i2c-net: Using Instance-Level Neural Networks for Monocular Category-Level 6D Pose Estimation

Alberto Remus, Salvatore D'Avella, *Graduate Student Member, IEEE*, Francesco Di Felice, Paolo Tripicchio, and Carlo Alberto Avizzano

***Abstract*—Object detection and pose estimation are strict requirements for many robotic grasping and manipulation applications to endow robots with the ability to grasp objects with different properties in cluttered scenes and with various lighting conditions. This work proposes the framework *i2c-net* to extract the 6D pose of multiple objects belonging to different categories, starting from an instance-level pose estimation network and relying only on RGB images. The network is trained on a custom-made synthetic photo-realistic dataset, generated from some base CAD models, opportunely deformed, and enriched with real textures for domain randomization purposes. At inference time, the instance-level network is employed in combination with a 3D mesh reconstruction module, achieving category-level capabilities. Depth information is used for post-processing as a correction. Tests conducted on real objects of the YCB-V and NOCS-REAL datasets outline the high accuracy of the proposed approach.**

***Index Terms*—Perception for grasping and manipulation, deep learning for visual perception, RGB-D perception.**

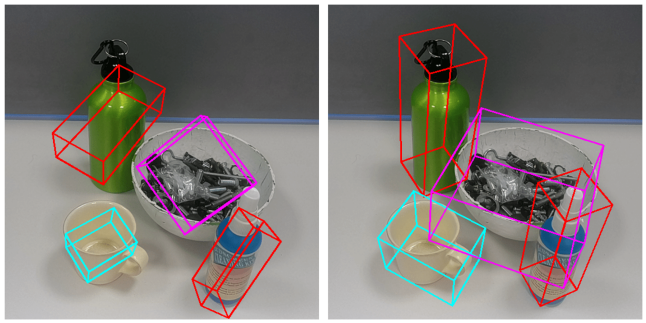


Fig. 1. On the left, an instance-level network that cannot generalize to unseen instances in the wild. On the right, the same instance-level enriched with the presented *i2c-net* framework shows category-level capabilities.

I. Introduction

NOWADAYS, robots are used in a wide range of applications, including advanced manufacturing [1], human-robot [2] collaboration, and logistics [3] that require a high level of autonomy. Two of the primary tasks for robots in such applications are object grasping and manipulation, and robots have to show the ability to adapt to the changing environment while interacting with the surroundings to perform such tasks efficiently in line with the concept of Industry 4.0. A key factor is the grasping of a variety of objects. In order to foster the gap, an important aspect for autonomous and reliable grasping of arbitrary objects involves object detection and pose estimation, which are challenging tasks as objects can present different sizes, material properties, and texture appearances, and they can be occluded in cluttered scenes with different lighting conditions. In addition, a desirable factor is that the method should be fast enough, especially for stringent cycle times in industries. Therefore, even though a lot has been done in recent years, pose estimation for autonomous and reliable grasping of different objects remains an open challenge [4].

Thanks to the advantages of deep learning methods [5], classification, detection [6], and segmentation [7] of objects from images received a significant step forward in the past decades. Instead, pose estimation from a single image is not yet a mature field, and there is still space for improvements toward a reliable solution. One problem is that extracting 3D information from a single color image is an ill-posed problem since the structures of the objects are retrievable only up-to-scale. The other aspect is the lack of labeled real data, whose collection is a difficult and time-consuming task. Humans can rely on stereo-vision or eye motion, and they can also exploit a strong knowledge of the surrounding environment. In that direction, some approaches exist that use multiple points of view to extract the pose of an object or rely on point clouds [8]. However, the computation time is long and increases with the number of viewpoints. An alternative to manually labeled data is the use of synthetic photo-realistic datasets [9] that, in combination with the sim-to-real transfer, allow for a high number of training data and can also be applicable to real-world scenarios.

Estimating the 6D pose of an object from an RGB image, taking into account also the scale factor, requires adding 3D information. One of the most diffused approaches is to use 3D Computer Aided Design (CAD) models of the objects composed of vertices and faces. Doing this, most of the approaches are constrained to objects whose CAD models are used during

Manuscript received 19 October 2022; accepted 7 January 2023. Date of publication 27 January 2023; date of current version 6 February 2023. This letter was recommended for publication by Associate Editor M. Li and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported by the Leonardo Company S.p.A. under Grant LDO/CTI/P/0026995/21, July 2nd, 2021. *(Corresponding author: Salvatore D'Avella.)*

The authors are with the Department of Excellence in Robotics and AI, Mechanical Intelligence Institute, Scuola Superiore Sant'Anna, 56127 PI Pisa, Italy (e-mail: alberto.remus@santannapisa.it; salvatore.davella@santannapisa.it; francescodifelice96@gmail.com; p.tripicchio@santannapisa.it; carlo@sssup.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3240362, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3240362

training, thus hampering generalization, even to objects of the same category [10]. If available, an alternative is to use depth information adding a higher computational burden. However, simulated depth images are more exposed to the sim-to-real problem than RGB images.

