Rheinische
Friedrich-Wilhelms-
Universität Bonn

Institute for Computer Science
Department VI
Autonomous Intelligent Systems

Rheinische
Friedrich-Wilhelms-Universität Bonn

Bachelor Thesis

Learning Implicit Probability Distribution Functions for 6D Object Poses

Author:
Luis Denninger

First Examiner:
Prof. Dr. Sven Behnke

Second Examiner:
Prof. Dr. Florian Bernard

Date: January 17, 2023

Declaration

I hereby declare that I am the sole author of this thesis and that none other than the specified sources and aids have been used. Passages and figures quoted from other works have been marked with appropriate mention of the source.

Place, Date

Signature

Abstract

The 6D object pose estimation is an indispensable requirement for many robotic applications, such as autonomous manipulation or motion planning. Object symmetries create ambiguities in an object's pose, increasing the complexity of the pose estimation task. Existing approaches predict only a single pose which makes it unable to reason about object symmetries in downstream tasks. Recently, modeling object orientations implicitly as non-parametric probability distributions on the rotation manifold $\mathbf{SO}(3)$ has shown impressive capabilities to capture geometric symmetries.

This thesis proposes *ImplicitPosePDF* model which aims to model poses by extending the approach to the space of all rigid body transformation $\mathbf{SE}(3)$ to capture object symmetries. The estimation of the distributions is decoupled in two models estimating an orientation and a translation distribution respectively. Each distribution is parametrized through a neural network. Using an efficient equi-volumetric sampling strategy for the rotation manifold $\mathbf{SO}(3)$ and the translation space $\mathbb{R}^3$, the pose distribution is approximated as a histogram over the respective space. The models are trained to estimate the likelihood of a single orientation, respectively translation, hypothesis taken from a set of ground-truth poses representing the object symmetries.

Acquiring ground-truth labels, especially multiple symmetrical poses for each frame, for the pose estimation task remains a bottleneck in the training process. To produce ground-truth labels to train our model, this thesis further introduces a three-stage pipeline that generates multiple symmetrical pseudo ground-truth poses for each training image without the supervision of object poses or symmetries. Given an RGB-D image and a 3D object model, the pipeline produces the set of pseudo ground-truth poses through a two-stage point cloud registration process with a succeeding render-and-compare validation stage.

The pose labeling scheme and *ImplicitPosePDF* model are evaluated on a photorealistic dataset and the T-Less dataset. Through thorough experimental evaluations, we demonstrate the strengths of our model to capture arbitrary symmetries in realistic scenarios and highlight the advantages of the proposed pose labeling scheme.

Contents

1 Introduction **1**

2 6D Pose Estimation **4**
2.1 Object Symmetries 4
2.2 Problem Definition 6

3 Related Work **7**
3.1 6D Pose Estimation 7
3.2 Ground-Truth Acquisition 8

4 ImplicitPosePDF Model **10**
4.1 Methods 10
4.2 Training 15
4.3 Visualization 15

5 Automatic Pose Labeling Scheme **17**
5.1 Problem Definition 17
5.2 Methods 18
5.3 Pipeline 21

6 Evaluation **23**
6.1 Datasets 23
6.2 Evaluation Metrics 26
6.3 Implementation Details 27
6.4 Automatic Pose Labeling Scheme Results 29
6.5 Backbone Ablation Studies 30
6.6 6D Pose Distribution Estimation 32
6.7 Single 6D Pose Estimation 36
6.8 Effectiveness of Pseudo Ground-Truth Labels 38
6.9 T-Less Evaluation 39

7 Conclusion **41**

1 Introduction

In our daily life, we continuously estimate the relative locations and orientations of objects in our surroundings. For instance, grasping a coffee mug requires us to estimate the relative location and orientation of the mug to our hand. If the handle of the mug is not visible, the orientation estimation is not unique but rather introduces uncertainty about the orientation. Consequently, we are not only aware of the pose of an object but also of the uncertainty around it. We have a notion of symmetry exhibited by the mug and can estimate the possible orientations that hide the handle. With the increase of automatization in the everyday life, autonomous systems require similar capabilities to interact with their surroundings.

The 6D object pose estimation becomes essentially important in the field of robotics. With the goal of building autonomous agents in human environments, robots need to reason about an object's 3D location and 3D orientation for manipulation or motion planning tasks. While there are very successful approaches to generating accurate poses from RGB-D images, the pose estimation from RGB images still faces open challenges. The low costs and small sensor sizes, while conveying a broad range of information, makes RGB cameras particularly interesting for robotic applications. Visual data conveys vital information about an object's pose which is extracted in recent approaches using Convolutional Neural Networks (CNNs) as feature extractors [1] [2]. In real-world scenarios, such data is strongly impacted by external influences such as lighting conditions, occlusion, object texture or pose variations arising from symmetries. Object symmetries introduce ambiguities in the pose, meaning that the visual representation of an object can be mapped to several correct poses and vica versa. While most recent 6D pose estimation approaches focus on estimating a single correct 6D pose of symmetric objects [3] [4] [2], modeling object symmetries is still an open challenge.

Furthermore, to efficiently train such CNN models through deep learning, large-scale datasets of ground-truth annotated images are required. Acquiring ground-truth labels for large datasets still remains the bottleneck in the training process. This problem is further aggravated when dataset labels should include multiple poses representing the symmetries of arbitrary objects, especially without prior knowledge about the symmetries.

Murphy *et al.* [5] proposed the Implicit Probability Distribution Function (Implicit-

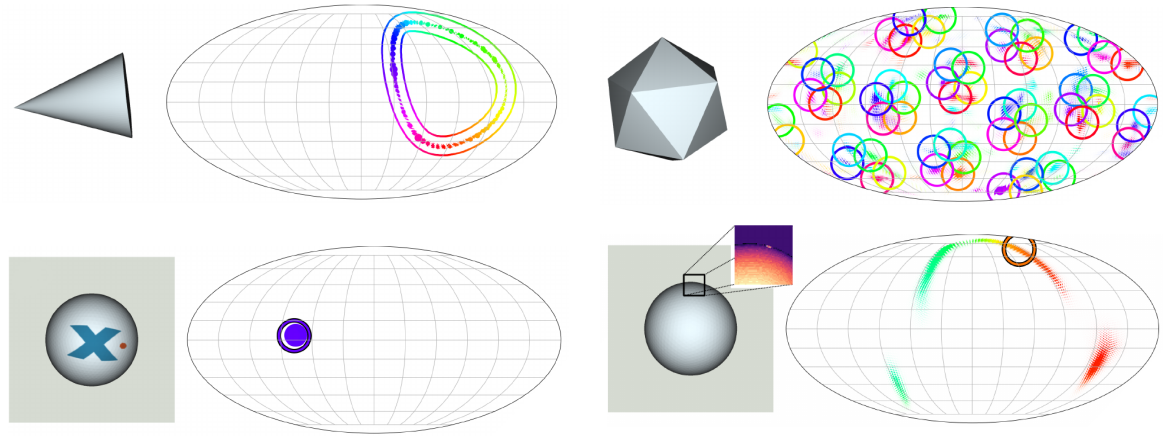


Figure 1.1: **Predicted Distributions of the Implicit-PDF Model.** The circles and continuous lines represent the ground-truth orientations. The dots indicate the estimated orientations with the size corresponding to the probability propagated by the model. **Upper Row:** The Implicit-PDF model is able to express the continuous symmetry of the cone and the 60 discrete symmetries of the icosahedron. **Lower Row:** The visual features break the visual symmetries, showing that Implicit-PDF model estimates an orientation distribution representing the visual symmetries. [5]

PDF) model for rotation manifolds that constructs non-parametric probability distributions over the rotation manifold $\mathbf{SO}(3)$ implicitly to model an object's orientation. This representation is parametrized by a neural network that can be trained by only observing a single object orientation for each training image. The model shows impressive capabilities of representing complex symmetries of platonic solids, such as cones, spheres and icosahedrons, shown in Figure 1.1. The qualitative results show that the model expresses symmetries arising from ambiguities in the visual features of the object. Unfortunately, the model is limited by visual features expressed through object textures, shown in the lower row of Figure 1.1. The distribution fails to capture geometric symmetries that are particularly interesting for robotic applications. Moreover, the model was only constructed to represent distributions over the rotation manifold $\mathbf{SO}(3)$ and not $\mathbf{SE}(3)$, the group containing all rigid body transformations. The training process requires a ground-truth annotated dataset which poses an additional limitation, especially in real-world scenarios when dealing with symmetric objects. This thesis tackles the previously mentioned limitations of the model by making the following contributions:

1. The *ImplicitPosePDF* model which constructs arbitrary pose distributions over $\mathbf{SE}(3)$ using an adaption of the Implicit-PDF model for estimating the translation in $\mathbb{R}^3$ (*Translation-IPDF*) combined with the original Implicit-PDF model

(*Rotation-IPDF*),

2. an adjusted training procedure to model geometric symmetries of arbitrary objects,

3. the *Automatic Pose Labeling Scheme* that produces a pseudo ground-truth set consisting of multiple poses for each training image, eliminating the necessity of ground-truth annotated training data,

4. evaluation on multiple real-world objects exhibiting different types of symmetry, showing that the model is able to construct arbitrary pose distributions over $\mathbf{SE}(3)$ invariant to object textures and negligible geometric features.

The *Automatic Pose Labeling Scheme* and the implicit probability distribution functions for modeling object orientations developed during the thesis were published at the IEEE International Conference on Robotic Computing (IRC) [6].

2 6D Pose Estimation

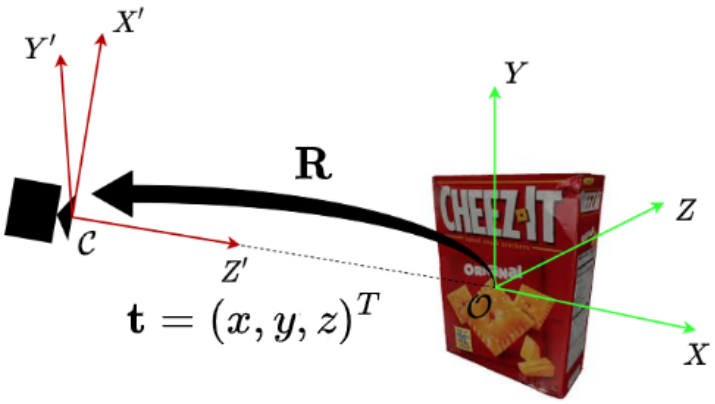


Figure 2.1: **6D Pose Estimation Problem.** The task of 6D pose estimation is to find a rigid transformation consisting of a rotation $\mathbf{R}$ and translation $\mathbf{t}$ transforming the object coordinate system $\mathcal{O}$ to the camera coordinate system $\mathcal{C}$. The resulting transformation is called the pose $\mathbf{P}$ of the object in the camera frame.

Before discussing the pose estimation task itself, we introduce a common notion of symmetry in Section 2.1, meaning we need to define a formal criterion that describes whether two different poses $\mathbf{P}_i$ and $\mathbf{P}_j$ are considered to be symmetric. In the case of two poses being symmetric, both poses might be defined as correct poses in the 6D pose estimation task. First, we define a pose $\mathbf{P} \in \mathbf{SE}(3)$ of an object as a rigid transformation from the object coordinate system $\mathcal{O}$ to the camera coordinate system $\mathcal{C}$, as shown in Chapter 2. The pose consists of a rotation $\mathbf{R} \in \mathbf{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, $\mathbf{P} = (\mathbf{R}, \mathbf{t})$. In this thesis, the rotation is represented by a 3×3 rotation matrix to avoid discontinuities and the translation by three independent coordinates x, y, and z. As shown by Murphy *et al.* [5], the representation through rotation matrices also yields the best results of the Implicit-PDF model.

2.1 Object Symmetries

Symmetries can be split into two categories. Visual symmetries arise due to the lack of distinctive visual features. In this case, an object placed in poses $\mathbf{P}_i$ and $\mathbf{P}_j$ produces

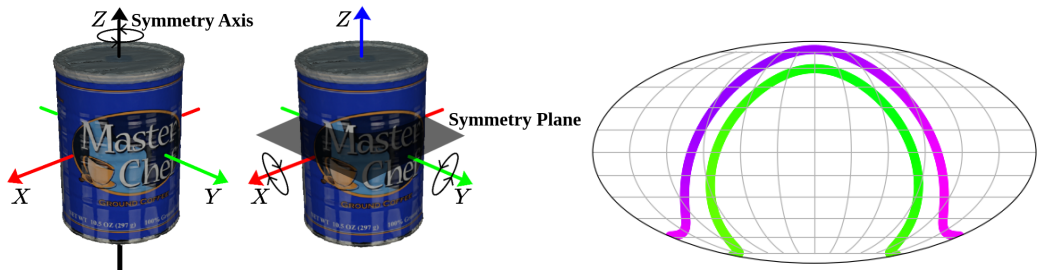
the same image $\mathcal{I}$:

$$\mathcal{I}\left(\mathcal{O}, \mathbf{P}_i\right) = \mathcal{I}\left(\mathcal{O}, \mathbf{P}_j\right). \tag{2.1}$$

Geometric symmetries on the other hand are still present if an object has distinctive visual features. These are solely determined by the geometry of the object. Given the 3D model consisting of n 3D points $\mathbf{x}$ of an object $\mathcal{O}$, two poses $\mathbf{P}_i$ and $\mathbf{P}_j$ are considered geometrical symmetric if they have a small mean closest point distance:

$$\frac{1}{n} \sum_{\mathbf{x}_1 \in \mathcal{O}} \min_{\mathbf{x}_2 \in \mathcal{O}} ||\mathbf{P}_i \mathbf{x}_1 - \mathbf{P}_j \mathbf{x}_2|| \approx 0. \tag{2.2}$$

In Figure 2.2 the exemplar *Can* object from the YCB-Video dataset exhibits multiple geometric symmetries while the texture prevents visual symmetries. The *Can* object can be continuously rotated around the z-axis and rotated by 180° around the x and y-axis respectively to produce symmetric poses. The resulting symmetric orientations, shown in Figure 2.2 (b), form two continuous sets over the rotation manifold $\mathbf{SO}(3)$. The symmetries are further appreciated in the provided video.[1]



(a) Object Symmetries (b) Symmetric Orientations

Figure 2.2: **Can Object Symmetries.** (a) The *Can* object exhibits a continuous symmetry around the z-axis and two discrete flip symmetries along the x and y-axes respectively. (b) The resulting symmetric orientations form two continuous sets of orientations over $\mathbf{SO}(3)$

This thesis follows the proposal of Bregier *et al.* [7] to define the *proper symmetries* $\mathcal{M}_{geo}$ regarding geometric symmetries as a group of rigid transformations:

$$\mathcal{M}_{geo} = \{\mathbf{m} \in \mathbf{SE}(3) \mid \forall \mathbf{P} \in \mathbf{SE}(3), \frac{1}{n} \sum_{\mathbf{x}_1 \in \mathcal{O}} \min_{\mathbf{x}_2 \in \mathcal{O}} ||\mathbf{P}\mathbf{x}_1 - \mathbf{m} \cdot \mathbf{P}\mathbf{x}_2|| \approx 0\}. \tag{2.3}$$

Visual and geometric symmetries arise from ambiguities in the rotational component of a pose. Thus, the symmetries are only expressed through the rotational component. The symmetric poses share a common translation vector.

