RGB-D-Based Categorical Object Pose and Shape Estimation: Methods, Datasets, and Evaluation

Leonard Bruns[a,*], Patric Jensfelt[a]

[a]*Division of Robotics, Perception and Learning (RPL), KTH Royal Institute of Technology, Teknikringen 14, 11428 Stockholm, Sweden*

Abstract

Recently, various methods for 6D pose and shape estimation of objects at a per-category level have been proposed. This work provides an overview of the field in terms of methods, datasets, and evaluation protocols. First, an overview of existing works and their commonalities and differences is provided. Second, we take a critical look at the predominant evaluation protocol, including metrics and datasets. Based on the findings, we propose a new set of metrics, contribute new annotations for the Redwood dataset, and evaluate state-of-the-art methods in a fair comparison. The results indicate that existing methods do not generalize well to unconstrained orientations and are actually heavily biased towards objects being upright. We provide an easy-to-use evaluation toolbox with well-defined metrics, methods, and dataset interfaces, which allows evaluation and comparison with various state-of-the-art approaches (https://github.com/roym899/pose_and_shape_evaluation).

Keywords:
pose estimation, shape estimation, shape reconstruction, RGB-D-based perception

1. Introduction

Estimating an actionable representation from raw sensor data is a long standing problem at the intersection of robotics and computer vision. Classic grasp and motion planning algorithms often assume full knowledge of the environment, making them difficult to apply with partial sensor information or requiring elaborate setups to acquire the necessary data. To bridge this gap from limited sensor data to an actionable representation of the environment various algorithms have been proposed often aiming at estimating various object-based quantities, such as bounding boxes, masks, poses, and shapes.

In this work, the problem of categorical[1] pose and shape estimation is considered. Originating from pose estimation of known object instances (in which the shape and appearance is known a priori), categorical pose and shape estimation aims to generalize to arbitrary shape and appearance variations within a category, while still arriving at a posed mesh (i.e., a full object mesh and its pose aligning it with the sensor data) of an object of interest.

Compared to just estimating bounding boxes or instance masks, estimating the full shape and pose of objects promises to further close the gap from partial sensor information to an actionable representation. While pure shape estimation can be sufficient for some tasks [1], categorical pose estimation further provides a per-category reference frame (see Section 3) that can be used in downstream tasks, such as aligning objects [2] and pose-dependent grasp computation (e.g., an upside-down mug has to be grasped differently from an upright mug).

In the last two years, various learning-based categorical pose and shape estimation methods have been proposed. With a large number of changes between different works (i.e., varying loss functions, network architecture, training protocols, etc.) the main contributions of individual works and their relation to prior work can become hard to identify. We provide an overview of existing methods and identify notable commonalities and differences, which so far have been less discussed in the literature.

Regarding datasets, most of the current methods are trained and evaluated on the two datasets proposed by [3], called CAMERA and REAL. The CAMERA dataset is a large dataset of real RGB-D tabletop scenes with synthetic objects superimposed on top of the table. The REAL dataset is a smaller real-world dataset of tabletop sequences with objects that have been scanned and tracked for the purpose of evaluating categorical pose estimation. Notably, both datasets only contain upright objects, which opens the question of how well existing methods generalize to less constrained settings.

To answer this question, we contribute a set of annotations to evaluate unconstrained 6D pose and shape estimation. Our annotations consist of meshes and poses for handheld objects of three categories in the Redwood

*Corresponding author.
Email addresses: `leonardb@kth.se` (Leonard Bruns), `patric@kth.se` (Patric Jensfelt)

[1] The terms categorical, category-level, and per-category are used interchangeably in the literature.

dataset [4]. The objects in this dataset are freely rotated in front of the camera by a human hand causing occlusions. The orientations vary significantly more than in the datasets by [3].

Most methods evaluate pose estimation by following the same evaluation protocol as originally proposed by [3]. The method in [3] combines mask detection and pose estimation into a single network and therefore evaluates pose estimation with *average precision* (AP), which is a common detection metric. However, many of the subsequent methods assume that the mask is an input to their method, making AP an unnatural evaluation metric, as it makes the results unnecessarily difficult to interpret. Therefore, we propose a set of simpler metrics that use ground-truth masks and categories to evaluate pose estimation.

Pose estimation with shape reconstruction was first demonstrated by [5] and [6]. Both methods independently used *chamfer distance* as their reconstruction metric, which is now commonly used by methods performing shape reconstruction. However, [7] showed that chamfer distance is not a good measure of reconstruction quality and other metrics better correlate with perceived reconstruction quality. Therefore, in this work, we advocate for the use of F-score to evaluate reconstruction quality and propose a new evaluation protocol for both shape reconstruction and pose estimation.

To summarize, our contributions are:

- an extensive overview of existing methods for categorical pose and shape estimation,
- a well-defined evaluation protocol,
- a challenging set of annotations to evaluate unconstrained pose and shape estimation,
- a fair evaluation of various state-of-the-art methods, and
- an open-source evaluation toolbox for categorical pose and shape estimation including methods, datasets, and metrics.

2. Related Work

Since the introduction of the BOP benchmark suite [8], great progress has been made in the task of instance-level pose estimation, where a mesh of the target object is available. However, the more general task of category-level pose estimation has only recently received more attention. Compared to instance-level pose estimation, category-level pose estimation is more challenging due to the large possible variations in shape and appearance.

Wang et al. [3] introduced the first deep learning-based method to address the 6D pose estimation problem at the per-category level. They introduced two datasets: CAMERA, a synthetic dataset that combines real scenes with meshes from the ShapeNet dataset [9] and REAL, a smaller real-world dataset that is mainly used for fine-tuning and evaluation. Their method is based on the *normalized object coordinate space* (NOCS), in which objects of one category have a common alignment. The projection of the NOCS coordinates in the image plane (also called NOCS map) is predicted by extending Mask R-CNN [10] with an additional head. From this prediction, the 6D pose and scale can be estimated by employing the Umeyama algorithm [11] with RANSAC [12] for outlier removal.

In [3], the NOCS map was predicted using only RGB information. Since geometry typically varies less than appearance for a fixed category, several methods were proposed to more directly incorporate the observed point set into the prediction. Chen et al. [5] introduced *canonical shape space* (CASS), which refers to a latent shape representation learned in a variational autoencoder [13] framework. Their method regresses the latent shape, orientation, and position directly from the cropped image and the set of observed points. As a byproduct of their method, they also reconstruct the full canonical point set. Tian et al. [6] introduced *shape prior deformation* (SPD), which uses a canonical point set and predicts a deformation based on the observed RGB-D information.

Various follow-up works modify SPD's network architecture and training scheme to achieve further improvements. For example, CR-Net [14] uses a recurrent architecture to iteratively deform the canonical point set, SGPA [15] uses a transformer architecture to more effectively adjust the canonical point set, and DPDN [16] employs consistency-based losses for additional self-supervised learning.

Another notable line of work follows an analysis-by-synthesis approach. Analysis-by-synthesis approaches can be seen as the category-level analog to mesh-based pose refinement [17, 18, 19, 20] for instance-level pose estimation. Since at category-level no ground-truth mesh is available for refinement, analysis-by-synthesis approaches typically integrate a generative shape model into the pipeline to jointly optimize a latent shape representation and the 6D pose at inference time.

Chen et al. [21] proposed the first such analysis-by-synthesis framework in which the latent representation of a generative model is iteratively optimized to fit the observed color image. The generative model allows to generate novel views of the object, but a full reconstruction is not readily available. Instead of minimizing the RGB discrepancy, iCaps [22] and SDFEst [23] perform iterative optimization based on the depth observation. A full RGB-D-based optimization is proposed in [24].

Aside from these RGB-D-based methods, a few RGB-based methods have been proposed. Manhardt et al. [25] estimate pose and a point set from monocular images. While their method only uses RGB information during inference, they show that unannotated depth data can be used to close the synthetic-to-real domain gap in a self-supervised fashion. Going beyond point set-based shape representations, Lee et al. [26] extend Mesh R-CNN

[27] to predict pose and mesh from a single RGB image, while Engelmann et al. [28] propose shape reconstruction in a representation-agnostic way by classifying the closest matching object from a database.

Other methods such as [29, 30, 31, 32, 33] predict pose and bounding box without reconstructing the full shape of the object. For the evaluation presented in this work, we limit ourselves to methods that perform both reconstruction and pose estimation, although our evaluation protocol could in principle be used for pure pose estimation methods as well.

Other works perform shape estimation on video sequences. FroDO [34] employs DeepSDF [35] to represent the shape and uses tracked keypoints from an RGB video to optimize the latent shape description such that the keypoints lie on the 0-isosurface of the signed distance field. NodeSLAM [2] uses the depth data to optimize the shape of multiple objects and the camera pose jointly. Both of these works simplify the pose estimation problem, by assuming objects to be upright on a planar surface.

In this work, the focus is on categorical 6D pose and shape estimation methods, which do not explicitly constrain the poses. An extensive overview of existing methods is provided in Section 4.

Evaluation. The most established benchmark dataset for categorical pose estimation is the REAL275 dataset proposed by [3]. We will take a critical look at that dataset in Section 5.2.1 and show that it only evaluates a constrained set of orientations, hiding inherent difficulties of the task, such as multimodal orientation distributions. [3] also proposed average precision as a metric to evaluate pose estimation.

To evaluate shape reconstruction most papers currently use chamfer distance [36, 5, 6], which was introduced to measure the difference of point sets by [37]. However, Tatarchenko et al. [7] noted that chamfer distance is not robust to outliers, in that outlier points can skew the resulting metric based on their distance to the ground truth. Therefore, the authors advocate using a robust thresholded metric such as F-score [38] to measure the quality of reconstruction.

3. Problem Definition

Let $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ be an RGB image, $\mathbf{D} \in \mathbb{R}^{H\times W}$ be a depth map, and $\mathbf{P} \in \mathbb{R}^{3\times 4}$ be the projection matrix of the associated camera. Further, let ${}^{i}\mathbf{T}_{j}$ be the homogeneous transformation matrix, that transforms a point ${}^{j}\mathbf{p}$ from frame j to frame i, that is, ${}^{i}\mathbf{p} = {}^{i}\mathbf{T}_{j}\,{}^{j}\mathbf{p}$. Note that depending on the context, ${}^{i}\mathbf{T}_{j}$ can also be interpreted as the 6D pose of frame j in frame i. Let ${}^{i}\mathbf{R}_{j}$ and ${}^{i}\mathbf{t}_{j}$ further denote the rotation matrix and translation vector of which ${}^{i}\mathbf{T}_{j}$ is composed.

We will use o to denote the object's coordinate frame and c to denote the camera's coordinate frame. We will use $\mathcal{O}$ to denote a 3D object and $\mathcal{B}(\mathcal{O})$ to denote the axis-aligned bounding box of $\mathcal{O}$ in $\mathcal{O}$'s frame o. We define the origin of frame o as the center of $\mathcal{B}(\mathcal{O})$. We assume that 3D objects and bounding boxes are defined such that transforms can be applied to them, for example, ${}^{\mathrm{c}}\mathcal{O} = {}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}\mathcal{O}$. Following this notation, note that there is a difference between $\mathcal{B}({}^{\mathrm{c}}\mathcal{O})$ and ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}\mathcal{B}(\mathcal{O})$. The first one is an axis-aligned bounding box (AABB), the second is an oriented bounding box (OBB).

Problem 1. *(Categorical Pose and Shape Estimation) Given* $(\mathbf{I}, \mathbf{D}, \mathbf{P})$ *imaging an object* $\mathcal{O}$ *of known category* c *at pose* ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}$*, and given the mask* $\mathbf{M}$ *of visible points of the object in the image, find estimates* $\widetilde{\mathcal{O}}$ *and* ${}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}$ *of* $\mathcal{O}$ *and* ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}$*, respectively.*

Similarly, one could define the problems of categorical pose estimation (estimate ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}$ only) and categorical pose and size estimation (estimate ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}$ and $\mathcal{B}(\mathcal{O})$). Extensions to multiple images are possible by introducing a world coordinate system, but are not further considered in this work.

