

A 3D Vision Framework for Generalized Workpiece Localization in Robotic Manufacturing

by

Alejandro Hernandez Villa

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Department of Mechanical Engineering)

UNIVERSITY OF MANITOBA

(Winnipeg)

July 2024

© Alejandro Hernandez Villa, 2024

Abstract

Automated workpiece localization through CAD-based Point Cloud Registration (PCR) plays a key role in enabling adaptive robotic tasks, e.g., mapping tool-paths from a CAD model to the current workpiece location. However, existing approaches often lack the robustness required for full autonomy and need fine-tuning and operator intervention especially in the presence of corrupted data due to scene complexity, missing workpiece information, or sensor inaccuracies. These shortcomings are more evident when generalizing to the diversity of objects in highly variable scenarios such as in high-mix manufacturing. In this work, we address this problem by solving the CAD-to-workpiece alignment via a partial-to-partial PCR using 3D vision data from a deep learning and optimization standpoint. Our registration pipeline consists of three modules. First, we leverage pre-trained computer vision models to provide workpiece-agnostic segmentation. Then, a coarse image-guided alignment step is applied, followed by a novel registration architecture with joint global feature embedding and multi-head cross attention. Additionally, we introduce a novel gradient-based optimization method for fine point cloud registration. The quality of registration is primarily evaluated based on registration success rate, rotation, and translation accuracy via isotropic and anisotropic errors as well as runtime. The registration robustness is demonstrated through testing across three distinct datasets aimed to assess the proposed solution with 1) industrial object real-data, 2) diversity of industrial-looking objects through a proposed rule-based synthetic data generation engine with common point cloud corruptions, and 3) on an object-centric dataset standard for benchmarking in the point cloud research community. Our optimization-based method achieves a 99.8% success rate

and significantly improves localization robustness compared to existing techniques, demonstrating its potential for real-world scenarios without parameter fine-tuning. Moreover, our proposed optimization-based strategy shows the ability to register partial overlapping point clouds from a coarse-to-fine manner without an initial guess. Our findings suggest that our pipeline, hybrid or purely optimization-based, shapes a step towards fully autonomous systems in manufacturing.

Table of Contents

Abstract	ii
Table of Contents	iv
List of Tables	vii
List of Figures	viii
Acknowledgements	xii
1 Introduction	1
1.1 Background and research motivation	1
1.2 Problem statement and research objectives	3
1.3 Proposed solution and approach	4
1.4 Thesis layout	6
2 Literature review	7
2.1 Overview	7
2.2 Point cloud registration	7
2.2.1 Traditional/optimization-based registration	8
2.2.2 Learning-based registration	10
2.3 Sensor data fusion	13
2.4 Robustness of point cloud tasks	13
2.5 Registration validation	14
2.6 Industrial object datasets	15
2.7 Summary	16

3	Learning-based registration	17
3.1	Overview	17
3.2	Vision-based automated part localization	17
3.3	Module 1: Workpiece segmentation	20
3.4	Module 2: Global registration via learning-based image keypoint matching	21
3.5	Module 3: Registration network with joint point cloud encoding	26
3.6	Datasets	29
3.6.1	Synthetic data generation engine	29
3.6.2	Synthetic partiality and corruptions	30
3.7	Registration evaluation metrics	33
3.8	Results and discussion	34
3.9	Summary	37
4	Optimization-based registration	39
4.1	Overview	39
4.2	Optimization-based global registration	39
4.2.1	Optimization variables	43
4.2.2	Data preprocessing	44
4.2.3	Multi-seed initialization	45
4.2.4	Loss function	46
4.2.5	Stop criteria	51
4.2.6	Selection of the best alignment candidate	51
4.3	Datasets	51
4.4	Results and discussion	52
4.4.1	Testing on composite part dataset	53
4.4.2	Testing on synthetic dataset	62
4.4.3	Testing on ModelNet40 dataset	64
4.4.4	Loss ablation study	65
4.5	Summary	67

Table of Contents

5	Conclusions	69
5.1	Completed work and key contributions	69
5.2	Future research direction	70
	Bibliography	72

List of Tables

3.1	Datasets for training and testing of the proposed registration network.	33
3.2	Test results and comparison of the proposed network on the sample workpiece dataset against existing learning-based baselines.	36
4.1	Datasets for testing our optimization-based method.	52
4.2	Test results on random rotation misalignment.	53
4.3	Test results on random translation misalignments.	54
4.4	Comparison of our proposed optimization-based method against ex- isting baselines.	56
4.5	Test results on increasing the number of parallel optimization seeds.	57
4.6	Test results of the severity factor on the industrial synthetic dataset.	62
4.7	Test results on the rotational misalignment on the industrial syn- thetic dataset.	63
4.8	Test results on clean and noisy samples from the ModelNet40 dataset.	64
4.9	Test results on the loss ablation study.	66

List of Figures

1.1	Big picture of our proposed registration pipeline.	5
3.1	Robot-agnostic transformations for part localization. Chain of transformation involves the robot flange position, 3D camera mount position, and workpiece capture.	18
3.2	CAD-based point cloud registration.	19
3.3	Point cloud workpiece segmentation from mask.	21
3.4	Outline of our global registration via learning-based image keypoint matching.	22
3.5	Matched keypoints between the CAD model (top) and the workpiece scan (bottom).	25
3.6	Architecture of our proposed registration neural network.	26
3.7	Free-form models generated synthetically.	31
3.8	Cutouts and Gaussian noise on the test case 3D model.	32
3.9	Sample captures in the composite part dataset.	33
3.10	Qualitative results of our learning-based registration pipeline with a downsampled point cloud of the test part dataset.	37
4.1	Overview of our proposed gradient-based registration method. . . .	42
4.2	Automatic statistical outlier removal. Red points denote outliers in the sample composite part.	44
4.3	Initialization of seeds in the rotation solution space using Euler angle variations around the y- and z-axis.	46
4.4	Operation of the proposed F1-Overlap Score on point clouds. . . .	50

4.5	Success rate as a function of the initial misalignment.	55
4.6	Success rate comparison of our method (green) against existing tra- ditional (purple) and learning-based (blue) baselines on real-world data.	56
4.7	Success rate and runtime (per pairwise alignment) as a function of the number of seeds.	57
4.8	Success rate as a function of percentage overlap on the sample com- posite part dataset.	58
4.9	Qualitative results on the composite part dataset.	59
4.10	Convergence of loss function.	60
4.11	Gradients as a function of number of iterations.	60
4.12	Optimization landscape translating the source in the x- and y-axis.	61
4.13	Optimization landscape translating the source in the x- and z-axis.	61
4.14	Optimization landscape translating the source in the y- and z-axis.	62
4.15	Success rate as a function of severity factor on the synthetic dataset.	63
4.16	Qualitative results on our custom synthetic dataset.	64
4.17	Qualitative results on the ModelNet40 dataset.	65
4.18	Qualitative results of best alignment based on two registration metrics.	67

Acronyms

CD Chamfer Distance. 47, 48, 66

DGCNN Dynamic Graph Convolutional Neural Network. 26

DL Deep Learning. 16

DOF Degree of Freedom. 40

FPFH Fast Point Feature Histograms. 8

GICP Generalized Iterative Closest Point. 33, 52, 56

GPU Graphics Processing Unit. 28, 41, 53

HMLV High-Mix Low-Volume. 1

HOUV Hybrid Optimization with Unconstrained Variables. 56

ICP Iterative Closest Point. 58

LBFGS Memory-Limited Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.
66

LiDAR Light Detection and Ranging o Laser Imaging. 13

NBV Next-best-view. 14

PCA Principal Component Analysis. 8

PCR Point Cloud Registration. ii, 9, 16, 39, 41, 54

- RanICP** Feature Matching Registration based on Random Sample Consensus
Followed by ICP. 56
- RGB** Red, green and blue. 13
- ROI** Region of Interest. 35
- SF** Severity Factor. 30
- SGD** Stochastic Gradient Descent. 66
- STL** STereoLithography File Format. 29
- SURF** Speeded Up Robust Feature. 13
- UPGReg** Unsupervised Parallel Gradient-based SE(3) Registration. 39, 40, 42,
43, 53, 54, 64

Acknowledgements

I would like to thank my research supervisor, Dr. Matt Khoshdarregi, for his unwavering support and guidance and for believing in me since our first contact back in 2019 as a MITACS intern. I am thankful to my lab colleagues in the Intelligent Digital Manufacturing (IDM) Lab at the University of Manitoba and my friends who made my academic experience unforgettable.

I would like to thank our research and funding partners including Research Manitoba, MITACS, Natural Sciences and Engineering Research Council of Canada (NSERC), and the Vehicle Technology Center (VTC) for their financial support and contributions to this work.

Finally, I am infinitely thankful to my parents and family who always supported me throughout this journey in Canada to the distance.

This thesis has been proofread using the assistance of OpenAI's ChatGPT [1], however, the original material has been fully written by the author.

Chapter 1

Introduction

1.1 Background and research motivation

While industrial robots have excelled at automating repetitive tasks in mass production, there is a growing interest in deploying robots in High-Mix Low-Volume (HMLV) manufacturing applications where there is significant variability in the arbitrary workpiece placement. This expansion in the dynamic environment of High-Mix manufacturing presents challenges to robots when it comes to adaptability. Hence, 3D machine vision systems and automated part localization play a key role in enabling adaptive robotic tasks in these scenarios.

Automated part localization in robot cells allows for higher flexibility, enhanced agility, and mass customization. In addition, vision-based part localization reduces the need for extensive manual program refinements and repeatable fixturing. Transitioning towards fully autonomous and agile systems can strengthen the productivity and competitiveness of manufacturing companies.

Many industrial sectors can benefit from flexible robot cells equipped with automated part localization. For example, the heavy vehicle manufacturing industry fabricates a high mix of fiberglass parts [2] such as those used in this research.

This work focuses on robust part localization in robot cells, i.e., finding the workpiece position relative to the robot base, for accurate task execution using 3D vision techniques. In particular, recognizing the criticality of point cloud registration for a successful automated localization process, this work primarily focuses on CAD-based point cloud registration, also referred to as CAD-part matching.

