Doctoral Dissertation (Shinshu University)

Subset focused rigid transformation search for registration of low overlapping 3D point clouds

Luis Antonio Peralta Miranda

Supervisor: Prof. Kiyoshi Tanaka

Interdisciplinary Graduate School of Medicine, Science and Technology
Department of Science and Technology, Energy Systems Unit
Shinshu University

Doctor of Engineering

September 2022

Acknowledgement

First and foremost, I would like to express my gratitude to all the people that have supported me in my journey to achieve a doctorate: to my parents and siblings for encouraging me to pursue higher education, enjoy the process of learning, and making me realize the significance of my accomplishments; to my friends in Mexico that despite the distance never forgot about our good times; to the new friends I made, Hassan Arafat, Yuri Marca, Hugo Monzon, Jaime Sandoval, Elena Balderas, Lucia Sandoval, and Jessica Piora, for all the adventures, good times, and put up with me my temper; to professor Kiyoshi Tanaka for allowing me to come to Japan to pursue my graduate studies and believing in me to become a high level professional; and lastly, to Hikari Duncan, for joining me in the adventure of my life, cheering me up in my difficult moments, putting up with my meaningless complaints, and unconditionally loving me. No words can express my appreciation for all of you, but I am forever grateful for you being part of my life.

This work was partially supported by JST SPRING, Grant Number JPMJSP2144.

Abstract

The advancement in computer hardware and the employment of 3D sensors has conveyed a wide range of applications and opportunities for development in image processing, resulting in new branches of research such as 3D image processing. One of the common types of data obtained from 3D sensors is the 3D point cloud, a collection of points in 3D space representing an object's or environment's shape. They are widely utilized in developing perception systems for self-driving cars, Autonomous Intelligent Vehicles, and Unmanned Aerial Vehicles. Also, in creating content for Virtual Reality experiences and Digital Twins.

The registration of 3D point clouds is a fundamental task in 3D image processing that looks for the transformation that optimally aligns the 3D point clouds to form a more extensive and detailed representation, being a core process for applications that need digital reconstruction. For example, autonomous intelligent vehicles perform System Localization and Mapping (SLAM), which estimates the vehicle's location and orientation while building a map of the environment comprised of hundreds or thousands of registered 3D point clouds obtained from a 3D sensor. Although there is a broad range of registration methods to date, only a few can successfully determine a good transformation that aligns 3D point clouds with partial overlaps below 40

The importance of developing registration methods for 3D point clouds with low overlap lies in the possibility of performing the full digital reconstruction of objects and environments with fewer data. Particularly useful in developing perception systems for truly robust Autonomous Intelligent Vehicles, which must perform SLAM without interruption in the presence of potentially low overlap 3D point clouds. Alternatively, it is desired when the storage resources or time to acquire the data are limited and compel us to obtain a few 3D point clouds from the most significant angles of an object or scene - having fewer 3D point clouds to work with to assemble an entire reconstruction.

The main reason why most typical registration approaches fail in a low overlap scenario is due to a high rate of incorrect data association (i.e., defining what points have to be aligned). The most common method to perform this data association is through a Nearest-neighbor search for each point in the point cloud to be aligned. However, it is an inadequate method when the 3D point clouds are initially far from the optimal alignment, or there are a lot of

potential outliers coming from noise or not overlapping areas. Other registration approaches substitute the nearest-neighbor search for pipelines making use of key points and feature point descriptors matched on the descriptive space and filtered by random sampling iterative methods. Nevertheless, these also fail in the low overlap conditions because the true overlap of low overlapping 3D point clouds lies between boundary surfaces and surfaces within the boundaries, which have very different point dispersion that governs the descriptiveness of the features, making them a challenge to associate correctly.

On the one hand, in recent years, Artificial Neural Networks have also been applied to perform the registration of 3D point clouds, bringing outstanding advancements to 3D image processing for digital reconstruction. Among them, some even focus on the low overlap problem. However, as with most Artificial Neural Network methods, these require extensive and costly datasets to train and high-end hardware to run due to the high computational complexity of convolving operations in 3D. On the other hand, the Hough transform is a well-known, conceptually simple, easy to implement, and flexible technique for feature extraction in image processing not fully explored in the 3D counterpart that does not require any training or high-end hardware to run. Furthermore, in 3D point cloud registration, these mechanisms allow us not to make direct hard point correspondences and drastically reduce the number of evaluation units to find the desired solution transformation.

This thesis presents the formulation details, implementations, experimental evaluations, and results of a couple of registration methods focused on aligning point subsets from the overlapping areas of 3D point clouds. The core idea of both methods relies on the likelihood of finding close to optimal transformations when aligning point subsets derived from the proper overlapping areas, regardless of the point dispersion. This work is divided into two parts to make it clear to the reader. The first part introduces the first developed method proposing the Supervoxel Segmentation technique to divide the 3D point clouds into subsets, registering them and evaluating the corresponding transformation via an inlier-focused error maximization objective function to find the best solution. The second part refers to the second developed method, which uses the same segmentation technique but improves the computational complexity by reducing the number of subsets to align. It implements a voting mechanism to determine the solution and replaces the evaluation of the transformations for the minimization of a metric that guarantees the maximum overlap and minimum error between corresponding points. The second part also covers a study on the effect of replacing the conventional quaternion rotation representation by the Euler angles over the second method. Furthermore, it extends to ablation and comparative studies that showcase how the method performs under its ruling parameters and compares it against the first method and other publicly available state-of-the-art methods.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

With the exponential development and increasing availability of relatively more powerful computing tools, there has also been an exponential interest in developing algorithms to process n-dimensional (nD) information to make technologies that, decades ago, belonged to science fiction. Image processing has a wide range of achieved milestones when processing two-dimensional (2D) information. Such as helping us read a text in different languages on the go, calibrating our smartphones' compass via image recognition, and even detecting malignant signs of tumors in a matter of seconds, to name a few. Nonetheless, the development of three-dimensional (3D) image processing has not yet achieved the same maturity due to the higher complexity imposed by the third dimension, which requires more computational power. Hence, it still is a relatively new field in constant development that takes advantage of the newest computing tools every year.

Digital reconstruction is one of the main development interests in 3D image processing since it is well employed in many fields and industries, from media production to medical imaging, archaeology, and robotics. To carry it out, first, obtaining data through 3D sensors is necessary, whose popularity and availability have increased in the last decade. Light Detection and Ranging (LiDAR) sensors, also known as laser scanners, and Time of Flight (ToF) sensors are the most commonly employed 3D ones that can digitize real-world environments and objects with relative ease (see **Figure 1.1**). As shown in **Figure 1.2**, nowadays, LiDAR and ToF sensors are common components of perception systems of Autonomous Intelligent Vehicles (AIVs), Unmanned Aerial Vehicles (UAVs), and other robotic systems. The obtained images from these sensors are digitized representations known as *3D point clouds*, a collection of 3D points $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N_\mathcal{P}}$ that hold the shape and geometric characteristics of the original object or environment.

(a) Velodyne® LiDAR **(b)** Realsense® LiDAR **(c)** Helios® ToF sensor

Figure 1.1: Common 3D sensors.

(a) ZOOX® self-driving car **(b)** AIV **(c)** UAV

Figure 1.2: Systems that utilize 3D sensors.

1.1 3D point cloud registration

As shown in **Figure 1.3**, due to limitations of the field of view and occlusion effects, 3D point clouds directly obtained from a sensor are partial representations rather than entirely digitized objects or environments. In order to retrieve the original shape, the partial representations are aligned and combined through a process known as registration. Informally speaking, registration consists of looking for the rigid transformation that aligns a source point cloud onto a target point cloud, preserving the original shape but extending the visual representation of the source with the target, forming a more extended and detailed 3D point cloud.

When the process is carried out only between a source and a target 3D point cloud, it is known as pairwise registration; meanwhile, if it involves three or more, it is known as multi-view registration. The latter is what actually can retrieve the full original shape, as shown in **Figure 1.4**. However, it is an extension of the pairwise process that aligns consecutive pairs of 3D point clouds in sequence, often improved through pose graph optimization in SLAM applications [1]. Therefore, many researchers and developers emphasize developing robust pairwise registration that inherits less error to the multi-view stage, reducing the need for post-optimization to improve the reconstruction.

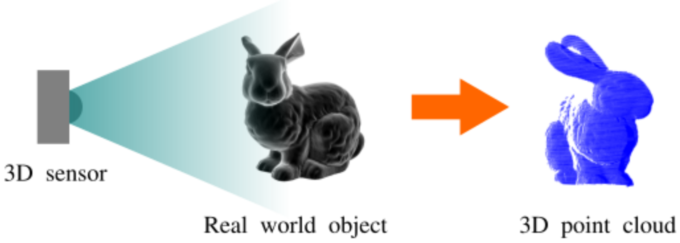


Figure 1.3: 3D point cloud acquisition by a ToF sensor.

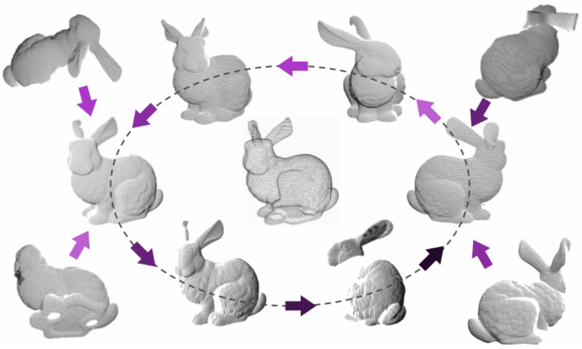


Figure 1.4: Full digital reconstruction of an object.

1.2 Pairwise registration

Formally, we can define the pairwise registration of 3D point clouds in the following manner: having two point clouds, the *source* $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^{N_\mathcal{P}}$ and the *target* $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^{N_\mathcal{Q}}$, along with a set of *point correspondences* $\mathcal{C} = \{(i,j)\}$, we look for the rigid body transformation matrix $\mathbf{T}(\mathbf{R},\mathbf{t})$ that minimizes the sum of squared errors:

$$\arg\min \sum_{(i,j)\in\mathcal{C}} \left\|\mathbf{q}_j - \mathbf{R}\mathbf{p}_i - \mathbf{t}\right\|^2. \tag{1.1}$$

Reordering $\mathcal{P}$ and $\mathcal{Q}$ and considering the set of point correspondences $\mathcal{C}$ with the index n, we can define the point sets $\{\mathbf{p}_n\}\{\mathbf{q}_n\}$ and reduce the problem to estimate the rigid body transformation

$$\bar{\mathbf{p}}_n = \mathbf{R}\mathbf{p}_n + \mathbf{t} \qquad n = 1,\ \ldots,\ |\mathcal{C}| := N, \tag{1.2}$$

which describes the transformation of points $\{\mathbf{p}_n\}$ into $\{\bar{\mathbf{p}}_n\}$. Hence, getting as close as possible to $\{\mathbf{q}_n\}$ minimizing the sum of squared *point-to-point* distances

$$\arg\min \sum \|\mathbf{q}_n - \bar{\mathbf{p}}_n\|^2. \tag{1.3}$$

Remark 1.1: Special case of the Absolute Orientation Problem

Registration is a special case of the Absolute Orientation Problem

$$\bar{\mathbf{p}}_n = \lambda_s \mathbf{R}\mathbf{p}_n + \mathbf{t}, \tag{1.4}$$

where we only want the parameters of the rigid body transform $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$. Hence, the scale parameter is $\lambda_s = 1$.

1.2.1 Direct optimal solution: known data association

Having perfect data association (*i.e.*, we know the perfect set of point correspondences $\mathcal{C}$), it is possible to directly find an optimal solution by computing a shift involving the center of masses of both 3D point clouds and performing a rotational alignment using Singular Value Decomposition (SVD). Formally speaking: considering a set of optional weights $W = \{w_n\}_{n=1}^{N}$ so that the squared error minimizes as

$$\arg\min \sum \|\mathbf{q}_n - \bar{\mathbf{p}}_n\|^2 w_n, \tag{1.5}$$

the center of masses of both 3D point clouds (*i.e.*, the means of the point sets) are

$$\mathbf{p}_0 = \frac{\sum \mathbf{p}_n w_n}{\sum w_n} \qquad \mathbf{q}_0 = \frac{\sum \mathbf{q}_n w_n}{\sum w_n}. \tag{1.6}$$

Hence, the cross-covariance matrix is

$$\mathsf{H} = \sum (\mathbf{p}_n - \mathbf{p}_0)(\mathbf{q}_n - \mathbf{q}_0)^{\mathsf{T}} w_n. \tag{1.7}$$

From which it is possible to compute $SVD(\mathsf{H}) = \mathsf{UDV}^{\mathsf{T}}$, directly computing the rotation matrix and translation vector as

$$\mathbf{R} = \mathsf{UV}^{\mathsf{T}} \qquad \mathbf{t} = \mathbf{q}_0 - \mathbf{R}\mathbf{p}_0. \tag{1.8}$$

This approach solves the registration problem directly and optimally. Nevertheless, perfect data association is nearly impossible in real-world scenarios as it would mean that both 3D point clouds overlap entirely and represent the same surfaces, being unpractical for digital reconstruction. In reality, we want the 3D point clouds to represent different parts of the object or environment and partially overlap to complement each other to form the

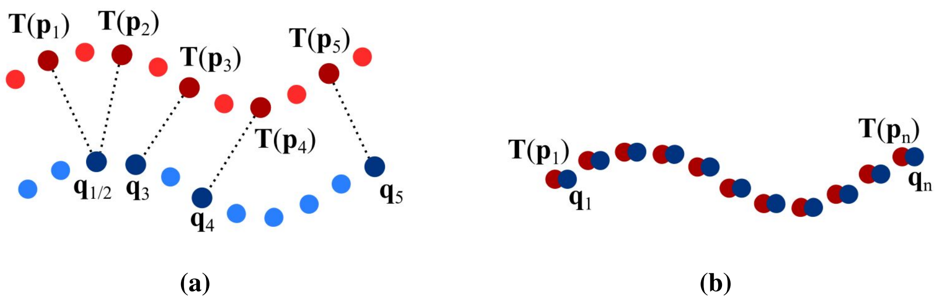


Figure 1.5: Initial guess (a) and converge (b) of ICP.

complete reconstruction. Hence, this method solves the problem based on the stringent and ideal constraints of having a perfect set of point correspondences $\mathcal{C}$.

1.2.2 No direct optimal solution: unknown data association

In practice, we face unknown data associations for the registration process. It means that we do not have prior knowledge of a perfect set of point correspondences $\mathcal{C}$, and no direct optimal solution exists. The Iterative Closest Points (ICP) algorithm [2] is the most widely used technique for registering 3D point clouds which tries to estimate the data association and corresponding transform. However, since the correct data association is unknown and, unlike in the previous case, it is impossible to determine the solution in one single step, ICP makes iterative guesses of the correspondences and the parameters of the transformation, gradually reducing the sum of squared errors between point correspondences. **Figure 1.5** depicts how from an initial guess of $\mathbf{T}$ with guessed point correspondences, depicted in dark red and dark blue, ICP converges to the alignment of the entire point sets with different and better correspondences.

In detail, from an initial guess, ICP at each iteration, first estimates $\mathcal{C}$ by the point-to-point nearest-neighbor function, and then updates $\mathbf{T}$ based on the least-squares estimation method by Umeyama [3]. These two steps are performed until one of three possible termination criteria are met:

1. The number of iterations has reached a maximum of imposed iterations.
2. The difference between the previous and current transformation is smaller than an imposed threshold known as the *Transformation Epsilon* ε_T.

3. The sum of squared errors is smaller than a defined threshold known as the *Euclidean Fitness Epsilon* ε_f.

Remark 1.2: Estimation of the overlapping ratio

The overlapping ratio is the percentage of overlapping points, defined as

$$\xi = (2N/(N_{\mathcal{P}} + N_{\mathcal{Q}})) \cdot 100 \qquad \text{or} \qquad \xi = (N/N_{\mathcal{P}}) \cdot 100. \tag{1.9}$$

The definition on the left, considers both point clouds as $\mathcal{C}$ is formed by points from both of them. The second definition only considers the source and is more commonly used as the set $\mathcal{C}$ is built from using the source points as the base. However, both estimations are very similar in practice.

1.2.3 Conventional registration algorithms

ICP is the most well-studied registration algorithm with the most variations [4–8]. Its variants improve speed, tolerance to noise and outliers, basin of convergence, and stability. *Point-to-plane registration* (N-ICP) [9], *Trimmed ICP* (TrICP) [10], and *Levenberg-Marquardt ICP* (LM-ICP) [11] are the variants that present direct improvements on the error metric, data association, and optimization techniques which, along with ICP, are considered the *Conventional Registration Algorithms* (CRAs). N-ICP performs similarly to ICP, but it requires the point normal vectors of $\mathcal{Q}$ to estimate spanning planes and define nearest-neighbor based on the *point-to-plane distance*. Modifying the objective function as

$$\arg\min \sum \left\| (\mathbf{q}_n - \bar{\mathbf{p}}_n) \cdot \mathbf{n}_q \right\|^2 . \tag{1.10}$$

TrICP improves tolerance to noise and speed of by consistently using the Least Trimmed Squares (LTS) approach at each step of ICP. Here, LTS arranges $\mathcal{C}$ in ascending order and selects the number of points N_{po} that can be paired to estimate their sum LTS. LM-ICP is an improved ICP robust to the initial positions of the 3D point clouds, which estimates the correspondences the same way as ICP, but employs the Levenberg-Marquardt algorithm to minimize the sum of squared errors. **Table 1.1** shows a comparison between the four CRAs.

The core operation of CRAs is an iterative local optimization that makes them prone to get stuck in local solutions that may or may not correctly align the 3D point clouds. Because of it, CRAs cannot correctly align 3D point clouds whose initial positions are far from the global solution, or partial overlap is low. In order to perform any registration of 3D

Table 1.1: Comparison between conventional registration algorithms

	ICP	**N-ICP**	**TrICP**	**LM-ICP**
Error metric	Point-to-point distance	Point-to-plane distance	Point-to-point distance	Point-to-point distance
Minimization optimization	Umeyama least-squares algorithm	Umeyama least-squares algorithm	Umeyama least-squares algorithm	Levenberg-Marquardt algorithm
Ideal conditions	Perfect set $\mathcal{C}$ or $N_{\mathcal{P}} = N_{\mathcal{Q}}$	Perfect set $\mathcal{C}$ or $N_{\mathcal{P}} = N_{\mathcal{Q}}$	Known ξ since $N_{po} = \xi N_{\mathcal{P}}$	Perfect set $\mathcal{C}$ or $N_{\mathcal{P}} = N_{\mathcal{Q}}$

point clouds, these must overlap so that they can be aligned based on the geometry of the overlapping surfaces. The proportion of points in these surfaces is known as the *overlapping ratio*, and they also represent the true point correspondences that lead to the globally optimal alignment. However, it has been shown that as the overlapping ratio decreases, the likelihood of obtaining a good alignment with a CRA decreases due to poor definition of $\mathcal{C}$ in the presence of many outliers that belong to not-overlapping areas [8, 12].

1.2.4 Non-conventional registration algorithms

To estimate better point correspondences other non-conventional registration algorithms [13, 14] follow a pipeline that consists of:

1. Estimating key points.

2. Computing the feature descriptors corresponding to the key points.

3. Making correspondences by matching points in the feature space.

4. Rejecting wrong correspondences based on random sampling iterative methods such as RANSAC [15].

5. Estimating a solution using the SVD method, assuming that the resulting correspondences are almost perfect.

This kind of approaches generally work well with partial overlaps, but can still yield wrong correspondences and incorrect alignments with low overlapping ratio 3D point clouds. Principally, due to shape-dependent parameter tuning, the use of normals-dependent information from their key points/descriptor estimators, and the difference of point densities between the overlapping areas which affects the point normals and, consequently, the descriptiveness

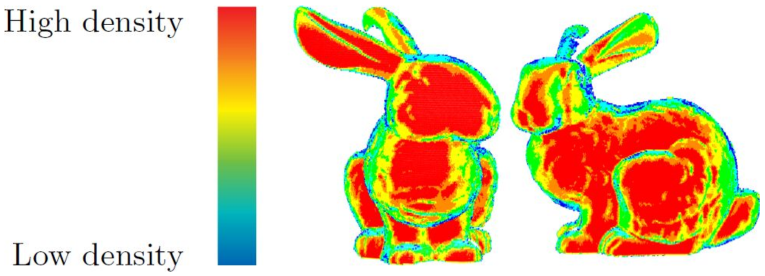


Figure 1.6: Differences of point densities in low overlapping 3D point clouds.

of the point descriptors. Furthermore, as shown in **Figure 1.6**, in low overlapping conditions the point densities of the overlapping areas mismatch from one point cloud to the other, which also leads to false positives when making hard point-to-point correspondences with feature descriptors.

With the increasing popularity of artificial Neural Networks and Deep Learning, along with the availability of more powerful GPUs, and more extensive datasets [16–18], robust point descriptors [19, 20] and 3D point clouds registration methods [21–24] based on these techniques have been developed in the late years. However, most of these are trained on relatively small datasets that may or may not have an specific domain of our interest, and even if we use a pre-trained model and adopt transfer learning we can face limitations such as negative transfer. It means that the initial and target problems are not similar enough, making the pre-training irrelevant. Furthermore, whenever transfer learning is used, the distributions of the pre-training and test data have to be similar or not vary too much, and the number of training data should not over-fit the model [25, 26]. Therefore, we could assume that pre-trained Deep Learning models for 3D point clouds are a viable option when there is sufficient training data of a new type of dataset that is close to a pre-trained domain. Additionally, due to the third dimension, Deep Learning methods for 3D point clouds are considerably resource heavy and demand vast amounts of Video Random Access Memory (VRAM) from a Graphics Processing Unit (GPU) for training.

Remark 1.3: Low overlapping ratio

The overlapping ratio can vary from full overlap (100%) to no overlap at all (0%) between the 3D point clouds. To define what percentages are high, medium, or low overlap there is no actual set rule. However, most literature defines low overlapping ratio from 10% to about 40%.

1.3 Motivation

Currently, there is a wide range of registration methods for 3D point clouds, but not many consider the low overlap problem: an scenario that imposes the challenge of finding a local alignment for a small quantity of data points equivalent to the global alignment for all the points in the datasets, and allows us to complete the digital 3D reconstruction of objects or environments with less 3D point clouds, as shown in **Figure 1.8**.

When performing digital 3D reconstruction it is a common practice to register many 3D point clouds with high overlapping ratios covering all possible angles to preserve as much details as possible from the original shape (*cf.* Figure 1.4). Hence, if it is possible to simply do this to complete the reconstruction, why bother on using less datasets with low overlap? Mainly because there are situations where registering less 3D point clouds with low overlap may be ideal:

- Acquiring 3D point cloud data is still costly nowadays, imposing limitations on the time given to acquire the data. Hence, we may have to face situations where we do not have enough time to obtain many 3D point clouds to cover every single angle of an object or environment – having to work with less 3D point clouds to achieve the full reconstruction.

- Considering that unprocessed 3D point clouds obtained from high definition 3D sensors can be in the order of Megabytes or even Gigabytes per seconds of scanning, we may also face data storage limitations where also having 3D point clouds covering most of the surfaces and relatively low overlap between them is the best option.

- Low overlap registration is also desired to develop robust perception systems for AIVs and UAVs performing Simultaneous Localization and Mapping (SLAM) [27]. Equipping them with the ability to recover from sudden external interruptions that can potentially produce low overlapping 3D point clouds of the environment and lost the localization of the system, as shown in **Figure 1.7**.

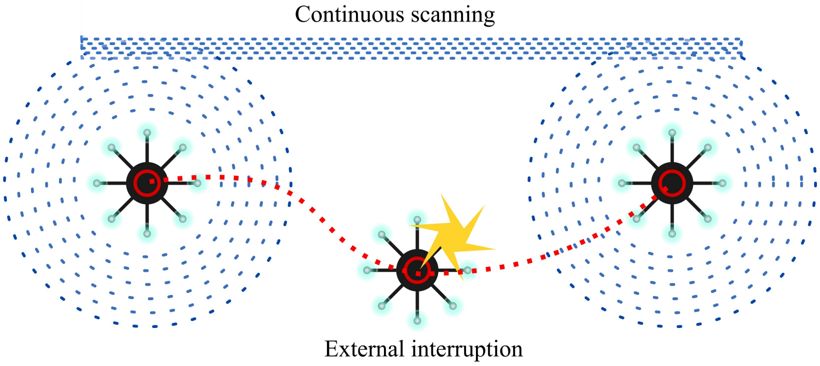


Figure 1.7: Schematic of an UAV continuously scanning after an external interruption.

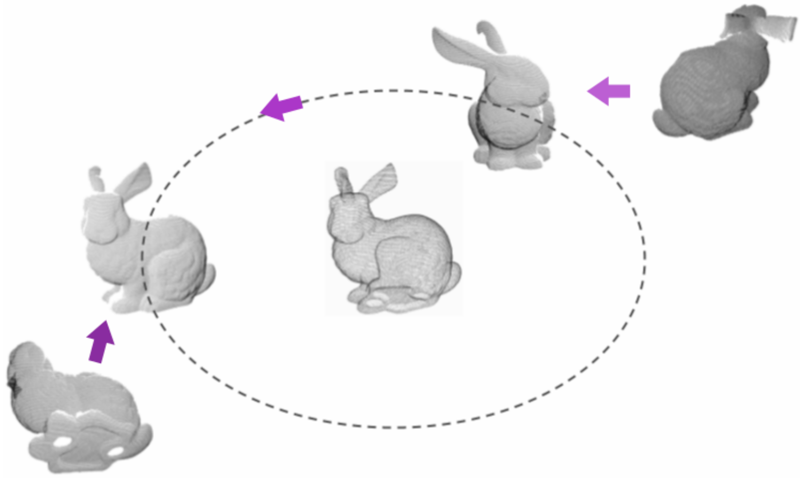


Figure 1.8: Full digital reconstruction of an object using low overlapping 3D point clouds.

It is worth mentioning that the registration of low overlapping 3D point clouds could resemble another interesting problem known as fragment repair [28, 29], which consists of digitally reconstructing an object from individual pieces. However, to the best of our knowledge, fragment repair works on volumetric objects. Thus, the formulation of this problem and solutions are out of the scope of this work.

1.4 Contributions

During the last years we have worked on the employment of subset-search-based methods for the registration of 3D point clouds, focusing on the low overlap problem. Unlike CRAs and feature-based registration approaches, a subset-focused approach gives us the possibility to not make hard point-to-point correspondences and leverage the neighboring interaction of the points at a local scale driven by the shape of the 3D point clouds. Providing the ability of finding close-to-optimal solutions derived from the alignment of the true overlapping areas. The present work contains the theoretical foundations, formulations, implementation details, experimental setups and results that form part of the following contributions:

- A registration method that uses a multi-scale and inlier-only error maximization search, leveraging the assumption of perfect data association between individual subsets from both 3D point clouds.
- A descriptor-free pairwise registration method that uses a voting mechanism to define the solution transform derived from the alignment of subsets of the overlapping areas.
- An extensive ablation and a comparative study of the final version of the proposed descriptor-free and voting-based registration method leading to the statement of a usage guideline.
- A study on the effect of the Euler angles to align low overlapping 3D point clouds and the significance of their order under the framework a voting-based registration approach.
- An evaluation of the proposed registration methods on datasets that gradually reduce their overlap to define the overlap working limits.

1.5 Outline

After this introduction, **Chapter 2** comprehends the problem reformulation that states the core idea behind the focus on aligning subsets in the low overlap problem, and the first registration method developed on it. It rigorously describes the approach itself, and the executed experimental evaluation. **Chapter 3** first states the applied countermeasures for the most obvious issues of the first method. Then, describes a voting-based one with a preliminary experimental evaluation. On **Chapter 4** there is an extensive evaluation of the second method, where ablation studies and comparatives were carried on to fully comprehend the capabilities and limitations of this approach. **Chapter 5** is dedicated to a parallel iteration of the second approach, where the rotation representation in the voting space is replaced for the Euler angles. Corresponding implementation details and experiments are also described in this chapter. **Chapter 6** complements this work with an evaluation of the overlapping limits of the proposed registration methods. Lastly, **Chapter 7** presents the definitive conclusions and open problems of this work that can be used as the base for new elemental and application research.

Chapter 2

Pairwise registration using supervoxel segmentation

This chapter focuses on the registration approach introduced in "Pairwise Registration of Low Overlapping 3D Point Clouds Using Supervoxel Segmentation" and published in IIEEJ Transactions on Image Electronics and Visual Computing, Vol. 7, No. 2.

