TU WIEN Informatics

Transparent Object Pose Refinement Using Differentiable Rendering

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Logic and Computation

by

Negar Layegh Khavidaki
Registration Number 12031846

to the Faculty of Informatics

at the TU Wien

Advisor: Markus Vincze, Univ.Prof. Dipl.-Ing. Dr.techn.
Assistance: Jean-Baptiste Weibel, Dr.techn. M.Sc.
Dominik Bauer, Dr.techn. Dipl.-Ing.

Vienna, 30th January, 2024

Negar Layegh Khavidaki

Markus Vincze

Acknowledgements

I am grateful to Prof. Markus Vincze for the valuable opportunity to contribute to the V4R lab and his consistent support throughout my academic journey. Under the guidance of Dominik Bauer and Jean-Baptiste Weibel, I have had the privilege of obtaining knowledge and experience. Their mentorship and dedication made this thesis possible. I am delighted to be part of a lab with outstanding colleagues who readily offer assistance and engage in pleasant conversations. I want to express my gratitude to the Tracebot group for allowing me to participate in such an exciting project.

Special appreciation goes to my friends—Clara, Konstantin, Maren, Mareike, and Sabrina—for their support and constructive feedback, which significantly influenced the development of my thesis. None of this would have been achievable without the unconditional support of my family: my parents, Bita and Morteza, and my sisters, Naghmeh and Yas, who stood by me kilometers away.

A heartfelt thank you to my partner, Sergio, for being so supportive and soothing throughout my studies. His presence truly helped me get through it all.

I feel lucky to have wonderful people in my life.

Kurzfassung

Vom Beginn unseres Tages mit dem Trinken eines Glases Wasser, dem Packen unserer Lunchbox oder dem Durchqueren einer Glastür, transparente Materialien sind allgegenwärtig in unserem täglichen Leben. Die Identifizierung dieser Objekte kann jedoch komplex sein, insbesondere unter unterschiedlichen Licht- oder Szenenbedingungen, und noch herausfordernder für Roboter und deren autonome Wahrnehmungssysteme.

Die Herausforderung liegt in der Transparenz solcher Objekte. Diese Eigenschaft führt oft zu äußerst fehlerbehafteten oder fehlenden Tiefeninformationen, was traditionelle tiefenbasierte Ansätze unzureichend macht. Andererseits stoßen texturbasierte Methoden auf Schwierigkeiten aufgrund von Reflexionen und der Notwendigkeit einer präzisen Hintergrundmodellierung.

Wir addressieren diese Probleme mit einer Differentiable-Rendering Pipeline, um die Position und Ausrichtung transparenter Objekte aus RGB-Bildern innerhalb einer Szene robuster zu erkennen. Basierend auf 3D Modellen erstellt unsere Differentiable-Rendering Pipeline eine 3D-Repräsentation einer Szene zu erstellen. Der Vorteil dieses Ansatzes besteht darin, dass direkt im Bildraum unter Verwendung der Parameter der 3D-Szene optimiert wird, basierend auf der Position und dem Aussehen von Objekten. Darüber hinaus wird der Optimierer durch die Verwendung der 3D-Repräsentation angeleitet, (deleted) realistische räumliche Anordnungen sicherzustellen (z.B. das Verhindern, dass ein Objekt in einem anderen liegt). Die Kombination von 3D-Szenen und RGB-Bildern, zusammen mit der Optimierung im Bildraum, erhöht die Flexibilität unseres Ansatzes und ermöglicht Pose Refinement für transparente Objekte ohne zeit- oder datenintensivem Training.

Im Rahmen dieser Arbeit werden drei Szenen aufgezeichnet und annotiert, um verschiedene Szenarien zu evaluieren. Diese Szenen dienen als wertvolle Daten zur Bewertung und Validierung des vorgeschlagenen Ansatzes. Die Flexibilität und Effektivität unserer Pipeline machen sie insbesondere in der Robotik anwendbar, in denen die Bestimmung der genauen Position von transparenten Objekte entscheidend für viele Anwendungen ist.

Abstract

From starting our day by drinking a glass of water, packing our lunch box, or passing a glass door, transparent materials are prevalent throughout our daily lives. However, identifying these objects can be complex, especially under varying lighting or scene conditions, and even more challenging for robots or any autonomous perception system. The challenge lies in the transparency of such objects. This property influences their depth information, making traditional depth-based approaches insufficient; in such methods, estimating the missing depth information is also a complex problem. Additionally, texture-based methods face difficulties due to reflections and the need for precise background modelling.

To overcome these issues, a pipeline is proposed that uses differentiable rendering to refine the position and orientation of transparent objects from Red-Green-Blue (RGB) images within a scene. The pipeline leverages Three Dimensional (3D) models and a differentiable renderer to create a 3D representation of the scene of interest. The advantage of this approach is that it optimizes directly in the image space using the parameters of the 3D scene, such as the position and appearance of objects. Furthermore, by utilizing the 3D representation, the optimizer is guided to avoid unnatural collisions between objects, ensuring realistic spatial arrangements (e.g., preventing one object from being inside another). The combination of 3D scenes and RGB images, along with optimization within the image space, enhances the flexibility of our approach, liberating it from the need for extensive datasets and predetermined object shapes.

Three scenes containing transparent canisters are recorded and annotated to support this thesis. These scenes serve as valuable data for evaluating and validating the proposed approach. The flexibility and effectiveness of the proposed pipeline make it applicable in various domains, including robotics and autonomous systems, where the accurate position of transparent objects is crucial.

Contents

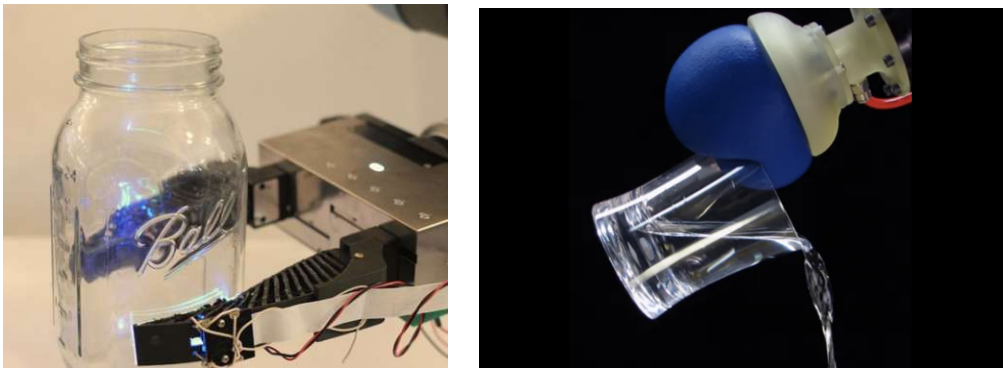
CHAPTER 1

Introduction

Transparent objects are an integral part of our surroundings, ranging from glassware and bottles to medical equipment like syringes and catheters. They are frequently encountered in both laboratory and household settings, as their transparency offers valuable insights into their contents. However, this very characteristic poses challenges in their perception, as it introduces complexities associated with the reflection and refraction of their surrounding environment [Bun21, LCL20].

In our daily interactions, we engage with objects by grasping them, holding them in our hands, and using them for various tasks such as drinking or tidying up. In this procedure, perceiving an object is the initial step, and our perception may vary based on factors like the object's surface material, lighting conditions, and other environmental influences. Over time, we learn and adapt our perception based on these parameters, enhancing our ability to recognize objects more accurately in the future. The next step is deciding on an optimal angle, which allows for the most efficient manipulation, considering factors such as stability, ease of control, and the object's intended use. Similar to object recognition, our skill in grasping objects is acquired through experience, adjusting our approach based on previous attempts and outcomes, and learning the optimal angles and methods for manipulation.

In contrast, the process for robots differs. Object perception for robots is a challenging task, especially with environmental factors and the robot's viewpoint influencing it. This challenge becomes more pronounced with transparent objects. The transition from perceiving objects to effectively manipulating them introduces a unique set of challenges for robots. While recognizing objects is crucial, the subsequent step of accurately manipulating them adds a layer of complexity. This complexity arises from the fact that robots not only need to perceive objects precisely but also detect their pose, representing their position and orientation. This challenge is particularly notable when dealing with transparent objects, as robots must precisely identify them and understand how to grasp them securely without causing damage or spillage.



(a) (b)

Figure 1.1: A robotic gripper, grabbing 1.1a a transparent jar from a correct angle, and 1.1b a glass of water from the wrong angle.[1]

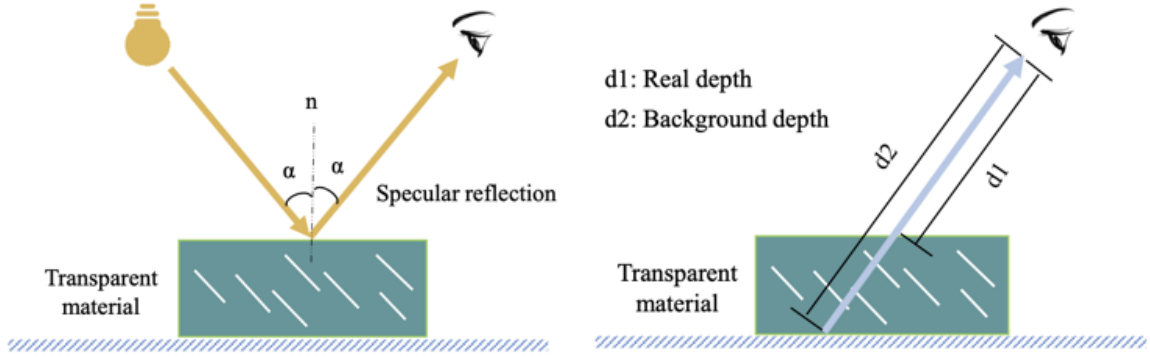
Considering a scenario where a robot is tasked with tidying up a misplaced glass by picking it up (shown in Figure 1.1a) and returning it to the sink. While humans instinctively choose the most stable angle to grasp the glass from its side, this task presents challenges for a robot, mainly if it contains hazardous liquid. Accurately estimating the pose of the glass allows the robot to calculate the optimal grasping angle, ensuring the safe and successful handling of the object and preventing undesirable consequences (Figure 1.1b). Pose estimation in this context refers to determining the object's orientation and location relative to the robot's viewpoint.

The difficulty in manipulating transparent objects arises from their visual properties. Transparent objects lack distinct surface features and rely heavily on the background, as shown in Figure 1.3. For example, in Figure 1.3a, poor lighting, thin glasses, and striped patterns in the container affect glass visibility. The second glass, lying at the container's bottom, can be easily confused with surface reflections. Container textures dominate the scene. In Figure 1.3b, lighting and container material impact the glass in the middle, and its pose may be misinterpreted as if it is facing us.

Furthermore, transparent materials defy the Lambertian assumption [SLR+80] underpinning optical 3D sensors, such as LiDAR and RGB-D cameras. This assumption relies on objects reflecting light uniformly in all directions, resulting in consistent surface brightness from any viewing angle. However, transparent objects disrupt this model by reflecting and refracting light, shattering the Lambertian assumption. This disruption poses significant challenges for obtaining accurate depth data from depth sensors, resulting in either invalid data or unpredictable noise.

Figure 1.2 illustrates the two main reasons for corruption in depth information. Firstly (1.2a), the depth of specific regions is missing due to specular reflection on the surface

[1] 1.1a Sandra Liu and Edward Adelson, MIT (https://news.mit.edu/2022/flexible-way-grab-items-feeling-0415) 1.1b https://spectrum.ieee.org/universal-jamming-gripper#toggle-gdpr



(a) (b)

Figure 1.2: Cause of depth error of transparent material: 1.2a depth error is caused by specular reflection; and 1.2b depth error is caused by light passing through the transparent material. [XCY+20]

of a transparent object. Secondly (1.2b), instead of capturing the actual depth on the object's surface, the false distorted depth on the background behind the object is captured because light passes through the transparent material and refraction occurs. This issue further complicates the perception of transparent objects. For instance, considering a scene from the Tracebot dataset (discussed in Chapter 4), illustrated in Figure 1.3c. Apart from the background's influence on the canister's appearance, the corrupted depth image in Figure 1.3d reveals that the canister's depth information is notably absent from the scene's depth image. Depth image in this context refers to an image containing the distance of each pixel in the observed scene from the camera view point.

To tackle this challenge, a method is proposed to refine the pose from RGB images after a pose estimator. This approach ensures a more accurate pose, which can help identify the best grasping angle for each object. To enable refining using RGB images, a differentiable rendering is employed, which pertains to rendering engines that generate images in a way that allows differentiation. The goal is to optimize rendered images within the scene to align with observed segmentation, edges, and overall scene-level plausibility. It liberates us from the necessity of depth information and the intricacies posed by textureless objects.

This thesis focuses on the pose refinement of transparent objects. To achieve this goal, a collection of loss functions is introduced that enables addressing this challenge through inverse rendering, ultimately estimating and refining the Six Degrees of Freedom (6DoF) poses of transparent objects. The 6DoF represents an object's motion in 3D space, which includes linear translation and axial rotation. Every loss function capitalizes on scene information and enforces it as a constraint on the optimizer, guiding it to align the positions of objects in both the rendered and authentic images.

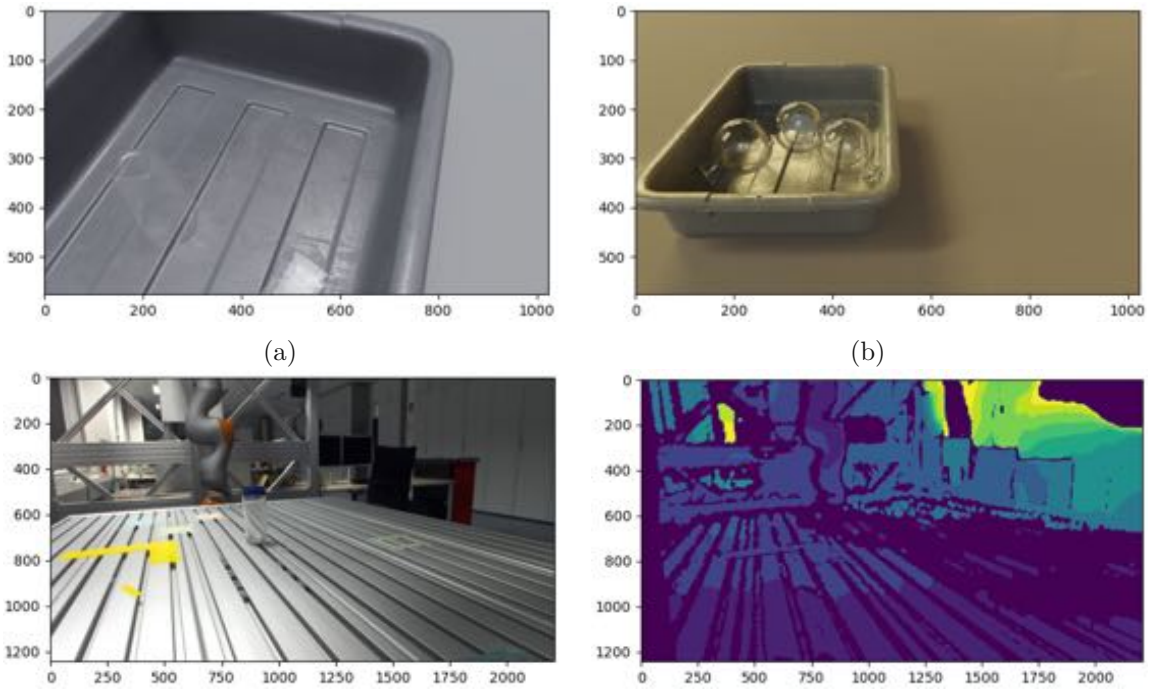
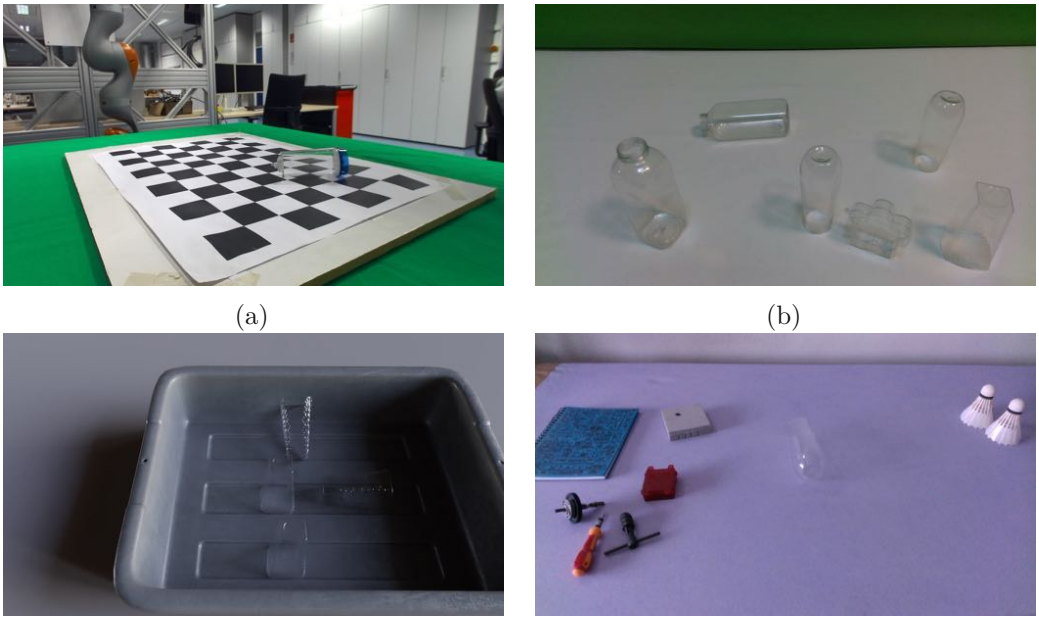


Figure 1.3: Different applications of transparent object perception. 1.3a Two barely visible glasses inside a grey container [SMP+19]. 1.3b Three circular glasses in a grey container with a golden glow [SMP+19]. 1.3c A scene from the Tracebot dataset 4 and how the background affects the perception of the canister. 1.3d The depth image of the scene does not show any information about the canister.

1.1 Challenges and Research Questions

Transparent objects exhibit distinctive optical properties that set them apart from opaque objects. Object perception relies on light reflecting from an object's surface and reaching our eyes. Transparency allows more light to pass through, resulting in a clearer view of both the object and its background. As a result, transparent objects' appearance is heavily influenced by the surrounding scene through reflection and refraction. This relationship is reversed for opaque objects.

The background's texture affects transparent object perception in RGB images, which is observable bleeding through (Figure 1.4a). The impact of the background on transparent object perception varies based on the object's shape, material, surface patterns, orientation and background details (Figure 1.4b). The impact of orientation is noticeable in Figure 1.4b, where a horizontally placed bottle is more visible than a vertical one. Similarly, a star-shaped object appears denser and more noticeable than a square glass. Moreover, the background can also affect object visibility, as seen in Figure 1.4d, where a colourful background can blur object edges. The transparent object's positioning, surface



(a) (b)

(c) (d)

Figure 1.4: Various factors affecting the perception of transparent objects are depicted in the following figures. Figure 1.4a demonstrates the influence of background patterns on the appearance of a transparent canister from the Tracebot dataset 4. Figure 1.4b showcases the impact of different shapes and materials, while Figure 1.4c highlights the role of lighting conditions and surface patterns. Lastly, Figure 1.4d illustrates the effect of background color on object appearance. The final three scenes are sourced from the Cleargrasp dataset [SMP+19].

pattern, and illumination of the scene (Figure 1.4c) can significantly impact its visibility, making it more easily perceived than other objects when leaning on the container.

As shown, creating a comprehensive dataset covering all these variables is impractical, time-consuming, and less adaptable when dealing with novel transparent object classes. In contrast, collecting datasets for opaque objects is more straightforward due to the limited impact of surface and background patterns on object perception. This absence of a generalized dataset impacts the effectiveness of Neural Network approaches, which are extensively employed. Neural networks require vast and diverse datasets for optimal performance, as they are susceptible to insufficient data and tend to overfit. To ensure robust generalization, it's essential to encompass a broad spectrum of object shapes and patterns.

Concerning depth images, the background's depth information may be inadvertently captured instead of the object's, or valid measurements may prove elusive when dealing with transparent objects. In contrast, depth images enhance pose estimation and refinement for opaque objects, contributing vital details that improve the method's performance.

However, this advantage does not apply to transparent objects. Efforts have been made to rectify this limitation by addressing the depth information issue beforehand and subsequently incorporating it into the estimation process. Yet, this remains a complex problem, necessitating extensive time and experimentation.

A novel approach is used to utilize RGB images to address the limitations of relying on extensive generalized datasets and the absence of depth information. This can be achieved by leveraging the impacts of transparent objects on their environments, as depicted in Figure 1.4. Despite their complexity, transparent objects influence their backgrounds, providing valuable visual cues for pose estimation and refinement.By including this influence in the rendered image, meaningful gradient information can be generated for differentiable rendering. Consequently, this method becomes more adaptable and efficient in scenarios where traditional approaches face limitations.

1.2 Approach and Contribution

Earlier in the previous section, the need for an approach that minimizes reliance on extensive datasets, accommodates new objects effectively and minimizes contextual factors affecting object visibility was highlighted. This thesis presents an innovative solution for refining the pose of transparent objects to address these challenges. This method employs a pipeline utilizing differentiable rendering. Differentiable rendering enables calculating and propagating gradients for 3D objects through 2D images. This technique generates a scene image for comparison with the actual captured view. Three different loss functions have been formulated to define this pipeline, where each loss represents metrics indicating deviation from the reference configuration, facilitating the use of differentiable rendering for refinement.

The integration of loss functions with differentiable rendering allows us to optimize each scene individually without requiring a learning algorithm. This process closely mirrors how a human, relying on visual perception, comprehends an object by attempting to fit their internal representation of the object to the observed scene [HZPB20]. Similarly, when encountering an object, this method seeks similarities between the remembered object and the observed one. These similarities involve imagining the object in a specific position, aligning its shape in the scene, estimating the distance to the eye (or camera), identifying edges, or recognizing its resting position on a table or other objects.

This approach leverages mesh and RGB information, combined with differentiable rendering to simulate this perception. To get to the perfect match between the visible scene and the object mesh, loss functions are defined to minimize the error from the perfect match by utilizing image space and physical constraints to refine the object pose.