[1] https://uni-bonn.sciebo.de/s/cjNsVTuiOBAkqlB

2.2 Problem Definition

This definition of symmetry allows us to define the 6D object pose estimation problem in two different ways: Given an RGD image $\mathbf{x}$ depicting an object, (a) estimate a single pose $\mathbf{P} \in \mathbf{SE}(3)$ or (b) estimate the complete set of proper symmetries $\mathcal{M}_{geo}$. Using the definition (a) has a major advantage: symmetric and non-symmetric objects can be treated the same in terms of network architecture, training scheme and inference, only the training loss has to be adjusted. Using the definition (b) makes it necessary to define the symmetries either explicitly or implicitly, making it arguably the harder task.

Figure 2.2 (b) shows a projection of the rotation manifold $\mathbf{SO}(3)$. The symmetries exhibited by the *Can* object depicted in Figure 2.2 (a) result in multiple correct poses. In this case in two continuous sets of symmetric poses with a unique translation vector. Using problem definition (a), every pose coming from these sets is considered to be a correct pose. The challenge of this task is to estimate one of the correct poses as accurately as possible. If we defined the problem as stated in definition (b), the coverage of all correct poses is the objective. In this thesis, we will refer to the problem definition (a) as the *Single 6D Pose Estimation* task and to the problem definition (b) as the *6D Pose Distribution Estimation* task.

3 Related Work

3.1 6D Pose Estimation

In recent years the performance of 6D pose estimation methods was significantly improved through the emergence of deep learning. Larger networks and increasing computational power allow us to train more accurate models. Especially in combination with *Convolutional Neural Networks* (CNNs) the 6D pose estimation from RGB data has seen significant progress.

In our human-made world objects often exhibit some kind of rotational symmetry. These symmetries create ambiguities in the ground-truth pose which might hamper the learning abilities of such approaches. Thus, it is evident that in order to efficiently train deep learning models, symmetries have to be incorporated into the training process. Among the recent approaches, strategies have been proposed to either implicitly or explicitly utilize symmetry annotations.

Xu *et al.* [4] and Li *et al.* [3] employ the ShapeMatch-Loss [1] which copes object symmetries during training. It implicitly selects a pose closest to the predicted pose from the proper symmetry set as the correct pose. While this does not require any prior knowledge about object symmetries, this variability in the ground-truth pose still hampers the learning ability. Opposed to the implicit approach, Pitteri *et al.* [8] and Amini *et al.* [2] mapped the symmetric rotations to a single "canonical" rotation. The downside of such approaches is the explicit definition of object symmetries. Object symmetries are typically not known prior to training or must be defined by hand, which makes them untractable for large datasets.

Another common explicit approach is modeling symmetries using pre-defined symmetry classes. Rad and Lepetit [9] defined an additional auxiliary task to classify the type of symmetry an object exhibits that benefits the single 6D pose estimation task with additional properties. Esteves *et al.* [10] and Saxena *et al.* [11] proposed methods that learn visual features equivariant to specific symmetry classes. While using pre-defined symmetry classes may improve the 6D pose estimation accuracy, it is limited in the scope of modeling arbitrary symmetries. The previously mentioned approaches modeled symmetries to improve the precision of single 6D pose estimations. Different strategies were proposed to cope with object symmetries to increase

the learning ability of such models. They made no approach toward estimating and modeling object symmetries during inference.

The approach by Corona *et al.* [12] modeled the pose estimation task as an image comparison task. Given an input image, a codebook of images is used for comparison. They trained a model to estimate a similarity score between two RGB images which is then used to find the best matching codebook image. The pose of the best matching codebook image is taken as the pose estimation during inference. In the presence of symmetry, multiple codebook images with a high similarity score are chosen. While this approach does not need explicit symmetry definition to model object symmetries, it requires a large codebook of images for accurate pose estimation. This comes at the cost of a high inference time requirement. To improve the inference time Sundermeyer *et al.* [13] introduced an Augmented Autoencoder to learn a low-dimensional latent space representation of the images used for image comparison. This improves the inference time but does not make it real-time capable and usable for robotic applications. The major drawback of such image comparison approaches is that they only model visual symmetries and not geometric symmetries which are more interesting in robotic applications. Instead of making a single prediction, Manhardt *et al.* [14] trained their model to predict a set of poses. The model predicts up to five different visual symmetric poses. This is neither sufficient to cover the complete set of proper symmetries nor to reason about the type of symmetry exhibited by an object.

Recent approaches model symmetries by estimating probability distributions over the rotation manifold $\mathbf{SO}(3)$. Deng *et al.* [15] and Gilitschenski *et al.* [16] modeled multiple pose hypotheses as a mixture of Bingham distributions and trained a CNN model to estimate the distribution parameters from RGB-D images. Since a single Bingham probability distribution function describes a uni-model distribution on the rotation manifold, a high number of Bingham distributions is required for an accurate approximation of more complex object symmetries, such as continuous symmetries. The computation time needed during inference is strongly increased by objects exhibiting a high degree of symmetry. Okorn *et al.* [17] modeled multiple pose hypotheses using Fisher distributions. These methods suffer from overhead in the computation of the normalization term. Moreover, using parametrized probability distributions makes a prior assumption on the pose distribution which makes it unlikely to be extended to arbitrary complex object symmetries.

3.2 Ground-Truth Acquisition

The aforementioned learning approaches are trained in a supervised manner observing a single ground-truth pose. This requires ground-truth annotated datasets.

Sun *et al.* [18] showed that an increase in the dataset size results in a logarithmical increase in model performances. In ablation studies, Murphy *et al.* [5] confirmed this proposition. Increasing the dataset size logarithmically increases the Implicit-PDF model's performance, stagnating after a dataset size of 50,000 images. In real-world scenarios, acquiring such large-scale ground-truth annotated datasets constitutes a bottleneck in the training process. Xiang *et al.* [1] annotated the YCB-Video dataset by manually annotating the first frame of each video. The remaining frames are annotated by tracking the camera trajectory revolving around the object and finally deriving the ground-truth pose analytically. The quality of the ground-truth pose annotations strongly relies on the accuracy of the camera trajectory and precise camera calibration. Hodan *et al.* [19] manually annotated the T-Less dataset by reconstructing the 3D scene using the RGB-D images and CAD object models. Misalignments in the scene reconstruction are repeatedly corrected by hand until the alignment is satisfactory. For small datasets like the T-Less dataset which only consists of 1,296 training images for each object, this method is tractable. It is evident that such an approach is not tractable for large-scale datasets. Overall, both methods are able to annotate ground-truth poses with manual supervision but miss the ability to output multiple ground-truth annotations to characterize the symmetries present in the images. Extending those methods to also define ground-truth symmetries would require prior knowledge about object symmetries.

4 ImplicitPosePDF Model

The *ImplicitPosePDF* (IPPDF) model presented in this thesis is based upon the Implicit-PDF (IPDF) model proposed by Murphy *et al.* [5]. The IPDF model has shown tremendous capabilities of expressing object symmetries implicitly. To approach 6D pose estimation, we extend the IPDF model to estimate a pose distribution over $\mathbf{SE}(3)$. Murphy *et al.* [5] noted that the translation may be predicted simultaneously within the same model. In a series of experiments, we observed that the incorporation of translation and rotation into one model compromises the ability of the model to capture object symmetries while making accurate pose estimations. Thus, the rotation and translation estimation is decoupled into two separate models, the *Rotation-IPDF* and *Translation-IPDF* model. The two models and methods used are discussed in detail throughout Section 4.1. In the following, we present our adjusted training procedure in Section 4.2 and the visualization method used to evaluate the pose distributions in Section 4.3. The ground-truth labels utilized for training are generated by the *Automatic Pose Labeling Scheme* which will be introduced in Chapter 5. Finally, we put the *ImplicitPosePDF* model to a test in Chapter 6. We highlight the strengths of the IPPDF model to predict accurate 6D poses while being able to capture the complete set of proper symmetries $\mathcal{M}_{geo}$.

4.1 Methods

The *ImplicitPosePDF* model, illustrated in Figure 4.1, expresses 6D pose distributions by modeling two separate non-parametric probability distribution functions for the orientation and translation respectively. The orientation distribution generated by the *Rotation-IPDF* model is responsible for capturing the object symmetries. The *Translation-IPDF* generates a distribution that approximates the unique translation vector as accurately as possible. Ideally, this distribution describes a highly peaked uni-modal distribution.

At the center of the two models stands a multi-layer perceptron that implicitly represents the probability distribution. Both models are similar with respect to the used methods and network architecture. The probability distributions are approximated as a histogram. The rotation manifold $\mathbf{SO}(3)$ and translation space $\mathbb{R}^3$ are

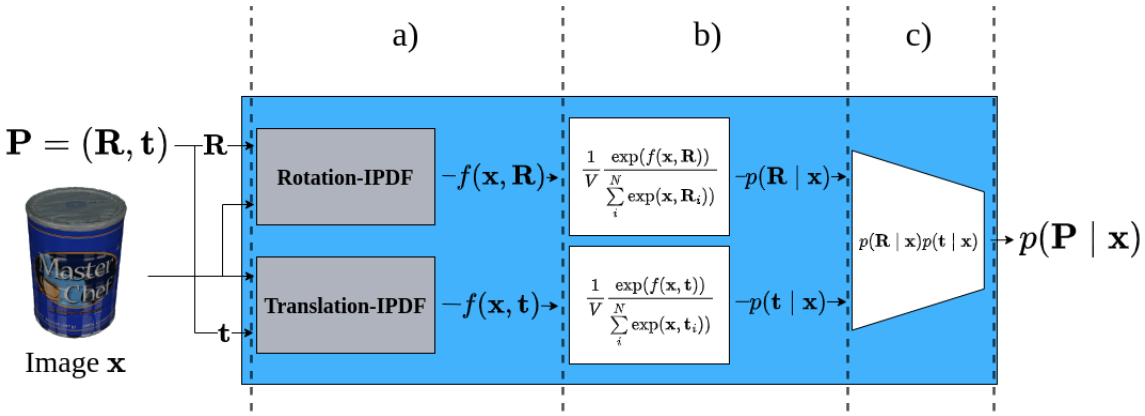


Figure 4.1: **ImplicitPosePDF Model.** The model's objective is, given an image $\mathbf{x}$, to estimate the probability of the given pose hypothesis $\mathbf{P}$. The rotation and translation probability are estimated separately. The *Rotation-IPDF* and *Translation-IPDF* estimate the unnormalized joint log probability. Using the pose queries, we retrieve the normalized probabilities that are multiplied to yield the probability of the pose hypothesis $p(\mathbf{P} \mid \mathbf{x})$

discretized into bins of the histogram and the probability of each bin is approximated by our model. In part a) of Figure 4.1, the two models estimate the unnormalized joint log probability function $f : \mathbf{x} \times \mathbf{SO}(3) \to \mathbb{R}$ and $f : \mathbf{x} \times \mathbb{R}^3 \to \mathbb{R}$ given an input image $\mathbf{x} \in \mathcal{I}$ and pose hypothesis $\mathbf{P} \in \mathbf{SE}(3)$. The sampling techniques described later allow us to efficiently approximate the normalization term. The resulting normalized probability functions $p(\mathbf{R} \mid \mathbf{x})$ and $p(\mathbf{t} \mid \mathbf{x})$ are finally combined to produce the pose probability distribution function $p(\mathbf{P} \mid \mathbf{x})$.

To prevent repetitions we introduce $\mathbb{G}$ as a placeholder for either the rotation manifold $\mathbf{SO}(3)$ or the translation space $\mathbb{R}^3$. Equally, we define $\mathbf{G}$ as a placeholder for either the rotation $\mathbf{R}$ or the translation $\mathbf{t}$.

Mathematical Derivation. In the following we derive the method used in parts b) and c) of the *ImplicitPosePDF* model presented in Figure 4.1. We obtain $p(\mathbf{G}|\mathbf{x})$ from the network output using the product rule:

$$p(\mathbf{G}|\mathcal{I}) = \frac{p(\mathbf{G}, \mathbf{x})}{p(\mathbf{x})}. \tag{4.1}$$

The joint distribution $p(\mathbf{x}, \mathbf{G})$ is derived through:

$$p(\mathbf{x}, \mathbf{G}) = \alpha \exp(f(\mathbf{x}, \mathbf{G})), \tag{4.2}$$

using the normalization term α. Note that the computation of α is infeasible because it would require integration over the space of images $\mathcal{I}$. $p(\mathbf{x})$ is estimated by marginalizing over $\mathcal{G}$, and approximating the integral with a discrete sum:

$$\begin{aligned} p(\mathbf{x}) &= \int_{\mathbf{G}\in\mathcal{G}} p(\mathbf{x},\mathbf{G}) d\mathbf{G} \\ &= \alpha \int_{\mathbf{G}\in\mathcal{G}} \exp(f(\mathbf{x},\mathbf{G})) d\mathbf{G} \\ &\approx \alpha \sum_{i}^{N} \exp(f(\mathbf{x},\mathbf{G}_i)) V. \end{aligned} \tag{4.3}$$

The queries $\{\mathbf{G}_i\}$ are the centers of an equi-volumetric partitioning of either $\mathbf{SO}(3)$ or $\mathbb{R}^3$ with N partitions of volume $V = \frac{\pi}{N}$ or $V = \frac{V_{sub}}{N}$. V_{sub} is the volume a sub-space $\mathbf{T} \subset \mathbb{R}^3$ containing the possible translation vectors $\mathbf{t} \in \mathbf{T}$.