2.1 Background and related work

2.1.1 Problem reformulation

As stated in Chapter 1, in the low overlap problem the definition of $\mathcal{C}$ is key to succeed. In one hand, the common nearest-neighbor search is not optimal because true correspondences originally are distant from the optimal alignment and outliers from the not overlapping areas can easily "fool" the search. On the other hand, descriptor-based search are potentially affected by the mismatch of point densities between the true overlapping areas. We also know that CRAs easily succeed if $N_{\mathcal{P}} = N_{\mathcal{Q}} = N$ as hypothetically every point $\mathbf{p}_i$ has a unique correspondence $\mathbf{q}_j$ (*i.e.*, $\{\mathbf{p}_i\}\{\mathbf{q}_j\} = \{\mathbf{p}_n\}\{\mathbf{q}_n\}$). Thus, *how can we achieve this ideal condition without using any point density-dependent feature descriptor?*

Clearly, the only way is to isolate the overlapping areas from the 3D point clouds and register these. We could focus the attention to specific areas assuming that these are always the areas of interest in low overlapping ratio 3D point clouds, as proposed by Wu et al.[30] that the overlap is concentrated at the borders. But, this assumption is not always true and does not clearly delimit the beginning and end of the borders. Nevertheless, we can agree that a pair of corresponding subsets from the overlapping areas, regardless if they lay on the borders or not, when registered produce a transformation that optimally aligns the full overlap and point clouds. Thus, if $\mathcal{P}$ and $\mathcal{Q}$ are divided into subsets it is possible to retrieve the solution transformation from those subsets that perfectly align. Now, the emerging question is: *what defines a perfect alignment?*

A perfect alignment is when having the least sum of squared errors between the true correspondences. These can be set following the definition of Ref. [12]: C_{pq} is built by aligning the 3D point clouds with the solution $\mathbf{T}$ and doing a radius-search with radius ε of two times the *Cloud Resolution* (CR) in $\mathcal{Q}$ for each $\mathbf{p}_i \in \mathcal{P}$. That is, for each point in the source which nearest neighbor in the target lays within ε, there is a correspondence and the total of correspondences considering both 3D point clouds is $2N$.

Remark 2.1: Cloud Resolution

The cloud resolution is the mean Euclidean distance between all the points and their nearest neighbors in a 3D point cloud.

$$CR = \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} \|\mathbf{p}_i - \mathbf{p}_{i_{nn}}\|^2 \tag{2.1}$$

It is widely used when the metric or scanning precision of the sensor is unknown.

Considering ε as the *inlier threshold* that defines the correspondences, it is possible to set an objective function that estimates the sum of squared errors only between the correct overlapping points:

$$E\left(\mathbf{T} \mid \mathcal{C},\, \varepsilon\right) = \sum_{i=1}^{N} \left(\|\mathbf{q}_n - \bar{\mathbf{p}}_n\|^2 \leq \varepsilon \right). \tag{2.2}$$

In this sense, by measuring the Euclidean error only between true overlapping points, the best transformation $\mathbf{T}_b$ actually maximizes the sum of squared errors. Therefore, the $\mathbf{T}$ that produces the best alignment is the one that meets

$$E^* = \max_{\mathbf{T}} E\left(\mathbf{T} \mid \mathcal{C},\, \varepsilon\right). \tag{2.3}$$

2.1.2 Practical registration of LiDAR scans

Due to their partial overlapping and complex structures, 3D point clouds obtained from LiDAR scanners require practical considerations to ease and accelerate their registration. Cai et al. [31] considered that in practice relative rotations are constrained to the azimuth in terrestrial systems, removing two *Degrees of Freedom* (DOF) and making $\mathbf{T}$ defined only by $\mathbf{t} \in \mathbb{R}^3$ and $R \in SO(1)$ around the vertical axis. They also follow the definition of the sum of squared errors in **Equation 2.2** as it robustly estimates it and rejects false correspondences. In their case, ε is given by the precision of the scanning sensor, but do not consider the low overlap problem.

2.1.3 Supervoxel segmentation

There are many ways in which we can divide a 3D point cloud into subsets, such as selecting random points and use these as centers to find their k-nearest neighbors, or their radial neighbors. However, techniques like these are not consistent due to the random selection of initial points, and their processing time depends of the number of initial points and range of neighbors. Also, the likelihood to make subsets with repeated points (*i.e.*, the subsets cross their boundaries) increases with the number of initial points and range of neighbors. Hence, it is desired to adopt a method which run time is fast enough to use as a pre-processing and consistently produces subsets which boundaries do not cross between each other. The Voxel Cloud Connectivity Segmentation, better known as Supervoxel Segmentation (SVS) [32] is a method for the 3D space that has the desired characteristics and divides 3D point clouds via fast volumetric over-segmentations known as supervoxels using voxel relationships.

More specifically, SVS divides the 3D space into a voxelized grid space with a resolution of *seed radius* R_{seed}, where the voxels have a resolution of *voxel radius* R_{voxel} and a kd-tree-based adjacency graph is built in any given voxel, then it evenly places a set of seed points in the grid space from which expands to the clusters. This expansion consists of assigning voxels to each cluster using a local k-means clustering but constrained by the adjacency graphs and the flow defined from the expansion distance

$$\Delta_P = \sqrt{\lambda \frac{D_c^2}{m^2} + \mu \frac{D_s^2}{3R_{seed}^2} + \varepsilon D_{HiK}^2}, \qquad 2.4$$

Where λ controls the influence of color information as a Euclidean color distance D_c in the CIELab space [33], μ controls the spatial distance D_s between points, and ε controls the geometric similarity of the points measured by the *Fast Point Feature Histogram* (FPFH) [14] descriptor. The simplified version of the expansion distance in **Equation 2.5** is computed in a feature space consisting of the color Euclidean distance D_c from the normalized RGB space [34], the space distance D_s normalized by R_{seed}, and the normal distance D_n that measures the angle between normal vectors. Moreover, w_c, w_s, and w_n replace λ, μ, and ε respectively, being weights that individually control the effects of the distances.

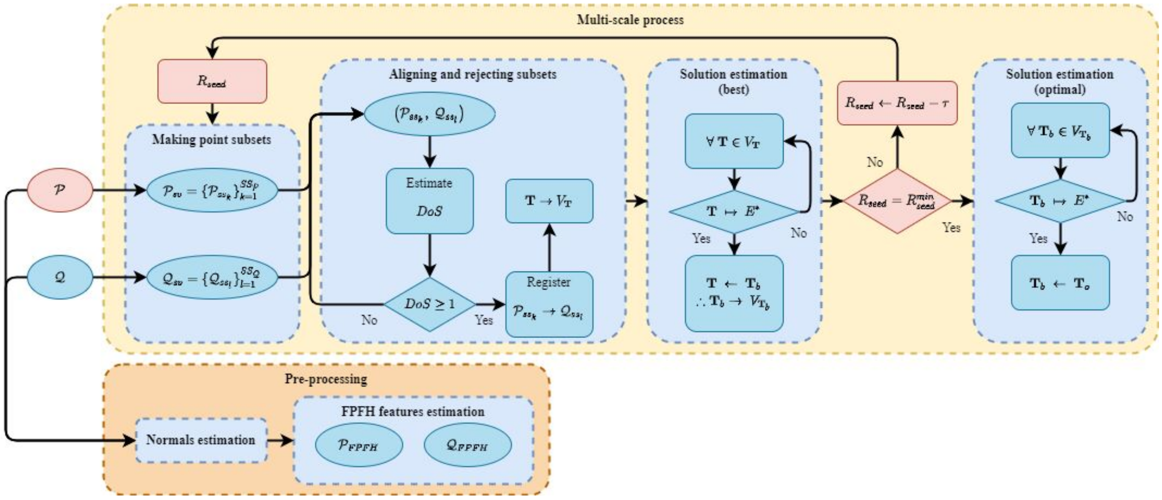

Multi-scale process
R_{seed}
Making point subsets
$\mathcal{P}$
$\mathcal{Q}$
$\mathcal{P}_{sv} = \{\mathcal{P}_{sv_k}\}_{k=1}^{SS_P}$
$\mathcal{Q}_{sv} = \{\mathcal{Q}_{ss_l}\}_{l=1}^{SS_Q}$
Aligning and rejecting subsets
$(\mathcal{P}_{ss_k}, \mathcal{Q}_{ss_l})$
Estimate
DoS
$DoS \geq 1$
No
Yes
Register
$\mathcal{P}_{ss_k} \to \mathcal{Q}_{ss_l}$
$\mathbf{T} \to V_{\mathbf{T}}$
Solution estimation
(best)
$\forall \mathbf{T} \in V_{\mathbf{T}}$
$\mathbf{T} \mapsto E^*$
Yes
No
$\mathbf{T} \leftarrow \mathbf{T}_b$
$\therefore \mathbf{T}_b \to V_{\mathbf{T}_b}$
$R_{seed} \leftarrow R_{seed} - \tau$
No
$R_{seed} = R_{seed}^{min}$
Yes
Solution estimation
(optimal)
$\forall \mathbf{T}_b \in V_{\mathbf{T}_b}$
$\mathbf{T}_b \mapsto E^*$
Yes
No
$\mathbf{T}_b \leftarrow \mathbf{T}_o$
Pre-processing
Normals estimation
FPFH features estimation
$\mathcal{P}_{FPFH}$
$\mathcal{Q}_{FPFH}$

Figure 2.1: Subset-based registration method flowchart.

$$\Delta'_P = \sqrt{w_c D_c^2 + w_s \frac{D_s^2}{3R_{seed}^2} + w_n D_n^2}, \qquad 2.5$$

The consistency in SVS is achieved thanks to its flow constrained clustering, but also by making good use of the voxelized space, as the occupied voxels closer to the evenly placed seed points in the voxelized space become the seed voxels. Hence, the definition of the seeding voxels from where the subsets expand is governed by the position and shape of the 3D point cloud in the voxelized space, and only changes if any of these properties changes. Moreover, the voxelized spaced along with the iterative expansion distances allow to precisely cluster planar and curved surfaces; meanwhile the adjacency graphs of the voxels help to avoid boundary crossing between subsets.

From the different parameters of SVS R_{seed} and R_{voxel} are the sizing parameters that determine the size and quantity of the supervoxels/subsets, but R_{seed} has a higher influence on defining their size as it controls the resolution of the grid where the seed points are placed.

2.2 Proposed method

With basis on the aforementioned problem reformulation we propose to exploit the local minima convergence and ideal condition of CRAs by clustering the 3D point clouds into point subsets via SVS and look for the pair of these that due to its similarity in shape and size perfectly align and registration produces the **T** that when applied to the full 3D point clouds fulfills **Equation 2.3**. The proposed registration method is depicted in the flowchart of **Figure 2.1** and described in the following subsections.

2.2.1 Pre-processing

The first step consists of simply computing the point normals of $\mathcal{P}$ and $\mathcal{Q}$ with a *normal radius* $R_{normals}$ of 10 CR, and the FPFH descriptors for all the points with an *feature radius* R_{FPFH} of also 10 CR.

2.2.2 Main process

Making point subsets

Next, both point clouds divide into subsets $\mathcal{P}_{sv} = \{\mathcal{P}_{ss_k}\}_{k=1}^{SS_{\mathcal{P}}}$ and $\mathcal{Q}_{sv} = \{\mathcal{Q}_{ss_l}\}_{l=1}^{SS_{\mathcal{Q}}}$ using SVS ruled by the seed radius R_{seed}.

Aligning and rejecting subsets combinations

Afterward, we obtain the transformations $\mathbf{T}$ from the registration of all the possible subsets combinations $(\mathcal{P}_{ss_k},\ \mathcal{Q}_{ss_l})$ using LM-ICP, with the Identity matrix as initial rotation $\mathbf{R}_0 = \mathbf{I}_3$ and a null initial translation $\mathbf{t}_0 = (0,\ 0,\ 0)$, and save them in a vector $V_{\mathbf{T}}$. The size of this vector is $SS_{\mathcal{P}} \times SS_{\mathcal{Q}}$ if all the combinations are registered. However, registering all combinations implies a high probability of obtaining transformations from subsets that do not correspond between each other. Thus, to avoid combinations of unrelated surfaces we set a *degree of geometric similarity DoS*, which measures how similar are a pair of subsets by counting the descriptor correspondences between them. In theory, the higher the value of DoS the more similar and geometrically related are the subsets. Hence, a pair $(\mathcal{P}_{ss_k},\ \mathcal{Q}_{ss_l})$ is rejected if DoS is zero, or registered if otherwise.

To estimate DoS between a pair of subsets $(\mathcal{P}_{ss_k},\ \mathcal{Q}_{ss_l})$ we need their one-to-one related descriptor subsets $(\mathcal{P}_{FPFHss_k},\ \mathcal{Q}_{FPFHss_l})$. Using the latter, point correspondences are estimated in the same manner as Buch et al.[35] and Lowe [36]: for each point descriptor $\mathbf{p}_{FPFH}$ in $\mathcal{P}_{FPFHss_k}$, the first and second nearest neighbors in $\mathcal{Q}_{FPFHss_l}$ are found. Then, if the ratio between their distances is lower than a *correspondence threshold* C_{thr}, the first nearest neighbor is considered a correspondence, and DoS increases by one.

Remark 2.2: LM-ICP for the registration of subsets

For the registration of the subsets, LM-ICP was chosen due to its robustness to the initial positions. A property desired for the registration of subsets that may be part of the same surface but have a different orientation in their respective point clouds.

Solution estimation

Lastly, the transformation that achieves the best alignment $\mathbf{T}_b$ is defined by evaluating all $\mathbf{T}$s in $V_{\mathbf{T}}$ with **Equation 2.1**. Then, $\mathbf{T}_b$ is the $\mathbf{T}$ that meets **Equation 2.2**.

2.2.3 Multi-scale process

All transformations are obtained from the registration of point subsets relatively similar in size. Thus, their size plays a significant role in the method. SVS's parameter R_{seed} controls the size of the subsets: the larger it is, the less but bigger are the generated point subsets and vice-versa. Since there is no predefined optimal R_{seed} value that leads to the optimal transformation $\mathbf{T}_o$, it is necessary to find them through a multi-scale evaluation. It means, iterating over the stages of the main process decreasing the value of R_{seed} at a step rate τ, from a maximum R_{seed}^{max} to a minimum R_{seed}^{min}.

In this multi-scale fashion, at each iteration R_{seed} has a value that produces different point subsets to register as described in the main process. Then the resulted transformations are evaluated to find $\mathbf{T}_b$ for each scale. All $\mathbf{T}_b$ from each iteration are saved in a vector $V_{\mathbf{T}_b}$ in order to perform the same evaluation with **Equation 2.1** and define the optimal transformation $\mathbf{T}_o$. That is, $\mathbf{T}_o$ is the $\mathbf{T}_b$ that satisfies **Equation 2.2**.

Remark 2.3: Definition of R_{seed}^{max} and R_{seed}^{min}

R_{seed}^{max} can be as significant enough to at least generate a couple of subsets at the first iteration, but due to the spatial relationship between supervoxels and the voxel grid in the segmentation process $R_{seed}^{min} = R_{voxel} + \tau$.

2.3 Experimental evaluation

2.3.1 Implementation and datasets

The proposed method was implemented in C++ using the Point Cloud Library (PCL) [37], along with OpenMP [38] for parallel processing of the subsets rejection, registration, and transformations evaluation. The experiments were conducted in a computer equipped with an Intel® Core i7-7700HQ (2.80 GHz × 8) and 32 GB of RAM on Ubuntu.

The utilized datasets were model 3D point cloud pair of *Armadillo*, *Bunny*, *Dragon*, and *Happy Buddha* from the Stanford 3D Scanning Repository [39]. Moreover, a laser scan pair

Table 2.1: Utilized datasets.

	Datasets	Sizes (points)	ξ_{gt} (%)
Armadillo	$\mathcal{P}$: `ArmadilloStand_0` $\mathcal{Q}$: `ArmadilloStand_270`	$N_{\mathcal{P}} = 28{,}220$ $N_{\mathcal{Q}} = 24{,}034$	26.66
Bunny	$\mathcal{P}$: `bun_0` $\mathcal{Q}$: `bun_270`	$N_{\mathcal{P}} = 40{,}256$ $N_{\mathcal{Q}} = 31{,}701$	30.90
Dragon	$\mathcal{P}$: `dragonStandRight_0` $\mathcal{Q}$: `dragonStandRight_288`	$N_{\mathcal{P}} = 41{,}841$ $N_{\mathcal{Q}} = 24{,}573$	29.64
Happy Buddha	$\mathcal{P}$: `happyStandRight_0` $\mathcal{Q}$: `happyStandRight_288`	$N_{\mathcal{P}} = 78{,}056$ $N_{\mathcal{Q}} = 72{,}346$	19.88
Stairs	$\mathcal{P}$: `Hokuyo_0` $\mathcal{Q}$: `Hokuyo_3`	$N_{\mathcal{P}} = 181{,}077$ $N_{\mathcal{Q}} = 187{,}959$	33.04

of *Stairs* from the ASL Datasets [40]. Each 3D point cloud pair has an overlapping ratio ξ_{gt} below 40%, measured based on the definition of **Equation 1.9** that considers both 3D point clouds. **Table 2.1** shows the specific datasets, the source-target arrangement, number of points, and corresponding ξ_{gt}.

We ran experiments recording the E and ξ of $\mathbf{T}_b$ at each iteration, as well as the processing time and the *Mean Squared Error with Penalty*, which is defined as

$$MSE_p = \frac{1}{N_{\mathcal{P}}} \sum_{i=1}^{N_{\mathcal{P}}} d_{pq} \qquad d_{pq} = \begin{cases} \left\|\mathbf{q}_j - \bar{\mathbf{p}}_i\right\|^2 & \text{if } d_{pq} \leq \varepsilon \\ 1 & \text{otherwise} \end{cases} \tag{2.6}$$

It measures the mean squared error between the true overlapping points and gives a penalty of 1 for each $\mathbf{p}_i \in \mathcal{P}$ that does not have a correspondence $\mathbf{q}_j \in \mathcal{Q}$ within ε. In low overlapping 3D point clouds MSE_p allows to estimate small values of error of alignment for point clouds that are correctly registered, and vice-versa.

Regarding to the parameters, we set the values shown in **Table 2.2** in terms of CR for the left column. C_{thr} represents a ratio $\in [0,1]$, and w_c, w_c, w_c the weights for SVS. Parameter w_c was set to zero because the datasets do not have color information, and C_{thr} as 0.1 to guarantee a strict rejection of unrelated subsets by the *DoS*. R_{seed}^{max} and τ, were defined large enough to generate at least two subsets at the first iteration, and small enough to gradually evaluate a wide range subset sizes. In the case of the laser scan R_{voxel} and R_{seed}^{max} were scaled up by ten because the size of this dataset is several times larger than the others.

Table 2.2: Parameters values

Parameter	Value [CR]	Parameter	Value
R_n	10	w_c	0
R_{FPFH}	10	w_w	2.5
R_{voxel}	2.5	w_n	0.5
R_{seed}^{max}	100	C_{thr}	0.1
τ	1		
ε	2		

2.3.2 Results and discussion

Table 2.3 the quantitative results obtained from the proposed method, comparing them to the ground-truth and three CRAs, and **Table 2.5** shows the qualitative results depicting $\mathcal{P}$ in red and $\mathcal{Q}$ in blue. It is worth mentioning that although TrICP is considered a CRA, it is not included in the comparison because it requires of the pre-knowledge of ξ which in a practical scenario is unknown.

Qualitatively, unlike CRAs, the proposed approach in most datasets keeps the shape of the datasets, obtaining alignments very close to the ground-truth. In the cases of *Bunny*, *Dragon*, *Happy Buddha*, and *Stairs*, the resulting alignments are visually almost perfect, and can be corroborated with the quantitative results of Table 2.3 since the metrics of the alignments from the proposed method are significantly closer to the ground truth. Although there are still some noticeable differences, as in the case of *Bunny*, these do not indicate a wrong result, but that there is still room for improvement in the alignment, which can be achieved by a *fine registration* stage.

With the same color pattern as in Table 2.5, **Figures 2.2(b)**, **2.3(b)**, **2.4(b)**, **2.5(b)**, and **2.6(b)** show the subsets that led to $\mathbf{T}_o$ in every dataset; meanwhile the left side corresponds to the complete subset segmentations (each subset is indicated by a color) generated by SVS with the value of R_{seed} indicated in **Table 2.4**. Examining *Armadillo*'s ground-truth and the leading subsets in Figure 2.2(b) we can notice that because the overlapping surfaces are mostly round, these are almost completely featureless. Therefore, at the subset rejection stage, many subset combinations coming from this parts pass to be aligned because their *DoS* points them as geometrically related, without considering if these arise from opposite sides of the surfaces causing an alignment with drift. A simple analogy to this is the registration of the two halves of a sphere, since both do not have any other geometric description than their roundness, the final registration will produce a bowl shape instead of the desired sphere. In figures 2.2 to 2.6 it is also clear that the method fulfills the expectation that registered point subsets of equal or similar size and shape satisfy Equation 2.2.

Table 2.3: Comparison of quantitative results between the ground-truth, proposed method, and CRAs

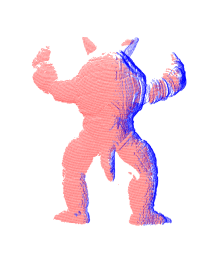
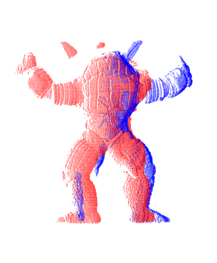
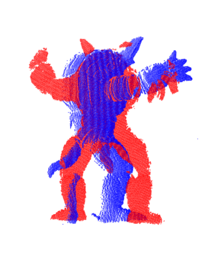
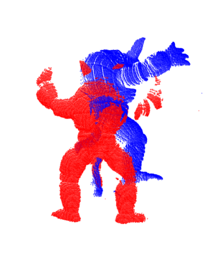
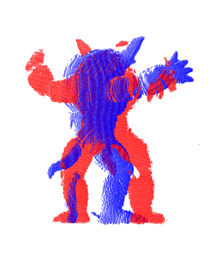
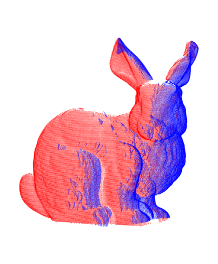
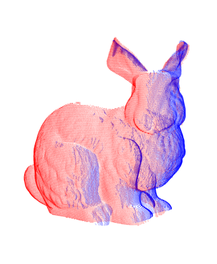
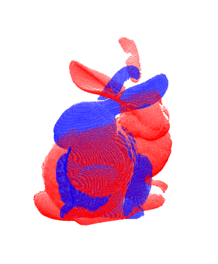
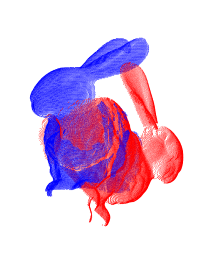
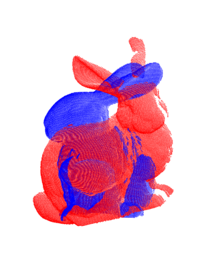
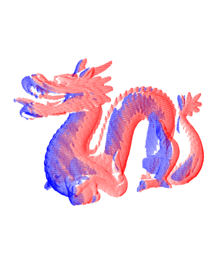
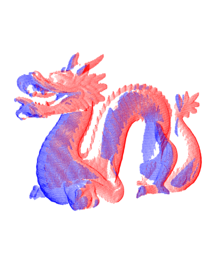
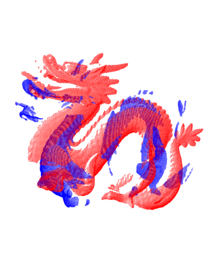
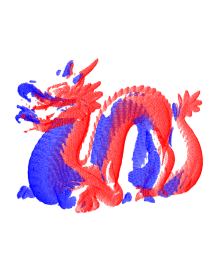
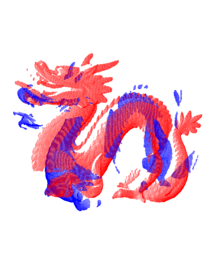
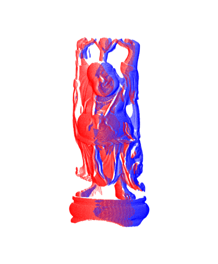
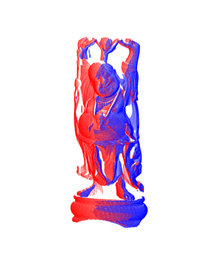
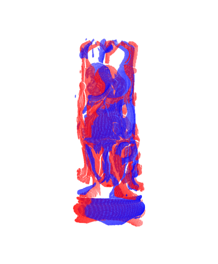
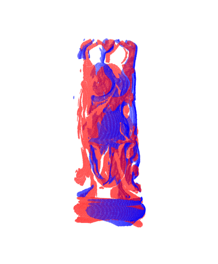
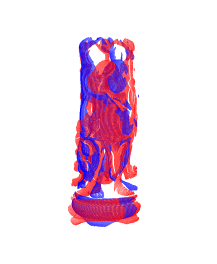
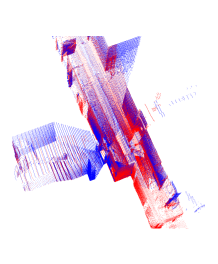
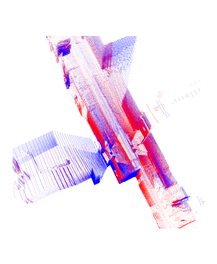
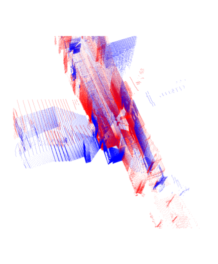
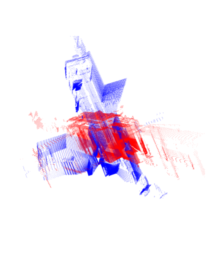
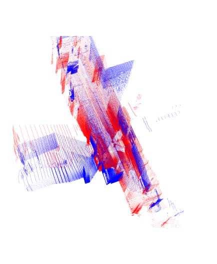
	Metric	**Ground-truth**	**Proposed method**	**ICP**	**N-ICP**	**LM-ICP**
Armadillo	E	4.56	4.03	0.64	0.77	0.64
	ξ (%)	26.66	23.63	3.67	4.61	3.63
	MSE_p	0.7533	0.7813	0.9660	0.9573	0.9663
Bunny	E	6.89	4.74	1.26	1.55	1.25
	ξ (%)	30.9	19.20	5.06	6.21	4.99
	MSE_p	0.7239	0.8284	0.9547	0.9444	0.9554
Dragon	E	5.48	5.62	1.41	1.45	1.43
	ξ (%)	29.64	30.18	6.27	6.24	6.36
	MSE_p	0.7648	0.7605	0.9502	0.9504	0.9495
H. Buddha	E	6.08	6.78	3.11	3.41	2.99
	ξ (%)	19.88	22.70	9.55	10.67	9.20
	MSE_p	0.8084	0.7813	0.9080	0.8971	0.9113
Stairs	E	896.42	812.27	132.32	96.11	128.05
	ξ(%)	33.04	26.31	4.09	2.93	3.87
	MSE_p	0.6682	0.7363	0.9589	0.9706	0.9611

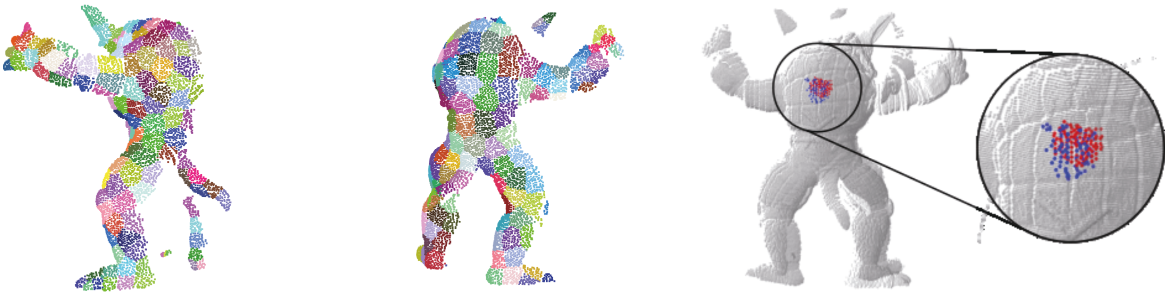
Table 2.4: Optimal R_{seed} and processing times

	R_{seed} $[CR]$	t_T **(sec)**	t_m **(sec)**
Armadillo	31	94.75	0.73
Bunny	70	444.74	2.47
Dragon	58	136.88	1.04
Happy Buddha	31	552.60	3.08
Stairs	927	5,925.2	0.98

Regarding the R_{seed} value where $\mathbf{T}_o$ originates from and the processing time, Table 2.4 shows the resulting optimal sizes and corresponding *total* t_T and *median* t_m processing times. Here it is expressed that the proposed method takes only a few minutes (between 10 min to 1.5 min) to process the model datasets, but the scaled-up values of R_{seed}^{max} and R_{voxel} for the laser scan dataset drastically impact on t_T because more values of R_{seed} have to be processed, taking approximately 1.64 hours to find a definitive solution.