To summarize, the main contributions of this thesis are:

- The transparent objects' pose refinement method is presented based on differentiable rendering by defining loss functions using scene images, rendered images, and

physical constraints.

- A dataset of transparent objects used in the Tracebot project has been recorded and annotated.

1.3 Organization

This section provides an overview of the upcoming chapters' structure. Chapter 2 offers a brief introduction to the concept of inverse rendering, the BOP dataset and the evaluation metrics employed in this study. It is followed by a literature review of related work. Chapter 3 presents an in-depth explanation of our pose refinement pipeline using inverse rendering techniques. Chapter 4 explores the transparent object dataset, outlining its setup and the variety of objects it encompasses. Finally, Chapter 5 discusses various design choices and the outcomes of our experimental evaluations.

CHAPTER 2

Background and Related Work

This chapter delves into the fundamental concepts and existing research relevant to the proposed method for transparent pose refinement using differentiable rendering. It begins by providing background information on Inverse Rendering, the BOP dataset, and the Evaluation Metrics employed in this thesis. Subsequently, related works in three key areas are reviewed: Pose Estimation, Pose Refinement, and Inverse Rendering.

2.1 Background

This section briefly overviews inverse rendering, explains the evaluation methodology used for the experiments in Chapter 5, and delves into the BOP dataset, explicitly focusing on T-Less for enhanced familiarity.

2.1.1 Inverse Rendering

Inverse rendering is a computational technique in computer vision and graphics that entails estimating the three-dimensional properties of a scene or object based on a two-dimensional reference image. Unlike traditional rendering, which generates 2D images from 3D models, inverse rendering works bi-directionally. It starts by assuming 3D structures, lighting conditions, and material properties that could produce a given 2D image. If the assumption is incorrect, the method learns from the mistake and iteratively refines the estimated 3D properties until they closely align with the observed 2D images. This technique is particularly valuable for tasks like pose estimation, which is crucial for determining objects' precise position and orientation in a scene using RGB images. Figure 2.1 illustrates the components of an inverse renderer (Mitsuba 2 [NDVZJ19]), showcasing the iterative process of refining scene parameters to match a reference image through rendering, loss calculation, and backpropagation loop. For this thesis, we used Pytorch3D differentiable renderer [RRN+20].

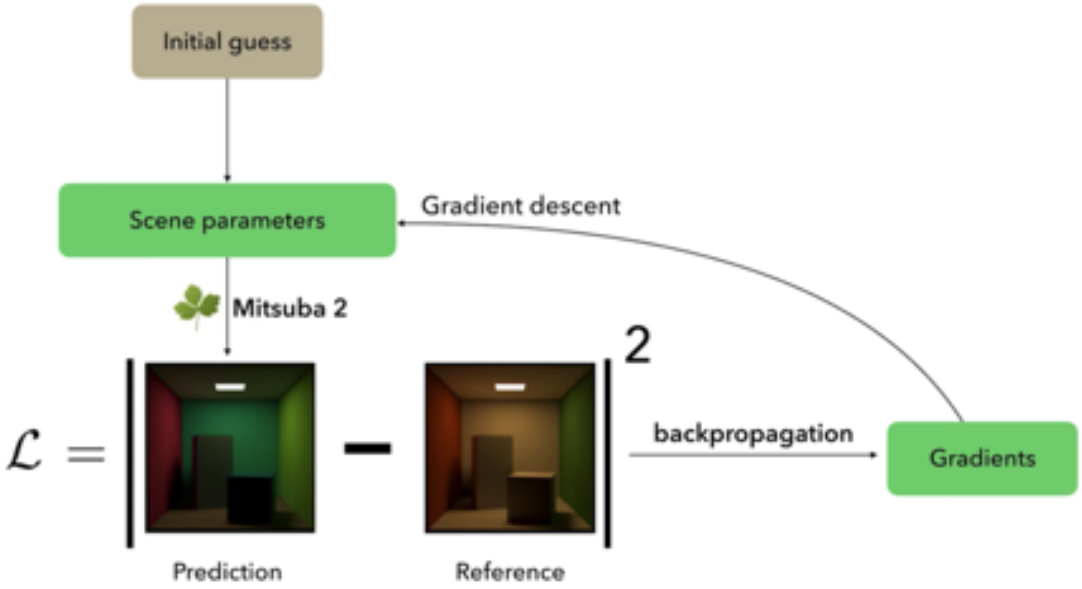


Figure 2.1: Optimization loop of an inverse renderer (Mitsuba 2 [NDVZJ19]) refining scene parameters in relation to a 2D reference image.

2.1.2 BOP Dataset

The Benchmark for 6D Object Pose Estimation (BOP) dataset is a comprehensive and widely used computer vision benchmark designed explicitly for evaluating 6DoF object pose estimation methods. The dataset contains diverse objects from different view points, each associated with annotated ground truth poses. It provides RGB-D images, depth maps, camera calibration information, and 3D models for each object. The dataset is challenging due to its lighting conditions, object appearances, and location variability. Therefore, It is extensively used to assess the robustness and accuracy of algorithms in estimating the 6DoF pose of objects in cluttered and realistic scenes.

One subset of the BOP benchmark is T-Less [HHO+17], featuring thirty distinct industry-relevant objects characterized by a lack of notable texture or discriminative colour. These objects display symmetries and mutual similarities in shape or size; some are composite structures. While this thesis focuses on only a few scenes from T-Less to establish a baseline and ensure comparability with the current state of the art in object pose estimation, it is noteworthy that the dataset does not include transparent objects. Figure 2.2 showcases sample images of scenes one and two extracted from the T-Less dataset used in this thesis.

2.1.3 Evaluation Metrics

In this section, we examine the evaluation metrics employed in this thesis.

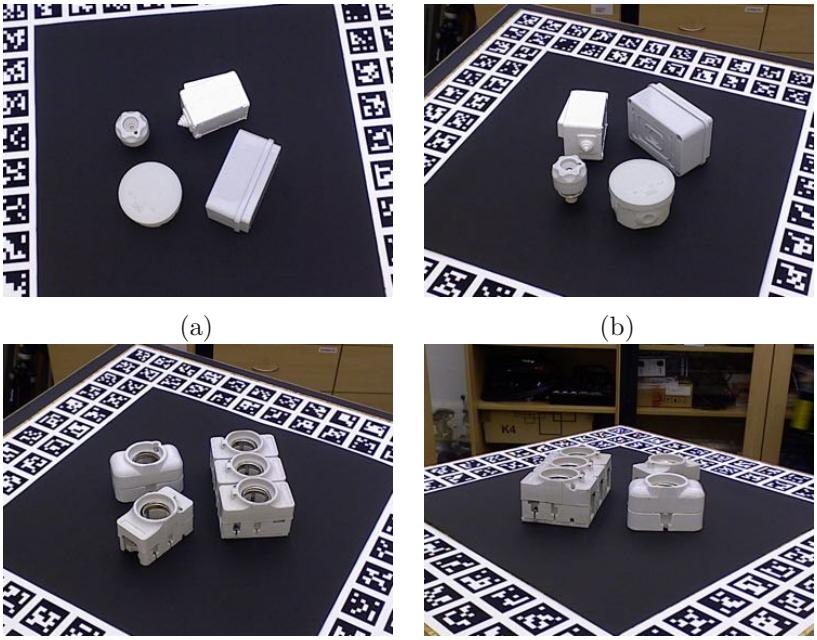


Figure 2.2: Samples from the T-Less dataset [HHO+17] showcasing scenes one (a, b) and scene two (c, d), each captured from distinct viewpoints.

Average Distance of Model Points. Average Distance of Distinguishable Model Points (ADD) is the most widely used evaluation method introduced by Hinterstoisser et al. (2013) [HLI+13]. This method evaluates the error between the estimated pose $\hat{P}$ and the ground truth pose $\bar{P}$ of an object model M without indistinguishable views. The ADD metric is computed as the average distance across corresponding model points, expressed by the formula:

$$e_{ADD}(\hat{P}, \bar{P}; M) = \underset{x \in M}{\text{avg}} \|\bar{P}_x - \hat{P}_x\|_2 \tag{2.1}$$

In cases where the model M has indistinguishable views, the Average Distance of Indistinguishable Model Points (ADI) error is calculated as the average distance to the closest model point:

$$e_{ADI}(\hat{P}, \bar{P}; M) = \underset{x_1 \in M}{\text{avg}} \min_{x_2 \in M} \|\bar{P}_{x_1} - \hat{P}_{x_2}\|_2 \tag{2.2}$$

A smaller ADD and ADI value implies a better alignment between predicted and ground truth poses. An object model M is considered to have an indistinguishable view ([HMO16]) if:

$$\exists P, \exists P', \exists C : d(v_c[PM], v_c[P'M]) \leq \epsilon \wedge f(P, P') \geq \rho \tag{2.3}$$

where $v_c[M] \subseteq M$ is the part of the model surface visible from the camera C, the function d measures a distance between two surfaces with poses P and P', ϵ being the tolerance controlling the level of distinguishable details, and ρ being the minimum required distance f between the poses. The threshold ρ is required since they are many nearly identical poses for which the surface distance is below ϵ.

Maximum Symmetry-Aware Surface Distance (MSSD) . If the symmetry transformations for a specific object are identified, the MSSD [DUB+17] enables the explicit assessment of errors related to any equivalent, symmetrical pose of the model (indistinguishable views) by:

$$e_{MSSD}(\hat{P}, \bar{P}, S_M, V_M) = \min_{S \in S_M} \max_{x \in V_M} \|\hat{P}_x - \bar{P}_{S_x}\|_2 \tag{2.4}$$

where the set S_M contains global symmetry transformations of the object model M and V_M is a set of the model vertices.

Maximum Symmetry-Aware Projection Distance (MSPD) . Similar to the MSSD, but here the symmetrical pose of the model after projection to the 2D pixel space is taken into account [HSD+20]:

$$e_{MSPD}(\hat{P}, \bar{P}, S_M, V_M) = \min_{S \in S_M} \max_{x \in V_M} \|proj(\hat{P}_x) - proj(\bar{P}_{S_x})\|_2 \tag{2.5}$$

where the function proj(.) is the 2D projection.

Visible Surface Discrepancy (VSD) . This metrics [HSD+20], considers the disparity of the rendered depth images of the object under ground-truth pose $\bar{p}$, donating $\bar{D}$ and estimated pose $\hat{p}$, contributing $\hat{D}$, by:

$$e_{VSD}(\hat{D}, \bar{D}, \hat{V}, \bar{V}) = \underset{p \in \hat{V} \cup \bar{V}}{\text{avg}} \begin{cases} 0, & \text{if } p \in \hat{V} \cap \bar{V} \wedge |\hat{D}(p) - \bar{D}(p)| < \tau \\ 1, & \text{otherwise} \end{cases} \tag{2.6}$$

where the τ is a misalignment tolerance.

Average Recall . In addition to considering multiple thresholds, this metric uses three different error functions: the MSPD, the MSSD and the VSD. The average recall rates over ten thresholds are computed per the error function. The overall performance score is the average recall over the three resulting sub-scores. See [HSD+20] for a definition of

the thresholds used per sub-score. For the calculation of the average recall, BOP toolkit[1] is used.

2.2 Related Work

This section reviews the state-of-the-art methods for 6DoF pose estimation, a crucial step preceding pose refinement, followed by pose refinement methods used for both opaque and transparent objects. In the end, it finishes by reviewing approaches which use inverse rendering in their methods.

2.2.1 Pose Estimation

Different methods for estimating poses are being studied in this subsection. Some approaches focus on estimating object pose without leveraging the object model [GNZN23, WSH+19, CLWX21, YCFB+21]. This section further explores techniques employing an object model, aligning more closely with the approach discussed in subsequent sections of this thesis.

Opaque Objects

The 6DoF object pose estimation challenge has been extensively explored in robotics and computer vision. A prevalent strategy involves establishing 2D-3D correspondences, subsequently applying a variant of the PnP/RANSAC algorithm [SSF+22, BR19, TSF18, NGSL23, PPV19]. According to the BOP challenge, GDR-Net [WMTJ21] stands out as the state-of-the-art pose estimation method. Alternatively, some approaches adopt an end-to-end paradigm, directly predicting the pose as the network's output [LWJ+18, TCM15, XQL+19, DMW+21, LMM+22]. Another category utilizes the point cloud representation of the scene, establishing a 3D-3D correspondence between scene points and the object mesh [HB19, QHZ+23, HH, GLH+21]. However, this method is not effective for transparent objects due to the noisy depth images, making scene point cloud construction impractical.

Transparent Objects

Transparent objects have unique optical properties that make it difficult to estimate their depth and recognize them accurately. This is because they reflect and refract light in a way that produces variations in their appearance. As a result, detecting and determining the position and orientation of transparent objects are complex tasks. Some primary works have attempted to estimate the pose of a transparent object using traditional feature extraction methods such as edge detection [LEB13, LR13] and SIFT features [GHJYAJ19]. However, these approaches are less effective than using high-level deep features. Later, researchers tried to eliminate depth image errors and enhance

[1] https://github.com/thodan/bop_toolkit

the accuracy of the estimation. One approach [SMP+20] used a deep convolutional network to adjust the depth image and combine the outputs of this network with mask images, surface normals, and boundary detections. Another approach [XCY+20] used surface norms, plane information, and UV maps combined with colour feature extraction to estimate the object's pose of interest. Other approaches extract 2D [BKC22] or 3D [LJAK20] key points from RGB or stereo images rather than using depth images. Finally, Transnet [ZOC+22] explored the effectiveness of the category-level approach in transparent object pose estimation and used a two-stage method to estimate the object's pose by reconstructing the depth image and feeding it to a transformer-based point cloud embedding module. The latest method [CJS+23] used stereo cameras directly instead of depth image reconstruction.

Overall, the field is moving from traditional feature extraction to deep learning-based methods, focusing on addressing the depth estimation challenges of transparent objects by reconstruction [CWX+23], enhancement, or using RGB stereo inputs.

2.2.2 Pose Refinement

Refinement methods for 3D pose are founded on the premise that the projection of a 3D object model aligns with the object's appearance in the image when the correct 3D pose is applied. The following discussion will explore pose refinement methods for opaque and transparent objects.

Recent research studies [ZSI19, SMD+18, XSNF17, SSH20, LWJ+18, MKNT18, CJS+23] have shown that incorporating a pose refinement network after the initial pose estimator is effective for 6DoF object pose estimation. Various methods, such as PoseCNN [XSNF17] and AAE [SMD+18], utilize an ICP algorithm [Zha21] with depth information to realign the known object model to the observed depth image. In contrast, others, like SSD-6D [KMT+17] and HybridPose [SSH20], refine the pose by optimizing a modification of the reprojection error.

Some other methods [ZLS14, LWJ+18, MKNT18, ZSI19, LCAS20] first render a 2D object image based on the initial pose and then compare it with the observed image using pixel-wise correspondence [ZLS14] or a CNN [LWJ+18, MKNT18, MKNT18, ZSI19, LCAS20] to estimate the residual pose. However, many of these RGB-CNN-based methods require extensive training data and may need to be more robust in practical scenarios. Additionally, they rely on a cumbersome CNN for pose regression, sacrificing efficiency. A later study [ILK+21] proposed a method that reuses image features extracted by a CNN and defines a loss based on the texture colour on a pixel level to address these challenges. However, these methods have limited performance as they cannot generalize to 3D poses or 3D models that were not included in the training data. The final method we explored is StereoPose [CJS+23], which specifically tackles the challenge of transparent objects. This method utilizes the Parallax magnitude of a transparent object as observed in stereo images. Parallax is affected by the shape and depth of the objects and represents the

apparent shift in an object's position relative to a background due to a change in the observer's viewpoint.

2.2.3 Differentiable Rendering

Differentiable rendering [MST+21, NDVZJ19, RRN+20, LB14] is a powerful concept that provides inverse graphics capabilities by computing gradients for 3D scene parameters from 2D image observations. This novel technique recently gained popularity for 3D reconstruction [KLR18, NPLBY18, LLCL19], scene lighting estimation [ALKN19, LADL18], and texture prediction [KTEM18, YHZ+18] as well as camera pose estimation [YCFB+21].

In the previous section 2.2.2, some methods employing rendering to compare the estimated pose of an object to the reference image for refinement were mentioned. However, they calculate the gradient separately from the renderer. Recent approaches based on differentiable rendering have overcome this limitation [PBCC18, BEM23, WKY21, GWZ+20]. These methods exploit knowledge about the 3D scene geometry and the projection pipeline for optimization. The method [PBCC18] uses a renderer after an encoder that estimates an object's pose and category. They use the mask space to refine the pose of an object by employing a category-level 3D object mesh instead of the accurate mesh. NeMO [WKY21] proposes a pose refinement method by learning the texture of a 3D model via contrastive loss. Another approach is using the feature space. Authors in [BEM23] utilize a 2D feature correspondence between rendered and reference images using a local gradient (Jacobian), although it only works for non-feature-less objects. On the other hand, [GWZ+20] employs a small convolution network for feature extraction, but their method is still limited to ideal scenarios. The most related work to ours is Diff-DOPE [TWB+23], which uses RGB, mask, and depth images for calculating their loss function. The idea behind this paper is very similar to this thesis. It is worth noting that their work was published on September 30, 2023, indicating that both had similar ideas.

CHAPTER 3

Inverse Rendering Pipeline for Pose Refinement

This thesis proposes a novel solution to tackle the challenges associated with transparent object pose estimation and refinement. This approach involves using a pipeline that leverages differentiable rendering, producing a rendered view of the scene that can be compared to the real scene captured view. By utilizing differentiable rendering, loss definitions can be calculated, which, in its simplest form, quantifies the difference between the reference and predicted images. Furthermore, the gradient of this loss can be computed within the image space. This gradient provides valuable information that guides the method in adjusting the predictions to minimize the loss and bring the predicted image closer to the actual image. Calculating the gradient directly in the image space offers several advantages. It eliminates the dependency on similar training images, reduces the need for extensive training data, and enables the inclusion of new class objects without prior exposure.

Figure 3.1 outlines the critical steps in the pipeline for a single image iteration in scene 1 containing a canister. The input combines mask, and depth images with the 3D representation of the object. To address sensitivity to local changes and illumination, the mask images were extracted from the RGB image through object detection methods. Mask image in this context refers to a 2D image in black and white, showing the pixels containing the object of interest in white. The iteration begins with an initial estimated pose matrix (Figure 3.1(I)) representing the 6DoF pose (see details in Section 3.1). This pose is applied to the object's 3D representation, producing the transformed 3D representation (Figure 3.1 (II)). Passing through the renderer, this representation generates the prediction mask image (Figure 3.1 (III)), forming the basis for the mask loss calculation (Figure 3.1 (IV.b)). Combining the transformed 3D object representation with the depth image yields a 3D scene representation (Figure 3.1 (IV.a)). This is used to calculate the collision loss, constraining object positioning relative to the plane (Figure 3.1

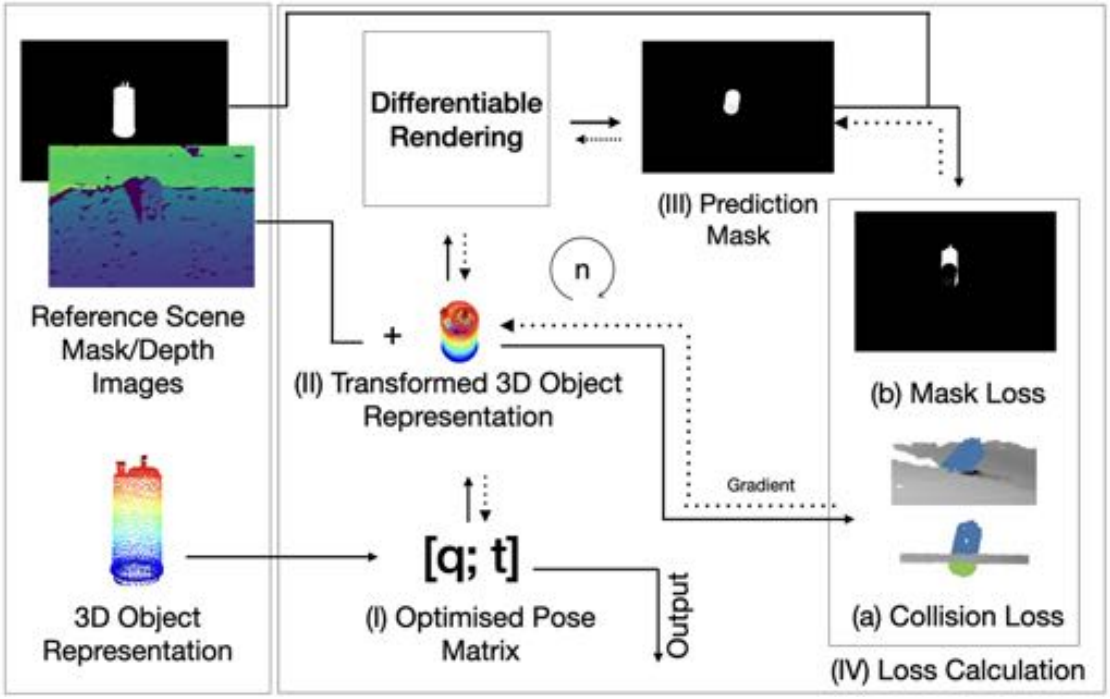


Figure 3.1: Overview of a single image iteration for scene 1. (I) The initialized optimized pose matrix is applied to (II) the 3D object representation and passed through a differentiable renderer to produce (III) a rendered mask image based on the current pose. Loss is calculated between the rendered and reference masks (IV.b) for mask loss, and the scene depth with the transformed 3D object (II) for collision loss (IV.a) is used to guide the object onto the table. The losses flow back to refine the pose matrix (I) for the next iteration. After n iterations, the optimized pose matrix is output.

(IV.a)), and others (for multi-object scenes). In Figure 3.1, the green part of the canister (IV.a) represents the constraint area. In this particular iteration, the canister is shown to be within the table, which is unrealistic. However, the canister will be forced to rest on the table as desired in subsequent iterations. Following the loss calculations, the gradient flows through the renderer and collision loss to the learned pose matrix (b.I) for refinement, initiating the next iteration with the updated pose. The optimization loop continues until it reaches a specified number of iterations (n) or a predefined minimum loss value.

In the subsequent subsections, individual explanations for each part and component of the pipeline have been provided.

3.1 Initialization

The optimization process commences with an initial pose, which can be obtained through two approaches. It can either start with a pose provided by a pose estimation method, which subsequently requires refinement or employ a heuristic process that begins with one of the object's most likely poses.