Note that Problem 1 only applies to object categories for which a reference frame can be defined. For example, for mugs the orientation can be defined by the opening and handle, while the position can be defined as the center of the tight bounding box. Similar definitions are possible for many human-made objects. However, objects without a well-defined category or significant intra-category variations exist. For such objects pure shape estimation such as demonstrated in [39] is more applicable.

4. Methods

An overview of proposed methods to solve Problem 1 and their most notable differences is given in Table 1. Only methods that estimate both pose and shape and do not require an initial pose estimate are included. Therefore, pure categorical pose estimation methods such as [3, 29, 33, 59] are excluded; tracking methods that require an initial pose estimate such as [60, 61, 62] are excluded; and finally, methods that explicitly rely on objects being upright such as [34, 2, 63] are excluded.

Notably, only SDFEst [23] supports multi-view setups, only iCaps [22] supports tracking over time, and only three methods include the detection part in their pipeline [25, 50, 24]. The other methods assume that an off-the-shelf detector (typically Mask R-CNN [10]) is available, but do not train it end-to-end with the pose and shape estimation part. This observation prompts us to question the predominant average precision-based evaluation protocol (see Section 5).

We note that in some circumstances modular pipelines can be more desired than end-to-end pipelines. For example, by training the detection part separately from the pose and shape estimation part, larger detection-only datasets

Table 1: Overview of categorical pose and shape estimation methods sorted by publication date. URLs to open-source implementations are included in the references. **Bold** methods are included in our toolbox and evaluation. See text for further explanation and discussion.

Method	Input	Shape	Symmetry Handling	Properties*			Open source†		Notes
				Det.	MV	Tr.	Infer.	Train	
CPS++ [25]	RGB	Point set	CD [18]	✓	✗	✗	✗	✗	Self-supervised sim-to-real (mask, depth)
Chen et al. [21]	RGB	Novel views	Input	✗	✗	✗	✓	✓	RGB-only analysis-by-synthesis
CASS [5]	RGB-D	Point set	CD [18]	✗	✗	✗	✓	✗	Variational autoencoder [13] to learn canonical shape space
SPD [6]	RGB-D	Point set	Norm. [40]	✗	✗	✗	✓	✓	Deformation of shape prior
SAR-Net [41]	D	Point set	Norm. [40]	✗	✗	✗	✓	✓	Symmetry-based shape completion
DISP6D [42]	RGB	Novel views	Codebook [43]	✗	✗	✗	✓	✓	Implicit shape and pose learning
CR-Net [14]	RGB-D	Point set	Norm. [40]	✗	✗	✗	✓	✓	Recurrent extension of SPD [6]
Lee et al. [26]	RGB	Mesh	Norm. [40]	✗	✗	✗	✗	✗	Monocular metric mesh estimation
SGPA [15]	RGB-D	Point set	Norm. [40]	✗	✗	✗	✓	✓	Transformer extension of SPD [6]
6D-ViT [44]	RGB-D	Point set	Norm. [40]	✗	✗	✗	○	○	Transformer-based feature extraction for SPD [6]
ACR-Pose [45]	RGB-D	Point set	Norm. [40]	✗	✗	✗	✗	✗	Adversarial extension of SPD [6]
ASM-Net [36]	D	Point set	CD [18]	✗	✗	✗	✓	✗	Active shape models [46]
iCaps [22]	D	Cont. SDF	Codebook [43]	✗	✗	✓	✓	✓	Alternating pose refinement and shape estimation over time
He et al. [47]	D	Mesh	CD [18]	✗	✗	✗	✗	✗	Fully self-supervised (depth)
GPV-Pose [32]	D	Point set	Sym. axis [29]	✗	✗	✗	✓	✓	Consistency-based losses and pointwise bounding box prediction
OLD-Net [48]	RGB	Point set	Norm. [40]	✓	✗	✗	✗	✗	Monocular extension of SPD [6]
Peng et al. [49]	RGB-D	Cont. SDF	Norm. [40]	✗	✗	✗	✓	✓	Self-supervised sim-to-real (depth) with DeepSDF [35]
CenterSnap [50]	RGB-D	Point set	Norm. [40]	✓	✗	✗	✓	✓	Single-stage detection, pose, and shape estimation
RePoNet [51]	RGB-D	Point set	Norm. [40] / CD [52, 53]	✗	✗	✗	✓	○	Self-supervised sim-to-real (mask)
SDFEst [23]	D	Disc. SDF	Disc. SO(3)	✗	✓	✗	✓	✓	Depth-only analysis-by-synthesis
DPDN [16]	RGB-D	Point set	Norm. [40]	✗	✗	✗	✓	✓	Self-supervised sim-to-real (consistency)
SSP-Pose [54]	D	Point set	Sym. axis [29] / Min. [3]	✗	✗	✗	✗	✗	Symmetry-aware and direct pose regression extension of SPD [6]
ShAPO [24]	RGB-D	Cont. SDF	Norm. [40]	✓	✗	✗	✓	✓	RGB-D analysis-by-synthesis
RBP-Pose [55]	D	Point set	Sym. axis only [29]	✗	✗	✗	✓	✓	Integration of SPD [6] into GPV-Pose [32]
gCasp [56]	D	Cont. SDF	Min. [3]	✗	✗	✗	✓	✓	Iterative optimization with semantic primitives [57]
Zhang et al. [58]	RGB-D	Mesh	CD [18]	✗	✗	✗	✓	✓	Fully self-supervised (RGB, depth, mask, consistency)

* Whether methods support detection (Det.), multi-view setups (MV) and tracking (Tr.) over time.

† ○ denotes cases in which the paper mentions publishing the code, but it could not be found as of July 10, 2023.

can be leveraged, whereas combined training of the detection part and the pose / shape estimation part typically requires annotations for both, which is harder to obtain at a large scale. Therefore, not supporting detection should not be seen as a major disadvantage for a given method.

Similar arguments can be made for tracking and multi-view handling. For example, most methods could be combined with a specialized tracking method such as [60] or [61] to enable tracking over time. However, most of the methods in Table 1 are purely discriminative in that they regress pose and shape in a feed-forward manner. Such methods are generally less flexible to extend to multi-view settings than methods that include a generative shape model in their pipeline, such as [5, 22, 23, 24, 56].

A notable line of work focuses on self-supervised approaches. Following the early work by Manhardt et al. [25], most works [49, 51, 16] focus on self-supervised learning in a sim-to-real context. That is, they first train in fully supervised fashion on synthetic data, and subsequently finetune without or with limited annotations on real data. He et al. [47] and Zhang et al. [58] have proposed fully self-supervised approaches that do not require initial training on synthetic data. In general, methods in this category differ in the used modality (see **Notes** column in Table 1). That is, methods formulate losses based on the depth data, mask, various consistencies that should hold, or combinations thereof. Self-supervised approaches are closely related to analysis-by-synthesis approaches [21, 23, 24], which often employ similar losses, however, to optimize the estimate at inference time instead of the network at training time.

Next, we will discuss two main dimensions in which the methods differ. First, the utilized shape representation and second, the handling of symmetries, and, in a wider sense, pose ambiguities.

4.1. Shape Representation

Various shape representations have been proposed in the literature. Shape representations vary in downstream usability (e.g., collision checking or grasp computation), efficiency, and flexibility. The latter referring to the fact that some representations can easily be converted into another, whereas the conversion for others is more involved or not well defined. Efficiency is often a trade-off with quality and depends on the employed hardware. Therefore, it is difficult to make general statements about efficiency.

Point sets. Most methods use point sets of fixed size to represent the shape. However, the exact way to predict the point sets varies. Most methods follow SPD [6] and predict an offset (often referred to as deformation) for each point in a mean shape that is defined per category. Notable exceptions are CASS [5], which instead uses a FoldingNet-based [64] variational autoencoder [13] to predict the point set, and ASM-Net [36] which regresses the parameters of a previously learned active shape model [46].

Point sets have the disadvantage of being a sparse shape representation and are therefore not a reliable representation for collision detection or grasp computation. Introducing an intermediate dense representation can alleviate this limitation; however, conversion quality is limited by the density of the point set. In our experiments, we observe that some methods predict outlier points or non-uniform point sets (i.e., uneven density). In general, high uniform density, and outlier-free point sets improve the downstream usability.

Meshes. Lee et al. [26] follow Mesh R-CNN [27] and represent the shape as a mesh. Since meshes are difficult to parameterize directly, Mesh R-CNN first predicts a discretized occupancy grid, which defines the topology of the mesh. This mesh is subsequently refined (similar to the deformation approach for point sets) through multiple refinement stages. Meshes are a dense surface representation and are hence well-suited for geometric downstream tasks.

Novel views. Instead of explicitly regressing the shape, Chen et al. [21] and DISP6D [42] employ neural networks that generate RGB and optionally depth views given a latent shape representation and viewing direction. In contrast to neural field representations (see *Continuous SDFs* below), multi-view consistency is not enforced by these methods, that is, there is no guarantee that the resulting views are consistent with each other. In principle, generated views can be converted to point sets or other representations by employing reconstruction methods on multiple synthesized views. However, this has not been demonstrated in any of the aforementioned works. Therefore, generating novel views on its own is typically not sufficient for many downstream tasks.

Discretized SDFs. SDFEst [23] represents shape as a discretized signed distance field of fixed resolution. Similar to [5] a variational autoencoder is trained to learn the shape model. Compared to point sets and novel views, signed distance fields can readily be used for collision detection and grasp computation. Furthermore, the 0-isosurface of a signed distance field can be converted to a mesh using the marching cubes algorithm [65].

Continuous SDFs. Following promising research on neural fields [66] for shape representation [35, 67, 68], iCaps [22], ShAPO [24], and gCasp [56] employ SDF-based neural fields in the context of categorical pose and shape estimation to represent the shape. Neural fields are coordinate-based neural networks (e.g., $f_\theta : \mathbb{R}^3 \mapsto d$) that for a given 3D coordinate predict a value (in this case the signed distance to the closest surface) at that coordinate. By conditioning such fields on a latent vector, they allow interpolation between shapes and mesh extraction at arbitrary resolution using the marching cubes algorithm [65].

4.2. Symmetry Handling

A major difference between existing methods is their approach to handle ambiguities. Ambiguities can occur at a category-level (e.g., bowls and bottles typically have a symmetry axis), but also due to occlusions (including view-dependent self-occlusion). In principle, such ambiguities require learning a one-to-many mapping [69]. However, most approaches assume one-to-one mappings, which creates issues at training time when different targets occur for the same or similar inputs. In general, the ideal prediction for networks that can only predict a single estimate is poorly defined, when presented with contradicting targets, and therefore they typically fail to converge to a correct pose estimate.

Different approaches to this issue have been proposed, requiring various levels of annotation and providing various degrees of flexibility. Some approaches only handle predefined category-level symmetries, whereas other approaches can, in principle, handle general ambiguities. The first being likely more data-efficient, however, only applies to constrained settings, the latter being more general. Here we summarize the different approaches and discuss their respective advantages and disadvantages.

Normalization. One way of handling symmetries at a per-category level is to introduce a constraint that removes the degree of freedom around the symmetry axis leaving only one correct orientation. This approach and an analytic solution has first been proposed for instance-level pose estimation by Pitteri et al. [40] and has subsequently been introduced to category-level pose estimation by Tian et al. [6]. Most works in the field that employ NOCS to estimate the object pose adopt the same symmetry handling. This approach has two practical disadvantages: it requires a priori specified category-level symmetries, and it cannot handle other forms of ambiguities. The former makes the assumption that all instances of a category exhibit the same axis-symmetry, which might not be the case. This could be alleviated, by per-instance symmetry annotations, which, however, requires additional annotation effort. The latter creates an issue when training on less constrained datasets. For example, when side-views of mugs are included, there can be two contradicting targets for such views. However, for view-constrained datasets with consistent per-category symmetries like the REAL275 dataset, this approach represents an efficient way of making an ambiguous dataset unimodal.