Although extensive research has been carried out on the point cloud registration

problem, existing techniques often lack the robustness for full autonomy, need fine-tuning to adapt to the diversity of objects, or need operator intervention. These limitations arise from two primary factors, as follows:

- **Point cloud acquisition quality:** The quality of data plays a significant role when dealing with real-world point clouds from 3D vision cameras. Challenges in point cloud acquisition include sensor inaccuracies (more predominantly noise and specular voids, although ambient light interference and limitations in dimensional trueness could also be considered), scene complexity (presence of clamping devices, occlusion, background remains, non-rigid deformation of the workpiece), and missing workpiece information when performing registration with partially overlapping point clouds (low fidelity of the nominal 3D model to the actual workpiece, or in scenarios where only a partial section of the workpiece is captured and we rely on this information to perform global localization of the entire workpiece). Beyond these challenges, intricate workpiece geometry and non-prominent or similar part features across the same workpiece may also pose a challenge for robust localization.
- **Registration methods:** There are two general approaches have been adopted in the literature related to registration, namely optimization-based and learning-based. Optimization-based techniques are computationally expensive, struggle with local minima, and are sensitive to outliers and partially overlapping environments. Additionally, conventional optimization-based registration methods do not necessarily converge to the globally optimized point cloud alignment, especially in complex scenarios. On the other hand, learning-based approaches mainly struggle with low generalization and require model retraining and fine-tuning. Recent learning-based models that aim to tackle robust partial-to-partial registration have become computationally intensive, which prevents them from being readily used in practical industrial applications. Moreover, there is a lack of sufficient training datasets representing

industrial parts.

With this thesis, we aim to devise a partial-to-partial point cloud registration strategy to robustly handle real-world data while maintaining generality across diverse industrial objects shaping a step towards fully autonomous systems for part localization in manufacturing.

1.2 Problem statement and research objectives

The primary objective of this thesis is to propose strategies to enable automated part localization using State-of-the-Art (SoTA) 3D vision cameras and a novel registration framework. Having in mind the prevalent challenges in part localization, this study seeks to provide insights into robust registration with the following objectives:

1. **Comprehensive review of learning-based registration:** We provide a compilation of learning-based methods with a focus on object-centric registration assessed for practical scenarios.
2. **Synthetic data generation engine:** We explore a technique to automatically generate synthetic point clouds of industrial objects to be able to test our robust registration across a diversity of objects.
3. **Dealing with missing workpiece information:** We provide insight into the application of pre-trained SoTA computer vision models to provide partiality perceptual ability into point cloud registration tasks.
4. **Handling noisy point cloud data:** We delve into deep learning approaches and an optimization strategy to deal with point cloud corruptions and provide accurate workpiece-agnostic segmentation.

As a case study, in this thesis we are particularly interested in free-form composite parts commonly used in the body of heavy vehicles such as trucks, buses, RVs, agricultural machinery, etc.

By addressing these objectives, we aim to enhance partial-to-partial point cloud registration with robustness against corruption.

1.3 Proposed solution and approach

In this thesis, we propose a coarse-to-fine registration pipeline to enhance point cloud registration robustness under real-world data and partiality while maintaining generalizability across industrial objects in a three-module registration pipeline. Taking the best of both realms, learning- and optimization-based global registration, our proposed registration strategy tackles the point cloud alignment problem from these two perspectives. Our pipeline is created to combine both approaches, i.e., image deep perceptual models and gradient-based optimization, to achieve a solution for autonomous workpiece localization.

First, this research examines the emerging role of large pre-trained computer vision models in the context of workpiece localization to aid the registration tasks. We propose a proof-of-concept application of these models that incorporates image data into the registration pipeline. The models provide workpiece-agnostic segmentation and global (coarse) image-guided alignment via image keypoint matching between the actual workpiece and synthetically generated views of the nominal CAD model. Additionally, on the deep learning side, we introduce a neural network architecture with a novel joint global feature embedding and multi-head cross-attention layer.

Then, we propose a novel gradient-based optimization method and loss function for robust point cloud fine registration, as a subsequent stage in our registration pipeline, leveraging existing deep learning machinery. This stage can be interchanged with our proposed neural network module to provide a hybrid learning and optimization-based strategy, or used standalone without an initial guess to provide accurate registration.

For domain-specific benchmarking of our registration algorithms, we introduce a rule-based synthetic point cloud engine that randomly generates industrial-

looking objects for assessment across a diversity of objects. Additionally, this data is subject to adversarial attacks through a severity factor to mimic common point cloud corruptions on synthetic data.

Figure 1.1 outlines our hybrid pipeline incorporating both point cloud and image data. The main modules of our proposed pipeline are numbered including 1) segmentation, 2) global image-guided registration, and 3) localization refinement using the proposed neural network or our optimization-based method.

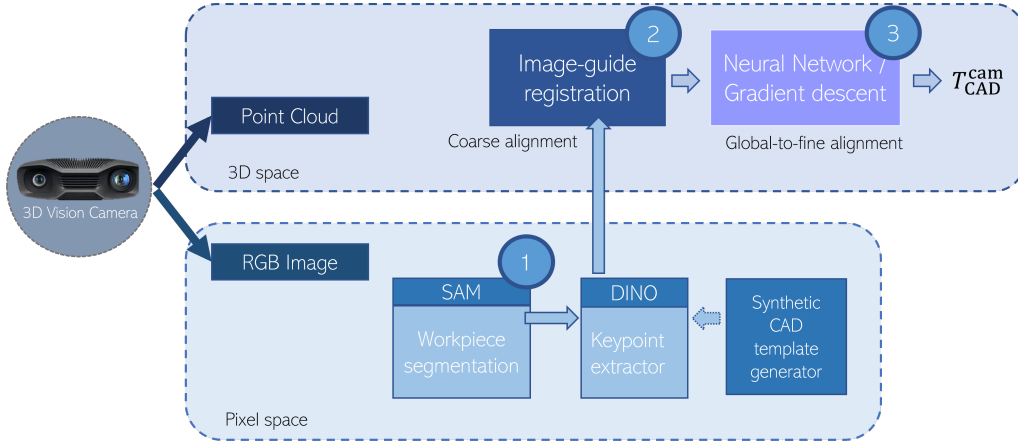


Figure 1.1: Big picture of our proposed registration pipeline.

The key contributions of this work are summarized, as follows:

1. A thorough review of recent learning-based registration techniques for industrial part localization.
2. Hybrid workpiece localization framework robust to real-world data corruptions with a novel registration overlap loss function.
3. Proof-of-concept application of SoTA vision models on point cloud registration tasks under partial workpiece scenarios.
4. Synthetic point cloud generation engine of industrial objects.
5. A comprehensive evaluation of our pipeline on real-world data, a large synthetic industrial dataset under common point cloud corruption, and a standard dataset.

1.4 Thesis layout

The remaining of this thesis is structured as follows. Chapter 2 presents a comprehensive review of recent point cloud registration techniques with a focus on 3D deep learning highlighting registration robustness and their feasibility in automated localization with real-data. Subsequently, chapter 3 presents a learning-based technique using vision models to aid coarse part localization by incorporating image data and a proposed registration neural network with joint global feature embedding. Next, chapter 4 introduces an optimization-based registration method targeted for fine registration with robustness against corruptions. Finally, chapter 5 concludes the thesis with a summary of completed work and future research directions.

Chapter 2

Literature review

2.1 Overview

This chapter presents the literature review of the two major realms of point cloud registration, optimization- and learning-based methods.

Given the advent of industrial AI and increasing emergence of learning-based registration methods, we elaborate a comprehensive review of recent data-driven approaches specifically oriented on object-centric registration that are found to be applicable for generalized workpiece localization of industrial objects.

Furthermore, we present a review of the current landscape with respect to optimization-based registration methods and current commercial solutions.

Finally, this chapter is summarized with the main key findings in the literature and research gaps that inspired the proposed solutions in this thesis.

2.2 Point cloud registration

CAD-based registration is central to automated workpiece localization. In our work, registration is the task of aligning two partially overlapping surface point clouds, referred to as the source (3D model) and target (workpiece scan), into a common reference frame using a rigid transformation. In some literature, localization is also referred to as workpiece pose estimation.

Typically, PCR comprises two steps, global and fine registration. Global registration is trivial with a close-form solution when point correspondences between the two point clouds are known. However, the correspondence matching step is a prevalent problem, especially under the low-overlap regime, with cross-source

data, e.g. due to the incomplete 3D scan data. Correspondence-free registration methods have been explored by researchers as an alternative approach [3].

Numerous registration methods have been proposed to tackle these problems. Huang et al. [4] present a comprehensive survey of various PCR methods. Here, we will survey literature related to the two main realms of registration, i.e., optimization-based and learning-based methods.

2.2.1 Traditional/optimization-based registration

Traditional algorithms such as the pioneering ICP, primarily used for local registration, assume outlier-free data and full overlap between the CAD model and scan. Thus, multiple ICP variants have been proposed to tackle issues such as density dissimilarity, outliers, and local minima. Generalized-ICP [5] merges ICP point-to-point and ICP point-to-plane. Go-ICP [6] was mainly developed to overcome the vulnerability of local-minima in the vanilla ICP. In this method, global alignment is achieved using a branch-and-bound (BnB) strategy which is appealing due to its theoretical global convergence. Nonetheless, even ICP variants tend to fail when there is a large initial transformation or a low overlap ratio between the two point clouds. [7]. Furthermore, global alignment may come at a higher computation cost and parameter fine tuning. As a further improvement of ICP, multi-scale ICP [8] adopts a strategy in which successive iterations of ICP with varying ball query thresholds are chosen to progressively refine the alignment. In addition, coarse-to-fine size voxel filters are applied in each step to introduce more geometrical details of the input point clouds. Principal Component Analysis-based registration (PCA) offers a coarse point cloud matching approach through the alignment of the principal axes and it has served as a foundation for modern techniques [9].

Other traditional approaches adopt a two-step procedure, correspondence searching and transformation estimation. Fast Point Feature Histograms (FPFH) descriptor matching with RANSAC + ICP stands as the go-to method for global

registration without relying on a rough initialization. However, the handcrafted nature of FPFH [10] obscures the registration power of this method when subject to cross-source data due to dissimilar point density, scale variation and inherent noise of real-world data. In [8], this approach was utilized for robust workpiece localization. Authors in [11] describe FGR, the Fast Global Registration algorithm. This method also serves as one of the go-to methods for cross-source registration. It leverages FPFH and alternating optimization to solve registration as a minimization problem, promising alignment of partially overlapping 3D surfaces without initialization and with fast computation time. [12] uses a certifiable algorithm to solve PCR based on correspondences under large outlier rates for real-time applications. In the literature, partially overlapping point clouds are frequently globally aligned using a randomized alignment or overlap ratio estimation approach. [13] presents an improved randomized alignment-based technique based on four-point congruent sets and an intelligent indexing strategy. However, the above-mentioned works are prone to failure in low-overlap and noisy environments. Moreover, the original CAD model may need to be manually modified to better match the overlapping regions and produce accurate registration results.

Probabilistic registration has also been an area of research. [14] offers implementations of stochastic models for PCR with even non-rigid capabilities. However, these methods tend to fail when tested on real-world data or point cloud sparsity.

From an optimization perspective, the global registration problem is characterized by a non-convex and complex solution space. With the advent of deep learning and the need for efficient training, differentiable optimization has gained significant attention leveraging existing deep learning libraries.