The heuristic approach begins with estimating the object's potential positions within the scene by calculating its stable poses. These stable poses represent configurations where the object can stably rest on a surface, as opposed to falling. Depending on their shape, objects may have multiple stable poses that are in close proximity. To simplify the process, the Reverse Cuthill-McKee (RCM) algorithm [CM69] has been employed to cluster these poses, selecting the most significant ones.

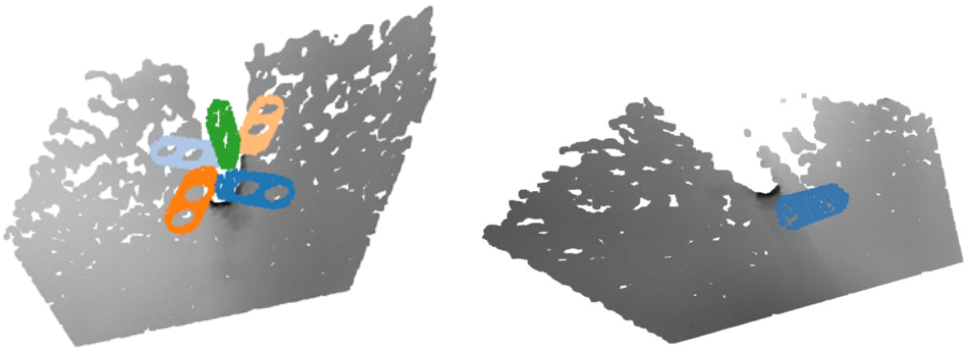
The RCM is a reordering strategy for sparse matrix computations to make them more efficient for storage and processing. The fundamental principle of the RCM is to rearrange the nodes of a graph so that the resulting adjacency matrix [1] has a lower bandwidth [2]. The problem has been reshaped into a graph format to group the stable poses using the RCM algorithm. An adjacency matrix was constructed to accomplish this by categorizing the angles with an angular difference of 15 degrees and excluding the in-plane rotation. Later, by implementing the RCM algorithm, similar poses can be clustered, and the most prominent ones for the initial pose would be chosen. Figure 3.2a showcases the different stable poses for the drain tray object from Tracebot dataset 4.

The subsequent step in the heuristic initialization method involves positioning the object as close as possible to the reference coordinates. By position of the object, values of the $t = [x, y, z]$ coordinate distances with respect to the camera frame is meant, which is called a translation matrix. Estimating the translation matrix utilizes information from the mask, depth, RGB images, and the camera's intrinsic parameters. Algorithm 3.1 provides pseudocode for the heuristic estimation of x and y translations. Depending on the object's chosen stable pose, z translation would be set to the object height. The inputs for this algorithm include plane coefficients (planeCoeff), mask image (maskImg), and camera intrinsics matrix (camInsMatrix).

The plane coefficients and point cloud are obtained from the scene's depth and RGB using the RANdom SAmple Consensus (RANSAC) [Can81]. The initial step involves determining the middle pixel within the masked area of the object. This choice ensures an average point, offering a reliable estimate of the distance to the camera, even when one object is partially obscured by another. The intersection of this middle point of the mask image and the table plane is the x and y translation of the object. Subsequently, the identified point is transformed into the camera frame and positioned on the plane.

[1] An adjacency matrix is a square matrix whose elements indicate whether pairs of vertices are adjacent in the graph

[2] The bandwidth of a matrix counts the non-zero entries in each row and determines the maximum separation.



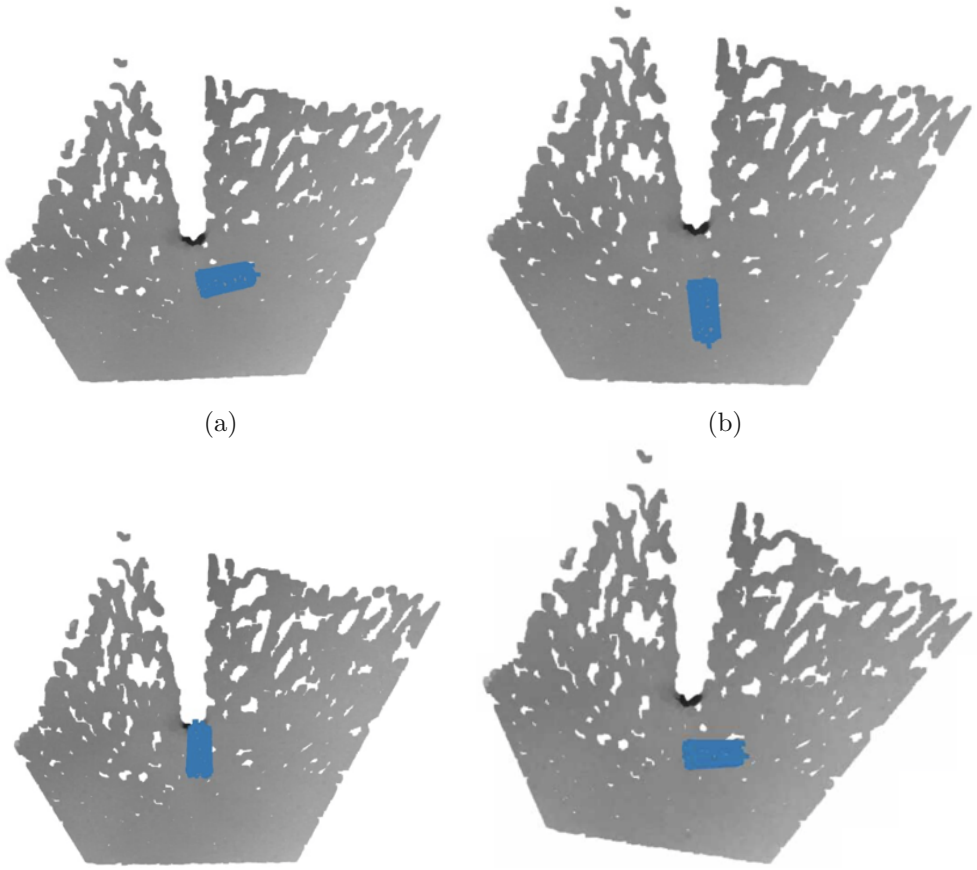
(a) (b)

Figure 3.2: The application of a heuristic for initializing the pose of an object in the initial step. The left image showcases all stable poses for the drain tray object, while the right image exhibits the canister's initial pose. The canister's pose is randomly selected from stable poses and positioned using the algorithm outlined in pseudocode 3.1. Both objects are from the Tracebot dataset 4.

Algorithm 3.1: Estimating the in-plane x and y translation of the object of interest

Input: planeCoeff, maskImg, camInsMatrix
Output: $x_{translation}$, $y_{translation}$

$x_{array} = x \text{ indexes of } maskImg > 0$;
$y_{array} = y \text{ indexes of } maskImg > 0$;
$y = mid(y_{array})$;
$x = mid(x_{array})$;
$p = inv(camInsMatrix).[x, y, 1]$;
$p_{norm} = p$;
$t = (-planeCoeff[-1])/(planeCoeff[0] * p_{norm}[0] + planeCoeff[1] * p_{norm}[1] + planeCoeff[2])$;
$x_{trans} = p_{norm}[0] * t$;
$y_{trans} = p_{norm}[1] * t$;
return x_{trans}, y_{trans};



(a) (b) (c) (d)

Figure 3.3: Representation of the multi-initialization process in the optimization, employing heuristics. Four stable poses are randomly selected and positioned within the scene using Algorithm 3.1.

Figure 3.2b illustrates the initial optimization step for a single canister within a scene, following the pseudocode 3.1 and by choosing one of the canister's stable poses.

For the pipeline with heuristic initialization, the optimization process with various stable poses is initialized to ensure a comprehensive exploration of stable poses and prevent potential local optima during the optimization process(Figure 3.1). These poses are individually optimized in sequence, and the best result is selected after the optimization process.

3.2 Loss Definitions

The next step involves defining the optimisation objectives by determining the loss functions. This section describes the loss definitions and calculations that are used in the

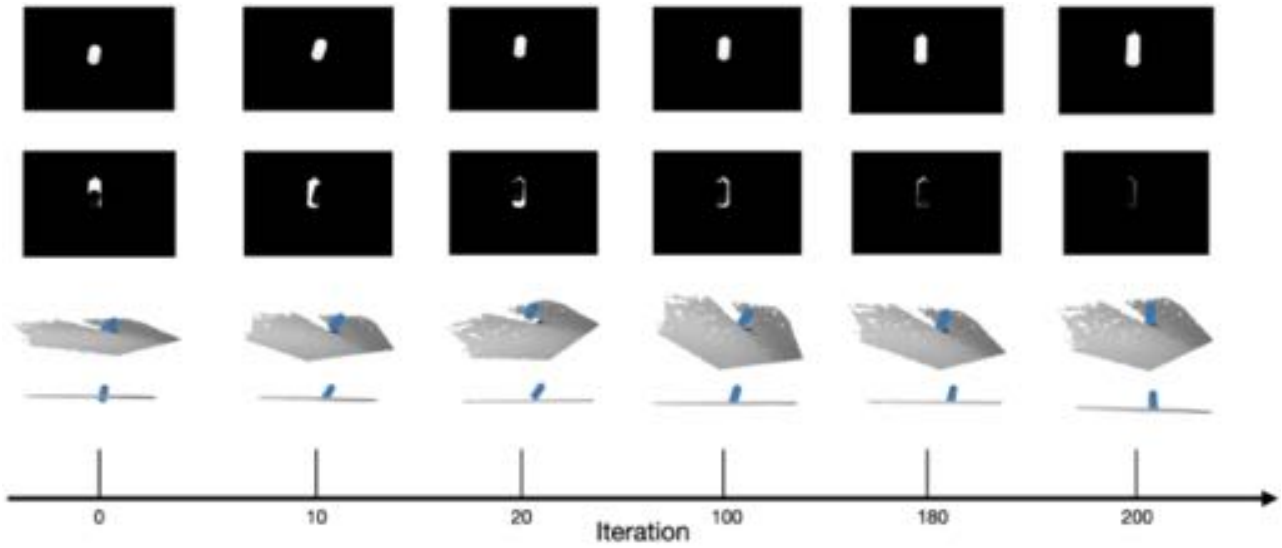


Figure 3.4: Optimization loop output on six different iterations for the scene in Figure 3.1(a. II) (a. II). First row: predicted masks in each iteration. Second row: corresponding mask losses. Third row: corresponding collision losses.

proposed method.

3.2.1 Silhouette Objective

The Silhouette Objective represents the most basic loss definition, mask loss. To compute this loss, the optimized pose (or initial pose for the first iteration) and the object mesh is provided to the renderer, generating a corresponding rendered image. The mask images is rendered instead of RGB to reduce the sensitivity towards local changes and lighting variations. The use of a mask image minimizes environmental influences on object visibility. The loss would be calculated from the difference between the reference mask (I_{ref}) and the estimated mask image(I_{est})(eq 3.1), called difference mask.

$$d_{x,y} = I_{ref} - I_{est} \tag{3.1}$$

The pixel values of $d_{x,y}$ can be one of three possible values: 0, 1, or -1. A pixel value of 1 signifies that a portion of the object was not accurately estimated and is missing in the estimated image, thereby contributing to the loss. On the other hand, a pixel value of -1 indicates that the estimated parts are incorrectly placed and appear in the estimated image but not in the reference image. Lastly, a pixel value of 0 indicates that the parts are estimated correctly and do not contribute to the loss or belong to the background.

Figure 3.5 presents the three components of mask loss calculation within a single optimization step. Following the rendering of the estimation mask in Figure 3.5b, a comparison is made with the reference mask displayed in Figure 3.5a, resulting in the difference mask

(a) Reference mask (b) Estimation mask (c) Difference mask

Figure 3.5: Three components of the mask loss are depicted. Beginning with the reference mask in Figure 3.5a, followed by the estimation mask after rendering in Figure 3.5b, the resulting difference mask is illustrated in Figure 3.5c. This difference mask is subsequently utilized to compute the mask loss.

illustrated in Figure 3.5c. This difference mask is subsequently employed for mask loss calculation. Figure 3.4 showcases a sequence of optimization steps. Figure 3.4(I) displays the estimation mask throughout the optimization steps, while Figure 3.4(II) exhibits the corresponding mask losses computed from the reference mask and the respective estimation masks.

As illustrated, the loss calculation depends on the target object's pose, and the accuracy of the 6DoF pose refinement is evaluated by comparing the rendered mask with the ground-truth mask. This loss can be calculated through three different equations which are called Sum of Squared Mask Difference Loss (L_{SSMD}) (eq. 3.2), Sum of Positive Squared Mask Differences Loss (L_{SPSMD}) (eq. 3.3), and Symmetric Difference over Union (L_{SDU}) (eq. 3.5) respectfully.

L_{SSMD}, is defined as follows:

$$L_{SSMD} = \sum_{x} \sum_{y} d_{x,y}^2 \tag{3.2}$$

By defining the loss from the above equation, the loss value is within the interval $[0, 4t]$ where t describes the number of pixels representing the object of interest in the mask image. Therefore, the value of the loss depends on the object's size in the mask image.

The following loss definition, L_{SPSMD} , is defined as follows:

$$L_{SPSMD} = \frac{\sum_x \sum_y d_{x,y}[d_{x,y} > 0]^2}{\sum_x \sum_y I_{ref}} \tag{3.3}$$

According to the definition in the numerator of the above equation, this loss function only considers the parts of the object that still need to be correctly estimated from the reference image. It disregards the parts in the estimated mask that are not found in the

reference mask. In addition, the loss is normalized by dividing it by the object's size in the reference image, ensuring that the loss value falls within the interval $[0, 1]$.

And the last loss definition, L_{SDU}, is defined as follows:

$$U_{mask} = I_{ref} \cup I_{est} \tag{3.4}$$

$$L_{SDU} = \frac{\sum_x \sum_y d^2_{x,y}}{U_{mask}} \tag{3.5}$$

This loss function is a normalized version of L_{SSMD}, calculated using symmetric difference over union. The numerator of this loss falls within the interval $[0, 4t]$, where t represents the number of pixels that make up the object. In contrast, the denominator falls within the interval $[0, 2t]$. As a result, the loss value falls within the interval $[0, 2]$.

3.2.2 Collision Objective

Real-world constraints dictate that objects rest on surfaces and should not float in the air. In the case of multiple objects, they should not be positioned inside each other. This constraint is addressed by defining a loss and employing differentiable rendering to enforce realistic object placements.

To compute the collision loss (described in Algorithm 3.2), the initial step involves using the depth image and the RANSAC [Can81] algorithm to detect the table and calculate the plane's frame matrix (M_{plane}). This computation transforms object points from the camera frame ($P_{c-frame}$) to the plane frame ($P_{p-frame}$), after applying the estimated pose matrix (M_{est}) to the points. This provides the distance of each object point to the resting plane, which is assumed to be the table in this project. If an object point is inside the plane, it will have a negative value. For each individual object (object b), its collision with other objects (object o) in the scene is calculated. If a collision occurs, those points also have negative values and are added to an array of all points which object b collides with($A_{obj-signdis}$). Finally, the function calculates the collision array ($A_{signed-dis}$) by finding the objects each object collides with the most. In Figure 3.4(III), the position of a canister relative to the plane is illustrated. In the first iteration (first image from the left), the object appears to be inside the plane (and thus inside the table), which is unrealistic. Consequently, in subsequent iterations, the collision loss acts to guide the object out of the plane and onto its surface, ensuring contact and preventing the object from floating in the air. In scenarios involving two objects, the loss ensures that the objects do not intersect from within, avoiding a non-realistic situation. In Figure 3.6, the black points represent the k-nearest neighbour points to the canister from the second object. The vectors from the canister's points indicate the direction of the distance between them and points from the other object, verifying that these points are positioned outside the canister.

Algorithm 3.2: Calculating the signed distance of scene's objects towards each other

Input: M_{plane}, $P_{c-frame}$, M_{est}
Output: $A_{signed-dis}$

$P_{p-frame} = M_{plane}$ @ (M_{est} @ $P_{c-frame}$);
if *more than one object in the scene* **then**
  foreach *object b in the scene* **do**
    $A_{obj-signdis} = P_{p-frame}[b]$;
    foreach *object o other than b within the scene* **do**
      nearest = k nearest neighbouring points of the object o with respect to the object b;
      distance = the distance of the respective points ;
      if *nearest points of object o inside object b* **then**
        $distance = -1 * distance$; /* changing the sign of distance for points inside */
      end
      $A_{obj-signdis}.add(min(distance))$;
    end
    $A_{signed-dis}.add(min(A_{obj-signdis})$;
  end
end
return $A_{signed-dis}$;

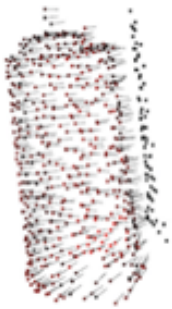


Figure 3.6: Signed distance between points of two objects. Black points show k-nearest neighbours on the second object to canister points. Vectors from canister (red) points represent the distance direction to the neighbour points, positioning neighbours outside the canister surface.

The collision loss is incorporated into the pipeline by adding it to any of the mask loss (L_{mask}) calculations from the previous subsection. The combined loss is referred to as the Masked Signed Distance Loss (L_{MSD}):

$$L_{MSD} = L_{mask} + \gamma * min(A_{signed-dis}) \tag{3.6}$$

This equation considers the maximum collision distance between any two objects in the scene. Here, γ is a coefficient that balances the mask loss and collision loss contributions.

3.2.3 Edge Objective

Edges are crucial for perceiving an object's pose. Matching interior and exterior edges while fitting the object's mesh helps determine the best pose estimate. Therefore, when the reference and estimated mask images have no overlap, the edge information is utilized to guide the optimizer toward aligning the object with the reference edge map. To generate the reference edge map, first, an edge image of the object is created, and then the distance of each pixel to the nearest edge pixel is computed. This yields a 2D image indicating the proximity of each pixel to the object's correct position. Figure 3.7 displays example edge map images. Later, the reference edge map image is multiplied by the edge image from the rendered prediction to calculate the loss, measuring deviation from the ideal position. However, this loss calculation is currently unstable and needs refinement. Therefore, it is excluded from the pipeline representation earlier in this section and would not be used in the experiments.

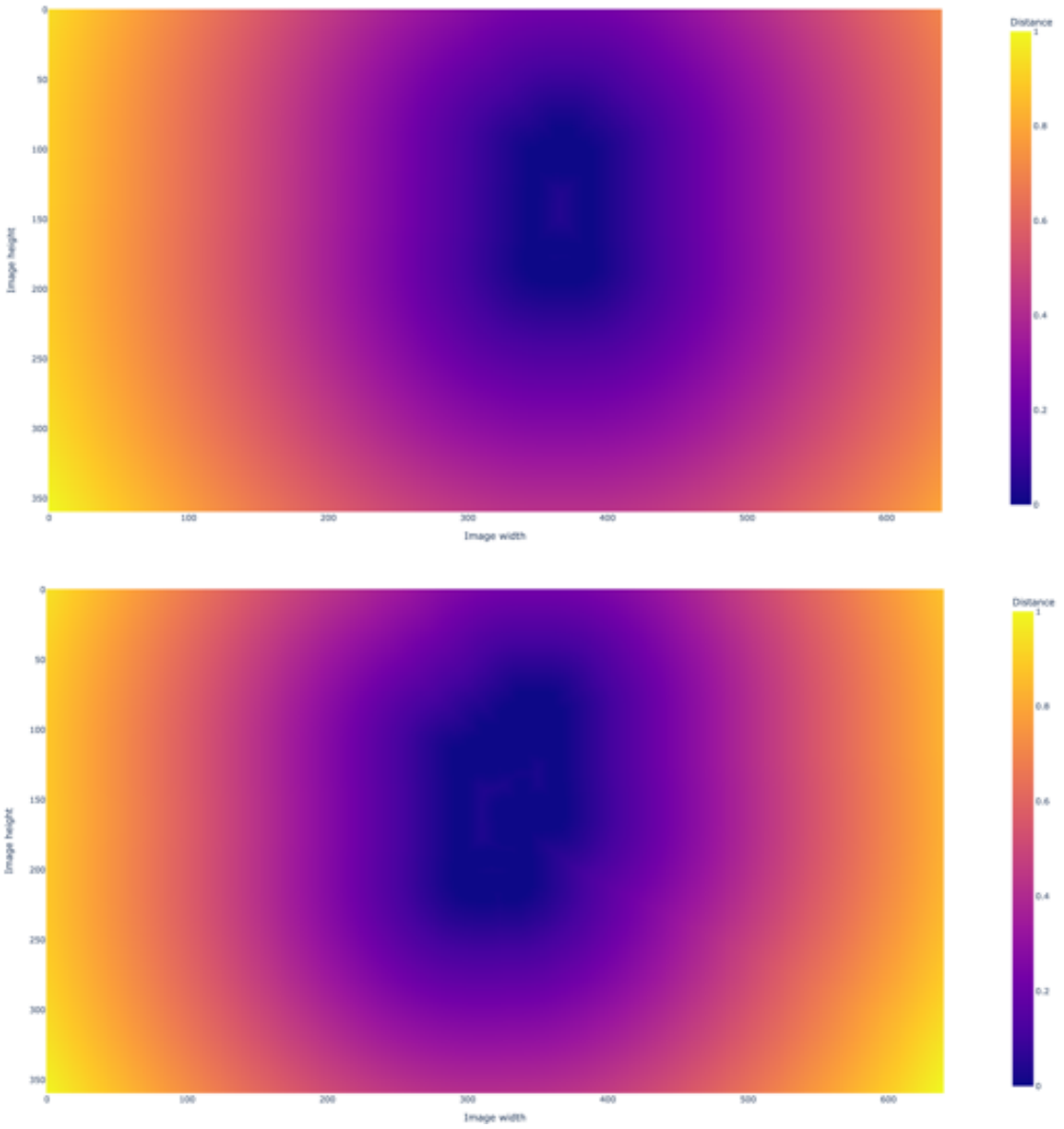


Figure 3.7: The figure portrays the reference edge map images from scene 1 (top) and scene 2 (bottom), indicating the proximity of each pixel to the nearest object edge within the respective scenes.

CHAPTER 4

Transparent Object Dataset

This chapter presents an overview of the Transparent Object Dataset collected for the Tracebot project[1], which includes some of the scenes used in this thesis. The Tracebot project, Traceable Robotic Handling of Sterile Medical Products, encompasses a variety of transparent and non-transparent objects commonly found in medical laboratory settings.