Minimum. Wang et al. [3] proposed to handle symmetries by generating multiple ground-truth targets by rotating the ground-truth pose around the predefined symmetry axis in discrete steps. The loss will then be computed for each target, and only the smallest of these losses will be used to update the weights. Compared to the normalization approach, this approach introduces a small overhead at training time. Furthermore, an ambiguous interval remains due to the discrete nature of the approach.

Symmetry axis. Following FS-Net [29] some approaches [32, 54, 55] tailor the orientation representation to the predefined axis-symmetries. Specifically, these methods parameterize the rotation by two axes (similar to [70]), which is sufficient to construct a rotation matrix (i.e., the third vector must be orthogonal to the first two). For objects with category-level symmetries, one of the axes will be defined as the symmetry axis (e.g., the up-axis of a bottle), and only that one is supervised during training and used during inference.

Chamfer distance (CD). Instead of directly supervising the pose, some methods [5, 36] employ chamfer distance (see Section 5.1.1 for definition) between transformed point sets as their loss function. Since the chamfer distance is based on the minimum distance between nearest neighbors, it automatically handles symmetries. However, chamfer distance as a loss function introduces local minima that do not exist for direct pose supervision, which can yield undesired solutions.

Codebook. Codebook-based pose estimation was introduced by Sundermeyer et al. [43] in the context of instance-based pose estimation. In short, an autoencoder is trained on synthetic data, which learns to reconstruct a given input image. Indistinguishable views will map to the same points in the latent space. A codebook is stored, which maps orientations to the corresponding latent representation. At inference time, the orientation can be inferred by finding the closest stored latent representation (and its associated orientation) to the encoded input. This approach automatically learns any form of ambiguity and has been adapted to categorical pose estimation by iCaps [22] and DISP6D [42]. iCaps uses a categorical reference object when training the decoder (i.e., implying category-level ambiguities), whereas DISP6D modifies the codebook at inference time, which in principle supports inference of object-level ambiguities.

Input. As discussed in the beginning of this section, one-to-many mappings are more difficult to learn. However, many-to-one mappings are well defined and, therefore, easier to learn. Chen et al. [21] propose a network that takes as input a latent shape description and the pose of the object and predicts the expected RGB view. To infer the pose, a large number of poses need to be sampled, and the resulting RGB image is compared with the observed one. While simplifying the learning, inferring the pose using such a network is more difficult and less efficient than with feed-forward approaches. However, similar to the codebook approach, all ambiguities are, in principle, supported, and no symmetry annotations are required.

Discretized SO(3). SDFEst [23] proposes to estimate the orientation by classifying into which cell of a discretized SO(3) grid the orientation falls. Similar to the codebook approach, this approach is, in theory, capable of handling general forms of ambiguities, since it learns all ambiguities purely from data and can represent arbitrary distributions over SO(3). A downside of this approach is that learning such a classification is likely less data-efficient compared to implicit rotation learning with a codebook as noted by [43].

4.3. Methods Evaluated

Next, we will summarize the categorical pose and shape estimation methods that are included in our evaluation. We limited ourselves to methods which are available open-source and perform both pose and shape estimation.

CASS. Canonical shape space (CASS) [5] represents one of the first methods for estimating pose and shape. It models the shape using a generative model (based on a variational autoencoder [13]) and regresses the pose, parametrized by a rotation matrix and translation. In contrast to most NOCS-based methods (see below), the pose is predicted in a correspondence-free manner, without the need for subsequent alignment with the Umeyama algorithm [11].

SPD. Shape prior deformation (SPD) [6] forms the basis of many subsequent methods. It is based on a categorical prior shape (represented as a point set) and predicts an offset (i.e., a deformation) for each point in the prior shape based on the observed image crop and masked 3D points. In addition, the method predicts an assignment matrix, assigning each observed point to a point in the prior shape. The assignment and deformation yield points in the NOCS [3] which can be used to retrieve the similarity transform (i.e., 6D pose and scale) using RANSAC with the Umeyama algorithm. Approaches following this approach are also referred to as correspondence-based, since dense correspondences between points in the observation and in a reference space (i.e., NOCS) are established.

CR-Net. Wang et al. [14] generally follow SPD [6] to represent the shape and estimate the pose, but modify the network architecture to iteratively deform the prior shape instead of predicting the full deformation in a single step.

SGPA. Structure-guided prior adaptation (SGPA) [15] also follows SPD [6] to represent the shape and estimate the pose, however, they make several architectural changes. Most notably, PointNet++ [71] is used to extract features from the prior shape and input point sets, and a transformer is used to extract features to estimate the deformation and assignment matrix.

ASM-Net. Akizuki and Hashimoto [36] propose to represent the shape using active shape models (ASM) [46]. To predict the pose, ASM-Net uses a correspondence-free estimation approach similar to CASS [5]. Compared with other works, they further use the iterative closest point (ICP) algorithm to align their estimated shape with the observed points.

iCaps. Deng et al. [22] proposes a pipeline for pose and shape estimation focused on tracking an object over time. It is the first method to use a coordinate-based neural field representation (DeepSDF [35]) in the context of categorical pose and shape estimation. The approach alternates between shape estimation given a pose and pose optimization given the shape. In this way, it iteratively optimizes the pose and shape estimates. We only evaluate iCaps' single-view performance for fair comparison.

SDFEst. Bruns and Jensfelt [23] describe a two stage pipeline for depth-based estimation of pose and shape. In the first stage the coarse pose and shape is estimated. In the second stage, these estimates are refined in a render-and-compare fashion. Similar to iCaps, SDFEst uses signed distance fields to represent the shape, however, a discretized representation is used instead of a coordinate-based neural field representation. The approach is trained using only synthetic data and only uses depth information for pose and shape estimation. Furthermore, the refinement step readily supports optimization with multiple views. However, similar to iCaps, we only evaluate the single-view performance for fair comparison.

DPDN. Deep prior deformation network (DPDN) [16] follows SGPA [15] in using PointNet++ [71] as their underlying feature extractor. Their network follows SPD [6] in predicting an assignment and deformation, however, pose estimation is learned instead of solved using the Umeyama algorithm. This allows them to directly supervise the pose, instead of just the assignment matrix. Furthermore, self-supervised learning is employed by augmenting the input point sets in two different ways, which enables the use of a consistency loss when no ground truth is available.

RBP-Pose. RBP-Pose [55] generally follows FS-Net [29] in directly regressing the pose in a correspondence-free manner, but extends it with shape reconstruction and various additional losses. In particular, the pose is estimated by two parallel branches: first, by a simple regression branch (like FS-Net [29]); second, by predicting the deformation and assignment (like SPD [29]) and subsequently the residual pointwise bounding box projections (hence RBP; see [32] for details on pointwise bounding box projection). The second part can be interpreted as an additional refinement step, based on the previously predicted pose and shape. This second branch is, however, only used for additional supervision during training; during inference, only the correspondence-free pose regression is used and optionally the shape deformation branch can be evaluated to retrieve the shape reconstruction.

5. Evaluation Protocol

In this section, the existing and proposed evaluation protocol will be discussed. Following the notation introduced in Section 3, we will start by introducing the metrics that define the evaluation protocols; subsequently the evaluation datasets are discussed.

5.1. Metrics

Various metrics exist to assess how well a method solves Problem 1. Currently, the predominant evaluation metric is *average precision* [3, 5, 14, 15, 36, 31, 29, 26] for pose estimation and *chamfer distance* [37, 5, 14, 15, 36, 26] for shape reconstruction. In this section, we will introduce these metrics and discuss several issues with them. Subsequently, we will advocate for *accuracy* (contrary to average precision) and *F-score* to evaluate pose estimation and shape reconstruction, respectively.

We first define similarity measures for transforms and objects. These are later used to define the evaluation metrics for Problem 1.

Definition 1. *Let $d({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}})$ denote the translation error between the ground-truth transform and the estimated transform, that is,*

$$d({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}) = \|{}^{\mathrm{c}}\mathbf{t}_{\mathrm{o}} - {}^{\mathrm{c}}\tilde{\mathbf{t}}_{\mathrm{o}}\|_2. \tag{1}$$

Definition 2. *Let $\delta({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}})$ denote the rotation error between the ground-truth transform and the estimated transform, that is,*

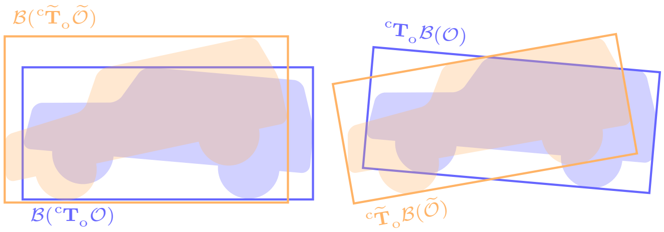
$$\delta({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}) = \left| \frac{\mathrm{trace}({}^{\mathrm{c}}\mathbf{R}_{\mathrm{o}}{}^{\mathrm{c}}\widetilde{\mathbf{R}}_{\mathrm{o}}^{-1}) - 1}{2} \right|. \tag{2}$$

Definition 3. *Let $\mathrm{IoU}(\mathcal{B}_1, \mathcal{B}_2)$ denote the true intersection over union (IoU) of two oriented bounding boxes [72]. Further, let the axis-aligned IoU between two objects be defined by*

$$\mathrm{IoU}^{+}({}^{\mathrm{c}}\mathcal{O}, {}^{\mathrm{c}}\widetilde{\mathcal{O}}) = \mathrm{IoU}\left(\mathcal{B}({}^{\mathrm{c}}\mathcal{O}), \mathcal{B}({}^{\mathrm{c}}\widetilde{\mathcal{O}})\right), \tag{3}$$

and the true (oriented) IoU using oriented bounding boxes by

$$\mathrm{IoU}({}^{\mathrm{c}}\mathcal{O}, {}^{\mathrm{c}}\widetilde{\mathcal{O}}) = \mathrm{IoU}\left({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}\mathcal{B}(\mathcal{O}), {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}\mathcal{B}(\widetilde{\mathcal{O}})\right). \tag{4}$$


$\mathcal{B}({}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}\widetilde{\mathcal{O}})$
$\mathcal{B}({}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}\mathcal{O})$
${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}\mathcal{B}(\mathcal{O})$
${}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}\mathcal{B}(\widetilde{\mathcal{O}})$

(a) Axis-aligned IoU

(b) Oriented IoU

Figure 1: Visualization of axis-aligned IoU and true (oriented) IoU in 2D. Note that axis-aligned IoU is less accurate and depends on the object shape.

The difference between these two IoU definitions is visualized in Figure 1. The current evaluation protocol [3] uses axis-aligned IoU$^+$ instead of oriented IoU, although the former is less accurate. Our implementation follows [72] and computes oriented IoU.

5.1.1. Chamfer Distance

In the context of shape reconstruction, *chamfer distance* was introduced by [37] to differentiably measure the difference of point sets.

Definition 4. *Let $\mathcal{S} \subset \mathbb{R}^3$ and $\widetilde{\mathcal{S}} \subset \mathbb{R}^3$ denote point sets sampled from the surfaces of $\mathcal{O}$ and $\widetilde{\mathcal{O}}$, respectively. We define chamfer distance as*

$$\mathrm{CD}(\mathcal{S}, \widetilde{\mathcal{S}}) = \frac{1}{2}\mathrm{AD}(\mathcal{S} \to \widetilde{\mathcal{S}}) + \frac{1}{2}\mathrm{AD}(\widetilde{\mathcal{S}} \to \mathcal{S}) \tag{5}$$

based on the directed average distance

$$\mathrm{AD}(\mathcal{X} \to \mathcal{Y}) = \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|_2. \tag{6}$$

It is easiest to interpret as the mean Euclidean distance from a point in one point set to the closest point in the other set. Note that slightly different chamfer distance versions exist, such as squared versions and ones using the sum instead of the arithmetic mean.