Table 2.5: Comparison of qualitative results between the ground-truth, proposed method, and CRAs

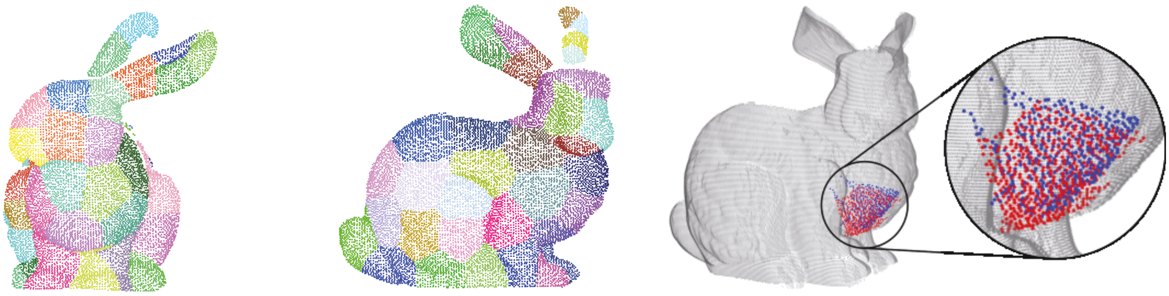
	Ground-truth	**Proposed method**	**ICP**	**N-ICP**	**LM-ICP**
Armadillo					
Bunny					
Dragon					
H. Buddha					
Stairs					



(a) Source and Target subsets

(b) Best subsets

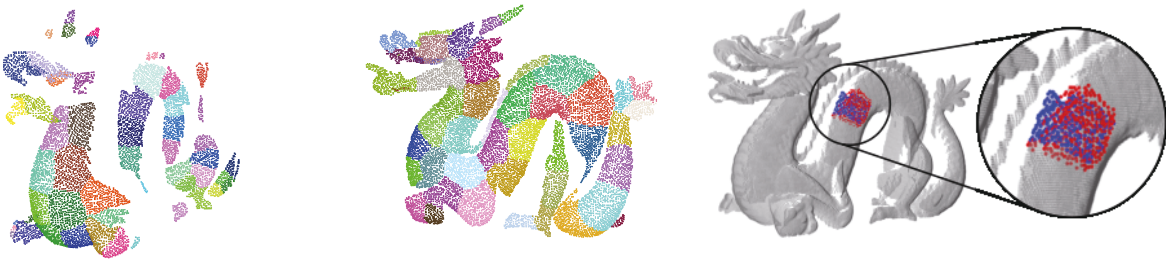
Figure 2.2: Segmentations and point subsets that lead to $\mathbf{T}_o$ in *Armadillo*



(a) Source and Target subsets

(b) Best subsets

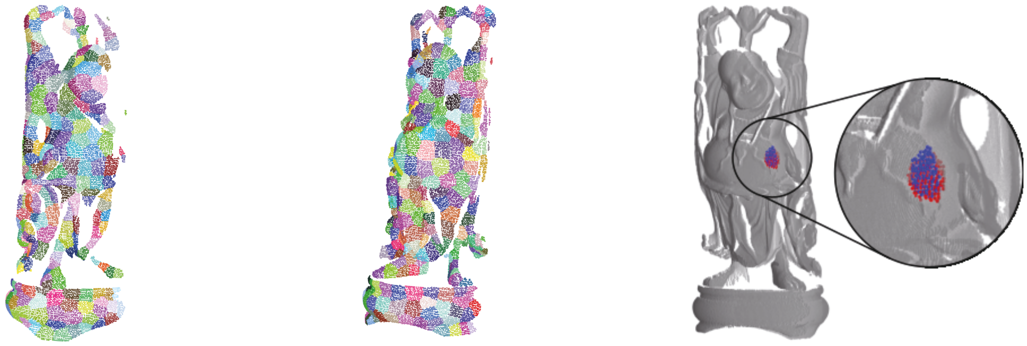
Figure 2.3: Segmentations and point subsets that lead to $\mathbf{T}_o$ in *Bunny*



(a) Source and Target subsets

(b) Best subsets

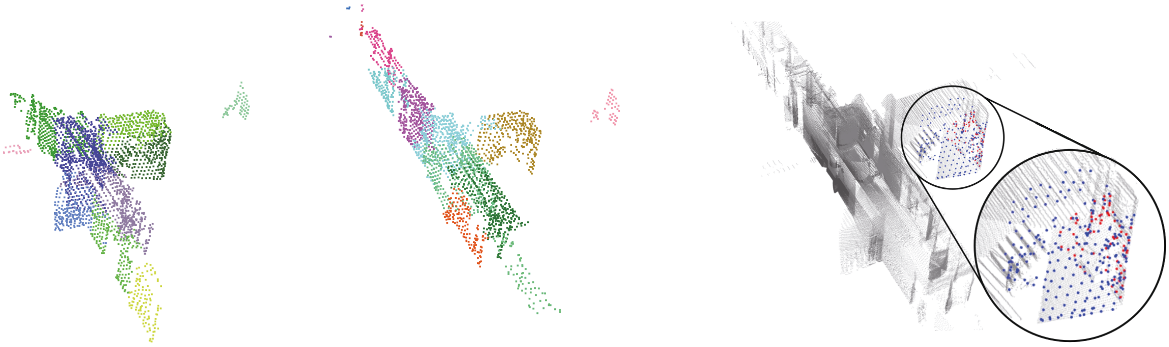
Figure 2.4: Segmentations and point subsets that lead to $\mathbf{T}_o$ in *Dragon*



(a) Source and Target subsets

(b) Best subsets

Figure 2.5: Segmentations and point subsets that lead to $\mathbf{T}_o$ in *Happy Buddha*



(a) Source and Target subsets

(b) Best subsets

Figure 2.6: Segmentations and point subsets that lead to $\mathbf{T}_o$ in *Stairs*

2.4 Conclusion

The proposed registration method can find transformations that register a pair of 3D point clouds with a low overlapping ratio close to the optimal alignment. Although the result in one of the tested datasets is not close to the optimal alignment, it is significantly better than CRAs. In reality, the proposed method excels the CRAs in all cases. Nevertheless, it might not be the optimal with large datasets due to its relatively long processing time. Furthermore, because of its core subset-to-subset registration, it is weak against 3D point clouds which shapes have large featureless surfaces or repeated patterns.

Chapter 3

Descriptor-free voting-based registration

This chapter comprehends the conception details, and preliminary experimentation of the second devised approach, presented in "Descriptor-free Voting-based Registration of Low Overlapping 3D Point Clouds" and published in Proceedings of the 48th Annual Conference of the Institute of Image Electronics Engineers of Japan.

3.1 Background and related work

3.1.1 Problem reformulation

The registration method introduced in the previous chapter satisfactorily achieves the main goal of registering low overlapping 3D point clouds. Nonetheless, it comprises several things that can be improved:

- The value for R_{seed}^{max} has to be set manually.
- The objective function can be satisfied by incorrect alignments.
- It utilizes feature point descriptors.

To tackle the first issue, we simply applied the multi-scale usage of SVS proposed by Ref.[41] which simply finds the mean maximum distances from the bounding boxes of the 3D point clouds to estimate R_{seed}^{max}. However, doing it arises a new problem: the total number of subsets to evaluate is very high because the search space given by all the possible sizes of R_{seed} increases, also increasing the total processing time. **Table 3.1** shows R_{seed}^{max} and total processing time t_T in each dataset before and after applying the automatic estimation.

Table 3.1: R_{seed}^{max} and t_T before and after automatic setting.

	R_{seed}^{max} **before (CR)**	R_{seed}^{max} **after (CR)**	t_T **before (sec)**	t_T **after (sec)**
Armadillo	100	290	94.75	275.02
Bunny	100	213	444.74	675.52
Dragon	100	256	136.88	310.46
H. Buddha	100	302	552.60	1,212.36
Stairs	1000	2183	5,925.2	13,057.87

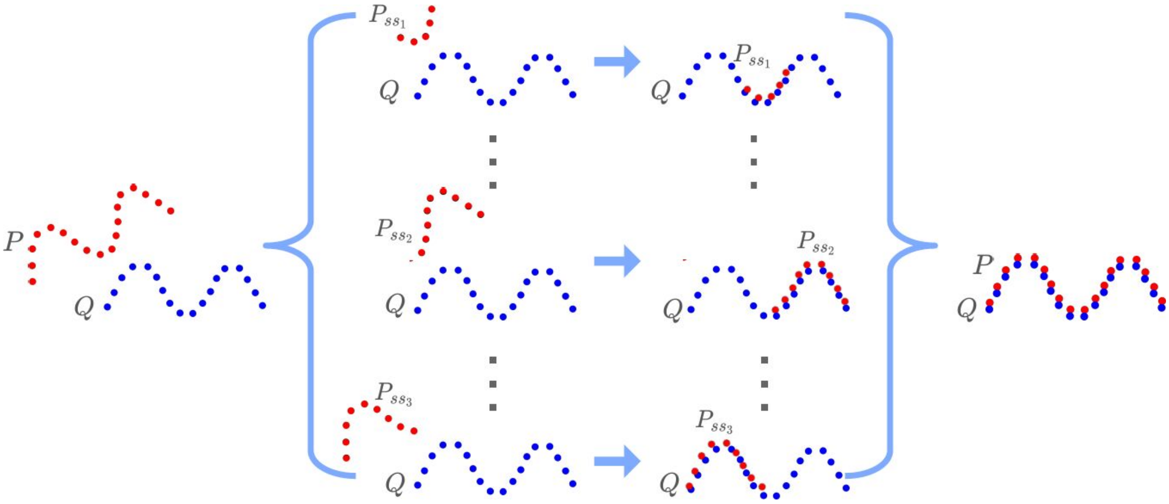


Figure 3.1: Core idea: same transformation from the alignment of different subsets.

By setting to perform SVS only in the source $\mathcal{P}$ it is possible to reduce in half the total number of subset registration to make, but it implies registering small subsets with a larger 3D point cloud and not having one-to-one full data association as originally intended. Nevertheless, we know that subsets $\mathcal{P}_{ss_k}$ still perfectly align somewhere in the true overlapping surfaces of the target $\mathcal{Q}$, and the corresponding transformations of these alignments are relatively similar or the same since they correspond to the registration of the overlapping areas. Thus, we can retrieve the alignment led by the subsets via a voting scheme without the need of feature descriptors to compute any similarity between subsets. In other words: by segmenting the source 3D point cloud and registering every subset with the target, the resulting transformations from the subsets belonging to the truly overlapping areas are rather similar and represent the solution transformation (*cf.* **Figure 3.1**) which can be retrieved via Hough voting.

To solve the matter of concern with the objective function in **Equation 2.3** it is necessary to describe good transformations more robustly. Thus, by keeping the definition of true point correspondences (*i.e.*, source points $\mathbf{p}_i$ with a target neighbor $\mathbf{q}_j$ that lays within the radius threshold ε of two times the cloud resolution of $\mathcal{P}$), and combining MSE_p (*cf.* **Equation**

2.6) with the objective function defined by Chetverikov et al.[42], we define a metric that describes *good transformations* as those that minimize the squared error between true point correspondences and maximize the overlapping ratio:

$$\psi = \frac{MSE_p}{\xi^{\lambda+1}}. \tag{3.1}$$

Remark 3.1: Value of λ

λ in the definition of ψ is a preset parameter $\lambda \geq 0$ that controls the influence of the overlapping ratio ξ. Increasing it allows to avoid undesirable alignments from featureless surfaces, and throughout this work we follow the setting $\lambda = 2$.

3.1.2 Hough voting

The *Hough Transform* (HT) [43] is a popular technique used in computer vision originally proposed for line detection in 2D images and later on was modified to detect also circles and ellipses. Its voting scheme and core idea is to vote for the image features in the parameter space of the shape to detect. The corresponding votes are accumulated into an array (the accumulator) which dimensions depend on the number of parameters of the desired shape. Then, the desired shape is found by the local maxima of votes in the accumulator.

A simple example is to understand the concept is the detection of lines in 2D images: given the edge points $(x_i,\ y_i)$ the task is to detect a line $y = mx\ +\ c$. Considering a point $(x_i,\ y_i)$ that passed through the desired line, we can write the equation $y_i = mx_i\ +\ c$ and rewrite it as $c = -mx_i + y_i$. Since $(x_i,\ y_i)$ are known, the latter equation allows us to look at the problem in two different spaces (the image space and the parameter space) and map between them. Then, each point in the image space corresponds to a line in the parameter space, and all the points in a line of the parameter space correspond to all the points that pass through a point in the image space. Hence, taking more points in the image space will give another line in the parameter space, and the crossing point of these lines in the parameter space correspond to the parameters that describe a line in the image space (see **Figure 3.2**). The algorithm for this line detection is described in **Algorithm 1**, where the parameter space is quantized and referenced by the accumulator $A\,(m,\ c)$.

Regarding the mechanics of HT, it is important to consider the impact of the size of bins in the accumulator: *if it is too big different shapes may be merged, but if it is too small, noise causes shapes to be missed.* Furthermore, the parameterization must be chosen carefully as parameters with infinite possibility of values require large accumulators, more memory, and

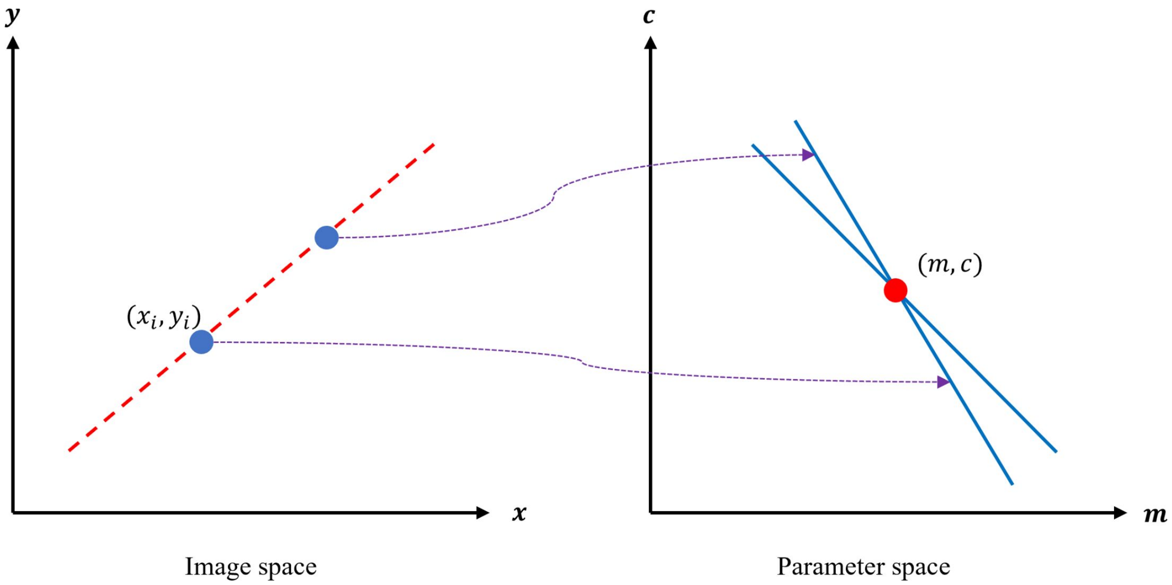


Figure 3.2: Mapping between the image space and parameter space for line detection.

computation. In theory, the latter also applies to shapes with many parameters as it would require a very sparse and high-dimensional accumulator of low computational efficiency and high memory demand.

Despite the wide use of HT in 2D images [44–47], in 3D is not broadly used due to the high number of parameters in 3D shapes. Originally HT was only used to detect 3D shapes with a relatively small number of parameters such as cylinders [48] and spheres [49]. Nevertheless, it has been show that the technique can be employed to detect free-from shapes in the 3D spaces with some arrangements hat reduce the size of the Hough space [50]. Additionally, we must consider that nowadays memory constraints tend to be relative to the hardware we have on hand and it is not necessarily a problem in most modern consumer level computers anymore.

Algorithm 1 Hough transform for line detection

```
Quantize parameter space $(m, c)$
Create accumulator array $A(m, c)$
Set $A(m, c) = 0$ for all $(m, c)$
for each edge point $(x_i, y_i)$ do
    if $(m, c)$ lies on the line $c = -mx_i + y_i$ then
        $A(m, c) = A(m, c) + 1$                                  ▷ Voting
    end if
end for
Find local maxima in $A(m, c)$
```

3.1.3 Voting-based registration

The introduction of the HT for registration 3D point clouds is relatively new [12]. The core idea is to leverage the high likelihood of CRAs to converge to an almost optimal alignment when having one-to-one data association and exploit the repeatability of transformations from the registration of subset pairs $(\mathcal{P}_{ss_k},\ \mathcal{Q}_{ss_l})$. Being those perfectly aligned subsets originated from the truly overlapping areas, the transformations binned into a parameter space represent the correct alignment of the full 3D point clouds. However, naive definitions of the point subsets might lead to extremely high time complexities.

One of the main contributions of Ref.[12] is to replace the 16 parameters matrix representation of the rigid body transformation for a concatenated vector of only 7 parameters. The key is to split $\mathbf{T}$ into the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, and replace the rotation matrix for its quaternion representation $a+b\hat{\mathbf{i}}+c\hat{\mathbf{j}}+d\hat{\mathbf{k}}$. Thus, the transformation can be represented by the vector $\mathbf{T}_{7D}=\left\{a,\ b,\ c,\ d,\ t_x,\ t_y,\ t_z\right\}$ which makes the accumulator $A\left(\mathbf{T}_{7D}\right)$ seven-dimensional.

In terms of quantization of the parameter space, because of the replacement to a quaternion representation the terms for the rotation are delimited by $[-1,1]$, but the translation can be infinite along any direction. To solve this issue Ref.[12] also proposes a delimitation of translation $\left[-t_{limit},\ t_{limit}\right]$ based on the size of the bounding boxes both 3D point clouds, where t_{limit} is one and a half times the difference between the overall minimum and maximum vertices of the bounding boxes.

Lastly, the bin in which every parameter of the transformation $\mathbf{T}_{7D}$ fits in $A\left(\mathbf{T}_{7D}\right)$ is defined by $b_i=\left(\rho-\rho_{min}\right)/N_b$. Where ρ is the corresponding parameter to fit and N_b the number of bins in every dimension of the accumulator $A\left(\mathbf{T}_{7D}\right)$.

3.2 Proposed method

Considering the reformulation aforementioned as well as related work, we present a 3D point clouds registration method that is descriptor-free and employs Hough voting to retrieve the solution from the alignment of source cloud subsets build by SVS. The proposed approach is detailed in the following subsections and shown in the flowchart of **Figure 3.3**.

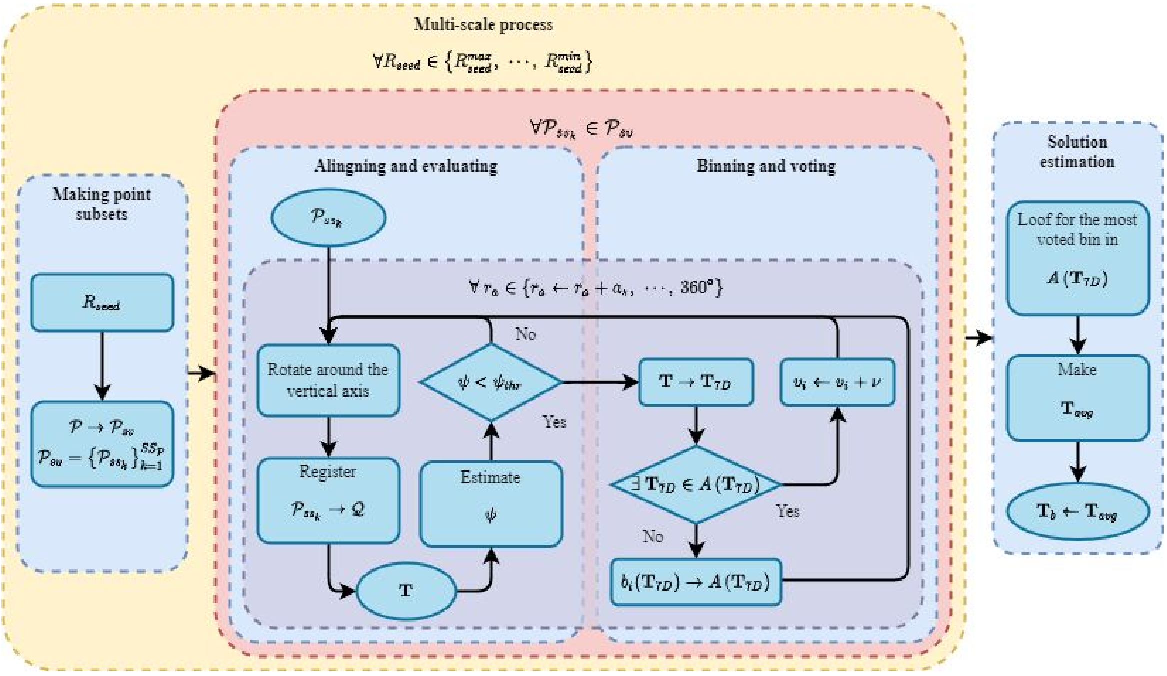

Multi-scale process
$\forall R_{seed} \in \{R_{seed}^{max}, \cdots, R_{seed}^{min}\}$
$\forall \mathcal{P}_{ss_k} \in \mathcal{P}_{sv}$
Making point subsets
R_{seed}
$\mathcal{P} \rightarrow \mathcal{P}_{sv}$
$\mathcal{P}_{sv} = \{\mathcal{P}_{ss_k}\}_{k=1}^{SS_\mathcal{P}}$
Alingning and evaluating
$\mathcal{P}_{ss_k}$
$\forall r_a \in \{r_a \leftarrow r_a + a_s, \cdots, 360°\}$
Rotate around the vertical axis
No
$\psi < \psi_{thr}$
Yes
Register
$\mathcal{P}_{ss_k} \rightarrow \mathcal{Q}$
Estimate
ψ
$\mathbf{T}$
Binning and voting
$\mathbf{T} \rightarrow \mathbf{T}_{7D}$
$v_i \leftarrow v_i + \nu$
$\exists \mathbf{T}_{7D} \in A(\mathbf{T}_{7D})$
Yes
No
$b_i(\mathbf{T}_{7D}) \rightarrow A(\mathbf{T}_{7D})$
Solution estimation
Loof for the most voted bin in
$A(\mathbf{T}_{7D})$
Make
$\mathbf{T}_{avg}$
$\mathbf{T}_b \leftarrow \mathbf{T}_{avg}$

Figure 3.3: Voting-based registration method flowchart.

3.2.1 Main process

Making point subsets

The first step consists of making point subsets only from the source $\mathcal{P}_{sv} = \{\mathcal{P}_{ss_k}\}_{k=1}^{SS_\mathcal{P}}$, by setting a value of *seed radius* R_{seed} via *Supervoxel Segmentation* (SVS).

Aligning and evaluating

In this stage, an initial rotation $\mathbf{R}_0$ corresponding to a rotation around the vertical axis at an angle r_a is applied to each subset $\mathcal{P}_{ss_k}$ and then registered to $\mathcal{Q}$ via ICP with null initial translation $\mathbf{t}_0 = (0,\ 0,\ 0)$ to obtain the corresponding $\mathbf{T}$ of their alignment. Since no features are used, $\mathbf{R}_0$ changes the initial position of the subset $\mathcal{P}_{ss_k}$ and improves ICP's basin of convergence. However, because the best initial position of $\mathcal{P}_{ss_k}$ is undefined, we perform several registrations with initial rotations $\mathbf{R}_0$ from different angles, progressively increasing r_a at an angle step a_s.

Next, the resulting $\mathbf{T}$ is applied to $\mathcal{P}$ and evaluated with the descriptive metric in **Equation 3.1**. At this point it is still possible to obtain wrong $\mathbf{T}$'s from the alignment of unrelated surfaces. Therefore, we discard these if $\psi < \psi_{thr}$, being ψ_{thr} a *false transformations threshold.*

Remark 3.2: Definition of the angle step a_s

The angle step at which the rotation angle r_a increases is defined as $a_s = 360°/N_r$, where N_r is a *number of rotations* set manually.

Binning and voting

Having a putative good $\mathbf{T}$, it is changed to a its seven-dimension vector representation $\mathbf{T}_{7D}$ and binned into a seven dimension accumulator $A\left(\mathbf{T}_{7D}\right)$ in the same manner as Ref.[12]. From this point the process uses the $\mathbf{T}_{7D}$ representation, but from now on it will referred as $\mathbf{T}$ for simplification purposes.

The voting process is a straightforward Hough voting scheme where the *votes* v_i of each *bin* b_i are increased by ν if the corresponding $\mathbf{T}$ exists in $A\left(\mathbf{T}_{7D}\right)$. Otherwise, b_i is created and its votes initialized as $v_i = \nu$, being ν a *voting step* of 1.

Solution estimation

The last stage consists of looking for the highest local maxima in the parameter space, it means locating the b_i that has the *maximum number of votes* v_{max} in the accumulator. Because v_{max} derives from a series of similar $\mathbf{T}$s we estimate an *averaged transformation* $\mathbf{T}_{avg}$ formed from all these, which represents them as $\mathbf{T}_b$. Notwithstanding, in the event of finding several local maxima with equal v_{max} $\mathbf{T}_{avg}$ is estimated for each ot these bins, then evaluated via ψ, and lastly the $\mathbf{T}_{avg}$ that satisfies $\psi \rightarrow \min$ is defined as the solution transformation $\mathbf{T}_b$.

3.2.2 Multi-scale process

Because the size of the subsets build by SVS is controlled by R_{seed}, in the same fashion of the method described in **Chapter 2** we perform stages the first three stages of the main process in a multi-scale approach from a *maximum seed radius* R_{seed}^{max} to a *minimum* R_{seed}^{min}. The latter are estimated based on the multi-scale usage of SVS [41] introduced in **Section 3.1.1**.

3.3 Experimental evaluation

3.3.1 Implementation and datasets

The present approach was also implemented in C++ using PCL [37] along with OpenMP [38], and evaluated with 3D point clouds pairs of the models *Armadillo*, *Bunny*, *Dragon*, and *Happy Buddha* from the Stanford 3D Scan Repository [39]. However, in comparison to the previous method, the datasets were arranged in inverse order, it means that the 3D point clouds labeled as 0 are the target $\mathcal{Q}$, meanwhile the others are the source $\mathcal{P}$. Experiments were carried out on the same system specified in **Section 2.3.1**.

In this method the two parameters that most influence the process are number of rotations N_r for the subset preregistration and the number of bins N_b in the accumulator $A\left(\mathbf{T}_{7D}\right)$. Therefore, we carried out preliminary experiments with four different values of $N_r = \{4,\ 8,\ 16,\ 32\}$, two $N_b = \{50,\ 100\}$, and set the threshold $\psi_{thr} = 0.1$. As for the parameters of SVS, we kept them as originally set in the experimental setup of **Section 2.3.1**, and empirically set $R_{seed}^{min} = 10\ R_{voxel}$ since evaluations below that scale do not generate subsets large enough to be registered by ICP.

Lastly, to assess the quality of the results, we evaluated the solution $\mathbf{T}_b$ and the ground-truth $\mathbf{T}_{gt}$ with **Equation 3.1** and estimated their absolute difference:

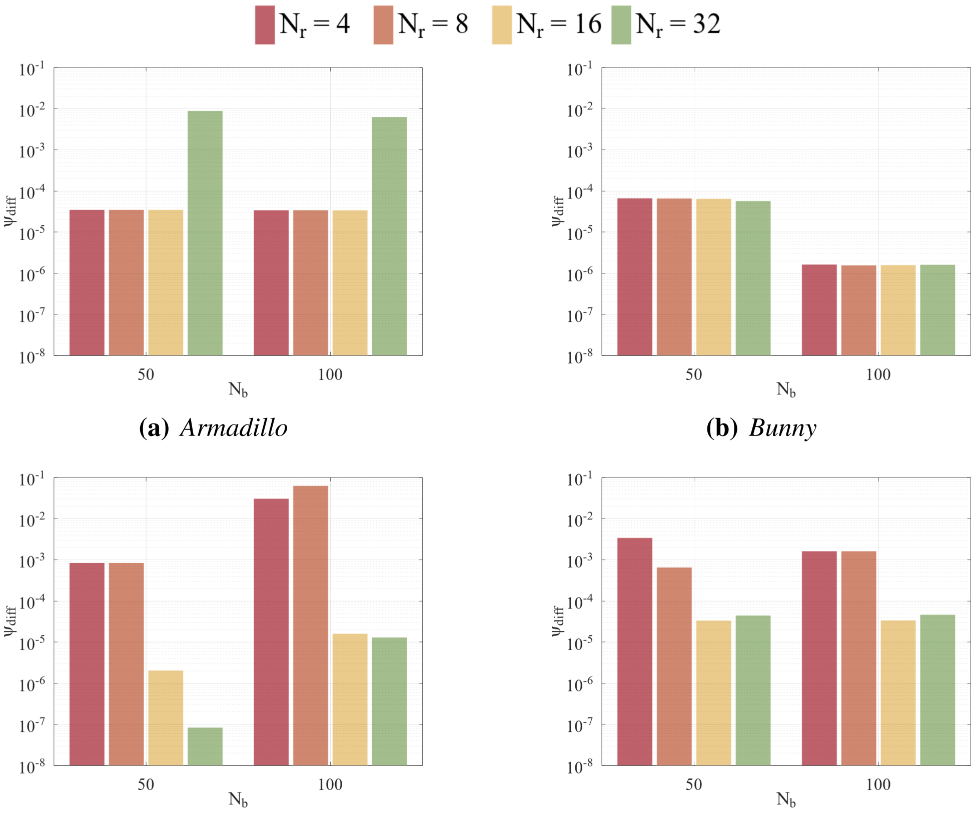
$$\psi_{diff} = \left|\psi_{gt} - \psi_b\right|. \qquad 3.2$$

3.3.2 Results and discussion

Table 3.2 shows the obtained qualitative results distinguishing the source $\mathcal{P}$ in red, the target $\mathcal{Q}$ in blue, and the overlap $\mathcal{C}$ in black. Charts in **Figure 3.4** summarize the quantitative results where the vertical axis shows ψ_{diff} in logarithmic scale. An analysis to the qualitative results clearly indicates that the proposed method can obtain solution transformations $\mathbf{T}_b$ that align the 3D point clouds very well in most cases, meanwhile the numeric results infer that these are significantly close to the optimal since ψ_{diff} ranges between $\left(10^{-7},\ 10^{-1}\right)$.