As previously discussed in Chapter 1, the unique features of transparent objects, such as reflecting and refracting, along with their shape, background, and overall scene setting, significantly impact the perception of these objects. Furthermore, given the distinct objects in the Tracebot project that are unavailable in other transparent object datasets, we decided to record our dataset. As depicted in Figure 4.1, the dataset gathered for this thesis contains two different objects: a canister and a drain tray. This dataset comprises real-world scenes with objects' ground-truth poses, which are beneficial for pose estimation and refinement methods. The dataset includes object models, visible masks, depth and RGB images, camera information, and scene parameters. It has three distinct scenes, each featuring 104 diverse images.

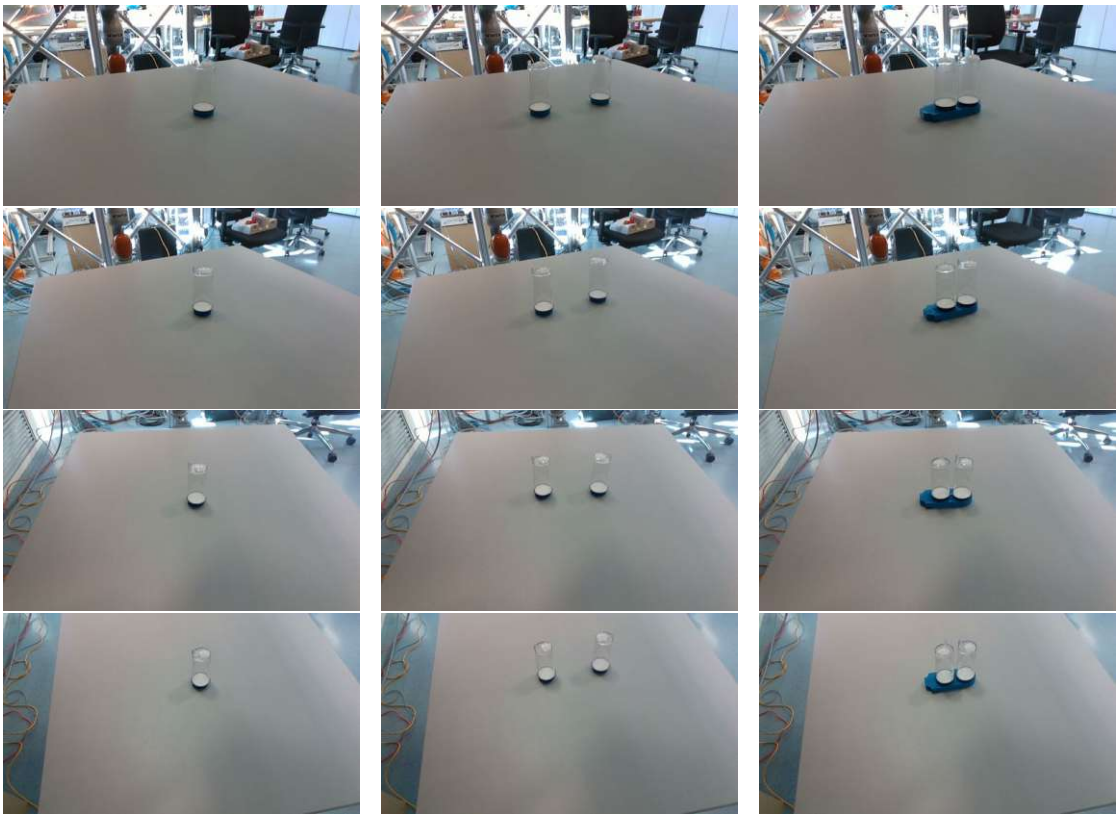
Recording Setup The dataset was recorded using the Kuka LBR IIWA 14 R280 robotic arm (Figure 4.2) and the Intel RealSense D435 camera. The camera was mounted on the robotic arm, with the objects placed on a table and the camera's area of interest focused on the objects within the scene (Figure 4.3). The recording involved four different camera heights and a full 360-degree rotation around the objects, resulting in 26 images per height and a total of 104 images per scene. Figure 4.4 shows that the camera heights increase from top to bottom for the three scenes. The first scene featured a single canister; the second included two canisters, and the third used two canisters placed on top of the drain tray.

[1] https://www.tracebot.eu

Figure 4.1: Objects included in the Tracebot dataset. On the top, we show authentic images of the mentioned objects. From left to right, there is a canister and drain tray. On the bottom, we show the rendered CAD models for each object.

Figure 4.2: Kuka LBR IIWA 14 R280 robotics arm

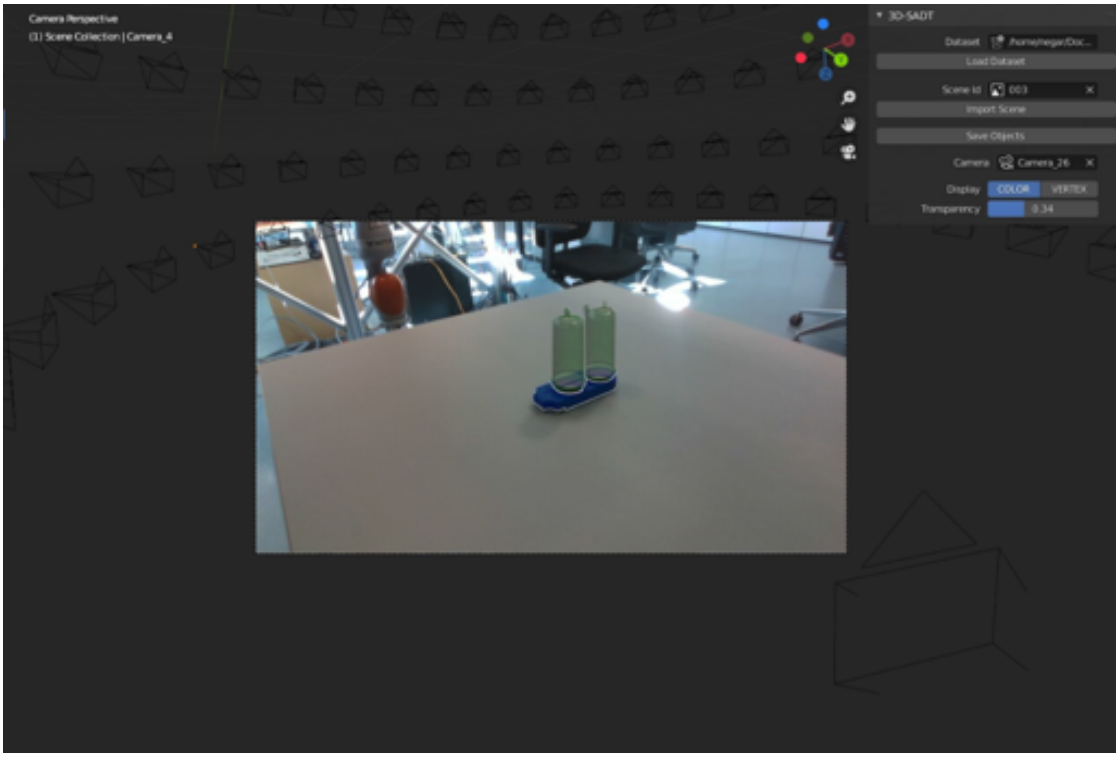
Figure 4.3: Scene recording setup. The camera, mounted on the robot arm, captures images of the canister placed on the table, ensuring a clear focus on the object of interest.



(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 4.4: The three different scenes are shown in this figure. Scene 1 containing a single canister, scene 2 two canisters and scene 3 two canisters and a drain tray. From top to down the height of the camera increases.

(a) Camera views



(b) Camera view 26

Figure 4.5: Annotation process using 3D-DAT [SNS+23]

Object Annotation To annotate the 6D pose of the objects and generate object-wise segmentation masks, we utilized the 3D-DAT tool [SNS+23]. We began by importing the scene of interest and the object models and camera positions into Blender using 3D-DAT. Next, 3D-DAT created cameras with identical viewpoints to the actual images (Figure 4.5a). We annotated each scene separately by switching our view from one camera to another. Figure 4.5b shows the view from camera number 26, corresponding to image 26. In this scene, all three objects have already been annotated.

CHAPTER 5

Results

This chapter investigates the various design choices and their associated limitations to choose the best setting. Later, the most promising design option is compared to two other state-of-the-art refiners according to BOP Challenge 2020.

5.1 Ablation Study - Noise

The following ablation studies aim to provide insight into various design decisions by outlining the tradeoffs of each approach. This allows the selection of the optimal pipeline configuration later. This section compares different loss calculation methods and performs a convergence basin analysis. For both analyses, rotational and translational noise is added along the three camera coordinate axes - the x-axis pointing left, the y-axis upwards, and the z-axis into the image plane.

5.1.1 Mask Loss

Initially, the study begins with the central loss of our optimization, the mask loss. In order to evaluate the effectiveness of various mask loss calculations as defined in chapter 3.2.1, three types of losses were used - Sum of Squared Mask Difference Loss (L_{SSMD}), Sum of Positive Squared Mask Differences Loss (L_{SPSMD}), and Symmetric Difference over Union (L_{SDU}). The experiment involved adding noise to the translation matrix ($[0, 0, 50mm]$) and rotating the matrix along the x-axis by 45 degrees, which was then applied to the ground truth pose. Subsequently, the outcomes across three scenes and two varying learning rates $(0.02, 0.015)$ were compared using the Average Distance of Indistinguishable Model Points (ADI) 2.2 metric, where a lower value indicates a better configuration.

Figures 5.1, 5.2 and 5.3 present a comparative analysis of the performance of different loss calculations for each scene in the dataset using the ADI metrics.

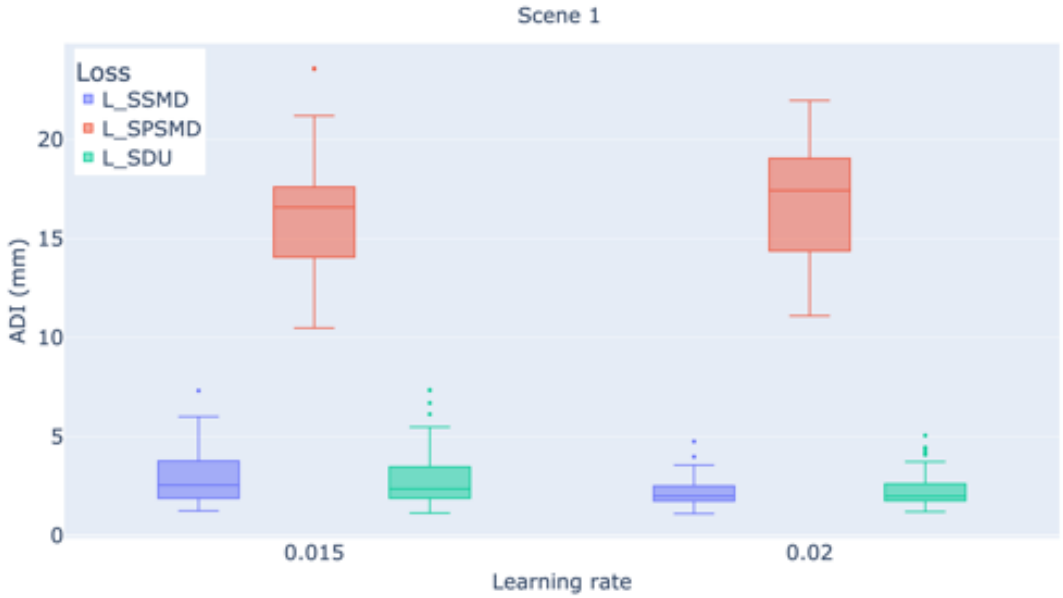


Figure 5.1: The ADI results of different mask loss calculations using two distinct learning rates, 0.015 and 0.02 for scene 1. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

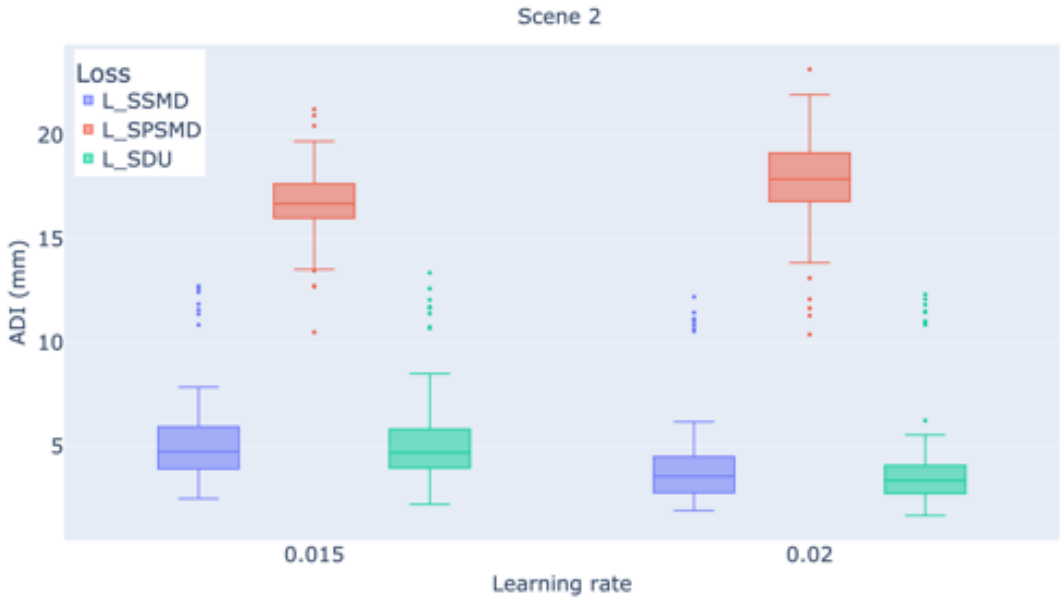


Figure 5.2: The ADI results of different mask loss calculations using two distinct learning rates, 0.015 and 0.02 for scene 2. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

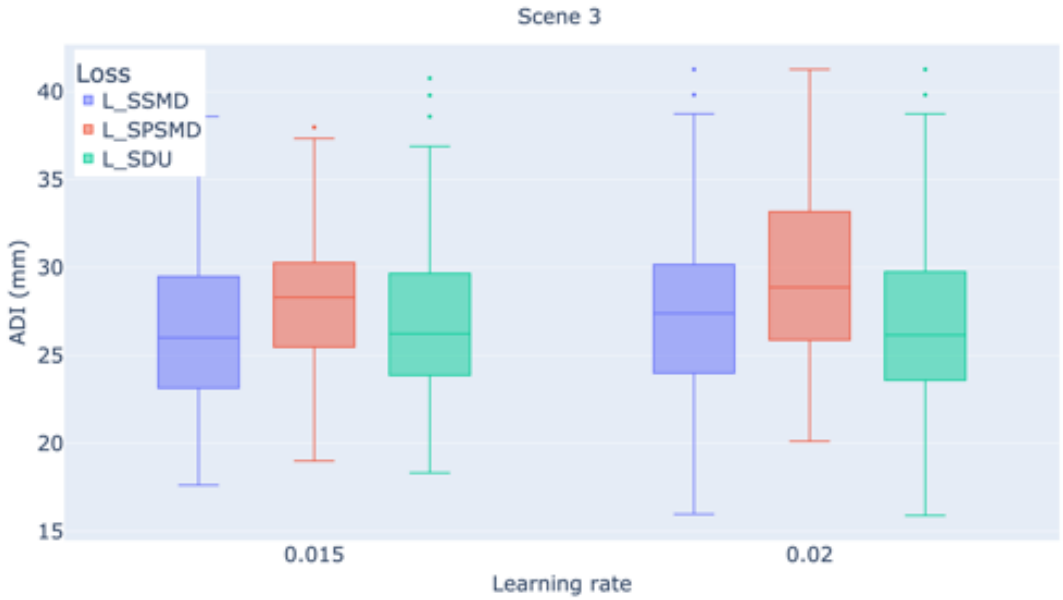
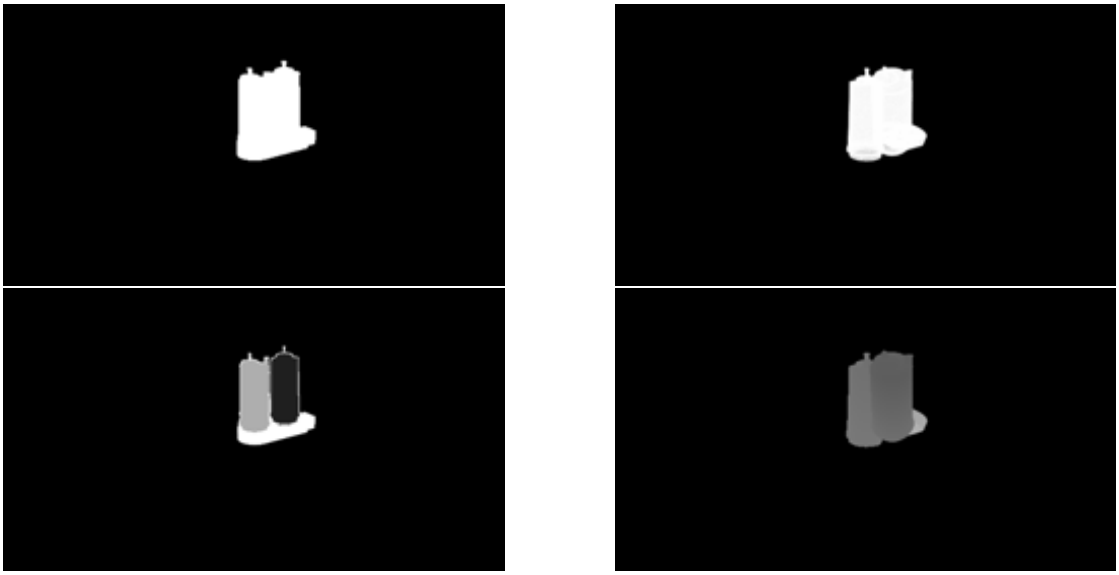


Figure 5.3: The ADI results of different mask loss calculations using two distinct learning rates, 0.015 and 0.02 for scene 3. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

Figure 5.1 demonstrates that L_{SDU} attains the best performance with a learning rate of 0.02 for scene 1. This is supported by its lower average ADI compared to the other losses and the narrower range of ADI values for this configuration, although L_{SSMD} performs similarly well. It is clear that a learning rate of 0.015 is less consistent and more likely to converge to local minima than a learning rate of 0.02. Moreover, the figure highlights that L_{SPSMD} (represented by the orange line) significantly underperforms. This can be attributed to the definition of L_{SPSMD} (see section3.3), which does not take into account the insights that could be derived from the estimated mask. Consequently, it does not yield as effective results as the other losses.

In scene 2, as depicted in Figure 5.2, L_{SDU} with a learning rate of 0.02 outperforms other configurations, while L_{SPSMD} also underperforms for the same reason as for the scene 1. This figure also reveals that the second scene's average ADI for all the configuration exceeds the first. This is because the first scene is relatively simple, containing a solitary object—the canister—whereas the second scene incorporates two objects, increasing complexity.

The results for scene 3 depicted in Figure 5.3 highlight that generally mask loss is most beneficial for the first two scenes, which exclusively feature the canister, as opposed to the third scene, which contains two canisters and a drain tray and objects are in closer proximity. This conclusion drives from the fact that the average ADI for the first two scenes using the mask loss calculations is lower compared to the average ADI for the



(a) Reference mask image (b) Estimated mask image

Figure 5.4: Reference masks (left) and final estimated masks (right) corresponding to image 12 from scene 3. The figure illustrates the challenges in pose refinement for this scene, accentuated by the close proximity of objects. The bottom mask outlines the individual positions of each object within the composite mask.

scene 3.

The third scene's complexity is further depicted in Figure 5.4, which displays the outcome of the final iteration for image number 12. All the mask loss calculations use the combined mask image of the entire scene (shown at the top) to compute the loss. The bottom images in the Figure 5.4 showcase the placement of each object within the mask individually. A closer look at 5.4a versus 5.4b reveals that due to the loss calculation not considering each object's mask separately, the algorithm struggles to position the objects based on the collective scene mask accurately. Consequently, it attempts to bridge the gaps in the "white space" by broadening the objects to better match the white space in the reference mask. As a result, as seen in 5.4b, the canisters are estimated to be closer to the camera, thereby occupying a larger portion of the mask resulting in a smaller loss. At the same time, the drain tray is positioned behind them.

After analyzing Figures 5.1, 5.2, and 5.3, it is evident that the box plots' range using L_{SDU} with a learning rate of 0.02 is influenced by two factors. Firstly, the complexity of the scenes increases from the first scene to the last, as discussed earlier. Secondly, the camera height also plays a role. As mentioned in Chapter 4, the Tracebot dataset includes recordings from four different camera heights for each scene. The average ADI for the three scenes at different camera heights is presented in Table 5.1. The table indicates that the average ADI increases for the second and third scenes as the camera

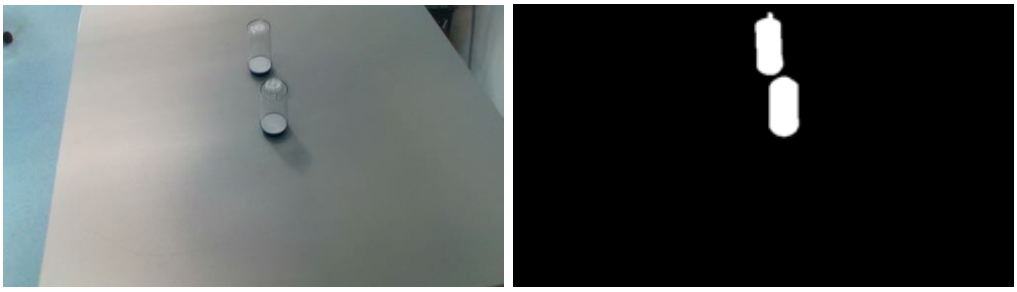


Figure 5.5: RGB (top) and mask (bottom) images of image 102 from scene 2. The mask of the canister at the bottom of the mask image does not encode information about the canister's rotation.

height increases.

	Height 1	Height 2	Height 3	Height 4
Scene 1	2.12	2.30	2.43	2.01
Scene 2	3.35	3.23	3.43	5.92
Scene 3	24.20	26.43	29.31	28.95

Table 5.1: The average ADI for the three scenes, recorded from four different camera heights sorted from first as the lowest and fourth as the highest.