In Figure 2(b) we visualize potential issues with chamfer distance as an evaluation metric. Consider the two mugs with different handles (denoted 1 and 2) as reconstructions of the mug without handle (denoted GT). The relative quality of these reconstructions measured by chamfer distance varies significantly depending on the number of samples. Furthermore, a large number of samples is required for chamfer distance to converge.

Notably, the number of ground-truth samples is left unspecified by most methods. This problem becomes further amplified because methods perform reconstruction by predicting a varying number of samples as noted by [36]. Furthermore, Tatarchenko et al. [7] note that chamfer distance is not robust to outliers, since it is an average of distances (i.e., a single outlier can strongly influence the metric). Therefore, we discourage further use of chamfer distance for evaluation purposes.

Manhardt et al. [25] introduce a size-normalized reconstruction metric, closely related to chamfer distance[2].

Definition 5. *Let $\mathcal{S} \subset \mathbb{R}^3$ and $\widetilde{\mathcal{S}} \subset \mathbb{R}^3$ denote point sets sampled from the surfaces of $\mathcal{O}$ and $\widetilde{\mathcal{O}}$, respectively. We define normalized average distance (NAD) as*

$$\mathrm{NAD}(\mathcal{S}, \widetilde{\mathcal{S}}) = \max\left(\frac{\mathrm{AD}(\mathcal{S} \to \widetilde{\mathcal{S}})}{\mathrm{diam}(\mathcal{S})}, \frac{\mathrm{AD}(\widetilde{\mathcal{S}} \to \mathcal{S})}{\mathrm{diam}(\widetilde{\mathcal{S}})}\right) \tag{7}$$

based on the directed average distance (6) *and the diameter of the point sets*

$$\mathrm{diam}(\mathcal{X}) = \max_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|_2. \tag{8}$$

In general, this metric exhibits similar pitfalls as chamfer distance, however, normalizing metrics (or, equivalently, setting thresholds) based on the diameter can be a useful tool for datasets with large size differences between objects.

5.1.2. Reconstruction F-Score

Tatarchenko et al. [7] introduced F-score in the context of reconstruction as a robust alternative to chamfer distance.

Definition 6. *Let $\mathcal{S} \subset \mathbb{R}^3$ and $\widetilde{\mathcal{S}} \subset \mathbb{R}^3$ denote point sets sampled from the surfaces of $\mathcal{O}$ and $\widetilde{\mathcal{O}}$, respectively. Given an application-specific threshold Δ, we define reconstruction recall as*

$$r_\Delta(\mathcal{S}, \widetilde{\mathcal{S}}) = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}} \left[\min_{\mathbf{y}\in\widetilde{\mathcal{S}}} \|\mathbf{x} - \mathbf{y}\|_2 < \Delta\right] \tag{9}$$

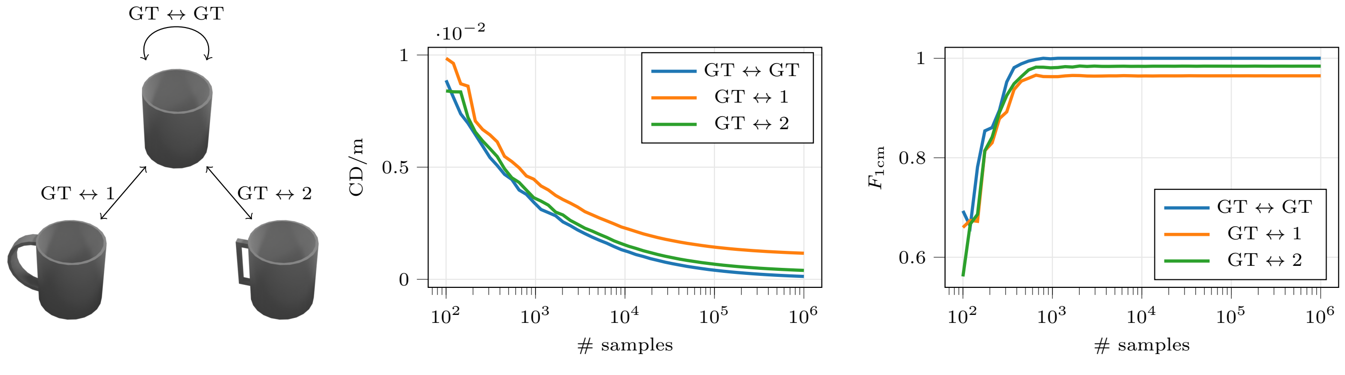
and reconstruction precision as

$$p_\Delta(\mathcal{S}, \widetilde{\mathcal{S}}) = \frac{1}{|\widetilde{\mathcal{S}}|}\sum_{\mathbf{y}\in\widetilde{\mathcal{S}}} \left[\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{x} - \mathbf{y}\|_2 < \Delta\right], \tag{10}$$

where $[\cdot]$ denotes the Iverson bracket. Finally, we define F-score as the harmonic mean of precision and recall

$$F_\Delta = \frac{2}{p_\Delta(\mathcal{S}, \widetilde{\mathcal{S}}^{-1}) + r_\Delta(\mathcal{S}, \widetilde{\mathcal{S}})^{-1}}. \tag{11}$$

[2] Manhardt et al. [25] introduced this metric as *average distance of predicted point sets* in an already thresholded manner. We have changed it to a non-thresholded metric, which can be thresholded later, yielding the same result as the originally proposed metric. This fits better into our formalism.



(a) Meshes (b) Chamfer distance (c) Reconstruction F-score

Figure 2: Visualization of the effect of varying number of samples on the chamfer distance (b) and reconstruction F-score (c). We consider two reconstructions (denoted by 1 and 2) of the ground truth (denoted by GT). Note that particularly the relative difference between $\mathrm{CD}(\mathcal{S}_{\mathrm{GT}}, \mathcal{S}_1)$ and $\mathrm{CD}(\mathcal{S}_{\mathrm{GT}}, \mathcal{S}_2)$ varies significantly. This is because the majority of the error stems from sparse sampling, not from actual differences in geometry. All mugs have been scaled to be 10 cm tall.

Note that Δ should be adjusted depending on the application and the sensor. For the table-top items contained in the datasets, we propose to use $\Delta = 1\,\mathrm{cm}$.

In Figure 2(c) we show $F_{1\mathrm{cm}}$ for varying numbers of samples for the meshes in Figure 2(a). $F_{1\mathrm{cm}}$ converges significantly faster than chamfer distance and can easily be interpreted as the percentage of correct (i.e., error below $\Delta = 1\,\mathrm{cm}$) surfaces or points [7].

Due to these reasons, following [7], we advocate for using F-score instead of chamfer distance to evaluate shape reconstruction. Furthermore, note that shape reconstruction can be evaluated in the object frame or in the camera frame, taking into account ${}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}$ and ${}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}$. Previous works evaluated in the object frame (i.e., assuming perfect alignment based on the canonical reference frame); however, we believe it is better to evaluate posed reconstruction in the camera frame, as it correlates more directly with downstream usability of the full estimate.

All metrics so far (i.e., d, δ, IoU, CD, NAD, and F_Δ) assess the quality of a single estimate. Next, we discuss average precision and accuracy, which attempt to summarize a method's performance on a dataset. In principle, one could also compute different averages of the aforementioned metrics, but those are typically affected by outliers and harder to interpret in comparison to thresholded evaluation metrics that classify estimates as correct or incorrect.

5.1.3. Average Precision

Average precision (AP) summarizes precision-recall curves in a single value [73] and has been the standard evaluation metric for object detection on the PASCAL VOC [74] and COCO datasets [75]. In general, average precision is calculated based on the interpolated precision-recall curve, which is constructed by varying a confidence threshold.

Wang et al. [3] proposed to use AP with different thresholds on IoU^+, d, and δ to evaluate their pose estimation (all specified thresholds must hold for a prediction to count as a true positive). Their method includes a Mask R-CNN architecture to detect objects and therefore had a confidence threshold to compute AP. However, most of the following pose and shape estimation methods do not include such a confidence threshold[3]. Instead they assume $\mathbf{M}$ to be given as stated in Problem 1.

To still follow the same evaluation protocol as [3], all other methods rely on the same, suboptimal Mask R-CNN predictions that [3] provided. This protocol effectively limits the achievable AP due to wrong classifications, missing detections and poor masks. Furthermore, AP is inherently difficult to interpret compared to simpler metrics.

Therefore, we believe that AP is ill-suited to compare pose and shape estimation methods that do not include detection. Rather, simpler metrics, such as accuracy (see below), should be used, assuming that mask $\mathbf{M}$ and category c are provided.

5.1.4. Proposed Metric: Accuracy

We propose to use accuracy, contrary to average precision, to assess categorical pose and shape estimation.

Definition 7. *Given inputs* $(\mathbf{I}^i, \mathbf{D}^i, \mathbf{P}^i, \mathbf{M}^i, c^i)$, *ground truths* $(\mathcal{O}^i, {}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}^i)$, *and associated predictions* $(\widetilde{\mathcal{O}}^i, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}^i)$ *with* $i = 1, \ldots, N$, *let accuracy be defined as*

$$\mathrm{ACC} = \frac{\sum_{i=1}^{N} \left[\mathrm{c}(\mathcal{O}^i, {}^{\mathrm{c}}\mathbf{T}_{\mathrm{o}}^i, \widetilde{\mathcal{O}}^i, {}^{\mathrm{c}}\widetilde{\mathbf{T}}_{\mathrm{o}}^i)\right]}{N}, \tag{12}$$

where $[\cdot]$ *denotes the Iverson bracket and* c *determines whether a prediction is correct or not based on a single or multiple thresholds on* d, δ, IoU, CD, NAD, *and* F_Δ.

Accuracy measures the percentage of correct estimates based on thresholds on translation error, rotation error, IoU, NAD and F-score. Note that we can use this simpler metric instead of average precision because we decouple pose estimation from detection and classification.

[3] Only methods marked **Det.** in Table 1 include a confidence threshold.

Table 2: Overview of categorical pose and shape estimation datasets. All datasets include RGB-D data and categorical pose ground truth. **Bold** datasets are included in our toolbox and evaluation. See text for further explanation and discussion.

Dataset	**# Instances**	**# Sequences**	**# Categories**	**Meshes**	**Real**	**Train**	**Notes**
CAMERA25 [3]	1085	31*	6	✓	✗	✓	Synthetic objects on table
REAL275 [3]	36	13	6	✓	✓	✓	Sequences of multiple objects on table
TOD [76]	15	600	3	✓	✓	✓	Transparent objects on plane
PhoCaL [77]	60	24	8	✓	✓	✓	Robotic ground-truth annotations; polarization data; transparent and reflective objects
Wild6D [51]	162	486	5	✗	✓	○†	Dataset for self-supervised learning
REDWOOD75	15	15	3	✓	✓	✗	Annotations for Redwood dataset [4]; objects freely rotated in hands
HouseCat6D [78]	194	41	10	✓	✓	✓	Larger dataset following PhoCaL [77]; better viewpoint coverage

* 31 different background scenes, 300K different object arrangements

† Statistics for evaluation split with ground-truth poses; larger training split only has mask annotations

5.1.5. Run Time

The run time of a method is the time it requires to infer the pose and shape for one object. In addition to its mean, the minimum, maximum and variance or standard deviation might also be reported, which can be meaningful when methods exhibit large variances in run time.

It is important to note that the run time can vary for different levels of optimization, and it is therefore strictly speaking only an evaluation metric of a method's implementation.

