TECHNISCHE
UNIVERSITÄT
WIEN

DIPLOMARBEIT

Robust Pose Estimation of 3D Objects with Symmetries

ausgeführt am

Institut für
diskrete Mathematik und Geometrie
TU Wien

unter der Anleitung von

Associate Prof. Dr. Christian Müller

durch

Philipp Schiller

[redacted]

Wien, am 14.05.2024

Kurzfassung

Die genaue Ermittlung der Position und Orientierung eines Objektes im Raum, besser bekannt als die Berechnung von Posen, ist eine fundamentale Problemstellung in vielen Bereichen wie Computer Vision, Robotik oder industrielle Automatisierung. Während sich viele der kürzlich entwickelten Methoden als sehr effektiv herausgestellt haben, werden symmetrische Objekte - trotz derer zahlreichen Präsenz im täglichen Leben - von den Autoren vielfach ignoriert. Aufgrund der Ambiguität der repräsentierenden Orientierungen stellen diese jedoch eine besondere Herausforderung dar und können bei vielen der aktuellen Methoden zu teils schwerwiegenden Problematiken führen. Um diese Lücke zu beheben, präsentieren wir einen Ansatz zur robusten Berechnung von Posen, der alle Arten von Objekten berücksichtigt, also auch symmetrische.

Typischerweise werden Posen in der Praxis mit rigiden Transformationen assoziiert. Diese Herangehensweise ist allerdings suboptimal bei symmetrischen Objekten. Wir wollen dieses Problem von Grund auf lösen, indem wir den Begriff der Pose neu definieren. Danach führen wir eine Metrik auf diesem Raum der Posen ein, welche die Symmetrie des Objektes berücksichtigt. Mit dieser Metrik weiten wir eine bestehende Methode zur Bestimmung von Posen auf alle Objekttypen aus.

Dazu demonstrieren wir zuerst die mangelhafte Funktionsweise des klassischen Zugangs mittels Experimenten mit synthetischen Daten. Danach zeigen wir die Effektivität unseres Algorithmus mit realen Daten. Zusätzlich vergleichen wir unseren Ansatz mit bestehenden Methoden und zeigen verbesserte Genauigkeit und Robustheit.

Abstract

The precise determination of an object's position and orientation in space, known as pose estimation, is fundamental in fields such as computer vision, robotics, and industrial automation. Despite the effectiveness of recent methods, authors frequently overlook the presence of symmetrical objects, which are ubiquitous in daily life. Due to the ambiguity of the representing orientation, such objects impose a special challenge when estimating their pose. To address this issue, we present a robust pose estimation approach applicable to all types of objects.

Typically, poses are associated with rigid transformations. However, this approach fails when dealing with symmetrical objects. We aim to tackle this problem at the core, by redefining the notion of pose. Subsequently, we introduce a metric on this space of poses, which accounts for the object's symmetries. Using this distance, we enhance a conventional method of robust pose estimation to accommodate symmetrical objects.

We demonstrate the insufficiency of a classical approach through experiments conducted with synthetic data. Furthermore, we validate the effectiveness of our proposed method through an experiment utilizing real-world data. This involves a comparative analysis with other state-of-the-art algorithms, showing improved robustness and accuracy.

Acknowledgment

I want to express my gratitude to the Austrian Institute of Technology (AIT), with whose collaboration this work was developed. It was part of the project AWARD - All Weather-AutonomousReal logistics operations and Demonstrations, which has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No 101006817. I am especially thankful to Katharina Ölsböck, my supervisor at the AIT, who guided me very carefully throughout this thesis. She gave me excellent mathematical support and many insights into the world of research, which were very valuable. Furthermore, I would also like to thank my supervisor, Christian Müller, for assisting me with all types of geometrical questions and for igniting my passion for geometry.

I also wish to thank Markus Murschitz, who gave me many new insights into programming, Michael Schwingshackl, who was a big help in providing the data, and Florian Wimmer, who was a mathematical support. Moreover, my deepest gratitude is to my family, who always encouraged me throughout this journey, and to everyone else that was part of my academic path.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am Datum

Name des Autors

Contents

1 Introduction

Estimating the *pose* of a 3D rigid object is a fundamental task appearing in numerous fields, including computer vision, robotics, augmented reality, and industrial automation. The pose of an object is widely regarded as its static state in space and is usually described in terms of position and orientation with respect to a fixed coordinate frame. Pose estimation typically refers to accurately detecting a specified object within a complex scene and correctly determining its pose. These scenes often originate from laser scans or depth images and contain significant clutter and noise, thus such processes are required to be robust. While traditional pose estimation techniques perform well for most objects, they tend to struggle when dealing with symmetrical objects because of the unique challenges posed by their special shape.

To grasp the problem better, it is essential to understand the notion of a pose. Even though a strict definition does not exist, poses are commonly agreed to be identified with rigid transformations. Those transformations leave the shape of the object unchanged, and are special forms of isometries of $\mathbb{R}^3$, known as the Special Euclidean group $SE(3)$. This approach works well for a non-symmetrical object, but entails some challenges if the object admits symmetries. In that case, multiple rigid transformations describe the same static state of the object. A sphere for example can be rotated arbitrarily but will never change its shape. In other words, the assignment of a rigid transformation to the shape of an object is not one-on-one anymore.

While this ambiguity does not present a challenge when *describing* poses, it does when a notion of *distance* is required. The space $SE(3)$ has been widely studied and is equipped with several common metrics [BK02], [Par95], which are commonly used as a form of distance on the pose space. However, for symmetrical objects this is of limited usage, as there is no unique element of $SE(3)$ to describe the object's state. This could lead to cases where two poses admit the same physical appearance but have a distance greater than 0 since the underlying rigid transformations are different. Therefore, a metric would be ill-defined.

Numerous popular robust pose estimation methods rely on such a concept of distance between poses. Local approaches based on local invariant features, such as [DUNI10], [CJ97], [TTKK14], and [RBB09], have proven to be highly efficient and have garnered significant interest in recent years. They consist of generating a large set of poses which approximately describes the ground truth. Processing this set of *candidates* usually requires a notion of distance or at least of similarity to perform operations like neighborhood queries. Global approaches circumventing this problem, like [PGBP10], [SWK07] or [RVDH05], are known for being rather inefficient or specified for very particular objects and have not attracted much attention. In the current deep learning era, many machine learning attempts like [HLI$^+$13], [KMT$^+$17], [DCPR18], or [TSF18] were made. While these methods also enjoy great success, they usually do not discuss the handling of symmetrical objects. According to [PRIL19] or [RL18], symmetries can be challenging due to the ambiguity of the poses and their respective representations, possibly leading to convergence problems of the algorithm. Here a notion of distance is of utterly importance as a form of loss function.

As we have stated above, many authors overlook the presence of symmetrical objects, although they occur frequently in real-world scenarios. Their methods often employ metrics that are ill-suited for symmetrical objects, potentially resulting in subpar outcomes. However, some attempts were made to circumvent this challenge. The approach of [PRIL19] is to normalize the pose rotation and [DI15] extend their method developed in [DUNI10] to geometric primitives, such as cylinders or spheres. Some authors like [HKLM22] or [dFMB15] focus on rotationally symmetric objects. That is, most approaches either specify one type of symmetry or only work for geometric primitives for which they are designed.

Our goal is to develop a method for robust pose estimation that is applicable to all types of objects. We believe that defining a symmetry-aware metric within the pose space is crucial to achieving this goal. Our work resembles the methods of [BDLC18] and [BDLC17] which precisely address this challenge. They adapt the widely adopted method for robust pose estimation proposed by [DUNI10] by employing their symmetry-aware distance to accommodate symmetrical objects. Our contributions involve employing a different clustering approach combined with using the voting scores as weights.

Defining a metric on the pose space requires a whole new definition of the pose itself, which will be thematized in Chapter 2. There we will explore the symmetries of 3D objects as well. Subsequently, in Chapter 3, we will introduce a metric on the pose space that comprehensively accounts for all symmetries. We will not only offer a theoretical framework but also present an easy and efficient method for practical computation. Moving forward, Chapter 4 will be devoted to the theory of robust pose estimation, where we aim to use our proposed distance to extend a given method to be valid for all types of symmetry. Finally, in the concluding chapter, we will validate our developed theory through practical testing, demonstrating the significant impact of considering symmetries on pose estimation in terms of robustness and accuracy.

2 The pose of any rigid object

Generally, throughout this thesis, we will speak about *rigid objects*, which we treat as any non-empty, bounded subset of $\mathbb{R}^n$, where n equals 3 most of the time. The term *rigid* should emphasize that we do not want the shape or geometry of the object to be changed. Mathematically speaking, we want the distance between any two points of the object to be preserved by any transformation, avoiding distortions.
The *pose* of an object refers to the position in space where the object is located. This chapter is devoted to finding a formal mathematical description of the pose of any rigid object, including those with symmetries.

2.1 The pose space

Especially in computer vision and robotics, the notion of pose is widely used. Even though it is somehow intuitively clear what is meant, it is not straightforward to find a proper description, and we are not aware of a general definition in the literature. Therefore, we will refer to the definition given by [BDLC18].

Definition 2.1.1 (Pose)**.** The *pose* of a rigid object is a distinguishable static state of this object. The set of all possible poses is called the *pose space* of the object and is denoted with $\mathcal{C}$.

Remark. In robotics literature, the pose space is often referred to as the *configuration space*, which is why we use the letter $\mathcal{C}$.

We will see that the pose space of an object is heavily dependent on the object itself. In the case of the object being a sphere, the pose space can be stated immediately by straightforward arguments.

Example 2.1.2. Let O be a sphere in $\mathbb{R}^n$, i.e. $O = \{x \in \mathbb{R}^n \mid \|x\| = r\}$ for any $r > 0$. Then the pose of O is uniquely determined by the center of O. Any point $c \in \mathbb{R}^n$ defines a unique distinguishable static state of O by being its center, therefore $\mathcal{C} \cong \mathbb{R}^n$.

In general, it will not be that easy to describe the pose space, since not only the *position* of an object plays a role, but also the *orientation*. In the case of a sphere, the orientation is omitted by its special shape. There are various classes of shapes that lead to different descriptions of the pose space.

Regardless of the shape of an object, the main tool to describe the pose space is the set of transformations that leave the object's shape fixed.

2.2 Rigid transformations

With the term *transformation* we refer to a map from $\mathbb{R}^n$ to itself. If we have a transformation T and an object $O \subset \mathbb{R}^n$, then the image $T(O) \coloneqq \{T(x) \mid x \in O\} \subset \mathbb{R}^n$ describes another object. For a general transformation, this image could be anything, but as we have stated before, we want the shape of our object to be preserved. Formally speaking, for any two points $x, y \in O$ the distance of $T(x)$ and $T(y)$ should be the same as of x and y. This leads to the following important definition, which is for example due to [GS07] where it is stated only for two dimensions, but the same definition can be easily extended to $\mathbb{R}^n$.

Definition 2.2.1 (Rigid transformation)**.** A rigid transformation T is a surjective map from $\mathbb{R}^n$ to $\mathbb{R}^n$ which is isometric, i.e.

$$\forall x, y \in \mathbb{R}^n : \; \|T(x) - T(y)\| = \|x - y\|.$$

First, we want to study the structure of rigid transformations.

Lemma 2.2.2. *Every rigid transformation is injective and therefore bijective.*

Proof. Let $x, y \in \mathbb{R}^n$ and $T(x) = T(y)$. Then we have

$$0 = \|T(x) - T(y)\| = \|x - y\|$$

and therefore $x = y$. □

Theorem 2.2.3. *The set of rigid transformations forms a group with the operation being the composition of maps.*

Proof. One can easily check that the identity $E : x \mapsto x$ serves as the neutral element. The composition of bijective maps is again bijective and if S, T are rigid transformations, then for any $x, y \in \mathbb{R}^n$, we have

$$\|S(T(x)) - S(T(y))\| = \|T(x) - T(y)\| = \|x - y\|,$$

which shows that $S \circ T$ is again an isometry. Since any rigid transformation T is bijective it admits an inverse T^{-1}, satisfying $T \circ T^{-1} = E = T^{-1} \circ T$. □

A simple, yet very important example of a rigid transformation can be given by translating a point by a fixed vector.

Definition 2.2.4. The map $T_a : \mathbb{R}^n \to \mathbb{R}^n, x \mapsto x + a$ is called a translation by the vector a.

Remark. One can easily verify for $x, y \in \mathbb{R}^n$, that

$$\|T_a(x) - T_a(y)\| = \|x + a - (y + a)\| = \|x - y\|$$

which shows that T_a is an isometry. The preimage of x is $x - a$, therefore T_a is surjective and a rigid transformation.

To give other examples of rigid transformations, we need a quick review of notation. If a transformation T is linear, then it can be represented by a matrix $A \in \mathbb{R}^{n \times n}$ via $T(x) = Ax$. Translations for example are not linear, therefore they cannot be represented by a matrix (at least not in the same dimension).

Definition 2.2.5. A matrix $A \in \mathbb{R}^{n \times n}$ is said to be orthogonal, if $A^T A = E = AA^T$.

Since for any matrix A it holds that $\det A = \det A^T$, the multiplication theorem of determinants shows $\det A = \pm 1$ in the case of A being orthogonal. It can be easily seen, that the product of orthogonal matrices is again orthogonal, therefore they form a group, the *Orthogonal group* $O(n)$.

Orthogonal matrices form another important class of rigid transformations. One quick way to see this is to recall that orthogonal matrices have norm 1, which immediately implies isometry. A different argumentation via the study of basis can be found in [Gal11]. The following well-known classification is due to [GS07].

Definition 2.2.6. An orthogonal matrix $A \in \mathbb{R}^{n \times n}$ is called a rotation if $\det(A) = 1$ and a reflection if $\det(A) = -1$.

For two rotation matrices R and S, we have that $\det(RS) = \det(R) = \det(S) = 1$, implying that the rotation matrices form a subgroup of $O(n)$. This is often referred to as the *Special Orthogonal group* $SO(n)$.

Together with translations, orthogonal matrices allow us to describe the set of rigid transformations more explicitly.

Theorem 2.2.7. *Every rigid transformation is the composition of a translation and an orthogonal transformation.*

Proof. Let T be a rigid transformation. We have $T(0) = a$ for some $a \in \mathbb{R}^n$. The rigid transformation $U := T_{-a} \circ T$ satisfies $U(0) = 0$. We need to show that U is orthogonal. First, note that U respects norms, since

$$\|U(x)\| = \|U(x) - U(0)\| = \|x - 0\| = \|x\|. \tag{2.1}$$

It also respects the scalar product. This can be seen in the following way. Let $x, y \in \mathbb{R}^n$. Since U is rigid, we have

$$\|x - y\|^2 = \|U(x) - U(y)\|^2.$$

Now we compute

$$\|x - y\|^2 = \langle x - y, x - y\rangle = \|x\|^2 - 2\langle x, y\rangle + \|y\|^2$$
$$\|U(x) - U(y)\|^2 = \langle U(x) - U(y), U(x) - U(y)\rangle = \|U(x)\|^2 - 2\langle U(x), U(y)\rangle + \|U(y)\|^2$$

and with the help of (2.1) we obtain

$$\langle x, y\rangle = \langle U(x), U(y)\rangle. \tag{2.2}$$

In the last step, we want to show the linearity of U. Let $\lambda \in \mathbb{R}$, then we obtain with (2.2)

$$\begin{aligned}\|U(\lambda x) - \lambda U(x)\|^2 &= \|U(\lambda x)\|^2 - 2\lambda\langle U(\lambda x), U(x)\rangle + \lambda^2\|U(x)\|^2 \\ &= \|U(\lambda x)\|^2 - 2\lambda\langle \lambda x, x\rangle + \lambda^2\|x\|^2 \\ &= \|U(\lambda x)\|^2 - \lambda^2\|x\|^2 \\ &= \|U(\lambda x)\|^2 - \|\lambda x\|^2 = 0.\end{aligned}$$

In the same manner one can show that

$$\|U(x+y) - U(x) - U(y)\|^2 = 0.$$

In total, we obtain $U(\lambda x + y) = \lambda U(x) + U(y)$ and therefore the linearity of U. The representing matrix of U in an orthonormal basis is orthogonal. Subsequently, T can be written as $T = T_a \circ U$. □

This theorem shows that any rigid transformation can be written as a tuple (R, T_a), where R is a reflection or a rotation. The set of all rigid transformations in dimension n is also called the *Euclidean group* of order n, written as $E(n)$. If we restrict R to being a rotation, we obtain the *Special Euclidean group* of order n, $SE(n)$. This set is also referred to as the set of *proper* rigid transformations.
In robotics, the Special Euclidean group is of wide interest since they preserve the handedness of an object. In this work, we will only focus on $SE(n)$, often omitting the term *proper* for rigid transformations. In the sense of Theorem 2.2.7 we will write $T = (R, t)$ for a proper rigid transformation T, where $R \in \mathbb{R}^{n\times n}$ is a rotation and $t \in \mathbb{R}^n$ is a translation vector.

2.3 Linking the pose space with $SE(n)$

The Special Euclidean group is highly related to the pose space, as proper rigid transformations will be the main tool to describe the state of an object. Connecting both terms is due to [BDLC18]. Let us consider an object O and an arbitrary pose $P_0 \in \mathcal{C}$. Applying any rigid transformation to the object at its reference pose will define a static state of the object. Therefore, every rigid transformation describes a pose. Conversely, suppose $P \in \mathcal{C}$ is an arbitrary pose. Then, P can be reached via a rigid transformation $T = (R, t)$ applied to the reference pose P_0. To be more precise, this works the following way. If $x \in \mathbb{R}^n$ is a point linked to the object at P_0, then the rigid transformation

$$T(x) = Rx + t$$

sends x to a point $T(x)$ assigned to the object at pose P. Therefore, we can use $SE(n)$ to fully describe the pose space.