Figure 5.5 displays the RGB and mask images for image number 102 within the second scene containing in recorded from the fourth camera height. As demonstrated, the mask image fails to convey the rotation of the second canister since the two knobs atop the canister are obscured due to the elevated angle of the camera. The other images that exhibit a high ADI face a similar predicament since the pose of the canisters within the estimation mask remains inaccurate despite the presence of white space in both the estimation and reference masks. This discrepancy emphasizes that the mask image alone may not provide sufficient information to accurately estimate an object's pose, particularly when the camera's perspective leads to occlusions or lack of distinguishable features.

Convergence Basin Analysis- Mask Loss

This test aims to determine the degree of error in translation or rotation matrices that would still lead to convergence in mask loss and the accuracy of this convergence. Since the mask loss did not work for the scene 3 and the L_{SDU} with learning rate 0.02 was performing better than other configurations, here the study in the first two scenes, using Symmetric Difference over Union (L_{SDU}) and learning rate 0.02 is performed. The influence of the noise is illustrated with both Average Distance of Distinguishable Model Points (ADD) and Average Distance of Indistinguishable Model Points (ADI). Primarily,

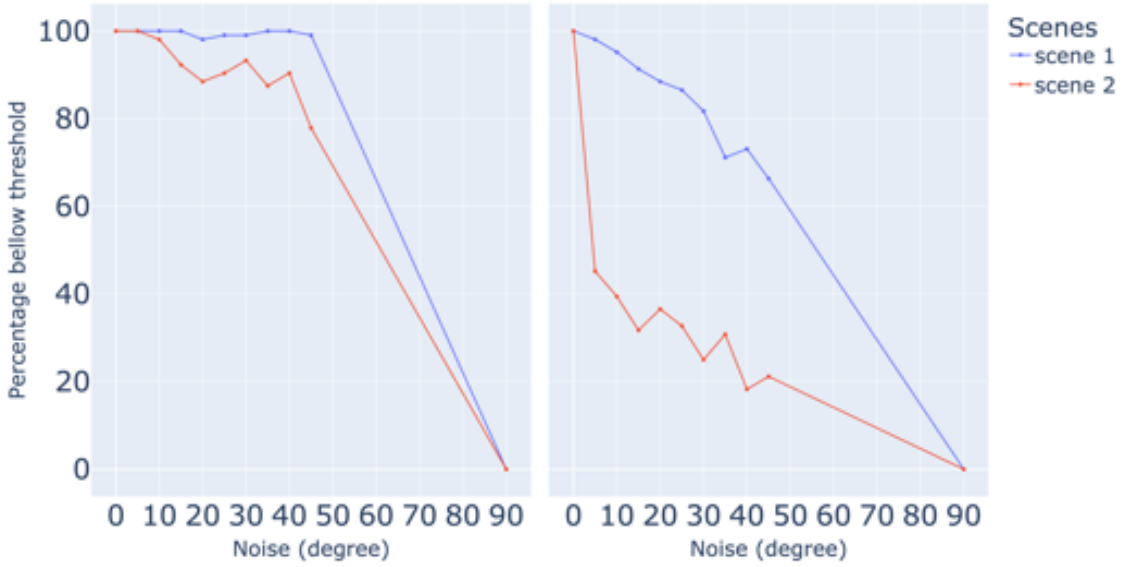


Figure 5.6: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the x-axis using L_{SDU}.

ADI was used, but ADD is also considered here for a more comprehensive comparison, despite minor differences between the canister knobs.

Rotation Noise along Axis - Mask Loss In this research, rotation noise was incrementally introduced along the three axes (x, y, and z), ranging from 5 to 45 degrees in 5-degree steps and 90 degrees. Quantitative results for both scenes regarding rotation along the x-axis are displayed in Figure 5.6. This figure represents the percentage of images with lower ADD or ADI than 5mm in both scenes combined.

Figure 5.6 reveals that assuming an indistinguishable view angle for the canister (using ADI lower 5mm) leads to image convergence above 85% accuracy for both scenes when noise is less than 45 degrees. Notably, scene 1 achieves superior convergence with accuracy exceeding 98% for noise below 90 degrees, likely due to its lower complexity. However, the ADD curve diverges from this trend, significantly dropping accuracy as noise increases along the x-axis. For scene 1, accuracy remains above 60% with noise under 45 degrees, but scene 2 falls below 50% for all noise levels. This difference can be attributed to the camera's viewpoint, where one object can partially obscure the other in scene 2(Figure 5.7 initial iteration). In such cases, the estimated pose of one object might converge to the other's position (Figure 5.7 final iteration), resulting in a high ADD error. ADI remains lower in such scenarios because its error calculation considers the minimum distance between estimated and ground truth object positions, favouring estimated object poses. Additionally, the close proximity of the objects contributes to a lower ADI compared to ADD.

Moreover, Figure 5.6 illustrates that the method maintains the correct pose without

(a) Initial iteratio (b) Final iteration

Figure 5.7: Convergence of image 1 from scene 2 - Initial and final iteration of object mesh positions, with the final position closely matching the ground truth of the other object for rotation noise of 45-degree along x-axis.

degradation upon initialization with no noise in both scenes. However, none of the images from either scene would converge the rotation noise degree of 90 on the x-axis with an ADD-ADI error below 5mm. In general, none of these scenes would converge for 90-degree noise along the x-axis. This is because, with a 90-degree rotation, there is minimal overlap between the estimated and ground truth mask images, making it difficult for our method to determine the direction towards the global optimum. Additionally, the lack of extra information the mask loss provides restricts the direction towards the global optimum. As a result, the optimizer moves randomly in search of the best pose, leading to no convergence for 90-degree noise along the x-axis in both scenes.

Figure 5.8 shows the quantitative results for both scenes when rotating along the y-axis. Along this axis, this method maintains the correct pose without degradation upon initialization with no noise in both scenes. Comparing the ADI and ADD curves reveals that the distinction between the knobs has the most significant influence along this axis, as evidenced by the more noticeable difference between the two metrics. In addition, compared to the rotation in the x-axis (Figure 5.6), the ADI error shows more robustness to noise along y-axis, resulting in accuracies above 90% for rotation noise below 90 for scene 2 and accuracies above about 99% for scene 1. The ADD accuracy curve does not follow the same pattern. The accuracy drops to 0% for both scenes after the noise degree is more than 20 for scene 2 and 40 for scene 1. The reason is that the canister's diameter is $45mm$, and placing one knob's estimated position on the other knob's place would result in a distance error of $40mm$. Moreover, since rotation along the y-axis rotates the canister in a way that the knobs' positions would rotate, it is tough for the method to differentiate these two knobs, resulting in displacing them since the overlap between the ground truth and estimated mask images remains high and results in lower loss.

Rotation noise of 90 degree along the y-axis converges better than the x-axis, although it is much more challenging than the other noise degrees and is still poor. For scene 1, the ADI error below 5mm achieves the accuracy of 25% and for scene 2 the accuracy below 5%.

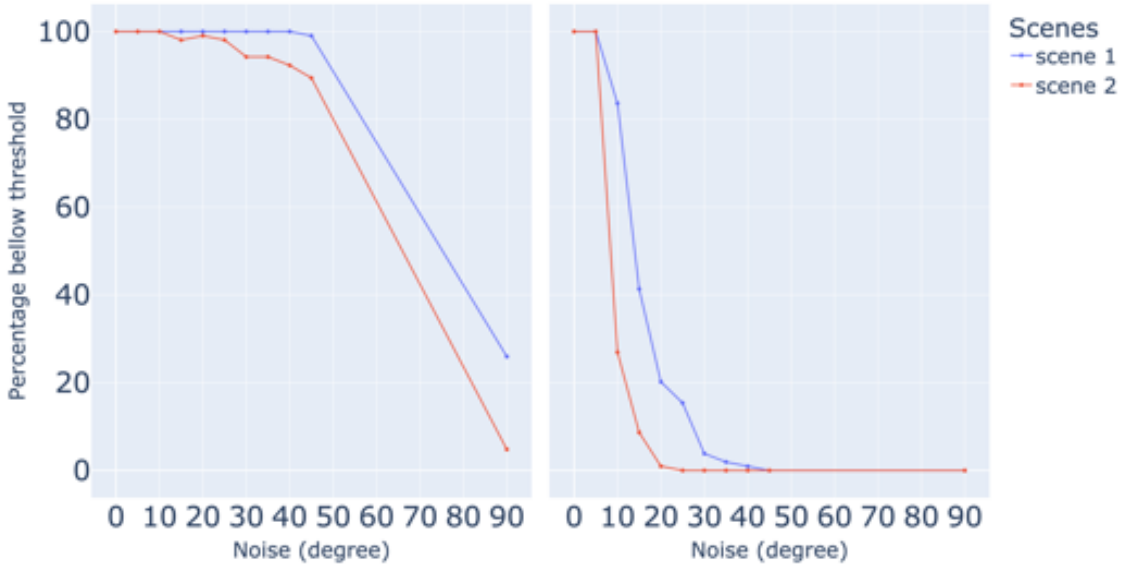


Figure 5.8: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the y-axis using L_{SDU}.

Finally, the results with noise along the z-axis are examined as shown in Figure 5.9. This axis displays greater noise robustness compared to the other two. Scene 1 with ADI below 5mm reaches the accuracy of 100% for all the rotation noises below 90 degree and 90% for the rotation noise 90 degree, which is remarkable compared to the other axes. Same for the scene 2, rotation noise of 90 degree results in 60% accuracy for ADI below 5mm. Noise along z, both high and low, leads to more mask overlap than noise along the other axes, which leads to prime robustness and convergence along z. The difference between the ADI and ADD in this axis is not as much as for the y-axis because, by rotation along this axis, the knobs' position does not get replaced by each other's positions. Their positions move along the axis. Therefore, there is less confusion.

An overall overview of all the rotation noises along all the axis, this method maintains the correct pose without degradation upon initialization with no noise. As a general pattern, firstly, none of the images from both scenes would converge the rotation noise degree of 90 on the x-axis. As discussed previously, a 90-degree rotation results in a minor mask image overlap, making it challenging for the method to determine the direction toward the global optimum. Secondly, the ADD value decreases for both scenes as the rotation error increases. However, there are exceptions due to the randomness of the optimizer's direction. Thirdly, comparing the accuracy of both measures for the two scenes reveals that scene 2 is more complex and more challenging to achieve high accuracy than scene 1. Finally, comparing results across axes, decreasing robustness towards noise is found; first, the z-axis, then the y-axis, and lastly, the x-axis. This implies the importance of mask image overlap in locating global optima since z-axis rotations with the most overlap show the highest robustness.

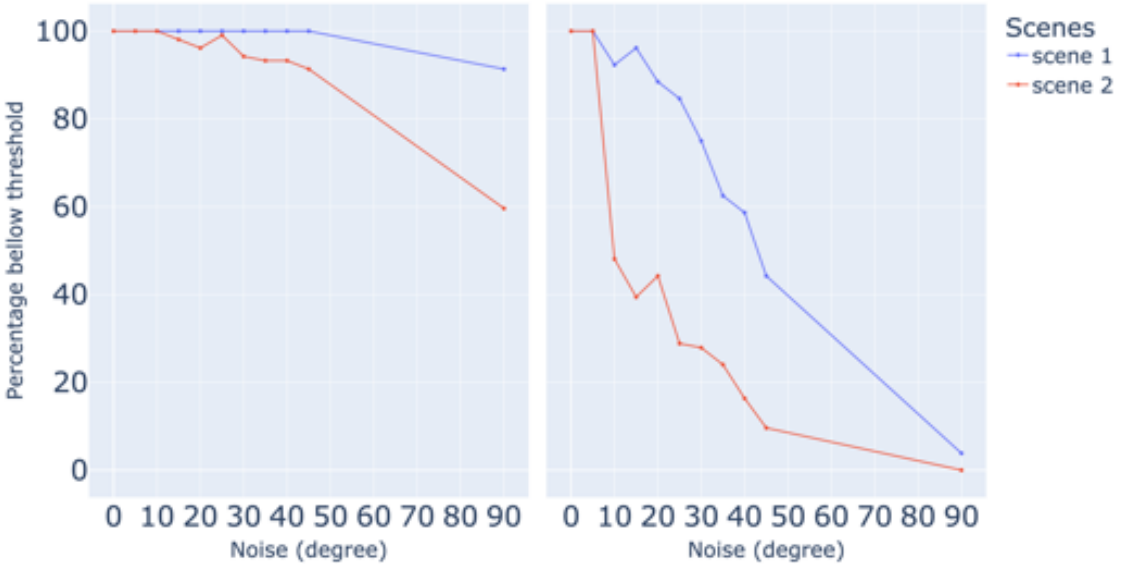


Figure 5.9: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the z-axis using L_{SDU}.

Translation Noise along Axis - Mask Loss In this analysis, translation noise was incrementally introduced along the three axes (x, y, and z), ranging from 10 to 80 mm in 10-mm steps. Quantitative results for both scenes regarding translation along the x-axis is displayed in Figure 5.10. This figure shows the accuracy decreases after more than 50mm of noise for both scenes. This is due to the method converging only when there is an overlap between the reference and estimated mask images. The canister's diameter is 45mm. Hence, any noise beyond that results in no overlap. Consequently, the optimizer drifts randomly and fails to find the optimal solution without image overlap for all images. For scene 2, after 50mm of translation noise, the situation becomes even more challenging. In this scene, there are two canisters placed close to each other. By introducing translation noise along the x-axis to both objects, there are some views where the initial pose of one object overlaps with the ground truth position of the other, leading to converging on the position of the second object instead of the correct one. Figure 5.11 demonstrates the situation for image number 20, with an initial translation noise of 60mm.

For translation noise under 50mm, the correct pose can be obtained with an accuracy of over 90%, where the scene 1 has an ADD less than 5mm. The accuracy for the same situation with ADI metrics reaches 100%. However, for scene 2, although accuracy above 98% is achieved for ADI below 5mm and translation noise less than 50mm, the ADD for the same situation has an accuracy below 60%. As previously discussed, the difference is due to the definition of ADD-ADI and the two canister knobs, as well as the partial coverage of one canister by the other due to the camera view. As observed, the translation along the x-axis is not robust and depends on the objects' diameters and mask overlap.

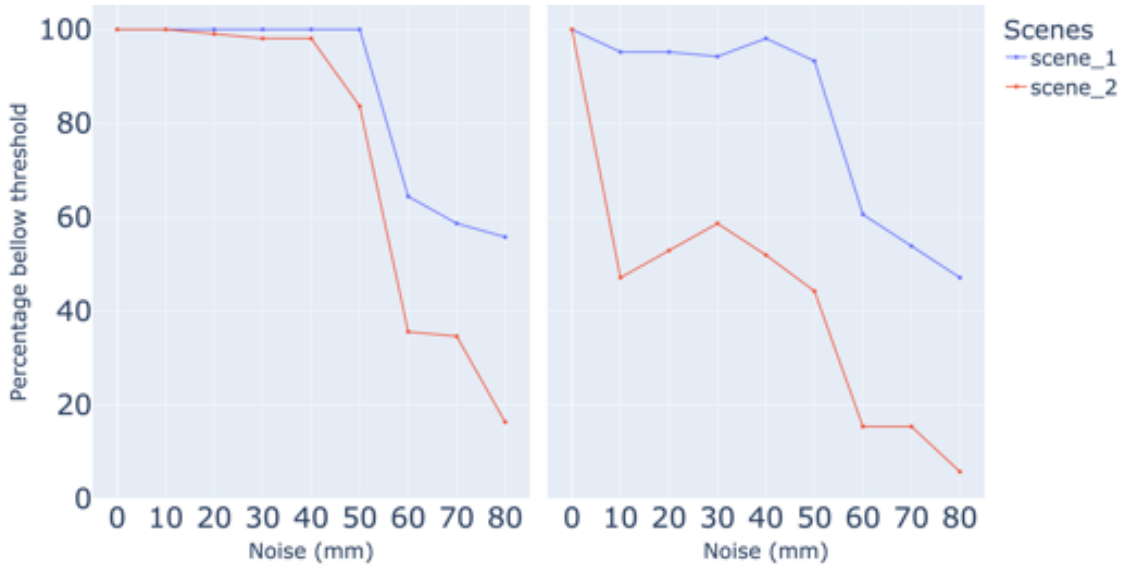


Figure 5.10: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the x-axis using L_{SDU}.

(a) Initial iteratio (b) Final iteration

Figure 5.11: Convergence of image 20 from scene 2 - Initial and final iteration of object mesh positions, with the final position closely matching the ground truth of the other object for translation noise of 60mm along x-axis.

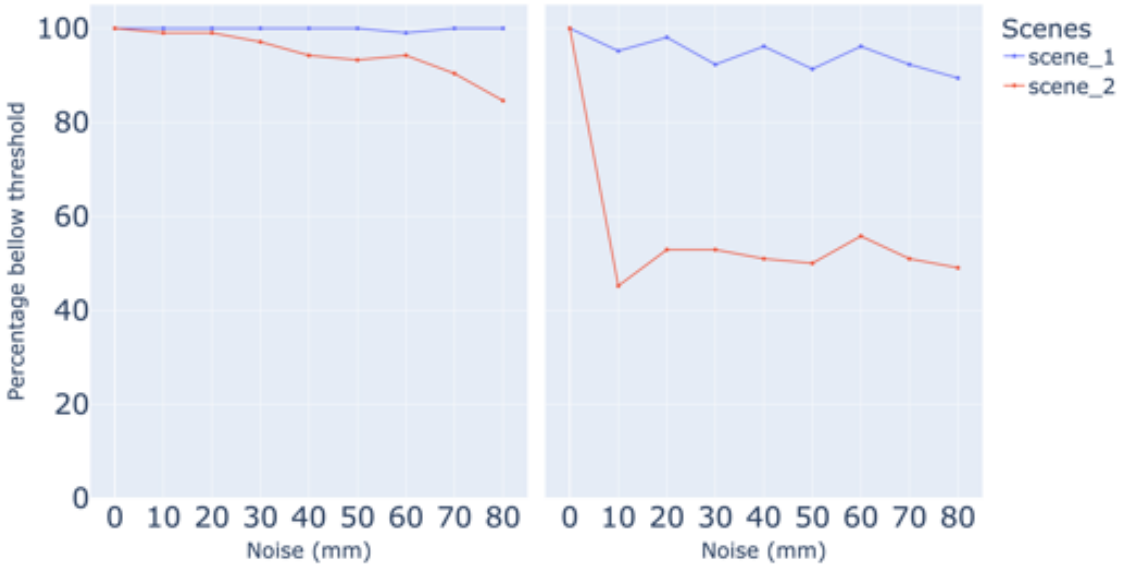


Figure 5.12: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the y-axis using L_{SDU}.

Figure 5.12 shows the accuracy curves for the translation noise along the y-axis having ADD-ADI below 5mm. For scene 1, the accuracy of the images converging with ADI below 5mm reaches the accuracy of 100%. Moreover, for the ADD below 5mm reaches the above 90%, this method can recover noise along the y-axis well for the first scene. For scene 2, the accuracy curve for ADI below 5mm decreases by increasing the translation noise, although it stays above 80% accuracy. For the accuracy curve having ADD below 5mm, the method can recover the correct pose with the accuracy below 60%. There is a considerable gap between the ADD and ADI accuracy curve for the scene 2. The reason behind this is that for some views like the one shown in Figure 5.13, after introducing the translation error along the y-axis, since the initial mask images of each objects has overlap with the ground truth mask images of the others, the method would not necessary converges on the correct position or as it shows in the Figure 5.13 last iteration, the method would try to just fit the objects within the scene in a way to have similar pixel-wise white spaces in the estimated mask image as the ground truth mask image. In the end, the method's ability to withstand translation noise along the y-axis can be observed in Figure 5.12. This is because the height of the canister is larger than the applied noise, resulting in the mask's overlap over all the noises and the method's convergence for all the scenes. However, it is essential to note that mask overlap plays a significant role in the convergence of this method.

Figure 5.14 shows the accuracy curves for translation noise along the z-axis. The figure displays the ADD-ADI values below 5mm. As it can be seen, scene 1 shows a similar pattern to the noise along the y-axis for both ADI and ADD measurements. However, in scene 2, the accuracy for images with ADI less than 5mm remains over 95%, and even

(a) Initial iteratio (b) Last iteration

Figure 5.13: Convergence of image 2 from scene 2 - Initial and final iteration of object meshes positions for translation noise of 80mm along y-axis.

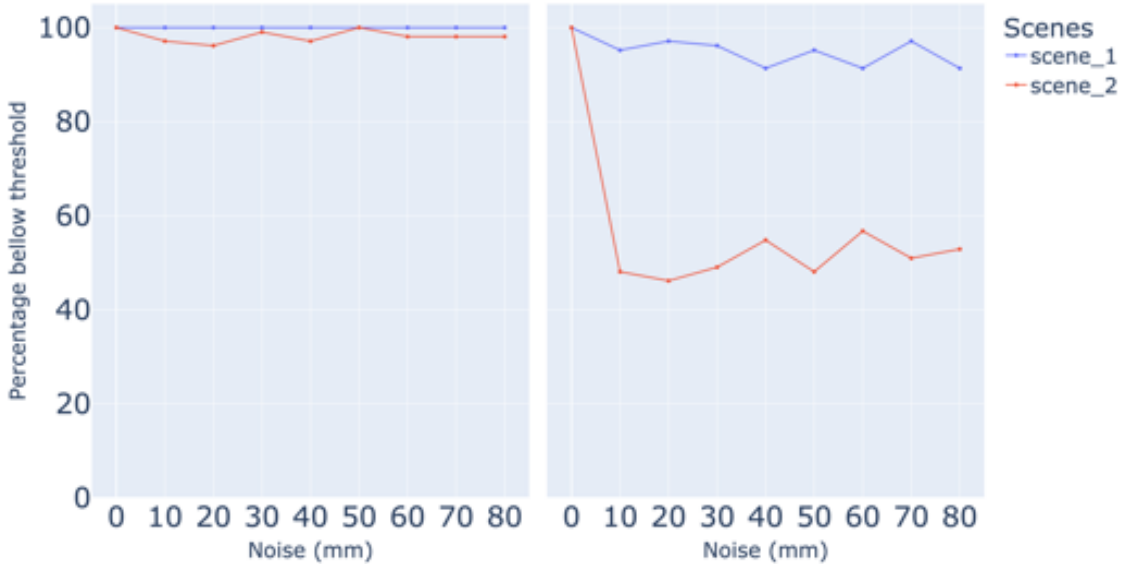


Figure 5.14: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the z-axis using L_{SDU}.

after a translation noise of 60mm, the accuracy remains the same. On the other hand, the accuracy curve for ADD values below 5mm behaves similarly to the translation noise along the y-axis. This method is robust to translation noise along the z-axis, similar to the y-axis because as we move along the z-axis, the ground truth and the initial mask image have the most overlap. By getting closer to the camera, this is because the object covers more pixels in the mask image. Conversely, by running away from the camera, it covers fewer pixels. Therefore, the method can recover the error better than along the x-axis.