Furthermore, run time depends heavily on the used hardware. Since our toolbox aims to make the baseline results easily reproducible, it also simplifies reporting run time comparisons on equal hardware.

5.1.6. Summary: Proposed Evaluation Protocol

We propose to evaluate categorical pose and shape estimation methods by calculating accuracy at a few informative thresholds of d, δ and F_Δ. Furthermore, this evaluation procedure can be adjusted for categorical pose estimation by using only d and δ and for categorical pose and size estimation by using IoU instead of F_Δ. The thresholds for these metrics must be adjusted based on application requirements, sensor accuracy, and annotation quality. Furthermore, when combining multiple thresholds, care should be taken that they are roughly equally strict.

Note that when evaluating δ and IoU, extra care must be taken with respect to the categories containing symmetric objects. We follow [3] and ignore rotations around the up-axis for the bottle, bowl, and can categories. Further issues with ambiguities are discussed in Section 7.

In addition, reporting the run time is encouraged taking into account the caveats mentioned in Section 5.1.5.

5.2. Datasets

An overview of existing datasets is given in Table 2. So far, most methods have been evaluated on the synthetic CAMERA25 dataset and on the smaller real-world dataset REAL275 [3]. The transparent object dataset (TOD) [76] focuses on transparent objects and provides the real erroneous depth data and depth data captured using opaque twins of the transparent objects. The PhoCaL dataset [77] is a larger, highly accurate dataset. Similar to the REAL275 dataset, most objects are upright on a table (see Section 5.2.1). However, it contains significantly more occlusions, and some objects in unconstrained poses. The recently released HouseCat6D [78] is a follow-up dataset, that significantly improves with regards to limited viewpoints. Finally, the Wild6D dataset [51] is a large-scale dataset for self-supervised learning. However, it does not contain ground-truth shapes, and is therefore not suited to assess shape estimation methods.

Given its popularity in the literature, we evaluate on the REAL275 dataset (see Section 5.2.1) and additionally on the REDWOOD75 dataset (see Section 5.2.2). All of REDWOOD75's categories are included in the REAL275 dataset, which simplifies evaluation across datasets. While not included in this evaluation, we note that PhoCaL, TOD, and especially HouseCat6D are promising datasets for future research on categorical pose and shape estimation.

5.2.1. REAL275

The REAL dataset was proposed by Wang et al. [3] and consists of 4300 training images (7 video sequences) and 2750 test images (6 video sequences). An additional validation set containing 950 images is mentioned in the publication, but has not been made publicly available. The dataset contains 6 categories (bottle, bowl, camera, can, laptop, and mug) and contains 4 to 7 objects per scene. Meshes for each object are provided, obtained using an RGB-D reconstruction algorithm. Since we are primarily interested in evaluation, we will focus on the evaluation split, called REAL275, from here.

Figure 3 shows point sets with their corresponding ground-truth annotations. Note that all objects are up-

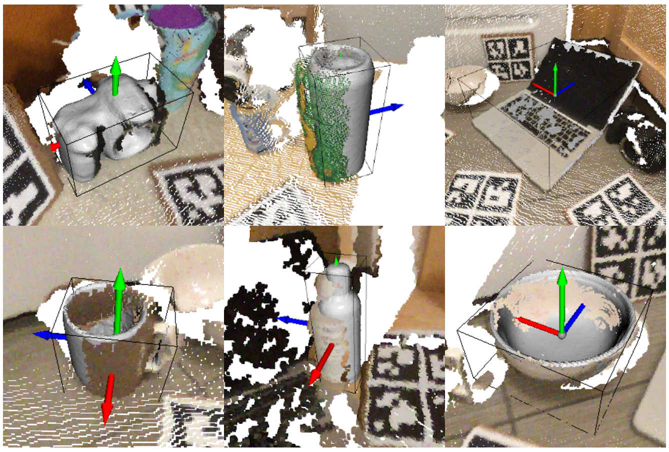


Figure 3: Examples of REAL275 samples for the 6 object categories. Note that all objects are positioned upright on a table.

right on planar surfaces. Similar constrained orientations can be found in the training splits of the CAMERA and REAL datasets. Figure 5(a) visualizes the distribution of orientations contained in the REAL275 dataset. Note that such constraints, present in training and test data, can significantly simplify the learning problem as pose and shape ambiguities disappear (e.g., upright or upside-down can).

The REAL275 dataset was originally proposed for pose estimation and was only later used to evaluate shape estimates. However, the ground-truth meshes included in the dataset are not complete. In particular, the whole bottom is missing (Figure 4, red meshes at bottom), which can cause correct estimates to be considered wrong. To fix this, we manually completed the missing surfaces in Blender [79] and include the fixed meshes in our toolbox.

5.2.2. Proposed Dataset: Redwood

To evaluate methods on less constrained orientations, we contribute annotations for a set of images in the Redwood dataset [4]. The Redwood dataset contains sequences of handheld objects being freely rotated in front of the camera. No ground-truth reconstructions are provided for these sequences.

We annotated pose and shape for 3 categories (bottle, bowl, mug) for 5 sequences each. These annotations were created by manually creating OBBs in multiple frames and exploiting potential symmetries of the object. Alignment of OBBs with previous annotations was sped up and refined by using the ICP algorithm. For each of the annotated sequences we took a subset of 5 frames covering various orientations. We will refer to this set of annotations as REDWOOD75.

To reconstruct the shape from the partially occluded and noisy depth data, we start from a dense voxel grid inside the bounding box and apply voxel carving using the annotated frames to remove hands and other temporary occlusions. The remaining voxel grid contains only voxels that are not observed to be free in any of the annotated frames. From this voxel grid, we extract a mesh and apply Laplacian smoothing. Figure 6 visualizes the annotation process.

Note that this method only approximates the real shape and is sensitive to misaligned bounding boxes and missing depth data. Especially thin surfaces and details, such as mug handles, are difficult to extract accurately due to sensor noise. Additionally, alignment errors can easily accumulate, resulting in too large or too small objects. However, the annotations are accurate enough to evaluate the performance of current methods on unconstrained orientations.

To produce the final ground truth, we compute tight bounding boxes based on the extracted meshes. For both datasets we normalize the orientation convention to be consistent with the ShapeNet [9] dataset. Figure 7 shows examples of the final annotations. In Figure 5 we compare the orientation distribution of REDWOOD75 and REAL275. Note that the orientations in REDWOOD75 are significantly less constrained than in REAL275.

6. Experiments

We follow our proposed evaluation protocol and compare the nine methods described in Section 4.3. All of the methods estimate 6D pose and reconstruct the shape. Most methods represent the shape as point sets; for iCaps [22] and SDFEst [23] we convert the estimated signed distance field first into a mesh and then into a point set by sampling 10000 points on the mesh surface.

For all methods, we closely followed the published inference code and verified that our method interface produced results similar to their evaluation code. For most methods we were able to achieve results on par with their published results. However, CASS' reconstructed point sets were significantly worse except for the laptop category and ASM-Net often predicted negative scales, which caused some reconstructions to be upside down, while the object frame o is predicted in the correct orientation.

We have implemented the metrics, and interfaces to the datasets and methods described in the previous sections using Open3D [80] and PyTorch [81]. We open-source our code as a benchmarking toolbox, with the goal of simplifying fair comparison with state-of-the-art methods. We plan to extend the toolbox as new methods are released.

6.1. Qualitative Results

Figure 8 shows randomly selected results on the REAL275 and REDWOOD75 datasets. For the REAL275 dataset, all methods perform pose estimation with a similar quality as shown in the respective publications. On REDWOOD75, on the other hand, only iCaps and SDFEst show generalization capability to arbitrary orientations. Only objects that are in a similar configuration as those in the REAL275 dataset are estimated successfully (see third row of Figure 8(b)) Most other methods predict upright objects consistent with the orientation distribution of REAL275 independent of the observed point sets.

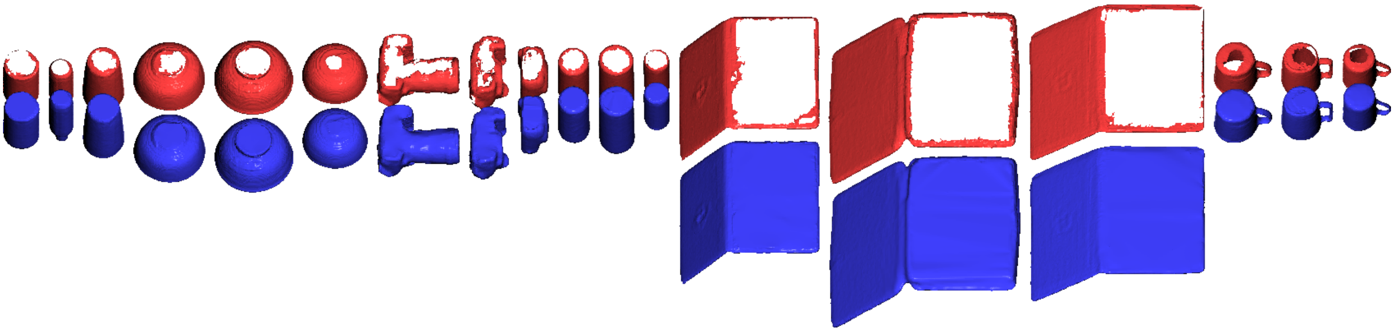
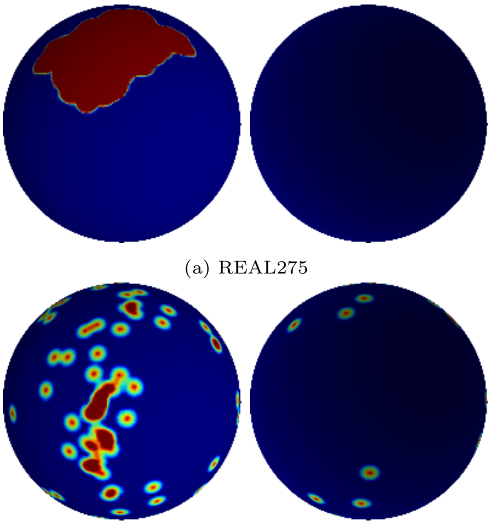


Figure 4: Bottom view of the original meshes of the REAL275 dataset (red) and our completed meshes (blue). Backfaces are not rendered highlighting the missing surfaces in the original meshes.



(a) REAL275

(b) REDWOOD75

Figure 5: Distribution of the up-axis in REAL275 and REDWOOD75 datasets. REDWOOD75 covers a larger variety of orientations.

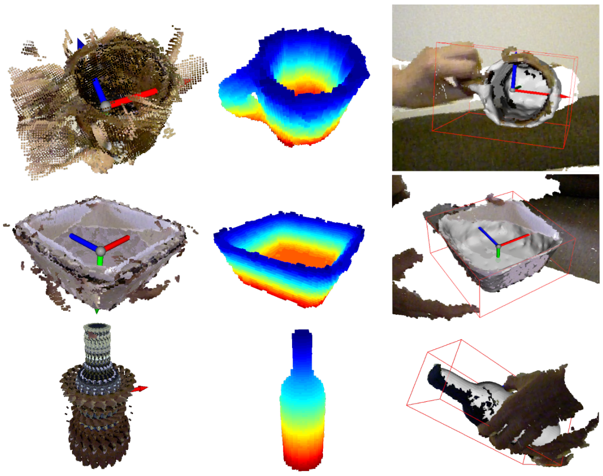


Figure 6: Manual annotations for Redwood dataset. The left column shows the cropped, accumulated point sets (including symmetries) extracted from annotated bounding boxes. The middle columns shows the voxel grid after carving. The right column shows the extracted mesh, overlaid with the point set.