This work presents *instance-to-category net* or *i2c-net* to extract the 6D pose of multiple objects belonging to certain categories starting from an instance-level pose estimation network and exploiting a custom-made synthetic dataset for training. The idea is to extract as much 3D information as possible from RGB images and available CAD models and use depth information for post-processing as a correction. Considering some known categories, like, for example, bottles or cans, a few $base\,CAD$ models, i.e., CAD models for known objects that encompass the diverse shapes of the objects in that category, are used as a starting point for a deformation procedure. Such deformations generate new object models that are enriched with real textures for domain randomization purposes. The instance-level pose estimation network can be trained on such augmented photo-realistic images, and at inference time it is used in combination with a 3D mesh reconstruction network achieving category-level capabilities. Several tests are conducted on real objects of the YCB-V [11] and NOCS-REAL 275 [12] datasets to assess the performance of the proposed framework. Fig. 1 shows a qualitative comparison between the proposed framework and the original GDR-Net, which demonstrated to have promising performance as an instance-level pose estimation network.

The rest of the paper is organized as follows: Section II describes the considered problem, related works, and open challenges; Section III details the structure of the presented pipeline, from a general viewpoint to a fine-grained description of the neural network models employed; Section IV collects experimental results with qualitative and quantitative comparisons with state-of-the-art approaches; and Section V is devoted to conclusions by summarizing the contribution of the work.

II. Problem Definition and Related Work

One of the key aspects of autonomous and reliable grasping is the pose estimation of the target object. Existing approaches that extract the 6D pose information from a single RGB(-Depth) image can be mainly distinguished in *instance-level* and *category-level* methods. The first type is biased by the CAD model and texture properties of the object used during training and does not properly work if changes are applied to such an object. The instance-level pose estimation problem can be formalized as follows: given a set of images $\mathbf{I}$ and a set of objects $\mathbf{O}$ for which the CAD model is available, the objective is to find, for each RGB(-D) image $I_j \in \mathbf{I}$ and object instance $M_i \in \mathbf{O}$, the mapping $(I_j, M_i) \longmapsto (\mathbf{R_{ij}}, \mathbf{t_{ij}})$, where $\mathbf{R_{ij}} \in SO(3)$ is the object rotation matrix and $\mathbf{t_{ij}} \in \mathbb{R}^3$ is the 3D translation vector for the particular object instance with respect to the camera frame. The presence of the 3D CAD model resolves the ill-posedness of the problem of extracting 3D information from a single color image, but jeopardizes the generalization capabilities of the method. Indeed, instance-level approaches may have practical applications in scenarios with a fixed number of objects, while their effectiveness drops with unseen instances. In the latest years, this type of network has witnessed an impressive improvement concerning accuracy and speed: from pioneering works in YOLO-6D [13], and Dope [10] to more recent approaches in GDR-Net [14], and SO-Pose [15]. Depth information can increase the accuracy of the estimation along with higher computational burden and lower real-time performance [11].

Passing from instance-level to category-level, the problem can be formalized as follows: given a set of images $\mathbf{I}$, a set of categories of objects sharing some common properties $\mathbf{C} = \cup_{k=1}^{N} C_k$, and a set of objects $\mathbf{O}$ belonging to such different categories, the objective is to find, for each RGB(-D) image $I_j \in \mathbf{I}$ and object instance $M_i \in \mathbf{O}$ the mapping $(I_j, M_i) \longmapsto (C_k, \mathbf{R_{ij}}, \mathbf{t_{ij}}, \mathbf{s_{ij}})$, where C_k is the category which the object M_i belongs to, and $\mathbf{s_{ij}} \in \mathbb{R}^3$ its size. It is worth noticing that the CAD model is not available for each object instance during inference otherwise the problem ends up in the instance-level setting. For that reason, the $\mathbf{s_{ij}}$ term represents the size of the 3D bounding box tightly surrounding the object to solve the scale ambiguity.

Such a scale factor is relevant for robotic grasping scenarios to determine whether the target object can fit the gripper opening. This family of problems leaves space for investigation, especially concerning the exploitation of RGB information. Indeed, state-of-the-art approaches, like Shape-Prior [16] (or SPD), and FS-Net [17], instead of using only color images during training, extract a point cloud from the depth image of the observed object and apply a set of 3D deformations to augment the available data and catch intra-category salient features. Existing methods can also be classified as single-stage or multiple-stage estimators. In the former, the training process of both detection and pose estimation outputs is performed jointly, as in the case of key-point-based approaches presented in [10] or [18]. Multiple-stage methods like [14] or [15], instead, can benefit from higher modularity considering the vast improvement that 2D detectors have been going through.

A dual problem to 6D pose estimation is 3D model reconstruction. *Explicit shape representation* [19] is a class of approaches that approximates a surface as a function of 2D coordinates and enhances the granularity of this approximation by increasing the number of edges, triangles, or vertices at the expense of additional processing time. Variations in the objects' topology within the same category can hamper performances due to the possible presence of holes and gaps inside the 3D model, thus leaving space for the so-called *implicit surface representations* [20]. Signed Distance Functions (SDF) are the most widespread implicit model that computes the distance to the closest surface for each considered 3D point and assigns a positive (negative) sign if the point is inside (outside) [21]. In addition, a novel approach to 3D reconstruction is constituted by Neural Radiance Field (NeRF) [8] that generates 3D scenes from a sequence of RGB images knowing the poses of the cameras. Then, by sampling 3D coordinates and 2D viewing directions for each camera ray, it is possible to feed a neural network and get an RGB-density image as an output to reconstruct the final 3D mesh. The main drawback of such an approach regards