However, this approach admits a certain problem in some cases. It would be desirable to assign to each pose exactly one rigid transformation to have a proper representation. But in general, two different rigid transformations do not automatically lead to two different distinguishable static states. For example, consider the simple case of the object being a sphere. Any two transformations $T_1 = (R_1, t_1), T_2 = (R_2, t_2)$ will lead to the same pose, as long as $t_1 = t_2$. The problem lies in the shape of the sphere: it admits symmetries.

2.4 The symmetry group

Formally speaking, symmetries are invariants of rigid transformations. The following definition can be found for example in [Ced04].

Definition 2.4.1. A symmetry of a given object O is a proper rigid transformation T, such that $T(O) = O$. The set of all symmetries of O is denoted as G_O and is called the proper symmetry group of O. The word proper shall emphasize the fact that we are excluding reflection symmetries. For simplicity, we will often refer to it as the symmetry group.

Remark. Since the identity is trivially a symmetry, the proper symmetry group is never empty. Indeed, it also forms a group with composition. If $S, T \in G_O$, then

$$S \circ T(O) = S(O) = O$$

and the inverse of a symmetry T is given by T^{-1}. Therefore, the proper symmetry group is a subgroup of $SE(n)$.

Example 2.4.2. The symmetry group of the sphere with a fixed radius is given by $SO(n)$.

Example 2.4.3. If $O = \mathbb{R}^n$, then $G_O = SE(n)$, that is why $SE(n)$ can be seen as the symmetry group of the space itself.

If the object is bounded, we can make an additional assumption about the proper symmetry group.

Theorem 2.4.4. *If the given object O is bounded, G_O is a subgroup of $SO(n)$.*

Proof. For any rigid transformation $T = (R, t)$ the condition $T(O) = O$ can only be fulfilled if $t = 0$, otherwise this would contradict the boundedness of O. Therefore T must be a rotation. □

Remark. This does not have to be the case if O is unbounded. Consider an infinite line, with direction vector $x \in \mathbb{R}^3$, $x \neq 0$. Then the symmetry group of the line is the set of all translations along that line and all rotations around the line.

2.5 Poses as equivalence classes of $SE(n)$

Now we want to formalize the connection of the pose as we defined it and the symmetry group, which is due to [BDLC18]. Heuristically speaking, the pose space is the special Euclidean group factorized by the symmetry group. It turns out to be fruitful to see the pose space as a set of equivalence classes of $SE(n)$. Rigid transformations are considered to be equivalent if they transform the object into the same pose.

Remark. There is an algebraic construction to factorize a group G by a subgroup H, notated by G/H which is defined by $G/H := \{gH \mid g \in G\}$. However, it turns out that such a construction is only possible if H is a *normal* subgroup, meaning that for every $g \in G$ the condition $gHg^{-1} = H$ is fulfilled. In general, G_O does not need to be normal. For example, $SO(3)$ does not contain any normal subgroup which is not trivial. Such groups are called simple. A proof of this can be found in [Sti08]. This is the reason we have to rely on equivalence classes rather than using this powerful algebraic tool.

Lemma 2.5.1. *Let O be an object in $\mathbb{R}^n$, then*

$$T_1 \sim T_2 :\Leftrightarrow T_1(O) = T_2(O) \tag{2.3}$$

defines an equivalence relation on $SE(n)$. The equivalence classes are precisely given by

$$[T]_\sim = \{T \circ S \mid S \in G_O\}. \tag{2.4}$$

Proof. The fact that $\sim$ is an equivalence relation follows directly from the properties of the equivalence relation " $=$ " in $\mathbb{R}^n$. The statement of the equivalence classes follows since for a given $S \in G_O$ it holds by definition that $S(O) = O$. □

This fact allows us to establish a deeper connection between the pose space and $SE(n)$.

Theorem 2.5.2. *The pose space can be identified with $SE(n)/\sim$, meaning that every pose corresponds to exactly one equivalence class of rigid transformations.*

Proof. Let O be an object with proper symmetry group G_O. At the beginning of Section 2.3 we have already seen that any distinguishable static state can be described with a rigid transformation T. By definition, every transformation $S \in [T]_\sim$ describes the same static state, therefore every pose corresponds to at least one equivalence class. If $S \nsim T$, then $S(O) \neq T(O)$ so they cannot describe the same pose, meaning that every pose can be identified with exactly one equivalence class of rigid transformations. □

Example 2.5.3. If G_O is trivial, i.e. the object admits no symmetries, then the pose space can be identified with the whole $SE(n)$.

2.6 Symmetry classes of bounded 3D objects

This chapter is devoted to obtaining a classification of possible symmetry groups. In general, the proper symmetry group can be very complicated, especially if the object is unbounded. Throughout this section, we will study a special case, namely the one of bounded objects in three dimensions. As we will see in Theorem 2.6.13, we can classify all possible finite symmetry groups. This result relies heavily on group theory, therefore we need a few preliminaries. Generally, we will use the multiplicative notion of a group, i.e. for a group $(G, \cdot)$ and $g, h \in G$ we will simply write gh for $g \cdot h$ and g^n for $\underbrace{g \cdot g \cdot \ldots \cdot g}_{n\text{-times}}$ omitting the notation for the operation. This chapter is due to [Arm97].

Definition 2.6.1. Let G be a group. A subgroup H of G is a subset of G which again forms a group with multiplication.

If we consider a group G and take one element x from it, we can consider the set $H := \{x^n \mid n \in \mathbb{Z}\}$. Then H forms a subgroup, since the neutral element is given by x^0 and it is closed under multiplication, because $x^m x^n = x^{m+n}$ for $m, n \in \mathbb{Z}$. This leads to the following definition.

Definition 2.6.2. Let G be a group, $x \in G$. The subgroup $\{x^n \mid n \in \mathbb{Z}\}$ is called generated by the element x and denoted by $\langle x \rangle$. If $\langle x \rangle = G$ then G is called *cyclic*. The cyclic group of order n (see Definition 2.6.4) is denoted by C_n.

Example 2.6.3. Let $g \in SO(3)$ be a rotation of 10 degrees around a given axis a. The group generated by g are all the rotations around a where the angle is a multiple of 10. This group contains exactly 36 elements and is cyclic by definition.

Definition 2.6.4. Let G be a group. The number of its elements is called the order of G, which could be possibly infinite. The order of an element $g \in G$ is the order of its generated group $\langle g \rangle$.

Example 2.6.5 (Dihedral group)**.** The group D_n of symmetries of a regular polygon with n sides is often referred to as the dihedral group of order n. Let r be a rotation of the polygon of $2\pi/n$ around its center and s a rotation of π around one of the axis of symmetry that lies in the plane of the polygon. The rotation s can also be seen as a reflection in the symmetry axis. Then the elements of D_n are

$$e, r, r^2, ..., r^{n-1}, s, rs, r^2s, ..., r^{n-1}s.$$

Clearly we have $ord(r) = n$ and $ord(s) = 2$. One can check geometrically that $sr = r^{n-1}s$ and since $r^{n-1} = r^{-1}$ we have $sr = r^{-1}s$. It can be shown that r and s generate D_n, which leads to the definition

$$D_n := \langle r, s \mid ord(r) = n, ord(s) = 2, sr = r^{-1}s \rangle. \tag{2.5}$$

Definition 2.6.6. Let G, H be two groups and $\phi : G \to H$ a map. We call ϕ a group homomorphism if

$$\text{for all } g, h \in G : \phi(gh) = \phi(g)\phi(h). \tag{2.6}$$

Definition 2.6.7. Let G be a group, X a set and S_X the group of all permutations of X, i.e. $S_X = \{\pi : X \to X \mid \pi \text{ bijective}\}$. A group homomorphism $\phi : G \to S_X$ is called an action of the group G on the set X.

Remark. The requirement of ϕ being a homomorphism allows a slight abuse of notation. Instead of writing $\phi(g)(x)$ for the image of the point $x \in X$ under the permutation $\phi(g)$ one often simply writes $g(x)$. Since ϕ is a homomorphism we get

$$\text{for all } g, h \in G, x \in X : g(h(x)) = \phi(g)(\phi(h)(x)) = \phi(gh)(x) = gh(x),$$

which justifies the notation. We therefore say G is acting on X.

Definition 2.6.8. Let G be a group acting on X and $x \in X$. The set of all images of x is called the orbit of x and will be denoted with $G(x) = \{g(x) \mid g \in G\}$. The subgroup G_x of all elements of G that leave the element x fixed is called the stabilizer of x, i.e. $G_x = \{g \in G \mid g(x) = x\}$.

Definition 2.6.9. Let G be a group with a subgroup H. The (left) cosets of H in G are given by $gH := \{gh \mid h \in H\}$ for every $g \in G$. The set of all cosets is denoted by $G/H = \{gH \mid g \in G\}$. The cardinality of G/H is called the index of the subgroup H.

Remark. The fact that G_x indeed forms a subgroup of G can be seen directly. If two elements $g, h \in G$ leave $x \in X$ fixed, then

$$gh(x) = g(h(x)) = g(x) = x.$$

For any element $x \in X$, there is an important connection between the size of its orbit and the size of its stabilizer. We do not want to dive too much into detail here, but the results rely on basic group theory.

Theorem 2.6.10. *Let G be a group acting on X and $x \in X$. If G is finite, then*

$$|G(x)| \cdot |G_x| = |G|. \tag{2.7}$$

Proof. A proof of this is given in [**?**, Chapter 17]armstrong1997groups □

Theorem 2.6.11 (Counting theorem / Lemma of Burnside)**.** *Let G be a finite group operating on a set X. Denote with $X^g := \{x \in X \mid g(x) = x\}$ the set of all fixpoints concerning the element g. Then the number N of distinct orbits is given by*

$$N = \frac{1}{|G|} \sum_{g \in G} |X^g| = \frac{1}{|G|} \sum_{x \in X} |G_x|. \tag{2.8}$$

Proof. A detailed proof of this well-known result can be found in [Arm97, Chapter 18]. □

Before we tackle the more general 3D case, we want to take a look at symmetries in two dimensions.

Theorem 2.6.12. *Let G be a finite subgroup of $O(2)$. Then G is either cyclic or dihedral.*

Proof. First, suppose that G lies in $SO(2)$, meaning that all the elements of G are rotations of the plane. Denote with A_θ the matrix representing the anticlockwise rotation of θ about the origin, where $0 \leq \theta < 2\pi$. Choose $\phi > 0$ as small as possible such that $A_\phi \in G$. We want to show that A_ϕ generates G. Let $A_\alpha \in G$ be any rotation in G. Divide α by ϕ to get $\alpha = k\phi + \psi$, where $k \in \mathbb{Z}$ and $0 \leq \psi < \pi$. The equation

$$A_\alpha = A_{k\phi+\psi} = A_\phi^k A_\psi$$

implies that

$$A_\psi = A_\phi^{-k} A_\alpha$$

which shows that $A_\psi \in G$. But $\psi > 0$ would contradict the minimality of ϕ which means $\psi = 0$. Therefore every element of G is a power of A_ϕ and G is cyclic.

Suppose G is not fully contained in $SO(2)$ and set $H = G \cap SO(2)$. Then H is a subgroup of G with index 2. The statement above implies that H is cyclic, so we can choose a generator A. Since the index of H is 2, for an element $B \in G \setminus H$ it holds that $B^2 = I$ and $B \notin SO(2)$, i.e. B represents a reflection. Now there are two cases. If $A = I$, then G is the cyclic group $\langle B\rangle$ containing two elements. If the order of A is $n \geq 2$, then $ord(A) = n$, $ord(B) = 2$. The elements of G are given by

$$I, A, ..., A^{n-1}, B, AB, ..., A^{n-1}B,$$

and satisfy $BAB^{-1} = A^{-1}$, so A and B are the generators of the dihedral group of order n, as the correspondence $A \mapsto r$, $B \mapsto s$ determines an isomorphism to a dihedral group as in Example 2.6.5. □

We can now finally present this chapter's main result, which can be found in [Arm97] or [Wey15].

Theorem 2.6.13. *Let G be a finite subgroup of $SO(3)$. Then G is isomorphic to one of the following groups:*

- *the cyclic group C_n for $n \in \mathbb{N}$*
- *the dihedral group D_n for $n \in \mathbb{N}$*
- *the rotational group of the tetrahedron (containing 12 elements)*
- *the rotational group of the cube (containing 24 elements)*
- *the rotational group of the icosahedron (containing 60 elements)*

Proof. Each element $g \in G$ represents a rotation of $\mathbb{R}^3$ about an axis a_g that passes through the origin. The axis a_g intersects the unit-sphere $\mathbb{S}_2 = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}$ in exactly two points, which are called the poles of g. The poles are exactly those points of $\mathbb{S}_2$ which are left fixed by g. Let X be the set of all poles of all elements of $G \setminus \{e\}$, where e denotes the identity. We will first show, that G is an action on X.

Suppose $g \in G$. For any $x \in X$, there is according to the definition of X, an $h \in G$, such that x is a pole of h. Since h leaves x fixed, the calculation

$$(ghg^{-1})(g(x)) = g(h(x)) = g(x)$$

shows that $g(x)$ is a fixed point of the element ghg^{-1}, i.e. a pole. But this means nothing else than $g(x) \in X$, therefore G is indeed acting on X.

The core idea is to use the Lemma of Burnside, Theorem 2.6.11, with the group G acting on X. Let N be the number of distinct orbits and choose one pole $x_1, ..., x_N$ of each of these. Every $g \in G \setminus \{e\}$ has exactly two poles, which implies $|X^g| = 2$. Additionally, the identity e fixes every point, therefore we obtain

$$N = \frac{1}{|G|}\sum_{g\in G}|X^g| = \frac{1}{|G|}\big(2(|G|-1) + |X|\big).$$

By our choice of $x_1, ..., x_N$ we have $|X| = \sum_{i=1}^{N}|G(x_i)|$. With Theorem 2.6.10 and some rearrangements we obtain

$$2\left(1-\frac{1}{|G|}\right)=N-\frac{|X|}{|G|}=N-\sum_{i=1}^{N}\frac{|G(x_i)|}{|G|}=N-\sum_{i=1}^{N}\frac{1}{|G_{x_i}|}=\sum_{i=1}^{N}\left(1-\frac{1}{|G_{x_i}|}\right). \tag{2.9}$$

If G is trivial there is nothing to prove, so let us assume G contains at least two elements. Therefore we have

$$1\leq 2\left(1-\frac{1}{|G|}\right)<2.$$

According to Theorem 2.6.10, each stabilizer G_{x_i} has at least order 2 if G is not trivial, which implies

$$\frac{1}{2}\leq 1-\frac{1}{|G_{x_i}|}<1, \text{ for } i=1,...,N. \tag{2.10}$$

Since the whole sum of the rightmost expression of (2.9) is smaller than 2, but every summand is larger than 1/2, N is either 2 or 3.

If $N=2$ we obtain from (2.9) that $2=|G(x_1)|+|G(x_2)|$ and there can only be two poles. This means there are exactly two distinct poles that define one axis a. Every element of G must be a rotation around this axis. Therefore, G is isomorphic to a finite subgroup of not only $O(2)$ but even $SO(2)$. According to Theorem 2.6.12 G must be cyclic.

Things get more complex if $N=3$. For simplicity, let us rename x_1,x_2,x_3 with x,y,z. Since (2.9) leads to

$$2\left(1-\frac{1}{|G|}\right)=3-\left(\frac{1}{|G_x|}+\frac{1}{|G_y|}+\frac{1}{|G_z|}\right)$$

which can be simplified to

$$1+\frac{2}{|G|}=\frac{1}{|G_x|}+\frac{1}{|G_y|}+\frac{1}{|G_z|}. \tag{2.11}$$

The left-hand side of this equation implies that

$$\frac{1}{|G_x|}+\frac{1}{|G_y|}+\frac{1}{|G_z|}>1. \tag{2.12}$$

This condition leads to 4 possible options, up to permutation. W.l.o.g. we assume $|G_x|\leq|G_y|\leq|G_z|$.

a) $|G_x|=2$, $|G_y|=2$ and $|G_z|=n$ for $n\geq 2$

b) $|G_x|=2$, $|G_y|=3$ and $|G_z|=3$

c) $|G_x|=2$, $|G_y|=3$ and $|G_z|=4$

d) $|G_x|=2$, $|G_y|=3$ and $|G_z|=5$

Let us consider these cases in detail. It shall be noted, that (2.11) immediately gives us the order of the group G.

a) If $n = 2$, then G is a group of order 4 and every element except the identity has order 2. Then, G must be isomorphic to a group called Klein's group, which is isomorphic to the dihedral group D_2.