A notable failure case is SDFEst's laptop prediction shown in Figure 8(a). While the predicted shape aligns well, the pose is incorrect, that is, the keyboard is aligned with the screen and vice versa. This shows a downside of unbiased depth-only approaches. Similar cases were observed by Deng et al. [22] for the can category, for which pure depth observations are not sufficient to determine whether the can is upright or not without introducing a bias into the estimation. In general, we observe more severe outliers produced by the unbiased methods, iCaps and SDFEst, on the REAL275 dataset.

ASM-Net's reconstructions are often flipped, but surfaces typically align well. As noted above, CASS completely fails to reconstruct any object except laptops. Most point set-based methods and, in particular, DPDN predict a few outlier points (outside the estimated bounding box) in their reconstruction.

6.2. Quantitative Results

We now present the results using the metrics introduced in Section 5.1. Following Section 5.1.6, we report accuracy with varying thresholds on translation error, orientation error, and reconstruction F-score in Table 3. To assess the joint estimate of pose and shape, we use $5^\circ, 1\,\text{cm}, 0.8$ and $10^\circ, 2\,\text{cm}, 0.6$ for δ, d, and $F_{1\text{cm}}$, respectively. To assess pose estimation in isolation (i.e., $F_{1\text{cm}} = \infty$), we use $5^\circ, 1\,\text{cm}$ and $10^\circ, 2\,\text{cm}$. We picked these tuples of thresholds such that all thresholds in a tuple are roughly equally strict. In the past, some methods used pairs such as 10°, 5cm, or 10°, 10cm, where only the 10° threshold practically mattered.

The results from Table 3 confirm the qualitative observations from before. DPDN performs best on REAL275, and SDFEst performs best on REDWOOD75. In general, the unbiased methods, SDFEst and iCaps, perform better on REDWOOD75 confirming limited generalization capability of methods trained on the CAMERA and REAL datasets. However, improved generalization performance comes with poorer performance on the REAL275 dataset, which opens the question, whether strong cross-dataset methods are possible. DPDN shows the third best performance on REDWOOD75, however, we note that

Table 3: Accuracy at varying position, orientation and F-score thresholds. The **best** 🟡, second best ⚪, and third best 🟤 are highlighted.

	REAL275				REDWOOD75			
$\delta, d \rightarrow$	10°, 2 cm		5°, 1 cm		10°, 2 cm		5°, 1 cm	
$F_{1\text{cm}} \rightarrow$	∞	0.6	∞	0.8	∞	0.6	∞	0.8
CASS [5]	0.3319	0.0293	0.0723	0.0000	0.0133	0.0000	0.0000	0.0000
SPD [6]	0.5439	0.4818	0.2082	0.1741	0.1867	0.1600	0.0400	0.0267
CR-Net [14]	0.5993	0.4781	0.2397	0.1777	0.2533	0.2400 🟤	0.0800	0.0800
SGPA [15]	0.6939 🟤	0.6302 ⚪	0.3142 ⚪	0.2497 ⚪	0.2267	0.2267	0.2000 ⚪	0.1733 ⚪
ASM-Net [36]	0.3255	0.2101	0.0678	0.0508	0.3067	0.1733	0.0667	0.0533
iCaps [22]	0.2976	0.2243	0.0565	0.0379	0.4133 ⚪	0.3467 ⚪	0.1200 🟤	0.0800
SDFEst [23]	0.5060	0.4710	0.2237	0.2008 🟤	**0.6533** 🟡	**0.6400** 🟡	**0.4667** 🟡	**0.4133** 🟡
DPDN [16]	**0.7589** 🟡	**0.6315** 🟡	**0.3336** 🟡	**0.2591** 🟡	0.3333 🟤	0.3200 🟤	0.0933	0.0933 🟤
RBP-Pose [55]	0.7397 ⚪	0.5559 🟤	0.2847 🟤	0.1514	0.2133	0.0933	0.0667	0.0267

Table 4: Accuracy for different categories with $\delta = 10°, d = 2\,\text{cm}, F_{1\text{cm}} = 0.6$. The **best** 🟡, second best ⚪, and third best 🟤 are highlighted.

	REAL275						REDWOOD75		
	Bottle	Bowl	Camera	Can	Laptop	Mug	Bottle	Bowl	Mug
CASS [5]	0.0018	0.0920	0.0000	0.0264	0.0000	0.0623	0.0000	0.0000	0.0000
SPD [6]	0.6268 🟤	0.8987	0.0543	**0.8560** 🟡	0.2269	0.2825	0.3200 🟤	0.0800	0.0800
CR-Net [14]	0.5963	0.9131	0.0498	0.6021	0.3936 ⚪	0.3656	0.2400	0.4000 🟤	0.0800
SGPA [15]	**0.7570** 🟡	0.9481 ⚪	0.0742 🟤	0.8534 ⚪	**0.4383** 🟡	0.7362	0.2000	0.4000 🟤	0.0800
ASM-Net [36]	0.1596	0.1114	0.0229	0.5808	0.1365	0.2328	0.1600	0.3600	0.0000
iCaps [22]	0.2136	0.5477	0.0214	0.2257	0.1187	0.2568	0.1600	0.7600 ⚪	0.1200 🟤
SDFEst [23]	0.3935	0.8679	0.0011	0.4984	0.1946	**0.8986** 🟡	**0.7200** 🟡	**0.8400** 🟡	**0.3600** 🟡
DPDN [16]	0.7004 ⚪	**0.9582** 🟡	0.1562 ⚪	0.8153 🟤	0.3137 🟤	0.8704 ⚪	0.4000 ⚪	0.4000 🟤	0.1600 ⚪
RBP-Pose [55]	0.4559	0.9338 🟤	**0.2023** 🟡	0.7340	0.2756	0.7707 🟤	0.0000	0.2000	0.0800

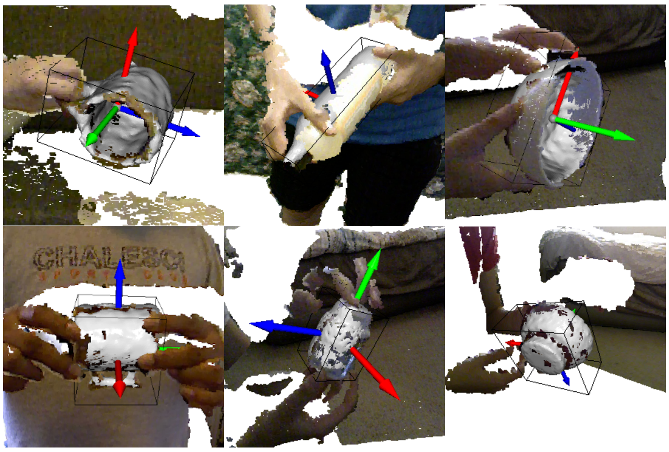


Figure 7: Examples of REDWOOD75 samples.

this mostly stems from objects with similar orientations as REAL275. RBP-Pose while giving strong results for pose estimation on REAL275, shows poor performance on REDWOOD75 indicating a strong dataset-specific bias. Furthermore, its reconstruction quality falls short of simpler methods such as SGPA.

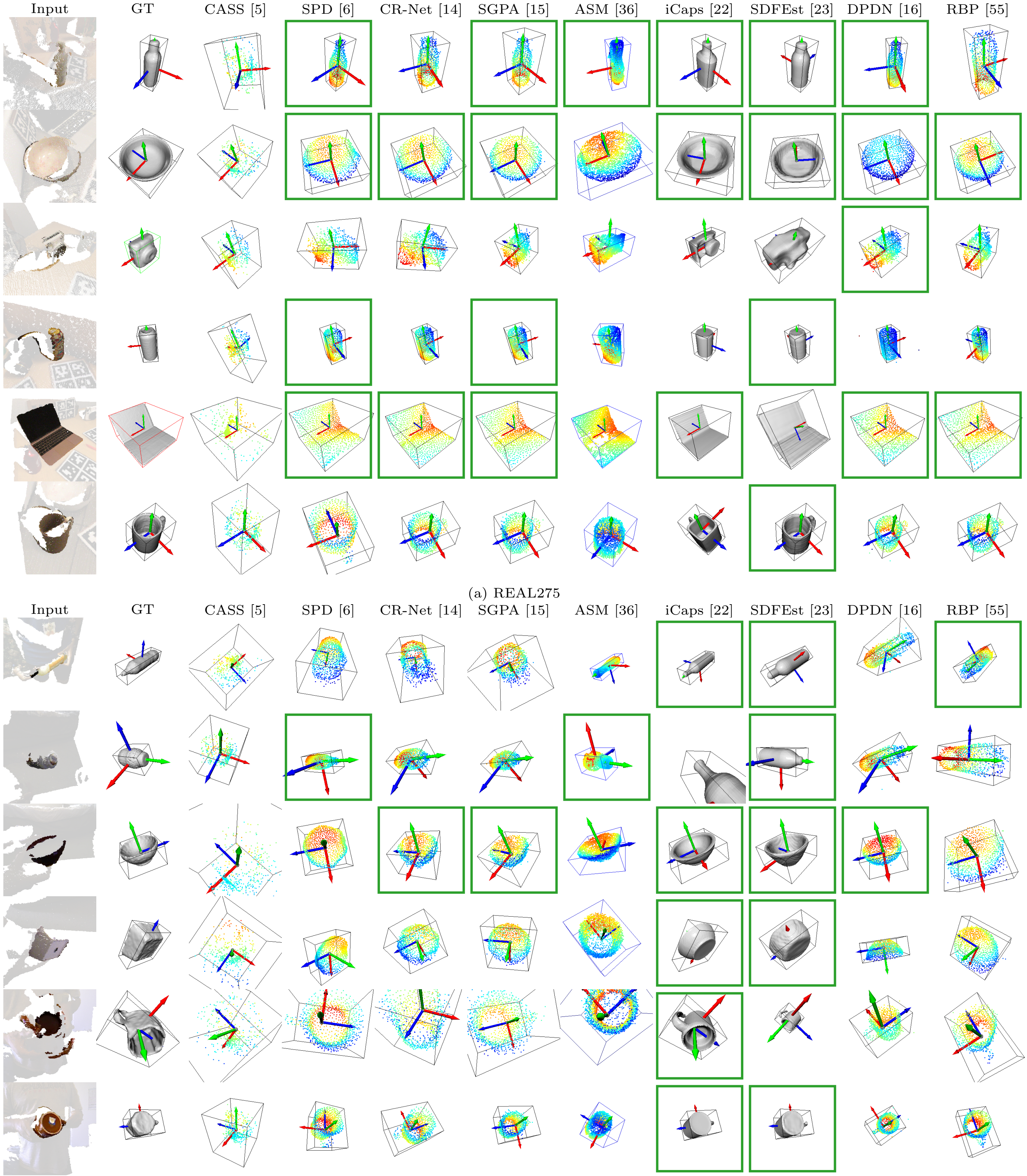
Note that for both datasets there is still a lot of room for improvement. Typically, significantly less than 50% of the estimates are of sufficient quality to be considered correct in pose and shape for the stricter thresholds. This shows that categorical pose and shape estimation is still an open problem, especially for unconstrained orientations.

To gain further insight into the estimation quality of the methods, we show detailed results for varying thresholds in Figure 10. It can be seen that the difference between the two datasets is most pronounced for orientation-based thresholding. This confirms the issue of constrained orientations discussed in Section 5.2 (see Figure 5). Unbiased methods (i.e., SDFEst and iCaps) that are trained on unconstrained synthetically generated data perform significantly better on REDWOOD75, however they level out earlier on REAL275, partly due to the aforementioned ambiguities, which other methods avoid through dataset specific biases. On REDWOOD75, ASM-Net which is trained on synthetic data (with the exact distribution unknown) performs comparatively well. Similarly DPDN, which is trained with augmented point sets, shows some generalization capability, however, less so than unbiased methods.