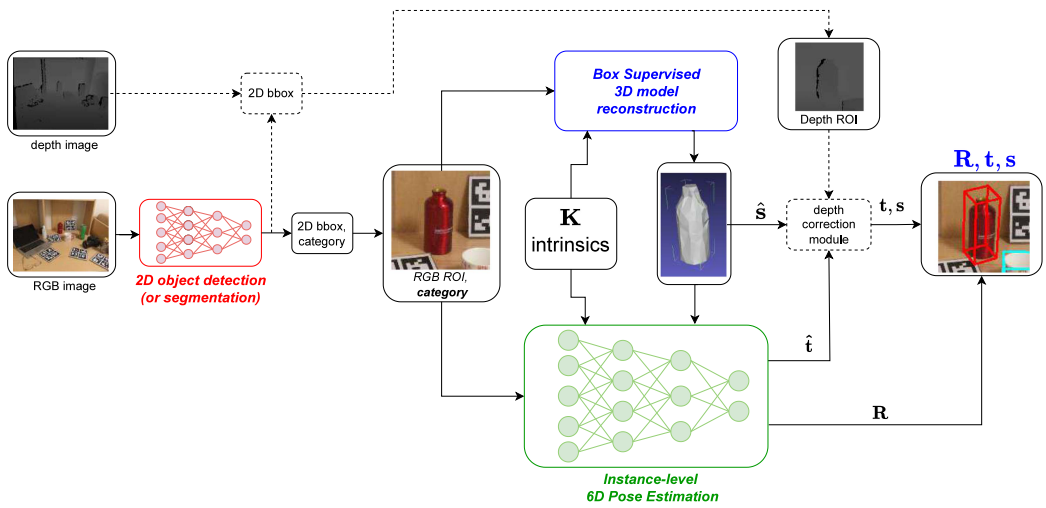


Fig. 2. The presented method aims to find the 6D pose and 3D object size from a single RGB image, with the depth image used to correct the estimation. It encompasses three neural network modules for 2D object detection (or segmentation), instance-level 6D pose estimation, and box-supervised 3D model reconstruction. Dashed components are used at inference time only.

the high computational burden, both during the training and testing phases, in the order of many seconds or minutes [8]. Instant-Nerf [22] goes in this direction by reducing inference time by a factor of 10 to 100, however, NeRF-based methods remain more suitable for 3D mesh reconstruction when multiple views are available, an assumption not always true in fixed camera robotics settings.

III. Methodology

The proposed approach presents a framework for 6D category-level pose estimation starting from an instance-level network (Fig. 2). The objective is to show that instance-level models that achieved high performance in 6D pose estimation, but suffer from low generalization capabilities, can still be effective for category-level tasks. The point is that, in general, RGB instance-level architectures encompass backbones that can have general-purpose applications like classification and detection. Therefore, they can distill the most salient features if exposed to various instances of the same category to output results in a latent space. From that, other neural networks can retrieve geometrical and shape information about the considered objects. The work investigates in the experimental campaign some categories that are common in the research community such as *banana*, *bottle*, *bowl*, *camera*, *can*, *laptop*, and *mug*. This section gives an overview of the proposed approach and details each module of the pipeline, as well as the dataset generation procedure required for the training.

The design of the architecture is modular, allowing changing the components with the most recent research advances in computer vision as they respect the same required interface. In particular, the pipeline starts from a 2D object detection module that takes as input an RGB image and outputs the 2D bounding box of the target. Then, a 3D model reconstruction module exploits the cropped 2D RGB image to generate the object 3D mesh with absolute scale along the coordinate axes $\hat{\mathbf{s}} \in \mathbb{R}^3$. Finally, the 6D instance-level estimation module combines the cropped 2D RGB image and the 3D mesh to obtain the 3D rotation matrix $\mathbf{R}$ and 3D translation vector $\hat{\mathbf{t}}$. A depth-correction module can be used in post-processing to improve the estimates $\hat{\mathbf{t}} \in \mathbb{R}^3$ and $\hat{\mathbf{s}} \in \mathbb{R}^3$ through cropped depth image of the considered target object to obtain final translation $\mathbf{t}$ and scale $\mathbf{s}$.

Given the significant recent advances in computer vision in the field of 2D object detection, the proposed approach relies on an off-the-shelf object detector, i.e., YOLOv5 [6] that takes the full camera frame in input and provides both the class of a given object and its 2D bounding box to the following stages of the network.

Cluttered scenes may cause a degradation in instance-level 6D pose estimation, and consequently, at inference time, it may be convenient to replace the 2D detection module with a 2D instance segmentation like Detectron [7]. This is useful to remove ambiguity in the single view by masking occluding objects, and increasing the attention on the object of interest. The benefits of segmentation are opportunely analysed in the experimental campaign.

A. Photo-Realistic Dataset

A custom-made purely photo-realistic dataset is generated to train the proposed framework. The dataset is built through BlenderProc, a tool based on Blender graphic engine capable of rendering RGB images with an acceptable sim-to-real gap, together with 6D object pose annotations [9]. The constructed dataset contains 9900 images for each of the considered categories, subdivided into 300 scenes with 33 images each, to simulate various camera viewpoints, illumination, and background conditions. The target is to make the network generalize to objects whose CAD is not available within the same category. The method employs 15 base models for the *camera* object, 300 for *laptop*, and 100 for the other NOCS' categories (*bottle*, *bowl*, *can*, *mug*) that are more related to manipulation tasks.