Quantitative results also reveal a trend to get better results as N_r increases regardless of the value on N_b in the datasets with stronger geometric features (*Dragon* and *Happy Buddha*). This trend does not apply in *Armadillo* nor *Bunny* (*cf.* **Figure 3.4(a)** and **Figure 3.4(b)**). However, the prominent round shape of *Bunny* allows to have a more stable behavior with the increase of rotations and better results with a more strict accumulator $A\left(\mathbf{T}_{7D}\right)$. In the case of *Armadillo*, because the metric in Equation 3.1 describes transformations that maximize ξ, the higher it is the lower ψ becomes and successfully passes through the threshold ψ_{thr}. Thus, because as N_r increases more subsets with little geometric features from the back of the model perfectly align producing many $\mathbf{T}$s that successfully pass through ψ_{thr} and yield most votes in $A\left(\mathbf{T}_{7D}\right)$, the solution converges to a local alignment.

With respect to the differences between the results of 50 and 100 bins, the accumulator $A\left(\mathbf{T}_{7D}\right)$ differentiates transformations more strictly as N_b increases. It seems that the parameter N_b does not have as big of an impact on the results compared to N_r, but may



(a) *Armadillo*

(b) *Bunny*

(c) *Dragon*

(d) *Happy Buddha*

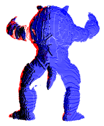
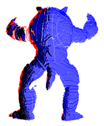
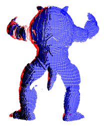
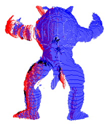
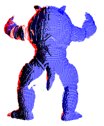
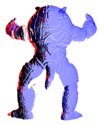
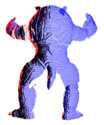
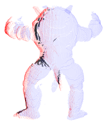
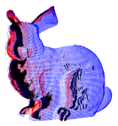
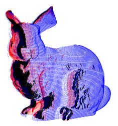
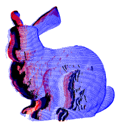
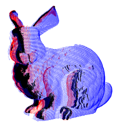
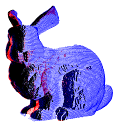
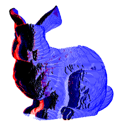
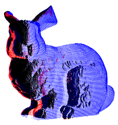
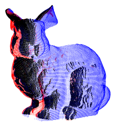
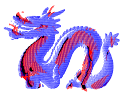
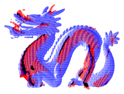
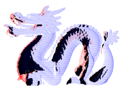
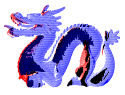
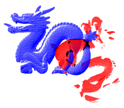
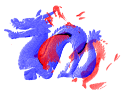
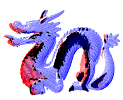
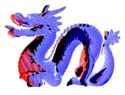
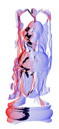
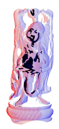
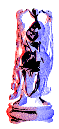
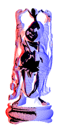
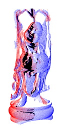
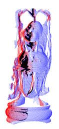
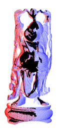
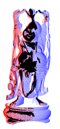
Figure 3.4: Quantitative results of preliminary experiments.

lead to local incorrect solutions when N_r does not provide good initial positions to improve the basin of convergence of ICP. This behavior is evident in the results of *Dragon* with $(N_r = 4,\ N_b = 100)$, and $(N_r = 8,\ N_b = 100)$. Lastly, comparing qualitative results with the approach presented in Chapter 2, this method can find an almost perfect solution for all mode-type datasets, as in the previous approach the somewhat featureless surface on the back of *Armadillo* turned out to be challenging.

3.4 Conclusion

The proposed descriptor-free voting-based registration method is capable to successfully find fairly good transformations for low overlapping ratio 3D point clouds of model-type datasets. Although the approach achieves its primary objective, there is a dependency on the parameters N_r and N_b, as well as in the 3D point clouds' shape to obtain good results. Hence, a more comprehensive analysis of its behavior under a wider range of function parameters is required to define a precise usage reference.

Table 3.2: Qualitative results of preliminary experiments.

	$\mathbf{N_b}$	$\mathbf{N_r = 4}$	$\mathbf{N_r = 8}$	$\mathbf{N_r = 16}$	$\mathbf{N_r = 32}$
Armadillo	50				
	100				
Bunny	50				
	100				
Dragon	50				
	100				
Happy Buddha	50				
	100				

Chapter 4

Evaluation of piece-wise voting for low overlapping 3D point clouds

The present chapter centers on the comprehensive evaluation carried out on the registration methods introduced in the previous chapter, which results were presented in "A Piece-wise Registration Method for Low Overlapping 3D Point Clouds" at the 7th IIEEJ International Conference on Image Electronics and Visual Computing (IEVC 2021); and extended in "A Registration Method on Piece-wise Voting for Low Overlapping 3D Point Clouds" for the IIEEJ Transactions on Image Electronics and Visual Computing, Vol. 9, No. 2.

4.1 Background and related work

4.1.1 Open questions

Going back to last chapter's experiments, these were designated as preliminary because do not provide a clear idea on how the framework can be improved and behaves under the variation of the parameters N_r and N_b. Additionally, a close analysis to the main process in Section 3.2.1 shows that the registration algorithm to align $\mathcal{P}_{ss_k}$ to $\mathcal{Q}$ also has an impact on the whole process. Therefore, we name this as *core registration* and open the possibility to use any of the CRAs to perform it. Considering the latter, the open questions to the proposed descriptor-free voting-based registration methods are:

- *How does it behave under variations of its ruling parameters N_r, N_b, and core registration?*

- *How does it perform compared to available state-of-the-art registration methods for the low overlap problem?*

4.1.2 State-of-the-Art on registration of low overlapping 3D point clouds

As previously mentioned in **Chapter 1**, to date the volume of work that addresses pairwise registration of low overlapping 3D point clouds is very scarce, but the interest in solving the problem increased in recent years. Addressing it in different fashions: using relatively robust point descriptors, on the basis of the interaction between neighboring points, and utilizing deep learning. Next, there is an introduction to some of these registration approaches that we can refer to as the *State-of-the-Art* (SoTA).

Descriptor-based

Despite the factors previously mentioned in **Section 1.2.4** about the conventional pipeline used for partial overlapping registration, there are feature-based methods for low overlap 3D point clouds. One of the pioneering works on the topic [30] addresses the problem assuming that the overlapping areas of the source cloud $\mathcal{P}$ are located at the borders. Thus, it samples border points and uses their Gaussian and mean curvatures [51] as local descriptors to match and make data association. Nevertheless, its dependence on non-robust local point descriptors makes it unstable method on 3D point clouds with uneven point densities – turning it not viable for noisy datasets obtained from laser scanners or low-end 3D sensors. Additionally, the experiments in this work were only carried out on the *Bunny* model of the Stanford 3D Scan Repository [39] which makes the extensibility of the approach inconclusive.

Sun et al. [52, 53] made point subsets from the source and target to decompose them via the K-SVD [54] algorithm, build sparse *patch features*, make data association by the sparse representations, and retrieve the solution $\mathbf{T}$ by registering matched subsets using the *Coherent Point Drift* [55] algorithm. Here, in a similar fashion as the approaches presented in this work, the subsets act as the registration unit to avoid potential mismatches when using the whole point clouds. The authors claim good performance and robustness to noise, but the experiments were carried on handcrafted datasets with consistent point densities that call into question the reported performance and reliability. Furthermore, for the registration problem itself it is worth to consider that the dictionary of K-SVD is defined by minimizing a high non-convex function that does not guarantee finding useful features that yield a good alignment [56].

The method developed by Prokop et al.[57] uses line features from both point clouds fed into a trial-and-error search to match the largest number of lines, and separates the Euclidean transformation's parameter space to separately look for each one via individual fitness functions – claiming being able to register 3D point clouds with overlapping ratios down to 20%. However, the approach is tailored to specific scanning conditions that delimit its applications. In it, there is an assumption that the 3D point clouds' coordinate system is aligned to the 3D sensor, which allows to estimate the rotation **R** only around the vertical z-axis and limit the translation **t** to the x-y plane. Besides, it also works under the assumption that straight edges strictly exist in the 3D point clouds, restricting its applicability to urban and man-made environments. Additionally, the approach's parameters are very susceptible to the dataset's shape, making it hard to tune in practical situations.

Neighboring interaction-based

An analogous approach to using point feature descriptors but without making hard point-to-point data association, is to exploit the information that neighboring points of the points in the overlapping areas possess, allowing to isolate from non-overlapping areas and focus on the region of interest. The HMRF-ICP [58] method exploits point neighboring interactions by using a *Hidden Markov Random Field* to infer data association via the *Expectation-Maximization* algorithm within the process of ICP. This approach reports good alignments on laser scan 3D point clouds but also failure cases driven by insufficient modeling of outlier points.

Deep learning-based

As mentioned in Section 1.2.4, deep-learning has been applied to develop 3D point cloud registration methods [21–24], but it was until very recently that the technique was utilized with focus on the low overlap registration problem. *Point-cloud REgistration with Deep Attention To the Overlap Region* (PREDATOR) [59] is a neural network architecture assembled to learn to detect the overlapping areas and focus on them to sample deep learning-based convolutional feature descriptors. In other words, the approach learns where to sample robust feature point descriptors for a more accurate data association and alignments. Another very recent approach combines deep learning with HT [60] to build a framework that, just like the approaches of this work, leverages Hough voting in a 6D transformation parameter space. It also extracts deep learning-based features to estimate a series of putative correspondences, and assemble a set of triplets from these to cast votes on the parameter space. Votes that are refined through a fully convolutional refinement module and finally predict the solution transformation by the consensus among the correspondences in the refined parameter space.

Both approaches outperform non-low-overlapping-focused registration methods in the low overlap problem on popular 3D deep learning benchmarks. Nevertheless, as most deep learning-based methods, these demand many datasets to train for a specific domain which makes them not viable when there is insufficient training data for a new type of dataset.

Moreover, nowadays deep learning models for 3D point cloud processing are considerably resource-heavy and require vast amounts of VRAM to train and operate. PREDATOR is so complex and demanding that the authors reported using a batch size of *one* in all the experiments because of memory restrictions even with 2021 top-of-the-line hardware.

4.2 Votes compensation

The experiments performed for the extended evaluation of this chapter utilize a somewhat improved version of the voting-based registration approach. In reality, the method is the same but replaces the voting step $\nu = 1$ for the definition $\nu = N_b/c_b$ that endures compensating votes as the bin size in the accumulator $A\left(\mathbf{T}_{7D}\right)$ decreases; where c_b is a *compensation factor*. This compensation is useful to allow more differentiation between potentially good transformations in the parameter space when the discrimination is rigorous.

4.3 Experimental evaluation

Since the main objective is to answer the questions raised in **Section 4.1.2**, we conducted two types of experiments: first, and *ablation study* to evaluate the behavior of the proposed voting-based registration methods under different values of the ruling parameters; and second, a *comparative study* between both methods proposed in this work and other SoTA approaches.

4.3.1 Datasets

In the ablation study the experiments were carried out on the same model-type datasets from the Stanford 3D Scan Repository used throughout this work, as well as in the pair of laser scan 3D point clouds of the environment *Stairs* from the ASL Datasets. However, there is a difference in the pair of 3D point clouds of the *Armadillo* model, as the source dataset `ArmadilloStand_270` was replaced for `ArmadilloStand_330` to attempt to obtain more shape descriptiveness and avoid potential drifted alignments result of the back of the model. Additionally, three point cloud pairs from the test set scenes *At Home*, *Study Room*, and *Maryland Hotel 1* of 3DLoMatch [59] dataset were included in order to test on a dataset tailored for the low overlap problem. **Table 4.1** specifies the employed 3D point clouds with their sizes and ground-truth overlapping ratio ξ_{gt}. It is worth mentioning that the overlapping ratio ξ_{gt} in this case compared to what is shown in **Table 2.1** differs due to switched order of source $\mathcal{P}$ and target $\mathcal{Q}$ clouds. In the case of Stairs, the change is particularly noticeable but the datasets were kept the same for uniformity purposes between

Table 4.1: Utilized datasets

	Datasets	**Sizes (points)**	ξ_{gt} (%)
Armadillo	$\mathcal{P}$: `ArmadilloStand_330` $\mathcal{Q}$: `ArmadilloStand_0`	$N_{\mathcal{P}} = 28{,}415$ $N_{\mathcal{Q}} = 28{,}220$	32.72
Bunny	$\mathcal{P}$: `bun_270` $\mathcal{Q}$: `bun_0`	$N_{\mathcal{P}} = 31{,}701$ $N_{\mathcal{Q}} = 40{,}256$	33.62
Dragon	$\mathcal{P}$: `dragonStandRight_288` $\mathcal{Q}$: `dragonStandRight_0`	$N_{\mathcal{P}} = 24{,}573$ $N_{\mathcal{Q}} = 41{,}841$	29.64
Happy Buddha	$\mathcal{P}$: `happyStandRight_288` $\mathcal{Q}$: `happyStandRight_0`	$N_{\mathcal{P}} = 72{,}346$ $N_{\mathcal{Q}} = 78{,}056$	20.37
Stairs	$\mathcal{P}$: `Hokuyo_3` $\mathcal{Q}$: `Hokuyo_0`	$N_{\mathcal{P}} = 187{,}959$ $N_{\mathcal{Q}} = 181{,}077$	50.98
At Home	$\mathcal{P}$: `cloud_bin_3` $\mathcal{Q}$: `cloud_bin_0`	$N_{\mathcal{P}} = 190{,}519$ $N_{\mathcal{Q}} = 425{,}577$	27.24
Study Room	$\mathcal{P}$: `cloud_bin_5` $\mathcal{Q}$: `cloud_bin_2`	$N_{\mathcal{P}} = 428{,}948$ $N_{\mathcal{Q}} = 334{,}279$	43.25
Maryland Hotel 1	$\mathcal{P}$: `cloud_bin_3` $\mathcal{Q}$: `cloud_bin_2`	$N_{\mathcal{P}} = 357{,}247$ $N_{\mathcal{Q}} = 630{,}366$	15.66

both proposed registration approaches. The target clouds $\mathcal{Q}$ from *At Home*, *Study Room*, and *Maryland Hotel 1* were moved to the ground-truth position, so in the experiments we look for the solution transformation $\mathbf{T}_b$ that aligns the source $\mathcal{P}$ to the global reference frame, which allows us to compare it to the ground-truth $\mathbf{T}_{gt}$. Furthermore, due to the significant larger size of *At-home*, *Studyroom*, and *Maryland Hotel 1*, voxel down-sampling with a leaf size of 0.1 applied in order to reduce the processing time.

4.3.2 Evaluation measures

To perform a better evaluation of any given alignment, we compare them to the ground-truth by ψ_{diff} of **Equation 3.2**, the *rotation error* R_{error} as the geodesic distance between the rotations [61], and the *translation error* t_{error} as the L-2 norm between the translations:

$$R_{error} = \arccos\left(\left|\frac{\mathrm{tr}\left(\mathbf{R}_b^{\top}\mathbf{R}_{gt}\right) - 1}{2}\right|\right) \qquad t_{error} = \left\|\mathbf{t}_{gt} - \mathbf{t}_b\right\|. \qquad 4.1$$

4.3.3 Ablation study

Ablation experiments setup

As mentioned earlier, the ruling parameters are the number of bins N_b, the number of rotations N_r, and the core registration algorithm, from which we can form configurations named *scenarios* composed as (dataset, N_b, N_r, CRA). To expand the range of evaluated values for each parameter, each dataset was tested with four values for the number of bins $N_b = \{10,\ 30,\ 60,\ 90\}$, five for the number of rotations $N_r = \{4,\ 8,\ 16,\ 32,\ 64\}$, and the four CRAs (*i.e.,* ICP, N-ICP, TrICP, and LM-ICP) as the core registration algorithm – resulting in a total of 640 scenarios to evaluate. With regard to the inner parameters, we kept $\psi_{thr} = 0.1$ and empirically set the compensation factor $c_b = 10$. These experiments were also conducted on the same system specifications introduced in Section 2.3.1.

Ablation results

The charts in **Figure 4.1** to **Figure 4.8** show the quantitative results on R_{error}, t_{error}, and ψ_{diff} of all scenarios. Each triplet of diagrams corresponds to one CRA as core registration, where the vertical axes express the evaluation measures (ψ_{diff} in logarithmic scale), the horizontal axes to N_b, and each series to N_r.

At a glance, it is clear that t_{error} does not vary along with the changes on the triplets (N_b, N_r, CRA), only between datasets. Resulting in about $0.001\ m$ $(1\ mm)$ in all model-type datasets and over $1\ m$ in the laser scan as its scale is significantly larger. Furthermore, like in the results of the preliminary experiments on Chapter 3, it seems that ψ_{diff} does not work as the definitive indication of the performance of the method. For example Bunny with TrICP, ψ_{diff} indicates that $N_b = 10$ gives the worst results but comparing to R_{error}, $N_b = 10$ performs better than $N_b = 90$. Nonetheless, this happens because of the discrimination level of the accumulator $A\left(\mathbf{T}_{7D}\right)$ and the fact that wrong alignments can still satisfy $\psi \leq \psi_{thr}$. As in low values of N_b **T**s with significant differences, good or wrong, can be placed into the same bin, then when retrieving the solution by $\mathbf{T}_{avg}$ the misalignment can be diminished or magnified depending on the amount of *good* and *bad* **T**s in the same bin - which is more noticeable in the $\mathbf{R}_{error}$. However, all the corresponding **T**s met $\psi \leq \psi_{thr}$ and the differentiation between good and bad ones does not transcend to ψ_{diff} as it is estimated from $\mathbf{T}_b$, not from an averaged ψ. Thus, we can resolve that assessing an alignment solely by the evaluation measure ψ_{diff} is not enough.

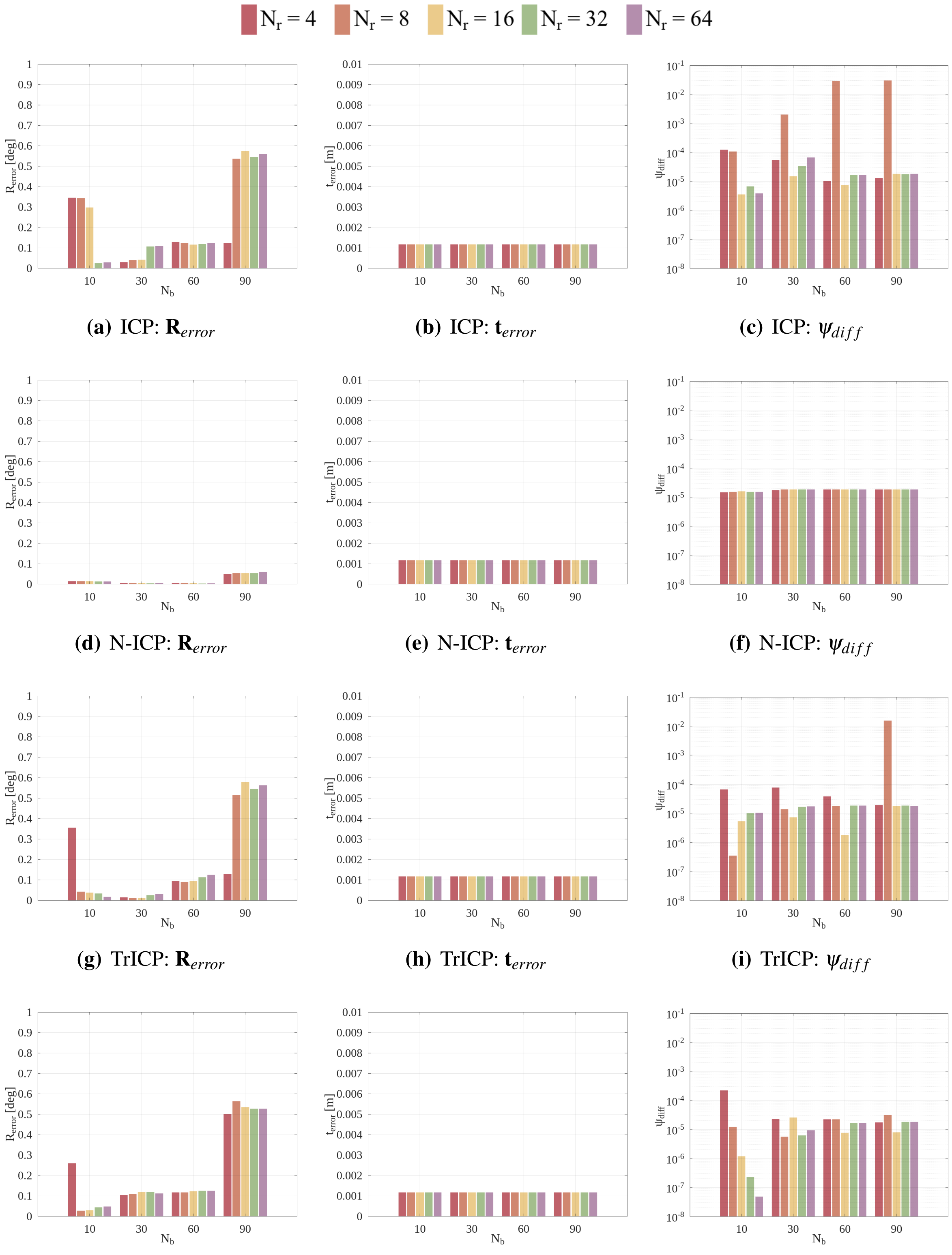
Results on R_{error} provide clearer indications on the performance of the method. N-ICP as core registration operates best in *Armadillo* and *Bunny* because this CRA leverages the uniform distribution of the normals on the large round surfaces of these datasets. Such occurrence also takes place in *Happy Buddha* as long as $A\left(\mathbf{T}_{7D}\right)$ does discriminate **T**s rigorously (*cf.* R_{error} in **Figure 4.4(a)** at $N_b = 90$). In the case of *Dragon* because its shape is more pronounced than the other models, there is higher variation in the results, but N-ICP and LM-ICP perform the best with $N_b \geq 30$. For Stairs, N-ICP converges to alignments with translation drift consequence of the large planar surfaces of this dataset. Overall, the number of rotations N_r seems to be dependent of the number of bins N_b and, more importantly, the core registration algorithm. However, $N_r \geq 16$ is steadier with $30 \leq N_b \leq 60$.

The datasets from 3DLoMatch showed a more steady behavior along the variations of $(N_b,\ N_r)$, due to the uniformity in point density obtained after the applied voxel down-sampling. However, with significantly larger errors in the three metrics as shown in Figures 4.6, 4.7, and 4.8. Most $\mathbf{R}_{error}$ values range from nearly 1° to 3°, which may not seem big for the scale of these datasets, but the $\mathbf{t}_{error}$ indicate significant translation errors between 1.5 to 5 m. ψ_{diff} values on these datasets also describe a poor performance, surpassing or being very close to $\psi_{thr} = 0.1$ in all cases. Notwithstanding, judging from the results in $\mathbf{R}_{error}$ and ψ_{diff} combined, one can tell that the method performs best in these datasets with N-ICP, $N_b \geq 60$, and $N_r = 64$ (*cf.* N-ICP in Figures 4.6, 4.7, and 4.8).

Although the results are somewhat noisy and allegedly no configuration consistently performs best, the scenarios with minimum R_{error}, t_{error}, and ψ_{diff} are:

- (*Armadillo*, 60, 32, N-ICP)
- (*Bunny*, 60, 8, LM-ICP)
- (*Dragon*, 30, 4, N-ICP)
- (*Happy Buddha*, 60, 32, N-ICP)
- (*Stairs*, 60, 4, ICP)
- (*At Home*, 10, 16, LM-ICP)
- (*Study Room*, 90, 64, N-ICP)
- (*Maryland Hotel 1*, 90, 64, N-ICP)

The corresponding alignments to these scenarios are shown in **Figure 4.11**, where the source $\mathcal{P}$ is depicted in red and the target $\mathcal{Q}$ in blue. It also clearly shows the translation drift $t_{error} \approx 1\ m$ in *Stairs*, result of many transformations from $\mathcal{P}_{ss_k}$ subsets originated at the planar surfaces that perfectly align to the large planar areas of the target $\mathcal{Q}$, as well as the misalignments in *At Home*, *Study Room*, and *Maryland Hotel 1*.



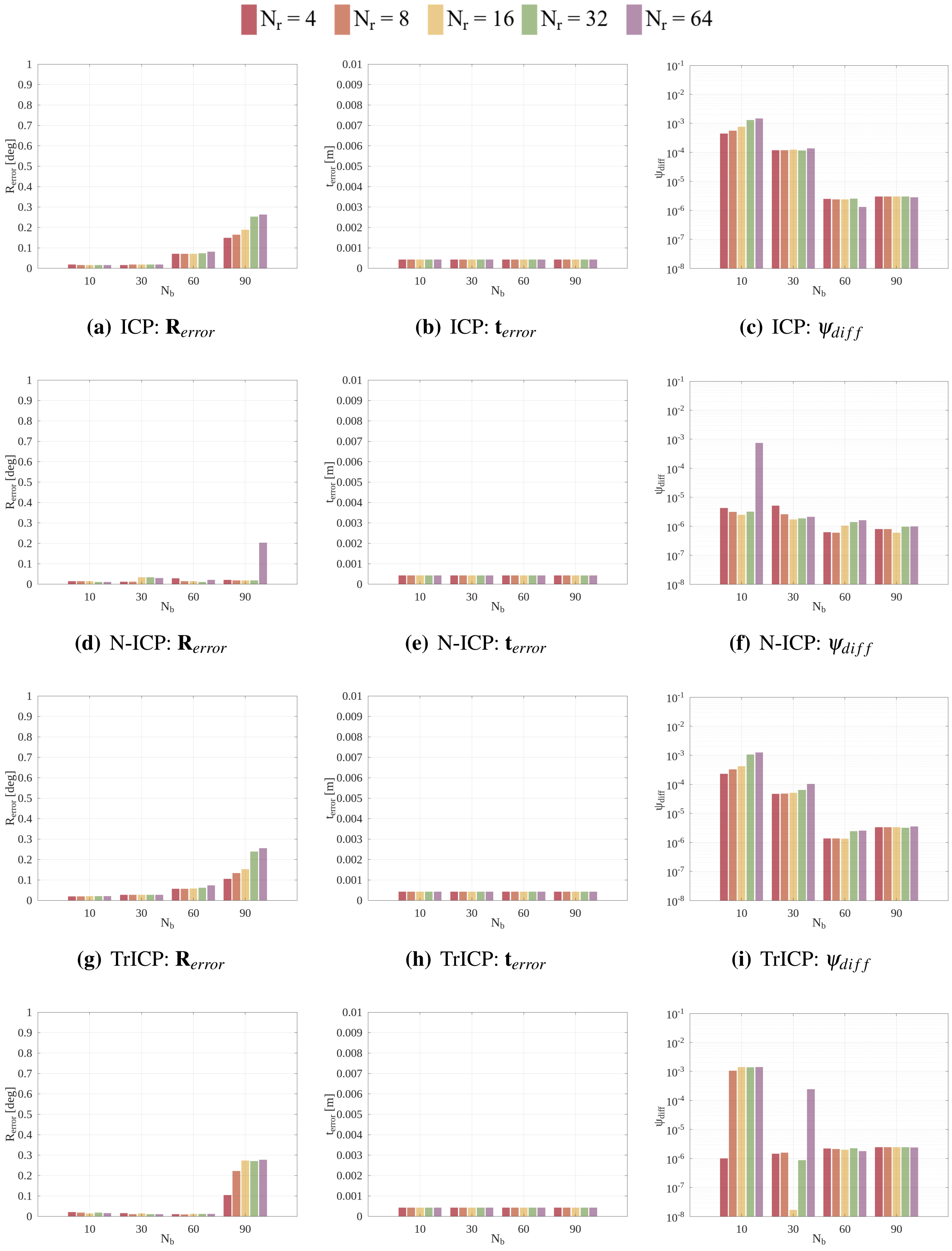
(a) ICP: $\mathbf{R}_{error}$ **(b)** ICP: $\mathbf{t}_{error}$ **(c)** ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$ **(e)** N-ICP: $\mathbf{t}_{error}$ **(f)** N-ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$ **(h)** TrICP: $\mathbf{t}_{error}$ **(i)** TrICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$ **(k)** LM-ICP: $\mathbf{t}_{error}$ **(l)** LM-ICP: ψ_{diff}