The normalization term α cancels out in the expression for $p(\mathbf{G}|\mathbf{x})$, leading to a formulation that can be computed by querying the neural network over the hypotheses $\{\mathbf{G}_i\}$:

$$p(\mathbf{G}|\mathbf{x}) \approx \frac{1}{V} \frac{\exp(f(\mathbf{x},\mathbf{G}))}{\sum_{i}^{N} \exp(f(\mathbf{x},\mathbf{G}_i))}. \tag{4.4}$$

On a small scale, we can assume that the rotation component is independent of the translation component, meaning $p(\mathbf{R}|\mathbf{x})$ and $p(\mathbf{t}|\mathbf{x})$ are independent of each other. Therefore, the final pose probability $p(\mathbf{P}|\mathbf{x})$ can be derived through:

$$p(\mathbf{P}|\mathbf{x}) = p(\mathbf{R}|\mathbf{x}) p(\mathbf{t}|\mathbf{x}). \tag{4.5}$$

Sampling. To make the integration over $\mathbf{G}$ feasible in Equation (4.3), the integral is approximated by a sum over an equi-volumetric partitioning of $\mathbf{SO}(3)/\mathbb{R}^3$. To construct an accurate probability distribution, it is crucial to precisely approximate the normalization term. This requires a dense grid which comes at the cost of computational effort. For training and pose estimation a sparse grid is sufficient for both models.

To sample from the rotation manifold $\mathbf{SO}(3)$, we follow the sampling method proposed by Murphy *et al.* [5] to produce an equi-volumetric grid $\{\mathbf{R}_i\} \subseteq \mathbf{SO}(3)$. The equi-volumetric sampling in $\mathbf{SO}(3)$ is produced in two steps. First, equal-area grids on the 2-sphere are produced using the HEALPix method [20]. The equal-area grids are produced in hierarchical order, leading to a convenient side effect of multi-resolution sampling. Second, the Hopf fibration [21] is used to create a great circle through

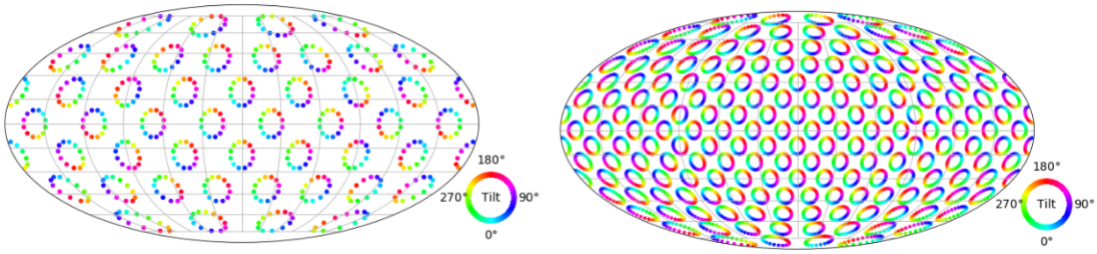


Figure 4.2: **Equi-Volumetric Sampling of SO(3).** The normalization term is approximated by computing the unnormalized probabilities on an equi-volumetric grid. The method proposed by [21] produces equal-area grids hierarchically on the 2-sphere using the HEALPix method [20] and finally samples rotations using the Hopf fibration [22]. Left: a grid with 576 rotations, right: 4608 rotations [5].

each point in the equal-area grid. Finally, the equi-volumetric sampling of $\mathbf{SO}(3)$ is generated in recursive steps with a grid of size 72 as a starting point. The size of the grid increases by the factor eight after each subdivision i, $|\{\mathbf{R}_i\}| = 72 \cdot 8^i$. Figure 4.2 displays the grid with $i = 1$ and $i = 2$.

To sample from the translation space $\mathbb{R}^3$, we restrict the translation vector in the spatial boundaries of the given scene. It is inefficient to include impossible translation vectors in the sampling process. We define $\mathbf{T} \subset \mathbb{R}^3$ as the subset containing all possible translation vectors:

$$\mathbf{T} = \{\mathbf{t} \in \mathbb{R}^3 \mid \mathbf{t}_x \in [t_0, t_1] \wedge \mathbf{t}_y \in [t_2, t_3] \wedge \mathbf{t}_z \in [t_4, t_5]\} \tag{4.6}$$

$t_0, \ldots, t_5$ mark the borders of the possible translation queries. These values can be chosen manually or also be learned before or during the training from the training data. Finally, the sampling of $\mathbf{T}$ is done using a regular grid with equi-volumetric, cubic cells. The centers of the cells are used as the translation queries for the *Translation-IPDF* model.

Inference Since the rotational and translational components of a pose are not intertwined, we decouple the estimation of the most likely pose as well as the pose distribution. For inference, the *Rotation-IPDF* model and *Translation-IPDF* model make separate estimations of either the most likely orientation/translation or the complete orientation/translation distribution.

To predict a 6D pose, the objective of both models is to find the single most likely pose. Using the differentiable network output $f(\mathbf{x}, \mathbf{G})$, both models optimize $\mathbf{G}$ using gradient ascent to yield:

$$\hat{\mathbf{G}}_\mathbf{x} = \arg\max_{\mathbf{G} \in \mathcal{G}} f(\mathbf{x}, \mathcal{G}). \tag{4.7}$$

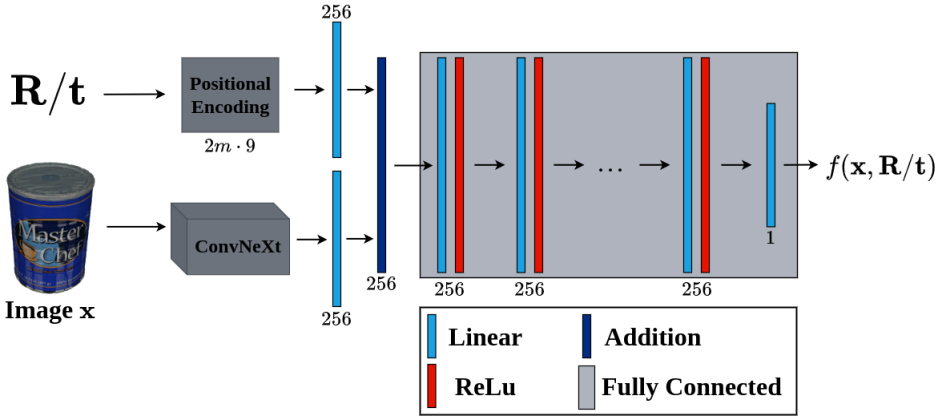


Figure 4.3: **Implicit-PDF Model.** Given an image $\mathbf{x}$ and an orientation hypothesis $\mathbf{R}$, respectively a translation hypothesis $\mathbf{t}$, the Implicit-PDF model is trained to generate the unnormalized joint log probability of the orientation/translation hypothesis and the image.

The single most likely rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$ are then combined to yield the pose prediction $\hat{\mathbf{p}}$:

$$\hat{\mathbf{p}} = (\hat{\mathbf{R}}, \hat{\mathbf{t}}). \tag{4.8}$$

The pose distribution $p(\mathbf{SE}(3) \mid \mathbf{x})$ is represented through two seperate distributions $p(\mathbf{SO}(3) \mid \mathbf{x})$ representing the orientation and $p(\mathbf{T} \mid \mathbf{x})$ representing the translation. To reconstruct each distribution, we generate a dense equi-volumetric sampling of $\mathbf{SO}(3)$ and the translation space $\mathbf{T}$ and evaluate Equation (4.4) on each query.

Neural Network. The *Rotation-IPDF* and *Translation-IPDF* models in part a) of Figure 4.1 share a common network architecture. The network architecture is depicted in Figure 4.3. The image is fed to a feature extractor which extracts the visual features. A standard state-of-the-art feature extractor is sufficient to create a visual representation used as input for the MLP. We carried out experiments with different feature extractors as the backbone to our model in Section 6.5, and finally settled for the ConvNeXt model proposed by [23] in the tiny configuration.

Positional encoding each element of the rotation or translation query with m elements proved to yield a better performance of our model. Together with the positional encoded query, the feature descriptor is fed to MLP. The MLP consists of n fully connected layers of size 256 with a ReLu activation function. The last layer outputs the unnormalized joint log probability for a given image and rotation/translation query.

4.2 Training

A standard loss to learn probability distribution functions is the negative log-likelihood $\mathcal{L}(\mathbf{x}, \mathbf{G}_{GT}) = -\log(p(\mathbf{G}_{GT} \mid \mathbf{x}))$. The log-likelihood measures how tightly two probability distributions fit, evaluated on a single point. Given a single ground truth, the model is trained to minimize the negative log-likelihood. This requires estimating the pose distribution as described in Equation (4.4). For normalizing the distribution, the pose distribution needs to be evaluated on a grid $\{\mathbf{G}_i\}$ in $\mathcal{G}$. The grid is finally rotated to include the ground truth $\mathbf{G}_{GT}$.

During training, the normalization term is not required to be approximated with high accuracy. A coarse grid is sufficient during training. Murphy *et al.* [5] showed that the model is even robust enough to be trained without an equi-volemtric grid. Randomly sampled $\{\mathbf{G}_i\}$ during training yield comparable model performances as the models trained with an equi-volumetric grid.

The model as proposed by Murphy *et al.* [5] is trained observing a single ground-truth pose for each image. In Chapter 6, we show that training the model in this manner does not yield the desired set of proper symmetries $\mathcal{M}_{geo}$. Using the *Automatic Pose Labeling Scheme*, which is later introduced in Chapter 5, we generate a set of pseudo ground-truth poses representative of the proper symmetries. The *Rotation-IPDF* model is then trained on varying single ground-truth annotations for each image. The training of the *Translation-IPDF* remains the same as the ground-truth translation is unique for each image.

4.3 Visualization

Finding a proper method to visualize poses in our use case is not trivial. The objective of the visualizations is to allow reasoning about the composition of a set of rotations or translation vectors and the propagated probabilities. Thus, besides the visualization of the rotation or translation itself, an additional degree of freedom is required to visualize the probability of each point. Furthermore, the visualization should facilitate a notion of distance between rotations or translations to reason about the accuracy of the estimated rotations or translations in relation to the ground truth.

Rotation Manifold SO(3). Rotation matrices have three degrees of freedom. Murphy *et al.* [5] proposed a method that represents two degrees of freedom as a 2-sphere. The third degree of freedom is represented using the Hopf fibration [22] by a great circle of points to each point on the 2-sphere. The 2-sphere is projected onto a plane using the Mollweide projection. Finally, the third degree of freedom is expressed through a color wheel that indicates the location of a point on the great circle. The

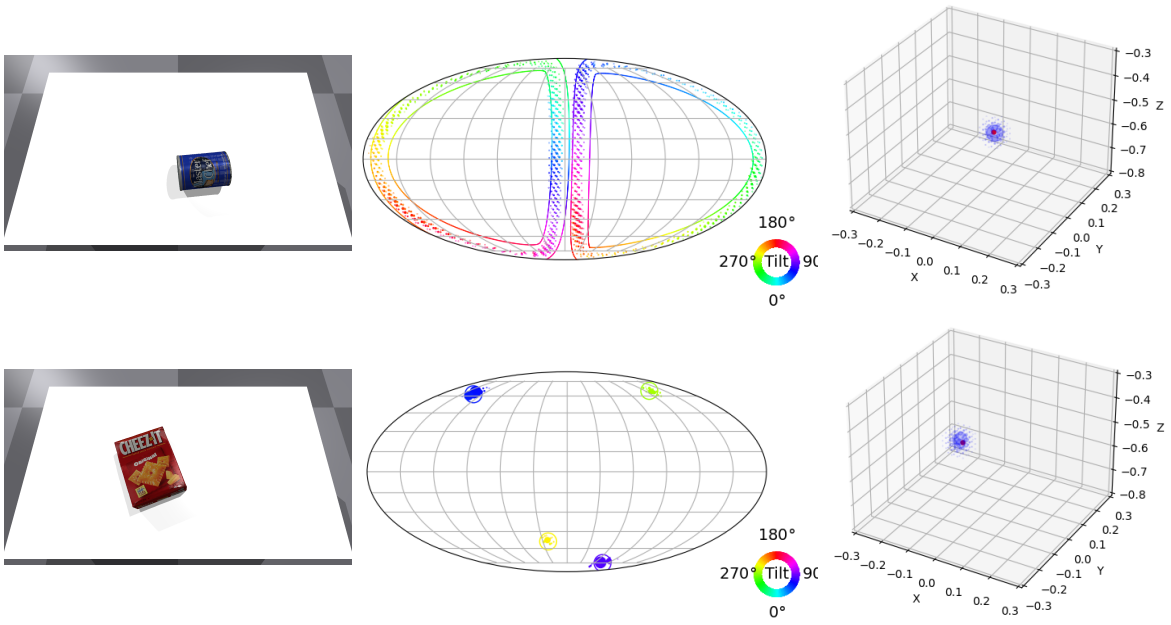


Figure 4.4: **Visualization Technique for Pose Distributions.** The estimated distributions for the orientation and translation are visualized separately. The dots in the visualizations show the estimated rotation/translation with their size corresponding to the propagated probability. The ground-truth rotations are marked through circles and continuous lines. The unique ground-truth translation is marked through a single red dot.

middle row of Figure 4.4 presents the visualization technique for the rotation manifold **SO**(3). The single dots represent the rotations estimated by the model with the size of the dots corresponding to the propagated probability. Through continuous lines and circles, the ground-truth symmetries are visualized on the rotation manifold. Finally, this method allows reasoning not only about single rotations but also about complete pose distributions compared to the underlying ground-truth distribution.

Translation space $\mathbb{R}^3$. The translation component is represented by three independent coordinates, x, y and z. These are visualized in a three-dimensional cartesian coordinate system. The resulting visualization can be seen in the right row of Figure 4.4. The points represent the predicted translation vectors with the size of the points indicating the estimated probability. The single ground-truth vector is represented through a red dot. Since the translation component is not compounded by symmetries, the translation component can be expressed through a unique translation vector. The ground-truth distribution is a uni-modal Gaussian probability distribution with a small standard deviation. Therefore, a three-dimensional projection on a plane is sufficient to reason about the estimation of the translation distribution.