If $|G_x| = |G_y| = 2$ and $|G_z| = n \geq 3$ then G must be of order $2n$. The axis through z and $-z$ is fixed by every element of the stabilizer G_z. We conclude that G_z must be a cyclic group of order n. Suppose g is the minimal rotation that generates G_z. Because of the minimality of g, all the points $x, g(x), g^2(x), ..., g^{n-1}(x)$ must be distinct. Since $g \in SO(3)$, it is an isometry and we obtain

$$\|x - g(x)\| = \|g(x) - g^2(x)\| = \cdots = \|g^{n-1}(x) - x\|.$$

Therefore, the elements $x, g(x), \ldots, g^{n-1}(x)$ form a regular polygon P with n vertices lying in a plane through the origin and perpendicular to the axis going through z and $-z$. The group G consists exactly of those $2n$ rotations that send P to itself, hence G must be the rotational symmetry group of P, which is the dihedral group of order n.

b) The group G has exactly 12 elements in this case while the orbit of z consists of 4 elements. Let us choose one point u of this orbit with $0 < \|u - z\| < 2$ and one generator g of G_z. Since $g(z) = z$ and g is an isometry, we obtain

$$0 < \|z - u\| = \|z - g(u)\| = \|z - g^2(u)\| < 2,$$

and therefore $u, g(u)$ and $g^2(u)$ are all equidistant from z. These points must form an equilateral triangle. If we switch our attention to u, we see that the points $z, g(u)$ and $g^2(u)$ are all equidistant from u and form an equilateral triangle as well. We conclude that $z, u, g(u)$ and $g^2(u)$ form a regular tetrahedron and G must be the rotational symmetry group of the tetrahedron.

c) Here G has order 24 and therefore the orbit of z contains 6 elements. Choose again u of this orbit, where u is neither z nor $-z$. Let g be a generator of G_z. With similar arguments as in the case before, we obtain that $u, g(u), g^2(u)$ and $g^3(u)$ must be the corners of a square. The only point left in the orbit of z has to be $-z$. Again, switching the focus to u shows directly that the pole $-u$ has to be in $G(u) = G(z)$. Since $\|g(u) - u\| = \|g^3(u) - u\| < 2$ we get that $g^2(u) = -u$. All in all, we obtain a regular octahedron with vertices $z, -z, u, g(u), g^2(u)$ and $g^3(u)$ and G is its symmetry group.

d) In the last case, G contains 60 elements and $G(z)$ contains 12 points. Let g be a minimal rotation generating G_z. We can choose a point $u \in G(z)$ where $u, g(u), g^2(u), g^3(u)$ and $g^4(u)$ are distinct and equidistant from z and form therefore a regular pentagon. We can do the same with a point $v \in G(z)$ that is further away from z than u and repeated action of g on v forms a pentagon again. Therefore we can choose u and v satisfying

$$0 < \|z - u\| < \|z - v\| < 2.$$

The twelfth point which is left in the orbit of z can only be $-z$. Now, as $G(u) = G(z)$ and $-u$ has to be in there as well, it has a distance of 2 to u and therefore can not be on

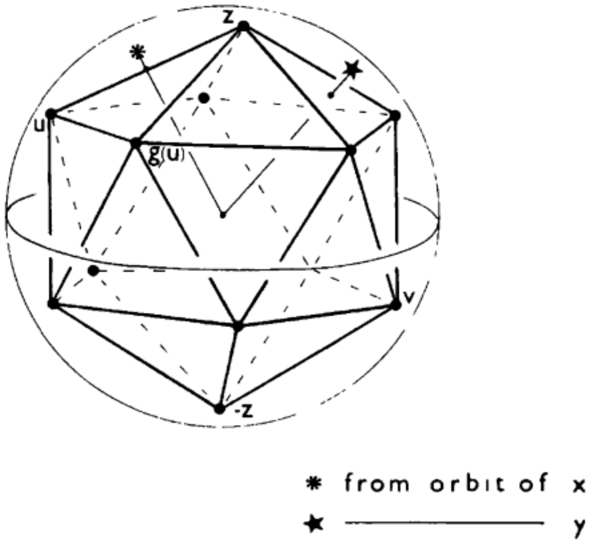


Figure 2.1: Symmetry group of the icosahedron, from [Arm97]

the pentagon containing u. It can neither be z nor $-z$, so it must be in the pentagon generated by v. W.l.o.g we assume $-u = v$, then $-g^r(u) = g^r(v)$. Now we change our attention to u. The five points which must be closest to u are $z, g(u), g^{-1}(u), g^2(v)$ and $g^3(v)$. Those must be equidistant as well and form another pentagon. All in all, it is easy to check that the twelve points form a regular icosahedron and G is the corresponding rotational symmetry group.

□

While we have successfully identified all finite subgroups of SO(3), the case is more complicated for infinite subgroups. According to [BDLC18] there is a classification of potential infinite proper symmetry groups that physically make sense. Namely, the spherical group that corresponds to the whole SO(3), and rotational invariance along a given axis by any angle. This is also called revolution symmetry, and can additionally include a reflection symmetry across a plane that is orthogonal to the axis of revolution, which is also called roto-reflection invariance. Other examples, like such that are not closed under the usual topology, will be omitted in this work. This leaves us essentially with three classes of symmetries that we will consider in this work:

- Spherical symmetry
- Revolution symmetry (with eventual roto-reflection invariance)
- Finite symmetry (containing the case of no given proper symmetry)

3 Defining a symmetry-aware distance on the pose space

The goal of this chapter is to define a metric on the pose space $\mathcal{C}$ of a given rigid object. A metric is a function $d : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ satisfying:

- $\forall P \in \mathcal{C} : \, d(P, P) = 0$
- $\forall P_1 \neq P_2 \in \mathcal{C} : \, d(P_1, P_2) > 0$
- $\forall P_1, P_2, P_3 \in \mathcal{C} : \, d(P_1, P_3) \leq d(P_1, P_2) + d(P_2, P_3)$
- $\forall P_1, P_2 \in \mathcal{C} : \, d(P_1, P_2) = d(P_2, P_1)$

As we will see, several classical approaches exist to define a metric on $SE(3)$, but they typically do not consider the object's shape or symmetry. This whole chapter is due to [BDLC18].

3.1 Prior works

In this section, we will shortly review an excerpt of the work done in this area and explain the relevant advantages and disadvantages.

3.1.1 Decomposition in translational and rotational part

A natural approach to defining a distance on $SE(3)$ is to decompose a rigid transformation into its translational and rotational parts. In other words, it is making use of the fact that $SE(3) \simeq \mathbb{R}^3 \times SO(3)$. By defining any metric d_{trans} on $\mathbb{R}^3$ and d_{rot} on $SO(3)$, one can fuse these to a metric on $SE(3)$ by considering some form of weighted generalized mean. For scaling factors $a, b > 0$, some exponent $p \in [1, \infty]$ and two rigid transformations $T_1 = (R_1, t_1)$ and $T_2 = (R_2, t_2)$ the function

$$d(T_1, T_2) = \sqrt[p]{a \cdot d_{\text{rot}}(R_1, R_2)^p + b \cdot d_{\text{trans}}(t_1, t_2)^p} \tag{3.1}$$

defines a metric. The most natural selection for d_{trans} is the Euclidean norm, while the selection for d_{rot} can be tricky. Various options are discussed and compared in [Huy09], but the most common would be the Riemannian distance over $SO(3)$. While this approach is usually easy to compute, it admits a clear disadvantage: symmetries in objects are not considered. While the translational part will not be affected by this, the rotational is. Also, it does not take the shape of the object into account but only measures the rotational angle. For example, the axis of rotation would make a big difference if an object is very distorted.

3.1.2 Geometric approach

A different approach would be to focus purely on the geometric aspect. Rather than measuring the distance between poses seen as transformations, it might be easier to calculate the distance between corresponding points. Let μ be a density function assigned to the object O and $V = \int \mu(x)\, dv$ be the volume of the object. Then for two transformations T_1, T_2 one can define for $p \geq 1$

$$d(T_1, T_2) := \frac{1}{V} \left(\int \mu(x) \|T_1(x) - T_2(x)\|^p \, dv \right)^{1/p} . \tag{3.2}$$

This expression has a strong physical meaning and is used in several applications. For example, [HLI$^+$13] uses the case $p = 1$ for the evaluation of pose estimation to name one. Nevertheless, while the interpretation of the metric is clear, the computation becomes difficult as there is no closed form of the integral. A possible approach used by [HLI$^+$13] is to consider only some vertices of the model to be able to calculate the integral explicitly. Moreover, this approach also disregards the existence of symmetries. Nevertheless, this problem can be fixed and we will adapt this idea to propose a symmetry-aware distance on the pose space.

3.2 Theoretical definition

This section proposes the formal definition of a symmetry-aware metric on the pose space $\mathcal{P}$ of a given object O. With G_O we will denote the proper symmetry group of O as in Definition 2.4.1 and let μ be a positive density distribution. Formally, we assume $\mu \circ G = \mu$ for all $G \in G_O$. First, we will start by defining a concept for distance on $SE(3)$.

Lemma 3.2.1. *Let O be an object and $T_1, T_2 \in SE(3)$. Then the surface integral*

$$d_{no\_symm}(T_1, T_2) := \sqrt{\frac{1}{S} \int_O \mu(x) \|T_1(x) - T_2(x)\|^2 \, ds} \tag{3.3}$$

defines a metric on $SE(3)$, where

$$S := \int_O \mu(x) ds.$$

Proof. The fact that $d(T_1, T_2) \geq 0$ is obvious, as well as $d(T_1, T_2) > 0$ for $T_1 \neq T_2$ and the symmetry. The triangle inequality is a direct application of Minkowski's inequality. □

The application of any other rigid transformation preserves this metric.

Lemma 3.2.2. *For $T_1, T_2, U \in SE(3)$ we have*

$$d_{no\_symm}(T_1, T_2) = d_{no\_symm}(U \circ T_1, U \circ T_2). \tag{3.4}$$

Proof. This follows straightforwardly from the fact that since U is a rigid transformation we have

$$\|U \circ T_1(x) - U \circ T_2(x)\|^2 - \|T_1(x) - T_2(x)\|^2$$

for any $x \in \mathbb{R}^3$. □

As the pose space is defined as equivalence classes of $SE(3)$ it is natural to define a metric on these.

Definition 3.2.3. Let O be an object, and G_O its proper symmetry group. Then the function $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}^+$ defined by

$$d(P_1, P_2) := \min_{G_1, G_2 \in G_O} d_{no\_symm}(T_1 \circ G_1, T_2 \circ G_2) \tag{3.5}$$

forms a metric on the pose space, where $T_1, T_2 \in SE(3)$ are representatives of the respective equivalence classes of P_1, P_2 and d_{no_symm} is the function defined in Lemma 3.2.1.

Proof. To justify our definition, we need to show the independence of the respective representatives of P_1 and P_2. Let $T_1, \tilde{T_1}$ be representatives of P_1 and $T_2, \tilde{T_2}$ of P_2. Then, since the minimum is taken over all elements of the proper symmetry group, we obtain

$$\begin{aligned} d([T_1], [T_2]) &= \min_{G_1, G_2 \in G_O} d_{no\_symm}(T_1 \circ G_1, T_2 \circ G_2) \\ &= \min_{G_1, G_2 \in G_O} d_{no\_symm}(\tilde{T_1} \circ G_1, \tilde{T_2} \circ G_2) = d([\tilde{T_1}], [\tilde{T_2}]). \end{aligned}$$

Moreover, the minimum is reached because the proper symmetry group is compact as a closed subgroup of the compact group $SO(3)$. Therefore d is well-defined. Lemma 3.2.1 shows immediately that d is a pseudo-metric.

If $d(P_1, P_2) = 0$, we have

$$\min_{G_1, G_2 \in G_O} d_{no\_symm}(T_1 \circ G_1, T_2 \circ G_2) = 0$$

and since $d_{\text{no\_symm}}$ is a metric there exist $G_1, G_2 \in G_O$ such that $T_1 \circ G_1 = T_2 \circ G_2$. In other words, $T_1 \approx T_2$ or $[T_1] = [T_2]$, so d is a metric. □

We have shown that the metric is independent of the underlying representative. Sometimes we omit the notation of equivalence classes, just writing $P = T$ for some representative T of the pose P.

It is possible to simplify the expression in some sense.

Lemma 3.2.4. *Let $P_1, P_2 \in \mathcal{P}$ with representatives $T_1, T_2 \in SE(3)$. Then*

$$d(P_1, P_2) = \min_{G \in G_O} d_{no\_symm}(T_1, T_2 \circ G) = \min_{G \in G_O} d_{no\_symm}(T_1 \circ G, T_2). \tag{3.6}$$

Proof. We will deduce the statement directly from (3.5). For two proper symmetries $G_1, G_2 \in G_O$ one can simply use the substitutions $x \leftarrow G_1(x)$ and $G \leftarrow G_2 \circ G_1^{-1}$ to obtain

$$\begin{aligned} d^2_{no\_symm}(T_1 \circ G_1, T_2 \circ G_2) &= \frac{1}{S} \int_O \mu(x) \|T_2 \circ G_2(x) - T_1 \circ G_1(x)\|^2 \, ds \\ &= \frac{1}{S} \int_{G_1(O)} \mu(G_1^{-1}(x)) \|T_2 \circ G(x) - T_1(x)\|^2 \, ds \\ &= \frac{1}{S} \int_O \mu(x) \|T_2 \circ G(x) - T_1(x)\|^2 \, ds \\ &= d^2_{no\_symm}(T_1, T_2 \circ G). \end{aligned}$$

Here we used that $\mu \circ G_1^{-1} = \mu$ and $G_1(O) = O$. The second equation can be proven in the same manner. □

3.3 Practical computation

The computation of d is in general rather complicated since it involves solving a surface integral. This is very unhandy for any practical use. Moreover, the symmetry group can be potentially infinitely large, therefore the minimization problem is not trivial either. This section will show how to compute the proposed distance efficiently.

Definition 3.3.1. Let P_i be poses for $i = 1, 2$. We call a finite subset $\mathcal{R}(P_i) \subset \mathbb{R}^N$ representatives of P_i, if

$$d(P_1, P_2) = \min_{p_1 \in \mathcal{R}(P_1), p_2 \in \mathcal{R}(P_2)} \|p_2 - p_1\|. \tag{3.7}$$

Definition 3.3.2. Let $x \in \mathbb{R}^N$, then the set

$$\text{proj}(x) := \arg\min_P \min_{p \in \mathcal{R}(P)} \|p - x\|^2 \tag{3.8}$$

is called the projection of x.

Representatives and projections act as the connection of the pose space with $\mathbb{R}^N$. One can see the benefit of having representatives of poses. The computation of the distance breaks down into computing a minimum of finite points in some $\mathbb{R}^N$, which can be done efficiently. The difficulty will be to find such representatives. We will see, that this is highly dependent on the symmetry class of the object and that this also determines N.

First, we need to take a look at $d_{\text{no\_symm}}$. Recall that for poses P_1, P_2 with representing rigid transformations T_1, T_2 and proper symmetries $G_1, G_2 \in G_O$

$$d_{\text{no\_symm}}(T_1 \circ G_1, T_2 \circ G_2) = \sqrt{\frac{1}{S} \int_O \mu(x) \|T_1 \circ G_1(x) - T_2 \circ G_2(x)\|^2 \, ds}.$$

If we write $T_i = (R_i, t_i)$ for $i = 1, 2$ we get

$$\|T_1 \circ G_1(x) - T_2 \circ G_2(x)\|^2 = \|R_1G_1x + t_1 - (R_2G_2x + t_2)\|^2$$
$$= \|R_1G_1x - R_2G_2x\|^2 + \|t_1 - t_2\|^2 + 2(t_1 - t_2)^T(R_1G_1 - R_2G_2)x.$$

Choosing the origin of the frame as the object's center of mass implies $\int_O \mu(x)x\, ds = 0$. Therefore the last term of the previous equation vanishes after integrating and we are left with

$$d^2_{\text{no\_symm}}(T_1 \circ G_1, T_2 \circ G_2) = \frac{1}{S}\int_O \mu(x)\|R_1G_1x - R_2G_2x\|^2 + \|t_1 - t_2\|^2\, ds,$$

and

$$d^2(P_1, P_2) = \|t_1 - t_2\|^2 + \underbrace{\min_{G_1,G_2 \in G_O} \frac{1}{S}\int_O \mu(x)\|R_1G_1x - R_2G_2x\|^2\, ds}_{:=d^2_{\text{rot}}(P_1,P_2)} \tag{3.9}$$

respectively.

So the metric can be decomposed into a translational and a rotational part. The translation is independent of the object, the interesting part lies in d_{rot}. In the following, we will investigate this more in detail.

3.3.1 The sphere

We start with the simplest example - the sphere. For a sphere with a fixed radius, the proper symmetry group is the whole $SO(3)$. To find representatives, let us examine the expression of the rotational distance. We have for rotations R_1, R_2 that

$$\min_{G_1,G_2 \in G_O} \frac{1}{S}\int_O \mu(x)\|R_1G_1x - R_2G_2x\|^2\, ds = 0,$$

since the minimum is reached for $G_1 = R_1^{-1}$ and $G_2 = R_2^{-1}$. If we define $\mathcal{R}(P) := t$, then (3.9) becomes

$$d(P_1, P_2) = \|t_1 - t_2\| = \|\mathcal{R}(P_1) - \mathcal{R}(P_2)\|$$

and we found a representative of the sphere. In this case, $N = 3$. The point of its center represents the pose of the sphere. Conversely, every point in $\mathbb{R}^3$ represents exactly one pose. Therefore, we have already answered the question of the projection of the sphere, being the point itself. Summing these observations up, we conclude the following theorem.

Theorem 3.3.3. *The pose space of the sphere is isomorphic to* $\mathbb{R}^3$.

Remark. All the previous arguments for the sphere are not restricted to the dimension of 3. In fact, it holds for any dimension n.

3.3.2 Obejcts without proper symmetries

Now we want to examine the other extreme - objects without any symmetries. This means that $G_O = \{I\}$. In that case every element $T = (R, t) \in SE(3)$ corresponds to exactly one pose. In order to find a representative of this pose we take a closer look at (3.9). As G_O is trivial, the rotational part of the metric can be reduced to

$$d^2_{\text{rot}}(P_1, P_2) = \frac{1}{S} \int_O \mu(x) \|R_1 x - R_2 x\|^2 \, ds. \tag{3.10}$$

In the following, we need a few facts from basic linear algebra.

Definition 3.3.4. The *trace* of a matrix $A \in \mathbb{R}^{n \times n}$ is defined as

$$\operatorname{Tr}(A) := \sum_{i=1}^{n} a_{ii}.$$

The Frobenius norm is defined via

$$\|A\|_F := \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2}.$$

Remark. The trace defines a linear operator on the set of matrices in $\mathbb{R}^{n \times n}$. It allows us to express the norm of a vector with its outer product. For $x \in \mathbb{R}^n$, the relation

$$\|x\|^2 = \sum_{i=1}^{n} x_i^2 = \operatorname{Tr}(xx^T) \tag{3.11}$$

can be immediately seen. Moreover, it is easy to see that the assignment

$$[A, B] := \operatorname{Tr}(A^T B)$$

defines a scalar product on $\mathbb{R}^{n \times n}$. In fact, this scalar product induces exactly the Frobenius norm, since

$$[C, C] = \operatorname{Tr}(C^T C) = \sum_{i,j=1}^{n} c_{ij}^2 = \|C\|_F^2. \tag{3.12}$$

Consequently, we can rewrite the inner part of the integral in (3.10).