Figure 4.1: Quantitative results of *Armadillo*



(a) ICP: $\mathbf{R}_{error}$

(b) ICP: $\mathbf{t}_{error}$

(c) ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$

(e) N-ICP: $\mathbf{t}_{error}$

(f) N-ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$

(h) TrICP: $\mathbf{t}_{error}$

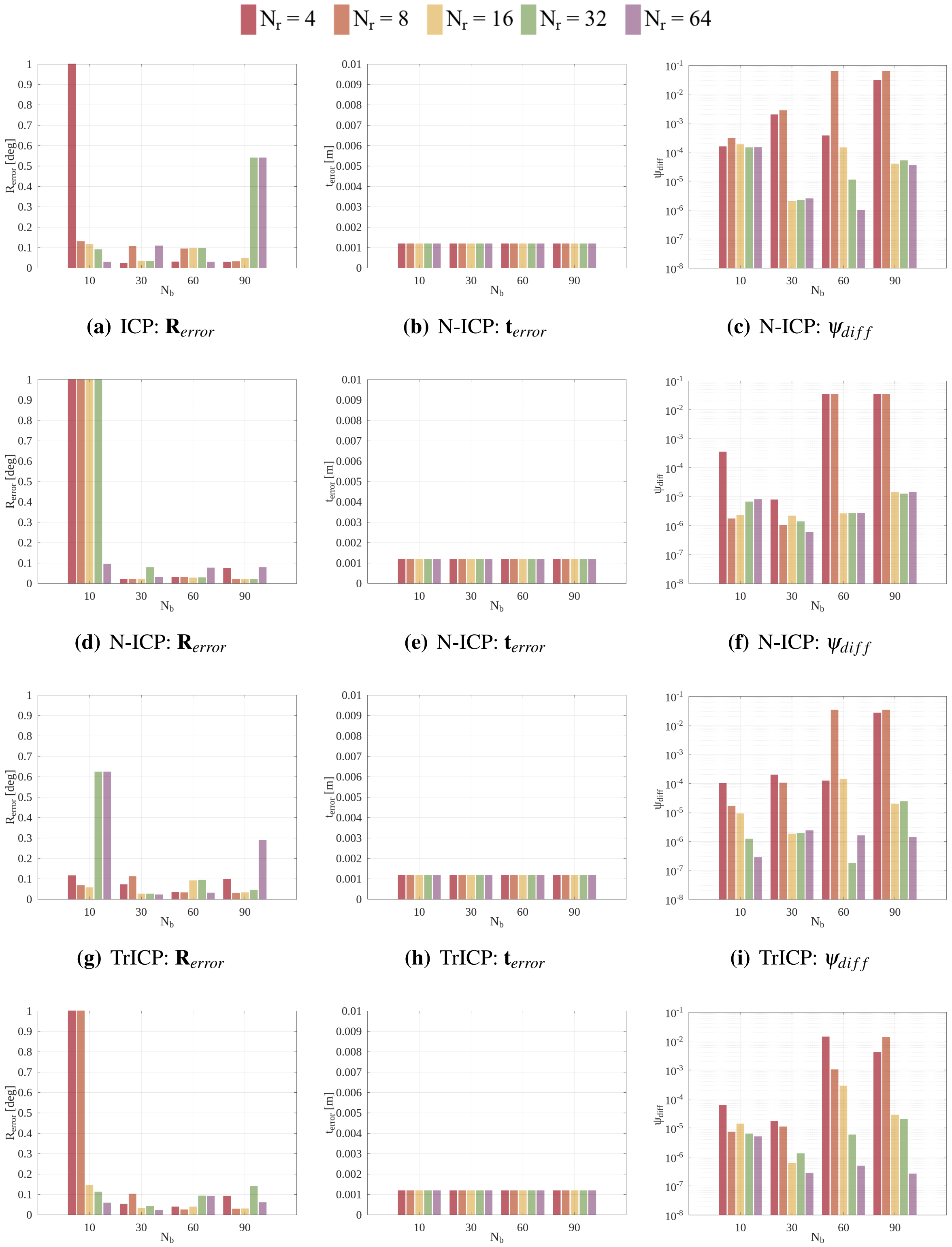
(i) TrICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$

(k) LM-ICP: $\mathbf{t}_{error}$

(l) LM-ICP: ψ_{diff}

Figure 4.2: Quantitative results of *Bunny*



(a) ICP: $\mathbf{R}_{error}$

(b) N-ICP: $\mathbf{t}_{error}$

(c) N-ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$

(e) N-ICP: $\mathbf{t}_{error}$

(f) N-ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$

(h) TrICP: $\mathbf{t}_{error}$

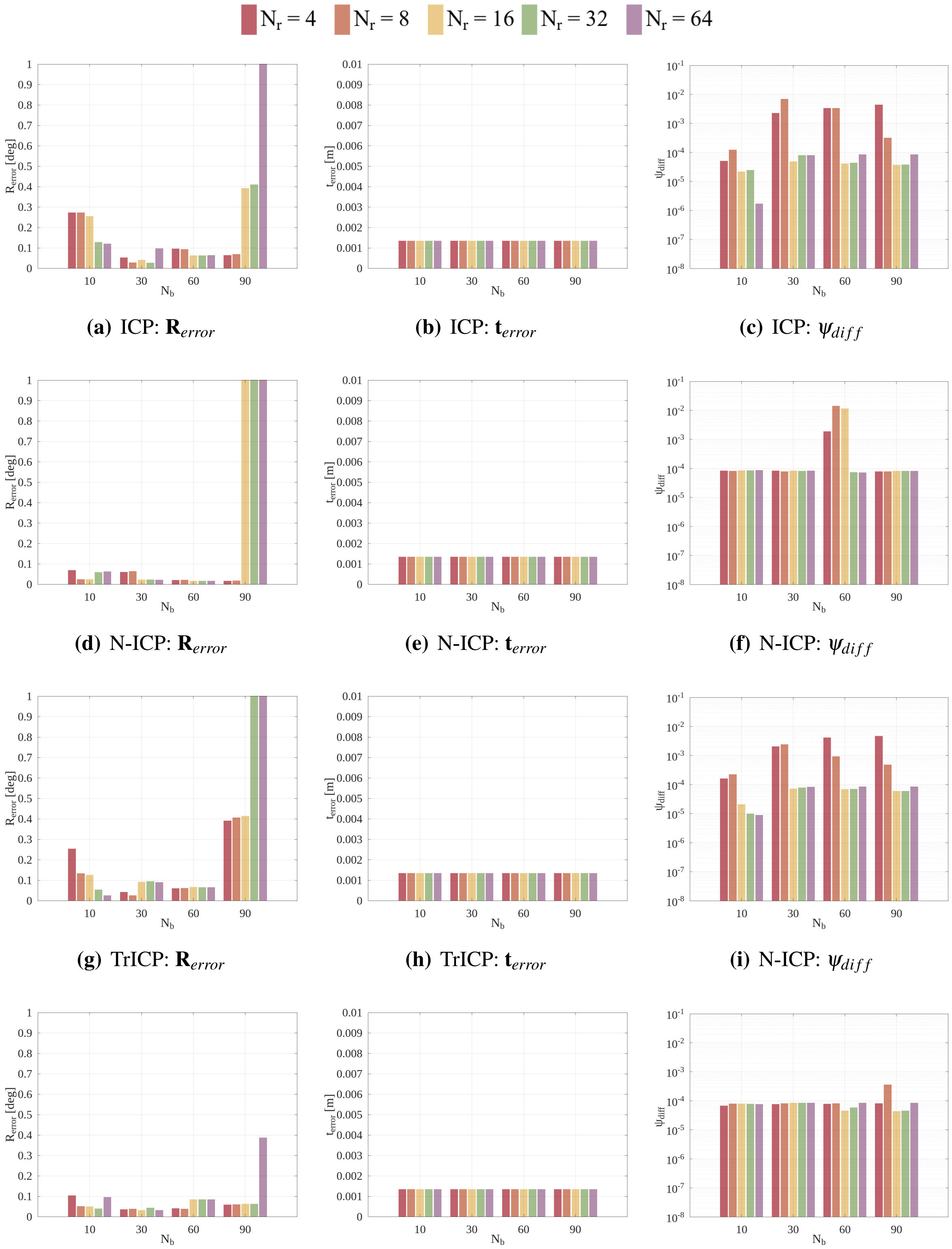
(i) TrICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$

(k) LM-ICP: $\mathbf{t}_{error}$

(l) LM-ICP: ψ_{diff}

Figure 4.3: Quantitative results of *Dragon*



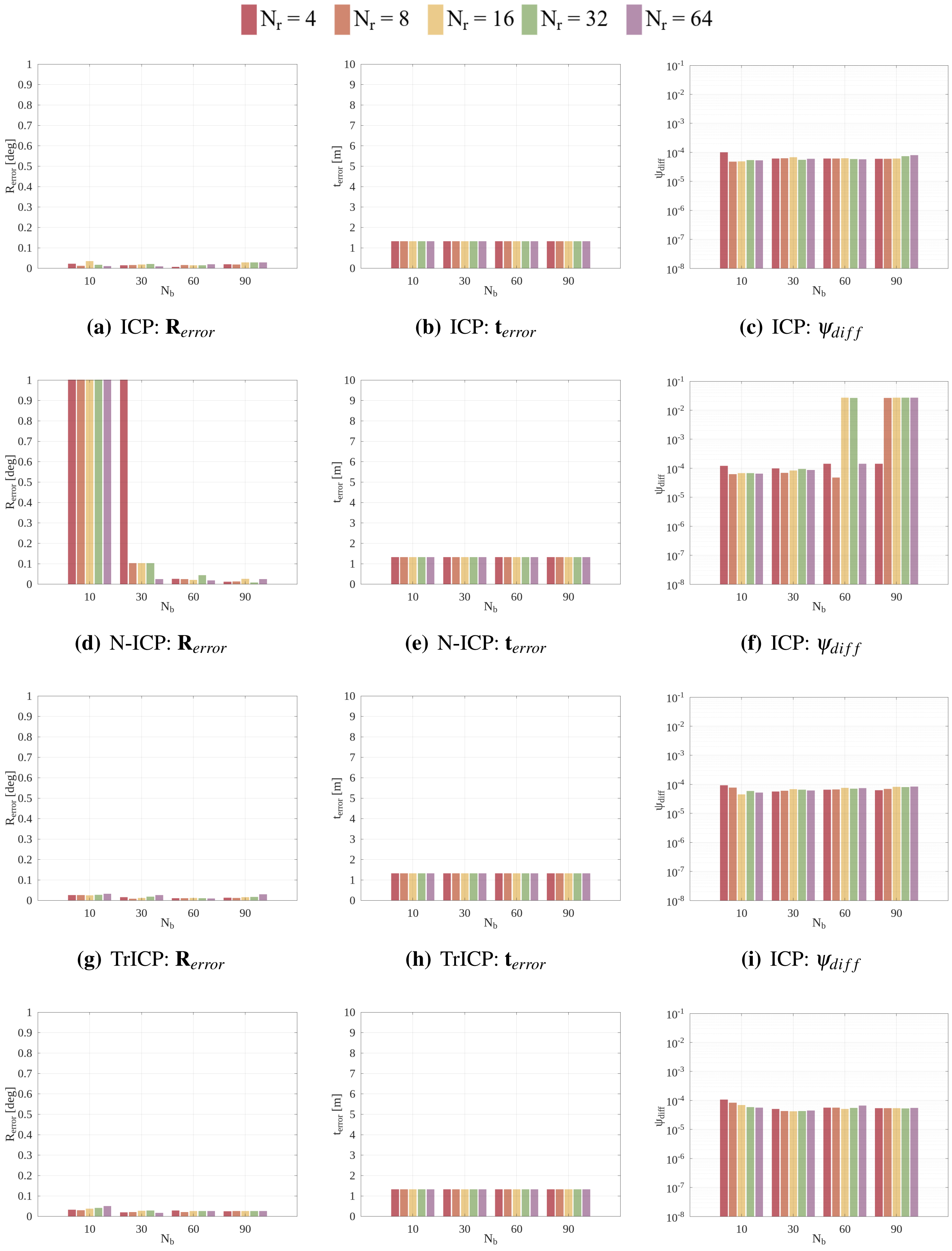
(a) ICP: $\mathbf{R}_{error}$ **(b)** ICP: $\mathbf{t}_{error}$ **(c)** ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$ **(e)** N-ICP: $\mathbf{t}_{error}$ **(f)** N-ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$ **(h)** TrICP: $\mathbf{t}_{error}$ **(i)** N-ICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$ **(k)** LM-ICP: $\mathbf{t}_{error}$ **(l)** N-ICP: ψ_{diff}

Figure 4.4: Quantitative results of *Happy Buddha*



(a) ICP: $\mathbf{R}_{error}$

(b) ICP: $\mathbf{t}_{error}$

(c) ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$

(e) N-ICP: $\mathbf{t}_{error}$

(f) ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$

(h) TrICP: $\mathbf{t}_{error}$

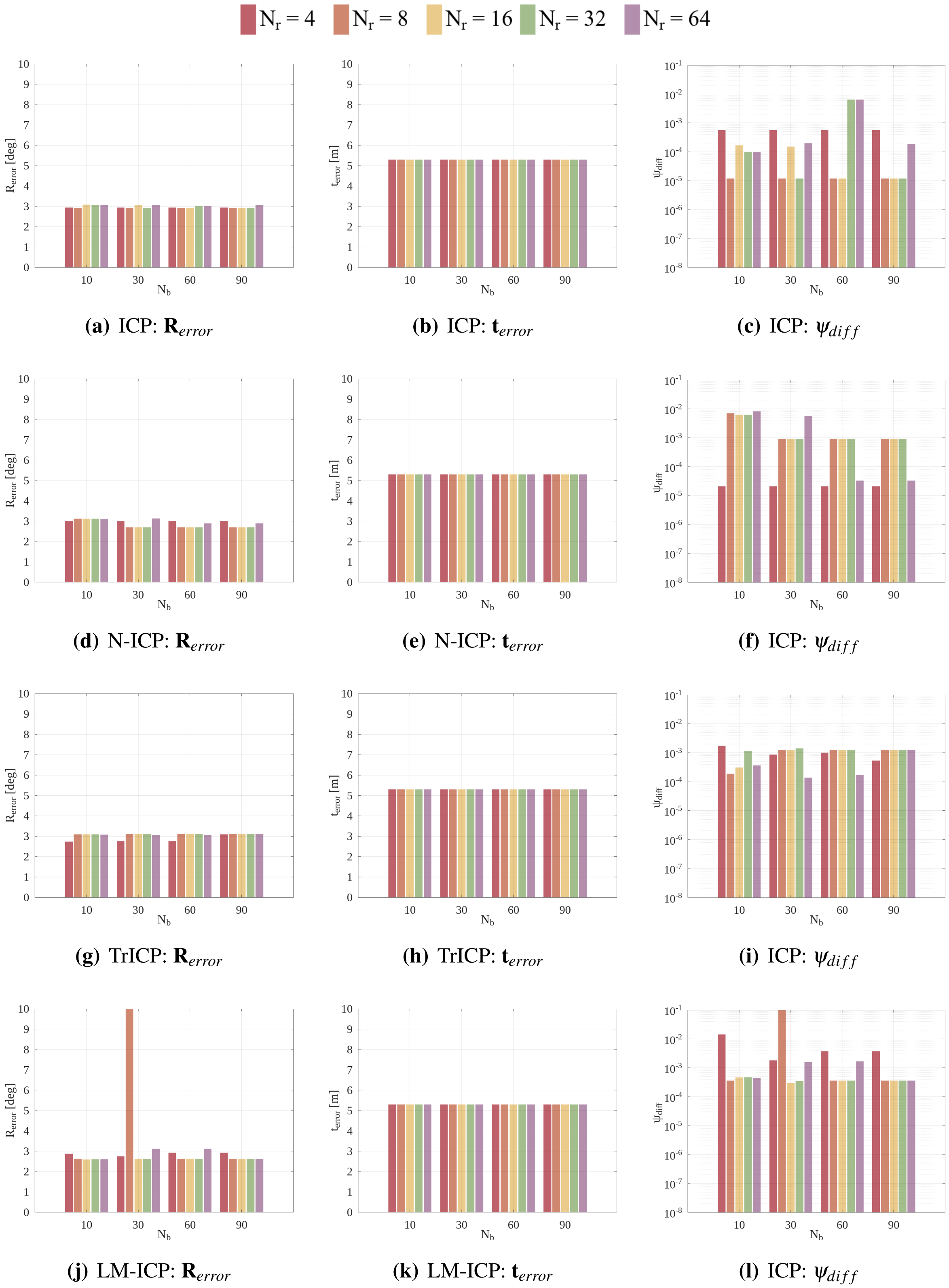
(i) ICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$

(k) LM-ICP: $\mathbf{t}_{error}$

(l) ICP: ψ_{diff}

Figure 4.5: Quantitative results of *Stairs*



(a) ICP: $\mathbf{R}_{error}$

(b) ICP: $\mathbf{t}_{error}$

(c) ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$

(e) N-ICP: $\mathbf{t}_{error}$

(f) ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$

(h) TrICP: $\mathbf{t}_{error}$

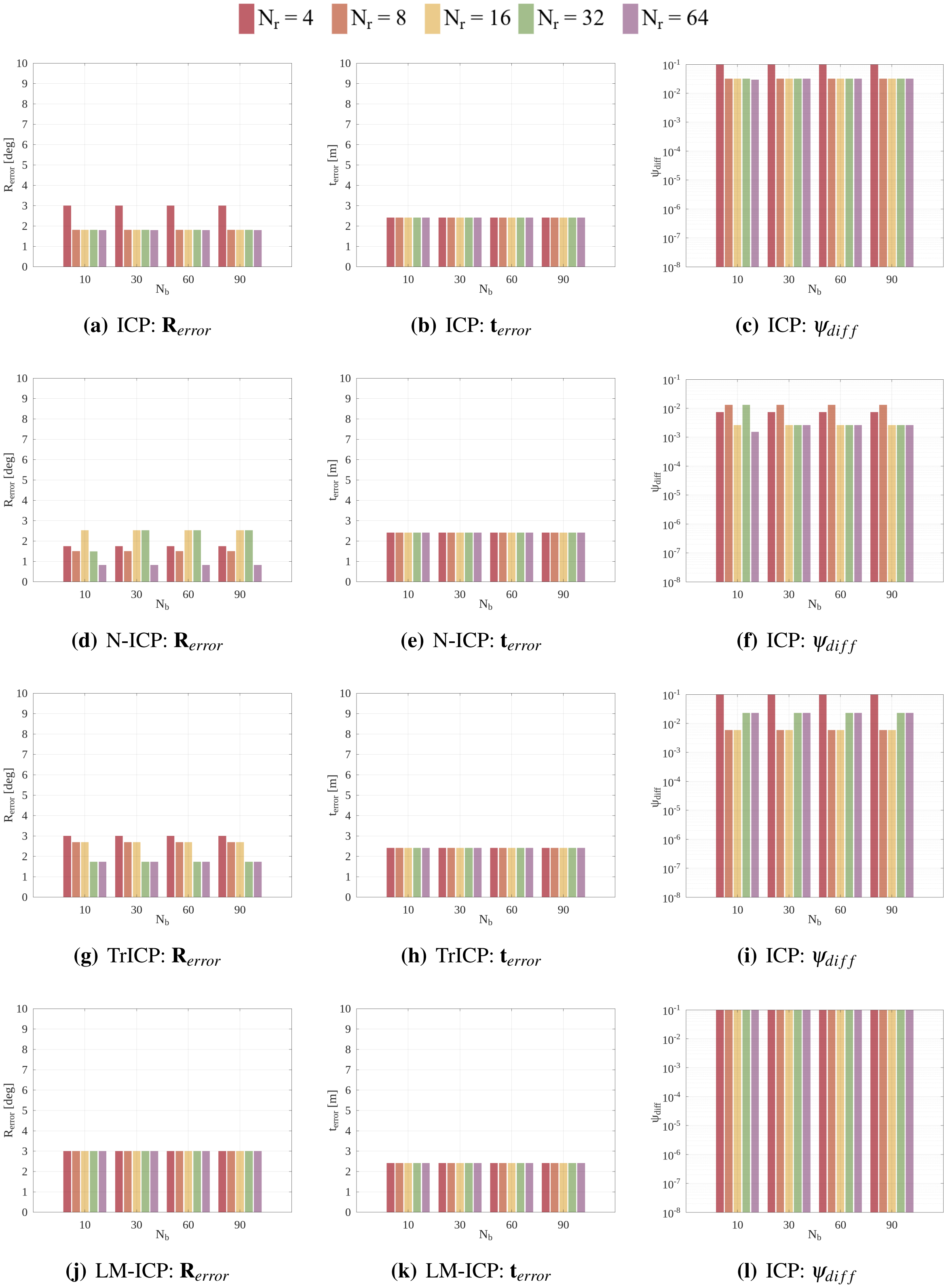
(i) ICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$

(k) LM-ICP: $\mathbf{t}_{error}$

(l) ICP: ψ_{diff}

Figure 4.6: Quantitative results of *At Home*



(a) ICP: $\mathbf{R}_{error}$

(b) ICP: $\mathbf{t}_{error}$

(c) ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$

(e) N-ICP: $\mathbf{t}_{error}$

(f) ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$

(h) TrICP: $\mathbf{t}_{error}$

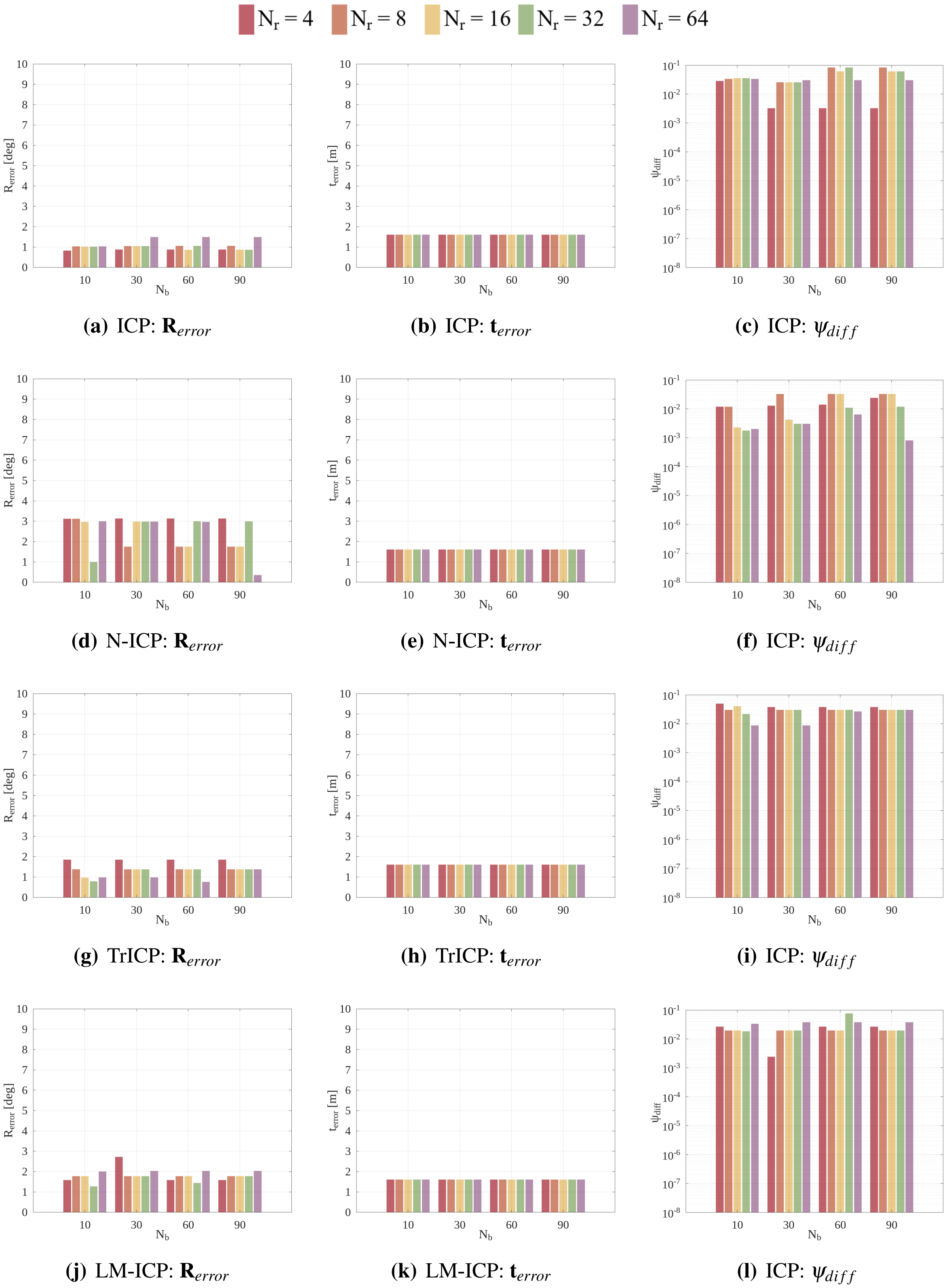
(i) ICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$

(k) LM-ICP: $\mathbf{t}_{error}$

(l) ICP: ψ_{diff}

Figure 4.7: Quantitative results of *Study Room*



(a) ICP: $\mathbf{R}_{error}$ **(b)** ICP: $\mathbf{t}_{error}$ **(c)** ICP: ψ_{diff}

(d) N-ICP: $\mathbf{R}_{error}$ **(e)** N-ICP: $\mathbf{t}_{error}$ **(f)** ICP: ψ_{diff}

(g) TrICP: $\mathbf{R}_{error}$ **(h)** TrICP: $\mathbf{t}_{error}$ **(i)** ICP: ψ_{diff}

(j) LM-ICP: $\mathbf{R}_{error}$ **(k)** LM-ICP: $\mathbf{t}_{error}$ **(l)** ICP: ψ_{diff}

Figure 4.8: Quantitative results of *Maryland Hotel 1*

Remark 4.1: DfVb and RwSVS connotation

- DfVb stands for *Descriptor-free Voting-based*.
- RwSVS stands for *Registration with Supervoxel Segmentation*.

4.3.4 Comparative study

Comparative experiments setup

Since the proposed voting-based approach can vary in the core registration algorithm, and to obtain CRA-specific results we divide this approach into four variants denoted as DfVb: ICP, DfVb: N-ICP, DfVb: TrICP, and DfVb: LM-ICP. Additionally we also added to the comparative study the method introduced in **Chapter 2** denoted as RwSVS.

Although all registration methods introduced in **Section 4.1.2** are SoTA that we could have included in this comparative study, we excluded the approach by Prokop et al.[57] since it requires exhaustive data-specific parameter tuning more difficult to set that the parameters of this work's approaches. Besides, most of the employed datasets do not follow their assumptions and their publicly available code implementation is incomplete which does not allow us to test it. The source code of the methods by Wu et al.[30] and Sun et al.[53] were not publicly released and implementing them by ourselves could be the source of incorrect results. Therefore, these two methods were not included in the comparison as well. The voting-based method [12] from which we borrowed the binning procedure was also not taken into consideration in the comparison because of its extremely long processing time. Last but not least, we do not include to the comparison the deep learning-based method [59, 60] because of their high demand on computational resources and training data.

Considering the before mentioned limitations, we carried out the comparative study only between the four variants of the voting-based method (DfVb), the registration with Supervoxel Segmentation RwSVS, HMRF-ICP, and a Global Registration (GlbReg) pipeline. The parameters of HMRF-ICP were set as described by the authors, and to represent DfVb variants in a general manner we use the median values of each core registration variant with all different combinations of (N_b, N_r). As for the GlbReg pipeline, it consists of computing the Scale Invariant Feature Transform (SIFT) key points [62], their corresponding FPFH descriptors, and estimating the transformation matrix via RANSAC. Its specific parameters for each dataset are in **Table 4.2**.

As mentioned in **Section 4.3.1** only the initial position of the target 3D point clouds from 3DLoMatch were modified to be aligned to the global reference frame. No additional

Table 4.2: Parameters utilized for Global Registration

Dataset	**Parameter**	**Value**	**Dataset**	**Parameter**	**Value**
Armadillo	Rn	5	*Stairs*	Rn	20
	R_{FPFH}	30		R_{FPFH}	50
	$SIFT_{minscale}$	0.001		$SIFT_{minscale}$	0.1
	$SIFT_{min\ contrast}$	0.0005		$SIFT_{min\ contrast}$	0.005
	$RANSAC_{iterations}$	1,000,000		$RANSAC_{iterations}$	1,000,000
Bunny	R_n	20	*At Home*	R_n	20
	R_{FPFH}	40		R_{FPFH}	50
	$SIFT_{min\ scale}$	0.001		$SIFT_{min\ scale}$	0.1
	$SIFT_{min\ contrast}$	0.0005		$SIFT_{min\ contrast}$	0.005
	$RANSAC_{iterations}$	1,000,000		$RANSAC_{iterations}$	1,000,000
Dragon	R_n	20	*Study Room*	R_n	20
	R_{FPFH}	40		R_{FPFH}	50
	$SIFT_{min\ scale}$	0.001		$SIFT_{min\ scale}$	0.1
	$SIFT_{min\ contrast}$	0.0005		$SIFT_{min\ contrast}$	0.005
	$RANSAC_{iterations}$	1,000,000		$RANSAC_{iterations}$	1,000,000
H. Buddha	R_n	15	*M. Hotel 1*	R_n	20
	R_{FPFH}	30		R_{FPFH}	50
	$SIFT_{min\ scale}$	0.001		$SIFT_{min\ scale}$	0.1
	$SIFT_{min\ contrast}$	0.0005		$SIFT_{min\ contrast}$	0.005
	$RANSAC_{iterations}$	1,000,000		$RANSAC_{iterations}$	1,000,000

rotation or translation were pre-applied to the other datasets. Hence, their initial positions are the original rotation and translation differences in reference to the local frame of the target $\mathcal{Q}$ as the global frame.

Comparative results

Figure 4.9 comprehends the quantitative comparison in R_{error}, t_{error}, ψ_{diff}, and processing time between all the evaluated approaches. In these charts each series represents a method, the horizontal axes correspond to the datasets, and the vertical axes to the evaluation measures in logarithmic scale. Depicting $\mathcal{P}$ in red and $\mathcal{Q}$ in blue, **Figure 4.10** shows ground-truth alignment and **Figures 4.9** to **4.14** show the corresponding qualitative results of the approaches.