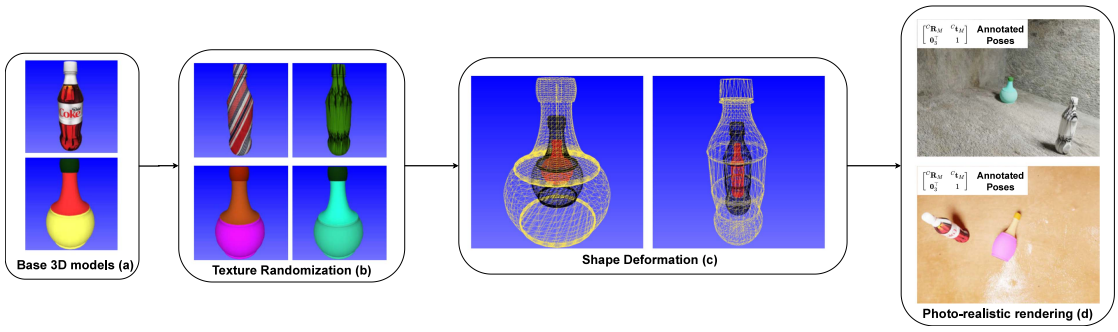


Fig. 3. Starting from a set of predefined CAD models [23] (a), a first randomization process (b) applies random textures to each instance, a shape randomization process (c) further augments the 3D dataset along the coordinate axes, and finally, a photo-realistic renderer (d) generates annotated RGB images.

Accordingly, ShapeNet [23] is a valuable source to gather the *base* CAD models. Taking inspiration from Shape-Prior [16] and FS-Net [17], such meshes are randomly enlarged or shrunk along their coordinate axes to augment the available dataset.

The deformation is applied to each side of the bounding box within a given range that is a tunable parameter selected depending on the required needs. A large range may be useful to push for generalization over the same category, but it is worth not exaggerating since a too broad range can lead to losing the main properties of the category's shape. The main innovation compared to other category-aware works is that during training, the proposed method uses only synthetic RGB images through an aleatoric set of colored textures applied to the deformed 3D models to increase generalization and reduce overfitting to particular surface patterns. In this way, the neural network model can focus more on the shape to reconstruct the geometry of the objects despite the intra-class variability. To this end, in industrial settings that work per category of object, the number of base models can be possibly reduced according to such diversity, also addressing new incoming products in line with the flexibility concept of Industry 4.0. Fig. 3 details the augmentation pipeline, where 3D CAD models can be endowed with some color and texture attributes picked at random. Such modified meshes are then passed to Blenderproc in charge of randomly scaling the models. In addition, Fig. 3 shows a qualitative representation of the shape deformation along the coordinate axes: the yellow and red wireframes outline the maximum and minimum deformation possible for a given category, compared to the base model depicted in black. Furthermore, Blenderproc places the objects in a photo-realistic environment, where a wide variety of backgrounds and illumination conditions are simulated. Finally, the software computes the 6D pose of each object with respect to each camera view. A strength of this method is the possibility to develop an *in-house* approach: all the data generation process is fully under the control of the user, and further developments are not constrained by the lack of access to real-world data annotation tools as in [24].

B. RGB 6D Pose Estimation

This module exploits an instance-level network to extract objects' 6D pose (3D rotation $\mathbf{R}$ and 3D translation $\hat{\mathbf{t}}$ with respect to the camera frame) from an RGB input, the associated 3D model, and the intrinsic parameters $\mathbf{K} \in \mathbb{R}^{3\times3}$ of the employed visual sensor.

Given the high number of instance-level networks, the choice of the particular model comes from a trade-off between accuracy, speed, and flexibility. In this work, Geometry-Guided Direct Regression Network or *GDR-Net* is the reference baseline. This architecture is based on an encoder-decoder encompassing a ResNet [25] backbone and a custom decoder to reconstruct an internal representation of the geometry of the seen object's feature space. Such a capability is appealing for an extension to category-level scenarios: the idea is to make the network learn the 3D geometry common to a given class instead of focusing on the details concerning a few instances of various categories, as in instance-aware settings. The model is a fully differentiable architecture enabling end-to-end training of the encoder-decoder and the Perspective-n-Point (PnP) modules in charge of finding the 6D pose of the 3D model given the features extracted from the RGB input [14]. Nonetheless, the recovery of the relative and, in particular, the absolute scale along the coordinate axes of the item from a single image is still beyond the possibilities of instance-level networks. To this end, the 3D reconstruction module becomes essential to carry out a successful category-aware estimation.

C. 3D Model Reconstruction

This neural model is trained over the same set of photo-realistic RGB images of the considered categories and learns how to infer the 3D mesh of unseen instances from a single viewpoint. In this work a *box-supervised 3D model reconstructor* is developed on top of Multi-Category Mesh Reconstruction (MCMR) [26] as in Fig. 4. In detail, the original architecture makes use of the weakly perspective projection model, mostly suitable when objects have a similar distance along the camera axis [27]. Moreover, it encompasses a fully connected network to regress 2D translation, 3D rotation, and 1D scaling factor. However, by delegating the pose estimation task to the instance-level network, it is possible to remove the above restriction and get the full 3D translation. Consequently, in order to retrieve the proper shape and scale of the 3D model, it is useful to exploit prior knowledge acquired during the learning phase and stored in 3D models called *meanshapes*. A meanshape can be regarded as a latent feature that condenses the salient information