DGR [15] is a three-stage PCR network that, as the last step, proposes a gradient-based $SE(3)$ optimizer with a differentiable loss for pose refinement without neighbor search, allowing for fast computation. HOUV [16] —the winning method of the MVP’21 Benchmark [17], proposed a hybrid optimization algo-

rithm for global registration with unconstrained rigid transformation variables. [7] presents a robust alignment error metric for registration via minimization under the observation that in any aligned shapes, straight lines intersecting one shape will also intersect the other at the same points. All the above works, however, deal with the trade-off of accurate global alignment at the expense of high computation time.

2.2.2 Learning-based registration

Learning-based point cloud tasks have gained significant traction in recent years given the introduction of PointNet [18], an encoder capable of processing the inherent complexities of point cloud data (irregular, unstructured, and unordered). Since then, numerous learning-based PCR networks have been proposed remarking the following categories:

I. End-to-end: Point cloud representation learning has driven the development of end-to-end global feature-based registration networks. These methods take the two point clouds as direct input to the network and through globally pooled features a transformation estimation module predicts the alignment transformation, frequently in an iterative fashion.

PointNetLK [19] exploits PointNet [18] as a backbone and the Lucas-Kanade algorithm to estimate the alignment. Similarly, PCRNet [20] leverages a siamese PointNet-based network [18] and iteratively estimates the alignment using an MLP head. AttentionPointResnet [8] improves [20] by adding sequential self-attention layers in the PointNet encoder coupled with skip connections for robust global registration in the workpiece localization task. DCP [21] leverages DGCNN [22] for point cloud representation and an SVD head assuming one-to-one correspondences and full overlap to estimate the transformation. DeepGMR [23] leverages probabilistic registration by minimizing the KL-divergence of unregistered point clouds modeled as Gaussian Mixture Model probability distributions. Nevertheless, the assessment of these architecture overlooks real-world data, and they tend to fail

in challenging low-overlapping scenarios.

In this category, authors have also been in pursuit of addressing partial-to-partial registration. As an extension of [21], PRNet [24] introduces a keypoint detection step and estimates the alignment via SVD matrix. Similarly, [25] adds a hybrid keypoint selection. Furthermore, OMNet [26] -a network with overlap region prediction capabilities, converts the partial-to-partial registration to the registration of the same shape problem and offers a strategy to mitigate over-fitting issues in CAD-based registration by sampling points from the CAD model twice separately for source and target. FNet [27] improves [26] by implementing partiality perception ability in a dual branch network, for rotation and translation, and a multi-level point-wise feature interaction module. RPM-Net [28] tackles partiality and outliers using a Sinkhorn normalization layer and soft correspondences to estimate the transformation via a SVD head repetitively. In practice, SVD shows computational instability for low overlapping point clouds, and repeatable keypoint detection is challenging specially in low-overlap scenarios. In [29, 30], inlier/outlier mask networks are detailed. By separating the overlap prediction from downstream registration, it reduces the computational efforts required for training, and thus stands as an alternative to overcome the partial-to-partial registration problem.

II. Local feature matching: Feature matching registration estimates the alignment based on matched point correspondences using learned features and similarity metrics.

Previous research has focused on high-overlap scenarios neglecting the challenges of partial-to-partial registration leading to misleading high registration accuracy due to permissive criteria ($\geq 30\%$ overlap) [31] or reliance on unrealistic assumptions for practical applications. Therefore, researches have devised more complex methods to counteract registration with low-overlapping point clouds.

In Predator [31], an encoder-decoder network with an overlap attention block for co-contextual information exchange between the two point clouds is proposed

to tackle the low-overlap registration regime. The key idea is to learn to detect the overlap region and focus on that region to sample feature points underlying the importance of sampling points that are not only salient but also lie in the overlap region for subsequent registration steps. Qin Z. et al. introduced GeoTransformer [32] following [31] in the low-overlap regime. It introduces a transformer architecture that encodes transformation-invariant geometrical structure of the point clouds. These methods are deemed SoTA and show robust performance on-par with [33–37] when primarily evaluated on indoor and odometry datasets.

DGR [15] makes use of FCGF [35] to predict the confidence of inlier correspondences followed by a differentiable weighted Procrustes algorithm to provide closed-form alignment estimation. Hatem A. et al. proposed Point-TTA [38], an innovative framework based on a meta-learning scheme and a test-time adaptation via self-supervised auxiliary tasks to boost generalization and performance on existing learning-based methods of unseen partially overlapping point clouds. However, the above works are computationally heavyweight, substantial training resources are required hindering ready deployment, and sufficient superpoints are required to lie in the overlap region and can be disturbed by uneven point density. Also, these methods lack to incorporate visual information from image data.

Even though learning-based SoTA models have improved over time on clean data, they have been found to be less robust under corruptions [39] that are inevitable in real-world deployments.

A practical consideration is that memory (≥ 24 GB VRAM) and computational burden of DL methods increase significantly with the number of points. Therefore, a downsampled point cloud, generally constrained to a maximum of 1024 points, needs to be input to the network, dismissing important geometric characteristics. Whereas objects in ModelNet40 [40] can be effectively represented with 1024 points, free-form industrial objects require a higher number of points for accurate registration.

2.3 Sensor data fusion

Recent advances in registration methods have been primarily driven by the application of the LiDAR technology, e.g., in self-driving cars. However, these sensors output only the point cloud data without color information. In contrast, 3D vision cameras such as Zivid open up an array of possibilities as they provide both the point cloud data and the RGB color image of the captured scene in which each pixel is translated into one point in the cloud. One of these possibilities is data fusion by leveraging color information from images to aid the point cloud registration tasks.

Authors have pointed out that multimodal data fusion has the potential to provide significant advantages for point cloud registration [41]. In this regard, previous works [42] have proposed a 2D-3D sensor fusion strategy using Speeded Up Robust Feature (SURF) descriptors to estimate the object pose using 3D points corresponding to the image keypoint matches. Nevertheless, the nature of hand-crafted image descriptor renders them unsuitable for dynamic scenarios. To this end, advances in the computer vision field with open-source foundational models based on unsupervised training of vision transformers [43–45] pose a promising alternative for keypoint and feature matching that aids registration tasks and add perceptual abilities.

2.4 Robustness of point cloud tasks

With the aim to enhance the understanding of how common sensor corruptions detrimentally affect the performance of point cloud tasks and to provide insights into mitigating these drawbacks, authors [39, 46] have proposed systematic approaches to benchmark and analyze the perception robustness of point cloud methods under corruptions. Moreover, they have proposed corruption taxonomy and algorithms to recreate this noise as adversarial attacks on synthetic data to mimic sensor corruptions and bridge the prevalent synthetic-to-real data gap for learning-

based point cloud strategies.

[47] acknowledges the existing limitations of learning-based 3D registration methods and elaborates on the reasons they fail to work with real-world data. It offers a thorough analysis and best practices to enable robustness into registration baselines, and the results show that it can effectively improve 3D registration perception under corruptions.

To mitigate occlusion, voids, and partiality, multi-viewpoint cloud assembly has been utilized [8]. In this technique, multiple snapshots of the same object are taken from a different point of view and then stitched together in a global space (multi-way registration), followed by merging to relieve over-densified regions. Next-best-view (NBV) planning is needed for the fully automated process.

Notably, one-shot global localization stands as an efficient process to reduce the data acquisition time, allowing for rapid workflow on the shop floor and maximum productivity gains. It is also necessary when the workpiece size exceeds the camera field of view and robot reach, relying therefore on the partial visibility of the workpiece.

2.5 Registration validation

One way to enable robust registration is through quality control measures that can be utilized to validate the accuracy of the registration results. This step is crucial in current localization pipelines to ensure the reliability of the transformed data.

Traditional approaches have used point cloud similarity metrics and thresholding to discern correct alignment. However, the process of determining a threshold level is generally a manual and empirical process. Furthermore, traditional metrics such as the RMSE or Chamfer Distance (CD) are not reliable metrics due to the uncertainty of correspondences or are ineffective especially when the point clouds are complex, or the objects have predominant planar surfaces. To overcome ineffective metrics, a secondary registration method as a safeguard can be added when the primary algorithm fails to fetch accurate point correspondences, as determined

by SVD [15].

Recently, researchers have proposed new approaches for detecting misaligned point clouds as they acknowledge its importance in fully automatic point cloud matching [48]. In [49], a binary logistic regression classifier was developed to discern whether two point cloud are correctly aligned or not. Similarly, in [50], a deep learning architecture was proposed to validate accurate registration results as a binary classification problem. However, these studies concluded that the accuracy of data-driven classifiers across the evaluated datasets was not sufficiently high to fully rely on this technique.

2.6 Industrial object datasets

When it comes to data-driven registration and fair benchmarking, domain specific and curated datasets become vital. In the field of object localization for manufacturing, however, researchers have identified a prevalent lack of industrial object datasets. The MVTec ITODD dataset [51] (28 small industrial models) has been historically used in object localization problems. Another commonly used dataset is ModelNet40 [40]. ModelNet40 has become the de facto standard for the evaluation of object-centric registration methods in the point cloud community. However, it mainly consists of surfaces of household appliances and large-scale objects. The ABC Dataset proposed in [52] is a large-scale dataset with more than 1 million models. Although it is infrequently used in point cloud tasks, it remains a promising alternative for geometric deep learning as its primary intended use. Consequently, evaluating registration pipelines for industrial applications solely based on these datasets may become unrealistic even when uniform jittery noise is added since they may not represent the combined noise from real sensors.

Researchers have also introduced generative models as a way to produce new CAD models. In [53], a transformer-based network randomly generates 3D models based on a prior probability distribution learned from training data represented as a sequence of basic computer-aided design operations. However, data-driven

approaches still require initial training data to generate new sets, making them inadequate when initial domain-specific data is unavailable or when intricate free-form surfaces are desired.

2.7 Summary

Two main trends are noted among DL networks that promise PCR in partial scenarios, i) learning to detect the overlapping region and then convert the partial-to-partial problem into a same-shape registration problem in a two-stage manner, and ii) learning jointly the point-wise features and the overlapping keypoint selection along with the alignment tasks. Although models in the latter tackle registration in a direct way and have shown robustness, they are computationally heavy and not suitable for real production. Also, the memory usage scales with the number of points due to the processing nature of point cloud data. In this context, partially overlapping surfaces still pose a challenge while maintaining high generality and low memory usage. Hence, efficient architectures are needed to make these models accessible and feasible for industrial applications. Learning the alignment transformation from image and point cloud data stands as a promising alternative to relieve registration using a holistic representation of whole point clouds or point-wise descriptors.

On a different category of registration, feature matching and optimization-based registration tend to fail when repetitive and similar point cloud structures are present. For that reason, developing an optimization-based registration method with a robust loss immune to noise, partiality, and overlap maximization stands as a promising strategy to tackle the registration problem. Combining the merits of both optimization-based and deep learning methods for registration is a favorable approach [4] which is investigated systematically in this thesis.