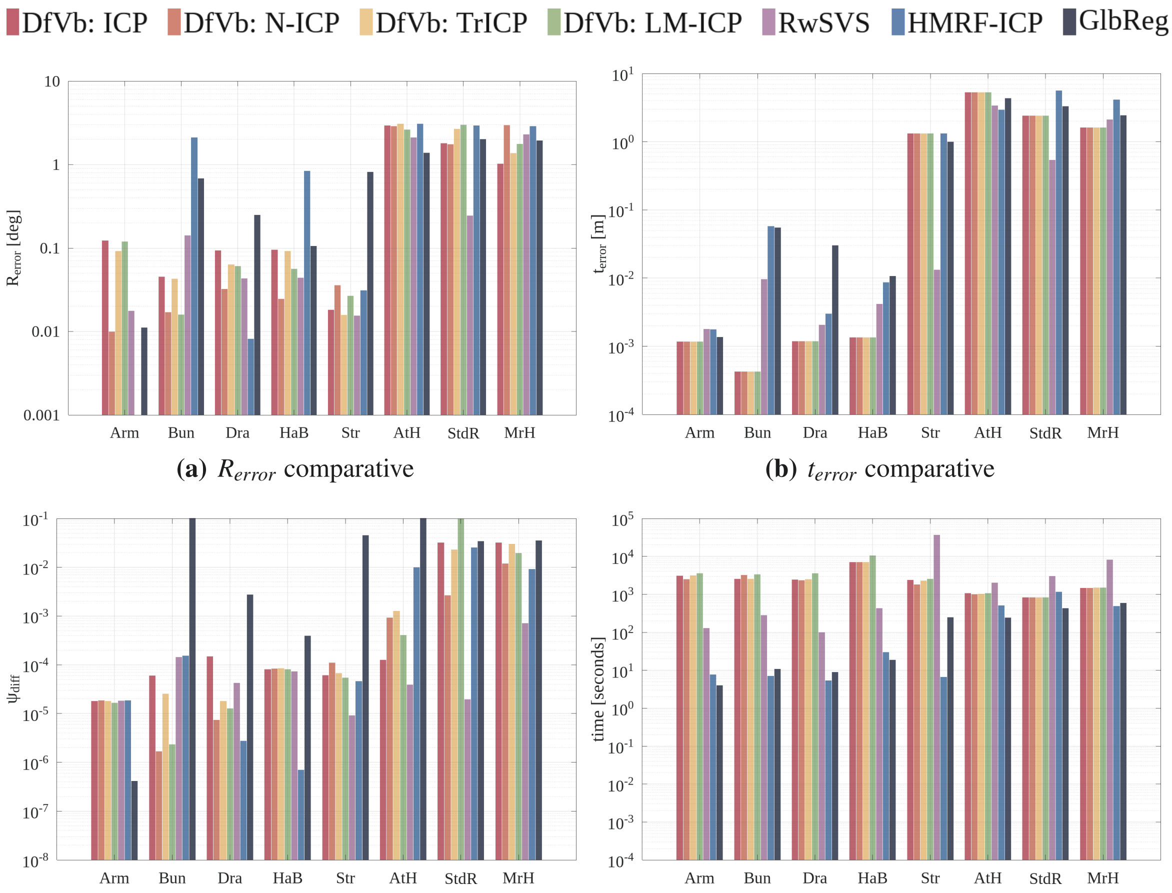
By analyzing **Figure 4.9(a)**, one can tell that HMRF-ICP drastically outperforms in the rotation estimation for *Armadillo* with $R_{error} = 0°$ and *Dragon* with $R_{error} \approx 0.01°$. However, as shown in the translation error comparative of **Figure 4.9(b)**, and ψ_{diff} comparative of **Figure 4.9(c)** it is not better than any of DfVb variants in terms of $\mathbf{t}_{error}$ and compares equally in ψ_{diff}. Still, the improved translation does not describe substantial differences between both methods in these datasets (*cf. Armadillo* and *Dragon* in **Figure 4.11(c)** and **Figure 4.13(c)**). Moreover, HMRF-ICP does not perform well for the rotation in *Bunny* and *Happy Buddha*, for the translation in *Stairs*, and for all metrics in *At Home*, *Study Room*, and *Maryland Hotel 1*.

In *Stairs* RwSVS performs the best with a significantly better $t_{error} \approx 0.01\ m\ (10\ mm)$. In the model-type datasets RwSVS achieves good alignments but converges to a local minima alignment in *Bunny* due to the large spherical surfaces on the front chest of this model (*cf.* **Figure 4.12(e)**). Furthermore, the datasets from 3DLoMatch also turned out to be challenging for this method as it also fails to obtain good alignments for these, but performs better than the other methods in *Study Room* (*cf.* **Figure 4.12(g)**), with a $R_{error} \approx 0.12°$, and a $\psi_{diff} \approx 10^{-5}$.

As expected, GlbReg is the only evaluated method that fails in almost all datasets. It outperforms the proposed methods DfVb and RwSVS only in *Armadillo* (*cf.* **Figure 4.14(a)**), with a $R_{error} \approx 0.01°$ and $\psi_{diff} < 10^{-4}$. However, it is also in *Armadillo* that all the evaluated methods obtain a good solution, which suggests that this dataset is not necessarily challenging despite having a low overlap. It is important to remark that better results could be obtained with different parameters for GlbReg, but it would require exhaustive trial-and-error tests that may not deliver significant improvements. As shown in the previous ablation study DfVb also requires of parameter tuning to deliver the best possible results, but, unlike GlbReg, the ruling parameters are easier to interpret as these are independent from the scale or size.

Although quantitative results of DfVb variants are the median values, it is clear that DfVb can estimate nearly optimal **T**s in all model-type datasets. Particularly, DfVb: N-ICP generally performs best as these have more detailed shapes that help to take advantage of the point normals. Nonetheless, it does not occur in *Stairs* because the dataset has many non-dense and large plane surfaces that can incorrectly align when divided into subsets. Furthermore and in general, DfVb fails in all scene type datasets, as clearly shown in Figure 4.11 and indicated by high $\mathbf{R}_{error}$ between 1° and 10°, $\mathbf{t}_{error} > 10m$, and $\psi_{diff} > 10^{-4}$ in the quantitative results of Figure 4.9.

Regarding the processing time, is clearly shown in **Figure 4.9(d)** that DfVb and RwSVS are significantly slower than HMRF-ICP and GlbReg. In one hand, RwSVS takes from 100 to $1,000$ seconds to converge in the model-type datasets and from $1,000$ to $100,000$ seconds, or about 16 minutes to 28 hours, in scene type datasets; DfVb variants take a median processing time between $1,000$ and $10,000$ seconds, or about 16 minutes to 3 hours in all datasets. In the latter case, DfVb converges faster to a solution in the datasets from 3DLoMatch than in the model-type datasets, but it is result of the down-sampling. On the other hand HMRF-ICP



(a) R_{error} comparative

(b) t_{error} comparative

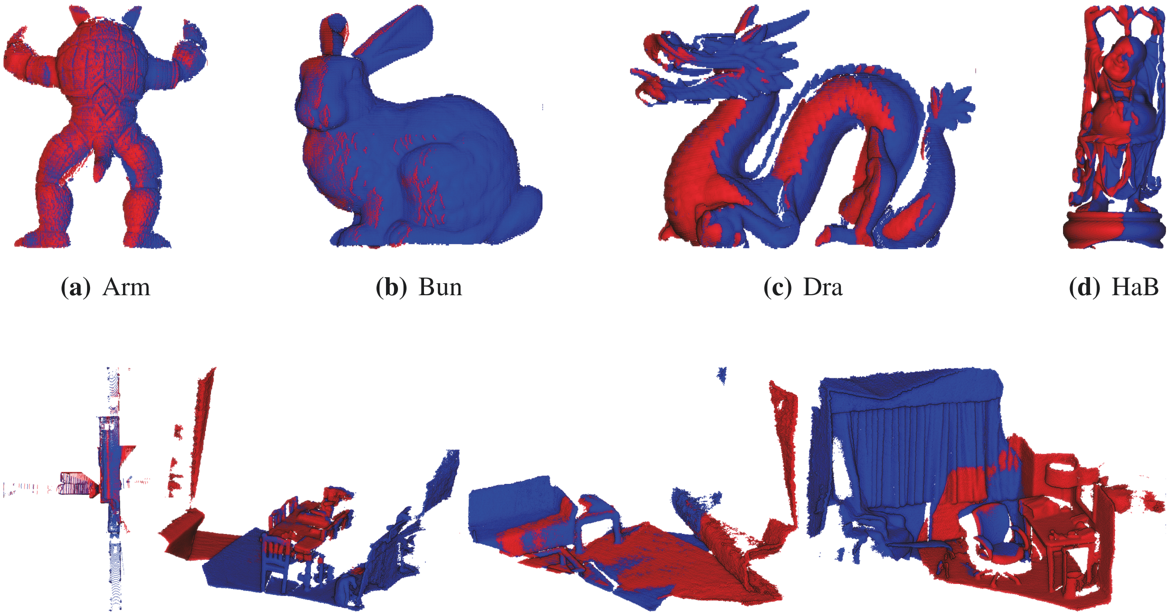
(c) ψ_{diff} comparative

(d) Processing time comparative

Figure 4.9: Comparative study quantitative results

and GlbReg are equally fast in most cases, converging to a solution within 1 and 1,000 seconds in all tested datasets.

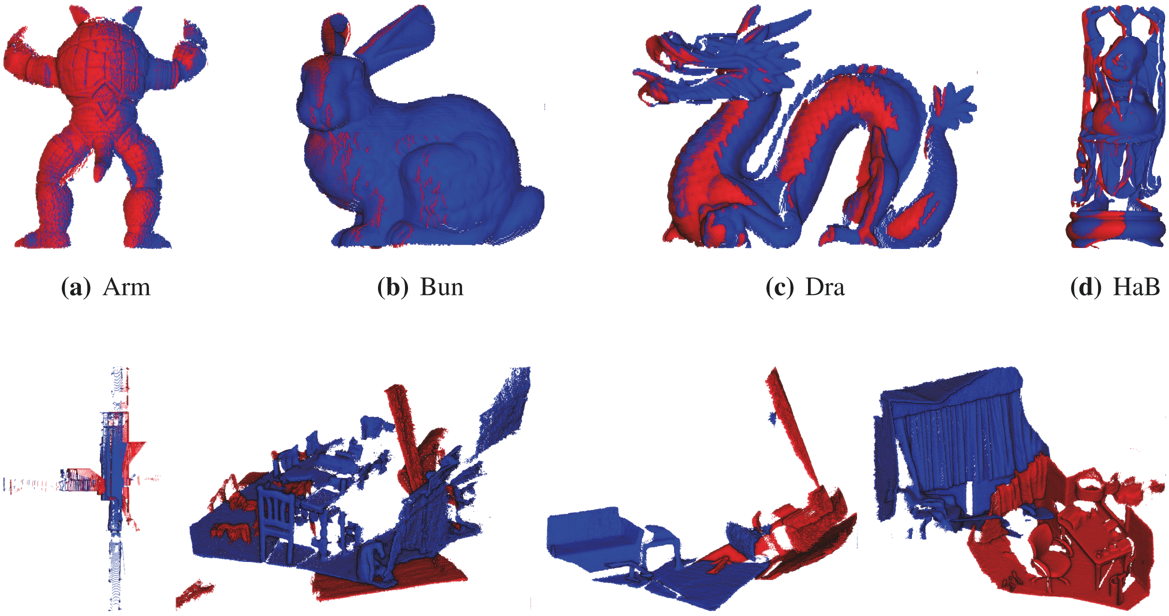
The time complexity of DfVb, $O\left(\left(R_{seed}^{max} - R_{seed}^{min}\right) \cdot SS_{\mathcal{P}} \log\left(N_r\right)\right)$, or in theory a log-linear time complexity $O\left(n \ \log n\right)$, reveals the reason for its sub-optimal processing time performance even with down-sampled 3D point clouds. In practice, its time complexity is ruled by the size of the input 3D point clouds, since it fixes the range of values for R_{seed} from a maximum R_{seed}^{max} to a minimum R_{seed}^{min}. Thus, the bigger are the point clouds, the longer the processing time will be (noticeable in the scene type datasets). Additionally, the number of rotations N_r also directly influences the bottleneck of the method, as more rotations to consider means more transformations to process. For its part, RwSVS's time complexity $O\left(\left(R_{seed}^{max} - R_{seed}^{min}\right) \cdot SS_{\mathcal{P}} \cdot SS_{\mathcal{Q}}\right)$, in theory $O\left(n\right)$, is significantly simpler but, in practice still depends on the size of the input 3D point clouds.



(a) Arm **(b)** Bun **(c)** Dra **(d)** HaB

(e) Str **(f)** AtH **(g)** StdR **(h)** MrH

Figure 4.10: Ground-truth alignments



(a) Arm **(b)** Bun **(c)** Dra **(d)** HaB

(e) Str **(f)** AtH **(g)** StdR **(h)** MrH

Figure 4.11: DfVb qualitative results

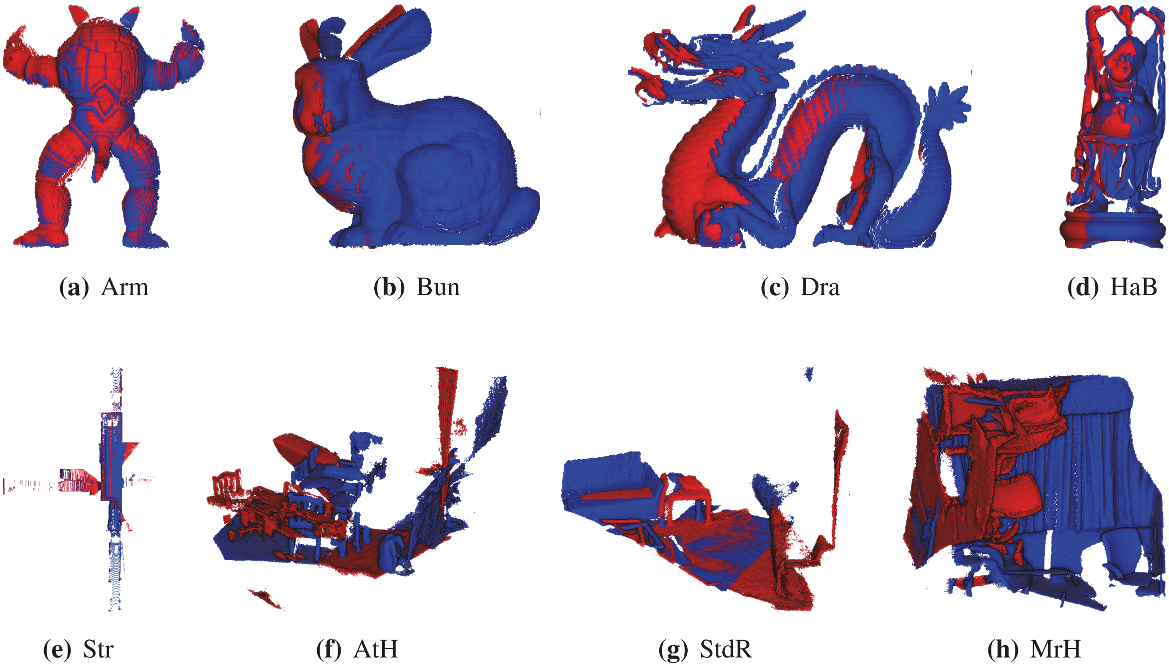

(a) Arm **(b)** Bun **(c)** Dra **(d)** HaB **(e)** Str **(f)** AtH **(g)** StdR **(h)** MrH

Figure 4.12: RwSVS qualitative results

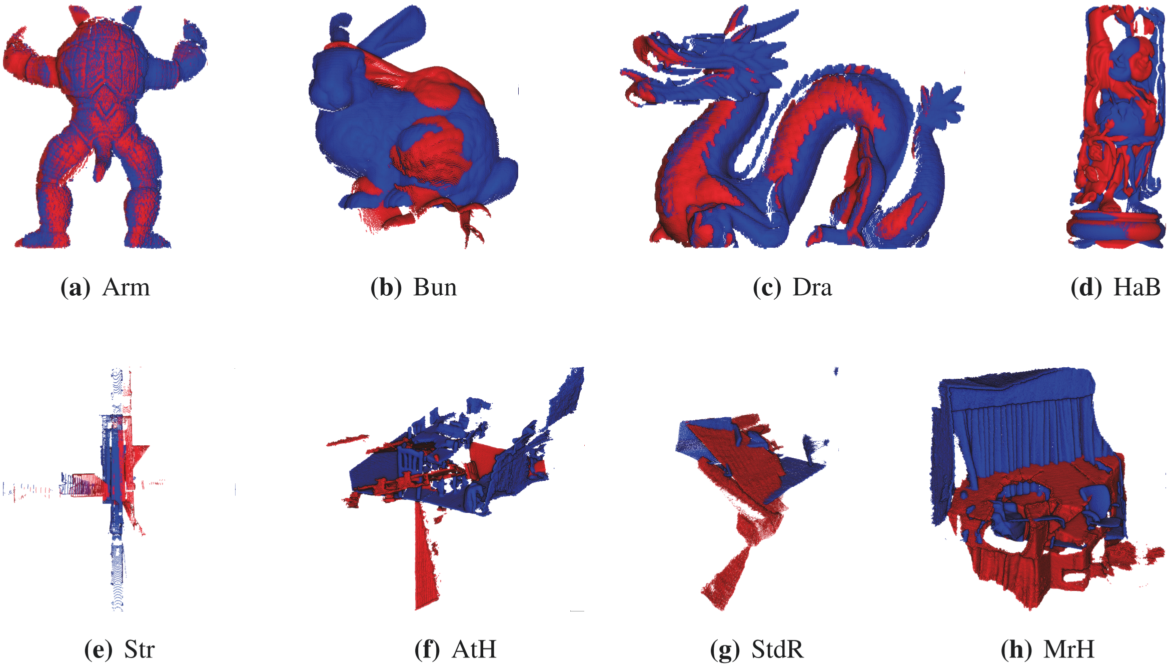
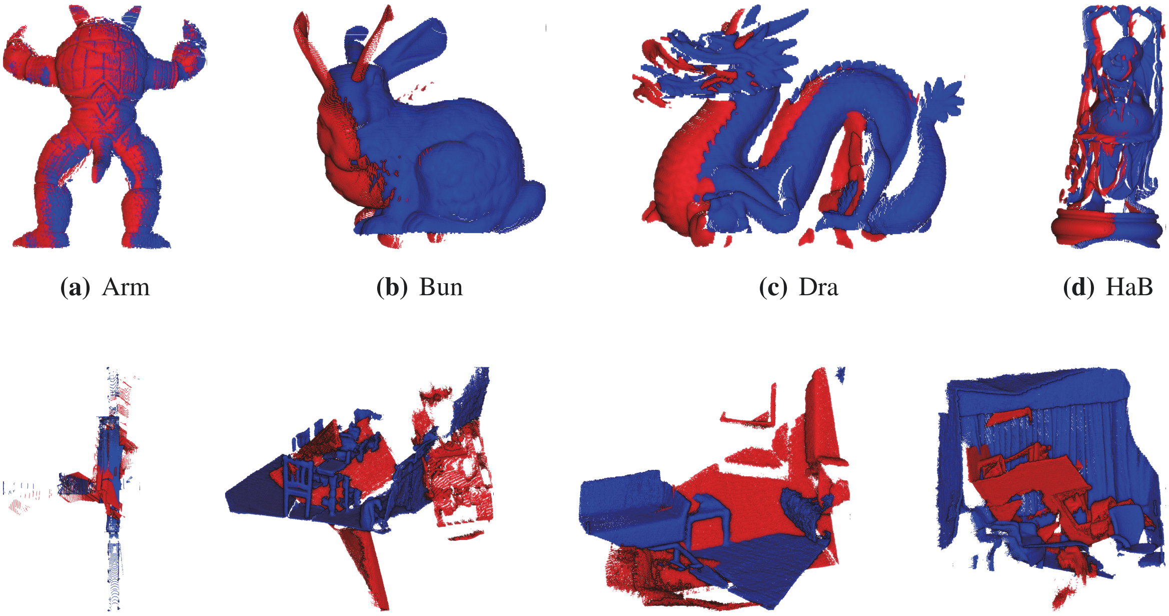

(a) Arm **(b)** Bun **(c)** Dra **(d)** HaB **(e)** Str **(f)** AtH **(g)** StdR **(h)** MrH

Figure 4.13: HMRF-ICP qualitative results


(a) Arm **(b)** Bun **(c)** Dra **(d)** HaB

(e) Str **(f)** AtH **(g)** StdR **(h)** MrH

Figure 4.14: GlbReg qualitative results

4.4 Usage guideline

This section establishes a guideline to adopt the proposed descriptor-free voting-based registration approach as the results of the ablation study do not present a clear pattern for an exemplary parameter configuration. The recommendations are the following:

1. Choose the algorithm for the core registration based on the shape of the 3D point clouds. However, N-ICP and LM-ICP are generally good starting points.

2. Set $30 \leq N_b \leq 60$ and $N_r \geq 16$ to achieve the best possible alignment, as these tend to perform good with any CRA as core registration.

3. To define the number of rotations N_r, consider the end application in which the methods is utilized as this parameter directly affects the processing time.

4. This registration method is more suitable for offline applications to process model-type datasets.

4.5 Conclusion

We conducted an ablation and comparative study to attempt to understand the behavior of the proposed voting-based registration approach under its ruling parameters, as well as a comparative study to evaluate its performance against the first approach proposed in this work, an SoTA method, and a Global Registration pipeline.

The ablation study comprehended an extensive experimentation that showcased the behavior broadly. Although its results did not clearly define the fittest parameter configuration, these allowed us to define recommendations about how to set up the parameters and use the approach to obtain the best possible results.

Results from the comparative study showed that the proposed approaches achieve better results than a Global Registration pipeline and similar results to HMRF-ICP in model-type datasets. Nevertheless, the voting-based approach might not be adequate for applications that utilize large datasets from environments or require real-time performance.

Moreover, the datasets from 3DLoMatch showed to be truly challenging for all the evaluated methods, but we should have in mind that these were down-sampled through voxel down-sampling which alters the density of the point clouds and organizes the points evenly in the 3D space. In DfVb and RwSVS down sampling affects SVS and the inlier threshold that defines the correspondences $\mathcal{C}$ - increasing the likelihood to retrieve false positive transformations. Which tells us that RwSVS and DfVb are sensitive to the point density.

Chapter 5

Evaluation of Euler angles in voting-based registration

The present chapter comprehends a parallel registration approach based on the proposed voting-based approach but that replaces the representation of the rotation for the Euler angles, which was introduced in "An Evaluation of Order of Significance of Euler Angles in Voting-based Registration of 3D Point Clouds" in Proceedings of the 49th Annual Conference of the Institute of Image Electronics Engineers of Japan.

5.1 Background and related theory

5.1.1 Motivation to replace the rotation representation

As described in **Section 3.2.1**, the proposed voting-based approach utilizes the quaternion as the rotation representation to map into the parameter space, reducing the number of parameters only for the rotation from nine to four. The quaternion representation is widely employed in 3D not only because it has a smaller computational footprint, but also because it allows to freely rotate an object in the 3D space without having to worry about a phenomenon known as gimbal lock. It means loosing one degree of freedom in the 3D space when the two of three axes are positioned into a parallel configuration, locking into rotating in a 2D space.

Nevertheless, the quaternion representation is difficult to understand when trying to comprehend the effects of a rotation on an object. Thus, when the main objective is to observe the effects of specific rotations simpler and straightforward representation are preferred, such as the *Euler angles* (EA). Under the proposed voting-based registration method, replacing the quaternion by the EA not only allows us to clearly understand and visualize the transformation that takes place to align the 3D point clouds correctly, but also reduce the dimension of the accumulator from seven to six. This is because the a rotation represented by the EA is composed by only three parameters, three rotation angles around the primary axes $(\theta_x,\ \theta_y,\ \theta_z)$. Hence, we can use a six-dimensional transformation vector representation $\mathbf{T}_{6D} = \left\{\theta_x,\ \theta_y,\ \theta_z,\ t_x,\ t_y,\ t_z\right\}$ along with a smaller accumulator $A\left(\mathbf{T}_{6D}\right)$. However, the order in which the EA are applied has an effect on the final positioning. Considering this possible modification to the proposed voting-based method, the question to answer is:

- *What is the effect of using the Euler angles in a voting-based registration approach?*

5.1.2 Rotation factorization

The EA, as mentioned before, represent angles to rotate about the corresponding axis. These individual rotations about the coordinate axes are

$$\begin{aligned}\mathbf{R}(\theta_x) &= \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x & \cos\theta_x\end{bmatrix}\\ \mathbf{R}(\theta_y) &= \begin{bmatrix}\cos\theta_y & 0 & \sin\theta_y\\ 0 & 1 & 0\\ -\sin\theta_y & 0 & \cos\theta_y\end{bmatrix}.\\ \mathbf{R}(\theta_z) &= \begin{bmatrix}\cos\theta_z & -\sin\theta_z & 0\\ \sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{bmatrix}\end{aligned} \tag{5.1}$$

Where $\theta_x/\theta_y/\theta_z > 0$ indicate counterclockwise rotations in their correspondent zero-leveled planes.

A rotation $\mathbf{R}$ can be factorized as the product of rotations about the coordinates axes, and it is possible to retrieve the rotation angles from it considering the factoring order. There are six factoring orders: XYZ, XZY, YXZ, YZX, ZXY, and ZYX. If, for example, we want the factoring order XYZ, it means that the rotation is composed by $\mathbf{R} = \mathbf{R}_x\left(\theta_x\right)\mathbf{R}_y\left(\theta_y\right)\mathbf{R}_z\left(\theta_z\right)$. Therefore, by setting the following notation $c_a = \cos\theta_a$, $s_a = \sin\theta_a$, and

$$\mathbf{R} = \begin{bmatrix}r_{00} & r_{01} & r_{02}\\ r_{10} & r_{11} & r_{12}\\ r_{20} & r_{21} & r_{22}\end{bmatrix}, \tag{5.2}$$

along with the individual rotations we can formally define the different factoring orders of **R** and estimate the EA [63].