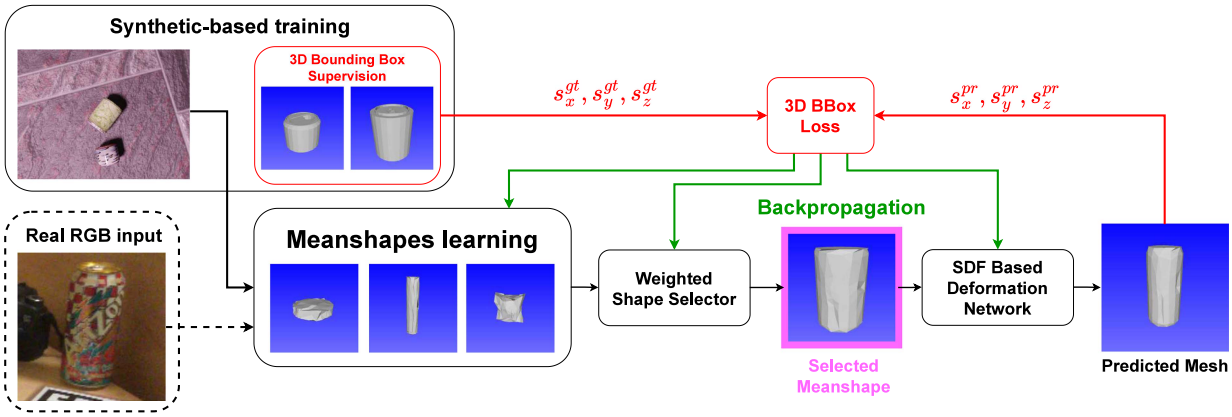


Fig. 4. Box-supervised 3D model reconstruction, from monocular synthetic RGB 6D pose estimation dataset, a set of meanshapes is learned. The newly introduced 3D bounding box loss (in red) allows backpropagating information (in green) about the object's size to the other neural components of the model. Dashed elements are used at inference time only.

about the structure of the objects seen at training time. It is possible to learn multiple meanshapes and let a classifier select the most suitable one with respect to the image features. In the presented approach, the loss function in [26] is completed with the 3D bounding box supervision by introducing the term $\mathcal{L}_{3Dbbox} = (s_x^{pr} - s_x^{gt})^2 + (s_y^{pr} - s_y^{gt})^2 + (s_z^{pr} - s_z^{gt})^2$, where s_k is the length of the side $k \in \{x, y, z\}$ of the object's predicted (pr) and ground truth (gt) 3D bounding boxes. In such a way, the information is back-propagated to all the neural components, and in particular to the SDF-based network [21] to regress the proper scale of the object as well as its shape, with a good real-time performance. In order to perform a comparison between predicted and ground truth meshes, it is convenient to align such 3D models in terms of position. Therefore, at every training step, the predicted mesh is shifted so that its bounding box's centroid becomes the 3D point [0,0,0], as by convention adopted in the NOCS dataset. In principle, the actual absolute scale $\mathbf{s}$ would be unknown, however, thanks to 3D box supervision, its approximation $\hat{\mathbf{s}}$ is retrievable up to the range of dimensions considered in the dataset generation process, as shown in Fig. 3.

D. Depth Correction Module

As detailed in Fig. 2, the 3D CAD model is not accessible at inference time. Consequently, the instance-level 6D pose estimation network needs to be completed with the above-mentioned 3D reconstruction module, which is sufficient for a proper 3D rotation regression, given the correct relative scale of the 3D model. On the other hand, a further correction may be performed on the 3D translation to compensate for pose estimation errors due to the absolute scale. To this purpose, it is convenient to exploit a rendered depth map obtained by rendering the predicted mesh, which comes from the 3D model reconstruction module, by using the predicted rotation and translation provided by the instance-level network. At this point, a correction based on stereo depth may be applied, as depicted in Fig. 5 and outlined in the following steps:

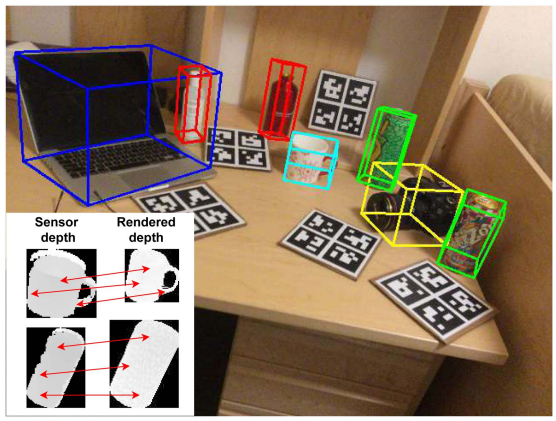


Fig. 5. Correction procedure of the pose estimation, the comparison between rendered and measured depth maps can be used to properly rescale translation coordinates.

- finding the *width* σ_u and *height* σ_v ratios between rendered and measured depth maps;
- sampling p 2D points on the rendered depth map z_r $\{(u_1^r, v_1^r), \ldots (u_p^r, v_p^r)\}$. For experiments $p = 8$ shows to be enough for each estimation;
- finding the corresponding points on the measured depth map z_m:

$$\begin{cases} u_i^m = \sigma_u \, u_i^r \\ v_i^m = \sigma_v \, v_i^r \end{cases}$$

- finding the final depth ratio as:

$$\sigma_z = \frac{1}{p} \sum_{i=1}^{p} \frac{z_m(u_i^m, v_i^m)}{z_r(u_i^r, v_i^r)}$$

so that both the 3D bounding box and the translation vector can be properly scaled to $\mathbf{t} = \sigma_z \hat{\mathbf{t}}$ and $\mathbf{s} = \sigma_z \hat{\mathbf{s}}$.