Chapter 3

Learning-based registration

3.1 Overview

In this chapter, we analyze and propose a registration pipeline with a focus on deep learning strategies. We start with laying out the foundations and symbology of automated workpiece localization that will be used throughout this work. Next, we delve into our proposed learning-based pipeline comprising three modules. Each module performs a key task to aid and tackle registration. The first two modules leverage a pre-trained foundational computer vision model to incorporate image data into our registration pipeline. In our third module, we propose a novel deep learning architecture by leveraging point cloud global encoders and a principal cross-attention layer, followed by a transformation regression to provide the point cloud alignment.

Subsequently, we present the two main datasets used in this chapter for training and evaluation of the proposed network. For the training dataset, we introduce a rule-based point cloud generation engine that is able to produce point clouds using a random CAD model generator of industrial-looking objects. For the evaluation of the registration network, we present a real-world dataset of a sample composite part. Finally, we discuss the results obtained from the experiments performed with our proposed pipeline.

3.2 Vision-based automated part localization

In the field of robotics, the position of objects within the robot envelope is commonly expressed using a 4×4 rigid homogeneous matrices in the 3D space. Hence,

robotic workpiece localization means to find the 6-DoF position (R, t) , of the object (CAD model reference frame) with respect to the robot base, where $R \in SO(3)$ and $t \in \mathbb{R}^3$. Furthermore, workpiece localization mainly consists of three different tasks: workpiece segmentation, global pose estimation, and pose refinement. These three tasks are addressed in this work.

In this thesis, our robotic setup is comprised of an ABB IRB-1300-10/1.15 serial robot and an industrial 3D camera (Zivid Two) in an eye-in-hand configuration. See Figure 3.1. The camera’s working principle is based on structured light for imaging. The camera outputs highly dense 2.5D point clouds as well as a color image of the scene. Then, workpiece localization can be attained by solving a chain of transformation matrices featuring three matrices, as follows:

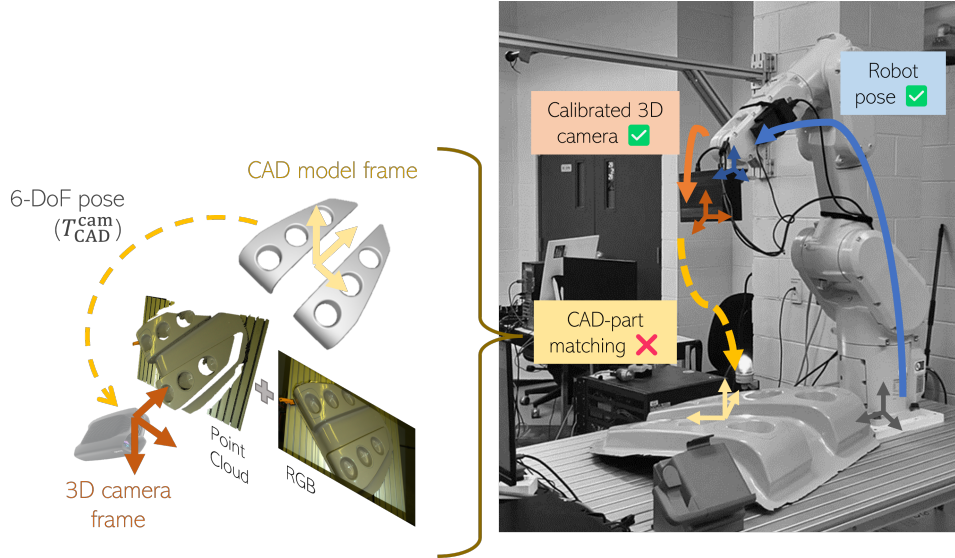


Figure 3.1: Robot-agnostic transformations for part localization. Chain of transformation involves the robot flange position, 3D camera mount position, and workpiece capture.

$$T_{CAD}^{base} = T_{flange}^{base} * T_{cam}^{flange} * T_{CAD}^{cam} \quad (3.1)$$

In this setting, the robot pose can be retrieved from the robot controller and the camera-to-flange relationship can be obtained through hand-eye calibration. Therefore, localization is the task of finding the T_{CAD}^{cam} matrix, i.e., the workpiece

position relative to the camera frame.

This missing homogeneous matrix, $T_{\text{CAD}}^{\text{cam}}$, is estimated through registration. Registration is the task of aligning two partially overlapping point clouds into a common reference frame. These point clouds are the 3D model represented as a set of points, namely, the source $\mathcal{S} \in \mathbb{R}^{M \times 3}$, and the actual capture of the workpiece from the camera known as the target point cloud $\mathcal{T} \in \mathbb{R}^{N \times 3}$. See Figure 3.2. $(R_{\mathcal{ST}}, t_{\mathcal{ST}})$ represents the rigid transformation from point cloud $\mathcal{S}$ to $\mathcal{T}$.

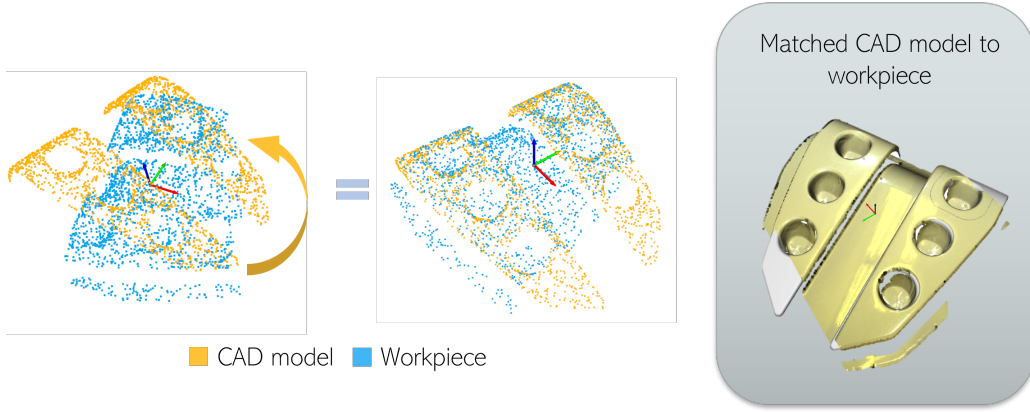


Figure 3.2: CAD-based point cloud registration.

For 3D scanning the workpiece, the robot moves to a pre-defined pose considering the optimal camera distance from the workpiece to get the best performance of the camera. The 3D camera serves as a feedback mechanism for the robot to adapt to the current workpiece setup. Throughout this work, we will use a sample composite part provided by the VTC network, shown in Figure 3.1, for the evaluation of our proposed pipeline. This test part features free-form surfaces, repetitive patterns, occlusion, and point voids to put our algorithms to the test.

Once Eq. 3.1 is solved, downstream operations like mapping pre-computed toolpaths from the CAD model to the actual location of the workpiece can be performed for automatic robot task execution. In this research, the following assumption is considered for the positioning of the workpiece.

Assumption 1 *The workpiece remains stationary and is placed arbitrarily within the robot workspace with a maximum deviation of $\pm 20^\circ$ around each rotation axis*

with respect to a nominal pose.

In the next section, we introduce the first module of our proposed pipeline for workpiece agnostic segmentation even in the presence of uncertainties such as fixturing devices and varying viewpoints.

3.3 Module 1: Workpiece segmentation

As the first module in our registration pipeline, we propose a learning-based workpiece-agnostic point cloud semantic segmentation step. To this end, we leverage SAM [54], the Segment Anything Model, a large pre-trained image segmentation model based on the vision transformer (ViT) architecture with our own adaptation on point cloud data, hence, *SAM-3D* in this work. The importance of accurate workpiece segmentation lies in passing clean point clouds free of the background remains to subsequent registration steps.

Let $I_{scan}(x, y)$ be the raw color image from the 3D camera (Zivid) of size $H \times W$, where x and y are spatial coordinates corresponding to the pixel locations. The model takes as input I_{scan} and a pixel coordinate as a prompt to select the workpiece. The pixel coordinate is computed as the silhouette mean of any object identified in the scene based on color or predominant spatial occupancy through thresholding. The model outputs a binary mask along with a confidence number signifying the correctness of segmentation. The mask, $M_{seg} \in \{0, 1\}^{H \times W}$, with the highest confidence is always selected, where 1 represents the object class.

On the 3D space, Zivid outputs a raw point cloud in a structured format $P_{raw} \in \mathbb{R}^{H \times W \times 3}$. This creates a straight-forward pixel-to-point between relation from which a segmented point cloud is obtained using element-wise multiplication as $P_{raw} \odot M_{seg}$. This process is visually outlined in Figure 3.3.

We evaluate the feasibility and segmentation accuracy of this method in Section 3.8. The following section provides the details of the second module of our proposed learning-based pipeline.

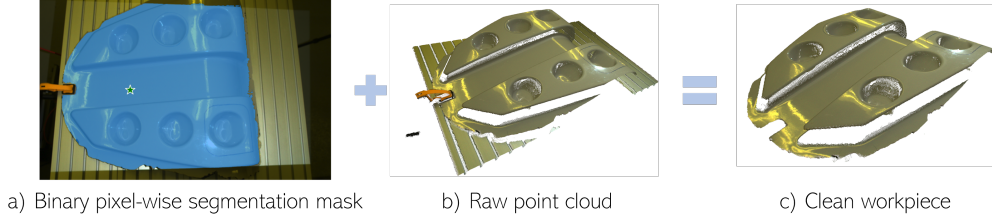


Figure 3.3: Point cloud workpiece segmentation from mask.

3.4 Module 2: Global registration via learning-based image keypoint matching

The objective of global registration is to provide an initial alignment between the CAD model and the workpiece scan. Global alignment is trivial if at least 3 sets of corresponding points that sit on non-tangential planes are known. Nonetheless, accurate keypoint matching still remains a challenge in complex scenarios under partiality. While workpiece localization is an effortless task for a human operator using visual information, this task is challenging for machines. Most currently used point cloud networks do not leverage image data. In this thesis, we designed module 2 to leverage the raw image from the camera, $I_{scan}(x, y)$, and extract keypoint information to guide the registration task.

To this end, we propose a global registration method based on keypoint matching. This method adopts a novel approach by leveraging image data of the scene and the 3D model through DINO [43] — a pre-trained foundational computer vision model— to output pixel correspondences that are then projected onto the 3D point clouds to obtain 3D matching points. The rationale behind this technique is to leverage high-level semantic information from images and enable image guidance into registration.

By doing so, we aim to demonstrate the feasibility and utility of deep image perceptual models and 2D image information to aid the global registration task, serving as a proof-of-concept. Large pre-trained models pose promising applications allowing us to generalize to diverse objects even without fine-tuning.

This method is functionally similar to workpiece localization via ArUco markers

[8] or manual picking of point correspondence but with the improvement that the workpiece keypoints are automatically selected and matched for global alignment without human intervention in the loop.