In Table 4 we further report the accuracy per category for the more lenient thresholds $\delta = 10°, d = 2\,\text{cm}, F_{1\text{cm}} = 0.6$. Note that CASS' reconstructions are relatively sparse and very noisy, and therefore rarely reach $F_{1\text{cm}} > 0.6$ (see also Figure 10). On REAL275, all methods fail at the camera category which contains significantly more shape variation than the other categories. RBP-Pose and DPDN show some encouraging results here, but still perform significantly worse compared to other categories. SDFEst shows a drop in performance for geometrically ambiguous categories, such as can and laptop, for which other methods perform comparatively better.

Run Time. Table 5 reports the run time results[4]. These times include the transfer time from CPU to GPU and

[4] Intel Core i7-6850K, NVIDIA TITAN X (Pascal)



(a) REAL275

(b) REDWOOD75

Figure 8: Randomly selected results on REAL275 and REDWOOD75 datasets. Results that are considered correct under $\delta = 10°, d = 2\,\text{cm}, F_{1\text{cm}} = 0.6$ thresholds are $\boxed{\text{highlighted}}$. Input shows the full point set, with points outside the instance mask grayed out.

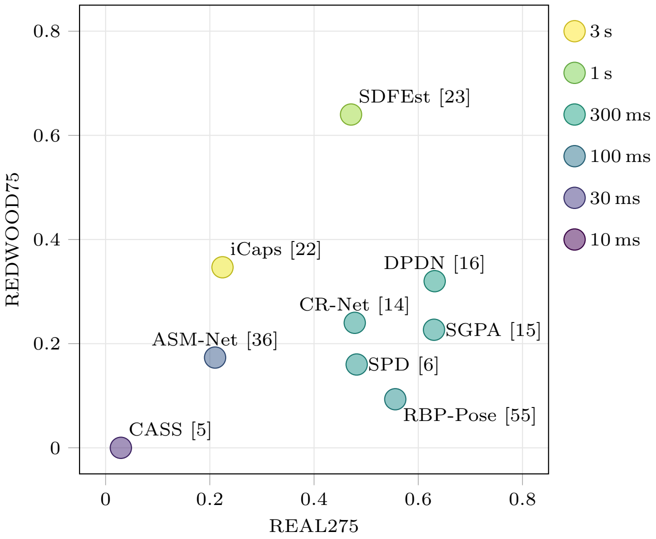


Figure 9: Accuracy of the evaluated methods on the REAL275 and REDWOOD75 dataset for $\delta = 10°, d = 2\,\text{cm}, F_{1\text{cm}} = 0.6$. The color of the circle indicates the mean run time of the method for one inference.

Table 5: Run time of the evaluated methods per object estimation on the REDWOOD75 dataset in seconds.

	Mean	Std	Min	Max
CASS [5]	0.0217	0.0247	0.0137	0.2294
SPD [6]	0.2079	0.1406	0.0976	0.9414
CR-Net [14]	0.2203	0.1454	0.1051	0.9266
SGPA [15]	0.2277	0.1432	0.1200	0.9211
ASM-Net [36]	0.0545	0.0234	0.0223	0.1436
iCaps [22]	2.5554	0.9255	2.0181	10.1082
SDFEst [23]	1.3458	0.0302	1.2588	1.4123
DPDN [16]	0.2581	0.1444	0.1338	0.8772
RBP-Pose [55]	0.1766	0.1487	0.0530	0.7680

mesh reconstruction for iCaps and SDFEst, but exclude the computation of the metrics. Most notably, the two methods including iterative optimization and mesh reconstruction (iCaps and SDFEst) exhibit the longest run times. CASS achieves the fastest run times, however also the worst results. Figure 9 further visualizes the mean run time and performance of all methods on both datasets.

7. Limitations

7.1. Comparability

The results from Section 6 suggest that training on synthetic data currently generalizes better to unconstrained orientations. This is expected, since synthetic data generation provides perfectly annotated and unconstrained training data. However, it introduces a synthetic-to-real domain gap, which needs to be addressed. This opens the question of how well methods such as DPDN and RBP-Pose would perform when trained on unconstrained synthetic images. In general, since methods currently vary significant parts of training datasets, architecture, pose parameterization, and losses, it is difficult to assess the contributions of individual changes.

7.2. Multimodal Distributions

Unconstrained pose estimation introduces significant difficulties in the task, which were hidden due to the constraints present in the CAMERA and REAL datasets. Consider, for example, the bottom left mug in Figure 7. From the given view, it is difficult to tell which way the opening of the mug faces. Another example are cans, which are geometry-wise nearly symmetric. Currently there are only few works [82, 69] that consider this problem of ambiguous poses. Evaluation of such methods that predict multimodal posteriors is difficult. One possible way of extending the presented framework to such methods would be to allow methods to generate N hypotheses. Two accuracy values ACC_{best} and $\text{ACC}_{\text{worst}}$ could be computed based on the best and worst hypothesis, respectively. A strong method would generate the same hypothesis N times if there is no ambiguity, leading to a higher ACC_{best} and $\text{ACC}_{\text{worst}}$. Similarly, if there is ambiguity, the correct hypothesis would still be contained in the set of hypotheses leading to a higher ACC_{best}.

7.3. Dataset Size

The REDWOOD75 dataset is limited in size, but the results suggest a clear lack of generalization capability of current approaches. This shows the need for larger datasets for unconstrained pose and shape estimation. It is an open question how such a dataset could be collected in the most efficient way.

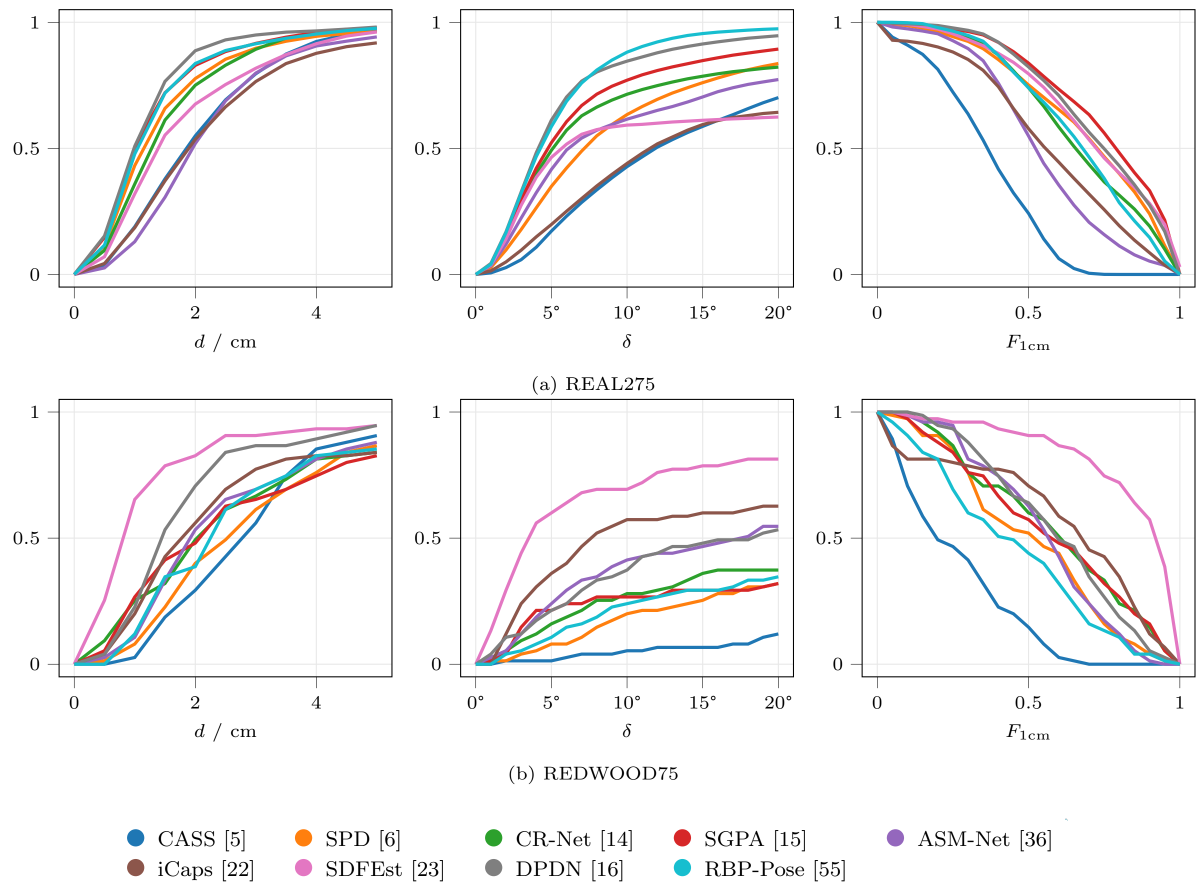
8. Conclusion and Outlook

In this work, we have discussed the current state of categorical pose and shape estimation and identified several limitations of the current evaluation protocol. In particular, existing evaluation datasets contain only a heavily constrained set of orientations, which simplifies the problem by removing pose and shape ambiguities. Furthermore, existing evaluation metrics are suboptimal and unnecessarily difficult to interpret. To alleviate these problems, we propose a new set of metrics and new annotations for the REDWOOD dataset, which contains less constrained orientations. We apply our evaluation protocol to nine state-of-the-art methods and confirm limited generalization capability as suggested by the constrained orientations in their training data.

Our experiments suggest that there is a need for larger, high-quality datasets for unconstrained pose and shape estimation as well as for robust methods that can handle unconstrained orientations and the resulting pose ambiguities in a principled way.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.



(a) REAL275

(b) REDWOOD75

Figure 10: Detailed accuracy results for varying thresholds of position, orientation and F-score thresholds.

References

[1] J. Varley, C. DeChant, A. Richardson, J. Ruales, P. Allen, Shape completion enabled robotic grasping, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, pp. 2442–2447.

[2] E. Sucar, K. Wada, A. Davison, NodeSLAM: Neural object descriptors for multi-view shape reconstruction, in: Proceedings of the International Conference on 3D Vision, 2020, pp. 949–958.

[3] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, L. J. Guibas, Normalized object coordinate space for category-level 6D object pose and size estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2642–2651.

[4] S. Choi, Q.-Y. Zhou, S. Miller, V. Koltun, A large dataset of object scans, arXiv preprint arXiv:1602.02481 (2016).

[5] D. Chen, J. Li, Z. Wang, K. Xu, Learning canonical shape space for category-level 6D object pose and size estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11973–11982. URL: https://github.com/densechen/CASS.

[6] M. Tian, M. H. Ang, G. H. Lee, Shape prior deformation for categorical 6D object pose and size estimation, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 530–546. URL: https://github.com/mentian/object-deformnet.

[7] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, T. Brox, What do single-view 3D reconstruction networks learn?, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3405–3414.

[8] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, J. Matas, BOP challenge 2020 on 6D object localization, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 577–594.

[9] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, ShapeNet: An Information-Rich 3D Model Repository, Technical Report 1512.03012, arXiv preprint, 2015.

[10] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[11] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 376–380.

[12] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (1981) 381–395.

[13] D. P. Kingma, M. Welling, Auto-encoding variational Bayes, in: Proceedings of the International Conference on Learning Representations, 2014.

[14] J. Wang, K. Chen, Q. Dou, Category-level 6D object pose estimation via cascaded relation and recurrent reconstruction networks, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2021, pp. 4807–4814. URL: https://github.com/JeremyWANGJZ/Category-6D-Pose.

[15] K. Chen, Q. Dou, SGPA: Structure-guided prior adaptation for category-level 6D object pose estimation, in: Proceedings of the International Conference on Computer Vision, 2021, pp. 2773–2782. URL: https://github.com/ck-kai/SGPA.

[16] J. Lin, Z. Wei, C. Ding, K. Jia, Category-level 6D object pose and size estimation using self-supervised deep prior deformation networks, in: Proceedings of the European Con-

ference on Computer Vision, Springer, 2022, pp. 19–34. URL: https://github.com/JiehongLin/Self-DPDN.