Lemma 3.3.5. *Let $A, B \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^n$. We write $X := (xx^T)^{1/2} \in \mathbb{R}^{n \times n}$ for the square root matrix of the outer product of x.*

$$\|Ax - Bx\|^2 = \operatorname{Tr}\left((A - B)X^2(A - B)^T\right) = \|AX - BX\|_F^2. \tag{3.13}$$

Proof. The first equality follows immediately from (3.11) for the vector $Ax - Bx$. For the second equation, we use (3.12) for the matrix $(A - B)X$ and the fact that $X = X^T$. Then we have

$$\operatorname{Tr}\left((A - B)X^2(A - B)^T\right) = \operatorname{Tr}\left((A - B)XX^T(A - B)^T\right) = \|AX - BX\|_F^2.$$

□

With this characterization, we immediately get a closed form for a representative. Denote with vec the vectorization of a matrix. For any matrix $A \in \mathbb{R}^{n\times n}$ it holds that $\|A\|_F^2 = \|\operatorname{vec}(A)\|^2$.

Theorem 3.3.6. *The unique representative of a pose $P = (R, t)$ for an object without any proper symmetries is given by*

$$\mathcal{R}(P) = \left(\operatorname{vec}(R\Lambda)^T, t^T\right)^T \in \mathbb{R}^{12}, \tag{3.14}$$

where

$$\Lambda := \left(\frac{1}{S}\int_O \mu(x)xx^T\, ds\right)^{1/2}. \tag{3.15}$$

Proof. Using Lemma 3.3.5 we obtain for two poses $P_i = (R_i, t_i)$, $i = 1, 2$

$$\begin{aligned} d_{rot}^2(P_1, P_2) &= \frac{1}{S}\int_O \mu(x)\|R_1x - R_2x\|^2\, ds \\ &= \frac{1}{S}\int_O \mu(x)\operatorname{Tr}\left((R_1 - R_2)xx^T(R_1 - R_2)^T\right)\, ds \\ &= \operatorname{Tr}\left(\frac{1}{S}\int_O \mu(x)(R_1 - R_2)xx^T(R_1 - R_2)^T\, ds\right) \\ &= \operatorname{Tr}\left((R_1 - R_2)\Lambda^2(R_1 - R_2)^T\right) \\ &= \|R_1\Lambda - R_2\Lambda\|_F^2. \end{aligned} \tag{3.16}$$

Altogether we calculate

$$d^2(P_1, P_2) = \|R_1\Lambda - R_2\Lambda\|_F^2 + \|t_1 - t_2\|^2 = \|\mathcal{R}(P_1) - \mathcal{R}(P_2)\|^2.$$

□

We see that in this case the set of representatives only contains one element, so each pose admits a unique representative. The map $\mathcal{R}$ is in this case an isometric embedding from the pose space into $\mathbb{R}^{12}$. The image of $\mathcal{R}$ is given by

$$\mathcal{R}(\mathcal{C}) = \left\{\left(\operatorname{vec}(R\Lambda)^T, t^T\right)^T \mid (R, t) \in SE(3)\right\} \subset \mathbb{R}^{12}.$$

This shows that the embedding is never surjective, as for example $0 \notin \mathcal{R}(\mathcal{C})$ if we assume $\Lambda \neq 0$. This observation shows the difficulty in trying to find a projection for any given point $x \in \mathbb{R}^{12}$. The problem breaks down into finding the closest rotation to an arbitrary matrix. Denote with $R_x \in \mathbb{R}^{n\times n}$ and $t_x \in \mathbb{R}^3$ those elements which satisfy $x = (\operatorname{vec}(R_x)^T, t_x^T)^T$. In other words, we split a 12-D vector into a rotational and a translational part. The first 9 dimensions account for the rotation and the last 3 for the translation. The projection of x is given by

$$\begin{aligned} \operatorname{proj}(x) &= \operatorname*{arg\,min}_P \|x - \mathcal{R}(P)\|^2 && (3.17) \\ &= \operatorname*{arg\,min}_{(R,t)\in SE(3)} \|R_x - R\Lambda\|_F^2 + \|t_x - t\|^2. && (3.18) \end{aligned}$$

This shows that the rotational and translational parts are independent of each other and the minimum is reached for $t = t_x$. This leaves us with minimizing the expression $\|R_x - R\Lambda\|_F$. There is a closed form of this problem, which is due to [Ume91], where a proof can be found.

Theorem 3.3.7. *The solution of the rotational part to the minimization problem (3.18) is given by*

$$\arg\min_{R\in SO(3)} \|R_x - R\Lambda\|_F^2 = USV^T$$

where UDV^T is a singular value decomposition of $R_x\Lambda$ satisfying

$$d_1 \geq d_2 \geq d_3 \geq 0$$

for $D = diag(d_1, d_2, d_3)$ and

$$S = \begin{cases} I & \textit{if } \det(UV) > 0 \\ diag(1,1,-1) & \textit{else.} \end{cases}$$

Moreover, this solution is unique, if $rank(R_x\Lambda^T) \geq 2$.

3.3.3 Objects with finite symmetry group

This section is a continuation of the previous one. Recall that the definition of d was

$$d(P_1, P_2) = \min_{G_1,G_2\in G_O} d_{\text{no\_symm}}(T_1 \circ G_1, T_2 \circ G_2)$$

and the representative of a pose of an object without any proper symmetry was given by

$$\mathcal{R}(P) = \left(\text{vec}(R\Lambda)^T, t^T\right)^T \in \mathbb{R}^{12}. \tag{3.19}$$

Now G_O is finite, so the minimum will be reached, which immediately yields the set of representatives.

Theorem 3.3.8. *For an object with a finite proper symmetry group and a pose $P = (R, t)$ the set of representatives is given by*

$$\mathcal{R}(P) = \left\{\left(\text{vec}(RG\Lambda)^T, t^T\right)^T \mid G \in G_O\right\} \subset \mathbb{R}^{12}. \tag{3.20}$$

Proof. This is straightforward since

$$\begin{aligned} d(P_1, P_2) &= \min_{G_1,G_2\in G_O} d_{\text{no\_symm}}(T_1 \circ G_1, T_2 \circ G_2) \\ &= \min_{p_1\in\mathcal{R}(P_1), p_2\in\mathcal{R}(P_2)} \|p_1 - p_2\|. \end{aligned}$$

□

We have already given the result for the projection of objects without proper symmetries in Theorem 3.3.7. Now this result obviously still holds for objects with finite proper symmetry group.

3.3.4 Objects of revolution without roto-reflection invariance

First, we want to consider objects of revolution without roto-reflection invariance. As usual, we consider the object's center of mass to be the origin. Moreover, we assume that the axis of revolution is the z-axis, i.e. the linear subspace generated by $e_z := [0,0,1]^T$. Let R_z^ϕ be the rotation around e_z of ϕ degrees. Then we can describe the proper symmetry group as

$$G_O = \{R_z^\phi \mid \phi \in \mathbb{R}\}.$$

In that case, the covariance matrix Λ defined in (3.15) admits a special form.

Lemma 3.3.9. *Let O be a revolution object with the axis of revolution being the z-axis. Then Λ is of the form*

$$\Lambda = \begin{pmatrix} \lambda_r & 0 & 0 \\ 0 & \lambda_r & 0 \\ 0 & 0 & \lambda_z \end{pmatrix} \tag{3.21}$$

where $\lambda_r, \lambda_z > 0$.

Proof. We will explicitly calculate the surface integral using standard techniques. Let $\gamma : [a,b] \to \mathbb{R}^3$ be a curve representing the object of revolution, i.e. we have the parameter representation

$$p(\theta,t) = \begin{pmatrix} \gamma_1(t)\cos\theta - \gamma_2(t)\sin\theta \\ \gamma_1(t)\sin\theta + \gamma_2(t)\cos\theta \\ \gamma_3(t) \end{pmatrix}, \theta \in [0,2\pi), t \in [a,b] \tag{3.22}$$

for $\gamma(t) = (\gamma_1(t), \gamma_2(t), \gamma_3(t))$. The partial derivates are given by

$$\frac{\partial p}{\partial \theta} = \begin{pmatrix} -\gamma_1(t)\sin\theta - \gamma_2(t)\cos\theta \\ \gamma_1(t)\cos\theta - \gamma_2(t)\sin\theta \\ 0 \end{pmatrix}$$

and

$$\frac{\partial p}{\partial t} = \begin{pmatrix} \gamma_1'(t)\cos\theta - \gamma_2'(t)\sin\theta \\ \gamma_1'(t)\sin\theta + \gamma_2'(t)\cos\theta \\ \gamma_3'(t) \end{pmatrix}.$$

An elementary computation shows

$$\|\frac{\partial p}{\partial \theta} \times \frac{\partial p}{\partial t}\| = \|\gamma_3'(t)(\gamma_1^2(t) + \gamma_2^2(t)) - \gamma_1(t)\gamma_1'(t) - \gamma_2(t)\gamma_2'(t)\|$$

and we see that this expression is independent of θ. Altogether we obtain

$$\begin{aligned} \int_O pp^T \, ds &= \int_a^b \int_0^{2\pi} pp^T \|\frac{\partial p}{\partial \theta} \times \frac{\partial p}{\partial t}\| \, d\theta dt \\ &= \int_a^b \|\frac{\partial p}{\partial \theta} \times \frac{\partial p}{\partial t}\| \int_0^{2\pi} pp^T \, d\theta dt. \end{aligned}$$

Now this integral can be computed explicitly. Using the fact that

$$0 = \int_0^{2\pi} \cos t \, dt = \int_0^{2\pi} \sin t \, dt = \int_0^{2\pi} \cos t \sin t \, dt$$

shows that Λ is 0 outside of the diagonal. For the inner integrals on the diagonal we obtain

$$\begin{aligned}
&\int_0^{2\pi} (\gamma_1(t)\cos\theta - \gamma_2(t)\sin\theta)^2 \, d\theta = \pi(\gamma_1(t)^2 + \gamma_2(t)^2),\\
&\int_0^{2\pi} (\gamma_1(t)\sin\theta + \gamma_2(t)\cos\theta)^2 \, d\theta = \pi(\gamma_1(t)^2 + \gamma_2(t)^2),\\
&\int_0^{2\pi} \gamma_3(t)^2 \, d\theta = 2\pi\gamma_3(t)^2
\end{aligned}$$

All in all, this yields for the expression of Λ that

$$\Lambda = \int_a^b \|\frac{\partial p}{\partial \theta} \times \frac{\partial p}{\partial t}\| \begin{pmatrix} \pi(\gamma_1(t)^2 + \gamma_2(t)^2) & 0 & 0 \\ 0 & \pi(\gamma_1(t)^2 + \gamma_2(t)^2) & 0 \\ 0 & 0 & 2\pi\gamma_3(t)^2 \end{pmatrix} dt \tag{3.23}$$

which proves our claim. □

With this knowledge, we can simplify the rotational part of the distance. It turns out to simply be a scaled distance of the different revolution axes, seen as 3D vectors.

Theorem 3.3.10. *Let O be a revolution object and $P_i = (R_i, t_i)$ for $i = 1, 2$ two poses with respective representatives. Then*

$$d_{rot}^2(P_1, P_2) = \lambda^2 \|R_1 e_z - R_2 e_z\|^2 \tag{3.24}$$

with $\lambda := \sqrt{\lambda_r^2 + \lambda_z^2}$ where λ_r, λ_z are given by Lemma 3.3.9.

Proof. We have seen in (3.16) that we can express d_{rot} as

$$d_{\mathrm{rot}}^2(P_1, P_2) = \min_{\phi_1, \phi_2} \|R_1 R_z^{\phi_1} \Lambda - R_2 R_z^{\phi_2} \Lambda\|_F^2. \tag{3.25}$$

Since the Frobenius norm is invariant under rotations we can rewrite it with $R := R_2^{-1} R_1$ to obtain

$$d_{\mathrm{rot}}^2(P_1, P_2) = \min_{\phi_1, \phi_2} \|R_z^{-\phi_2} R R_z^{\phi_1} \Lambda - \Lambda\|_F^2. \tag{3.26}$$

Parametrizing R with Euler angles gives us $R = R_z^{\psi_1} R_x^{\theta} R_z^{\psi_2}$. Injecting this into (3.26) and with a change of variables we obtain

$$d_{\mathrm{rot}}^2(P_1, P_2) = \min_{\phi_1, \phi_2} \|R_z^{-\phi_2} R_x^{\theta} R_z^{\phi_1} \Lambda - \Lambda\|_F^2. \tag{3.27}$$

Accoring to Lemma 3.3.9, Λ has a specific diagonal form. Therefore the Frobenius norm can be decomposed into two parts:

$$\|R_z^{-\phi_2} R_x^\theta R_z^{\phi_1}\Lambda - \Lambda\|_F^2 = \lambda_z^2 \underbrace{\|R_z^{-\phi_2} R_x^\theta R_z^{\phi_1} e_z - e_z\|^2}_{=:a_{\phi_1,\phi_2}}$$
$$+\lambda_r^2 \underbrace{(\|R_z^{-\phi_2} R_x^\theta R_z^{\phi_1} e_x - e_x\|^2 + \|R_z^{-\phi_2} R_x^\theta R_z^{\phi_1} e_y - e_y\|^2}_{=:b_{\phi_1,\phi_2}}).$$

With basic calculus, one can evaluate these terms to

$$\begin{aligned} a_{\phi_1,\phi_2} &= 2(1-\cos\theta) \\ b_{\phi_1,\phi_2} &= 4 - 2\cos(\phi_1+\phi_2)(1+\cos\theta). \end{aligned}$$

Minimizing both expressions leads to

$$\min_{\phi_1,\phi_2} a_{\phi_1,\phi_2} = \min_{\phi_1,\phi_2} b_{\phi_1,\phi_2} = 2(1-\cos\theta).$$

Using the fact that

$$2(1-\cos\theta) = \|Re_z - e_z\|^2 = \|R_2^{-1}R_1 e_z - e_z\|^2 = \|R_1 e_z - R_2 e_z\|^2$$

completes the proof, as

$$d_{\text{rot}}^2(P_1,P_2) = (\lambda_z^2+\lambda_r^2)2(1-\cos\theta) = (\lambda_z^2+\lambda_r^2)\|R_1e_z - R_2e_z\|^2.$$

□

This expression allows us immediately to find an expression for the representative.

Corollary 3.3.11. *The representative of a pose $P=(R,t)$ of a revolution object without roto-reflection invariance is given by*

$$\mathcal{R}(P) = \left(\lambda(Re_z)^T, t^T\right)^T \in \mathbb{R}^6, \tag{3.28}$$

where λ is given by Theorem 3.3.10.

Proof. We obtain for poses $P_i = (R_i, t_i)$ for $i=1,2$

$$\begin{aligned} d^2(P_1,P_2) &= \lambda^2\|R_1e_z - R_2e_z\|^2 + \|t_1-t_2\|^2 \\ &= \|\mathcal{R}(P_1) - \mathcal{R}(P_2)\|^2 \end{aligned}$$

if $\mathcal{R}$ is defined as claimed. □

Again, we want to examine the map $\mathcal{R}$ further. Since the norm of Re_z for any rotation R is one, we get that the image is given by

$$\mathcal{R}(\mathcal{C}) = \lambda\mathcal{S}^2 \times \mathbb{R}^3 \tag{3.29}$$

with $\lambda S^2 = \{x\in\mathbb{R}^3 \mid \|x\| = \lambda\}$ being the unit sphere scaled by λ. This leads immediately to the following:

Corollary 3.3.12. *The pose space of a revolution object without roto-reflection invariance is isometrically isomorphic to $\lambda\mathcal{S}^2 \times \mathbb{R}^3$.*

This provides a clue on determining the projection of any given point $x \in \mathbb{R}^6$. Once more, we can decompose it into a form where $x = (a_x^T, t_x^T)^T$, with $a_x, t_x \in \mathbb{R}^3$. Here, a_x signifies the axis and t_x denotes the object's translation. Our objective is to project a_x onto the rotational part of the image of $\mathcal{R}$, essentially scaling it to a length of λ. In other words, we have

$$\begin{aligned}\operatorname{proj}(x) &= \arg\min_P \|x - \mathcal{R}(P)\|^2 \\ &= \arg\min_{t,a\in\mathbb{R}^3,\|a\|=1} \|a_x - \lambda a\|^2 + \|t_x - t\|^2 \\ &= \left((\frac{a_x}{\|a_x\|})^T, t_x^T \right)^T .\end{aligned}$$

This solution to the minimization problem is unique, as long as $a_x \neq 0$. This leads to the following result.