5 Automatic Pose Labeling Scheme

As shown later throughout the evaluation in Chapter 6, the *ImplicitPosePDF* model trained with a single ground-truth pose for each image is not able to represent the complete set of proper symmetries. Training the ImplicitPosePDF directly with the ground-truth symmetries enables the model to construct pose distributions representative of object symmetries. In this case, the ground-truth symmetries are derived analytically and a set of poses, accurately approximating the object symmetries, is used during training. Defining ground-truth symmetries analytically requires prior knowledge about the object's symmetries and the object's pose. Since it is impossible to define every source of object symmetry, this approach is not scalable to arbitrary objects.

As an approach to make the ground-truth symmetries available to us during training without any prior knowledge, this thesis proposes the *Automatic Pose Labeling Scheme* (APLS). In the following, the complete pose labeling pipeline is described in detail. First, this thesis formally defines the problem intended to be solved by the pose labeling scheme in Section 5.1. Next, the methods used in each stage of the pipeline are described and motivated in Section 5.2. Finally, these methods are incorporated into the three-stage pipeline and their implementation is discussed in detail in Section 5.3. To evaluate the quality of the resulting pseudo ground-truth poses, we show experimentally in Chapter 6 that the APLS generates a set of ground-truth poses that accurately covers the set of proper symmetries. The effectiveness of the pseudo ground-truth labels to train the IPPDF model is shown by comparing it against models trained with analytically derived ground-truth symmetries.

5.1 Problem Definition

First, we need to define the task intended to be solved by the APLS to motivate the choice of methods. We assume the segmentation information and camera intrinsic values corresponding to each image to be given. The objective of the APLS is to register a given 3D object model $\mathcal{O}_{model}$ against the observed point cloud $\mathcal{O}_{obs}$ to output a pose from the set of proper symmetries $\mathcal{M}_{geo}$. Formally, the goal is to

output a pose $\mathbf{P}$ that satisfies the condition:

$$\frac{1}{n} \sum_{x_1 \in \mathcal{O}_{obs}} \min_{x_2 \in \mathcal{O}_{model}} ||x_1 - \mathbf{P}x_2|| \approx 0, \tag{5.1}$$

meaning that pose $\mathbf{P}$ is one of the symmetric poses corresponding to the observed point cloud. As already described, a single ground-truth label for each image is not sufficient to train the IPPDF model. Thus, it should be able to generate a set of ground-truth poses that are representative of the object symmetries. Note that the APLS does not output the set of proper symmetries as defined in Equation (2.3), but rather outputs the final poses resulting from the symmetries. Using this dual representation of the geometric symmetries has the advantage that the IPPDF model directly propagates the pose distribution over all possible poses and does not simply define the geometric symmetries. Retrieving $\mathbf{m}$ from the set of ground-truth poses is as simple as multiplying all generated poses with the inverse of a fixed pose.

5.2 Methods

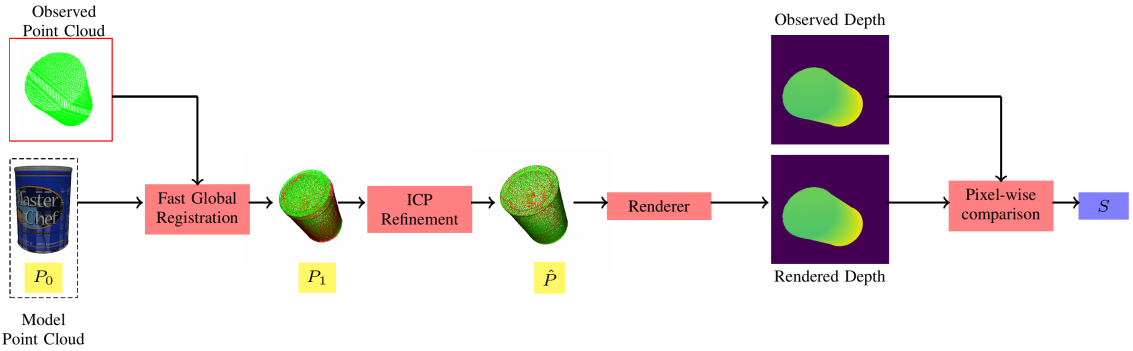


Figure 5.1: **Automatic Pose Labeling Scheme.** Given an RGB-D image without ground-truth labels, the APLS produces a pseudo ground-truth label in three stages. First, the object model is put in an initial pose P_0. The Fast-Global-Registration algorithm then generates an initial alignment P_1. Together with the initialization set $\mathbf{P}_{init}$, this pose is fed to the ICP algorithm to produce the final poses $\hat{P}$. In the final step, the observed depth map is compared against the depth map rendered using poses $\hat{P}$ to yield the similarity score S. The pose with the smallest S is taken as pseudo ground-truth label [6].

The *Automatic Pose Labeling Scheme*, shown in Figure 5.1, aligns a given object model $\mathcal{O}$ with the observed point cloud $\mathcal{O}_{obs}$ using a three-stage pipeline. For this purpose, we assume that an RGB-D image with segmentation information is given. Additionally, camera intrinsic values need to be provided. In the first stage, the

object model is globally registered against the observed point cloud to yield a rough alignment. In the second stage, this alignment is refined to produce an accurate pose of the object. To determine the quality of this pose, the tightness of the alignment is checked in the final stage. In the following, the methods used in each step of the pipeline are discussed. The two-stage alignment process is described in a bottom-up manner to motivate design choices.

Observed point cloud. From the RGB-D image the observed point cloud $\mathcal{O}_{obs}$ is extracted by unprojecting the depth map $\mathcal{D}(x, y)$ into 3D. Knowing the camera intrinsics (focal length (f_x, f_y) and focal point (c_x, c_y)), the 3D points, corresponding to each pixel $p = (p_x, p_y)$, can be recovered using a pinhole camera model:

$$\begin{aligned} x &= \frac{c_x - p_x}{f_x} \cdot d(x, y) \\ y &= \frac{c_y - p_y}{f_y} \cdot d(x, y) \\ z &= \mathcal{D}(x, y). \end{aligned} \tag{5.2}$$

Finally, by using the segmentation information the 3D points belonging to the depicted object are extracted to obtain the observed point cloud $\mathcal{O}_{obs}$.

Iterative-Closest-Point algorithm. A common method for registration tasks is the *Iterative-Closest-Point* (ICP) algorithm. Given two point clouds, namely the *source* point cloud $\mathcal{O}_{source}$ and *target* point cloud $\mathcal{O}_{target}$, the ICP algorithm iteratively matches the points in the *source* point cloud to the closest point in the *target* point cloud. Iteratively, the algorithm finds the rotation $\mathbf{R}$ and translation $\mathbf{t}$ that minimizes,

$$\frac{1}{n} \sum_{x_1 \in \mathcal{O}_{source}} \min_{x_2 \in \mathcal{O}_{target}} ||x_1 - \mathbf{R}x_2 + \mathbf{t}||. \tag{5.3}$$

Ideally, a perfect alignment of the point clouds yields a distance, as defined in Equation (5.3), of approximately zero. Comparison with Equation (5.1) shows that a converged ICP solution yields a pose from the set of proper symmetries. Moreover, it shows that the poses from the proper symmetry set ideally create local optima in the optimization domain which causes the ICP algorithm to converge to different symmetric poses depending on the initialization of ICP. This variability in the resulting poses comes with the downside of the ICP solutions being only locally optimal. Without proper initialization with a pose close to one of the symmetric poses, ICP tends to diverge to local optima producing poses that do not correspond to the proper symmetry set. This requires a previous global registration step to produce an initial

alignment such that ICP converges to the correct local optima.

Fast-Global-Registration algorithm. To retrieve an initial global alignment, we chose the *Fast-Global-Registration* algorithm [24] due to its computational efficiency. Before the optimization process, correspondences between the points of the two point clouds are established through rapid feature matching using Fast Point Feature Histogram features [25]. During the optimization process, the transformation is iteratively optimized, minimizing an objective function, without recomputing these correspondences. This leads to an extremely efficient alignment algorithm that yields a rough alignment of two point clouds. This initial alignment needs to be further refined to produce an accurate pose of the object.

Pixel-wise comparison. To ensure proper ground-truth labeling, we need to assess the quality of the final pose generated by the ICP algorithm. A naive implementation of a quality measure is the evaluation of the alignment of two corresponding points from the point clouds using the L2 metric or similar distance metrics. Using the nearest neighbors as corresponding points, this metric yields the objective function from the ICP algorithm. Since the observed point cloud is extracted from the depth map of the image, this point cloud suffers from a high degree of self-occlusion, leading to potentially wrong correspondences. In these scenarios, a point-wise comparison fails to distinguish between bad and good alignments. To eliminate the effect of self-occlusion, we chose a pixel-wise comparison approach, comparing the observed depth map $\mathcal{D}_{obs}$ with a rendered depth map $\mathcal{D}_{rend}$ of the object put in the pose resulting from the ICP algorithm. The rendered depth map is generated using the Stillleben framework [26]. From the observed and rendered depth map, the edges are extracted using the *Canny Edge Detector* [27] and dilated to make it more robust to minor differences in the contour of the object. Finally, a similarity score S computed by measuring the L2 distance between the edge pixels $\{\mathbf{e}^i_{obs}\} \in \mathbf{E}_{obs}$ from $\mathcal{D}_{obs}$ and the nearest neighbour from the edge pixels $\{\mathbf{e}^i_{rend}\} \in \mathbf{E}_{rend}$ coming from $\mathcal{D}_{rend}$ and vica versa:

$$S = \frac{1}{|\mathbf{E}_{obs}| + |\mathbf{E}_{rend}|}\left(\sum_i \min_j(||\mathbf{e}^i_{obs} - \mathbf{e}^j_{rend}||_2) + \sum_j \min_i(||\mathbf{e}^j_{rend} - \mathbf{e}^i_{obs}||_2)\right). \tag{5.4}$$

The similarity score expresses the quality of the alignment of the point clouds produced using the ICP algorithm and can be utilized to determine good alignments from bad alignments. A lower score indicates a better alignment.

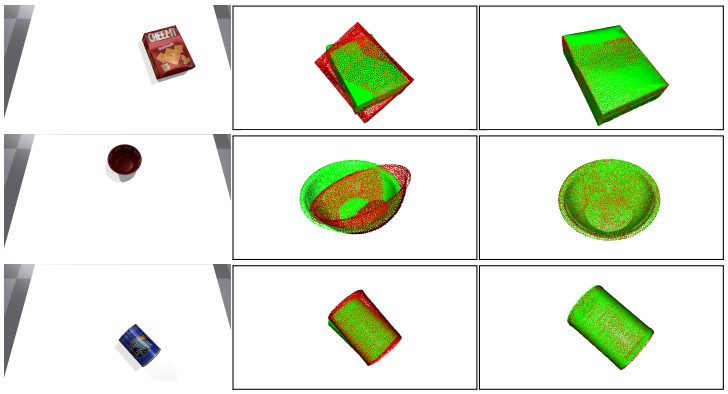


Figure 5.2: **Pseudo Ground-Truth Generation Process.** The observed point cloud $\mathcal{C}_{obs}$ is colored green and the point cloud resulting from each stage is colored red. (a) RGB image, the observed point cloud is extracted from (b) Pose generated by the Fast-Global-Registration algorithm. (c) Final pseudo ground-truth pose generated by the APLS [6].

5.3 Pipeline

The complete pipeline is depicted in Figure 5.1. The results of each stage are shown in Figure 5.2. The starting point of the three-stage pipeline is the extraction of the observed point cloud $\mathcal{O}_{obs}$ from the image in column (a) of Figure 5.2. The object model $\mathcal{O}_{model}$ is put into an initial random pose P_0 to ensure that the pipeline converges to different poses. In the first stage of the pipeline, the *Fast-Global-Registration* algorithm is utilized to produce an initial alignment of the object model and the observed point cloud. The rough alignment of the two point clouds can be seen in column (b) of Figure 5.2 The resulting pose P_1 roughly aligns the object axes of the object model with the observed point cloud. This pose can then be used as an initial alignment for the *Iterative-Closest-Point* algorithm.

Before the next stage, an initialization set $\mathbf{P}_{init}$ of poses is generated to force the ICP algorithm to converge to different local optima and to determine good alignments in the final stage. To generate the initialization set, we take pose P_1 as a starting point and rotate it iteratively eight times by $45°$ around the x,y, and z-axis respectively. This set of poses is then used as initial alignments for the ICP algorithm, which refines the poses and produces the final poses $\hat{P}$. In the last stage of this pipeline, the quality of these poses is assessed. For each pose, the similarity score is computed and subsequently compared. Finally, the pose with the smallest score is chosen as the final pseudo ground-truth pose.

Our experiments in Section 6.4 show that the majority of final poses correspond

to one of the ground-truth symmetries. To make the pipeline computational more efficient in practice, we introduced a threshold $\mathcal{T}$ to determine good alignments. Using this method to distinguish between good and bad alignments, all good alignments are included in the set of pseudo ground-truth poses and consequently, the number of iterations of the APLS for each image is reduced. The thresholds however need to be determined experimentally prior to the pseudo ground-truth generation process. We found that a threshold of $\mathcal{T} \approx 5$ works well for most objects.

6 Evaluation

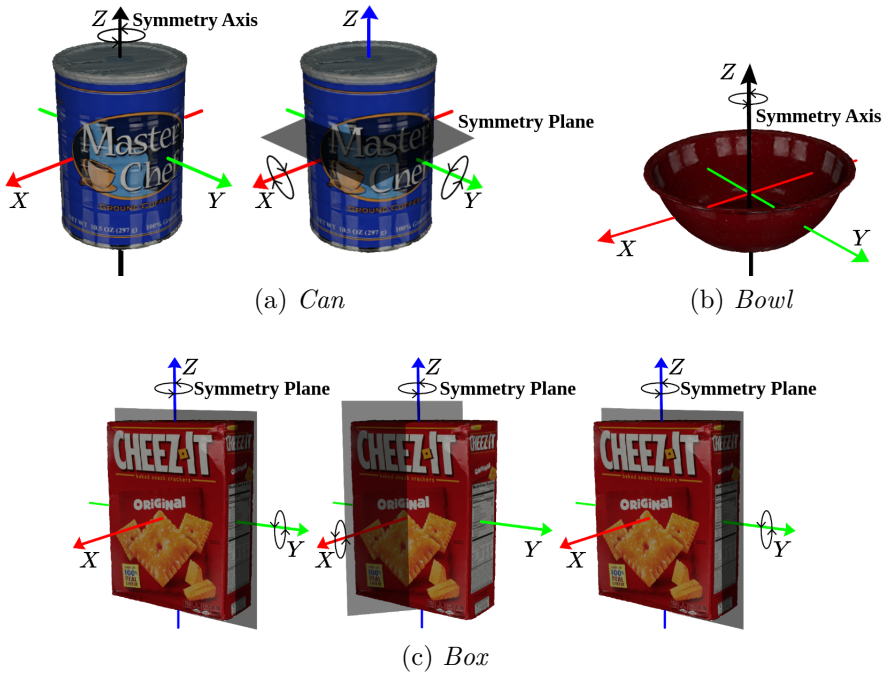
Murphy *et al.* [5] concluded that the *Implicit-PDF* model is able to represent arbitrarily complex distributions for 3D orientations. An implicit representation of the orientation distributions yields better results than existing state-of-the-art methods. The model allows us to reason about object symmetries and pose uncertainties. This thesis extends the approach to estimating pose distributions over $\mathbf{SE}(3)$ expressive of geometric symmetries found in real-world applications.