Despite the proposed method is not designed for heavily cluttered scenes, it can address self-occlusions, as well as mild occlusions where the 2D bounding box's size is not affected

TABLE I
PERFORMANCE COMPARISON ON NOCS REAL-275 DATASET [12] BETWEEN *I2C-NET* AND VARIOUS STATE-OF-THE-ART APPROACHES, SUBDIVIDED INTO METHODS REQUIRING REAL (R) ANNOTATIONS (ANN.) OR JUST SYNTHETIC (S) ONES

Method	Ann. type	$3D_{25}$	$3D_{50}$	5° 5 cm	10° 5 cm	10° 10 cm
NOCS [12]	R+S	84.9	80.5	10.0	25.2	28.8
SPD [16]	R+S	83.4	77.3	21.4	54.1	56.2
FS-Net [17]	R+S	95.1	92.2	**28.2**	**60.8**	64.6
Gao et al. [31]	**S**	68.6	27.7	7.8	17.1	-
CPPF [32]	**S**	78.2	26.4	16.9	44.9	46.0
i2c-net	**S**	**99.96**	**92.46**	24.62	49.42	**67.05**

TABLE II
ABLATION STUDY ON THE IMPACT OF DEPTH CORRECTION AND INSTANCE SEGMENTATION'S REMOVAL ON $n°$, $N\ cm$ METRIC TESTED ON NOCS REAL, ALONG WITH HOW MUCH THE METHOD IS CAPABLE TO COMPENSATE FOR THE ABSENCE OF 3D CAD MODEL

Row no.	3D CAD available	Depth correct.	2D seg	**5° 5 cm**	**10° 5 cm**	**10° 10 cm**
1	✓	✓	✓	44.93	71.40	75.25
2	✓	✓	✗	36.02	62.97	66.52
3	✓	✗	✓	37.48	56.37	71.75
4	✓	✗	✗	23.32	39.65	56.16
5	✗	✓	✓	24.62	49.42	67.05
6	✗	✓	✗	22.33	43.65	60.70
7	✗	✗	✓	8.84	14.33	30.45
8	✗	✗	✗	10.41	18.03	31.66

by other objects, since if points are present in the sensor depth map but not in the rendered depth map, then another pair is sampled. The assumption behind this choice is the possibility to rely on a grasping policy that prioritizes less occluded objects to reduce the clutter for the next grasping. Another relevant feature of this approach is the reduced complexity compared to other corrections methods like Iterative Closest Point (ICP) on 3D point clouds [28].

IV. EXPERIMENTAL RESULTS

To assess the performance of the presented method, NOCS-REAL 275 [12] dataset is used as a widespread benchmark. The proposed architecture is trained on a custom-made photo-realistic dataset, which is not related to NOCS-REAL's test subset used for quantitative evaluation. In addition, YCB-V [11] test set is used to show that the method can also work with categories beyond the NOCS dataset.

The following metrics are introduced to allow a comparison with other approaches:

- 3D Intersection over Union (3D IoU) measures the average precision of the ratio between intersection and union of the predicted and ground truth 3D bounding boxes [29].
- $n\ cm$, $n°$ is the average precision of the predictions with a roto-translational error below n centimeters and n degrees; a symmetric-aware version of the metric can relax the error in case of ambiguities along the axes of symmetry [29];

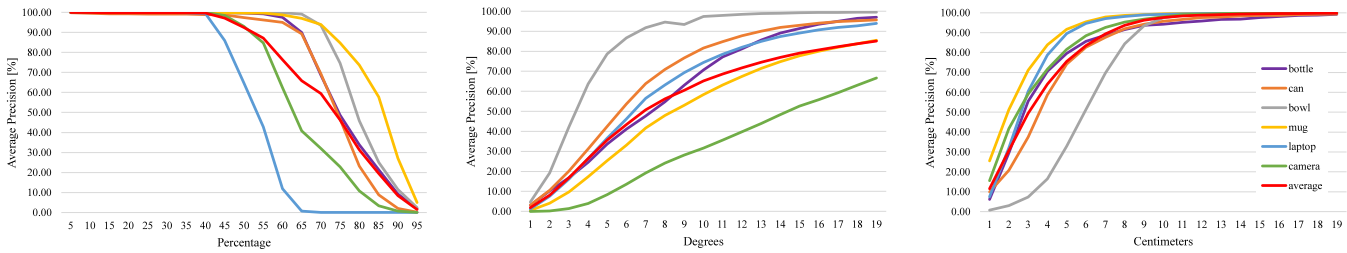
The reference framework for the experimental campaign is PyTorch. Training is carried out on an NVIDIA RTX 3090 GPU (24 GB), while inference on an RTX 3080 Laptop GPU (8 GB). For real-world experiments in the wild, the used device is an RGB-D Luxonis OAK-D camera [30].