XYZ factoring order

$$\mathbf{R}(\theta_x)\,\mathbf{R}(\theta_y)\,\mathbf{R}(\theta_z) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z & -c_y c_z & s_y \\ c_z s_x s_y + c_x s_z & c_x c_z - s_x s_y s_z & -c_y s_x \\ -c_x c_z s_y + s_x s_z & c_z s_x + c_x s_y s_z & c_x c_y \end{bmatrix} \tag{5.3}$$

The most straightforward equivalence is $r_{02} = s_y$, then $\theta_y = \arcsin(r_{02})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_y \in \left(-\frac{\pi}{2},\, \frac{\pi}{2}\right)$, then $c_y \neq 0$ and $c_y(s_x,\, c_x) = (-r_{12},\, r_{22})$.

$$\therefore \quad \theta_x = \arctan\left(\frac{-r_{12}}{r_{22}}\right), \quad \theta_y = \arcsin(r_{02}), \quad \theta_z = \arctan\left(\frac{-r_{01}}{r_{00}}\right). \tag{5.4}$$

2. When $\theta_y = \frac{\pi}{2}$, then $s_y = 1,\ c_y = 0$ and

$$\begin{bmatrix} r_{10} & r_{11} \\ r_{20} & r_{21} \end{bmatrix} = \begin{bmatrix} c_z s_x + c_x s_z & c_x c_z - s_x s_z \\ -c_x c_z + s_x s_z & c_z s_x + c_x s_z \end{bmatrix} = \begin{bmatrix} \sin(\theta_z + \theta_x) & \cos(\theta_z + \theta_x) \\ -\cos(\theta_z + \theta_x) & \sin(\theta_z + \theta_x) \end{bmatrix}. \tag{5.5}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_y = \frac{\pi}{2}, \quad \theta_z + \theta_x = \arctan\left(\frac{r_{10}}{r_{11}}\right). \tag{5.6}$$

3. When $\theta_y = -\frac{\pi}{2}$, then $s_y = -1,\ c_y = 0$ and

$$\begin{bmatrix} r_{10} & r_{11} \\ r_{20} & r_{21} \end{bmatrix} = \begin{bmatrix} -c_z s_x + c_x s_z & c_x c_z + s_x s_z \\ c_x c_z + s_x s_z & c_z s_x - c_x s_z \end{bmatrix} = \begin{bmatrix} \sin(\theta_z - \theta_x) & \cos(\theta_z - \theta_x) \\ \cos(\theta_z - \theta_x) & -\sin(\theta_z - \theta_x) \end{bmatrix}. \tag{5.7}$$

There is only one degree of freedom and factorization is not unique.

$$\therefore \quad \theta_y = -\frac{\pi}{2}, \quad \theta_z - \theta_x = \arctan\left(\frac{r_{10}}{r_{11}}\right). \tag{5.8}$$

XZY factoring order

$$\mathbf{R}(\theta_x)\,\mathbf{R}(\theta_z)\,\mathbf{R}(\theta_y) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z & -s_z & c_z s_y \\ s_x s_y + c_x c_y s_z & c_x c_z & -c_y s_x + c_x s_y s_z \\ -c_x s_y + c_y s_x s_z & c_z s_x & c_x c_y + s_x s_y s_z \end{bmatrix} \tag{5.9}$$

The most straightforward equivalence is $r_{01} = -s_x$, then $\theta_z = \arcsin(-r_{01})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_z \in \left(-\frac{\pi}{2},\, \frac{\pi}{2}\right)$, then $c_z \neq 0$ and $c_z(s_y,\, c_y) = (r_{02},\, r_{00})$.

$$\therefore \quad \theta_x = \arctan\left(\frac{r_{21}}{r_{11}}\right), \quad \theta_y = \arctan\left(\frac{r_{01}}{r_{00}}\right), \quad \theta_z = \arcsin(-r_{01}). \qquad 5.10$$

2. When $\theta_z = \frac{\pi}{2}$, then $s_z = 1,\ c_z = 0$ and

$$\begin{bmatrix} r_{10} & r_{12} \\ r_{20} & r_{22} \end{bmatrix} = \begin{bmatrix} s_x s_y + c_x c_y & -c_y s_x + c_x s_y \\ -c_x s_y + c_y s_x & c_x c_y + s_x s_y \end{bmatrix} = \begin{bmatrix} \cos(\theta_y - \theta_x) & \sin(\theta_y - \theta_x) \\ -\sin(\theta_y - \theta_x) & \cos(\theta_y - \theta_x) \end{bmatrix}. \qquad 5.11$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_z = \frac{\pi}{2}, \quad \theta_y - \theta_x = \arctan\left(\frac{-r_{20}}{r_{22}}\right). \qquad 5.12$$

3. When $\theta_z = -\frac{\pi}{2}$, then $s_z = -1,\ c_z = 0$ and

$$\begin{bmatrix} r_{10} & r_{12} \\ r_{20} & r_{22} \end{bmatrix} = \begin{bmatrix} s_x s_y - c_x c_y & -c_y s_x - c_x s_y \\ -c_x s_y - c_y s_x & c_x c_y - s_x s_y \end{bmatrix} = \begin{bmatrix} -\cos(\theta_y + \theta_x) & -\sin(\theta_y + \theta_x) \\ -\sin(\theta_y + \theta_x) & \cos(\theta_y + \theta_x) \end{bmatrix}. \qquad 5.13$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_z = -\frac{\pi}{2}, \quad \theta_y + \theta_x = \arctan\left(\frac{-r_{20}}{r_{22}}\right). \qquad 5.14$$

YXZ factoring order

$$\mathbf{R}(\theta_y)\,\mathbf{R}(\theta_x)\,\mathbf{R}(\theta_z) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z + s_x s_y s_z & c_z s_x s_y - c_y s_z & c_x s_y \\ c_x s_z & c_x c_z & -s_x \\ -c_z s_y + c_y s_x s_z & c_y c_z s_x + s_y s_z & c_x c_y \end{bmatrix} \qquad 5.15$$

The most straightforward equivalence is $r_{12} = -s_x$, then $\theta_x = \arcsin(-r_{12})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_x \in \left(-\frac{\pi}{2},\, \frac{\pi}{2}\right)$, then $c_x \neq 0$ and $c_x(s_y,\, c_y) = (r_{02},\, r_{22})$.

$$\therefore \quad \theta_x = \arcsin(-r_{12}), \quad \theta_y = \arctan\left(\frac{r_{02}}{r_{22}}\right), \quad \theta_z = \arctan\left(\frac{r_{10}}{r_{11}}\right). \qquad 5.16$$

2. When $\theta_x = \frac{\pi}{2}$, then $s_x = 1,\ c_x = 0$ and

$$\begin{bmatrix} r_{00} & r_{01} \\ r_{20} & r_{21} \end{bmatrix} = \begin{bmatrix} c_y c_z + s_y s_z & c_z s_y - c_y s_z \\ -c_z s_y + c_y s_z & c_y c_z + s_y s_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_z - \theta_y) & -\sin(\theta_z - \theta_y) \\ \sin(\theta_z - \theta_y) & \cos(\theta_z - \theta_y) \end{bmatrix}. \quad 5.17$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_x = \frac{\pi}{2}, \quad \theta_z - \theta_y = \arctan\left(\frac{-r_{01}}{r_{00}}\right). \quad 5.18$$

3. When $\theta_x = -\frac{\pi}{2}$, then $s_x = -1,\ c_x = 0$ and

$$\begin{bmatrix} r_{00} & r_{01} \\ r_{20} & r_{21} \end{bmatrix} = \begin{bmatrix} c_y c_z - s_y s_z & -c_z s_y - c_y s_z \\ -c_z s_y - c_y s_z & -c_y c_z + s_y s_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_z + \theta_y) & -\sin(\theta_z + \theta_y) \\ -\sin(\theta_z + \theta_y) & -\cos(\theta_z + \theta_y) \end{bmatrix}. \quad 5.19$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_x = -\frac{\pi}{2}, \quad \theta_z + \theta_y = \arctan\left(\frac{-r_{01}}{r_{00}}\right). \quad 5.20$$

YZX factoring order

$$\mathbf{R}(\theta_y)\mathbf{R}(\theta_z)\mathbf{R}(\theta_x) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z & s_x s_y - c_x c_y s_z & c_x s_y + c_y s_x s_z \\ s_z & c_x c_z & -c_z s_x \\ -c_z s_y & c_y s_x + c_x s_y s_z & c_x c_y - s_x s_y s_z \end{bmatrix} \quad 5.21$$

The most straightforward equivalence is $r_{10} = s_z$, then $\theta_z = \arcsin(r_{10})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_z \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right)$, then $c_z \neq 0$ and $c_z(s_x,\ c_x) = (-r_{12},\ r_{00})$.

$$\therefore \quad \theta_x = \arctan\left(\frac{-r_{12}}{r_{11}}\right), \quad \theta_y = \arctan\left(\frac{-r_{20}}{r_{00}}\right), \quad \theta_z = \arcsin(r_{10}). \quad 5.22$$

2. When $\theta_z = \frac{\pi}{2}$, then $s_z = 1,\ c_z = 0$ and

$$\begin{bmatrix} r_{01} & r_{02} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} s_x s_y - c_x c_y & c_x s_y + c_y s_x \\ c_y s_x + c_x s_y & c_x c_y - s_x s_y \end{bmatrix} = \begin{bmatrix} -\cos(\theta_x + \theta_y) & \sin(\theta_x + \theta_y) \\ \sin(\theta_x + \theta_y) & \cos(\theta_x + \theta_y) \end{bmatrix}. \quad 5.23$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_z = \frac{\pi}{2}, \quad \theta_x + \theta_y = \arctan\left(\frac{r_{21}}{r_{22}}\right). \tag{5.24}$$

3. When $\theta_z = -\frac{\pi}{2}$, then $s_z = -1,\ c_z = 0$ and

$$\begin{bmatrix} r_{01} & r_{02} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} s_x s_y + c_x c_y & c_x s_y - c_y s_x \\ c_y s_x - c_x s_y & c_x c_y + s_x s_y \end{bmatrix} = \begin{bmatrix} \cos(\theta_x - \theta_y) & -\sin(\theta_x - \theta_y) \\ \sin(\theta_x - \theta_y) & \cos(\theta_x - \theta_y) \end{bmatrix}. \tag{5.25}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_z = -\frac{\pi}{2}, \quad \theta_x - \theta_y = \arctan\left(\frac{r_{21}}{r_{22}}\right). \tag{5.26}$$

ZXY factoring order

$$\mathbf{R}(\theta_z)\mathbf{R}(\theta_x)\mathbf{R}(\theta_y) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z \\ c_z s_x s_y + c_y s_z & c_x c_z & -c_y c_z s_x + s_y s_z \\ -c_x s_y & s_x & c_x c_y \end{bmatrix} \tag{5.27}$$

The most straightforward equivalence is $r_{21} = s_x$, then $\theta_x = \arcsin(r_{21})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_x \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right)$, then $c_x \neq 0$ and $c_x(s_y,\ c_y) = (-r_{20},\ r_{22})$.

$$\therefore \quad \theta_x = \arcsin(r_{21}), \quad \theta_y = \arctan\left(\frac{-r_{20}}{r_{22}}\right), \quad \theta_z = \arctan\left(\frac{-r_{01}}{r_{11}}\right). \tag{5.28}$$

2. When $\theta_x = \frac{\pi}{2}$, then $s_x = 1,\ c_z = 0$ and

$$\begin{bmatrix} r_{00} & r_{02} \\ r_{10} & r_{12} \end{bmatrix} = \begin{bmatrix} c_y s_z - s_y c_z & c_z s_y + c_y s_z \\ c_z s_y + c_y s_z & -c_y c_z + s_y s_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_y + \theta_z) & \sin(\theta_y + \theta_z) \\ \sin(\theta_y + \theta_z) & -\cos(\theta_y + \theta_z) \end{bmatrix}. \tag{5.29}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_x = \frac{\pi}{2}, \quad \theta_y + \theta_z = \arctan\left(\frac{r_{02}}{r_{00}}\right). \tag{5.30}$$

3. When $\theta_x = -\frac{\pi}{2}$, then $s_x = -1,\ c_x = 0$ and

$$\begin{bmatrix} r_{00} & r_{02} \\ r_{10} & r_{12} \end{bmatrix} = \begin{bmatrix} c_y s_x + s_y c_z & c_z s_y - c_y s_z \\ -c_z s_y + c_y s_x & c_y c_z + s_y s_z \end{bmatrix} = \begin{bmatrix} \cos(\theta_y - \theta_z) & \sin(\theta_y - \theta_z) \\ -\sin(\theta_y - \theta_z) & \cos(\theta_y - \theta_z) \end{bmatrix}. \tag*{5.31}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_x = -\frac{\pi}{2}, \quad \theta_y - \theta_z = \arctan\left(\frac{r_{02}}{r_{00}}\right). \tag*{5.32}$$

ZYX factoring order

$$\mathbf{R}(\theta_z)\,\mathbf{R}(\theta_y)\,\mathbf{R}(\theta_x) = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} c_y c_z & c_z s_x s_y - c_x s_z & c_x c_z s_y + s_x s_z \\ c_y s_z & c_x c_z + s_x s_y s_z & -c_z s_x + c_x s_y s_z \\ -s_y & c_y s_x & c_x c_y \end{bmatrix} \tag*{5.33}$$

The most straightforward equivalence is $r_{20} = -s_y$, then $\theta_y = \arcsin(-r_{20})$ and the three cases to consider to retrieve the EA are:

1. When $\theta_y \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right)$, then $c_y \neq 0$ and $c_y(s_x,\ c_x) = (r_{21},\ r_{22})$.

$$\therefore \quad \theta_x = \arctan\left(\frac{r_{21}}{r_{22}}\right), \quad \theta_y = \arcsin(-r_{20}), \quad \theta_z = \arctan\left(\frac{r_{10}}{r_{00}}\right). \tag*{5.34}$$

2. When $\theta_y = \frac{\pi}{2}$, then $s_y = 1,\ c_y = 0$ and

$$\begin{bmatrix} r_{01} & r_{02} \\ r_{11} & r_{12} \end{bmatrix} = \begin{bmatrix} c_z s_x - c_x s_z & c_x c_z + s_x s_z \\ c_x c_z + s_x s_z & -c_z s_x + c_x s_z \end{bmatrix} = \begin{bmatrix} \sin(\theta_x - \theta_z) & \cos(\theta_x - \theta_z) \\ \cos(\theta_x - \theta_z) & -\sin(\theta_x - \theta_z) \end{bmatrix}. \tag*{5.35}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_y = \frac{\pi}{2}, \quad \theta_x - \theta_z = \arctan\left(\frac{-r_{12}}{r_{11}}\right). \tag*{5.36}$$

3. When $\theta_y = -\frac{\pi}{2}$, then $s_y = -1,\ c_y = 0$ and

$$\begin{bmatrix} r_{01} & r_{02} \\ r_{11} & r_{12} \end{bmatrix} = \begin{bmatrix} -c_z s_x - c_x s_z & -c_x c_z + s_x s_z \\ c_x c_z - s_x s_z & -c_z s_x - c_x s_z \end{bmatrix} = \begin{bmatrix} -\sin(\theta_x + \theta_z) & -\cos(\theta_x + \theta_z) \\ \cos(\theta_x + \theta_z) & -\sin(\theta_x + \theta_z) \end{bmatrix}. \tag*{5.37}$$

There is only one degree of freedom and the factorization is not unique.

$$\therefore \quad \theta_y = -\frac{\pi}{2}, \quad \theta_x + \theta_z = \arctan\left(\frac{-r_{12}}{r_{11}}\right). \qquad 5.38$$

5.2 Modifications to the methods

The main modification necessary to make the voting-based methods to work with the EA consist of, first, computing the EA based on the chosen factoring order, and form along with the translation vector **t** a six-dimensional transformation vector $\mathbf{T}_{6D}$. Next, T_{6D} is evaluated by the descriptive metric in **Equation 3.1** but following the chosen factoring order when applying $\mathbf{T}_{6D}$ to $\mathcal{P}$. The rest of the process keeps the same as in the original approach but considering the factoring order also when binning into the accumulator $A\left(\mathbf{T}_{6D}\right)$, and limiting the EA within the range $[-\pi,\ \pi]$ when creating a new bin b_i.

5.3 Experimental evaluation

5.3.1 Implementation and datasets

This alternative version of the voting-based methods adds one more parameter to the ruling parameters (*i.e.*, N_b, N_r and core registration algorithm), the EA's *factorization order* (*cf.* **Section 5.1.2**). Since the main goal is to evaluate the effect of the EA representation in the proposed approach, we performed experiments using the six factorization orders, with a couple of configurations that are reliable in the original approach, $N_b = 90$, $N_r = 32$ with all four CRAs. For the datasets and metrics, we utilized the model datasets as in the original approach, along with the evaluation of R_{error}, t_{error} and ψ_{diff} between the ground-truth and resulting alignments.

5.3.2 Results and discussion

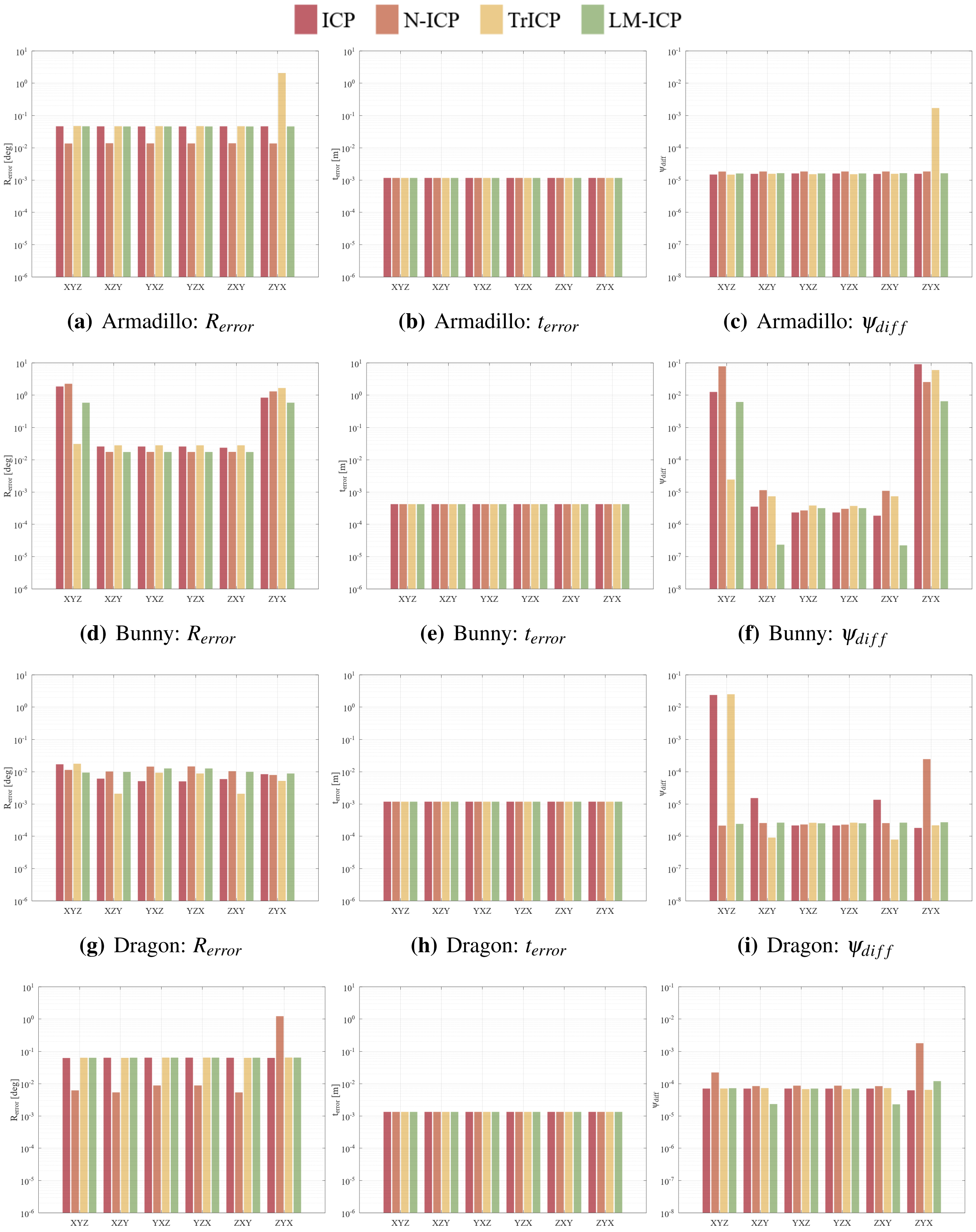
Qualitative results are shown in **Tables 5.1** to **5.4**, depicting in red the source $\mathcal{P}$, in blue the target $\mathcal{Q}$, and the overlap or correspondences $\mathcal{C}$ in black. Quantitative results are outlined in the charts of **Figure 5.1**, the vertical axis corresponds to evaluation metrics in logarithmic scale and the horizontal axis to the factorization orders. By looking at ψ_{diff}, in most cases (85 out of 96) the obtained transformation $\mathbf{T}_b$ produces an alignment very close to the optimal solution with a difference to the ground-truth in the range of $\left(10^{-6},\ 10^{-4}\right)$.

In the eleven cases resulting in misalignment, there is a clear pattern at the factorization orders XYZ and ZYX, specially in the *Bunny* dataset. In this particular case, the corresponding qualitative results in **Table 5.2** do not show an evident misalignment with TrICP as core registration with order XYZ. However, quantitative results on R_{error} and ψ_{diff} (*cf.* **Figures 5.1(d)** and **(f)**) indicate that compared to the other orders, the alignment is slightly worse with $\psi_{diff} > 10^{-5}$. In this case R_{error} does not seem to be a better indicative compared to ψ_{diff}, but rather a complement as it follows a similar pattern with order XYZ and ZYX.

The reason behind the misalignment only on orders XYZ and ZYX is related to the necessary rotation to align the datasets. In reality, the biggest difference between the original positions on all datasets is in the rotation around the vertical y-axis. Therefore, the orders that apply θ_y first are leading to the most significant required rotation at the beginning of the full alignment, and then apply less significant rotations from θ_x and θ_z. This happens the other way around in the opposite case. On the other hand, XYZ and ZYX first apply a less significant rotation, then attempt to amend it with the most significant one but worsen the complete alignment with the third and also less significant rotation. This phenomena is clearer in datasets with large primitive-like surfaces (*i.e.*, spheres, planes, cylinders) that are prone to drift, such as *Bunny* which has many round sphere-like surfaces. In the other datasets, the XYZ and ZYX pattern is not as prominent, but these two factoring orders are where the eleven cases of misalignment come from.

5.4 Conclusion

We investigated the influence of the Euler angles on a voting-based registration methods for low overlapping 3D point clouds. Results indicate that this rotation representation can be adopted and still obtain significantly good alignments, but also that, as expected, the angles' factoring order has a significant impact. Showing that *the factoring order has to be chosen depending on the largest rotation difference between the original positions of the 3D point clouds*. Being necessary to choose an order that does not apply the most significant rotation secondly in the sequence. Furthermore, observations on the quantitative and qualitative results indicate that none of the chosen metrics individually best describes a good alignment. Rather, the judgment should be based on a joint indication between all metrics.



(a) Armadillo: R_{error}

(b) Armadillo: t_{error}

(c) Armadillo: ψ_{diff}

(d) Bunny: R_{error}

(e) Bunny: t_{error}

(f) Bunny: ψ_{diff}

(g) Dragon: R_{error}

(h) Dragon: t_{error}

(i) Dragon: ψ_{diff}

(j) Happy Buddha: R_{error}

(k) Happy Buddha: t_{error}

(l) Happy Buddha: ψ_{diff}

Figure 5.1: Quantitative results of $(N_b = 90,\ N_r = 32,\ CRA)$ on different factoring orders

Table 5.1: Qualitative results between factoring orders in *Armadillo*

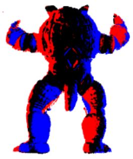
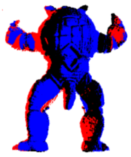
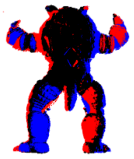
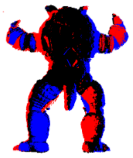
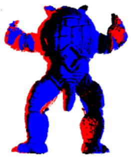
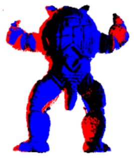
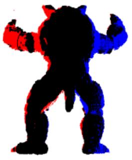
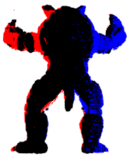
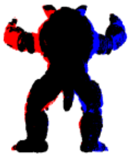
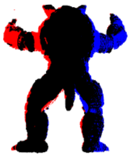
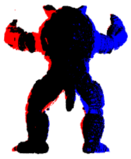
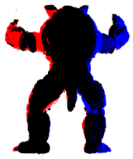
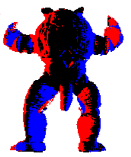
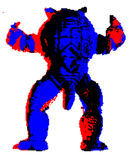
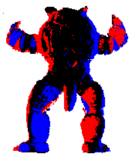
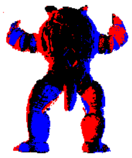
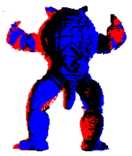
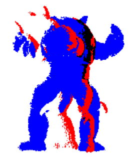
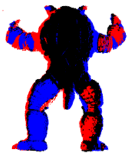
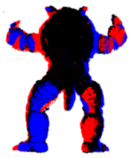
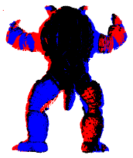
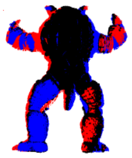
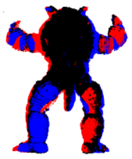
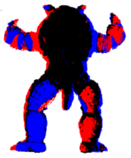
	XYZ	XZY	YXZ	YZX	ZXY	ZYX
ICP						
N-ICP						
TrICP						
LM-ICP						

Table 5.2: Qualitative results between factoring orders in *Bunny*

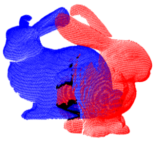
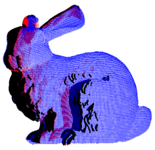
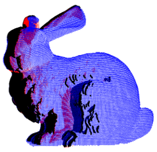
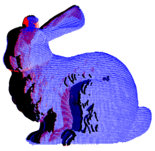
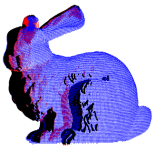
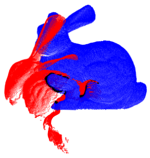
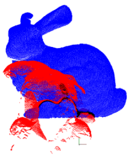
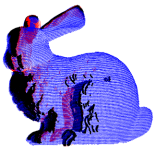
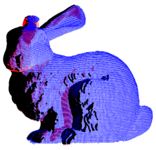
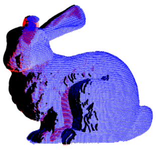
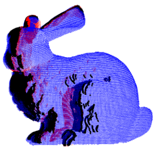
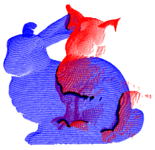
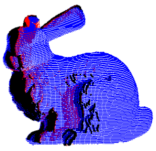
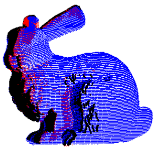
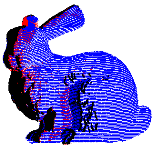
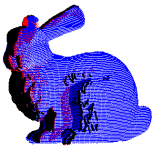
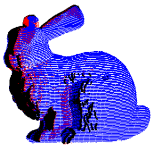
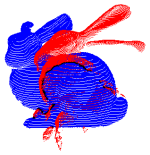
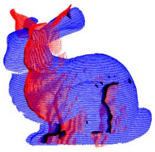
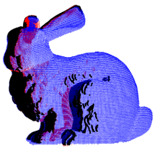
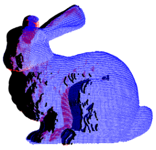
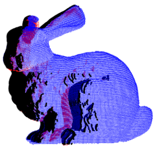
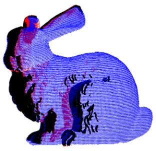
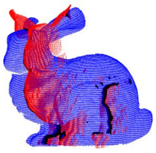
	XYZ	XZY	YXZ	YZX	ZXY	ZYX
ICP						
N-ICP						
TrICP						
LM-ICP						

Table 5.3: Qualitative results between factoring orders in *Dragon*

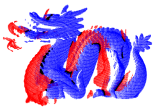
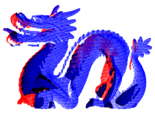
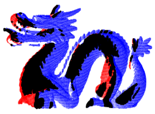
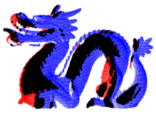
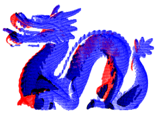
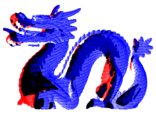
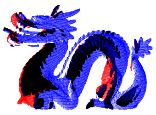
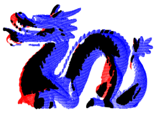
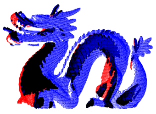
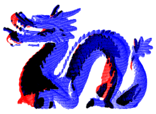
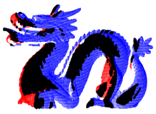
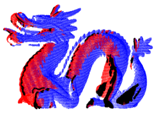
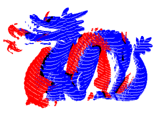
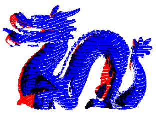
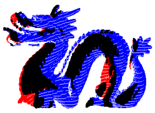
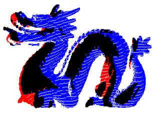
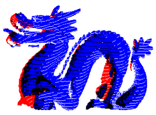
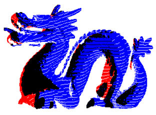
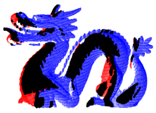
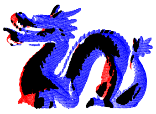
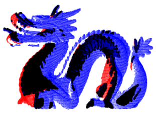
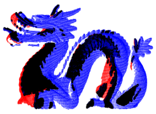
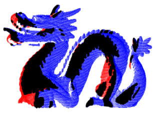
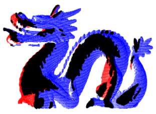
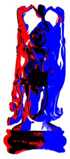
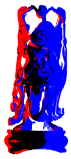
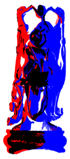
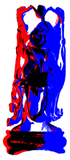
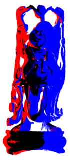
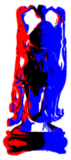
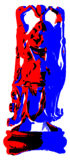
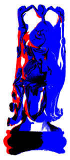
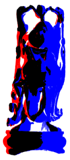
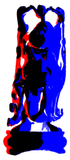
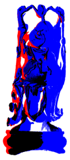
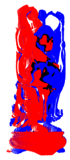
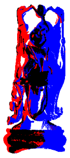
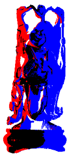
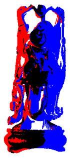
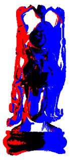
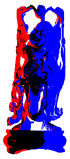
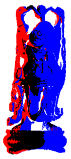
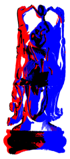
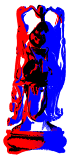
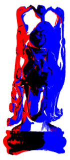
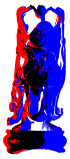
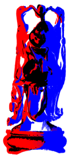
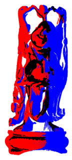
	XYZ	XZY	YXZ	YZX	ZXY	ZYX
ICP						
N-ICP						
TrICP						
LM-ICP						

Table 5.4: Qualitative results between factoring orders in *Happy Buddha*

	XYZ	XZY	YXZ	YZX	ZXY	ZYX
ICP						
N-ICP						
TrICP						
LM-ICP						

Chapter 6

Overlapping limits

This chapter evaluates the overlapping limitations of the registration methods proposed in this work. The contents are an extension that complements the analysis and understanding of these registration methods for low overlapping 3D point clouds.

6.1 Background

Both registration approaches presented in this work were conceived with a focus on low overlapping 3D point clouds, being tested on 3D point clouds with overlapping ratios ranging from 15% to nearly 50% (*cf.* Table 4.2), and resulting in solutions close to the global optimal in model type datasets. As shown in Chapter 4, the proposed registration methods are not optimal for large 3D point clouds from scene-type datasets, mainly due to their time complexity and low success rate on these datasets. Nevertheless, their success on model-type datasets raises the question:

- *Up to what percentage of overlapping ratio the proposed approaches are successful?*

6.2 Experimental evaluation

To answer the previous question, one would like to see how the approaches perform in terms of $\mathbf{R}_{error}$, $\mathbf{t}_{error}$, and ψ_{diff} as the overlapping ratio ψ smoothly transitions from a high to a low percentage, making noticeable the points where the methods fail. Nonetheless, the original datasets of the used model type 3D point clouds do not provide a smooth transition of overlapping ratio (*i.e.*, the 3D point clouds in the complete dataset transition in larger steps, for example, from 100% to 75%, to 50%, to 25%). Therefore, we made our sets of 3D point

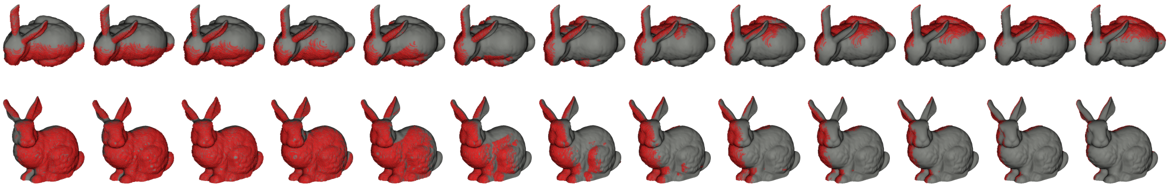


Figure 6.1: Top and isometric views of simulated scan 3D point clouds of the *Bunny* model

clouds from the Armadillo, Bunny, Dragon, and Happy Buddha with smooth overlapping ratio transitions.