The proposed method consists of a series of steps: 1) Reference image generation, 2) image keypoint matching, 3) pixel to 3D point projection, and 4) alignment estimation. These steps are outlined in the diagram in Figure 3.4 and presented in greater detail below.

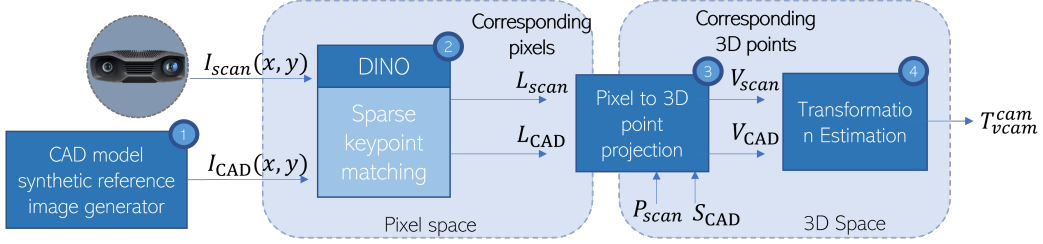


Figure 3.4: Outline of our global registration via learning-based image keypoint matching.

1) Reference image generation

First, in order to perform keypoint matching two images are needed; one is the capture from the camera and the second one is a reference image of our CAD model, $I_{CAD}(x, y)$. Hence, we synthetically generate an image representation of the 3D model from a single generic point of view, $I_{CAD}(x, y)$. This reference image of the CAD model is generated using a virtual camera pointing at the CAD model from a bird's-eye view powered by a simulated twin environment in RoboDK [55]. It is noteworthy that the relative position of the virtual camera with respect to the CAD model, T_{vcam}^{CAD} , may require human input and is frequently known in real world scenarios. Also, a rough estimate of the workpiece position is assumed to be known with respect to the robot base, T_{CAD}^{base} and the intrinsic parameters of the virtual camera are set to those of the real camera. Our implementation does not only provide the reference image but also the source point cloud $\mathcal{S}$ of the 3D model seen from the virtual camera position in a grid-like structure. In this scenario, the

chain of rigid transformations to find the CAD position with respect to the robot base is as follows:

$$T_{\text{CAD}}^{\text{base}} = T_{\text{flange}}^{\text{base}} * T_{\text{cam}}^{\text{flange}} * T_{\text{vcam}}^{\text{cam}} * T_{\text{CAD}}^{\text{vcam}} \quad (3.2)$$

By using registration, we estimate the transformation $T_{\text{vcam}}^{\text{cam}}$ that aligns the CAD model in the virtual camera frame to that of the real camera.

2) Image keypoint matching

Second, the image keypoint matching procedure is performed as presented in Algorithm 1 [43]. This algorithm defines the main steps to extract the pixel correspondence pairs. DINO employs the Vision Transformer architecture (ViT-big) with a patch size of 16 pixels as its main backbone and returns patchwise descriptors. The algorithm takes two RGB images as inputs to match sparse pixel keypoints as implemented in [44]. In our implementation, these images are the scan image, $I_{\text{scan}}(x, y)$, and the CAD image, $I_{\text{CAD}}(x, y)$. To reduce the processing load, these images may need to be scaled down to a lower resolution. Subsequently, the cosine similarity between all possible pairs of the descriptors of both images is computed in chunks. Lastly, the pixel location of the n most similar descriptors is retrieved back from the position of each patch descriptor in $L_{\text{scan}} = \{p_1, p_2, \dots, p_n\}$ and L_{CAD} , where these vectors hold the coordinates of the pixel keypoints of the scan and CAD model, respectively. Here, $L_{\text{scan},i}$ and $L_{\text{CAD},i}$ are corresponding points.

Figure 3.5 shows a set of 6 matched keypoints between the scan and the 3D model. Note that same-color points indicate the matching keypoints.

3) Pixel to 3D point projection

Third, once the keypoints in the pixel space are obtained, the corresponding 3D points are directly retrieved from P_{raw} at the same pixel location of the keypoints for the workpiece, and the same applies for the CAD model. By doing so, we obtain

Algorithm 1: Simplified sparse keypoint matching from DINO [43]

Require: $I_{\text{scan}}, I_{\text{CAD}}, n\text{Keypoints} = 6$,
 $I_{\text{scan}} \leftarrow \text{resizeAndNormalize}(I_{\text{scan}})$
 $I_{\text{CAD}} \leftarrow \text{resizeAndNormalize}(I_{\text{CAD}})$
Extract Descriptors:
 $\text{Descriptor}_{\text{scan}} \leftarrow \text{extractDescriptors}(I_{\text{scan}})$
 $\text{Descriptor}_{\text{CAD}} \leftarrow \text{extractDescriptors}(I_{\text{CAD}})$
Extract Salient Maps:
 $\text{SaliencyMap}_{\text{scan}} \leftarrow \text{extractSaliencyMaps}(I_{\text{scan}})$
 $\text{SaliencyMap}_{\text{CAD}} \leftarrow \text{extractSaliencyMaps}(I_{\text{CAD}})$
Compute Cosine Similarities:
 $\text{similarities} \leftarrow \text{chunk_cosine_sim}(\text{Descriptor}_{\text{scan}}, \text{Descriptor}_{\text{CAD}})$
Extract Mask of Best Pixel Matches:
 $\text{bbs_mask} \leftarrow \text{find_best_matches}(\text{similarities})$
Filter Best Matches:
 $\text{filtered_bbs_mask} \leftarrow \text{filter_by_saliency}(\text{bbs_mask}, \text{SaliencyMap}_{\text{scan}}, \text{SaliencyMap}_{\text{CAD}})$
Apply K-means:
 $\text{clusters} \leftarrow \text{kmeans}(\text{filtered_bbs_mask}, \text{Descriptor}_{\text{scan}}, \text{Descriptor}_{\text{CAD}}, n\text{Keypoints})$
Rank Pairs:
 $\text{top_pairs} \leftarrow \text{rank_pairs}(\text{clusters}, \text{SaliencyMap}_{\text{scan}}, \text{SaliencyMap}_{\text{CAD}})$
Extract Coordinates:
 $\text{corrPxScan}, \text{corrPxScanCAD} \leftarrow \text{get_coordinates}(\text{top_pairs})$

the vectors $V_{\text{scan}} = \{p_1, p_2, \dots, p_n\}$ and V_{CAD} , where these are the vectors holding the projected 3D points in space pertaining to the scan and 3D model, respectively. Given that P_{raw} may contain invalid points as NaN , in these instances, the nearest 3D point in the pixel space is retrieved from the original pixel location using a *cKDTree* for rapid search with a maximum search radius of 3 pixels.

Then, a higher-order filter based on triangle similarity is implemented to prune out outliers. The Euclidean distance between all possible pair of 3D points in V_{scan} and V_{CAD} are computed separately to obtain the pairwise distance matrices D_{scan} and D_{CAD} , each of size $n \times n$, as expressed in Eq. 3.3.

$$\text{Dist}_{i,j} = \|p_i - p_j\|_2 \quad \text{for } i, j = 1, 2, \dots, n \quad (3.3)$$

The absolute difference between the corresponding distances is computed to identify discrepancies between the two sets of points, as represented in 3.4.

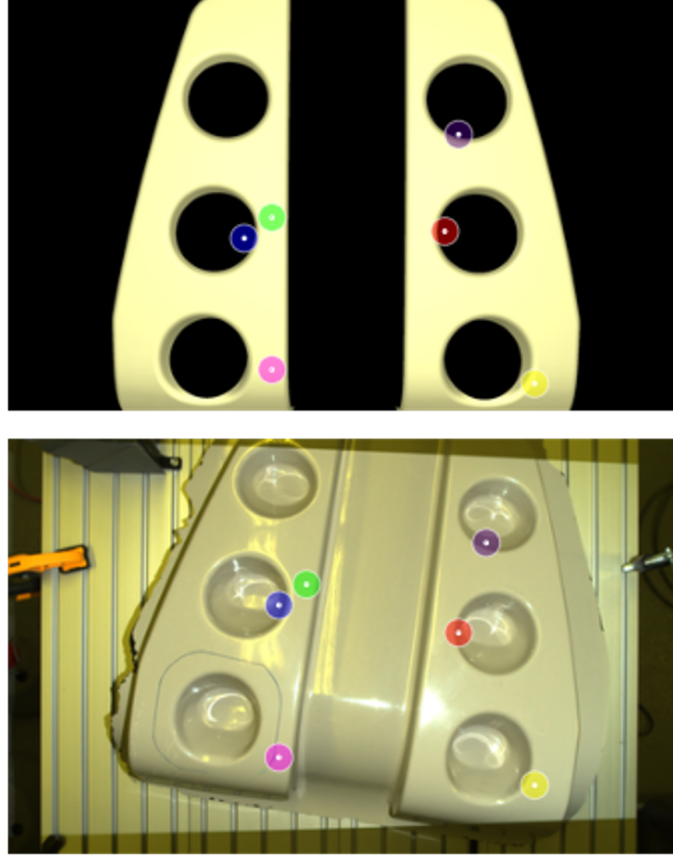


Figure 3.5: Matched keypoints between the CAD model (top) and the workpiece scan (bottom).

$$\text{Diff}_{i,j} = |D_{\text{scan},i,j} - D_{\text{CAD},i,j}| \quad \text{for } i, j = 1, 2, \dots, n \quad (3.4)$$

Then, a maximum allowable distance, α , is set between corresponding point pairs to reject outliers, and the inlier keypoint indexes are obtained as in Eq. 3.5, where $1\{\cdot\}$ is the indicator function.

$$\text{Inlier} = \{i \in \{1, 2, \dots, n\} : \sum_{j=1}^n 1\{\text{Diff}_{i,j} > \alpha\} \leq \text{max_exceeded}\} \quad (3.5)$$

A similar approach is used with triangle internal angles, although this time a combination of triplet points is used from V_{scan} and V_{CAD} . The internal angles of each triangle formed by the triplets are then computed and subtracted by their corresponding angles between those of the scan and CAD model. Similarly, a

threshold is employed to identify outliers.

4) Alignment estimation

In this step, the global alignment transformation is estimated from the filtered 3D corresponding points using the orthogonal Procrustes algorithm, as described and packed in [56] using the `TransformationEstimationPointToPoint()` function.

In the following section, we present the design of the proposed deep learning-based architecture in this pipeline.

3.5 Module 3: Registration network with joint point cloud encoding

Although image-guided registration provides an initial alignment transformation, a second registration stage is proposed to refine the alignment. In this third proposed module, we leverage the power of neural networks and deep learning to estimate the alignment transformation between the CAD model ($\mathcal{S}$) and the scan from the camera $\mathcal{T}$. The proposed architecture is illustrated in Figure 3.6.