Theorem 3.3.13. *The projection of a given point $x = (a_x^T, t_x^T)^T \in \mathbb{R}^6$, where $a_x \neq 0$ is given by*

$$\operatorname{proj}(x) = \left((\frac{a_x}{\|a_x\|})^T, t_x^T \right)^T . \tag{3.30}$$

3.3.5 Objects of revolution with roto-reflection invariance

The case of revolution objects with roto-reflection invariance, such as a cylinder for example, can be treated very similarly to the one without roto-reflection invariance. We take the same assumptions as in the previous chapter, i.e. the axis of revolution being the z-axis. We denote with R_x^α the rotation by α around the x-axis. Then the proper symmetry group is given by

$$G_O = \left\{ R_x^\alpha R_z^\phi \mid \alpha \in \{0, \pi\}, \phi \in \mathbb{R} \right\}.$$

Theorem 3.3.14. *The representatives of a pose $P = (R, t)$ of a revolution object with roto-reflection invariance are given by*

$$\mathcal{R}(P) = \left\{ \left(\pm\lambda(Re_z)^T, t^T\right)^T \right\} \subset \mathbb{R}^6. \tag{3.31}$$

Proof. Knowing the proper symmetry group, the distance between two poses P_1, P_2 can be written as

$$d(P_1, P_2) = \min_{\alpha_a,\alpha_2\in\{0,\pi\};\phi_1,\phi_2\in\mathbb{R}} d_{\text{no\_symm}} \left((R_1 R_x^{\alpha_1} R_z^{\phi_1} e_z, t_1), (R_2 R_x^{\alpha_2} R_z^{\phi_2} e_z, t_2) \right).$$

Using Corollary 3.3.11 the distance of two poses $P_i = (R_i, t_i)$ for a revolution object can be expressed as

$$d(P_1, P_2) = \min_{\alpha_1,\alpha_2\in\{0,\pi\}} \|p_1^{\alpha_1} - p_2^{\alpha_2}\|, \tag{3.32}$$

where $p_i^\alpha = \left(\lambda(R_i R_x^\alpha e_z)^T, t_i^T\right)^T$. Now we have $R_x^0 e_z = e_z$ and $R_x^\pi e_z = -e_z$ which proves the theorem. □

The question for the projection can be immediately answered with the result for revolution objects without roto-reflection invariance. The given projection consisting of a scaled axis and the translation of the center of mass also holds if the object admits roto-reflection invariance. Therefore we can immediately deduce the result from Corollary 3.3.13.

3.3.6 Symmetries within representatives

We have already seen that objects with non-trivial finite proper symmetry group and revolution objects with roto-reflection invariance have multiple representatives for a single pose. This is because the information about the symmetry is not hidden in the representative itself but in the number of them. It is not surprising that the representatives themselves admit symmetries as well in the ambient space $\mathbb{R}^N$. Here we will especially focus on the case of a finite proper symmetry group. The set of representatives was given by

$$\mathcal{R}(P) = \left\{ \left(\operatorname{vec}(RG\Lambda)^T, t^T\right)^T \mid G \in G_O \right\} \subset \mathbb{R}^{12}. \tag{3.33}$$

Now it is easy to define a symmetry group on $\mathcal{R}(P)$ for a pose P.

Definition 3.3.15. Let O be an object with finite proper symmetry group G_O. Then the set

$$G_{\mathcal{R}} \coloneqq \{ s_G \mid G \in G_O \} \tag{3.34}$$

where

$$\begin{aligned} s_G : \mathbb{R}^{12} &\to \mathbb{R}^{12} \\ (\operatorname{vec}(M)^T, t^T)^T &\mapsto (\operatorname{vec}(MG)^T, t^T)^T \end{aligned}$$

forms a group with the composition operation.

Remark. The property of $G_{\mathcal{R}}$ being a group follows directly from the fact that G_O is a group.

It would be desirable that for a representative r the set $\{s_G(r) \mid G \in G_O\}$ is the whole set of representatives. Before we can prove that, we need a lemma.

Lemma 3.3.16. *For any proper symmetry G we have*

$$\Lambda G = G\Lambda. \tag{3.35}$$

Proof. By the definition of Λ^2 we get

$$G\Lambda^2 = \frac{1}{S} \int_O \mu(x) G x x^T \, ds.$$

Now we can transform the integral via $x \leftarrow G^{-1}x$ and since $G(O) = O$ and the invariance of μ this evaluates to

$$\begin{aligned} G\Lambda^2 &= \frac{1}{S}\int_O \mu(x)x(G^{-1}x)^T\, ds. \\ &= \frac{1}{S}\int_O \mu(x)xx^T G^{-T}\, ds = \Lambda^2 G. \end{aligned}$$

In this calculation we used that G is a rotation, i.e. $G^{-T} = G$. Now Λ^2 is positive semi-definite and therefore admits an eigenvalue decomposition $\Lambda^2 = UDU^T$ where D is diagonal and $U \in SO(3)$. Injecting this decomposition into the previous equation leads to

$$\Lambda^2 = (G^T U)D(G^T U)^T$$

which shows that $G^T U$ is also an eigenbasis of Λ^2. Since Λ is the principal square root of Λ^2 it shares the same eigenspace. Thus

$$\Lambda = (G^T U)D^{1/2}(G^T U)^T = G^T U D^{1/2} U^T G = G^T \Lambda G \tag{3.36}$$

and Λ commutes with G. □

Theorem 3.3.17. *The group $G_{\mathcal{R}}$ contains exactly as many elements as $\mathcal{R}(P)$ for any pose P. For any representative $r \in \mathcal{R}(P)$ it holds that*

$$\{s(r) \mid s \in G_{\mathcal{R}}\} = \mathcal{R}(P). \tag{3.37}$$

Proof. Let $(\mathrm{vec}(R\Lambda)^T, t^T)$ where $R \in \mathbb{R}^{3\times 3}$ and $t \in \mathbb{R}^3$ be a representative of a pose P. According to Lemma 3.3.16 we have

$$s_G((\mathrm{vec}(R\Lambda)^T, t^T)^T) = (\mathrm{vec}(R\Lambda G)^T, t^T)^T = (\mathrm{vec}(RG\Lambda)^T, t^T)^T. \tag{3.38}$$

Hence the left-hand side is a representative of the same pose. □

The elements of $G_{\mathcal{R}}$ themselves are special transformations of the ambient space $\mathbb{R}^N$.

Proposition 3.3.18. *Every $s \in G_{\mathcal{R}}$ is a bijective, isometric, linear transformation of $\mathbb{R}^N$.*

Proof. The linearity is clear, bijectivity can be seen since $(s_G)^{-1} = s_{G^{-1}}$ for any proper symmetry $G \in G_O$. Moreover, any $G \in G_O$ preserves the norm of a vector which implies the isometry of s. □

Remark. For revolution objects with roto-reflection, which admit two representatives, the same results can be deduced with

$$G_{\mathcal{R}} \coloneqq \{s_\delta \mid \delta = \pm 1\} \tag{3.39}$$

where

$$\begin{aligned} s_G &: \mathbb{R}^6 \to \mathbb{R}^6 \\ (a^T, t^T)^T &\mapsto (\delta a^T, t^T)^T. \end{aligned}$$

3.4 Pose averaging

The problem of averaging poses arises in various applications, such as denoising or interpolation. While it is straightforward to define an average for the translational part, it gets rather complicated for the orientation, since $SO(n)$ is not a vector space. One way to generalize the average to non-vector spaces is to consider the Fréchet mean.

Definition 3.4.1. Let $S = \{P_i\}_{i=1}^n$ be a finite set of poses and $w_i > 0$ strictly positive weights for $i = 1, ..., n$. For any metric $\hat{d}$ on the pose space the expression

$$\Phi(P) = \sum_{i=1}^{n} w_i \hat{d}^2(P_i, P) \tag{3.40}$$

is called the Fréchet variance at P with weights w_i. The Fréchet mean is given by

$$\mu_f(S) = \arg\min_{P \in \mathcal{C}} \Phi(P), \tag{3.41}$$

i.e. minimizing the Fréchet variance over all poses.

This expression is not necessarily well-defined since the minimum does not have to be unique. Such cases would occur in a few configurations like averaging two poses of opposite axes for a revolution object that does not admit roto-reflection invariance. However, such cases do typically not arise in real-world scenarios where a meaningful average is desired. For objects that do not admit any symmetries, pose averaging has been extensively studied by [SWR10]. They compare the differences for various metrics $\hat{d}$ on the rotation part. However, these attempts do not take the symmetry of objects into account. This is why we want to use our symmetry-aware metric d for the pose space.

Moreover, in the previous section, we established a deep connection between the pose space and some Euclidean space $\mathbb{R}^N$, namely via representatives and projections. Now, while the pose space is not a vector space, $\mathbb{R}^N$ is. Building an average there can be done by the arithmetic mean and then we can project back to the pose space. The only thing that has to be taken care of is that there can be multiple representatives of a pose in $\mathbb{R}^N$.

Theorem 3.4.2. *Let $S = \{P_i\}_{i=1}^n$ be a finite set of poses and $w_i > 0$ strictly positive weights for $i = 1, ..., n$. For a tuple $R = (r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ of representatives define*

$$m_R := \frac{\sum_i w_i r_i}{\sum_i w_i} \tag{3.42}$$

as the weighted arithmetic mean of R. Then

$$\mu_f(S) = \arg\min_{P \in \mathcal{A}} \Phi(P), \tag{3.43}$$

where

$$\mathcal{A} := \left\{ \operatorname{proj}(m_R) \mid R \in \prod_i \mathcal{R}(P_i) \right\}. \tag{3.44}$$

The case of objects admitting only a single representative simplifies to

$$\mu_f(S) = \operatorname{proj}\left(\frac{\sum_i w_i \mathcal{R}(P_i)}{\sum_i w_i}\right). \tag{3.45}$$

Proof. Plugging in the definition of the Fréchet variance leads to

$$\Phi(P) = \sum_i w_i \min_{r_i \in \mathcal{R}(P_i), r \in \mathcal{R}(P)} \|r_i - r\|^2. \tag{3.46}$$

Thanks to the symmetry of the distance we have for all $r \in \mathcal{R}(P)$

$$\Phi(P) = \sum_i w_i \min_{r_i \in \mathcal{R}(P_i)} \|r_i - r\|^2 = \min_{(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)} \sum_i w_i \|r_i - r\|^2. \tag{3.47}$$

Now we can develop the sum further using the arithmetic mean:

$$\begin{aligned}\sum_i w_i \|r_i - r\|^2 &= \sum_i w_i \|r_i - m_R + m_R - r\|^2 \\ &= \sum_i w_i \|r_i - m_R\|^2 + \sum_i w_i \|m_R - r\|^2 + \underbrace{2\sum_i w_i \langle r_i - m_R, m_R - r\rangle}_{=0}.\end{aligned}$$

Therefore, for a given tuple $(r_i)_{i=1}^n$ the minimization problem (3.41) splits into two terms, where the first sum is independent of r. Minimizing the second sum corresponds exactly to the projection to the vector m_R, which proves the claim. The statement for objects admitting only one representative can be deduced immediately. □

Theorem 3.4.2 provides an important tool to calculate the average of finitely many poses. Especially in the case of objects having only one representative, it breaks down to a calculation of an arithmetic mean in $\mathbb{R}^N$ and projecting back to the pose space. This can be done quickly and efficiently and requires no optimization techniques. Now we want to study the case of an object admitting multiple representatives.

One possible approach would be just to try all possible combinations and brute force it but since the number of possible tuples grows exponentially this is very inefficient. One method to circumvent such a problem is to consider only *consistent* tuples. While there are various definitions of consistency amongst tuples, we rely on the one given by [BDLC18] since it is not ill-defined as we will show.

Definition 3.4.3. A tuple $(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ is said to be consistent if

$$\text{for all } i, j = 1, ..., n \text{ and for all } q_j \in \mathcal{R}(P_j) \setminus \{r_j\} : \|r_j - r_i\| < \|q_j - r_i\|. \tag{3.48}$$

To summarize this definition, a consistent tuple is a set of representatives that are closer to each other than to every other representative.

Consistent tuples are unique, up to symmetry.

Proposition 3.4.4. *Let $(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ be a consistent tuple. Then the set of all consistent tuples is given by $\{(s(r_i))_{i=1\dots n} \mid s \in G_{\mathcal{R}}\}$.*

Proof. Let $(r_i)_{i=1}^n, (q_i)_{i=1}^n \in \prod \mathcal{R}(P_i)$ be two consistent tuples. If $r_i = q_i$ for some i, then (3.48) would lead to $\|r_i - r_j\| < \|r_i - r_j\|$ for some j. Thus, $(r_i)_{i=1}^n$ and $(q_i)_{i=1}^n$ must be pairwise disjoint. Now according to Theorem 3.3.17, we have exactly $|\mathcal{R}(.)|$ different representative combinations symmetric to $(r_i)_{i=1}^n$. Those are given by the set

$$\{(s(r_i))_{i=1}^n \mid s \in G_{\mathcal{R}}\}. \tag{3.49}$$

Now every $s \in G_{\mathcal{R}}$ is a linear transformation, implying that $(s(r_i))_{i=1}^n$ is again consistent. □

It is not surprising that the projection is invariant under the symmetry of representatives.

Theorem 3.4.5. *Let $x \in \mathbb{R}^{12}$ and $s \in G_{\mathcal{R}}$. Then*

$$\operatorname{proj}(x) = \operatorname{proj}(s(x)). \tag{3.50}$$

Proof. As usual, we split $x \in \mathbb{R}^{12}$ in a way such that $x = (\operatorname{vec}(M)^T, t^T)^T$ for $M \in \mathbb{R}^{3\times 3}$ and $t \in \mathbb{R}^3$. Let $s_G \in G_{\mathcal{R}}$ for a proper symmetry $G \in G_O$, then we can write

$$s_G(x) = \left(\operatorname{vec}(MG)^T, t^T\right)^T.$$

In Theorem 3.3.7, we have already seen the form of the solution of the projection problem of objects with finite symmetry group. Namely, the projection of x is the pose (R, t), with $R = USV^T$, where $UDV^T = M\Lambda$ is an SVD decomposition. To obtain the projection of $s_G(x)$, one needs the SVD of $MG\Lambda$. With the help of Lemma 3.3.16 we can simply plug in the SVD of R and obtain

$$MG\Lambda = M\Lambda G = UDV^T G = UD\bar{V}^T$$

with $\bar{V} = G^T V$. Keep in mind that since G and V are orthogonal, also $\bar{V}$ is orthogonal. Therefore we have

$$\operatorname{proj}(s_G(x)) = (US\bar{V}^T, t) = (RG, t) = (R, t) = \operatorname{proj}(x).$$

□

It turns out to be useful to define the minimal distance between multiple representatives of the same pose.

Definition 3.4.6. Let P be any pose and $r \in \mathcal{R}(P)$. The minimum distance T between representatives is defined as

$$T := \min_{q \in \mathcal{R}(P), q \neq r} \|r - q\|. \tag{3.51}$$

If an object admits only a single representative per pose we have the convention $T := \infty$. Notice that this definition is independent of the choice of pose and representative due to the underlying symmetry.

This definition allows us to find a simpler characterization for the consistency of a tuple.

Lemma 3.4.7. *Let $(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ be a tuple of pose representatives. If*

$$\textit{for all } i, j = 1, ..., n : \; \|r_i - r_j\| < T/2 \tag{3.52}$$

then the tuple is consistent.

Proof. Suppose that the tuple $(r_i)_{i=1}^n$ satisfies (3.52). Then we have for any $q_j \in \mathcal{R}(P_j)\backslash\{r_j\}$ that with the definition of T

$$2\|r_i - r_j\| < T \leq \|q_j - r_j\| \leq \|q_j - r_i\| + \|r_i - r_j\| \tag{3.53}$$

and hence

$$\|r_i - r_j\| < \|q_j - r_i\|$$

for any i which is exactly the condition for being consistent. □

Lemma 3.4.8. *Let $(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ be a tuple of pose representatives. If there is a $c \in \mathbb{R}^N$, such that*

$$\textit{for all } i = 1, ..., n : \; \|r_i - c\| < T/4, \tag{3.54}$$

then the tuple is consistent.

Proof. This is a simple consequence of the Lemma 3.4.7, since for any $i, j = 1, ..., n$

$$\|r_i - r_j\| \leq \|r_i - c\| + \|r_j - c\| < T/2.$$

□

We now have all the tools to give a meaningful definition of the mean of several poses for objects with multiple representatives. This construction relies on consistent tuples. We want to emphasize that consistent tuples do not always exist, hence that definition does not work for every arbitrary set of poses. However, we believe such cases do not have a strong physical meaning.

Theorem 3.4.9. *Given a consistent tuple $(r_i)_{i=1}^n \in \prod_i \mathcal{R}(P_i)$ of representatives and positive weights w_i, we can define the mean via*

$$\widehat{mean}(S) := \operatorname{proj}\left(\frac{\sum_i w_i r_i}{\sum_i w_i}\right). \tag{3.55}$$

Proof. We need to show that this is well-defined. According to Proposition 3.4.4 all consistent tuples are given by $(s(r_i))_{i=1}^n$ for $s \in G_{\mathcal{R}}$. The equation

$$\operatorname{proj}\left(\frac{\sum_i w_i s(r_i)}{\sum_i w_i}\right) = \operatorname{proj}\left(s\left(\frac{\sum_i w_i r_i}{\sum_i w_i}\right)\right) = \operatorname{proj}\left(\frac{\sum_i w_i r_i}{\sum_i w_i}\right) \tag{3.56}$$

shows that the mean is independent of the chosen consistent tuple, hence the expression is well-defined. □

Example 3.4.10. Consider as an object a cylinder. The two representatives of this object correspond to the two directions of the axis. When averaging two poses, one simply averages the vectors that define the axis of the poses. However, one has to be careful with the sign of the vectors. As depicted in Figure 3.1, the average pose (given by the purple vector) can be obtained by averaging the red and the blue representative. This leads to the meaningful average of the two poses and is according to our consistency definition since these two representatives are closer to each other than all the others. Violating this by averaging the yellow and the blue representatives would lead to a meaningless interpretation of the mean, shown with the green vector.

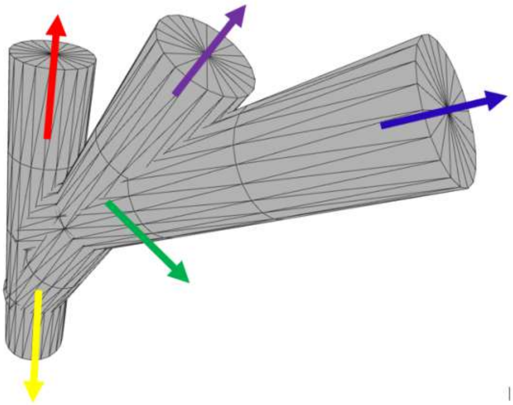


Figure 3.1: Averaging two poses of a cylinder

Remark. The fact that this definition of the mean corresponds to the Fréchet mean as defined in the minimization problem (3.41) is very likely, but we were not able to find a proof for it. We believe this relies on the fact that given a consistent tuple, one can extend this tuple by any weighted mean (meaning any point in the convex hull of these points) and get a consistent tuple again. It is not easy to find an exact description of consistent tuples as the rotational part of the representatives lies on a sphere in $\mathbb{R}^9$.