5.1.2 Mask Loss Combined with Signed Distance

Here, the experiment continues by studying the Masked Signed Distance Loss (L_{MSD}). As defined in the previous chapter in eq 3.6, L_{MSD} combines the mask loss with signed distance with a balancing factor between the two. Based on the previous section, we chose the Symmetric Difference over Union (L_{SDU}) to contribute as the mask loss since it outperformed other mask loss calculations. The experiment starts by examining the different balancing factors for the signed distance contribution to the whole loss (γ) from the interval $[1, 0.001, 0.0001]$ by introducing noise to the translation matrix ($[0, 0, 50mm]$) and the rotation matrix along the x-axis with 45 degrees, applied to the ground truth pose. Subsequently, the outcomes across three scenes and two varying learning rates, $[0.02, 0.015]$, are compared.

Figures 5.15, 5.16, and 5.18 present a comparative analysis of the performance of different γ values for each scene in the dataset using ADI 2.2 metrics.

Upon examining Figure 5.15, we can see that L_{MSD} with $\gamma = 0.001$ and learning rate 0.02 produced the best results for scene 1. It had a lower average ADI and a smaller range. In contrast, using $\gamma = 1$ to value the signed distance resulted in an inconsistent loss definition with a very high average ADI and long range. Similarly, for scene 2, L_{MSD} with $\gamma = 0.001$ also produced the best results with a lower average ADI and shorter range, as shown in Figure 5.16.

In the third scene, results depicted in Figure 5.18, the method fails to converge with L_{MSD}, much like it did with L_{SDU} (refer to Figure 5.3). Although incorporating the signed distance into the Mask loss definition somewhat improves the outcome with L_{MSD} and $\gamma = 1$ at a learning rate of 0.015, the fundamental similarity between L_{MSD} and L_{SDU} means the method still fails to converge. As previously discussed, scene 3 comprises two canisters and a drain tray in close proximity, with the canisters mounted on the tray. Emphasizing that physical constraints help the method for better convergence. Interestingly, a lower learning rate produces better results for this more complex third scene.

The Figure 5.18 illustrates that a better loss calculation can be achieved by combining the L_{SDU} with signed distance, which is also known as L_{MSD}. This combination results in a lower average and shorter range ADI for the first two scenes. However, it is essential to note that this loss combination only addresses physical constraints and does not solve the issue with the view, as seen in Figure 5.5.

Convergence Basin Analysis - Adding Signed Distance

Much like the Mask Loss 5.1.1, experiments are undertaken to evaluate the L_{MSD}'s ability to manage and converge with errors in translation or rotation matrices, as well as to evaluate the accuracy of its convergence. Since the L_{MSD} did not work for the scene 3 and the L_{MSD} with learning rate 0.02 and $\gamma = 0.001$ was performing better than other configurations, here studies the first two scenes, using L_{MSD} with learning rate 0.02 and

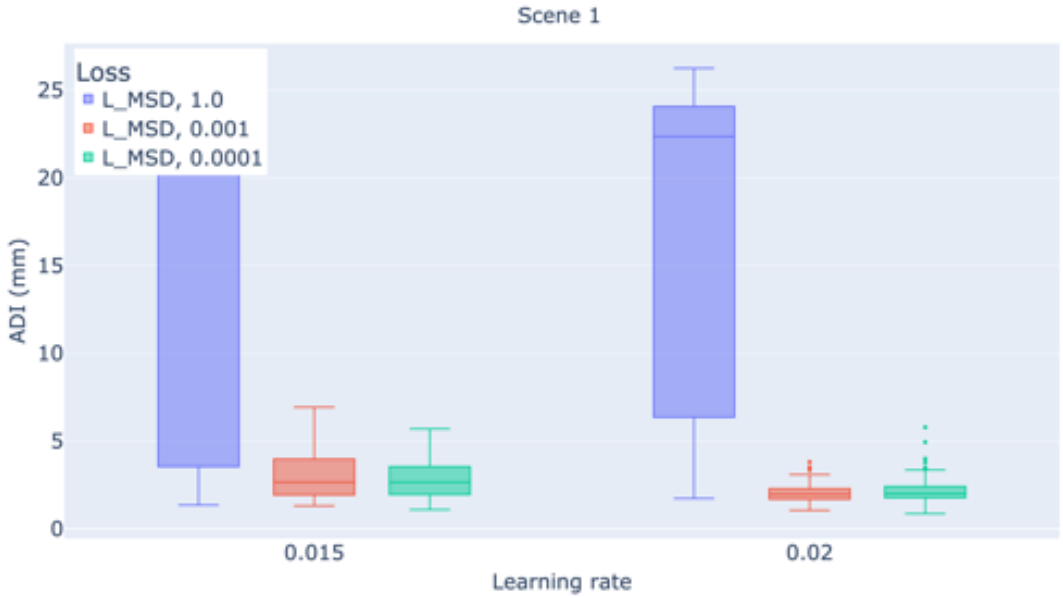


Figure 5.15: The ADI results of different γ values in L_{MSD} calculations using two distinct learning rates, 0.015 and 0.02 for scene 1. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

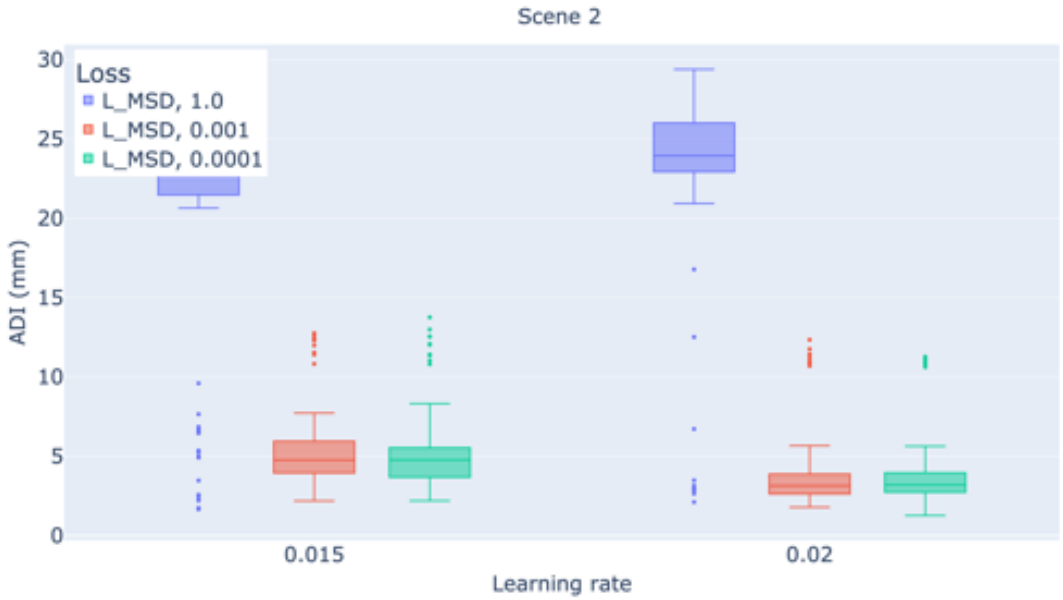


Figure 5.16: The ADI results of different γ values in L_{MSD} calculations using two distinct learning rates, 0.015 and 0.02 for scene 2. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

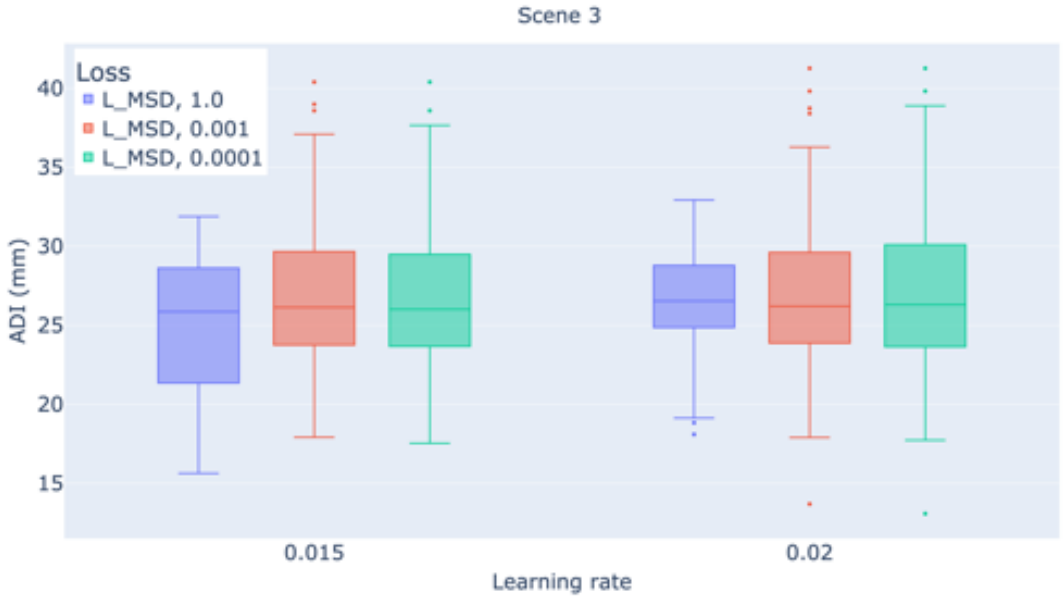


Figure 5.17: The ADI results of different γ values in L_{MSD} calculations using two distinct learning rates, 0.015 and 0.02 for scene 3. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

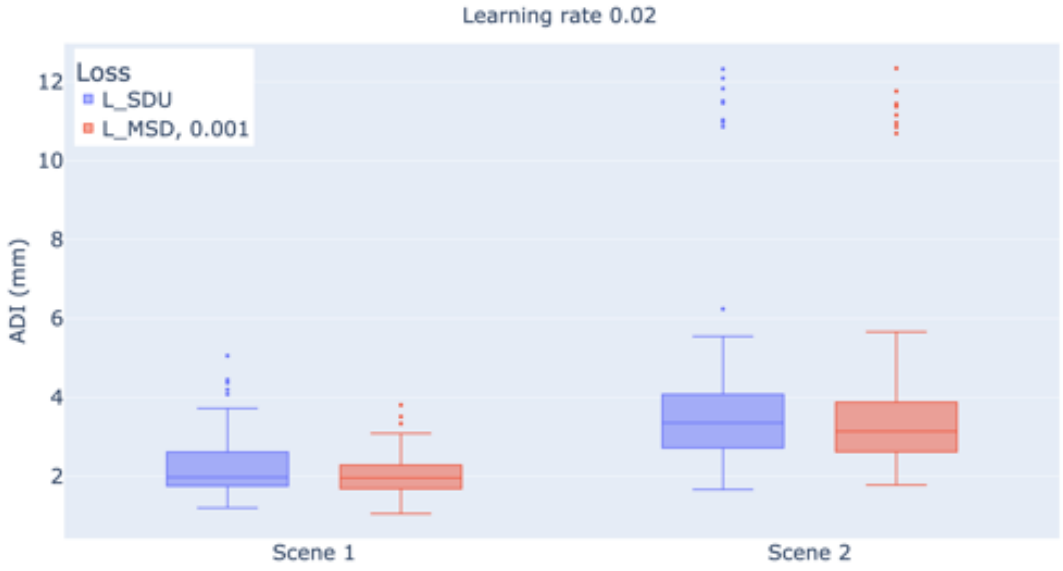


Figure 5.18: This figure illustrates the ADI result of L_{SDU} and L_{MSD} for scene 1 and scene 2

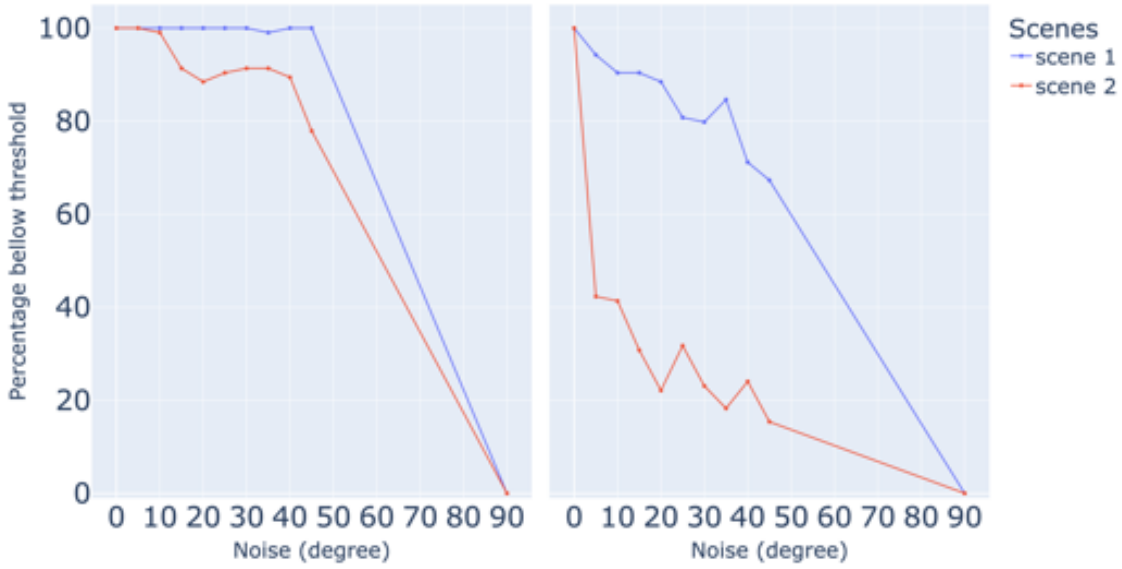


Figure 5.19: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the x-axis using L_{SDU}.

$\gamma = 0.001$. Both ADD and ADI are utilized for the noise influence illustrations. Despite the minor differences between the canister knobs, ADD is also considered for a more comprehensive comparison.

Rotation Noise along Axis - Adding Signed Distance In this experiment, rotation noise was incrementally introduced along the three axes (x, y, and z), ranging from 5 to 45 degrees in 5-degree steps and 90 degrees. Quantitative results for both scenes regarding rotation along the x-axis is displayed in Figure 5.19. It has been demonstrated that it maintains the correct pose in both scenes when our pipeline starts with no noise added to the ground truth, and we utilize the L_{MSD}. Additionally, it does not converge for the 90 degree rotation noise along the x-axis.

In the first scene results, for all rotation errors that are lower than 90 degrees, this method achieves an accuracy above 99% for an ADI error below 5mm, which is similar to the results obtained from the ablation study for mask loss 5.1.2 for rotation noise along the x-axis. However, in this scene, the accuracy for ADD lower than 5mm has a decreasing curve except for rotation noise of 35 degrees. This is interesting, especially compared to the results of the accuracy curve adding rotation noise along the x-axis using L_{SDU} in the mask loss ablation study 5.1.2 (Figure 5.6), where the accuracy drops below 80%, while here it remains above 80%. Table 5.2 presents the average ADD values for the scene 1 where rotation noise of 35 degree along the x-axis was added. The table shows the results for different camera heights. As illustrated, the maximum difference in the results is below 0.30mm. In addition, the table demonstrates that using signed distance positively affects having a lower error on a higher camera position for scene 1.

Loss definition	Scene 1				Scene 2			
	H1	H2	H3	H4	H1	H2	H3	h4
L_{SDU}	3.49	3.86	4.03	5.03	7.91	6.14	6.21	7.39
L_{MSD}, $\gamma = 0.001$	3.60	3.55	4.01	4.79	7.52	6.80	6.19	7.61

Table 5.2: Average ADD with 35 degree rotation noise along x-axis for both scenes.

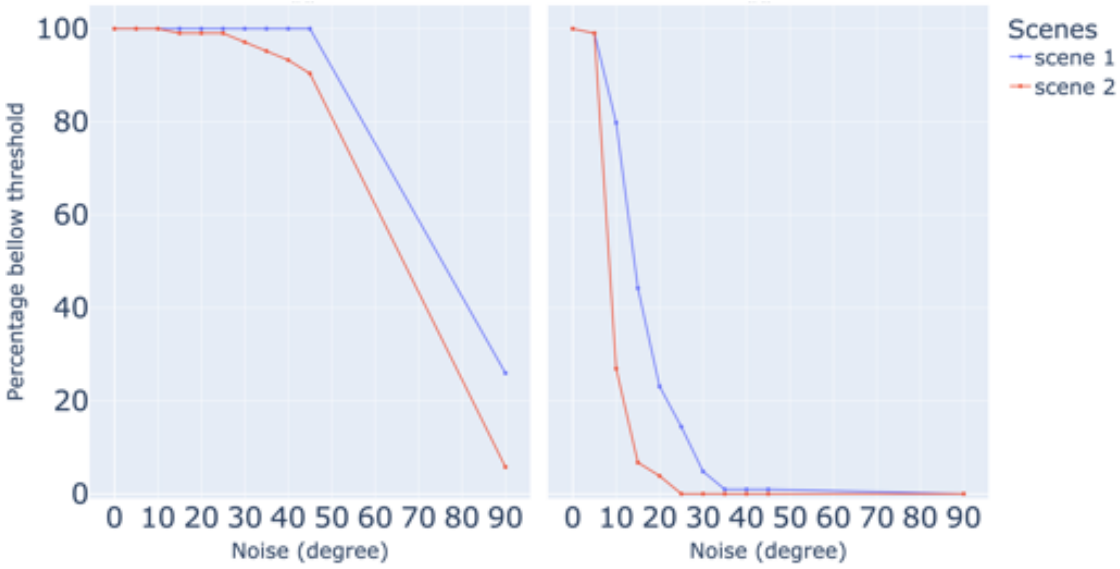


Figure 5.20: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the y-axis using L_{MSD}.

For the scene 2 results, the accuracy curve of ADI below 5mm shows a decreasing pattern and achieves above 80% accuracy for the noise below 45 degree, which is similar to the results of the rotation noise study along the same axis using L_{SDU}. The results for the ADD curve are also similar but with a higher decreasing slope. Specially for noise degree of 35, where previously using L_{SDU} method achieve the accuracy above 20% (Figure 5.6), but with L_{MSD} below 20%. The results of the average ADD values for this scene where rotation noise of 35 degree along the x-axis is also presented in Table 5.2. The results illustrate that for the second and fourth heights, the signed distance results obtain higher ADD but lower ADD for the other heights. This inconsistency could also be due to the optimizer's random movements for finding the global optima.

Quantitative results for both scenes regarding rotation along the y-axis is displayed in Figure 5.20. It has been demonstrated that it maintains the correct pose in both scenes when our pipeline starts with no noise added to the ground truth, and we utilize the L_{MSD}. Similar to the ablation study using mask loss only (Figure 5.8), comparing the ADI and ADD curves reveals that the distinction between the knobs has the most significant influence along this axis, which was predictable. Adding signed distance to the mask loss definition, only adds physical constrains and had no effect on distinguishing

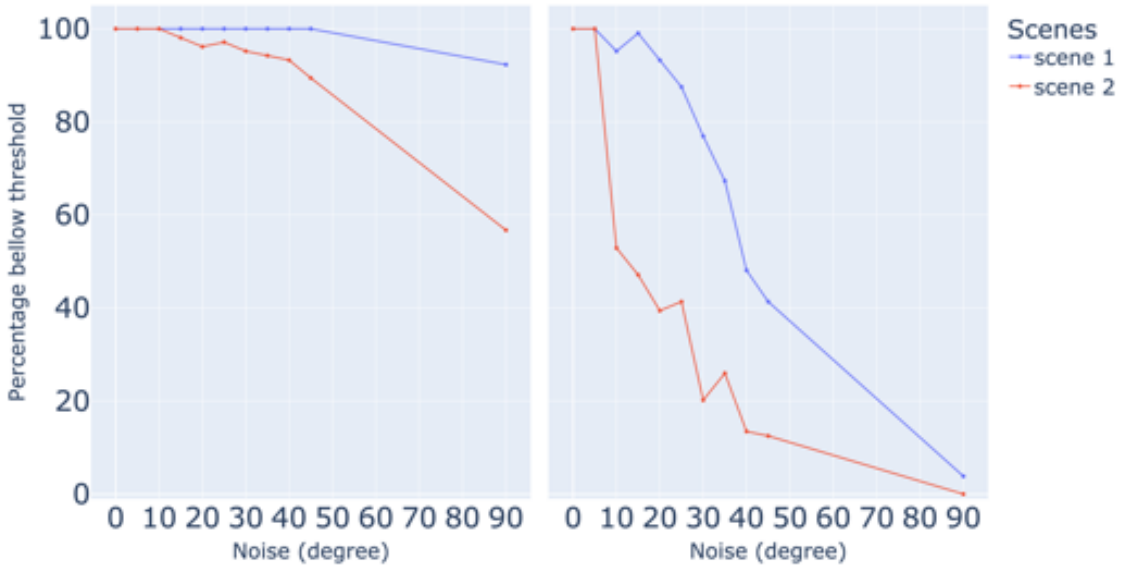


Figure 5.21: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to rotation noise along the z-axis using L_{MSD}.

the canister's knobs better.

The accuracy of ADD drops to 0% for both scenes when the noise degree is more than 25 for scene 2 and 35 for scene 1. This occurs because the diameter of the canister is $45mm$, and if one knob's estimated position is placed on the other knob's location, there is a distance error of $40mm$. Additionally, since rotating along the y-axis causes the canister to rotate in a way that the knob positions rotate, it becomes difficult for the method to differentiate between the two knobs, resulting in displacement. The overlap between the ground truth and estimated mask images remains high, resulting in a lower loss. Rotation noise of 90 degrees along the y-axis converges better than the x-axis, although it is much more challenging than the other noise degrees and is still poor. For scene 1, an ADI error below 5mm achieves an accuracy of 25%, and for scene 2, the accuracy is below 5%.