For evaluation, this thesis will examine the performance of the *ImplicitPosePDF* (IPPDF) model to construct arbitrary complex pose distributions over $\mathbf{SE}(3)$ on symmetric objects invariant to object textures. First, the datasets and evaluation metrics used are introduced in Section 6.1 and Section 6.2. The pseudo ground-truth labels for each dataset are generated using the *Automatic Pose Labeling Scheme* (APLS). The quality of the pseudo ground-truth labels is assessed in Section 6.4. Before examining the performance of the IPPDF model trained with the pseudo ground-truth labels, the implementation of the models used for evaluation is described in detail in Section 6.3. The backbone feature extractor was experimentally determined and the ablation studies are described in detail in Section 6.5. In addition, we further investigate the impact of different visual features. In the following Section 6.6, Section 6.7 and Section 6.9, the performance of the IPPDF model is evaluated on different symmetric objects on the *6D Pose Distribution Problem* as well as the *Single 6D Pose Estimation Problem*.

6.1 Datasets

To evaluate the performance of the APLS and the IPPDF model in real-world scenarios, we created a photorealistic dataset. Additionally, the T-Less dataset is used as a popular dataset to compare our approach with state-of-the-art methods.

Photorealistic Dataset. The dataset contains three objects, *can*, *box* and *bowl* from the YCB-Video dataset exhibiting different types of geometric symmetries. To simulate a realistic robotic environment, the objects are placed on a tabletop and put in randomly sampled physical-plausible poses. The possible translation vectors are



(a) *Can* (b) *Bowl*

(c) *Box*

Figure 6.1: **Object Symmetries in the Photorealistic Dataset.** (a) The *Can* object exhibits a continuous symmetry along the z-axis and two discrete flip symmetries around the x and y-axis respectively. (b) The *Bowl* object exhibits a single continuous symmetry along the z-axis. (c) The *Box* exhibits three discrete flip symmetries around each axis respectively.

restricted by the size of the tabletop and its distance from the camera. The range of the x and y-coordinates is $[-0.51\text{m}, 0.57\text{m}]$ and $[-0.33\text{m}, 0.42\text{m}]$ respectively. The depth values z range from 0.43 m to 0.76. Using the Isaac GYM framework [28], a training set of 15,000 photorealistic images and a validation set of 5,000 images were generated. Each image is rendered in Full-HD with an additional depth map. The segmentation information is additionally provided. The ground-truth poses as well as the object symmetries are available to us during evaluation. To evaluate the impact of object textures, we provide two datasets, the *Texture* dataset comprising objects rendered with material texture, and the *Uniform* dataset containing objects rendered in uniform red color.

For further understanding during the evaluation, the geometric symmetries of each object are visualized in Figure 6.1. The box object is compounded by three discrete flip symmetries around each axis respectively. The resulting four symmetric poses for each image are called the ground-truth symmetry. The *can* and *bowl* object exhibit a continuous symmetry around the z-axis. *Can* has two additional discrete flip symmetries around the x and y-axis. Both objects exhibit infinitely many symmetric poses for each image. Additionally, videos of the resulting symmetric orientations used as

ground-truth symmetries for each object during evaluation are provided.[1]

T-Less Dataset The T-Less dataset [19] consists of untextured industrial objects of varying sizes. For the evaluation of the IPPDF model on symmetric objects, we chose a subset of objects from the T-Less dataset that exhibit geometric symmetries. The objects are shown in Section 6.1. The objects are placed in isolation on a turntable with a black background. As the objects are turned, a camera mounted at different elevations captures the images. As a side-effect of this technique, the translational component does not express many variations. Therefore, we only evaluate the *Rotation-IPDF* model on the T-Less dataset. For each object, the training set contains 1296 images with depth information, ground-truth labels and camera intrinsic values.

To make the segmentation information available to us, each image is reproduced using the Stillleben framework [26]. Using the provided camera intrinsic values, a synthetic image of the object placed in the ground-truth pose is rendered. The segmentation information from the synthetic image yields an accurate segmentation image for the original image from the T-Less dataset. This thesis follows the evaluation strategy as implemented by Gilitschenski *et al.* [16] and Murphy *et al.* [5]. The training set of each object is split into a training and validation set. 1/6 of the training set is used for the validation set.

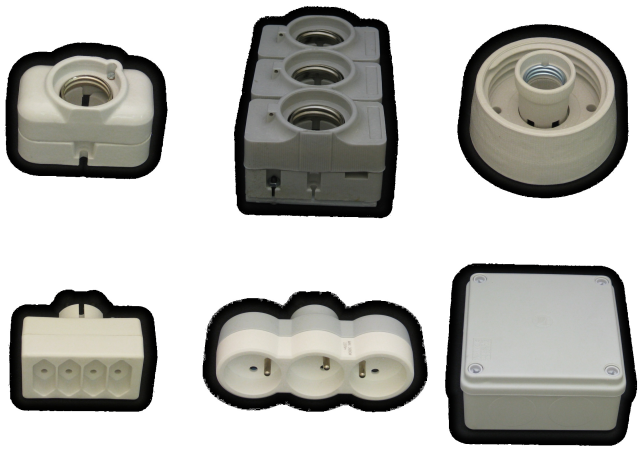


Figure 6.2: **T-Less Dataset Objects.** For evaluation, we use a subset of objects from the T-Less dataset that exhibit geometric symmetries.

[1] https://uni-bonn.sciebo.de/s/3moqVRuWwUcYk6S

6.2 Evaluation Metrics

The IPPDF model consists of the *Rotation-IPDF* and *Translation-IPDF* which estimate the rotation $\hat{\mathbf{R}} \in \mathbf{SO}(3)$ and translation $\hat{\mathbf{t}} \in \mathbb{R}^3$ separately. The images contained in the validation sets are annotated with a ground-truth poses $\mathbf{P}_{GT} = (\mathbf{R}_{GT}, \mathbf{t}_{GT})$. Additionally, the object symmetries are known during evaluation which yields the set of ground-truth symmetries $\{\mathbf{R}^i_{GT}\} \subseteq \mathbf{SO}(3)$. This set contains the correct rotations of an object with respect to the proper symmetries. In the case of continuous symmetries, we use a discretized set of 200 poses to make the computation of the evaluation metrics tractable. Note that the translation component is unique for all ground-truth symmetries. Symmetries are only expressed in the rotational component. Therefore, we represent the ground-truth symmetries as a set of rotations.

To evaluate the probability distribution generated by each model, we report the *log-likelihood* (LLH) separately for each model. The LLH metric measures the likelihood given to a single ground-truth rotation or translation:

$$\mathrm{LLH}(\mathbf{G}) = \mathbb{E}_{I \sim \mathcal{R}(I)} \, \mathbb{E}_{\mathbf{G} \sim \mathcal{P}_{\mathcal{GT}}(\mathbf{G}|\mathcal{I})} \log(\mathcal{P}(\mathbf{G}|\mathcal{I})), \tag{6.1}$$

where $\mathbf{G}$ is a placeholder for either the rotation $\mathbf{R}$ or translation $\mathbf{t}$. This gives valuable insight into the model's confidence in the estimated poses.

For the evaluation of the *ImplicitPosePDF* model on the *Single 6D Pose Estimation Problem*, we measure the precision of a single 6D pose estimation $\hat{\mathbf{P}}$. As a precision metric for the orientation estimation, the *mean absolute angular deviation* (MAAD) is reported. The MAAD metric is defined as:

$$\mathrm{MAAD}(\mathbf{R}) = \mathbb{E}_{\mathbf{R} \sim \mathcal{P}(\mathbf{R}|\mathcal{I})} \big[\min_{\mathbf{R}' \in \mathbf{R}_{GT}} d(R, R') \big], \tag{6.2}$$

with d being the geodesic distance between rotations. Separately, we measure the precision of the translation estimation $\hat{\mathbf{t}}$ using the L2 distance between the estimation and the ground-truth translation $\mathbf{t}_{GT}$.

To give further insight to the overall quality of the pose estimation, we examine the alignment of the object model placed in the estimated pose with the observed object model. $\mathcal{O}_{model}$ denotes a set of m 3D points corresponding to the object model. The average distance metric (ADD-S) as proposed by Corona *et al.* [12] evaluates the object model alignment on symmetric objects. Since the matching between points has ambiguities arising from the symmetries, the ADD-S metric utilizes the closest point distance between the estimated and observed object model. The ADD-S metric

is formally defined as:

$$\text{ADD-S} = \frac{1}{m} \sum_{\mathbf{x}_1 \in \mathcal{O}_{model}} \min_{\mathbf{x}_2 \in \mathcal{O}_{model}} ||(R_{GT}\mathbf{x}_1 + \mathbf{t}_{GT}) - (\hat{R}\mathbf{x}_1 + \hat{\mathbf{t}})||. \tag{6.3}$$

This metric compares against the complete set of proper symmetries by implicitly choosing the closest ground-truth pose to compare the alignment against. A pose is considered to be correct if the average distance is under a predefined threshold. We finally report the AUC metric for thresholds ranging from 0.001 to 0.02 meters.

To highlight the strength of our model to express object symmetries, we evaluate the performance on the *6D Pose Distribution Problem* using the Recall MAAD as a recall metric. The Recall MAAD expresses how accurately each rotation from $\{\mathbf{R}^i_{GT}\}$ is estimated in the orientation distribution. From the orientation distribution a set of predictions $\{\hat{\mathbf{R}}\}$ with a probability over $1e^{-3}$ is extracted. Formally, the Recall MAAD measures the mean absolute angular deviation for each ground-truth rotation in $\{\mathbf{R}^i_{GT}\}$ from the closest predicted rotation in $\{\hat{\mathbf{R}}\}$:

$$\text{RMAAD}(\mathbf{R}) = \mathbb{E}_{\mathbf{R} \sim \mathcal{P}(\mathbf{R}|\mathcal{I})}[\min_{\mathbf{R} \in \{\mathbf{R}|\mathcal{P}(\mathbf{R}|\mathcal{I}) \geq \Theta\}} d(\mathbf{R}, \mathbf{R}_{GT})]. \tag{6.4}$$

Furthermore, we assess the quality of the pseudo ground-truth labels produced by the *Automatic Pose Labeling Scheme* to examine the impact of the pseudo ground-truth labels on the training process of the *ImplicitPosePDF* model. As a precision metric, we report the MAAD metric as defined above for the rotation component and the L2 distance for the translation component. Additionally, we report the alignment quality of the APLS using the ADD-S metric. To motivate the variation within a set of pseudo ground-truth poses arising from object symmetries, we report the average *mean angular nearest neighbor distance* (MANN) within the pseudo ground-truth sets. This thesis defines the MANN metric for a single pseudo ground-truth set $\mathbf{R}_{PGT} = \{\mathbf{R}^i_{PGT}\}$ of size n as:

$$\text{MANN} = \frac{1}{n} \sum_{\mathbf{R}_i \in \mathbf{R}_{PGT}} \min_{\mathbf{R}_i \in \mathbf{R}_{PGT} \mathbf{R}_i} d(\mathbf{R}_i, \mathbf{R}_j), \tag{6.5}$$

where d is the geodesic distance between two rotations.

6.3 Implementation Details

The *ImplicitPosePDF* model and the *Automatic Pose Labeling Scheme* are implemented using the PyTorch library [29]. Additionally, the APLS uses the implementation of the Fast-Global-Registration algorithm from the Open3D library [30] and the

PyTorch3D [31] implementation of the Iterative-Closest-Point algorithm.

Automatic Pose Labeling Scheme. The pose labeling scheme runs offline once for each frame prior to training. The APLS runs at roughly 3 seconds for each frame with the render-and-compare stage being the heaviest workload in this process. In a post-processing step, close rotations within a pseudo ground-truth set are eliminated to ensure variation in the ground-truth pose. Rotations with an angular deviation under 5 degrees are substituted by a single rotation. This single rotation is generated by minimizing the Frobenius norm between the set of close rotation matrices through gradient descent.

ImplicitPosePDF. For the *Photorealistic* dataset the MLP of the *Rotation-IPDF* consists of three layers of 256 neurons while the *Translation-IPDF* uses a reduced version of two layers. The rotational component is encoded by three positional encoding terms and the translational component with two terms. Using the segmentation information, the background is masked out in each image. For the *Rotation-IPDF* model, a quadratic crop of 560×560 pixels around the object is extracted. The crop is then resized to the input size 224×224 of the feature extractor. To maintain information about the object's position, the *Translation-IPDF* model is fed the resized image without cropping. The images are then color-normalized and the gamma value is increased to suppress dominant visual features on the objects. The gradients are backpropagated through the backbone such that the feature extractor is included in the training process.

As a starting point the feature extractor is initialized with pre-trained weights for the ImageNet-1000 dataset [32]. The *Rotation-IPDF* model and the *Translation-IPDF* model are trained separately with separate feature extractors. For each object a separate model is trained. Using a combined model as done by Murphy *et al.* [5] did not perform properly with our adapted training procedure. The models are trained for 50 epochs with each epoch consisting of 200 iterations. In each iteration, the models are trained with a batch of 32 RGB images. We early stop the training after 30 epochs when the training progress starts stagnating. Both models are trained with an Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) and a base learning rate of 10^{-4} used for 20 iterations. The learning rate cosine decays to zero throughout the remaining iterations. To stabilize the training process of the neural network we additionally utilize gradient clipping and batch normalization. The *Rotation-IPDF* model is trained using a grid of cardinality $4,608$ and the *Translation-IPDF* is trained using a grid of cardinality $4,913$. The evaluation of the *Rotation-IPDF* model is completed with a grid of $294,912$ orientations and $97,336$ translation vectors for the *Translation-IPDF*.