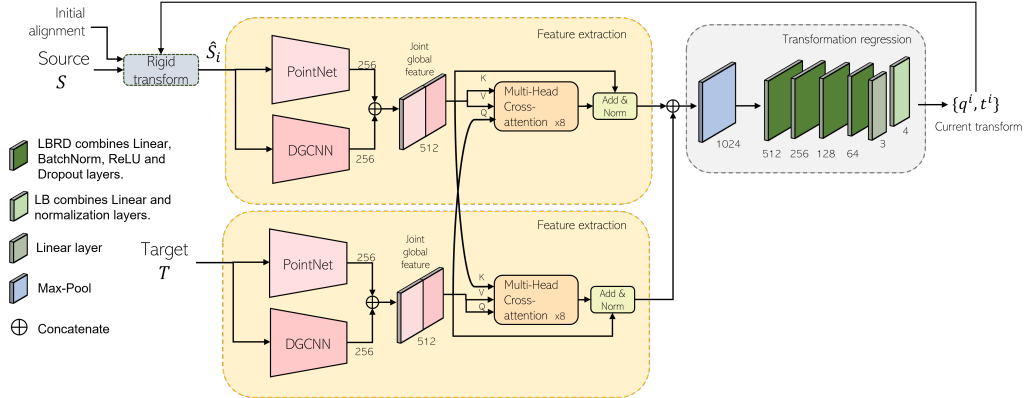


Figure 3.6: Architecture of our proposed registration neural network.

This network has been designed based on global feature alignment using a joint global point cloud feature, PointNet [18] and DGCNN [22]. Using joined features has been a recent technique in the literature introduced in [57]. To start,

a siamese network architecture is selected where the trainable weights are shared for the source and the target point cloud encoders. The source and target point clouds are input into the dual encoders each producing a per point feature with a dimension embedding of 256. The features are then joined using concatenation resulting in a per point feature space of size 512. Next, a multi-head cross-attention layer has been implemented to exchange the embedded information of the points between the source and target point clouds. The number of head was selected as eight for our network. Next, a concatenation operation is used this time to combine the source and target features followed by a max pool layer to finally obtain a global feature embedding per point cloud of dimension size 1024. Lastly, the global feature is passed through a multi-layer perceptron regressor to produce the translation and rotation components. The latter is expressed in the form of a unit quaternion following [24].

In this work, our contribution lies in the introduction of joint feature embedding followed by a cross-attention mechanism. In the latter, the scaled dot-product attention is employed as described in [58], where the Key and Values matrices are generated using the source while the Query is input from the target point cloud embedding, and vice versa, allowing for information exchange. By inputting per point embedding into the cross-attention mechanism we allow the network to attend each point of one point cloud with the rest of the counterpart point cloud. Additionally, this mechanism is adequate for point cloud data due to its permutation invariant characteristic. For point cloud registration networks, a common practice is to iteratively refine the alignment by passing the input point cloud repeatedly through the network. In this work, we have adopted the same strategy as in [20] using 4 iterations.

For training, we have chosen the Chamfer distance metric and the Frobenius Norm for our loss function to be minimized. The Chamfer distance is defined in Eq. 3.6, where p and q are points in the source and target point clouds, and $|\mathcal{S}|$ and $|\mathcal{T}|$ are the number of points in each of them, respectively.

$$\mathcal{L}_{CD}(\mathcal{S}, \mathcal{T}) = \frac{1}{|\mathcal{S}|} \sum_{p \in \mathcal{S}} \min_{q \in \mathcal{T}} \|p - q\|^2 + \frac{1}{|\mathcal{T}|} \sum_{q \in \mathcal{T}} \min_{p \in \mathcal{S}} \|p - q\|^2 \quad (3.6)$$

While the Chamfer distance operates directly on the point cloud data, the Frobenius Norm utilizes the ground truth and the predicted transformation matrices to guide the learning of the network into the right alignment pose. The Frobenius Norm loss function is defined in Eq. 3.7.

$$\mathcal{L}_{FrobeniusNorm} = \lambda * |T_{\mathcal{ST}}^\top T_{\mathcal{ST}}^* - \mathbf{I}|^2, \quad (3.7)$$

where $T_{\mathcal{ST}}^\top$ is the predicted transformation, $T_{\mathcal{ST}}^*$ is the ground truth, $\mathbf{I}$ is the identity matrix, and λ is a constant typically set to 16.

In order to address the lack of sufficient industrial object datasets for training, we propose a synthetic point cloud generation engine as presented in Section 3.6. We generated 10,000 random models for the training of this network. Moreover, common point cloud corruptions were applied as a data augmentation technique to mimic real-world dataset.

Training deep learning networks for point clouds is a memory-intensive and computationally heavy task. Therefore, input point clouds must be down-sampled significantly, e.g., to a maximum of 1024 points. To capture most of the important point cloud structures, we have sampled points in the CAD model using the farthest point sampling strategy presented in [24]. By using synthetic data for training, the ground truth can be generated automatically. Hence, we chose a random initial misalignment of the source point cloud in the range of $\pm 20^\circ$ using Euler angles, and a translation of ± 0.2 in a normalized scaled where the point cloud lies within a unit radius sphere.

We have trained our network for 300 epochs using an Adam optimizer with an initial learning rate of 0.05, and an ExponentialLR scheduler with a decay constant of 0.998. The training of the network and experiments have been performed using an NVIDIA GeForce GTX 1650 GPU. Our network has 3,436,999 trainable

parameters.

As discussed, one of the contributions of this thesis is the introduction of a synthetic generation engine to create a point cloud of industrial-looking objects. Section 3.6 delves into the procedure developed for random point cloud generation as well as the cloud corruption patterns as a severity factor that form the dataset for the training of the proposed network and for benchmarking in the subsequent chapter. Section 3.7 details the registration metrics used for the evaluation of our pipeline.

3.6 Datasets

3.6.1 Synthetic data generation engine

In this section, we describe the procedures employed in the development of our synthetic data generation engine. Additionally, we introduce algorithms to replicate point cloud corruption patterns to bridge the synthetic-to-real gap. The purpose of this engine is to create an extensive domain-specific dataset of industrial objects for training and evaluation of our proposed methods.

The engine was developed following a rule-based approach using the PythonOCC library [59]. By using computer-aided-design operations we build free-form surface objects that possess similar characteristics to those found in composite parts. The 3D models are exported to the STL format and then converted to point clouds using the Uniformly point sampling method from the 3D mesh (`sample_points_uniformly()` in Open3D [56]). The CAD models are generated randomly using the following steps. Sample models generated by the engine are shown in Figure 3.7.

- **Step 1:** As the first step, we outline a 2D geometry in the xy plane that will serve as the base contour of the workpiece. This geometry is arbitrarily selected among various shapes including pentagonal, rectangular, rectangular with a circle on one side, rectangular with double circles, rectangular with rounded corners, circular, annulus, triangular,

and trapezoidal. Each of these shapes is randomly sized using random controlled variables.

- **Step 2:** Subsequently, in order to create a free-form surface, two curve profiles are outlined in the orthogonal planes xz and yz . For the curves, we use an interpolated B-Spline with 6 knots, linear and quadratic functions with arbitrary coefficients. A B-Spline surface is then generated from the two contiguous profile curves which form its boundaries.
- **Step 3:** Next, the common geometry between the 3D surface and the xy base contour is obtained. Once the 3D surface is created, an L-shape wire with a fillet is drawn at one point of the surface contour and then swept to create a flange contouring the workpiece.
- **Step 4:** The 3D surface and flange are stitched together to create a single object.
- **Step 5:** Finally, in addition to the flange, in order to increase object diversity, the generated surfaces may be mirrored based on a primary axis to generate a new category of semi-symmetric objects. Notably, in all of the above-mentioned steps extensive conditioning and error-proof checking have been implemented to ensure the generated models have smooth double-curvature surfaces and the objects are free of geometric errors.

Figure 3.7 depicts various examples of randomly generated models using our engine.

3.6.2 Synthetic partiality and corruptions

A Severity Factor (SF) is introduced to mimic real-world point cloud corruptions on synthetic data. The factor is discretized on a scale from 1 to 5, in which 5 means high corruption aggressiveness. Although real-world point cloud corruptions

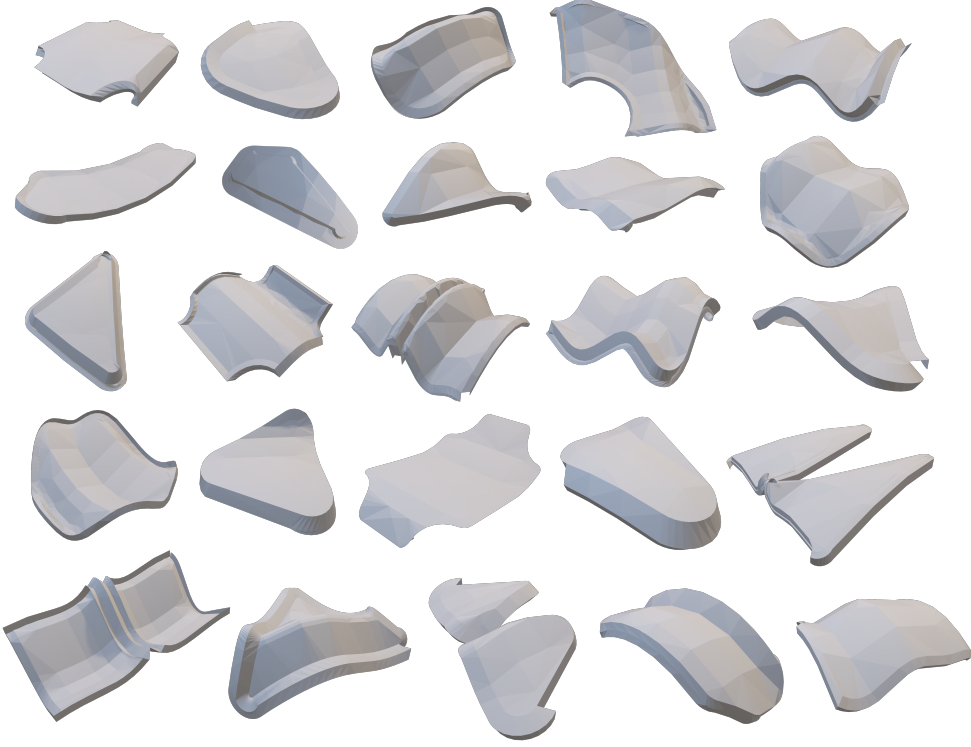


Figure 3.7: Free-form models generated synthetically.

have a rich variation and specific probability distribution, the presented corruption patterns provide a practical solution to analyze registration robustness, similar as in [46] for classification tasks.

The corruptions can be divided into two categories, i.e., density and noise corruption patterns [46]. The corruptions are applied in a combined fashion on the synthetic data, as explained below.