6.2.1 Datasets acquisition and preparation

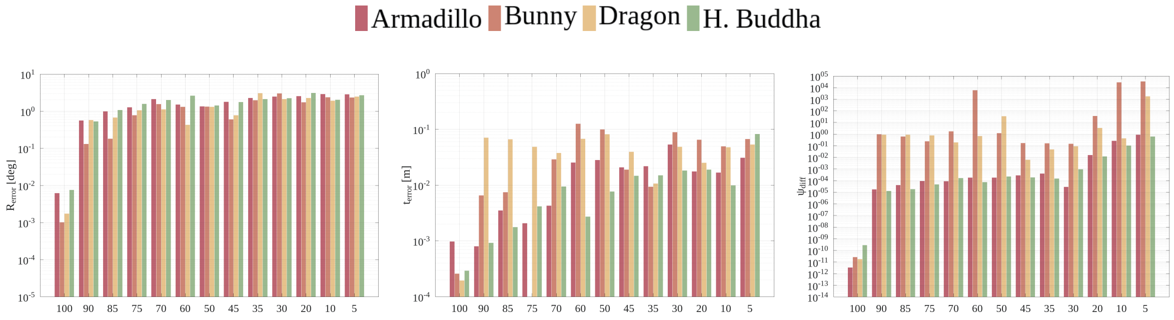
The Kinect 3D sensor simulation from the BlenSor [64] package was utilized to create a series of 3D point clouds which partial overlap that gradually increases/decreases. Each original full model mesh reconstruction of *Armadillo*, *Bunny*, *Dragon*, and *Happy Buddha* provided in the 3D Stanford Repository was imported into BlenSor, gradually rotated around their vertical axis in steps of 15° from 0° to 180°, and scanned in every step with a sensor resolution of 640×480 pixels. This resulted in thirteen 3D point clouds for each model, as shown in the example of **Figure 6.1**, where the 3D point clouds are depicted in red and the original model in gray.

6.2.2 Experimental setup

The experimental setup consisted in running both proposed methods setting the 3D point clouds at 0° as the target $\mathcal{Q}$ and the scans from 0° to 180° as the source $\mathcal{P}$ while evaluating $\mathbf{R}_{error}$, $\mathbf{t}_{error}$, and ψ_{diff}. For the second approach, its four variants were evaluated on the parameter configuration ($N_b = 30$, $N_r = 32$), which generally performs good according to the usage guideline of Section 4.4. Furthermore, once again, no initial rotation or translation was applied to the 3D point clouds before registration. As for the utilized hardware and software, these experiments were carried out in the same ones specified in Section 2.3.1.

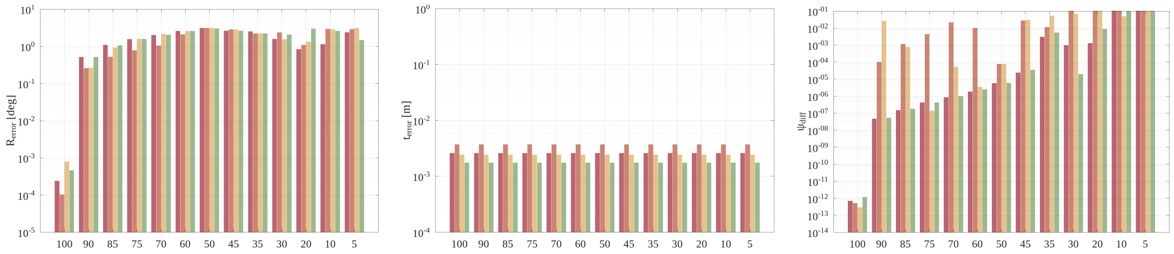
6.2.3 Results and discussion

The charts in **Figure 6.2** show the obtained results from RwSVS, where the horizontal axes correspond to the gradually decreasing overlapping ratio between 3D point cloud pairs, the vertical axes to the evaluation metrics in logarithmic scale, and each series to a dataset. Similarly, **Figures 6.3** to **6.6** show the results from each variant of DfVb.



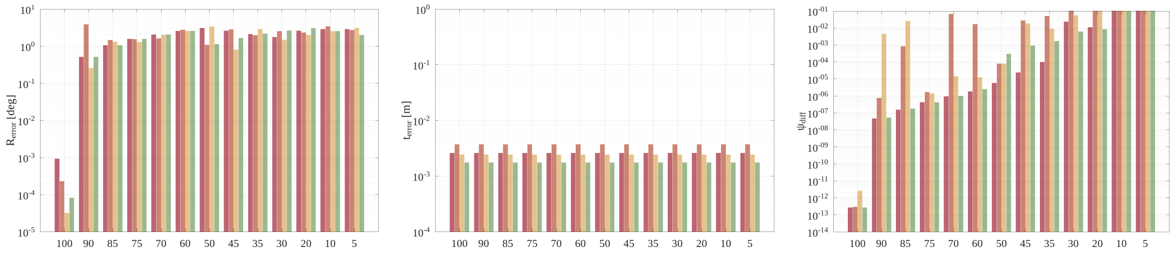
(a) $\mathbf{R}_{error}$ (b) $\mathbf{t}_{error}$ (c) ψ_{diff}

Figure 6.2: Overlapping limit evaluation of RwSVS



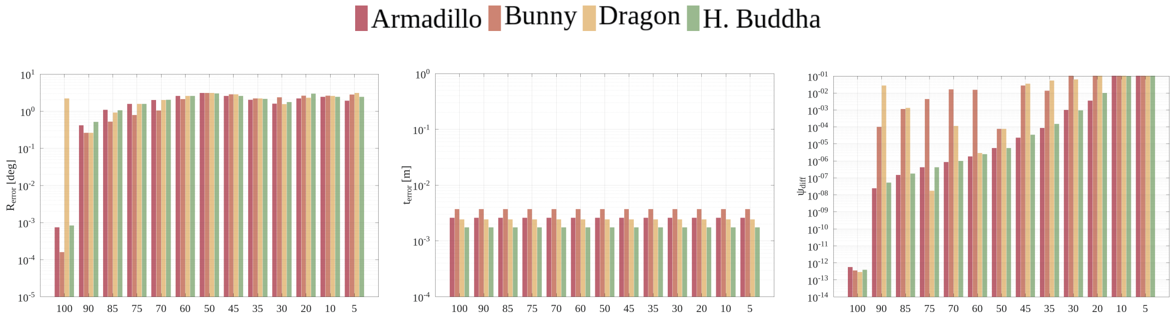
(a) $\mathbf{R}_{error}$ (b) $\mathbf{t}_{error}$ (c) ψ_{diff}

Figure 6.3: Overlapping limit evaluation of DfVb: ICP



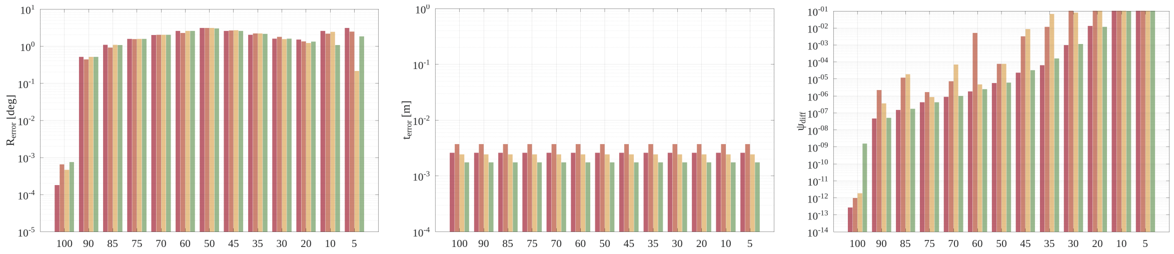
(a) $\mathbf{R}_{error}$ (b) $\mathbf{t}_{error}$ (c) ψ_{diff}

Figure 6.4: Overlapping limit evaluation of DfVb: N-ICP



(a) $\mathbf{R}_{error}$ **(b)** $\mathbf{t}_{error}$ **(c)** ψ_{diff}

Figure 6.5: Overlapping limit evaluation of DfVb: TrICP

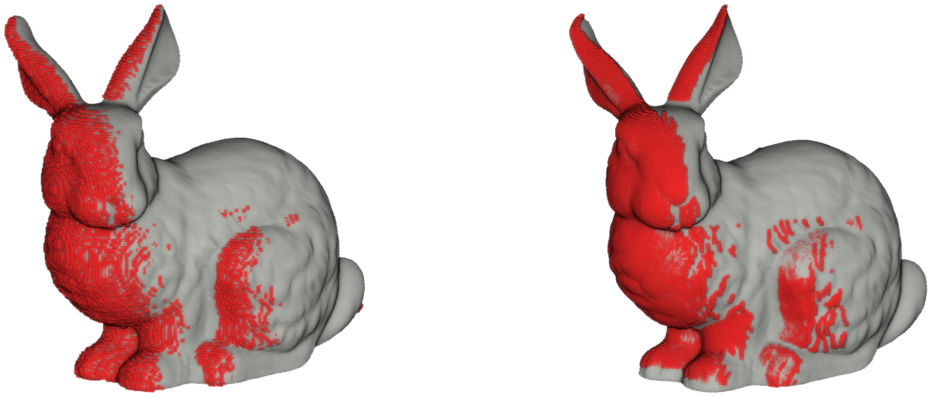


(a) $\mathbf{R}_{error}$ **(b)** $\mathbf{t}_{error}$ **(c)** ψ_{diff}

Figure 6.6: Overlapping limit evaluation of DfVb: LM-ICP

Figure 6.2(c) clearly depicts how RwSVS tends to perform worse as the overlapping ratio decreases in all datasets, which is clearer in *Armadillo* and *Happy Buddha*. *Bunny* and *Dragon* follow a similar tendency but the results are worse compared to the other two models. As shown in **Figure 6.2(a)** the tendency to decrease the quality of the alignment as ξ decreases is clearer with $\mathbf{R}_{error}$, ranging from about $0.1°$ to less than $10°$ in all partial overlaps. In terms of terror, RwSVS also shows a tendency to deteriorate as ξ decreases. In overall, there is a slight pattern in all datasets of decreasing the quality of the solution from 90% to 50% overlap, improving after 50% up to 30%, and decreasing the quality again from 30% to 5%. Therefore, one can say that for the utilized datasets the overlapping limits of RwSVS are $[50\%,\ 30\%]$. Furthermore, despite having $\xi_{gt} = 100\%$, RwSVS shows some error in all the metrics because the solution transformation derives from the alignment of subsets, which can present rotation or translation drift. However, these errors are insignificant due to their low values.

On the other hand, all variants of DfVb show a more steady behavior as ξ decreases, but the same tendency of worse results as there is less overlap between the 3D point clouds. However, it is only noticeable in the $\mathbf{R}_{error}$ and ψ_{diff} results as $\mathbf{t}_{error}$ does not change along



(a) Simulated **(b)** Real

Figure 6.7: Comparison between simulated scan and real scan 3D point clouds

with the overlapping ratio. The tendency in ψ_{diff} results does not help to find the overlapping ratio limits. But, as mentioned in the preliminary conclusion in **Section 4.5** it is better to judge the quality of a solution based on all three metrics. Despite it, by judging the quality of the solution on the $\mathbf{R}_{error}$ in **Figures 6.3(a)**, **6.5(a)**, one can tell that DfVb: ICP, DfVb: TrICP, and DfVb: LM-ICP variants tend to gradually perform worse from 90% to 50%. Then, between 50% to 20% the results improve and worsen below this point. The latter behavior tells us that the overlapping limits of these variants are $[50\%, 20\%]$. As for DfVb: N-ICP, the $\mathbf{R}_{error}$ results in **Figure 6.4(a)** follow a less prominent tendency of improvement at $[45\%, 30\%]$.

Comparing the results on $\mathbf{t}_{error}$ from both approaches, the most noticeable thing is the behavior difference. Meanwhile, RwSVS depicts a pattern similar to $\mathbf{R}_{error}$ and ψ_{diff}, $\mathbf{t}_{error}$ in DfVb remains the same along with the different overlapping ratios. This is an effect of the differences in where the transformations are retrieved from and the mechanism employed to find the solution. In the first method, the transformations derive from the alignment of individual subsets, which translations may differ from the translation of the global solution and present drifts. Meanwhile, in the second method the transformations are retrieved from the alignment of subsets with full 3D point clouds and then binned into a Hough space. It is because of the voting mechanism that the different translations get normalized and show a steady behavior along with the change of overlapping ratio.

Additionally, results also show that the changing points that indicate the overlapping limits are not very prominent. It may be caused by the smooth overlapping transitions achieved by the simulated scan datasets. But also, because as shown in **Figure 6.7** these datasets have rougher surfaces, fewer points, and noise at the borders. Simulated scan 3D point clouds are rougher due to some voxel-like artifacts that add more descriptiveness to

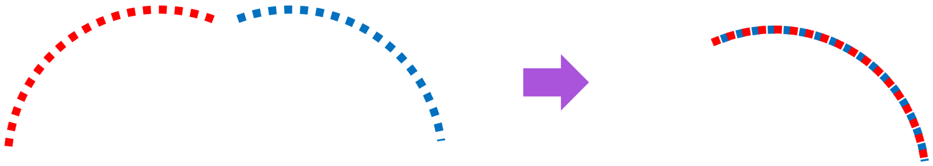


Figure 6.8: Drifted alignment ot two smooth round surfaces

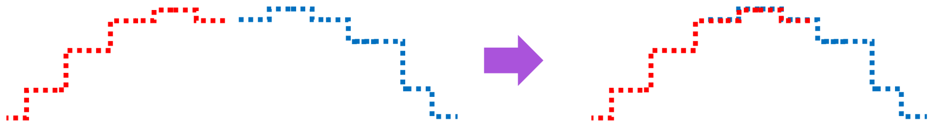


Figure 6.9: Non-drifted alignment ot two non-smooth round surfaces

surfaces that are very smooth in the real scan data, like in the chest of *Bunny*. This helps to avoid drifted alignments that could arise from subsets of featureless round surfaces and differentiate better small rotational differences when registering these, as shown in **Figures 6.7** and **6.9**.

6.3 Conclusion

The registration approaches proposed in this work showed success in overlapping ratios between 50% down to 30% or 20%, depending on the method or variant. It tells us that because the approaches were conceived to work on low overlapping ratios ($\xi \leq 40\%$), it is around these ratios that can perform well on model-type datasets from simulated 3D sensors. In addition, experimental results reinforce that to judge the quality of an alignment given by any transformation, it should be based on the analysis of the values obtained by all three evaluation metrics.

Chapter 7

Conclusion and open problems

In this work, we introduced our research on leveraging subset segmentation for the pairwise registration of 3D point clouds under low overlapping conditions. Unlike traditional and newer approaches, we proposed two methods that do not make direct point-to-point correspondences to retrieve a solution transformation. Instead, these replace the points for point subsets as the focus of attention, indirectly adding the neighbor interaction information to the subsets and inheriting semi-local shape features that can be paired with the ideal conditions of conventional registration algorithms to find a close to optimal alignment.

The first proposed methods can successfully retrieve transformations close to the optimal alignment for 3D point clouds with low overlapping ratios through a straightforward optimization multi-scale process with a linear time complexity dictated by the size of the input 3D point clouds. The second method automates the definition of the multi-scale process parameters and replaces the maximization optimization for a Hough voting scheme. The latter casts votes for transformations from subset alignments that minimize the error between putative point correspondences and maximize the overlapping ratio. It also results in alignments close to the global optima but only in model-type datasets and log-linear time complexity. As the latter approach allows a wide range of parameter configurations, we demonstrated its performance and behavior in a significantly large range of parameters, establishing a usage guideline for any end-user to obtain the best possible results without much effort.

Additionally, we included a parallel version of the second approach, which replaces the rotation representation in the parameter space for the Euler angles, showing that it can also obtain alignments close to the global optima in model-type datasets. However, in this case, the angles' factorization order is added as a parameter that must be chosen based on the most significant rotational difference between the initial positions of the 3D point clouds. Lastly, we evaluate both methods using 3D point clouds from simulated sensors on a smooth transition from high to low overlap to investigate their overlap limits and find the range where these succeed – confirming that their working range is in the low overlap category.

In summary, we have shown and proved the potential and limitations of addressing the registration of low overlapping 3D point clouds via subset-focused approaches. Nevertheless, although for model-type datasets, we have obtained good results in terms of quality of alignments based on the employed evaluation metrics, the proposed methods are not faultless, and we recognize several open problems:

- The first proposed method uses FPFH point descriptors to set a degree of similarity between point subsets. However, it only employs a somewhat binary threshold to judge if a pair of subsets are geometrically related or not. A better approach could be to define a weighted degree of similarity, where only point subsets with a "high weight" are justified to be locally registered. Notwithstanding, we still have to consider that point descriptors such as FPFH can still lead to false positives due to the differences in point densities in low overlapping conditions. Hence, it needs a robust local point description against point densities for better results or a fast and robust matching technique like the one proposed by Ref. [65] based on Robust Optimal Transport.

- We think the descriptive metric ψ introduced in Chapter 3 deserves to be studied more deeply. Preliminary conclusions point out that we cannot base the judgment of the correctness of a transformation solely on its difference from the ground-truth but consider the errors in rotation and translation. By studying its behavior under different shapes of 3D point clouds and point densities and determining the optimal value for its parameter λ, we could potentially make it more robust and improve its descriptiveness. Then, spread its usage to describe good transformations in the registration problem since there is no quantitative rule to describe a good transformation.

- The first obvious problem in the proposed voting-based approach is the lack of consistent behavior in different parameter values. We believe that the main reason behind it is the lack of a robust definition of *true overlapping points* that form the correspondences $\mathcal{C}$. Both methods utilize a radius search nearest-neighbor as the base to define them, which is affected by the initial positions. Therefore, when the voting-based method attempts to improve the initial positions by adding more rotations, we also add more possibilities of obtaining false-positive transformations. In order to solve this issue, we need some robust point descriptions to guarantee to align the subsets where they should, no matter what the initial positions are. Furthermore, it could also be helpful to reduce the processing time of both methods and discriminate subset better.

- The biggest drawback of the proposed methods is their unpractical time complexity. One of the reasons we did not compare both methods to more sophisticated Deep Learning-based approaches like Ref.[59, 60] was that the time complexity did not allow us to obtain results fast enough to freely test on larger datasets and evaluations with widely adopted metrics such as *Registration Recall*. We could say that the brute force search of both approaches is their biggest flaw. We think that adopting the proposed subset division concept and pairing it with robust feature descriptors and novel search techniques such as GORE [66] and TEASER [67] could drastically reduce the solution search space and deliver faster and more accurate results.

- Regarding processing datasets other than model-type, only the first method showed capabilities to find adequate solutions for laser scan datasets, but at the cost of long processing times. 3DLoMatch, a new dataset of indoor scene-type 3D point clouds tailored explicitly for the low overlap problem, proved to be a challenge for both methods. Nonetheless, these were down-sampled to reduce the processing time, affecting the cloud resolution and consequently the construction of the subsets by supervoxel segmentation and the inlier threshold that determines the correspondences $\mathcal{C}$ – potentially increasing the likelihood of retrieving many false-positive transformations. It indicates that both methods are sensitive to the point densities and that voxel down-sampling is not an adequate solution to reduce the processing time when registering large-scale 3D point clouds with the proposed methods.

- Lastly, we acknowledge that because the core registration in both methods is a conventional algorithm, these may have inherited the conventional algorithms' weak points. Hence, being weak against 3D point clouds whose original positions are significantly far away in terms of translation. It could also add to the poor performance on the point clouds from 3DLoMatch since these are significantly far away from each other at their original positions compared to the other datasets. Hence, both methods can be classified as local registration algorithms, and more work must be done to match up with other state-of-the-art methods such as Ref.[59, 60].

Bibliography

[1] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, 2006.

[2] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[3] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[4] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.

[5] J. Santamaría, O. Cordon, and S. Damas. A comparative study of state-of-the-art evolutionary image registration methods for 3d modeling. *Computer Vision and Image Understanding*, 115:1340–1354, 2011.

[6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34:133–148, 2013.

[7] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[8] L. Peng, W. Ruisheng, W. Yanxia, and T. Wuyong. Evaluation of the icp algorithm in 3d point cloud registration. *IEEE Access*, 8:68030–68045, 2020.

[9] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2724–2729, April 1991.

[10] D. Chetverikov, D. Svirko, D. Stephanov, and P. Kresk. The trimmed iterative closest point algorithm. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR 2002)*, volume 3, pages 545–545, 2002.

[11] A. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21:1145–1153, April 2002.

[12] L. Peralta, J. Sandoval, M. Iwakiri, and K. Tanaka. A preliminary study on low overlapping unorganized 3d point clouds registration using hough voting. *Reports of the 288th Technical Conference of the Institute of Image Electronics Engineers of Japan*, pages 8–13, 2019.

[13] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. *European Conference on Computer Vision (ECCV 2010)*, 2010.

[14] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. *The IEEE International Conference on Robotics and Automation (ICRA 2009)*, pages 3212–3217, 2009.

[15] M.A. Fishcler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commutations of the ACM*, 24(6):381–395, 1981.

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pages 3354–3361, 2012.

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 1912–1920, 2015.

[18] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 199–208, 2017.

[19] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. *IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, pages 8957–8965, 2019.

[20] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C. Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pages 6358–6366, 2020.

[21] Y. Aoki, H. Goforth, R.A. Srivatsan, and S. Lucey. Pointnetlk: Robust and efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 7156–7565, 2019.

[22] Y. Wang and J. Solomon. Deep closest point: Learning representations for point cloud registration. *IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, pages 3522–3531, 2019.

[23] G. Dias Pais, S. Ramalingam, V.M. Govindu, J.C. Nascimento, R. Chellappa, and P. Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 7193–7203, 2019.

[24] C. Choy, W. Dong, and V. Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*, pages 2511–2520, 2020.

[25] S. Hanneke and S. Kpotufe. On the value of target data in transfer learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[26] M. Yishay, M. Mehryar, R. Jae, T.S. Ananda, and W. Ke. A theory of multiple-source adaptation with limited target labeled data. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 2332–2340, 2021.

[27] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[28] A. Willis, D. Anderson, T. Thomas, T. Brown, and J. Marsh. 3d reconstruction of highly fragmented bone fractures - art. no. 65121p. *Proceedings of SPIE - The International Society for Optical Engineering*, 6512, 03 2007.

[29] A.J. Collings and K. Brown. Reconstruction and physical fit analysis of fragmented skeletal remains using 3d imaging and printing. *Forensic Science International: Reports*, 2:100114, 2020.

[30] Y. Wu, W. Wang, K. Lu, Y. Wei, and Z. Chen. A new method for registration of 3d point sets with low overlapping ratios. *Procedia CIRP*, 27:202–206, 2015.

[31] Z. Cai, T. Chin, A.P. Bustos, and K. Schindler. Practical optimal registration of terrestrial lidar scan pairs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:118–131, 2019.

[32] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 22–27, 2013.

[33] Cie 15: Colorimetry. Tecnical report 3, Commission Internationale De L'eclairage, 2004.

[34] R.W.G. Hunt. *The reproduction of color*. John Wiley & Sons, 6 edition, 2004.

[35] A.G. Buch, D. Kraft, J. Kamarainen, H.G. Petersen, and N. Krüger. Pose estimation using local structure-specific shape and appearance context. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2013)*, pages 2080–2087, 2013.

[36] D.G. Lowe. Distinctive image features from scale-invariant key-points. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[37] R. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). *The IEEE International Conference on Robotics and Automation (ICRA 2011)*, 2011.

[38] Openmp. https://www.openmp.org/. Accessed: 2022-04-30.

[39] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIG-GRAPH, ACM*, pages 303–312, 1996.

[40] F. Pomerlau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *The international Journal of Robotics Research*, 31(14):1705–1711, 2012.

[41] L. Peralta, J. Sandoval, M. Iwakiri, and K. Tanaka. Registration of low overlapping 3d point clouds boosted by simulated annealing. *2019 IEEE Shin-etsu Session*, 2019.

[42] D. Chetverikov, D. Stephanov, and P. Kresk. Robust euclidean alignment of 3d point sets: The trimmed iterative closest point algorithm. *Image and Vision Computing (IVC 2005)*, 23:299–309, 2005.

[43] P. Hough. Methods and means for recognizing complex patterns, 1962.

[44] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics and Image Processing*, 40(3):361–387, 1987.

[45] D.P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision (IJCV 1990)*, 5(2):195–202, 1990.

[46] H.J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

[47] Y.C. Hecker and R. Bolle. On geometric hashing and the generalized hough transform. *IEEE Transactions on Systems Man and Cybernetics*, 24:1328–1338, 1994.

[48] T. Rabbani and F. Van Den Heuvel. Efficient hough transform for automatic detection of cylinders in point clouds. In *ISPRS Ws. Laser Scanning*, pages 60–65, 2005.

[49] G. Vosselman, B. Gorte, G. Sithole, and T. Rababani. Recognising structure in laser scanner point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46:33–38, 11 2003.

[50] F. Tombari and L. Di Stefano. Hough voting for 3d object recognition under occlusion and clutter. *IPSJ Transactions on Computer Vision and Applications*, 4:20–29, 2012.

[51] A. Pressley. *Elementary Differential Geometry*. Springer-Verlag London, 2001.

[52] L. Sun, Y. Manabe, and N. Yata. Sparse group match for point cloud registration. *ITE Transactions on Media Technology and Applications*, 6(2):151–161, 2018.

[53] L. Sun, Y. Manabe, and N. Yata. Double sparse representation for point cloud registration. *ITE Transactions on Media Technology and Applications*, 7(3):145–158, 2019.

[54] M. Aharon, M. Elad, and A.M. Bruckstein. The k-svd: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[55] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Patter Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[56] R. Rubinstein, A.M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. In *Proceedings of the IEEE*, volume 98, pages 1045–1057, 2010.

[57] M. Prokop, S.A. Shaikh, and K. Kim. Low overlapping point cloud registration using line features detection. *Remote Sensing*, 12(1):Article 61, 2020.

[58] J. Stechschulte, N. Ahmed, and C. Heckman. Robust low-overlap 3-d point cloud registration for outlier rejection. *The IEEE International Conference on Robotics and Automation (ICRA 2019)*, pages 7143–7149, 2019.

[59] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, pages 4267–4276, 2021.

[60] J. Lee, S. Kim, M. Cho, and J. Park. Deep hough voting for robust global registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[61] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision (IJCV 2013)*, 103(3):267–305, 2013.

[62] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[63] D. Eberly. Euler angle formulas. Technical report, Geometric Tools, Redmond Washington 98052, 12 1999.

[64] M. Gschwandtner, R. Kwitt, and A. Uhl. Blensor: Blender sensor simulation toolbox. *Advances in Visual Computing: 7th International Symposium*, 2011.

[65] Z. Shen, J. Feydy, P. Liu, A.H. Curiale, R.S.J. Estepár, R.S.J. Estepár, and M. Niethammer. Accurate point cloud registration with robust optimal transport. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[66] Á. Parra and T. Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2868–2882, 2018.

[67] H. Yang, J. Shi, and L. Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021.

Publications

- Luis Peralta, Munetoshi Iwakiri, and Kiyoshi Tanaka. *An Experimental Evaluation on the Overlap Limitations of Subset focused Registration Methods for 3D Point Clouds*. Proceedings of the 50th Annual Conference of the Institute of Image Electronics Engineers of Japan. September 2022.

- Luis Peralta, Munetoshi Iwakiri, and Kiyoshi Tanaka. *A Registration Method on Piece-wise Voting for Low Overlapping 3D Point Clouds*. IIEEJ Transactions on Image Electronics and Visual Computing. Vol.9, No. 2, pp.58-68. December 2021. [**Journal**]

- Luis Peralta, Munetoshi Iwakiri, and Kiyoshi Tanaka. *A Piece-wise Registration Method for Low Overlapping 3D Point Clouds*. Proceedings of the 7th IIEEJ International Conference on Image Electronics and Visual Computing (IEVC 2021), 2A-1. September 2021.

- Luis Peralta, Munetoshi Iwakiri, and Kiyoshi Tanaka. *An Evaluation of Order of Significance of Euler Angles in Voting-based Registration of 3D Point Clouds*. Proceedings of the 49th Annual Conference of the Institute of Image Electronics Engineers of Japan. June 2021.

- Luis Peralta, Munetoshi Iwakiri, and Kiyoshi Tanaka. *Descriptor-free Voting-based Registration of Low Overlapping 3D Point Clouds*. Proceedings of the 48th Annual Conference of the Institute of Image Electronics Engineers of Japan. December 2020.

- Luis Peralta, Jaime Sandoval, Munetoshi Iwakiri, and Kiyoshi Tanaka. *Pairwise Registration of Low Overlapping 3D Point Clouds Using Supervoxel Segmentation*. IIEEJ Transactions on Image Electronics and Visual Computing. Vol.7, No. 2, pp.78-87. December 2019. [**Journal**]

- Luis Peralta, Jaime Sandoval, Munetoshi Iwakiri, and Kiyoshi Tanaka. *Pairwise Registration of Low Overlapping Unorganized 3D Point Clouds Using Supervoxel Segmentation*. Proceedings of the 6th IIEEJ International Conference on Image Electronics and Visual Computing (IEVC 2019), 4C-6. August 2019.