6.4 Automatic Pose Labeling Scheme Results

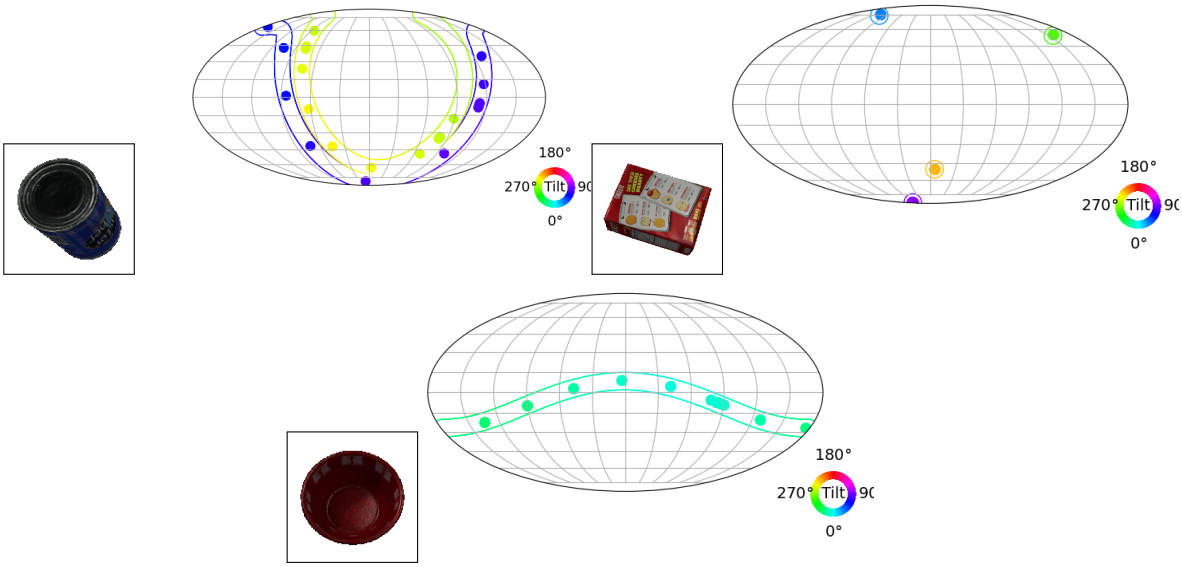


Figure 6.3: **Pseudo Ground-Truth Labels.** Visualization of the pseudo ground-truth labels compared against the ground-truth symmetries. The pseudo ground-truth labels within a single set are spread well across the ground-truth symmetries.

Object	Dataset	Number of Labels	MAAD[°]	MANN[°]	L2 [cm]
Can	*Texture*	15.59	1.44	31.81	0.33
	Uniform	15.71	1.41	31.73	0.33
Box	*Texture*	4.01	3.09	174.22	0.82
	Uniform	4.01	3.1	174.47	0.83
bowl	*Texture*	9.05	1.32	35.80	0.23
	Uniform	9.02	1.33	35.82	0.23
Average		9.565	1.95	80.64	0.46

Table 6.1: Evaluation of the Pseudo Ground-Truth Pose Labels

The *Automatic Pose Labeling Scheme* enables the *ImplicitPosePDF* model to be trained without ground-truth annotated data. As shown later in Section 6.6, the ability of the IPPDF model to represent the complete set of proper symmetries $\mathcal{M}_{geo}$ is highly dependent on the ground-truth labels. The objective of the APLS is to produce a pseudo ground-truth set for each frame that consists of accurate pose labels that are representative of object symmetries.

The evaluation of the APLS focuses on the *Photorealistic* dataset to highlight the strengths on highly symmetric objects in a realistic environment. The quantitative results for the *Photorealistic* dataset are presented in Table 6.1. The APLS achieves an overall precision of 1.95° for the rotational component and an error of $0.46cm$ in the translation component. To evaluate the overall quality pseudo ground-truth poses generate by the pose labeling scheme, we additionally report the ADD-S metric in Table 6.4. The average AUC of 90.57 shows that the APLS produces accurate pose labels. The slightly worse performance on the *Box* object is contributed to negligible geometric features such as small dents in the object model which cause further ambiguities in the point cloud registration process.

To highlight the ability of the pose labeling scheme to produce pseudo ground-truth sets representing the object symmetries, we additionally provide qualitative results in Figure 6.3. For discrete symmetries the pseudo ground-truth sets include an accurate pseudo ground-truth for each symmetric pose. With 4.01 pseudo ground-truth labels for each frame of the *Box* object, the 4 symmetric poses of the *Box* are well represented. The average distance between the rotations of 174.47° reflects the 180° rotation around one of the axes needed to transform between the symmetric rotations. The small error in both metrics can be contributed to a marginal number of false labelings that do not interfere with the learning ability of the IPPDF model.

The qualitative results indicate that the pseudo ground-truth labels are well spread across the continuous symmetries. Note that a close approximation is not required. A coarse sampling of the symmetric poses is sufficient to train the IPPDF model. On average, the pseudo ground-truth sets of the *Bowl* object consist of 9.04 pose labels with an average distance of 35.81 to the closest neighbor in the rotational component. Due to the additional discrete flip symmetries, the pseudo ground-truth sets for the *Can* are even larger with a similar distance to the closest neighbors.

The presented results show that the APLS is able to produce accurate pseudo ground-truth sets without prior knowledge of object symmetries. Discrete and continuous symmetries are well appreciated through multiple pose labels for each image. The size of each pseudo ground-truth set depends on the object symmetries and configuration chosen for the APLS. In Section 6.8 we will further highlight the effectiveness of the pseudo ground-truth labels in the training process.

6.5 Backbone Ablation Studies

The backbone feature extractor produces a visual feature descriptor used as an input for the MLP. State-of-the-art feature extractors use CNNs to produce visual features in a hierarchical order. The first layers extract low-level features at a high resolution,

Table 6.2: Comparison of different models as feature extractor.

	Model	Metric	ResNet [33]		ConvNeXt [23]				
			18	50	Tiny	Tiny-1	Tiny-2	Small	Base
FLOPs[1]			1.8G	3.8G	4.5G	-	-	8.7G	15.4G
can	Rotation	LLH	3.76	3.86	3.99	3.83	1.09	3.77	3.72
		MAAD [°]	2.24	2.46	2.44	2.52	21.5	2.55	2.79
	Translation	LLH	9.19	9.31	9.19	9.34	9.13	9.56	9.43
		L2 [cm]	0.71	0.79	0.48	0.52	0.53	0.6	0.57
box	Rotation	LLH	5.75	5.98	6.09	5.75	3.02	5.996	5.649
		MAAD [°]	5.2	4.73	5.24	5.97	34.27	4.28	8.609
	Translation	LLH	8.86	8.97	9.49	9.03	8.79	9.29	9.24
		L2 [cm]	1.07	1.1	0.88	1.04	1.02	0.94	0.99
bowl	Rotation	LLH	3.3	3.17	3.85	3.86	1.05	3.21	3.24
		MAAD [°]	2.96	3.17	2.47	2.7	22.84	2.88	3.22
	Translation	LLH	9.1	9.25	9.19	9.52	9.45	9.56	9.6
		L2 [cm]	0.58	0.63	0.49	0.53	0.53	0.67	0.7

[1]FLOP values are taken from [33] and [23].

whereas the last layers produced high-level features at a low resolution. Different model architectures produce different visual features that typically generalize well among different datasets and objects. To find the best-suited feature extractor for the *ImplicitPosePDF* model, we carried out numerous experiments with a variety of models.

The ResNet-50 model [33] was originally chosen by Murphy *et al.* [5]. Additionally, we experimented with the ResNet-18 and the ConvNeXt models [23] in different configurations. The last fully connected layer used for classification is removed from the feature extractors such that the visual feature descriptors are fed directly to the MLP. The ResNet-18 produces a feature vector of size 512 and the ResNet-50 outputs a feature vector of size 2048. In the tiny and small configuration, the ConvNeXt models produce a feature vector of size 768. The ConvNeXt-Base model generates a feature vector of size 1024. Table 6.2 presents the quantitative results of the *Rotation-IPDF* model and the *Translation-IPDF* model using different backbones. Overall, using different feature extractors results in good performances across the board. The features extracted seem to generalize well to our task.

Furthermore, we examined the impact of different features taken from different layers of the feature extractor. As tasks like object classification and object detection profit from high-level features, other tasks such as semantic segmentation benefit from low-level features. To determine the best-suited features for our task, we ad-

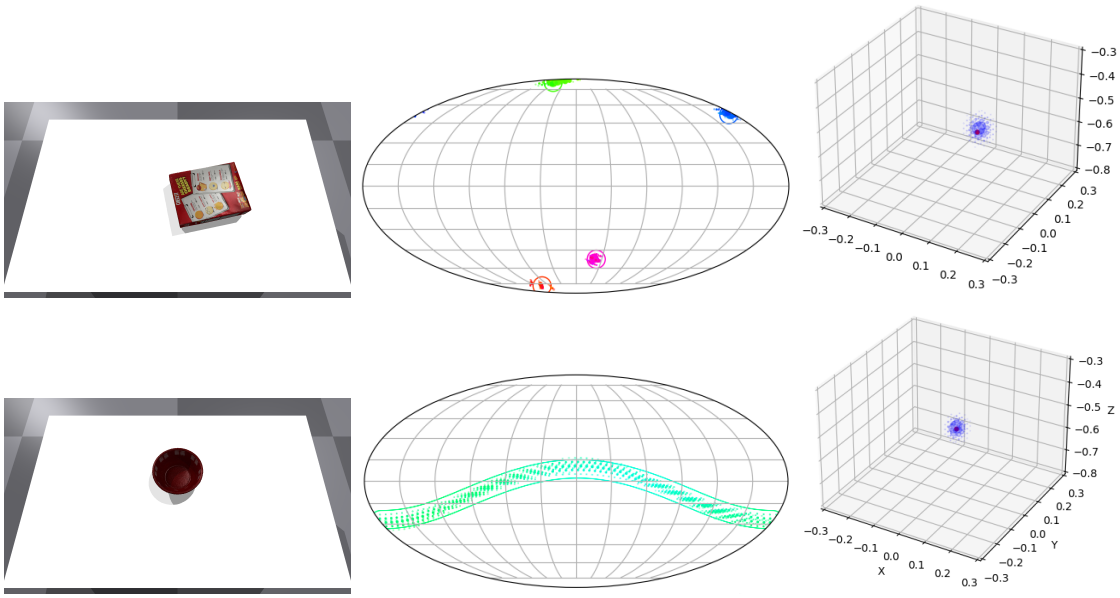
ditionally introduce the ConvNeXt-Tiny-1 model with the last average-pooling layer removed and the ConvNeXt-Tiny-2 model with the last average-pooling layer and ConvNeXt block removed. The features extracted by the ConvNeXt-Tiny-1 model are of shape $768 \times 7 \times 7$ and the features extracted by the ConvNeXt-Tiny-2 model of shape $384 \times 14 \times 14$. The quantitative results in Table 6.2 show that the features from the ConvNeXt-Tiny-2 model seem to not provide useful information about an object's orientation but give sufficient information about the translation. Taking low-level features from earlier layers seems to not improve our model's performance. We conclude that the final features extracted from the ConvNeXt-Tiny model are well-suited for our task.

With the goal of estimating orientation distributions, we decided in favor of the backbone yielding the best results of the *Rotation-IPDF* in terms of the log-likelihood. The *Translation-IPDF* model's objective is to predict a single translation as precisely as possible. Thus, we chose the backbone which produces the smallest error in the translation estimation. Ultimately, we decided in favor of the ConvNeXt-Tiny model without additional layers removed as our backbone feature extractor. The ConvNeXt-Tiny model outperforms the other feature extractors by a small margin for the *Rotation-IPDF* model and the *Translation-IPDF* model. The visual features extracted by this model seem to be the best suited for our task. Note that the feature extractor accounts for the majority of the computational effort of the *ImplicitPosePDF* model. With about three times fewer floating point operations, the ResNet-18 still poses an adequate alternative to speed up training and inference.

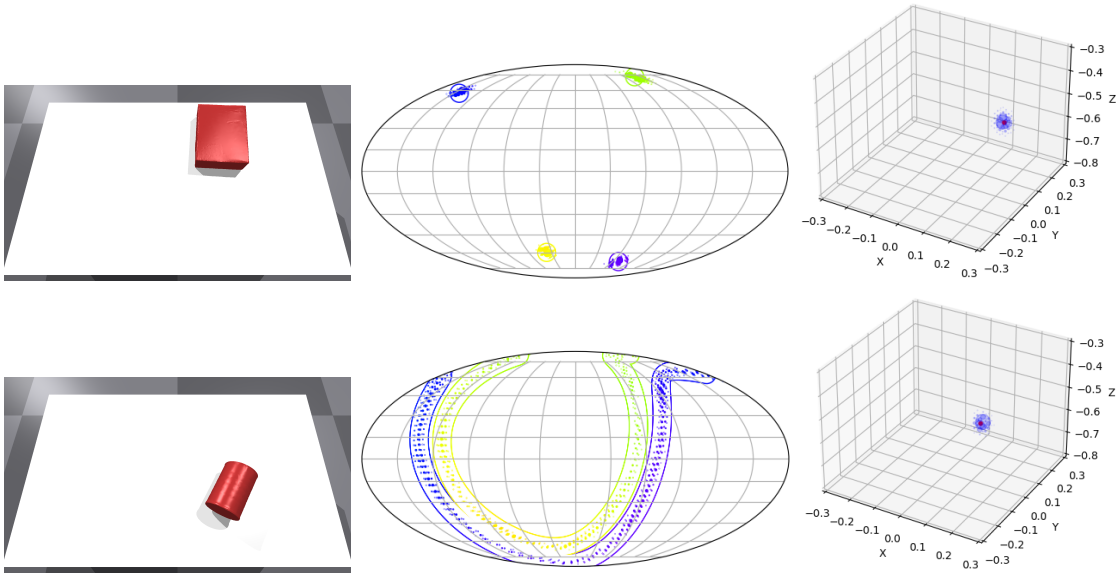
6.6 6D Pose Distribution Estimation

In order to evaluate the ability of the *ImplicitPosePDF* model to capture arbitrary object symmetries, we examine the performance of the model on the symmetric objects from the *Photorealistic* dataset. Without ground-truth annotations or prior knowledge of object symmetries, the training is fully automated through the *Automatic Pose Labeling Scheme*.