Density:

- **Camera viewpoint:** We sample a full 3D model from different viewpoints to generate partially overlapping data and mimic self-occlusion. Open3D `hidden_point_removal()` [56] function is employed for this task.
- **Random crop with plane:** We crop a point cloud with a random plane and keep a specified ratio r of points, where $r \in \{0.95, 0.85, 0.75, 0.55, 0.45\}$

for the five severity levels. This aims to replicate partially overlapping point clouds, for instance, when large objects exceed the camera’s field of view. As opposed to viewpoint, this pattern allows to have control over the number of points that are maintained.

- **Cutout:** We drop random globally picked chunks of points, where a is the number of chunks and b is the number of removed points per chunk. $(a, b) \in \{(2, 20), (3, 30), (5, 30), (7, 30), (10, 40)\}$ for all five levels. This aims to simulate missing local parts of the object due to specular voids.

Noise:

- **Gaussian noise:** We add a Gaussian noise $\phi \sim \mathcal{N}(0, \sigma^2)$ to each coordinate of the points, where $\sigma \in \{0.010, 0.015, 0.02, 0.025, 0.03\}$ and clipped to $[-0.05, 0.05]$ for the five levels.
- **Uniform noise:** We add uniform noise $\phi \sim \text{Uniform}(-a, a)$, where $a \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ for the five levels.

An illustration of the corruption patterns on a point cloud is shown in Figure 3.8.

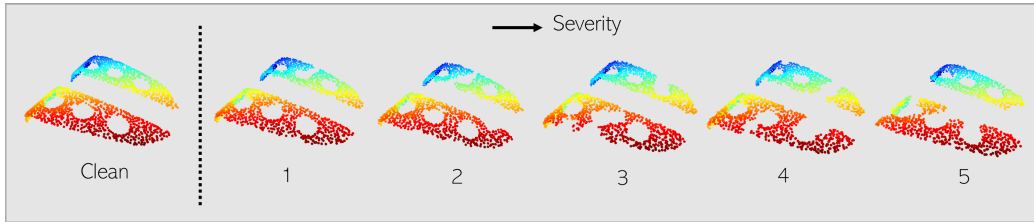


Figure 3.8: Cutouts and Gaussian noise on the test case 3D model.

Two datasets are used in this chapter: 1) 10,000 sample point clouds generated by our synthetic engine with common point cloud corruptions to train our proposed network, and 2) the sample composite part dataset to evaluate our registration pipeline. The composite part dataset consists of 40 actual Zivid Two snapshots of the test part and serves as the benchmark for real-world data. These datasets are summarized in Table 3.1.

Table 3.1: Datasets for training and testing of the proposed registration network.

Dataset	Real data	Num of samples	Corruptions/ partiality	Ground truth
Composite part dataset (testing)	✓	40	✓	Manual+GICP
Synthetically generated (training)	No	10,000	Synthetic	Synthetic

Some sample images of the composite part dataset including full and partial captures are shown in Figure 3.9.

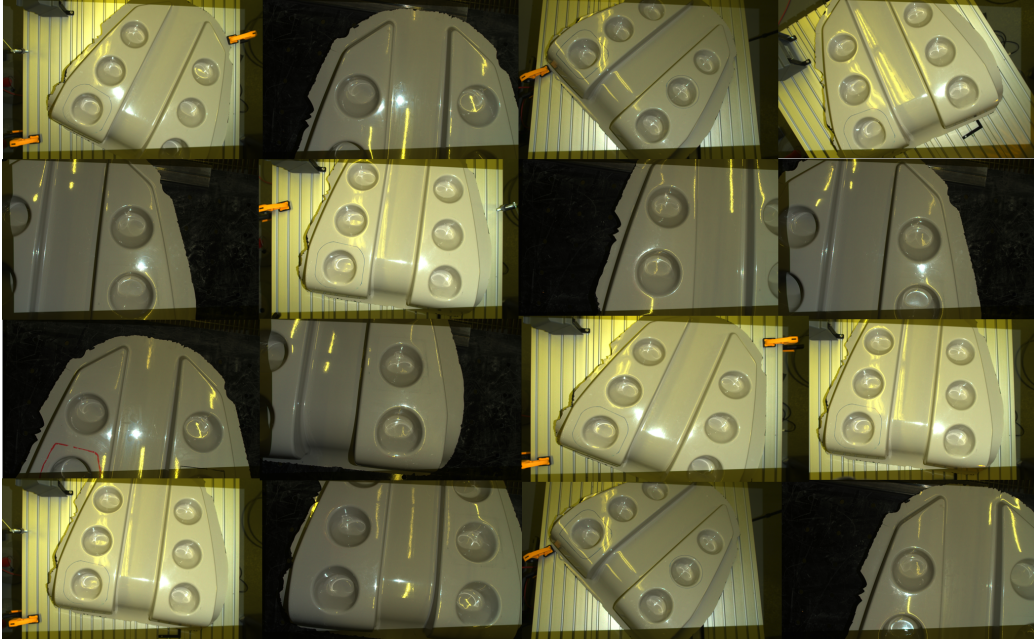


Figure 3.9: Sample captures in the composite part dataset.

3.7 Registration evaluation metrics

In workpiece localization, we are primarily interested in the alignment correctness, accuracy, and runtime. Hence, the registration evaluation metrics are chosen as registration success rate (SS), also known as recall, RR, isotropic and anisotropic rotation and translation errors.

Registration success rate is the ratio of successful alignment whose rotation error and translation error fall within some thresholds. See Eq. 3.8, where $\mathbf{T}_{est}$

and $\mathbf{T}_{GT}$ are the estimated and ground truth transformation, N is the number of times a registration test is performed, and $f(\cdot)$ is the isotropic errors with a threshold θ for translation and rotation. These thresholds have been empirically chosen as 1.6 and 2.3 for the translation and rotation error, respectively, which ensure a correct and tight alignment.

$$SS = \frac{1}{N} \sum_{i=0}^N 1\{f(\mathbf{T}_{est}, \mathbf{T}_{GT}) < \theta\} \quad (3.8)$$

Following [28], we report the mean of the isotropic rotation and translation errors which are expressed in Eq. 3.9 and Eq. 3.10, respectively.

$$\text{Error}(\mathbf{R}) = \frac{180}{\pi} \arccos \left(\frac{(\text{Trace}(\mathbf{R}_{est} \mathbf{R}_{GT}^\top) - 1)}{2} \right) \quad (3.9)$$

$$\text{Error}(\mathbf{t}) = \|\mathbf{t}_{GT} - \mathbf{t}_{est}\|_2 \quad (3.10)$$

where $\mathbf{R}_{est}$ and $\mathbf{R}_{GT}$ are the estimated and ground truth rotations, $\mathbf{t}_{est}$ and $\mathbf{t}_{GT}$ are the estimated and ground truth translations, and $\|\cdot\|_2$ is the Euclidean norm.

The anisotropic errors are defined as the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) of the rotation in its xyz degree Euler representation and translation vectors between the estimated and the ground truth, as stated in Eq. 3.11 and 3.12, respectively [21].

$$\text{RMSE}(\mathbf{t}) = \sqrt{\frac{\sum_{i=1}^N \|\mathbf{t}_{GT} - \mathbf{t}_{est}\|^2}{N}} \quad (3.11)$$

$$\text{MAE}(\mathbf{t}) = \frac{\sum_{i=1}^N \|\mathbf{t}_{GT} - \mathbf{t}_{est}\|}{N} \quad (3.12)$$

3.8 Results and discussion

To assess our registration method comprising the three modules described in this chapter, we have evaluated the proposed pipeline on a real-world dataset of the

sample composite part. The registration results are reported and analyzed below.

Evaluation of module 1 (segmentation)

The mean IoU as expressed in Eq. 3.13 was utilized to evaluate the segmentation performance of SAM 3D on the 40 sample images of the test part. In this equation, target are the manual ground truth masks of the composite workpiece in the pixel space.

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{\text{target}_i \cap M_{\text{seg } i}}{\text{target}_i \cup M_{\text{seg } i}} \quad (3.13)$$

Our evaluation shows an $mIoU$ of 96.14% which demonstrates the perception performance to distinguish the workpiece from the background and even exclude clamps from the segmentation mask without fine-tuning and the potential to yield background-free point clouds for the downstream registration steps. This represents a direct advantage compared to rudimentary approaches like box Region of Interest (ROI) cropping.

Evaluation of module 2 (sparse keypoint matching)

The robustness of the keypoint matching step was assessed based on the matching accuracy. Accuracy is defined as the ratio between the number of correct corresponding points and the total number of keypoints (n). We performed a series of experiments using the images of the composite part dataset with randomly sampled rotation in the range ± 20 deg and translation of ± 300 px as image augmentations to improve our results. The testing involves both the captures of the workpiece with a high overlap and captures with partial visibility (approximately 40% overlap).

Our results show an average matching accuracy of 83.32% in the sample images where the composite workpiece is captured in nearly its entirety as in Figure 3.5. This inlier accuracy is sufficient to estimate a robust global alignment between the scan and the 3D model. However, while the average matching accuracy

remains high in the high-overlap regime, the accuracy drops substantially to less than 16.67% when only a minimal partial section of the workpiece is matched against the reference full 3D model or when the image rotation varies significantly from the reference image. That is, this approach using the DINO model shows high sensitivity in the detection of accurate pixel correspondences to the relative rotation in the z-axis between the scan image and the 3D model. This accuracy decrease is due to the lack of common global context between the two input images and similar object features across the same workpiece, e.g., the cup shapes in the sample test part.

Given that the quality of the global alignment is directly impacted by the accuracy of the correspondences, the third module of this pipeline adds a second layer of registration aiming to overcome these previous challenges. Hence, the registration metrics will be reported on the results from the neural network using the result of the keypoint matching module 2 as the first initial guess.

Evaluation of module 3 (registration network)

After the training of the neural network presented in Section 3.5, we assessed our results using the registration metrics on the composite part dataset. This aims to benchmark the ability of our model to generalize to real-world data while trained on the synthetically generated dataset mimicking point cloud corruptions. The registration results are reported in Table 3.2.

Table 3.2: Test results and comparison of the proposed network on the sample workpiece dataset against existing learning-based baselines.

Model	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
iPCRNet	31.928	24.776	0.662	0.566	45.999	1.101	0.000
PRNet	5.728	3.581	0.115	0.082	8.296	0.165	0.109
PRMNet	23.311	8.730	0.331	0.160	18.609	0.322	0.067
Ours	4.701	2.634	0.070	0.042	5.491	0.088	0.361
Ours+ICP	4.333	2.455	0.076	0.043	5.109	0.090	0.410

Our results show that our network has a degree of generalizability to the test workpiece to be able to bring the source point cloud into a position where the

alignment is correct by visual inspection, although the success rate is low due to not being able to reach a tight alignment, hence a fine registration step such as ICP or any of its variants may be needed. Two sample qualitative results are shown in Figure 3.10.