4 Robust Pose estimation

In this chapter, we aim to use our metric combined with our results of pose averaging to tackle the problem of instance detection and robust pose estimation. This task is prevalent in computer vision and robotics and has various applications, such as augmented reality, object tracking, or gesture recognition. A broad survey can be found in [CF01].

4.1 Problem description

The task involves locating multiple instances of a given object within a 3D scene and determining their respective poses. Such a 3D scene can come from various sources, such as laser scans, or stereo systems, the data are typically presented as point clouds. To illustrate this concept, consider the example depicted in Figure 4.1 where we have the scan of a section of a trailer. Our objective is to identify the poses of the two wheels. In this scenario, the wheel serves as the object, and the goal is to detect the two instances of it in the point cloud and estimate their poses relative to a fixed coordinate frame.

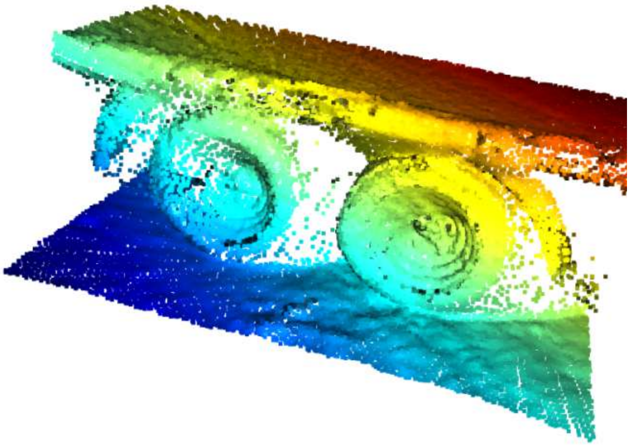


Figure 4.1: Point cloud of a trailer, obtained by LIDAR

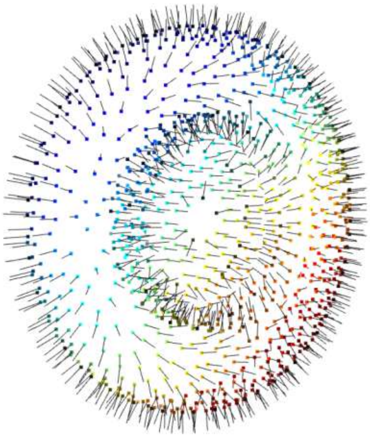


Figure 4.2: Model of a wheel

However, this problem imposes several challenges, including occlusions, clutter, and lighting variations. Therefore, we require our method to be robust. Another difficulty is the visibility of the wheels, they can only be seen from one side, so a careful choice of the model is crucial. One possibility is depicted in Figure 4.2, which was modeled in Blender [Com18], and 1000 points were randomly sampled from the surface. The wheel sets an additional difficulty, since it admits symmetries, making the pose described in the classical sense as a rotation and translation not unique.

Global approaches to robust pose estimation, such as [PGBP10], [SWK07], and [RVDH05], would typically not be affected by the symmetry of the object. However, these are typically not very efficient or precise. Especially in big point clouds the execution time explodes. On the other hand, local approaches, as [DUNI10], [HLI$^+$13], [HLRK16], [CJ97], [RBB09], seem to be more fruitful. Such methods are mostly based on local invariant features. Roughly speaking, these methods look for local structures of the object in the scene and try to estimate the pose from that. They follow an approach to generate votes for pose candidates, to obtain a distribution of poses that describes the ground truth. Then, the main modes of this distribution are identified, which then hopefully correspond to the actual poses of the instances. Detecting these modes is not an easy task, since the pose space has a high dimensionality and does not admit a vector space structure. Mode detection usually requires a notion of distance, or at least similarity, and this is where common approaches with metrics on $SE(3)$ fail.

Authors frequently overlook symmetric objects, despite their prevalence in real-world scenarios. Bottles, glasses, tables, and boxes are a few examples of things in our daily life that admit symmetries. However, a few attempts were made to deal with symmetric objects. Some focus on geometric primitives [DI15] such as cylinders or spheres, while others on rotationally symmetric objects [HKLM22], [dFMB15]. Nevertheless, most approaches either specify one type of symmetry or only work for geometric primitives for which they are specifically designed. We aim to develop a method that works for all types of objects, regardless of their symmetry.

4.2 Generalized Hough Transfrom with PPF

To achieve this goal, we stick to a method developed by [DUNI10], which is known as a popular approach for robust pose estimation that works well and is reliable. We aim to extend this method to function effectively with symmetric objects which we believe it does not.

They follow a Hough-like procedure to obtain a set of *candidate poses*. Each of these candidate poses represents a pose where the algorithm is confident it is close to one instance of the ground truth. In the final step, these candidate poses are clustered, where each cluster represents one pose of an instance. Poses in each cluster are averaged to obtain the final pose of each instance. Clustering as well as averaging accounts for the robustness of the method.

Our goal is to use our notion of distance in the pose space to cluster and to use our results of pose averaging. With that approach, we hope to be able to improve the existing method of [DUNI10]. They cluster in a way such that translation and rotation do not differ more than a given pre-defined threshold, but this does not take the symmetry of objects into account. Multiple poses of the same instance could be obtained. Therefore, we aim to use our proposed symmetry-aware distance to cluster the candidate-poses, to be able to deal with symmetric objects. The next section will describe the process of obtaining these candidate poses.

4.3 Obtaining candidate poses

The model and the scene are assumed to be finite point clouds admitting normals. If the data is given in the form of a mesh, such a representation can be easily computed. Having point clouds only, normals can be estimated in various ways. While there exist more sophisticated methods like [BM12], standard techniques like fitting a plane to k nearest neighbors can be applied as well. Due to readability reasons, we denote points of the model $\mathcal{M}$ with m_i and similarly points of the scene $\mathcal{S}$ with s_i.

The whole process of obtaining candidate poses consists of two phases: the *off-line* and the *on-line* phase. During the off-line phase, point pair features are used to obtain a global description of the model. Those point pair features will be described in more detail in the next section. One big advantage is that the off-line phase is independent of the scene and therefore has to be computed only once, which saves a lot of time. In the on-line phase, a set of *reference points* are chosen at random from the scene. The point pair feature of each reference point with every other point in the scene is computed and searched in the global model description. Each match votes for a specific pose to obtain a set of potential poses in a Hough-like manner. With this voting process, one obtains the optimal object pose for each reference point.

4.3.1 Point Pair Feature

Although multiple variations of point pair features exist, we use the one defined in [DUNI10]. A point pair feature (PPF) describes the relative position and orientation of two points. For any two points $p_1, p_2 \in \mathbb{R}^3$ with normals n_1, n_2 we set $d = p_2 - p_1$. With $\angle(v, w) \in [0, \pi]$ we denote the angle between two vectors v and w. We define a feature F with

$$F(p_1, p_2) = (\|d\|, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)) . \tag{4.1}$$

This feature will be used to make a global model description. As it will be essential to compare features efficiently, we introduce a discretized version. For this, we simply bin the distances and the angles. Fix $d_{dist} \in \mathbb{R}$ and $d_{angle} \in \mathbb{R}$ as the stepsize of the distance and the angle respectively. A reasonable choice for d_{angle} would be $d_{angle} = 2\pi / n_{angle}$ where n_{angle} is the number of desired bins. Then we define the discretized feature F_d via

$$F_d(p_1, p_2) = \left(\left\lfloor \frac{\|d\|}{d_{dist}} \right\rfloor , \left\lfloor \frac{\angle(n_1, d)}{d_{angle}} \right\rfloor \left\lfloor \frac{\angle(n_2, d)}{d_{angle}} \right\rfloor \left\lfloor \frac{\angle(n_1, n_2)}{d_{angle}} \right\rfloor \right) \in \mathbb{N}^4. \tag{4.2}$$

4.3.2 Global model description

This step is the off-line phase and has to be computed only once for a model $\mathcal{M}$. The main idea is to calculate the point pair feature between every possible pair of points in the model M and group point pairs with similar features together. Formally speaking, the global model description is a map $G : \mathbb{N}^4 \to 2^{\mathcal{M} \times \mathcal{M}}$ defined as

$$G(n_1, n_2, n_3, n_4) = \{(m_i, m_j) \mid F_d(m_i, m_j) = (n_1, n_2, n_3, n_4)\} . \tag{4.3}$$

The purpose of this map is to have quick access to all point pairs with similar PPF. In practice, this map can be seen as a hash table with the input (n_1, n_2, n_3, n_4) being the key.

Now if we have a scene pair $(s_i, s_j) \in \mathcal{S}^2$ we can simply calculate $F_d(s_i, s_j)$ and use this as a key to find all point pairs of the model with the same feature in constant time. Note that this set can possibly be empty.

4.3.3 Voting scheme

In the next step, a pre-defined number of scene points is chosen. We call this subset $\mathcal{R} \subset \mathcal{S}$ the *reference points*. If we assume $s_r \in \mathcal{R}$ is a point that lies on the object we want to detect, then there must be a corresponding point $m_r \in \mathcal{M}$. After aligning those points together with their normals, we are left with one degree of freedom to align the model with the scene. This is a rotation around the normal of s_r. We call a pair (m_r, α) the *local coordinates* of the model with respect to s_r, where α is exactly this rotation angle.

In the method, we want to align a point pair $(m_r, m_i) \in \mathcal{M}^2$ to a scene pair (s_r, s_i) where both pairs have a similar feature vector, i.e. $F_d(m_r, m_i) = F_d(s_r, s_i)$. Let $T_{x \to g}$ be the transformation that translates the point x to the origin and rotates the normal associated to x onto the x-axis. Then the transformation from the local model coordinates to the scene coordinates can be written as

$$s_i = T_{s \to g}^{-1} R_x^{\alpha} T_{m \to g}(m_i) \tag{4.4}$$

where R_x^{α} is a rotation by α around the x-axis. Now the goal is to find optimal local coordinates of a fixed reference point such that the number of scene points on the model is maximized. For this procedure, a method that is similar to a Generalized Hough Transform is used. For every reference point $s_r \in \mathcal{R}$ a so-called *accumulator array* is generated and filled with zeros. This array consists of $|\mathcal{M}|$ rows and n_{angle} columns. Recall that n_{angle} was the number of bins we used to discretize the angle. The accumulator array can be seen as the discretized space of local coordinates for a fixed reference point. For the actual voting process, the point s_r is paired with every other scene point $s_i \in S$. The feature $F_d(s_r, s_i)$ is calculated and searched for in the model. This is done via the global model description, i.e. $G(F_d(s_r, s_i))$ is calculated. The set of points with similar features is obtained and thus the question of where on the model the pair (s_r, s_i) could be is answered. As we know from the description of the local coordinates, this information is not enough, as we still need to compute the rotation angle α. Subsequently, for every pair $(m_r, m_i) \in G(F_d(s_r, s_i))$ the rotation angle α is obtained via (4.4). In the final step, the vote is cast in the accumulator array at the index (m_r, α) by increasing the value there by one. Repeating this procedure for all scene points s_i finishes the process of obtaining the accumulator matrix. Finally, the accumulator can be searched for maxima. For stability reasons, it is advised to take all peaks up to a certain threshold which is relative to the maximum peak. From each of these few peaks, the global pose can be retrieved to obtain so-called *candidate poses*. Each of those candidate poses comes with a score, namely the value of the peak in the accumulator array, which is an indication of how sure the algorithm is that a pose might be a correct one. This score can be used in the next step.

An illustration of such candidate poses can be seen in Figure 4.3. As a model, we used a rocket, where we cut off the upper part for efficiency reasons. The scene contains two copies of this rocket, lying somewhere in space. A total of 58 poses was retrieved by the procedure described above.

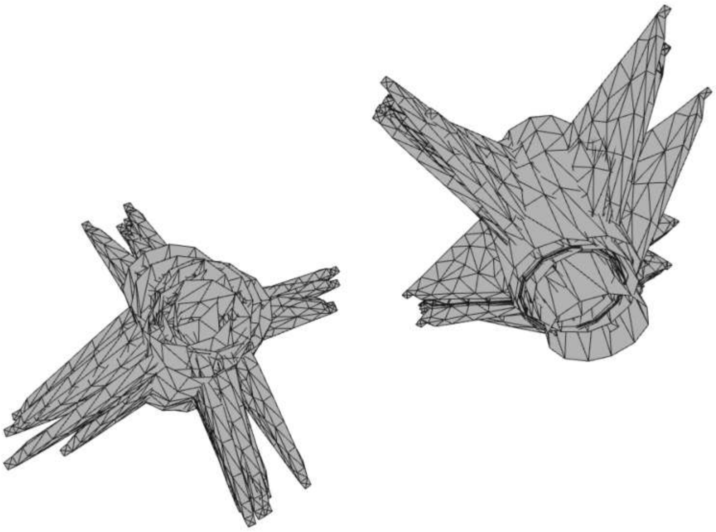


Figure 4.3: Candidate poses of a rocket

One can see that the candidate poses clearly form two clusters, each of them roughly describing one instance of the ground truth.

4.4 Clustering candidate poses

In the step above we assumed that the chosen reference point lies on the model. However, this does not need to be the case, as the scene may contain noise. To ensure that at least some reference points are on the objects that are to be detected, the set $\mathcal{R}$ has to be chosen reasonably large. Unfortunately, this leaves us with some incorrect poses. On the other hand, we have several poses within the candidate poses that only approximate the ground truth. For example, it would be a possibility to choose the candidate pose with the highest score as a final result. However, this approach is only of limited usage, since there could be multiple instances of the object in the scene, which would not be detected then. Another downside is that even the pose with the highest score might not be that accurate, due to rounding errors during the discretization.

An approach to increase the overall accuracy of the final result and to filter out incorrect poses is to cluster the candidate poses. Formally speaking, a clustering is a set of subsets. Let us denote with $\mathcal{K} = \{P_i\}_{i=1,...,n}$ the set of candidate poses. Then a clustering is a set of subsets $C_1, ..., C_n$ such that

$$\mathcal{K} = \bigcup_{i=1}^{n} C_i \text{ and } C_i \cap C_j = \emptyset \text{ for } i \neq j.$$

Such a structure is also known in mathematics as partition and it is clear that each candidate pose is assigned to exactly one cluster. Every cluster C_i represents one instance of the object in the scene, which can be more than one. Let us denote the score obtained by the Hough voting of the pose P_i with k_i. We define the score K_i of cluster C_i as

$$K_i := \sum_{P_j \in C_i} k_j \tag{4.5}$$

i.e. as the sum of all scores of the poses in that cluster. Incorrect poses likely form clusters with rather low scores, and can be easily filtered out. One example is using a threshold relative to the highest score of all clusters, accepting only clusters with a score higher than or equal to $c \cdot \max_i K_i$ for some constant $c \in [0, 1]$. While $c = 0$ corresponds to accepting every cluster, $c = 1$ means only accepting the best (ones).

Since we use a slightly different approach than [DUNI10], we want to describe the clustering procedure more in detail. While they cluster in a way such that poses in one cluster do not differ in translation and rotation more than a predefined threshold, this could lead to incorrect results in the case of symmetric objects. Multiple clusters would account for the same instance, delivering a wrong result of the algorithm. To solve this issue, we aim to use our symmetry-aware distance. However, many clustering algorithms rely on the fact that the underlying space is a vector space, which the pose space is not. A possible idea would be to use representatives, but as they are not unique in some cases this would cause several problems and be rather inefficient.

Instead, we propose a simple hierarchical clustering algorithm based on our proposed distance. This is due to several reasons. Clustering itself is an extensively studied topic, a broad survey of clustering can be found in [XW05], and it is not trivial how to find the best clustering. In fact, it always depends on the specific problem which form of clustering is better than the other, and in our case, we are trying to find obvious clusters, which is why we waive more sophisticated methods.

Our procedure reads as follows, and can be found in Algorithm 1. The candidate poses P_i are sorted in descending order with respect to their score k_i. Now we choose the first element in the list, i.e. P_1 and put all poses into one cluster whose distance to that initial pose is smaller than a predefined threshold K. So we have

$$C_1 = \{P_i \mid d(P_1, P_i) < K \text{ for } i = 2, ..., n\}.$$

Then we remove all those elements from the list and repeat this procedure. So the element with the highest score which is not contained in the first cluster becomes the initial pose for the second and so on. This is done until the list is empty and each element is assigned to one cluster.

It should be noted that this algorithm requires a predetermined threshold, which can vary a lot for different forms of problems. Nonetheless, even advanced techniques, such as DBSCAN, (see [KRA$^+$14]), necessitate an additional parameter, rendering it a challenge that is difficult to circumvent. One approach to simplify this is to scale the model (and the scene accordingly) to have a diameter of 1. That way, it is easier to compare results and set them into relation.