[17] Y. Li, G. Wang, X. Ji, Y. Xiang, D. Fox, DeepIM: Deep iterative matching for 6D pose estimation, International Journal of Computer Vision 128 (2020) 657–678.

[18] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, S. Savarese, DenseFusion: 6D object pose estimation by iterative dense fusion, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3343–3352.

[19] Y. Labbé, J. Carpentier, M. Aubry, J. Sivic, CosyPose: Consistent multi-view multi-object 6D pose estimation, in: European Conference on Computer Vision, Springer, 2020, pp. 574–591.

[20] I. Shugurov, I. Pavlov, S. Zakharov, S. Ilic, Multi-view object pose refinement with differentiable renderer, IEEE Robotics and Automation Letters 6 (2021) 2579–2586.

[21] X. Chen, Z. Dong, J. Song, A. Geiger, O. Hilliges, Category level object pose estimation via neural analysis-by-synthesis, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 139–156. URL: https://github.com/xuchen-ethz/neural_object_fitting.

[22] X. Deng, J. Geng, T. Bretl, Y. Xiang, D. Fox, icaps: Iterative category-level object pose and shape estimation, IEEE Robotics and Automation Letters 7 (2022) 1784–1791. URL: https://github.com/aerogjy/iCaps.

[23] L. Bruns, P. Jensfelt, SDFEst: Categorical pose and shape estimation of objects from RGB-D using signed distance fields, IEEE Robotics and Automation Letters 7 (2022) 9597–9604. URL: https://github.com/roym899/sdfest.

[24] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, A. Gaidon, ShAPO: Implicit representations for multi-object shape, appearance, and pose optimization, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 275–292. URL: https://github.com/zubair-irshad/shapo.

[25] F. Manhardt, G. Wang, B. Busam, M. Nickel, S. Meier, L. Minciullo, X. Ji, N. Navab, CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning, arXiv preprint arXiv:2003.05848 (2020).

[26] T. Lee, B.-U. Lee, M. Kim, I. S. Kweon, Category-level metric scale object shape and pose estimation, IEEE Robotics and Automation Letters 6 (2021) 8575–8582.

[27] G. Gkioxari, J. Malik, J. Johnson, Mesh R-CNN, in: Proceedings of the International Conference on Computer Vision, 2019, pp. 9785–9795.

[28] F. Engelmann, K. Rematas, B. Leibe, V. Ferrari, From points to multi-object 3D reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4588–4597.

[29] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, A. Leonardis, FS-Net: Fast shape-based network for category-level 6D object pose estimation with decoupled rotation mechanism, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1581–1590.

[30] X. Li, Y. Weng, L. Yi, L. J. Guibas, A. Abbott, S. Song, H. Wang, Leveraging SE(3) equivariance for self-supervised category-level object pose estimation from point clouds, Advances in Neural Information Processing Systems 34 (2021) 15370–15381.

[31] J. Lin, Z. Wei, Z. Li, S. Xu, K. Jia, Y. Li, DualPoseNet: Category-level 6D object pose and size estimation using dual pose network with refined learning of pose consistency, in: Proceedings of the International Conference on Computer Vision, 2021, pp. 3560–3569.

[32] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, F. Tombari, GPV-Pose: Category-level object pose estimation via geometry-guided point-wise voting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6781–6791. URL: https://github.com/lolrudy/GPV_Pose.

[33] T. Lee, B.-U. Lee, I. Shin, J. Choe, U. Shin, I. S. Kweon, K.-J. Yoon, UDA-COPE: Unsupervised domain adaptation for category-level object pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14891–14900.

[34] M. Rünz, K. Li, M. Tang, L. Ma, C. Kong, T. Schmidt, I. Reid, L. Agapito, J. Straub, S. Lovegrove, et al., FroDO: From detections to 3D objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14720–14729.

[35] J. J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: Learning continuous signed distance functions for shape representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 165–174.

[36] S. Akizuki, M. Hashimoto, ASM-Net: Category-level pose and shape estimation using parametric deformation, in: Proceedings of the British Machine Vision Conference, 2021, pp. 1–13. URL: https://github.com/sakizuki/asm-net.

[37] H. Fan, H. Su, L. J. Guibas, A point set generation network for 3D object reconstruction from a single image, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 605–613.

[38] A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: Benchmarking large-scale scene reconstruction, ACM Transactions on Graphics 36 (2017).

[39] C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, G. Gkioxari, Multiview compressive coding for 3D reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9065–9075.

[40] G. Pitteri, M. Ramamonjisoa, S. Ilic, V. Lepetit, On object symmetries and 6D pose estimation from images, in: Proceedings of the International Conference on 3D Vision (3DV), IEEE, 2019, pp. 614–622.

[41] H. Lin, Z. Liu, C. Cheang, Y. Fu, G. Guo, X. Xue, SAR-Net: Shape alignment and recovery network for category-level 6D object pose and size estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6707–6717. URL: https://github.com/hetolin/SAR-Net.

[42] Y. Wen, X. Li, H. Pan, L. Yang, Z. Wang, T. Komura, W. Wang, DISP6D: Disentangled implicit shape and pose learning for scalable 6D pose estimation, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 404–421. URL: https://github.com/fylwen/DISP-6D.

[43] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3D orientation learning for 6D object detection from RGB images, in: Proceedings of the European Conference on Computer Vision, Springer, 2018, pp. 699–715.

[44] L. Zou, Z. Huang, N. Gu, G. Wang, 6D-ViT: Category-level 6D object pose estimation via transformer-based instance representation learning, IEEE Transactions on Image Processing 31 (2022) 6907–6921. URL: https://github.com/luzzou/6D-ViT.

[45] Z. Fan, Z. Song, J. Xu, Z. Wang, K. Wu, H. Liu, J. He, ACR-Pose: Adversarial canonical representation reconstruction network for category level 6D object pose estimation, arXiv preprint arXiv:2111.10524 (2021).

[46] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, Active shape models-their training and application, Computer vision and image understanding 61 (1995) 38–59.

[47] Y. He, H. Fan, H. Huang, Q. Chen, J. Sun, Towards self-supervised category-level object pose and size estimation, arXiv preprint arXiv:2203.02884 (2022).

[48] J. Lin, Z. Wei, C. Ding, K. Jia, Object level depth reconstruction for category level 6D object pose estimation from monocular RGB image, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 220–236.

[49] W. Peng, J. Yan, H. Wen, Y. Sun, Self-supervised category-level 6D object pose estimation with deep implicit shape representation, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 2082–2090. URL: https://github.com/swords123/SSC-6D.

[50] M. Z. Irshad, T. Kollar, M. Laskey, K. Stone, Z. Kira, CenterSnap: Single-shot multi-object 3D shape reconstruction and

categorical 6D pose and size estimation, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2022, pp. 10632–10640. URL: https://github.com/zubair-irshad/CenterSnap.

[51] Y. Fu, X. Wang, Category-level 6D object pose estimation in the wild: A semi-supervised learning approach and a new dataset, Advances in Neural Information Processing Systems 35 (2022) 27469–27483. URL: https://github.com/OasisYang/Wild6D.

[52] Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes, in: Robotics: Science and Systems, 2018.

[53] G. Wang, F. Manhardt, F. Tombari, X. Ji, GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16611–16621.

[54] R. Zhang, Y. Di, F. Manhardt, F. Tombari, X. Ji, SSP-Pose: Symmetry-aware shape prior deformation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022, pp. 7452–7458.

[55] R. Zhang, Y. Di, Z. Lou, F. Manhardt, F. Tombari, X. Ji, RBP-Pose: Residual bounding box projection for category-level pose estimation, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 655–672. URL: https://github.com/lolrudy/RBP_Pose.

[56] G. Li, Y. Li, Z. Ye, Q. Zhang, T. Kong, Z. Cui, G. Zhang, Generative category-level shape and pose estimation with semantic primitives, in: Proceedings of the Conference on Robot Learning, 2022, pp. 1390–1400. URL: https://github.com/zju3dv/gCasp.

[57] Z. Hao, H. Averbuch-Elor, N. Snavely, S. Belongie, DualSDF: Semantic shape manipulation using a two-level representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7631–7641.

[58] K. Zhang, Y. Fu, S. Borse, H. Cai, F. Porikli, X. Wang, Self-supervised geometric correspondence for category-level 6D object pose estimation in the wild, in: Proceedings of the International Conference on Learning Representations, 2023. URL: https://github.com/kywind/self-corr-pose.

[59] Y. You, R. Shi, W. Wang, C. Lu, CPPF: Towards robust category-level 9D pose estimation in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6866–6875.

[60] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, Y. Zhu, 6-PACK: Category-level 6D pose tracker with anchor-based keypoints, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2020, pp. 10059–10066.

[61] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, L. J. Guibas, CAPTRA: Category-level pose tracking for rigid and articulated objects from point clouds, in: Proceedings of the International Conference on Computer Vision, 2021, pp. 13209–13218.

[62] B. Wen, K. Bekris, BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2021, pp. 8067–8074.

[63] K. Li, H. Rezatofighi, I. Reid, MOLTR: Multiple object localization, tracking and reconstruction from monocular RGB videos, IEEE Robotics and Automation Letters 6 (2021) 3341–3348.

[64] Y. Yang, C. Feng, Y. Shen, D. Tian, FoldingNet: Point cloud auto-encoder via deep grid deformation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.

[65] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, in: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, 1987, pp. 163–169.

[66] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, S. Sridhar, Neural fields in visual computing and beyond, in: Computer Graphics Forum, volume 41, Wiley Online Library, 2022, pp. 641–676.

[67] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger, Occupancy networks: Learning 3D reconstruction in function space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4460–4470.

[68] Z. Chen, H. Zhang, Learning implicit fields for generative shape modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5939–5948.

[69] H. Deng, M. Bui, N. Navab, L. Guibas, S. Ilic, T. Birdal, Deep Bingham networks: Dealing with uncertainty and ambiguity in pose estimation, International Journal of Computer Vision (2022) 1–28.

[70] Y. Zhou, C. Barnes, J. Lu, J. Yang, H. Li, On the continuity of rotation representations in neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5745–5753.

[71] C. R. Qi, L. Yi, H. Su, L. J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, Advances in Neural Information Processing Systems 30 (2017) 5099–5108.

[72] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, M. Grundmann, Objectron: A large scale dataset of object-centric videos in the wild with pose annotations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7822–7831.

[73] G. Salton, M. J. McGill, Introduction to modern information retrieval, McGraw Hill, 1983.

[74] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2010) 303–338.

[75] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 740–755.

[76] X. Liu, R. Jonschkowski, A. Angelova, K. Konolige, KeyPose: Multi-view 3D labeling and keypoint estimation for transparent objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11602–11610.

[77] P. Wang, H. Jung, Y. Li, S. Shen, R. P. Srikanth, L. Garattoni, S. Meier, N. Navab, B. Busam, PhoCaL: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 21222–21231.

[78] H. Jung, S.-C. Wu, P. Ruhkamp, H. Schieber, P. Wang, G. Rizzoli, H. Zhao, S. D. Meier, D. Roth, N. Navab, et al., Housecat6d–a large-scale multi-modal category level 6d object pose dataset with household objects in realistic scenarios, arXiv preprint arXiv:2212.10428 (2022).

[79] Blender Online Community, Blender - a 3D modelling and rendering package, Blender Foundation, Blender Institute, Amsterdam, 2022. URL: http://www.blender.org.

[80] Q.-Y. Zhou, J. Park, V. Koltun, Open3D: A modern library for 3D data processing, arXiv preprint arXiv:1801.09847 (2018).

[81] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019) 8026–8037.

[82] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, F. Tombari, Explaining the ambiguity of object detection and 6D pose from visual data, in: Proceedings of the International Conference on Computer Vision, 2019, pp. 6841–6850.