Lastly, the results for both scenes regarding rotation along the z-axis are examined, as displayed in Figure 5.21. For the result of ADI below 5mm, scene 1 achieves the accuracy of 100%, and scene 2 the accuracy above 90% for noise degree below 90. In contrast to the other axis results on 90 degree rotation noise, both scenes achieve the accuracy above 80% for the ADI below 5mm. For the ADD metrics, the accuracy drops for both scenes. Upon comparing the ADD-ADI figures along this axis with the results presented in the ablation study for mask loss (Figure **??**), the similarities between the two loss configurations are observed. It is noted that they do not have a significant impact on convergence. However, it is emphasized that the mask overlap continues to play a crucial role in the method's convergence.

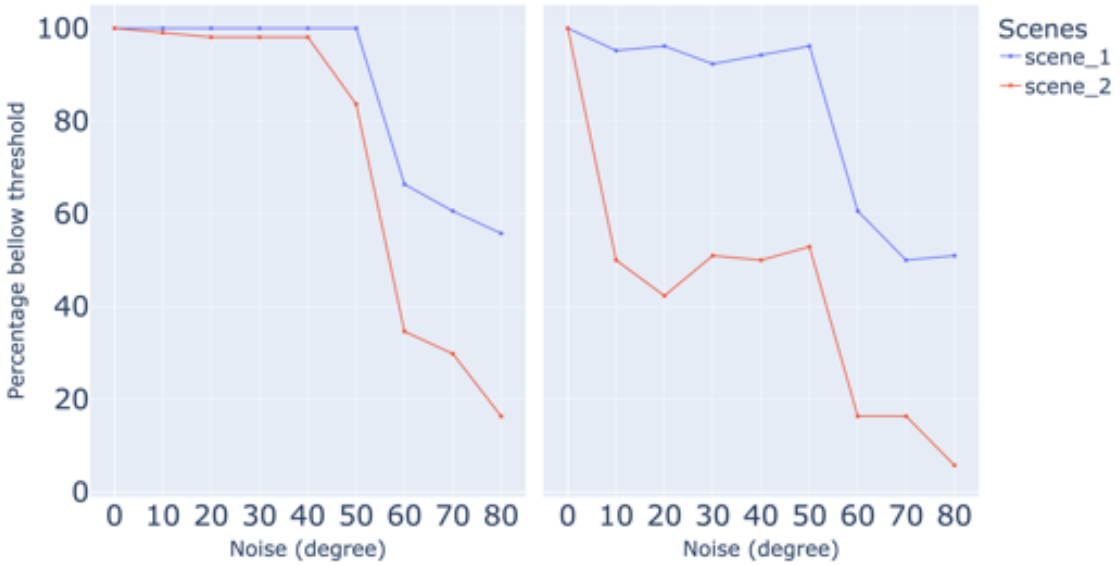


Figure 5.22: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the x-axis using L_{MSD}.

Translation Noise along Axis - Adding Signed Distance In this research, translation noise was incrementally introduced along the three axes (x, y, and z), ranging from 10 to 80 mm in 10-mm steps. Quantitative results for both scenes regarding translation along the x-axis, y-axis, and z-axis are displayed in Figures 5.22, 5.23, and 5.24 respectfully.

After analyzing the ADD-ADI values on the axes related to the figures presented in the ablation study for mask loss concerning translation error, it has been observed that both loss configurations are similar and do not have a significant impact on the convergence. However, L_{MSD} shows better robustness towards translation error. Nevertheless, it is still essential for the mask overlap to be considered, as it plays a crucial role in the method's convergence.

5.1.3 Separate Object Optimization

In the previous two sections, the optimization process treated the scene as a whole without distinguishing the position of each object within the mask images. This optimization approach led to the convergence of one object into the position of others, as illustrated in Figures 5.7 and 5.11. As a result, optimization was performed individually for each object within the scene using objects' separate visible mask images. Visible mask images in this context refer to the mask image that shows the parts of an object that are visible to the camera. If another object partially covers an object, the mask image of the second object will not include the covered part. The same loss calculation as the previous part, L_{MSD} with $gamma = 0.001$, has been used for this experiment. Figure 5.25 illustrates the ADI results of optimizing a scene united or per object using L_{MSD} with $\gamma = 0.001$

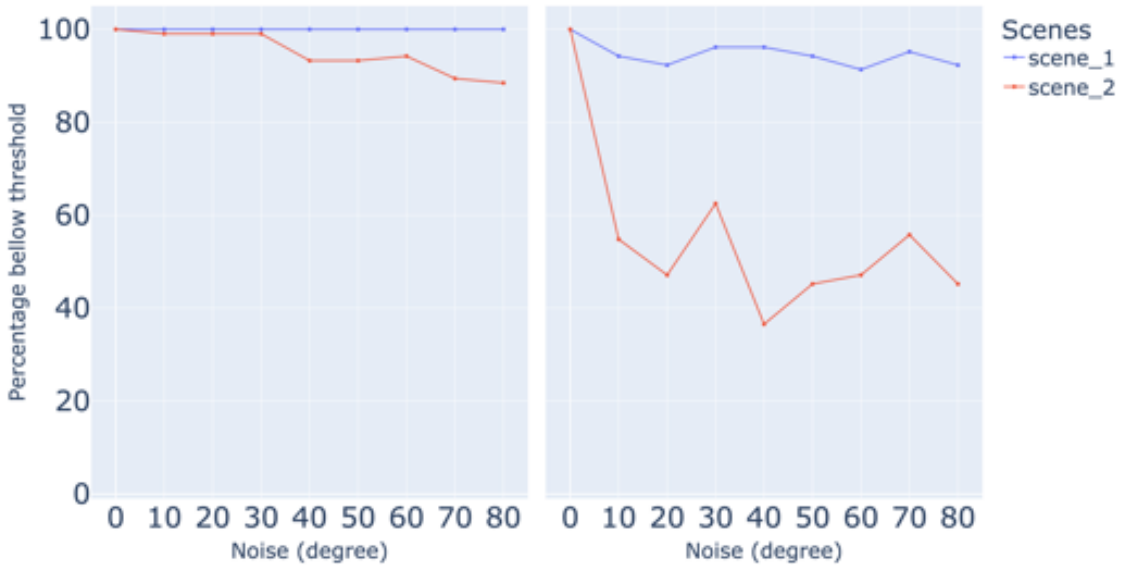


Figure 5.23: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the y-axis using L_{MSD}.

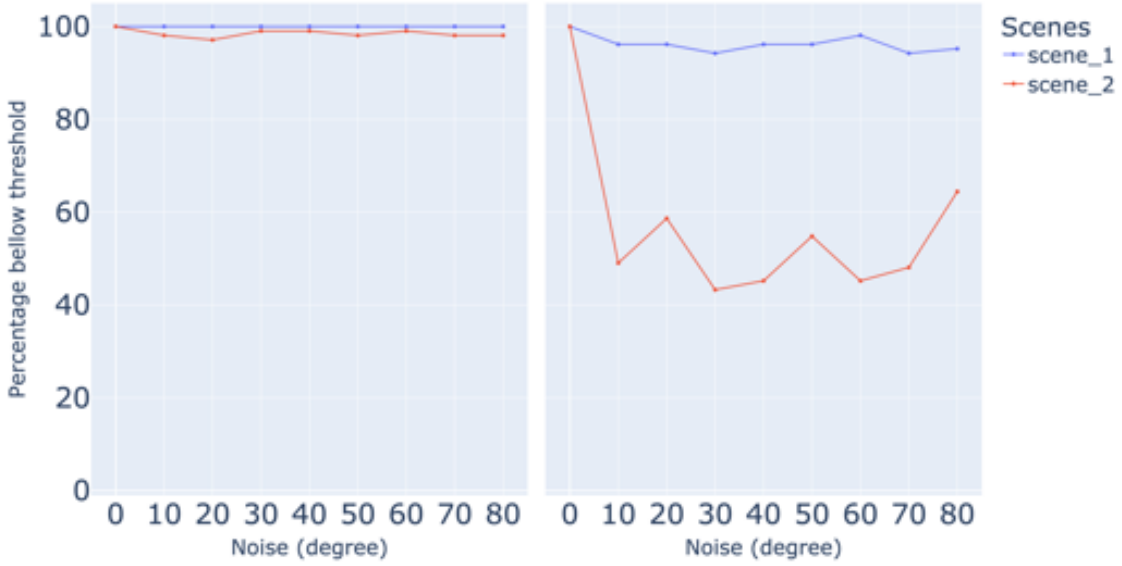


Figure 5.24: The accuracy curve for the first two scenes having ADI (left) and ADD (right) below 5mm showing response to translation noise along the y-axis using L_{MSD}.

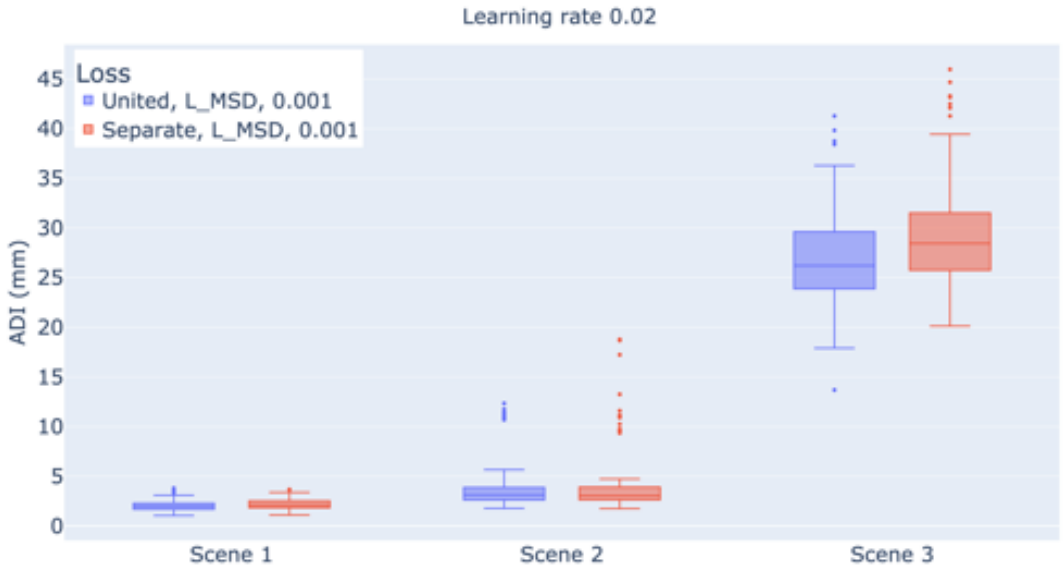


Figure 5.25: The ADI results of optimizing a scene as a whole or per object using L_{MSD} with $\gamma = 0.001$ for all three scenes. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

for all three scenes. The pipeline was initialized with 50mm of translation noise along the z-axis and 45 degree of rotation noise around the x-axis, applied to the ground truth pose.

As shown, optimizing objects separately within a scene yields lower ADI for scenes two and three. This technique does not affect scene one, which only contains one canister. Moreover, convergence is not achieved for scene 3 as the loss calculation in both techniques is identical. Consequently, the limitations of the pipeline continue with this technique. Therefore, the convergence in case of no overlap in mask images is similar to the previous subsection. Scene 1 is excluded from the experiments in this part since it only contains one object. Therefore, the results are identical to the earlier sections.

The results of the robustness of optimizing each object separately towards rotation noise compared to considering the scene as a unit is illustrated in Table 5.3 for scene two. This table shows the accuracy of estimations having ADI-ADD below 5mm for the rotation noise from 5 to 45 degree with 5 degree steps and 90 degree applied to all three axes. Similar to optimizing the scene as a unit, this optimization does not converge when applying a 90 degree rotation noise along the x-axis. However, it shows higher accuracy for 45-degree noise rotation along this axis, which is due to solving the problem depicted in Figure 5.7. Additionally, Table 5.3 shows that this configuration maintains the correct pose by starting the pipeline with no zero.

Noise (degree)	ADI						ADD					
	x		y		z		x		y		z	
	S	U	S	U	S	U	S	U	S	U	S	U
0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
5	100.00	100.00	100.00	100.00	100.00	100.00	46.15	42.31	100.00	99.04	99.04	100.00
10	99.03	99.04	100.00	100.00	100.00	100.00	30.76	41.35	35.57	26.92	50.96	52.88
15	93.26	91.35	99.03	99.04	93.26	98.08	40.38	30.77	7.69	6.73	55.77	47.12
20	93.26	88.46	99.03	99.04	94.23	96.15	39.42	22.12	1.92	3.85	46.15	39.42
25	84.42	90.38	96.15	99.04	92.30	97.12	37.50	31.73	1.92	0.00	36.54	41.35
30	93.26	91.35	92.30	97.12	94.23	95.19	25.96	23.08	0.00	0.00	32.69	20.19
35	90.38	91.35	92.30	95.19	94.23	94.23	23.07	18.27	0.00	0.00	25.00	25.96
40	90.38	89.42	91.34	93.27	92.30	93.27	21.15	24.04	0.00	0.00	22.12	13.46
45	80.76	77.88	90.38	90.38	92.30	89.42	14.42	15.38	0.00	0.00	19.23	12.50
90	-	-	5.7	5.7	74.03	56.73	-	-	0.00	0.00	0.00	0.00

Table 5.3: Illustrating the accuracy of estimations after rotational noise injection having ADI-ADD below 5mm for two different optimization techniques. In this table, "S" stands for separately optimizing objects and "U" for optimizing the scene as a unit.

As a general pattern, this optimization does not have a massive effect on the method's robustness considering ADI metrics compared to the previous optimization method. However, it shows a higher accuracy in estimation regarding the ADD metrics, specifically regarding rotation along the y-axis. The reason behind the minor difference between the two techniques is that they have the same loss foundation. However, using distinct object optimization ensures that the convergence of one object in the ground truth position of the others does not occur, leading to more reliable results.

Regarding the injection of translation noise along the three axes (x, y, and z), ranging from 10 to 60 mm in 10-mm increments, the accuracy of estimations having ADD-ADI below 3mm for the separate object optimization technique are presented in comparison to optimizing the entire scene as a unit in Table 5.4.

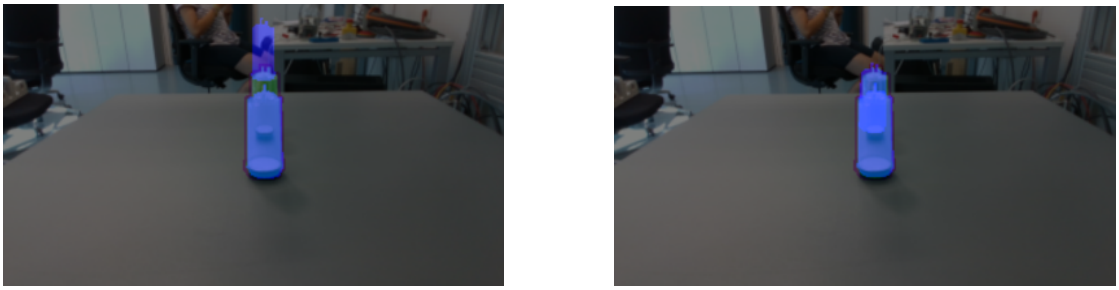
Comparing the results of two techniques along the x-axis shows that optimizing each object separately positively increases the accuracy of estimations having both ADD-ADI below 3mm. This optimization technique solves the problem of 60mm translation error in situations like 5.11, yielding in more accurate estimation.

When comparing the results of two techniques on the y-axis, it was observed that optimizing each object separately can increase accuracy, but only in some cases of applied noise values. For instance, when there was 10mm translation noise along this axis, 57.69% of all the separately optimized estimations had an ADI below 3mm. At the same time, this value increased to 72% when optimizing the scene as a whole.

This occurs because when one object is covered by another, it impacts its mask image by reducing the number of white pixels since the covered portion requires fewer pixels. A similar effect occurs in the ground truth visible mask, leading to two mask images

Noise (mm)	ADI						ADD					
	x		y		z		x		y		z	
	S	U	S	U	S	U	S	U	S	U	S	U
0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
10	69.23	57.69	57.69	72.12	57.69	63.46	4.81	2.88	2.88	2.88	5.77	1.92
20	64.42	56.73	64.42	59.62	70.19	69.23	1.92	2.88	1.92	4.81	0.00	3.85
30	65.38	66.35	65.38	74.04	68.27	53.85	2.88	0.96	0.96	3.85	3.85	0.96
40	63.46	57.69	61.54	50.00	58.65	63.46	3.85	2.88	3.85	1.92	5.77	3.85
50	63.46	61.54	64.42	57.69	65.38	66.35	5.77	6.73	2.88	1.92	6.73	0.96
60	32.69	22.12	70.19	59.62	68.27	61.54	1.92	0.96	6.73	4.81	6.73	1.92

Table 5.4: Illustrating the accuracy of estimations having ADI-ADD below 3mm for two different optimization techniques. In this table, "S" stands for separately optimizing objects, and "U" for optimizing the scene as a unit.



(a) Separate optimization (b) United optimization

Figure 5.26: Contrasting the final iteration of separately optimizing objects within a scene with the unified optimization of the entire scene in the convergence of image two from scene 2, it is observed that optimizing the scene as a whole provides more accurate pose estimation compared to separate optimization, for a translation noise of 10 mm along the y-axis.

with minimal white space that the method needs to align. In such instances, the method struggles to converge to the correct pose, resulting in higher ADD-ADI.

Figure 5.26 illustrates the final iteration of image two from scene 2 after applying 10mm of translation noise along y-axis. The united optimization technique results in the correct pose convergence, whereas the separate optimization technique encounters difficulties and displaces the object upward.

Optimizing each object within the scene separately relies on having additional information about the scene, the visible mask images of the objects. While this approach resolves the issue of one object converging to the ground truth position of another, it also has limitations, and its convergence is reliant on the distribution of white pixels in each object's visible mask image. Given that, on average, this approach produces more reliable results, the subsequent two sections employ the separate optimization technique.

5.2 Ablation Study - Heuristic

Thus far, as described in section 3.1, the method initialized by adding noise to the ground truth poses to find the best modification for the pipeline using different loss calculations. This chapter examines the result of starting the method using heuristic on the three scenes using the best loss calculation from the previous section, L_{MSD} with $\gamma = 0.001$ using separate optimization technique. As explained earlier in section 3.1, the initialization process simultaneously starts with four distinct stable poses for each object. Ultimately, the best result is chosen as the output. Figure 5.27 shows the ADI result of the three scenes in the Tracebot dataset. As can be seen, the method does not converge in any of the scenes using heuristics.

The heuristic randomly selects a stable pose from all available poses for an object's initialization to keep it general. However, hypothesis clustering could perform more effectively. Figure 3.2 showed the drain tray's stable poses, for example. For the canister, the situation is the same. Moreover, since lying down on the side is more stable than standing, and the canister could lay down in any direction, the chances to choose the standing stable pose are less than sitting on the side. As we discussed earlier in the ablation studies in the previous sections, laying on the side, meaning 90 degree rotation from the ground truth pose, leads to less mask image or no mask image overlap, which makes the pipeline not converge.

Moreover, since the object can be in any position within the scene, we cannot know with which stable pose to start the method. Having no other loss definition to guide the optimizer towards the optimum solution leads to the method being unable to converge in this situation. This pipeline is designed to refine the results obtained from a pose estimator, and it cannot estimate the position of each object within the scene.

5.3 Comparison to State of the Art - BOP

This section evaluates the performance of the pipeline by comparing it to two state-of-the-art pose refiners selected from the BOP challenge 2020, Pix2pose [PPV19] and Megapose [LMM+22], on the multi-object scene T-Less dataset. Although the T-Less dataset lacks transparent objects, it features textureless objects, which are particularly challenging to refine regarding their poses. To facilitate comparison, Pix2pose is chosen for its ability to handle textureless objects. Pix2pose's refiner relies on an Iterative Closest Point (ICP) method to minimize the difference between two point clouds by matching objects' point clouds obtained from RGB-D images and their point clouds constructed from objects' mesh after applying the estimated pose. Megapose is selected for its ability to work with novel objects, which are objects that were not seen during training. The Megapose refinement stage begins after the coarse pose estimation selects the best template. Afterward, Megapose refinement generates RGB and depth images of each object's pose within the template and iteratively refines the estimation. The refinement process involves choosing the template that yields the lower error. The refinement results

Figure 5.27: The ADI result of heuristic initialization of the pipeline over three different scenes.

of the pipeline would be compared with:

- Pix2pose refinement [1] over its estimator results [2]. In this case, the transparent pose refinement pipeline would start the refinement from the pose estimation results of Pix2pose.
- Megapose refinement [3] over its estimator results [4]. In this case, the transparent pose refinement pipeline would start the refinement from the pose estimation results of Megapose.

For this evaluation, eight scenes from the T-Less dataset are selected, specifically scenes $[1, 2, 3, 7, 9, 10, 11, 12]$ utilizing the ground truth visible masks. This selection is because our method cannot handle objects resting on other objects. The signed distance algorithm prevents physical impossibilities and enforces objects resting on the table. Therefore, scenes as such are excluded from the comparison. The comparison between this pipeline and the other methods is presented in Table 5.5.

This pipeline achieves an average recall of 0.17 when applied to Megapose [LMM+22] coarse pose estimation, while Megapose's refinement achieved a recall of 0.51. Similarly,

[1] https://bop.felk.cvut.cz/method_info/50/
[2] https://bop.felk.cvut.cz/method_info/49/
[3] https://bop.felk.cvut.cz/method_info/122/
[4] https://bop.felk.cvut.cz/method_info/115/

	MegaPose Estimator		Pix2Pose Estimator	
	L_{MSD}	MegaPose	L_{MSD}	Pix2Pose
Average Recall	0.17	0.52	0.22	0.56

Table 5.5: Comparative Analysis of Average Recall Across eight Scenes from the BOP Dataset – This table showcases a side-by-side comparison of recall metrics: transparent pipeline results following initialization with the Megapose pose estimator versus the Megapose refiner, and transparent pipeline results post initialization with the Pix2pose pose estimator as opposed to the Pix2pose refiner.

when applied to Pix2pose PPV19 coarse estimation, this pipeline achieved an average recall of 0.32, compared to Pix2pose ICP refiner's recall of 0.58. The results illustrate that the transparent pose refinement pipeline cannot reach the state-of-the-art level in the context of opaque objects. However, RGB information is utilized in this pipeline, while the other two methods employed depth information, significantly assisting in the refinement process. They also trained a neural network model using natural and synthetic data. Moreover, Megapose employed two refinement techniques sequentially, both requiring training, while our approach was considerably less complex. The transparent pose refinement pipeline is the first to use RGB images to refine transparent objects. No other method exists for transparency refinement.