4.5 Recovering poses

Having obtained clusters in the previous step, we subsequently need to recover the final pose of each cluster. Given that every pose in the cluster is an approximation of the ground

Algorithm 1 Clustering of candidate poses

```
Require: $[P_1, ..., P_n]$ a list of candidate poses, sorted in descending order according to
    their scores; symmetry-aware distance function $d$; threshold $K > 0$
Ensure: Clusters of poses, each cluster standing for one instance of an object
$sortedPoses \leftarrow [P_1, ..., P_n]$
$finalClusters \leftarrow$ empty list
while $sortedPoses$ is not empty do
    $cluster \leftarrow$ empty list
    $bestPose \leftarrow sortedPoses[1]$
    for $P_i$ in $sortedPoses$ do
        if $d(P_i, bestPose) < K$ then
            add $P_i$ to $cluster$
            remove $P_i$ from $sortedPoses$
        end if
    end for
    add $cluster$ to $finalClusters$
end while
return $finalClusters$
```

truth, it is natural to calculate the average of all poses contained in one cluster. Again, this problem is not trivial, since the pose space is in general not a vector space. While averaging the translational part is straightforward, it remains complicated for the rotational part. Several methods exist, for example, using SVD or rotation quaternions, see [Gra01]. We could not determine which method was used by [DUNI10]. However, to our knowledge, these methods also do not take symmetries into account, which is why we aim to use the theory obtained in Section 3.4. We keep the notation from the previous section, i.e. we have clusters $\{C_i\}_{i=1,...,m}$ with $C_i = \{P_1, ..., P_{n_i}\}$. With k_j being the score of P_j and a tuple $(r_j)_{j=1,...,n_i} \in \prod_{P_j \in C_i} \mathcal{R}(P_j)$, we can define the recovered pose $\mathcal{P}_i$ of cluster C_i as

$$\mathcal{P}_i \coloneqq \text{proj}\left(\frac{\sum_{j=1}^{n_i} k_j r_j}{\sum_{j=1}^{n_i} k_j}\right), \tag{4.6}$$

the weighted average of the poses within that cluster. Now a consistent tuple does not necessarily need to exist, but this problem can be circumvented by choosing a threshold K for clustering that is small enough.

Proposition 4.5.1. *If the threshold cluster K in Algorithm 1 is chosen to be smaller than $T/4$, with T as defined in (3.51), then there exists a consistent tuple for each cluster.*

Proof. Let $C = (P_1, ..., P_m)$ be a cluster, sorted by score, such that P_1 has the highest score. According to the algorithm we have for all $i = 2, ..., m$ that

$$d(P_1, P_i) = \min_{r_1 \in \mathcal{R}(P_1), r_i \in \mathcal{R}(P_i)} \|r_1 - r_i\| \leq K < T/4. \tag{4.7}$$

Choose one representative r_1 of P_1. According to Lemma 3.2.4 and (4.7) we can choose $r_i \in \mathcal{R}(P_i)$ such that

$$\|r_1 - r_i\| < T/4,$$

since the set of representatives is always finite and the minimum will be reached. But now r_1 is a proper choice for the center of a ball with a radius smaller than $T/4$ such that all representatives lie inside. This means the conditions for Lemma 3.4.8 with $c = r_1$ are fulfilled and therefore the tuple $(r_i)_{i=1,\ldots,m}$ is consistent. □

This constraint to the clustering gives us a solid theoretical foundation since the average is well-defined. One can waive the weighting of the average, but we believe it adds to the accuracy if we weigh the stronger candidate poses more. This step ensures that the algorithm is more accurate, and we believe the weighting of the average has not been done before.

5 Experiments

This chapter is devoted to applying the algorithm developed in the previous chapter. We aim to analyze the effect of the pose clustering, Section 4.4, which can be seen as a post-processing procedure. Here we differ from the method of [DUNI10]. The main focus lies on objects admitting symmetries, as the original paper ignores this case. However, symmetrical objects often appear in daily life as well and are not specially made up by mathematicians as an edge case. Things like glasses, bottles, boxes, pens, even the shape of the paper of this work - they all admit symmetries.

5.1 Methodology

As we have mentioned above, the focus lies on where we differ from the algorithm of [DUNI10]. Acquiring suitable candidate poses can present a significant challenge, entailing several difficulties. For example, the quality of the normals directly influences the candidate's quality, since they stand for the object's orientation in space. In practice, normals are often estimated, usually by fitting a plane through a neighborhood of the k nearest points. This works well as long as the surface is smooth enough but can lead to terrible results if there are fine details or corners. Consequently, if the normals are of bad quality, the recovered pose of the object is as well.

Other problems arising in real-world applications are the emergence of noise and clutter. As shown in Figure 4.1, such things appear naturally when scanning an object with a laser. Approaches made by [HLRK16] and [WYL20] tackle this problem by introducing new sampling and voting schemes. To sum it up, finding suitable candidates is an art in itself, and many improvements and adaptations have been made in that area (see [BI15], [VLM18] for example). However, studying all of this would exceed our scope by far, therefore we want to shift our focus to our post-processing step, namely clustering and averaging candidate poses based on our proposed distance. Hence, we employed a modified version of the methodology proposed by [DUNI10] to obtain the candidate poses. We used the code of [wha24] up to small adaptations. We performed our experiments based on those candidate poses. As we have stated above, there are certain ways to improve the quality of the candidates themselves, but this would exceed the scope of this thesis and we aimed to choose a baseline for our clustering experiments.

We want to mention that a symmetry-aware approach to pose recovery was made by [BDLC18], using the same distance. Instead of hierarchical clustering, they used an adaptation of Mean Shift, a popular technique to find the maxima of a density function, so-called *modes*. The density function is given by the candidate poses, and the modes represent the recovered poses. While the authors claim that by choosing a radius small enough they can guarantee the unambiguous estimation of the average, Mean Shift is to our knowledge not known for guaranteed convergence. The hierarchical approach offers the same theoretical guarantees but provides more explainability and directly uses the proposed distance. Furthermore, we can use a weighted average, favoring stronger candidates, which should account for robustness in noisy scenes. By using weights for pose averaging, we believe that our approach also differs from [BDLC17], although they did not clearly specify how they implemented the PPF method.

Our experiments are employed as follows. First, we contrast the algorithm's outcomes when taking the symmetries of the object into account versus when disregarding them. This will be done in a setting with synthetic data, where the circumstances are almost perfect. Three basic types of symmetries are tested. The recovered poses will be compared with the ground truth, using our proposed distance to have a precise error measure. Furthermore, we compare the outcome with the result when using the scores of the votes as weights during averaging, favoring poses where the algorithm is sure that they are an accurate description.

Second, we want to test our proposed clustering approach. At this point we consider the symmetries of the object correctly, the focus lies solely on recovering the final poses. We aim to compare our results with the adapted version of Mean Shift as proposed by [BDLC18]. Furthermore, we employ DBSCAN [EKS$^+$96], a widely used clustering algorithm that is density-based and robust to outliers. DBSCAN does not rely on a vector space structure and can be used with any metric, so it can be executed with our proposed distance.

This can be effectively tested when the candidates are not that perfect anymore. To achieve this in a setting with synthetic data, we add various forms of random noise to the scene points. This allows us to test the different methods precisely since the ground truth of the poses to recover is known. Then we shift to a real-world example, namely the one with the trailer and the wheel, see Figures 4.1 and 4.2. This should not only test the robustness of our method but also show that it applies to real-world applications.

5.2 Synthetic data without noise

This section should emphasize the effect of taking the symmetries of the object into account versus disregarding them. We made basic experiments on toy data. We test three different types of symmetry: revolution symmetry, finite symmetry, and no symmetry at all. Here we do not aim to test different clustering approaches since they would likely lead to the same result.

5.2.1 The torus

The torus stands for an object with revolution symmetry and roto-reflection invariance. The proper symmetry group is infinitely large, therefore infinitely many rotation matrices describe the same pose. Because of the roto-reflection invariance, we have two representatives in $\mathbb{R}^6$ for each pose. We chose a model containing 576 points and scaled it to have a diameter of 1. Normals were estimated, and we used an exact copy of the model for the scene. A random rigid transformation was applied to the model, which served as the ground truth. Figure 5.1 visualizes the model and the scene as point clouds.

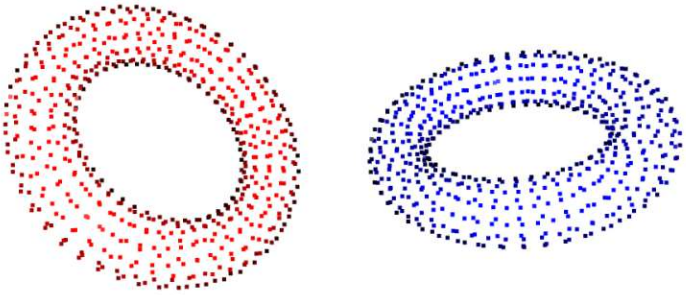


Figure 5.1: Model (red) and scene (blue) of the torus

To create the PPF model as described in Sections 4.3.1 and 4.3.2 we set $n_{angle} = 30$, i.e. we used 30 bins for the angle. The number of reference points was chosen to be 10% of the model points, which led to 57. After calculating the candidate poses, we took the 50 best, according to their score obtained by the Hough voting. In Figure 5.2 the 10 best are shown, each colored in a different shade of blue for visibility reasons. One can observe that the candidate poses already form a pretty good approximation of the ground truth.

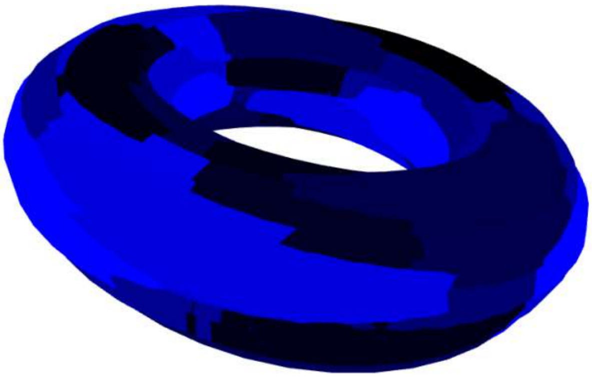


Figure 5.2: The 10 best candidate poses of the torus

We calculated $T \approx 0.42$ and chose $K = 0.1$ for the clustering threshold, satisfying $K < T/4$. In Table 5.1 the clustering result for when symmetries are not considered is shown. The process generated several clusters, the biggest ones containing only five elements. In each of these clusters pose averaging was performed using the theory developed before (Theorem 3.4.2 and Theorem 3.4.9), once using the scores as weights and once unweighted. The error of the retrieved poses was measured according to our proposed distance, with the symmetry taken into account. In general, the overall error is very good compared to the model size, and since the model diameter is 1 those can be interpreted directly as percentages. It can be observed that weighting the average does not make much of a difference, likely due to the good quality of all candidates. However, we obtained a total of 22 clusters which translates to 22 instances of the object, when there is only one, which is a clear disadvantage. Performing the clustering symmetry-aware produces a different result, as seen in Table 5.2. Here we got only one cluster, correctly accounting for one instance of the object in the scene. Not only is the issue with the instances resolved, but the overall error has also significantly improved. It is better than every single cluster before, mostly because more poses were considered for the averaging, making the result more robust and accurate.

Cluster Size	**Avg Score**	**Error weighted**	**Error unweighted**
5	941.8	0.004225	0.005224
5	938.8	0.010028	0.009321
5	913.2	0.006441	0.006323
4	891.8	0.014271	0.014000
4	863.0	0.019834	0.019112
4	858.0	0.007087	0.006418
2	1084.0	0.028415	0.028415
2	1083.0	0.007686	0.007713
2	1012.5	0.022053	0.021455
2	865.0	0.017756	0.017209
2	853.0	0.019278	0.018716
2	772.0	0.020748	0.020817
2	769.0	0.026302	0.026302
1	1084.0	0.021719	0.021719
1	1072.0	0.037673	0.037673
1	945.0	0.026336	0.026336
1	783.0	0.024810	0.024810
1	763.0	0.023975	0.023975
1	763.0	0.017800	0.017800
1	761.0	0.024973	0.024973
1	761.0	0.024449	0.024449
1	759.0	0.030353	0.030353

Table 5.1: Clustering the torus without symmetries

Cluster Size	**Avg Score**	**Error weighted**	**Error unweighted**
50	899.8	**0.001844**	0.001880

Table 5.2: Clustering the torus with symmetries

5.2.2 The 5-sided pyramid

The second experiment was conducted with a 5-sided pyramid. This should represent an object with a finite symmetry class. In this case, the proper symmetry group contains exactly five elements. To avoid problems with normals, we cut off the parts that are not smooth, i.e. the top and the vertices on the side. This left us with a model containing 315 vertices. Moreover, the model was scaled to have a diameter of 1. Again, an exact copy of the model was used for the scene, including the same normals. A random rigid transformation was applied to the model, which served as the ground truth. The model and scene can be seen in Figure 5.3.

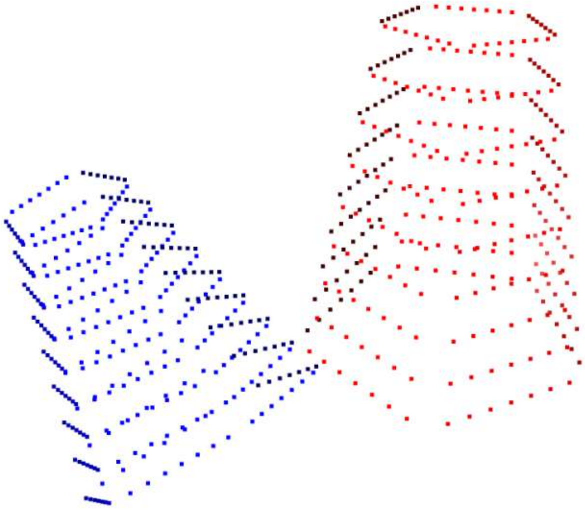


Figure 5.3: Model (red) and scene (blue) of the 5-sided pyramid

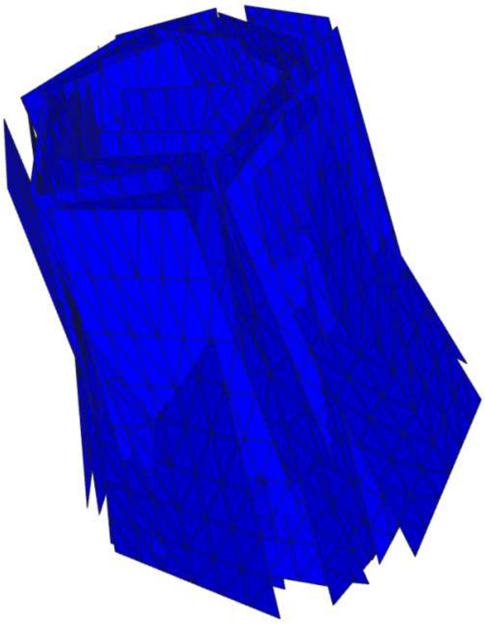


Figure 5.4: 10 best candidate poses of the pyramid

To create the PPF model as described in Sections 4.3.1 and 4.3.2 we set $n_{angle} = 30$, i.e. we used 30 bins for the angle. The number of reference points was chosen to be 20% of the model points, which was 63. After calculating the candidate poses, we took the 50 best, according to their score obtained by the Hough voting. Figure 5.4 shows the 10 best candidates. One can see, that they are not as accurate as for the torus. This is not surprising since the object is very primitive and essentially consists of five planes. Such voting algorithms usually do not perform well on primitives, since rounding errors and edge cases can be problematic. Therefore, for the hierarchical clustering, a bigger threshold of 0.2 was used to offset this effect. Unfortunately, the unambiguous estimation of the mean is not theoretically guaranteed anymore, since $0.2 > T/4 \approx 0.06$. This experiment is also there to show that the result can still make a lot of sense, even though lacking theoretical foundations. In practice, there is often a trade-off between theoretical soundness and applicability, and in that case, a small threshold would not go well with the quality of the candidates.

First, the clustering process was performed assuming the object does not admit any symmetries, which can be seen in Table 5.3. Second, the clustering was done with the same candidate poses, but accounting for the object's symmetry this time, see Table 5.4. Again, each cluster was averaged to retrieve the final poses.

Cluster Size	Avg Score	Error weighted	Error unweighted
20	594.4	0.022869	0.023061
9	624.7	0.022653	0.023853
9	580.9	0.035680	0.035121
8	619.6	0.021147	0.020164
4	558.5	0.031559	0.031378

Table 5.3: Clustering the pyramid without symmetries

Cluster Size	Avg Score	Error weighted	Error unweighted
50	598.6	0.020336	0.020416

Table 5.4: Clustering the pyramid with symmetries

In Table 5.3 we have five poses for the same instance, one for each of the five sides, where the pyramid looks the same. Each of these clusters is again a worse approximation of the ground truth than clustering everything at once with the symmetry-aware distance.

5.2.3 The bunny

The same experiment with identical conditions was performed with the Stanford Bunny [TL94]. We downsampled the mesh to 689 vertices to save computation time. The bunny does not admit any symmetries, therefore we have $T = \infty$. Figure 5.5 shows the candidates, colored by score. We see that they are pretty well distributed. Since the estimation of the mean is unique here, we used the same threshold of $K = 0.2$ as for the pyramid. Here we only took the 20 best candidates, due to the uniqueness of the pose in terms of symmetry.

As the bunny does not admit any symmetries, no comparison is necessary. Yet, Table 5.5 shows that the result is very good, with the total error only being 1% of the model diameter. However, weighing the average according to the voting score did not make any noticeable difference, with both results being very accurate.

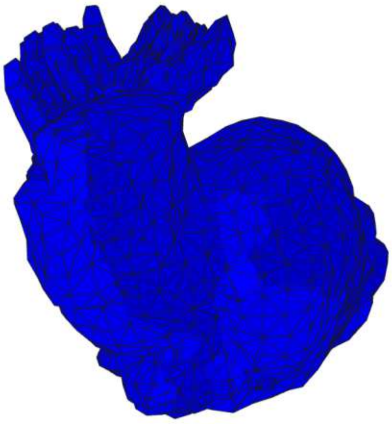


Figure 5.5: Candidate poses of the bunny

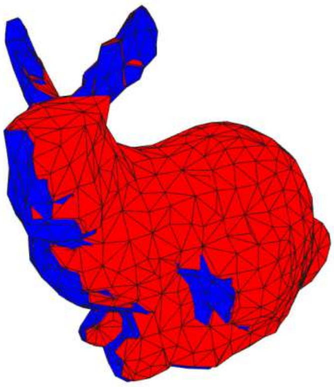


Figure 5.6: Recovered pose (blue)
Ground truth (red)

Cluster Size	Avg Score	Error weighted	Error unweighted
20	1087.8	0.010480	0.010490

Table 5.5: Clustering the bunny with symmetries

5.3 Synthetic data with noise

The previous example stands for a case where the set of candidates is almost perfect. In reality, however, this happens very rarely. The set of candidates will be more distributed and contain poses that do not describe any instance of the object at all. Obtaining the set of candidates can be disturbed by many factors, such as noise, clutter, or the quality of the normals. Sometimes objects are not fully visible. We want to simulate this by adding random Gaussian noise to the objects. One could also delete random points or add some artificial noise, which would lead to the same effect.