Table I shows i2c-net architecture compared to some state-of-the-art approaches, over category-level metrics. The reported performance is referred to the configuration exploiting both depth correction and instance-segmentation at inference time. i2c-net registers the highest accuracy on 3D intersection over union at 25% ($3D_{25}$) and 50% ($3D_{50}$). Concerning the $n°, n\ cm$ metric, i2c-net is the best on $10°, 10\ cm$. It is worth noticing that the majority of other works rely on real-world annotations during training, which hinders a proper extension to categories not included in the NOCS dataset. Conversely, recent methods like CPPF [32] exploit synthetic data only in the learning phase. However, compared to it, i2c-net outlines superior performances on all the reported measurements, thus highlighting that the presented approach is quantitatively effective in a real-world scenario. It must be noted that all the category-level state-of-the-art approaches reported in Table I rely on depth information.

Fig. 6 contains a quantitative performance breakdown over different categories in terms of average precision (AP) for 3D Intersection over Union, rotation error, and translation error. Different thresholds are used to extract the curves for each metric, where the closer the AP value is to 100%, the better. On the other hand, Fig. 7 shows qualitative results related to the analysis.

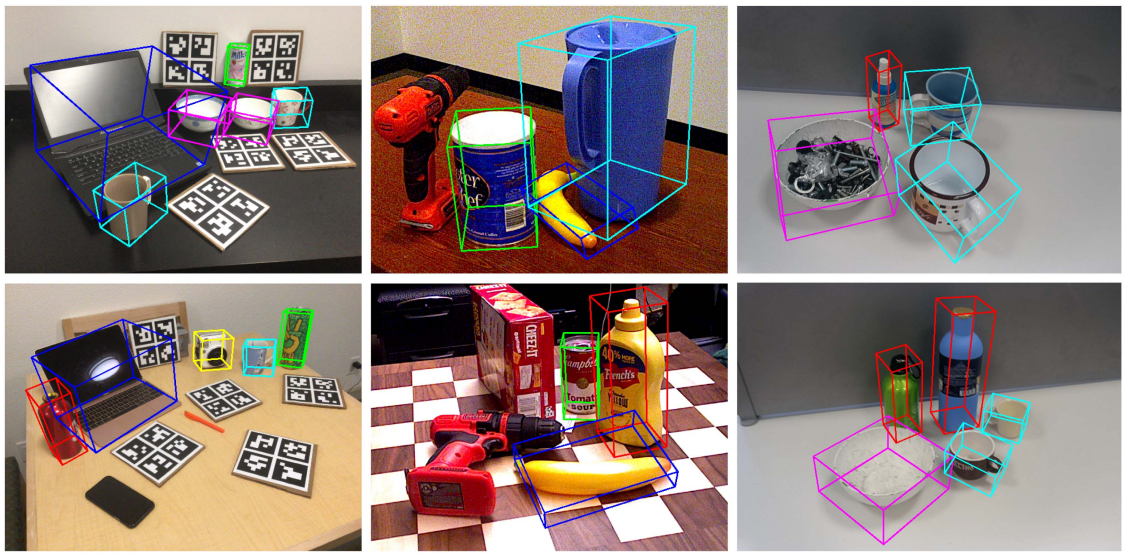
Translation errors show consistent performances among all the classes except for *bowl*, where the 3D reconstruction step may face difficulties in detecting the depth of the hollow from RGB images only. Further pose randomization techniques can be applied to reduce such ambiguities. Conversely, *bowl*, as well as *bottle* and *can* show results above the average regarding the rotation error that does not penalize the different predictions along the symmetry axes, not affecting grasping and manipulation tasks. In addition, symmetric objects benefit from fewer viewpoints required for a proper 3D shape reconstruction. Therefore, increasing that number during training for other categories can provide an improvement. Concerning the *camera* category, results show lower performances on rotation error, due to shape variability that is higher than in other categories. Conversely, the class *laptop* behaves properly on $n°, n\ cm$, while lags on 3D IoU, since the articulated structure of the object may lead to a low overlapping between reconstructed and ground truth 3D models, in spite of good pose estimation.

Table II reports an ablation study to highlight how much the performance of the presented architecture, expressed through the $n°, n\ cm$ metric, degrades by removing different modules. Such an analysis covers depth correction and instance segmentation to feed the network with a masked RGB input. In addition, the first 4 rows of the table consider the case in which the ground truth 3D model is available to give a baseline for the last 4 rows, quantitatively assessing how much the presented method can compensate for the lack of a 3D model by reconstructing it at inference time.



(a) 3D IoU error (b) Rotation error (c) Translation error

Fig. 6. Quantitative performance analysis over different NOCS categories.



(a) NOCS REAL-275 (b) YCB-V (c) Real-world images in the wild

Fig. 7. Qualitative results on some images from NOCS REAL-275 (first column) and YCB-V (second column) test sets, and real-world examples in the wild (third column). All the involved instances are not included in the training dataset.

By comparing rows 6 and 8, and rows 5 and 7, not using the depth correction module impacts between 11.92% and 36.6% in case the 3D model is not available. Similarly, instance segmentation, which can be convenient to counteract occlusions, provides a lighter improvement, between 2.29% and 16.72%, by comparing lines 1-3-5 against 2-4-6 respectively.

It is worth noticing that row 5 presents the best performance in a category-level condition, such as when depth correction and 2D segmentation are both used, while the 3D CAD is not available, as already presented in Table I.