CHAPTER 6

Discussion

This thesis designs a pipeline to refine transparent objects' poses without relying on depth images or object categories. This method can operate without the necessity of large-scale data training and can handle novel objects. An ablation study is conducted to select the best parameters and setup to determine the optimal configuration of the method. This pipeline can work on a single image from an unseen new scene, using the object's mesh and mask images without prior training.

A series of experiments is conducted on the Tracebot dataset, which includes three scenes. This dataset is recorded and described in detail in Chapter 4. The results of the conducted experiments are as follows.

6.1 Mask Loss

Initially, the effects of injecting 45-degree rotation noise to the x-axis and 50mm translation noise along the z-axis to the ground truth pose while applying each loss calculation is evaluated. Additionally, two learning rates, 0.015 and 0.02, are chosen for each loss calculation. The ADI metric is selected for measurement, given the identical canister knobs. The results illustrate a lower average ADI for the L_{SDU} with learning 0.02. L_{SDU} calculates the symmetric difference between the estimated and ground truth mask images over their union. This experiment shows that the average ADI rises with increasing the objects within the scene as the scene becomes more complex. In addition, this loss proves inefficient in the third scene, where the objects are in close proximity with two canisters mounting on top of a drain tray.

Later, an ablation study is conducted to measure the robustness of this loss against noise along different axes. The approach shows the highest robustness to rotation noise along axes for scene one. Regarding the ADI error with a 5mm threshold and rotation noise below 45 degrees, over 98% of all estimates fall below that threshold. However, when

exposed to a 90-degree rotational noise, the method exhibits non-convergence along the x-axis, resulting in decreased accuracy along the other axes. The reason is that the overlap of the estimated and ground truth masks influences the method's convergence.

Regarding the translation noise in scene one along the x-axis, all estimates fall below the 5mm ADI error threshold with translation noise below 50mm. Above 50mm, if the error exceeds the object width and there is no overlap between the ground truth pose and the initial pose, the method's accuracy decreases and converges randomly. However, for the other two axes, the method shows robustness and results in ADI below 5mm for all estimations since the noise along the y and z axes maintains the overlap between the ground truth pose and the initial pose.

The second scene is more complex than the first one. As the number of objects per scene increases, this method's accuracy is significantly influenced by factors such as the orientation and position of the objects, along with the initial pose error. With a 5mm ADI error threshold and rotation noise below 45 degrees, over 90% of estimates are within the threshold. Upon exposure to a 90-degree rotational noise, similar to scene one, the method fails to converge along the x-axis, leading to decreased accuracy in the other axes.

Regarding the translation noise in scene two along the x-axis, above 80% of all estimates fall below the 5mm ADI error threshold with translation noise below 50mm. Above 50mm, similar to scene 1, if the error exceeds the object width and there is no overlap between the ground truth pose and the initial pose, the method's accuracy decreases and converges randomly. In contrast to the first scene, rotating along the y-axis is more challenging in this scene because one object's initial position may overlap with the other's mask image, leading to false convergence. The translation noise smaller than 80mm along the y-axis has ADI below 5mm for at least 80% of all estimations. The method is most robust along the z-axis for noise smaller than 80mm in scene 2, with over 95% of instances having an ADI below 5mm.

6.2 Mask Loss combined with Signed-distance

The signed distance loss calculation is introduced to prevent unrealistic object collisions and ensure a realistic spatial arrangement of object positions. This loss ensures that objects rest on the plane rather than floating in the air and aims to avoid placing one object within another. This calculation is added to the L_{SDU} with a balancing factor of γ. Similar to the mask loss, the effects of injecting 45-degree rotation noise to the x-axis and 50mm translation noise along the z-axis to the ground truth pose are evaluated on the loss definition while applying the three γ values. The three gamma values chosen are [1, 0.001, 0.0001]. Two learning rates, 0.015 and 0.02, are chosen for each setting. The results illustrate a lower average ADI for $\gamma = 0.001$ with learning rate of 0.02. The average ADI for the first two scenes is lower than the L_{SDU}. However, this loss also proves inefficient for the third scene since the signed distance definition cannot handle objects mounted on top of each other, not resting on the plane. Regarding the noise

injection, the loss behaves similarly to the L_{SDU} but shows more robustness for scene one.

In general, if there is no overlap between the ground truth and the initial estimate, the method encounters failure because it cannot leverage object-specific features, given that the initial estimate is utilized to identify the image region.

6.3 Separate Object Optimization

In mask loss and mask loss combined with signed distance experiments, the optimization process treats the entire scene as a singular entity without accounting for the individual positions of each object within the mask images. This approach results in the convergence of one object into the position of others, as demonstrated in Figures 5.7 and 5.11. Consequently, optimization is conducted separately for each object within the scene using their respective visible mask images. In this context, visible mask images refer to the mask image that depicts the parts of an object visible to the camera. If another object partially obstructs an object, the mask image of the second object will exclude the covered portion. The same loss calculation, L_{MSD} with $gamma = 0.001$, has been used for this experiment.

While this optimization technique effectively addresses the issue of one object converging to the position of another and, on average, yields results with lower ADI-ADD, it encounters limitations (Figure 5.26). This is because when one object is covered by another, it affects its mask image by reducing the number of white pixels since the covered portion requires fewer pixels. A similar effect occurs in the ground truth visible mask, resulting in two mask images with minimal white space that the method needs to align. In such instances, the method struggles to converge to the correct pose due to the limited information derived from the mask image.

6.4 Heuristic Initialization

This section evaluates the outcomes of the pipeline initiated using a heuristic. The heuristic relies on the stable poses of objects to initialize their rotation matrices and employs the objects' mask images and camera intrinsics for initializing the translation matrices. The initialization process starts with four distinct stable poses for each object simultaneously. Utilizing the separate optimization technique, this approach is tested on three scenes using the best loss calculation identified in the previous section, L_{MSD} with $\gamma = 0.001$. However, it is found to be insufficient for all three scenes.

The heuristic randomly selects a stable pose from all available poses for an object's initialization to maintain generality. For instance, a canister lying down on its side is more stable than standing, and the canister could assume any orientation while lying down. Therefore, the likelihood of selecting the standing stable pose is lower than lying on the side. Ablation studies in preceding sections indicate that lying on the side, implying

a 90-degree rotation from the ground truth pose, results in less mask image overlap or no mask image overlap, preventing the pipeline from converging. Determining the appropriate stable pose to initiate the method becomes challenging since the object can be oriented differently in each scene. The absence of another loss definition to guide the optimizer toward the optimal solution prevents the method from converging in such situations. Nevertheless, it is essential to note that this pipeline is designed to refine results obtained from a pose estimator and does not have the capability to estimate the precise position of each object within the scene.

6.5 Comparison to State of The Art

In the evaluation of the T-Less dataset, this method achieves an average recall of 0.17 when applied to Megapose [LMM+22] coarse pose estimation, while Megapose's refinement achieved a recall of 0.52. Similarly, when applied to Pix2pose [PPV19] coarse estimation, this method achieves an average recall of 0.22, compared to Pix2pose ICP refiner's recall of 0.56. Even though these results are not competitive with current methods, not leveraging depth is a more complex problem, mainly due to scale and size invariance. However, this method can potentially close the gap towards RGB-D methods. Here, the RGB information is utilized, while the other two methods employ depth information, significantly assisting in the refinement process. Megapose [LMM+22] and Pix2pose [PPV19] also trained a neural network model using real-world and synthetic data. Moreover, Megapose employed two refinement techniques sequentially, both requiring training, while our approach was considerably less complex.

6.6 Limitations

The success of this pipeline depends on the degree of overlap between the estimated and ground truth masks. In cases where there is no overlap, the method fails to converge. Consequently, the pipeline's effectiveness is influenced by the scenes' arrangement, the relative distances between objects, and their impact on the mask image. To overcome the abovementioned limitations, an edge objective is introduced to guide the method towards the object's location. Unfortunately, this objective proved unstable and incompatible with the other two objectives—silhouette and collision—that formed the foundation of our loss computations.

Furthermore, the ablation studies conducted on scene 3 in preceding sections reveal that the method struggles when objects are positioned on top of each other rather than on a table. This limitation arises from the definition of signed distance, which forces each object to rest on the table, preventing them from floating in the air.

Moreover, the camera view angle can be misleading, resulting in imprecise estimations of an object's pose, particularly when the object's shape varies across different sides. For example, as depicted in Figure 5.5, the placement of canisters in the estimation mask is inaccurate due to the camera view angle covering the knobs of the canisters. This

inconsistency occurred despite the pixel-by-pixel similarity between the estimation and reference mask images.

CHAPTER 7

Conclusion and Future Work

The identification and pose estimation of transparent materials create significant challenges for robots and autonomous systems due to their reflective and refractive properties. Traditional depth-based methods are insufficient for transparent objects as their depth information is corrupted by transparency. Similarly, texture-based methods face challenges due to reflections and the necessity for precise background modelling.

To overcome the challenges posed by transparent objects, this thesis introduces a novel pipeline that uses differentiable rendering to refine the position and orientation of transparent objects from RGB images. The pipeline utilizes 3D models and a differentiable renderer to generate a 3D representation of the scene, optimizing directly within the image space using the parameters of the 3D scene. By adding physical constraints, the pipeline guides the estimations towards more realistic spatial arrangements, avoiding implausible object collisions. An advantage of this approach is its lack of dependency on large-scale training datasets, making it particularly attractive for robotics applications, such as scene-understanding tasks for tidying up.

The pipeline is evaluated on the Tracebot dataset, containing three scenes with transparent canisters, recorded and annotated for this thesis. Results indicate that the method's convergence depends on the overlap between estimated and ground truth mask images. For the first two scenes, with an ADI threshold below 5mm, the pipeline achieves over 95% estimation convergence above that threshold. When evaluated on selected scenes from the T-Less dataset, the pipeline achieves an average recall of 0.17 using the pose estimation results of Megapose [LMM+22] and 0.22 using the pose estimation of Pix2pose [PPV19].

However, the method's success depends on scene arrangement, object visibility, and camera perspective. Although it bridges the gap towards RGB-D general refinement methods, introducing an edge objective-based loss could potentially enhance accuracy to levels comparable to state-of-the-art methods. Despite some limitations, such as potential inaccuracies in pose estimation due to occlusions or lack of distinguishable features, the

pipeline's flexibility and effectiveness make it a valuable tool in domains where accurate positioning of transparent objects is essential.

While the current pipeline performs well on more straightforward scenes like the Tracebot dataset, future improvements could extend its capability to handle more complex scenes. The flexibility of the loss definition-based pipeline allows for easy adaptation and improvement by defining new loss calculations.

One possible direction is exploring other rendering techniques, like detecting interior edges, as an additional cue to enhance the robustness of pose alignment, particularly in cases of heavy occlusion. This approach could contribute to more robust pose alignment in general. Furthermore, the signed distance design approach, which enforces objects to rest on the table, could be enhanced to handle objects resting on each other or the table. This improvement could better manage complex scenes, such as scene three from Tracebot, where two canisters are mounted on a drain tray.

Another avenue for exploration is rendering RGB images in addition to mask images. This could provide valuable additional information considering the environment's impact on object visibility and could be incorporated into the pipeline as a new loss definition. However, this approach requires object meshes to contain information about object transparency to render RGB images.

In summary, while the current pipeline shows promise in refining the pose of transparent objects, these future directions have the potential to enhance its performance and applicability significantly. Exploring these avenues could bring us closer to solving the complex problem of transparent object pose estimation in diverse real-world scenarios.

List of Figures

List of Tables

List of Algorithms

Glossary

depth image An image containing the distance of each pixel in the observed scene from the camera view point. 5

manipulation Robotic manipulation refers to the ways robots interact with the objects around them. 1

mask image A mask image refers to a two dimensional image in black and white, showing the pixels containing the object of interest in white. 17–19, 22, 23, 26, 37, 39, 44, 50, 51, 61

renderer A renderer in computer graphics refers to a software component or engine that takes 3D models and scene descriptions as inputs and generates 2D images from the desired viewpoint. vii, 9, 15, 17, 18, 22

scene 1 First scene within the Tracebot dataset containing only one canister. 17, 18, 27, 31, 34, 35, 37–41, 43, 45–50, 53, 60, 67, 68

scene 2 Second scene within the Tracebot dataset containing two canisters. 27, 31, 34, 35, 37–47, 49, 50, 55, 60, 67, 68

scene 3 Third scene within the Tracebot dataset containing two canisters and a draintray. 31, 35–37, 45, 47, 53, 62, 67, 68

Acronyms

L_{MSD} Masked Signed Distance Loss. 26, 45–53, 56, 58, 61, 68

L_{SDU} Symmetric Difference over Union. 23, 24, 33, 35–38, 40–45, 47–49, 59–61, 67, 68

L_{SPSMD} Sum of Positive Squared Mask Differences Loss. 23, 33, 35

L_{SSMD} Sum of Squared Mask Difference Loss. 23, 24, 33, 35

2D Two Dimensional. 6, 9, 10, 12–15, 17, 26

3D Three Dimensional. vii, 2, 3, 6, 9, 10, 13–15, 17, 65

6D Six Dimensional. 14

6DoF Six Degrees of Freedom. 3, 10, 13, 14, 17, 23

AAE Augmented Auto Encoder. 14

ADD Average Distance of Distinguishable Model Points. 11, 37–44, 48–55, 61, 67–69

ADI Average Distance of Indistinguishable Model Points. 11, 33–57, 59–61, 65, 67–69

BOP Benchmark for 6D Object Pose Estimation. 9, 10, 13, 33, 56

CNN Convolutional Neural Network. 14

Diff-DOPE Differentiable Deep Object Pose Estimation. 15

GDR-Net Geometry-Guided Direct Regression Network. 13

ICP Iterative Closest Point. 14

ICP Iterative Closest Point. 56, 58, 62

LiDAR Laser imaging, Detection, and Ranging. 2

MSPD Maximum Symmetry-Aware Projection Distance. 12

MSSD Maximum Symmetry-Aware Surface Distance. 12

NeMO Neural Mesh Models for 3D reasoning.. 15

PnP Perspective-n-Point. 13

PoseCNN Pose estimation using Convolutional Neural Network. 14

RANSAC RANdom SAmple Consensus. 13, 19, 24

RCM Reverse Cuthill-McKee. 19

RGB Red-Green-Blue. vii, 3, 4, 6, 9, 14, 15, 17, 19, 22, 29, 37, 56, 58, 62, 65–67

RGB-D Red-Green-Blue-Depth. 2, 10, 56, 62, 65

SIFT Scale-Invariant Feature Transform). 13

SSD Single Shot multiBox Detector. 14

T-Less Texture-Less objects. 9–11, 56, 62, 65, 67

VSD Visible Surface Discrepancy. 12

Bibliography

[ALKN19] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2447–2456, 2019.

[BEM23] Ramchander Bhaskara, Roshan T Eapen, and Manoranjan Majji. Differentiable rendering for pose estimation in proximity operations. In *AIAA SCITECH 2023 Forum*, page 2317, 2023.

[BKC22] Munkhtulga Byambaa, Gou Koutaki, and Lodoiravsal Choimaa. 6d pose estimation of transparent object from single rgb image for robotic manipulation. *IEEE Access*, 10:114897–114906, 2022.

[BR19] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4322–4331, 2019.

[Bun21] Robert M Bunch. Wave optics and light propagation. In *Optical Systems Design Detection Essentials*, 2053-2563, pages 3–1 to 3–69. IOP Publishing, 2021.

[Can81] H Cantzler. Random sample consensus (ransac). *Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh*, 3, 1981.

[CJS+23] Kai Chen, Stephen James, Congying Sui, Yun-Hui Liu, Pieter Abbeel, and Qi Dou. Stereopose: Category-level 6d transparent object pose estimation from stereo images via back-view nocs. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2855–2861, 2023.

[CLWX21] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation, 2021.

[CM69] Elizabeth Cuthill and James McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172, 1969.

[CWX+23] Kang Chen, Shaochen Wang, Beihao Xia, Dongxu Li, Zhen Kan, and Bin Li. Tode-trans: Transparent object depth estimation with transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4880–4886. IEEE, 2023.

[DMW+21] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. in 2021 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pages 12376–12385, 2021.

[DUB+17] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 2200–2208, 2017.

[GHJYAJ19] Chen Guo-Hua, Wang Jun-Yi, and Zhang Ai-Jun. Transparent object detection and location based on rgb-d camera. In *Journal of Physics: Conference Series*, volume 1183, page 012011. IOP Publishing, 2019.

[GLH+21] Ge Gao, Mikko Lauri, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. Cloudaae: Learning 6d object pose regression with on-line data synthesis on point clouds. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11081–11087, 2021.

[GNZN23] Ning Gao, Vien Anh Ngo, Hanna Ziesche, and Gerhard Neumann. Sa6d: Self-adaptive few-shot 6d pose estimator for novel and occluded objects. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1572–1595. PMLR, 06–09 Nov 2023.

[GWZ+20] Alexander Grabner, Yaming Wang, Peizhao Zhang, Peihong Guo, Tong Xiao, Peter Vajda, Peter M. Roth, and Vincent Lepetit. Geometric correspondence fields: Learned differentiable rendering for 3d pose refinement in the wild, 2020.

[HB19] Frederik Hagelskjær and Anders Glent Buch. Pointposenet: Accurate object detection and 6 dof pose estimation in point clouds. *CoRR*, abs/1912.09057, 2019.

[HH] Frederik Hagelskjær and Rasmus Laurvig Haugaard. Keymatchnet: Zero-shot pose estimation in 3d point clouds by generalized keypoint matching.

[HHO+17] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[HLI+13] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*, pages 548–562. Springer, 2013.

[HMO16] Tomáš Hodaň, Jiří Matas, and Štěpán Obdržálek. On evaluation of 6d object pose estimation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 606–619. Springer, 2016.

[HSD+20] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 577–594. Springer, 2020.

[HZPB20] Martin N Hebart, Charles Y Zheng, Francisco Pereira, and Chris I Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nature human behaviour*, 4(11):1173–1185, 2020.

[ILK+21] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021.

[KLR18] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018.

[KMT+17] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017.

[KTEM18] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[LADL18] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

[LB14] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 154–169. Springer, 2014.

[LCAS20] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020.

[LCL20] Tingtian Li, Yuk-Hee Chan, and Daniel PK Lun. Improved multiple-image-based reflection removal algorithm using deep neural networks. *IEEE Transactions on Image Processing*, 30:68–79, 2020.

[LEB13] Ilya Lysenkov, Victor Eruhimov, and Gary Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. *Robotics*, 273(273-280):2, 2013.

[LJAK20] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects, 2020.

[LLCL19] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019.

[LMM+22] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022.

[LR13] Ilya Lysenkov and Vincent Rabaud. Pose estimation of rigid transparent objects in transparent clutter. In *2013 IEEE International Conference on Robotics and Automation*, pages 162–169. IEEE, 2013.

[LWJ+18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[MKNT18] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6d pose refinement in rgb, 2018.

[MST+21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[NDVZJ19] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), December 2019.

[NGSL23] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. *arXiv preprint arXiv:2311.14155*, 2023.

[NPLBY18] Thu H Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *Advances in neural information processing systems*, 31, 2018.

[PBCC18] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. End-to-end 6-dof object pose estimation through differentiable rasterization. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[PPV19] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.

[QHZ+23] Wei Qin, Qing Hu, Zilong Zhuang, Haozhe Huang, Xiaodan Zhu, and Lin Han. Ippe-pcr: a novel 6d pose estimation method based on point cloud repair for texture-less and occluded industrial parts. *Journal of Intelligent Manufacturing*, 34(6):2797–2807, 2023.

[RRN+20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[SLR+80] JA Smith, Tzeu Lie Lin, KJ Ranson, et al. The lambertian assumption and landsat data. *Photogrammetric Engineering and Remote Sensing*, 46(9):1183–1189, 1980.

[SMD+18] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)*, pages 699–715, 2018.

[SMP+19] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Cleargrasp: 3d shape estimation of transparent objects for manipulation, 2019.

[SMP+20] Shreeyak Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, and Shuran Song. Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International*

Conference on Robotics and Automation (ICRA), pages 3634–3642. IEEE, 2020.

[SNS+23] Markus Suchi, Bernhard Neuberger, Amanzhol Salykov, Jean-Baptiste Weibel, Timothy Patten, and Markus Vincze. 3d-dat: 3d-dataset annotation toolkit for robotic vision. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9162–9168. IEEE, 2023.

[SSF+22] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. *arXiv preprint arXiv:2203.09418*, 2022.

[SSH20] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

[TCM15] Shubham Tulsiani, João Carreira, and Jitendra Malik. Pose induction for novel object categories, 2015.

[TSF18] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 292–301, 2018.

[TWB+23] Jonathan Tremblay, Bowen Wen, Valts Blukis, Balakumar Sundaralingam, Stephen Tyree, and Stan Birchfield. Diff-dope: Differentiable deep object pose estimation. *arXiv preprint arXiv:2310.00463*, 2023.

[WKY21] Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. *arXiv preprint arXiv:2101.12378*, 2021.

[WMTJ21] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation, 2021.

[WSH+19] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation, 2019.

[XCY+20] Chi Xu, Jiale Chen, Mengyang Yao, Jun Zhou, Lijun Zhang, and Yi Liu. 6dof pose estimation of transparent object from a single rgb-d image. *Sensors*, 20(23), 2020.

[XQL+19] Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from shape: Deep pose estimation for arbitrary 3d objects, 2019.

[XSNF17] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[YCFB+21] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

[YHZ+18] Shunyu Yao, Tzu Ming Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, Bill Freeman, and Josh Tenenbaum. 3d-aware scene manipulation via inverse graphics. *Advances in neural information processing systems*, 31, 2018.

[Zha21] Zhengyou Zhang. Iterative closest point (icp). In *Computer vision: a reference guide*, pages 718–720. Springer, 2021.

[ZLS14] Xenophon Zabulis, Manolis IA Lourakis, and Stefanos S Stefanou. 3d pose refinement using rendering and texture-based matching. In *Computer Vision and Graphics: International Conference, ICCVG 2014, Warsaw, Poland, September 15-17, 2014. Proceedings*, pages 672–679. Springer, 2014.

[ZOC+22] Huijie Zhang, Anthony Opipari, Xiaotong Chen, Jiyue Zhu, Zeren Yu, and Odest Chadwicke Jenkins. Transnet: Category-level transparent object pose estimation. In *European Conference on Computer Vision*, pages 148–164. Springer, 2022.

[ZSI19] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019.