The torus from the previous example served again as the model. The scene was constructed by three copies of the torus where one of them was blurred by light noise, and one by heavy noise. With light and heavy we mean Gaussian noise with standard deviation of 0.01 and 0.02 respectively. The scene can be seen in Figure 5.7. After adding the noise, the normals were estimated again.

The Hough voting was performed in the same fashion as before, 5% of the 1728 scene points were used as reference points. Figure 5.8 shows the obtained candidates. Poses were colored according to the score, the darker the lower. The view of the scene is a bit rotated, to be able to recognize more in the image. While the candidates of the torus with no noise

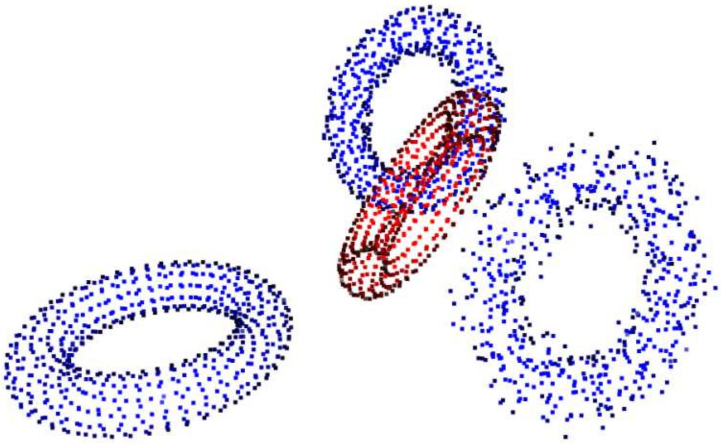


Figure 5.7: Model (red) and scene (blue) of tori with noise

are accurate, there are a few bad ones at the one with heavy noise. There is even one in between two poses which does not make any sense.

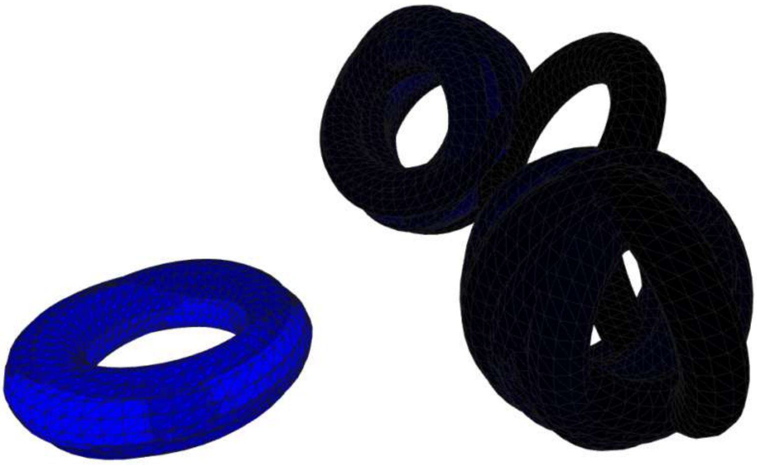


Figure 5.8: Candidate poses of the tori

For the hierarchical clustering process, we used the same threshold as above. Mean Shift was performed with the same radius. For DBSCAN, the epsilon parameter was set to 0.05. At the end of each clustering, small clusters representing outliers were removed. This was implemented by disregarding clusters with sizes smaller than 5% of the maximum cluster size. Table 5.6 shows the results of each process. While DBSCAN and our proposed hierarchical clustering performed similarly, Mean Shift could not deal with too much noise in the data. Several medium-sized clusters appeared as a result, all representing the same instance. Figure 5.9 shows the recovered poses of the hierarchical clustering, which are very accurate. The results show that adding noise hardly affects the overall performance of the pose recovery, underlining the robustness of the method. The total errors are all negligible and on the same scale. Also, weighing does not impact the result noticeably. Additionally, we see that all the approaches correctly filtered out the wrong pose.

Noise Std Dev	Hierachical	DBSCAN	Mean Shift
0	0.009223	0.009223	0.008842
0.01	0.006334	0.006803	0.008701
0.02	0.015097	0.010702	multiple detections

Table 5.6: Clustering the noisy tori

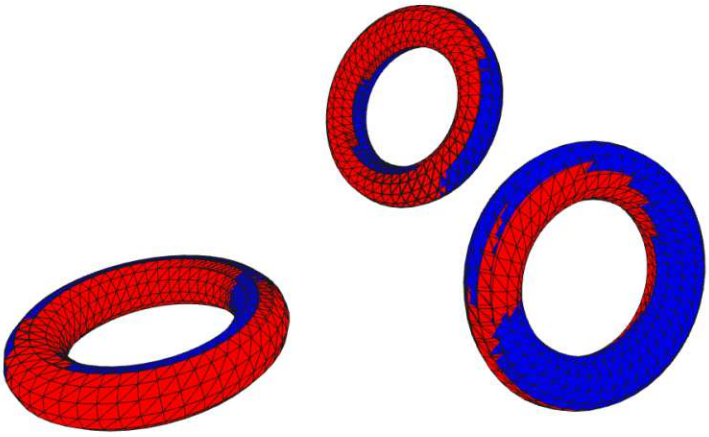


Figure 5.9: Recovered poses (red) of the tori

5.4 Real-world data

Finally, we want to analyze our method on real data. While the robustness can be tested on synthetic data by adding random noise or clutter, real-world data usually shows more difficulties than that. The scene we used was the scan of a trailer, obtained by a LIDAR scan, as seen in Figure 4.1. The task was to obtain the poses of the two wheels. This imposed several difficulties. Firstly, the scan was too big, so we manually had to segment the point cloud. In practice, a segmentation algorithm could be used as a preprocessing step. Even after cropping, the point cloud was too dense. To improve the execution time, points were sampled by voxel size, which resulted in a point cloud containing 5679 points. Normals were estimated before the sampling, to have better quality, and then aligned to point towards the same direction, in our case to the sensor.

The choice of the model was not so easy, as it is not advisable to take the whole wheel since in the scan the wheel is only visible from one side. Our approach was to model the front side of the wheel as accurately as possible in Blender, and then randomly sample points from the surface. The symmetry type of this object is a revolution object without roto-reflection invariance. The more points are chosen for the model, the more likely it is that the right point pair features are found in the scene. But this is in contrast to the computation time, which increases drastically with the model size. Here we chose to sample 1000 points, the resulting model was already shown in Figure 4.2. Another problem that arose was the lack of a ground truth. To circumvent this, we fitted the model by hand in the scene and extracted the ground truth visually. While this is far from perfect, it was sufficient for our purposes as we needed something to compare our results with. The sampled scene with the fitted models of the wheels can be seen in Figure 5.10. One can observe that there is still a lot of noise in the data, and also that the right wheel is a bit better captured than the left one.

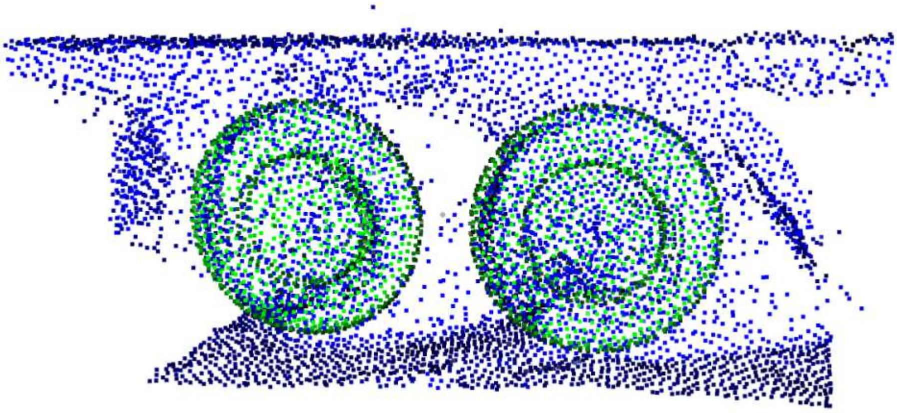


Figure 5.10: Scene (blue) plus fitted wheels (green)

To obtain the candidate poses, we chose the same parameters as in the previous experiments. The number of reference points was 1% of the scene, leading to 56 candidate poses. The retrieved poses, colored according to their score (the lighter, the better), are depicted in Figure 5.11. While many poses are part of the noise, most of them are very dark, meaning they have a low score. One can somehow guess that at the place where the wheels should be there are a few poses where the algorithm is more confident.

For the pose recovery, we used a larger threshold of 0.4 for the hierarchical clustering and as a Mean Shift radius. Since this type of symmetry only admits one representative, there are no problems with any ambiguities while averaging. The results in Table 5.7 show that our proposed hierarchical clustering performs better than DBSCAN and Mean Shift. Plots of the retrieved poses can be seen in Figure 5.12 to give a better impression. While Mean Shift correctly detected both wheels, the error was bigger than the one of our proposed method. This is likely because Mean Shift does not consider the scores of the poses, which is essential in this case to filter out the noise. DBSCAN, on the other hand, performed well in terms of total error but also recovered an additional pose somewhere on the floor in front of the trailer. Again, one disadvantage of DBSCAN is that the clustering process is done independently of the scores, whereas the hierarchical approach centers its clusters

Figure 5.11: Candidate poses of the trailer

around the strongest poses. This can also be seen in the respective cluster sizes.

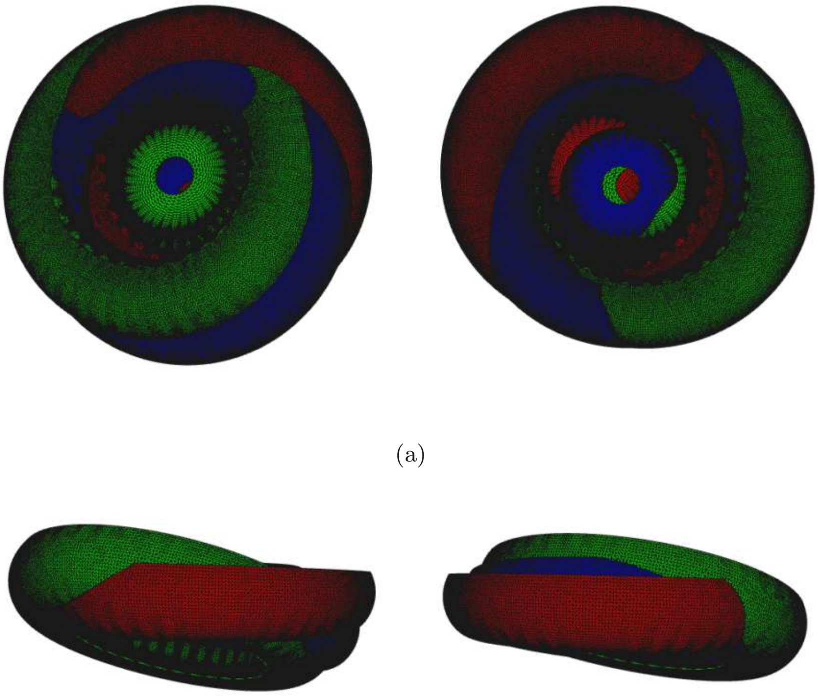
Clustering	**Hierachical**		**DBSCAN**		**Mean Shift**	
	Size	Error	Size	Error	Size	Error
right	27	**0.04189**	20	0.06025	19	0.06108
left	22	**0.06183**	19	0.05374	25	0.08407
noise	-	-	17	0.38816	-	-

Table 5.7: Clustering of the wheel

5.5 Discussion

Our experiments show that it is crucial to consider the object's symmetry for robust pose estimation. Based on a fixed set of candidate poses, we deduced that not only the correct number of poses are retrieved and duplicates of the same instance are avoided, but also that the overall accuracy and robustness increase drastically. The method works for objects admitting symmetries but is not specifically designed for such, as we have proven with the Stanford Bunny.

Moreover, our proposed hierarchical clustering appears to be more robust and accurate than common methods such as Mean Shift or DBSCAN. This is likely due to poses with a high score being favored in the clustering process, accounting for bigger clusters and therefore for more robustness. While the results on synthetic data with artificial noise were



(a)

(b)

Figure 5.12: Recovered poses of the proposed clustering (blue) and Mean Shift (green) together with the ground truth (red); (a) front view; (b) top view

similar to standard methods, it outperformed given methods in the case of real-world data, proving to be more robust to noise within the candidates. While the retrieved poses are already a very good estimation of the ground truth, those could be further refined with common methods like Iterative Closest Point (ICP) [CM92] that require a good estimation as a starting point.

Bibliography

[Arm97] Mark A Armstrong. *Groups and symmetry*. Springer Science & Business Media, 1997.

[BDLC17] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L Crowley. Symmetry aware evaluation of 3D object detection and pose estimation in scenes of many parts in bulk. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2209–2218, 2017.

[BDLC18] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L Crowley. Defining the pose of any 3D rigid object and an associated distance. *International Journal of Computer Vision*, 126(6):571–596, 2018.

[BI15] Tolga Birdal and Slobodan Ilic. Point pair features based object detection and pose estimation revisited. In *2015 International conference on 3D vision*, pages 527–535. IEEE, 2015.

[BK02] Calin Belta and Vijay Kumar. Euclidean metrics for motion generation on SE(3). *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 216(1):47–60, 2002.

[BM12] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. In *Computer graphics forum*, volume 31, pages 1765–1774. Wiley Online Library, 2012.

[Ced04] Judith Cederberg. *A course in modern geometries*. Springer Science & Business Media, 2004.

[CF01] Richard J Campbell and Patrick J Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[CJ97] Chin Seng Chua and Ray Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25:63–85, 1997.

[CM92] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[Com18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[DCPR18] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6Dpose: Recovering 6D object pose from a single RGB image. *arXiv preprint arXiv:1802.10367*, 2018.

[dFMB15] Rui Pimentel de Figueiredo, Plinio Moreno, and Alexandre Bernardino. Efficient pose estimation of rotationally symmetric objects. *Neurocomputing*, 150:126–135, 2015.

[DI15] Bertram Drost and Slobodan Ilic. Local Hough transform for 3D primitive detection. In *2015 International Conference on 3D Vision*, pages 398–406. IEEE, 2015.

[DUNI10] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[Gal11] Jean Gallier. *Geometric methods and applications: for computer science and engineering*, volume 38. Springer Science & Business Media, 2011.

[Gra01] Claus Gramkow. On averaging rotations. *Journal of Mathematical Imaging and Vision*, 15(1):7–16, 2001.

[GS07] Ana Irene Ramirez Galarza and José Seade. *Introduction to classical geometries.* Springer Science & Business Media, 2007.

[HKLM22] Lukáš Hruda, Ivana Kolingerová, Miroslav Lávička, and Martin Maňák. Rotational symmetry detection in 3D using reflectional symmetry candidates and quaternion-based rotation parameterization. *Computer Aided Geometric Design*, 98:102138, 2022.

[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.

[HLRK16] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going further with point pair features. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 834–848. Springer, 2016.

[Huy09] Du Q Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35:155–164, 2009.

[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.

[KRA$^+$14] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[Par95] Frank C Park. Distance metrics on the rigid-body motions with applications to mechanism design. 1995.

[PGBP10] In Kyu Park, Marcel Germann, Michael D Breitenstein, and Hanspeter Pfister. Fast and automatic object pose estimation for range images on the GPU. *Machine Vision and Applications*, 21:749–766, 2010.

[PRIL19] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, and Vincent Lepetit. On object symmetries and 6D pose estimation from images. In *2019 International conference on 3D vision (3DV)*, pages 614–622. IEEE, 2019.

[RBB09] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[RL18] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth, 2018.

[RVDH05] Tahir Rabbani and Frank Van Den Heuvel. Efficient Hough transform for automatic detection of cylinders in point clouds. *Isprs Wg Iii/3, Iii/4*, 3:60–65, 2005.

[Sti08] John Stillwell. *Naive Lie theory*. Springer Science & Business Media, 2008.

[SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[SWR10] Inna Sharf, Alon Wolf, and Miles B Rubin. Arithmetic and geometric solutions for average rigid-body rotation. *Mechanism and Machine Theory*, 45(9):1239–1251, 2010.

[TL94] Greg Turk and Marc Levoy. Polygonal approximation to implicit surfaces. *Proceedings of the SIGGRAPH '94 Conference on Computer Graphics and Interactive Techniques*, pages 303–308, 1994.

[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.

[TTKK14] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class Hough forests for 3D object detection and pose estimation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 462–477. Springer, 2014.

[Ume91] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.

[VLM18] Joel Vidal, Chyi-Yeu Lin, and Robert Martí. 6d pose estimation using an improved method based on point pair features. In *2018 4th international conference on control, automation and robotics (iccar)*, pages 405–409. IEEE, 2018.

[Wey15] Hermann Weyl. *Symmetry*, volume 104. Princeton University Press, 2015.

[wha24] whateverforever. model-globally-match-locally-python. https://github.com/whateverforever/model-globally-match-locally-python, 2024. GitHub repository.

[WYL20] Guokang Wang, Lei Yang, and Yanhong Liu. An improved 6d pose estimation method based on point pair feature. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 455–460, 2020.

[XW05] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.