To highlight the advantages of the IPPDF model trained with pseudo ground-truth labels over the training procedure proposed by Murphy *et al.* [5], we additionally trained the IPPDF model observing only a single ground-truth label for each frame which is referred to as the *single* model. The qualitative results presented in Figure 6.5 demonstrate that this model is not able to capture the proper symmetries. The pose distributions degenerate to a single pose estimation, even for uniformly colored objects. Table 6.3 presents the quantitative results of this model. The high average Recall MAAD of 121.2° and LLH score of 6.5 emphasize that the model estimates a



(a) *Texture* dataset.



(b) *Uniform* dataset.

Figure 6.4: **Pose Distributions on the Photorealistic Dataset.** Pose distributions predicted by the *ImplicitPosePDF* model on the *Texture* dataset and *Uniform* dataset. The visualizations are generated using 294,912 orientation and 97,336 translation hypotheses.

single pose with high confidence disregarding any object symmetries. We conclude that the model is not able to generalize from the visual features to predict complete pose distributions.

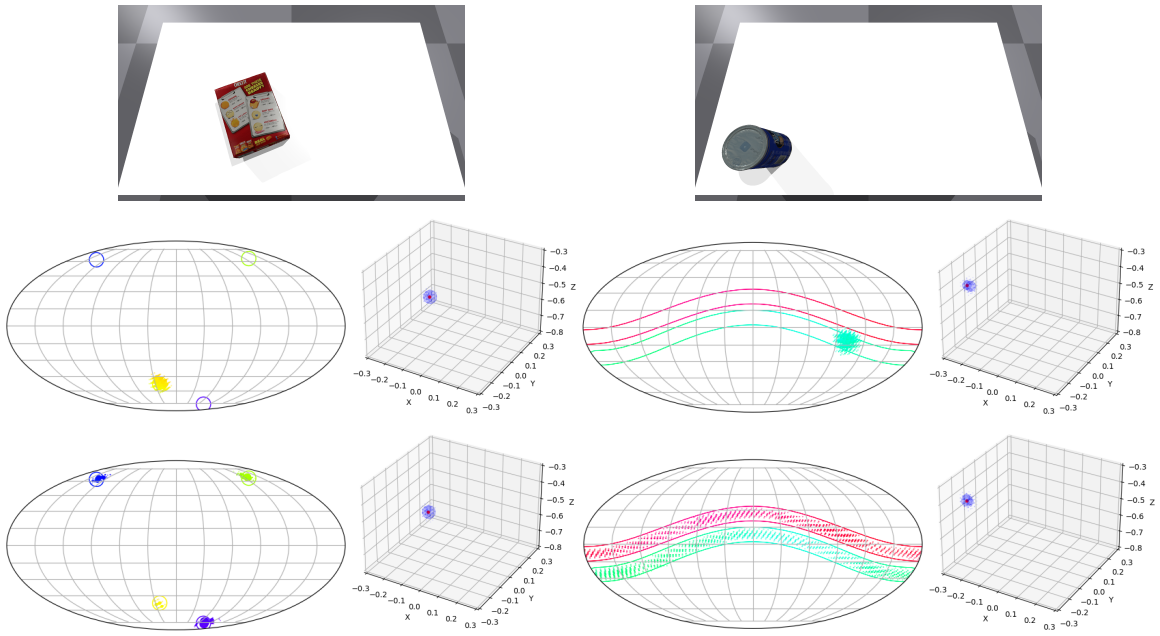


Figure 6.5: **Pose Distributions using Different Ground-Truth Labels**. **Upper Row:** Input image for the IPPDF model. **Middle Row:** Pose distribution predicted by the IPPDF model trained with a single ground-truth label for each frame. **Lower Row:** Pose distribution predicted by the IPPDF model trained with the pseudo ground-truth labels.

In contrast, the pose distribution predicted by the models trained with pseudo ground-truth labels shown in Figure 6.5 are able to represent the proper symmetries. The qualitative results in Figure 6.4 show that continuous symmetries of the *Can* and *Bowl*, and the discrete symmetries of the *Box* are well appreciated. Visual features arising from object textures do not interfere with the quality of the pose distributions. The IPPDF model is able to construct pose distributions invariant to visual features. A low average Recall MAAD of 2.27° emphasizes the ability of our model to accurately approximate the proper symmetries. The unique translation vector is estimated by a uni-modal distribution that propagates high probabilities for a small number of translation queries close to the ground-truth translation. In absence of occlusion, the *Rotation-IPDF* achieves a LLH score of 4.64. With more predicted orientations included in the distribution the confidence in a single orientation declines. Consequently, the LLH score of the *single* model, as well as the LLH score for the *Box* that only exhibits four discrete symmetries, is higher. This effect is also visible in

the *Translation-IPDF* model. The high LLH score of 8.84 of the *Translation-IPDF* model reflects the high confidence in a single translation estimation. On average the *Translation-IPDF* model predicts a translation with an error of 0.58 cm.

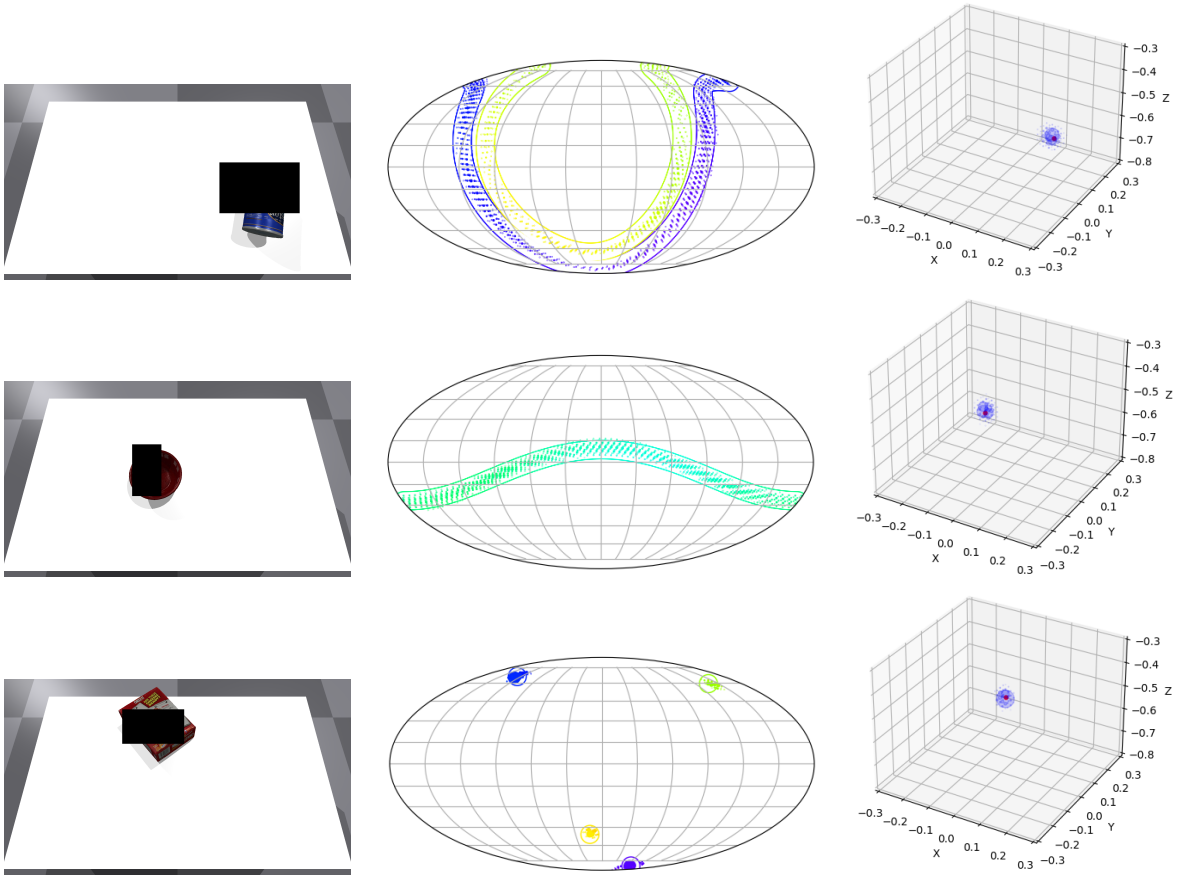


Figure 6.6: **Pose Distributions in the Presence of Occlusion.** The IPPDF model predicts the complete pose distributions unaffected by the occlusion.

The difficulty of computer vision tasks is intensified by more complex scenes. Real-world scenes typically include multiple objects or other disturbances that partially occlude the object of interest. Occlusion excludes vital information about the object's pose in the object. To further evaluate the performance of our model in the presence of occlusion, we add occlusion to 80% of the images. 10% to 50% of the cropped images are occluded. The quantitative results in Table 6.3, show that the presence of occlusion does not compromise the performance of the IPPDF model compared to the non-occluded scenes. The LLH and Recall MAAD show that even in heavily occluded scenes the IPPDF model is able to predict the object symmetries. The qualitative results in Figure 6.6 show that the IPPDF model predicts the complete pose distributions. Interestingly, the Recall MAAD of the *single* model is significantly smaller

Table 6.3: Results of models trained on different ground-truth Labels.

	GT	Without Occlusion					With Occlusion				
		LLH (Rot.) ↑	LLH (Trans.) ↑	MAAD [°] ↓	Recall MAAD [°] ↓	L2 [cm] ↓	LLH (Rot.) ↑	LLH (Trans.) ↑	MAAD [°] ↓	Recall MAAD [°] ↓	L2 [cm] ↑
can	Single	6.46	9.17	3.11	120.7	0.45	5.347	8.77	4.9	97.99	0.77
	Analytic	3.78	9.22	2.5	1.87	0.46	3.47	8.81	3.87	1.95	0.77
	Pseudo	3.99	9.19	2.44	1.86	0.48	3.55	8.83	4.16	1.99	0.74
box	Single	6.56	9.59	5.35	123.61	0.61	6.76	9.35	5.6	119.63	0.99
	Analytic	5.78	9.53	5.2	2.17	0.62	5.75	9.68	4.94	2.08	0.97
	Pseudo	6.09	9.49	5.24	2.17	0.88	5.68	9.56	5.66	1.95	0.88
bowl	Single	6.49	9.19	2.77	119.28	0.45	5.36	9.28	4.99	97.12	0.9
	Analytic	3.95	9.23	2.43	2.12	0.46	3.93	9.27	5.99	1.88	0.89
	Pseudo	3.85	9.19	2.47	2.05	0.49	3.76	9.21	6.89	2.54	0.97
Pseudo Avg.		4.64	9.29	3.38	2.03	0.62	4.33	9.2	5.09	2.16	0.86

↑ indicates higher value better, whereas ↓ indicates lower value better.

in the presence of occlusion. This can be attributed to the additional uncertainty introduced through the occlusion.

To further demonstrate the expressiveness of object symmetries, we additionally provide videos of the orientation distributions predicted by our model on a synthetic dataset rendered using the Stillleben framework [26].[2]

Overall, our model expresses the ability to construct complete pose distributions that are representative of object symmetries invariant to object textures. The *Rotation-IPDF* model implicitly learns to represent the proper symmetries while the *Translation-IPDF* learns a uni-modal distribution that precisely estimates a unique translation vector. Even in more complex scenes that include occlusion, the IPPDF model learns the distributions well for the rotational component, as well as the translational component. Combined with the APLS, the *ImplicitPosePDF* is able to capture arbitrary symmetries in realistic scenes without the supervision of object poses or symmetries.

To further demonstrate the capabilities of the *ImplicitPosePDF*

6.7 Single 6D Pose Estimation

Besides the ability to construct pose distribution, we want to highlight the advantages of implicitly modeling symmetries in the *Single 6D Pose Estimation Problem*. While state-of-the-art methods [3] [2] [1] only define symmetries during training, our model

[2] https://uni-bonn.sciebo.de/s/VTgZYfUAQYIxLVe

Object	Pseudo Ground-Truth	ImplicitPosePDF				
		Texture	Uniform	Single	Analytic	Occlusion
Can	93.40	91.61	91.61	91.63	91.42	91.94
Box	81.94	79.15	78.32	83.81	80.87	76.12
Bowl	96.37	91.43	90.66	93.93	93.57	85.63
Average	90.57	87.4	86.86	89.79	88.62	84.56

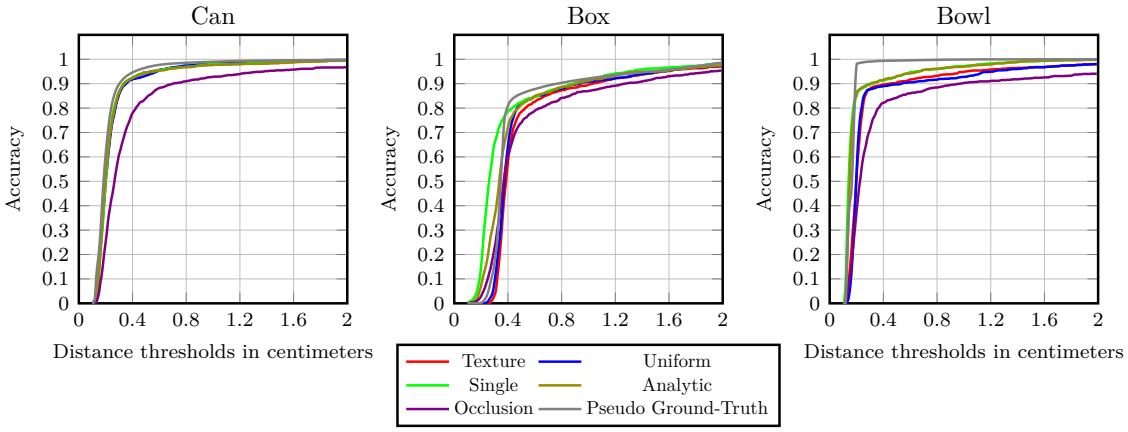


Table 6.4: Area under the accuracy-threshold (AUC) of the ADD-S metric on the *Photorealistic* dataset.

utilizes the implicit representation of the object symmetries during inference as well.