On the other hand, comparing lines 1-2 versus 5-6 respectively highlights that once the 3D model needs to be reconstructed, the performance drops between 8.2% and 21.98% when depth is available, increasing up to 42.04% when i2c-net cannot access any 3D information as in rows 7-8 versus rows 3-4, where, instead, at least the 3D model is available. Despite this difference is not negligible, the generalization gain obtained through 3D mesh reconstruction is consistent since the 3D model of the seen object is not available at inference time.

As depicted in Fig. 1, GDR-Net provides unsuccessful results with instances in the wild, confirming the superiority of i2c-net over its instance-level baseline in the presence of unseen instances not contained in the dataset for the considered categories (mug, bottle, and bowl).

A. Real-World Experiments

Fig. 7 highlights the capabilities of the network to generalize beyond instances seen during training coherently to the quantitative analysis. Since no ground truth is available, the oriented 3D bounding box can show qualitative results over different categories, thanks to the intensive domain, texture, and shape randomization carried out prior to the training phase. In addition, to show the extension of the pipeline to categories not included in NOCS REAL-275, testing on the YCB-V object *banana* is reported in Fig. 7(b), beside the classes in common with the former dataset.

Real-time experiments show an inference time of 60.3 milliseconds (ms) on the RTX 3080 laptop GPU for each pose estimation, averaging over 1000 evaluations. A more detailed breakdown highlights 12.8 ms for the 2D object detection with YOLOv5 small [6] and 27.7 ms for instance-level pose

estimation. The remaining time is split between 3D reconstruction (7.4 ms) and depth correction (12.4 ms) that, together, introduce a 50% overhead. However, even though this represents the price for moving from an instance-level to a more general category-level pipeline, real-time tasks can still be carried out without further optimization.

V. Conclusion

Two of the primary tasks for robots in many applications are object grasping and manipulation. Object detection and pose estimation are fundamental skills to give robots the ability to adapt to the changing environment and grasp a variety of objects that present different sizes, material properties, and texture appearances in cluttered scenes with different lighting conditions.

This work presents *i2c-net* to extract the 6D pose of multiple objects belonging to different categories, starting from an instance-level pose estimation network and exploiting a custom-made synthetic dataset for training. Such a dataset uses some base CAD models for known objects, encompassing the diverse shapes of the objects in that category as a starting point for a deformation procedure, which provides new object models enriched with real textures for domain randomization purposes. The selected instance-level pose estimation network can be trained on such augmented photo-realistic images, and, at inference time, it is used in combination with a 3D mesh reconstruction network achieving category-level capabilities. Depth information is used for post-processing as a correction. Tests conducted on real objects of the YCB-V and NOCS REAL-275 datasets show the high accuracy of the proposed method as well as good real-time performances.

References

[1] S. D'Avella, C. A. Avizzano, and P. Tripicchio, "ROS-industrial based robotic cell for industry 4.0: Eye-in-hand stereo camera and visual servoing for flexible, fast, and accurate picking and hooking in the production line," *Robot. Comput.- Integr. Manuf.*, vol. 80, 2023, Art. no. 102453.

[2] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Auton. Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[3] S. D'Avella, P. Tripicchio, and C. A. Avizzano, "A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper," *Robot. Comput.- Integr. Manuf.*, vol. 63, 2020, Art. no. 101888.

[4] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1677–1734, 2021.

[5] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[6] G. Jocher et al., "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022, doi: 10.5281/zenodo.7347926.

[7] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron 2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 405–421.

[9] M. Denninger et al., "Blenderproc," 2019, *arXiv:1911.01911*.

[10] 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 *Proc. Conf. Robot Learn.*, Oct. 29–31, 2018, vol. 87, pp. 306–316.

[11] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot.: Sci. Syst.*, 2018, doi: 10.15607/RSS.2018.XIV.019.

[12] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2637–2646.

[13] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 292–301.

[14] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16606–16616.

[15] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "SO-Pose: Exploiting self-occlusion for direct 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12396–12405.

[16] M. Tian, M. H. Ang Jr, and G. H. Lee, "Shape prior deformation for categorical 6D object pose and size estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 530–546.

[17] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis, "FS-Net: Fast shape-based network for category-level 6D object pose estimation with decoupled rotation mechanism," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1581–1590.

[18] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-stage keypoint-based category-level object pose estimation from an RGB image," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 1547–1553.

[19] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 52–67.

[20] A. Tewari et al., "Advances in neural rendering," *Comput. Graph. Forum*, vol. 41, no. 2, pp. 703–735, 2022.

[21] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[22] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022.

[23] A. X. Chang et al., "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[24] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7822–7831.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[26] A. Simoni, S. Pini, R. Vezzani, and R. Cucchiara, "Multi-category mesh reconstruction from image collections," in *Proc. IEEE Int. Conf. 3D Vis.*, 2021, pp. 1321–1330.

[27] Z. Zhang, *Weak Perspective Projection*. Berlin, Germany: Springer, 2014.

[28] P. Besl and H. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Mar. 1992.

[29] J. Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Comput. Surv.*, vol. 55, pp. 1–40, 2023.

[30] Luxonis, "Oak-d, depth-ai documentation," 2022. Accessed: Aug. 30, 2022. [Online]. Available: https://shop.luxonis.com/products/oak-d

[31] G. Gao, M. Lauri, Y. Wang, X. Hu, J. Zhang, and S. Frintrop, "6D object pose regression via supervised learning on point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3643–3649.

[32] Y. You, R. Shi, W. Wang, and C. Lu, "CPPF: Towards robust category-level 9D pose estimation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6866–6875.