stereo for Accurate Depth Estimation: An Efficient Semi-Supervised Deep Neural Network Approach," *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1120–11 208, 2018.

[52] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Asian Conference on Computer Vision (ACCV)*, 2012, pp. 548–562.

[53] T. Hodăn, J. Matas, and Š. Obdržálek, "On Evaluation of 6D Object Pose Estimation," in *European Conference on Computer Vision Workshops (ECCVW)*, 2016, pp. 609–619.

[54] P. Jaccard, "The Distribution of the Flora in the Alpine Zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[55] B. Calli, P. Abbeel, S. Member, A. M. Dollar, and S. Member, "The YCB Object and Model Set," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510—-517.

[56] F. Persia, F. Bettini, and S. Helmer, "An interactive framework for video surveillance event detection and modeling," 2017, pp. 2515–2518.