To assess the quality of the predicted rotation and translation separately, we report the estimation error of the orientation and translation estimation in Table 6.3. Compared to popular datasets used for pose estimation tasks such as the YCB-Video dataset [1] or the LineMOD dataset [34], the translation range in our dataset is limited. Thus, it is not surprising that the *Translation-IPDF* model is able to estimate a single translation with a small error of 0.58 cm. The estimation of the orientation on the other hand still poses a difficult challenge for the *Rotation-IPDF*. With a broad variety of orientations presented in our dataset, the *Rotation-IPDF* model is able to predict an orientation with an average error of 3.2°.

We further evaluate the accuracy of the pose using the area under the accuracy-threshold (AUC) of the ADD-S metric. The ADD-S metric is more sensitive to errors in the translation than to errors in the orientation estimation. Therefore, to compensate for the limited translation range and the resulting small estimation error, we use smaller thresholds than the state-of-the-art methods. The thresholds used for evaluation range from 0.1 cm to 2 cm. The results of the ADD-S evaluation are presented in Table 6.4. Our model achieves an average AUC of 87.4 on the *Texture* dataset and an average AUC of 86.86 on the *Uniform* dataset. The model tends to perform worse on the *Box* object. This can be attributed to the learned orientation distribution for

discrete objects. The distribution consists of a few orientations with a high probability as shown in Figure 6.4 which does not allow the computation of useful gradients to optimize the orientation such that the gradient ascent method gets stuck in local optima resulting in less accurate predicitons. Overall the model is still able to predict accurate poses on all objects in our dataset invariant to object textures. Compared to the *single* model, we achieve similar results, while being able to produce complete pose distributions.

While the occlusion does not affect the pose distributions predicted by our model, it still seems to compromise the accuracy of predicted poses to a small extent. The quantitative results in Table 6.3 show that occlusion increases the translation error by 0.24 cm and the angular error of the orientation estimation by 1.71°. Especially on objects with a high degree of symmetry, such as the *Can* and *Bowl*, the *Rotation-IPDF* model performs worse in presence of occlusion. Since the pose distributions are unaffected, we conclude that the occlusion mainly affects the gradients calculated during pose estimation. With faulty gradients and a high number of local optima, resulting from a high degree of symmetry, in the optimization domain $\mathbf{SO}(3)$, the gradient ascent method gets stuck in local optima and is unable to converge to an accurate pose estimation. Still, our model achieves an average AUC of 84.56 for the images affected by occlusion, showing its capabilities in more complex scenes.

Even though our model is designed to estimate pose distributions, it is able to predict accurate poses even in scenes including occlusion. Defining symmetries during inference seems helpful for the 6D pose estimation task for symmetric objects.

6.8 Effectiveness of Pseudo Ground-Truth Labels

Using pseudo ground-truth labels during training instead of single ground-truth labels has shown superior results with respect to the estimated pose distribution. To further highlight the effectiveness of the pseudo ground-truth labels, we examine a third model trained directly with the ground-truth symmetries of the object, called the *analytic* model. Knowing a ground-truth pose and the object symmetries, we define a set of analytically derived ground-truth labels. The quantitative results in Table 6.3 show that our model using pseudo ground-truth labels yields similar results as the analytic model. The differences in the quantitative results for the *Rotation-IPDF* and *Translation-IPDF* are marginal. The results also confirm that the drop in the LLH metric compared to the *single* model is linked to the fact that our model estimates a complete pose distribution and not a result of the utilization of the pseudo ground-truth labels.

The quantitative results from the single 6D pose estimation task presented in Ta-

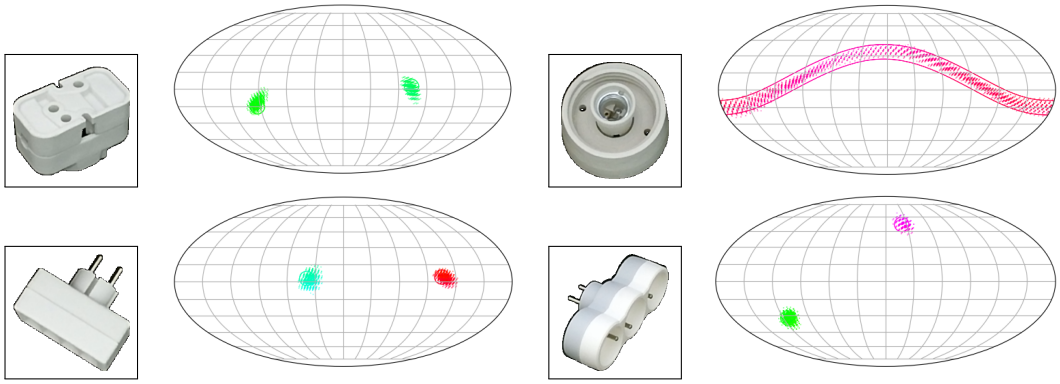


Figure 6.7: **Pose Distributions on the T-Less Dataset.** Pose distributions predicted by the IPPDF model trained with pseudo ground-truth labels.

ble 6.4 show that the pseudo ground-truth labels do not compromise the ability to predict accurate poses. The analytic model performs similar to our model. Moreover, the accuracy of the pseudo ground-truth labels is higher than the accuracy achieved by the analytic model. Thus, we can conclude that the precision of the *ImplicitPosePDF* model is not limited by the precision of the pseudo ground-truth labels but rather by the capacity of the model itself. The small error in the pseudo ground-truth labels reported in Section 6.4 does not affect the precision of the estimated poses.

In addition to the results of Section 6.4, we conclude that the pseudo ground-truth sets are sufficient to train the IPPDF. The symmetries appear to be well appreciated within the sets as the performance of the model trained with pseudo ground-truth labels is the same as the model trained with the ground-truth symmetries. The pseudo ground-truth labels pose an efficient alternative to using analytically derived ground-truth labels. In contrast to producing ground-truth labels analytically, the *Automatic Pose Labeling Scheme* does not need prior knowledge about an object's pose or symmetries and does not add any restrictions on the dataset acquisition process. The pose labeling scheme fully automates the training of the *ImplicitPosePDF* model and enables the learning of pose distributions of symmetric objects without ground-truth annotations or explicit symmetry annotations.

6.9 T-Less Evaluation

The T-Less dataset poses a difficult challenge to evaluate the *ImplicitPosePDF* model on real-world objects. With a relatively small training set of $1,000$ images, the qualitative results in Figure 6.7 show that the *Rotation-IPDF* model is able to express an orientation distribution expressive of the object symmetries. Each of the symmetric orientations is well approximated. To further investigate the performance of our

Table 6.5: Results of the *Rotation-IPDF* model on the T-Less dataset.

Method	LLH ↑	MAAD[°] ↓
Deng *et al.* [15]	5.3	23.1
Gilitschenski *et al.* [16]	6.9	3.4
Prokudin *et al.* [35]	8.8	34.3
Murphy *et al.* [5]	9.8	4.1
Analytic	6.2	1.7
Ours	6.09	3.22

model we state the quantitative results in Table 6.5 compared to the state-of-the-art methods. Overall, the *Rotation-IPDF* model achieves a LLH score of 6.09. Object symmetries introduce uncertainty in the orientations. Our subset of objects used for evaluation consists strictly of symmetric objects. Thus, our model performs worse in terms of the LLH metric but this does not hamper the ability of the *Rotation-IPDF* to accurately reconstruct orientation distributions. The MAAD score of 3.22° shows that the model predicts orientations with high precision. The results of the *Rotation-IPDF* model training with analytically derived ground-truth labels further show that the pseudo ground-truth labels are effective to train the *Rotation-IPDF* model on the T-Less dataset. We observed only a marginal difference in the LLH metric and a difference of 1.52° in the MAAD metric. This error in the orientation estimation can be attributed to a small error in the pseudo ground-truth labels but this impacts the model's performance only to a small extent. The small error in the pseudo ground-truth labels does not hamper the learning ability of the *Rotation-IPDF* model. Thus, using pseudo ground-truth labels poses an efficient alternative to analytically derived ground-truth labels.

7 Conclusion

This thesis presents the *ImplicitPosePDF* model which extends the approach by Murphy *et al.* [5] from the rotation manifold $\mathbf{SO}(3)$ to construct pose distributions over $\mathbf{SE}(3)$. The model is able to be trained without any supervision of the poses and symmetries through the usage of pseudo ground-truth labels. The *Automatic Pose Labeling Scheme*, producing the pseudo ground-truth labels, is the second contribution presented in this thesis.

The strengths of the *ImplicitPosePDF* were highlighted on a variety of symmetric objects from the Photorealistic dataset and the T-Less dataset. The model is able to construct complete pose distributions, capturing arbitrary geometric symmetries invariant to object textures. Eliminating the high sensitivity to object textures enables the IPPDF model to be employed in real-world scenarios. Additionally, we have shown the advantages of implicitly modeling symmetries for the single 6D pose estimation task. Even though the *ImplicitPosePDF* model is designed to estimate pose distributions, it is able to predict object poses accurately.

Through the APLS the IPPDF model can be trained without the supervision of object poses and object symmetries. Starting off with an unlabeled dataset, the pose labeling scheme is able to produce sets of accurate ground-truth poses representative of object symmetries. The pseudo ground-truth labels proved to be an efficient substitution for analytically derived ground-truth labels in the training process of our model. The APLS is able to fully automate the training pipeline of our model and eliminates the necessity of ground-truth labeled data.

The IPPDF model combined with the APLS has shown the capability of being employed in more application-oriented tasks. Future work can include incorporating our model into a grasp planning task to test its abilities in a robotic application. Moreover, there is room to improve the efficiency of the pose labeling scheme. A batch-wise computation of the pseudo ground-truth labels, further downsampling of point clouds and reduction of image sizes could improve the running time of the APLS.

List of Figures

1.1 Predicted Distributions of the Implicit-PDF Model 2

2.1 6D Pose Estimation Problem 4
2.2 *Can* Object Symmetries 5

4.1 *ImplicitPosePDF* Model 11
4.2 Equi-Volumetric Sampling of **SO**(3) 13
4.3 Implicit-PDF Model 14
4.4 Visualization Technique for Pose Distributions 16

5.1 Automatic Pose Labeling Scheme 18
5.2 Pseudo Ground-Truth Generation Process 21

6.1 Object Symmetries in the *Photorealistic* Dataset 24
6.2 T-Less Dataset Objects 25
6.3 Pseudo Ground-Truth Labels 29
6.4 Pose Distributions on the *Photorealistic* Dataset 33
6.5 Pose Distributions using Different Ground-Truth Labels 34
6.6 Pose Distributions in the Presence of Occlusion 35
6.7 Pose Distributions on the T-Less Dataset 39

List of Tables

6.1 Evaluation of the Pseudo Ground-Truth Pose Labels 29
6.2 Comparison of different models as feature extractor. 31
6.3 Quantitative Results using Different Ground-Truth 36
6.4 AUC of the ADD-S metric on the YCB-Video dataset 37
6.5 Results of the *Rotation-IPDF* model on the T-Less dataset. 40

Bibliography

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

[2] A. Amini, A. S. Periyasamy, and S. Behnke, "YOLOPose: Transformer-based multi-object 6D pose estimation using keypoint regression," in *International Conference on Intelligent Autonomous Systems (IAS)*, 2022.

[3] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[4] Y. Xu, K.-Y. Lin, G. Zhang, X. Wang, and H. Li, "RNNPose: Recurrent 6-DoF object pose refinement with robust correspondence field estimation and pose optimization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 880–14 890.

[5] K. A. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia, "Implicit-PDF: Non-parametric representation of probability distributions on the rotation manifold," in *International Conference on Machine Learning (ICML)*, PMLR, 2021, pp. 7882–7893.

[6] A. S. Periyasamy, L. Denninger, and S. Behnke, "Learning implicit probability distribution functions for symmetric orientation estimation from rgb images without pose labels," 2022.

[7] R. Bregier, F. Devernay, L. Leyrit, and J. L. Crowley, "Defining the pose of any 3D rigid object and an associated distance," *International Journal of Computer Vision (IJCV)*, vol. 126, pp. 571–596, 2018.

[8] G. Pitteri, M. Ramamonjisoa, S. Ilic, and V. Lepetit, "On object symmetries and 6D pose estimation from images," in *International Conference on 3D Vision (3DV)*, IEEE, 2019.

[9] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3828–3836.

[10] C. Esteves, A. Sud, Z. Luo, K. Daniilidis, and A. Makadia, "Cross-domain 3D equivariant image embeddings," in *International Conference on Machine Learning (ICML)*, PMLR, 2019, pp. 1812–1822.

[11] A. Saxena, J. Driemeyer, and A. Y. Ng, "Learning 3D object orientation from images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 794–800.

[12] E. Corona, K. Kundu, and S. Fidler, "Pose estimation for objects with rotational symmetry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[13] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *European Conference on Computer Vision (ECCV)*, 2018.

[14] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, and F. Tombari, "Explaining the ambiguity of object detection and 6D pose from visual data," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019.

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

[16] I. Gilitschenski, R. Sahoo, W. Schwarting, A. Amini, S. Karaman, and D. Rus, "Deep orientation uncertainty learning based on Bingham loss," in *International Conference on Learning Representations (ICLR)*, 2019.

[17] B. Okorn, M. Xu, M. Hebert, and D. Held, "Learning orientation distributions for object pose estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[18] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: an RGB-D dataset for 6D pose estimation of texture-less objects," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2017, pp. 880–888.

[20] K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann, "HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *The Astrophysical Journal*, p. 759, 2005.

[21] A. Yershova, S. Jain, S. Lavalle, and J. Mitchell, "Generating uniform incremental grids on SO(3) using the hopf fibration," *The International journal of robotics research*, 2010.

[22] D. W. Lyons, "An elementary introduction to the Hopf Fibration," *Mathematics Magazine*, 2003.

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2022.

[24] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 766–782.

[25] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *International Conference on Robotics and Automation (ICRA)*, IEEE, 2009, pp. 3212–3217.

[26] M. Schwarz and S. Behnke, "Stillleben: Realistic scene synthesis for deep learning in robotics," in *IEEE Internation Conference on Robotics and Automation (ICRA)*, 2020.

[27] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[28] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU based physics simulation for robot learning," in *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: an imperative style, high-performance deep learning library," 2019.

[30] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[31] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3D deep learning with PyTorch3D," in *European Conference on Computer Vision (ECCV)*, 2020.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[34] S. Hinterstoißer, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*, 2012.

[35] S. Prokudin, P. Gehler, and S. Nowozin, "Deep directional statistics: Pose estimation with uncertainty quantification," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 534–551.