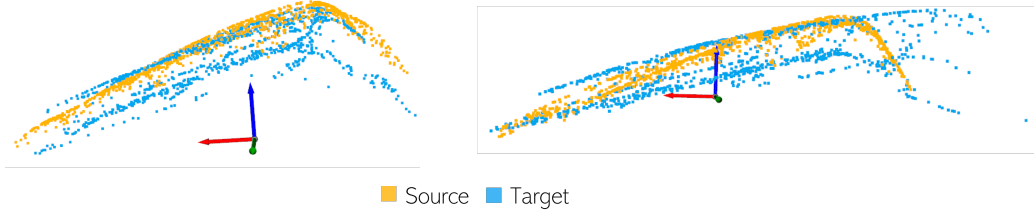


Figure 3.10: Qualitative results of our learning-based registration pipeline with a downsampled point cloud of the test part dataset.

It can be observed that even though a quasi-optimal alignment is attained, a tight fit is not guaranteed using a deep learning-based pipeline. Since ICP has known limitations when dealing with partially overlapping point clouds, applying a vanilla ICP potentially deviates the global alignment attained by the learning-based pipeline. It can be observed in Table 3.10 that even after using ICP as a refinement step, the registration success rate does not significantly increase. Such an effect may be caused by local minima close to the tight alignment pose, or the optimal solution by ICP does not match the one from visual inspection. Hence, a proposed coarse-to-fine alignment method based on gradient descent optimization is introduced in the next chapter to overcome the fine alignment issue presented here. Also, a comparison of the results and an in-depth analysis of our proposed optimization-based method are covered in the next chapter.

3.9 Summary

In this chapter, we described our proposed deep learning-based registration pipeline composed of three modules: 1) workpiece segmentation, 2) image keypoint matching, and 3) registration network. Additionally, two datasets were proposed for the training and evaluation of our pipeline. We introduced a rule-based synthetic

point cloud generation engine to produce 10,000 point clouds of industrial objects for training our network, and a real-world dataset of a composite test case for benchmarking on real-world data.

Our pipeline leverages two foundational computer vision models, SAM and DINO for accurate workpiece agnostic segmentation and image-guided registration by mapping pixel keypoints between the actual scan and the CAD model into 3D points to estimate the alignment transformation based on correspondences. Lastly, a second registration module was developed using a trained neural network based on point cloud encoders and multi-head cross-attention. Our results show a registration success rate of 41% on the sample workpiece dataset. While the global alignment attained with our learning-based pipeline is correct, the pipeline fails to attain a tight alignment, leading to a low registration success rate. Hence, in the next chapter, we propose an optimization-based registration strategy to overcome this challenge and achieve coarse-to-fine registration.

Chapter 4

Optimization-based registration

4.1 Overview

In this chapter, we propose a global-to-fine optimization-based registration strategy based on gradient descent. We call this method **UPGReg**, Unsupervised Parallel Gradient-based SE(3) Registration. We leverage deep learning to design a novel fully differentiable loss function that operates on a batch of point cloud data to maximize the overlap for tight alignment. Previous experiments with registration networks served as the foundation and key ideas to develop this novel strategy. While this method was initially developed to solve the fine alignment challenges commonly faced with learning-based registration pipelines, as discussed in the previous chapter, our results and experiments demonstrate that **UPGReg** has the potential to successfully perform registration even from a global manner. Moreover, **UPGReg** could be integrated into our learning-based pipeline as an interchangeable block with our Module 3, making our registration strategy hybrid by combining learning and optimization-based approaches. Solving the PCR from an optimization perspective provides deeper insights and toolsets to tackle the problem to complement our learning-based pipeline.

4.2 Optimization-based global registration

In this thesis, we propose **UPGReg**, a gradient-based optimization method with continuous variables to tackle the robust registration task free of correspondences. This method offers the merits of generalization to the diversity of objects without parameter fine-tuning and a novel differentiable F1-Overlap metric that shows

robustness to noise and point cloud corruptions of real-world data. Although learning-based methods have been on the rise, the industry still tends to favor deterministic solutions due to their predictability. Moreover, optimization-based registration has shown potential in solving complex scenarios under partiality in the literature [16].

We leverage insights and observations from our own experimentation on end-to-end learning-based registration networks and the previous research [47] to lead to the development and key ideas behind UPGReg, as follows:

- **Deep-learning machinery:** From a fundamental point of view, the training of deep neural networks is deemed a highly non-convex and multi-dimensional optimization problem conditioned on the training samples, and so is the 6-DOF registration problem conditioned to the input point clouds. Hence, we leverage the *auto-differentiation* capabilities of deep learning frameworks and *optimizers* to design a custom registration cost function to minimize the alignment error between two point clouds with respect to the rotation and translation parameters.
- **Parallel processing:** We use GPU acceleration for a multiple initialization of the optimization variables into sub-spaces in the pursuit of a collective optimal global solution.
- **Iterative approach:** Existing end-to-end deep-learning registration techniques employ the Chamfer distance as an unsupervised metric and rely on an iterative approach with inner-loops to find the best alignment, the same nature as gradient-descent.
- **Validation of unsupervised alignment loss function:** The initial development of this method was to experimentally validate the convergence of the proposed or existing unsupervised registration loss functions for optimal point cloud alignment with one training sample and let it overfit to one single inference sample, making this approach a differentiable optimization

solution.

With these ideas, we transition towards an optimization-based strategy while harnessing the capabilities of a deep-learning machinery.

The flowchart in 4.1 summarizes the steps of our optimization-based PCR strategy. This schematic formulates the PCR problem as a global optimization problem and leverages Graphics Processing Units (GPUs) to solve it via gradient descent with parallel seeds. In this context, a seed is one different initial alignment from which registration is started. This algorithm for pairwise registration will be used in all subsequent experiments.

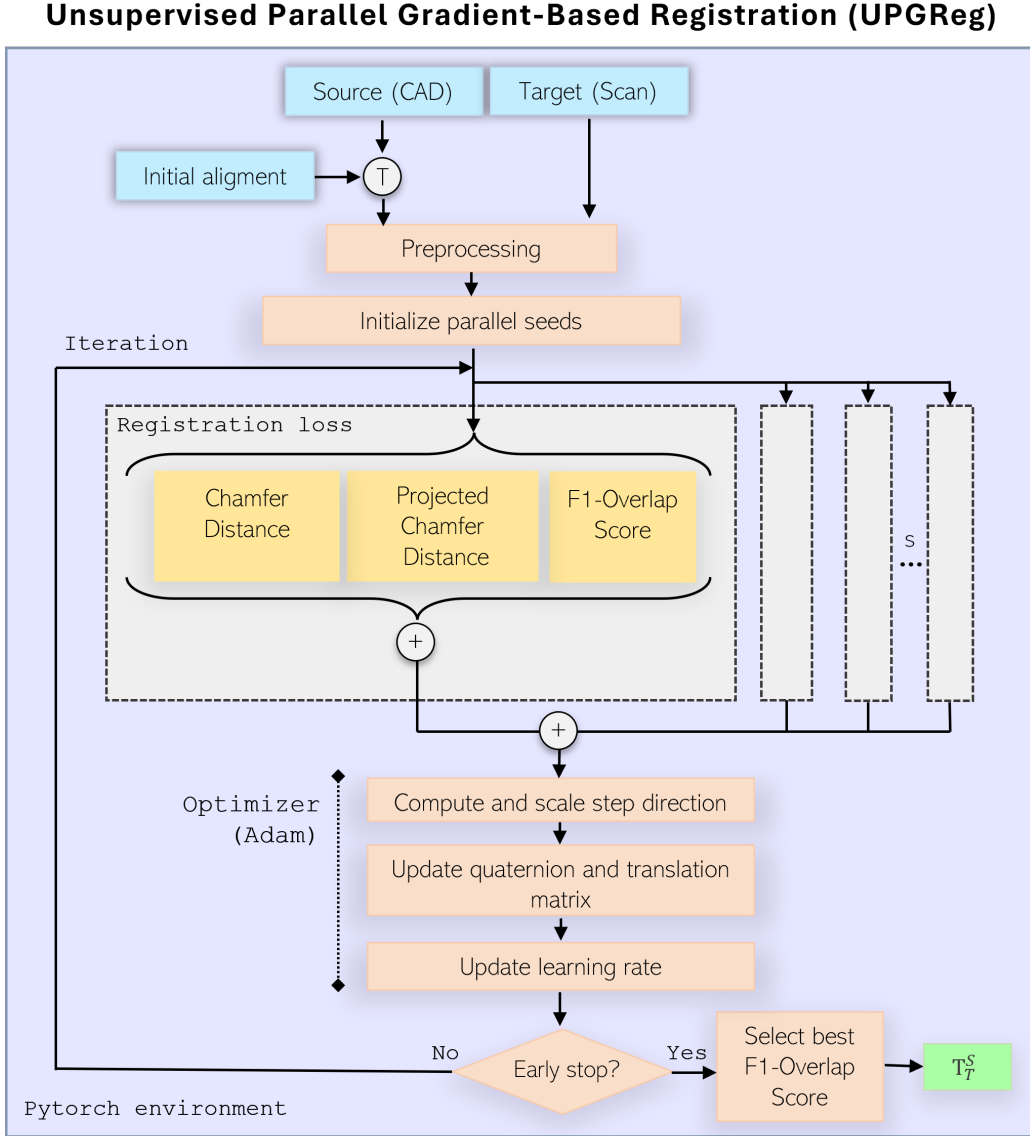


Figure 4.1: Overview of our proposed gradient-based registration method.

UPGReg takes as input the source ($\mathcal{S}$, CAD model) and target ($\mathcal{T}$, workpiece scan) points clouds, a series of hyperparameters (θ), and optionally an initial global alignment transformation such as the one provided from our learning-based keypoint detection module in Section 3.4.

$$f(\mathcal{S}, \mathcal{T}, \theta) \rightarrow \mathbf{T}_{\mathcal{T}}^{\mathcal{S}} \quad (4.1)$$

Then, using a gradient-descent algorithm [60], it seeks to minimize a robust

registration loss function given the input point clouds. In this step, backpropagation is run to compute the gradients of the registration loss with respect to the optimization parameters, allowing for their iterative update until convergence.

UPGReg is fully implemented in PyTorch [61] and leverages batched point cloud operations from Pytorch 3D [62]. Due to the non-convex nature of the registration solution space, the *Adam* optimizer is employed to improve the convergence of the optimization to a global optimal and escape from local minima or saddle points. Additionally, a *LambdaLR* learning rate scheduler is used to smoothen the convergence process as the optimization iterations elapse.

The initial learning rate, lr , of each parameter group is set to 0.055, and it decays as given by the expression in Eq. 4.2. A decay constant of 0.9989 is used. In our experiments, the maximum number of iterations is set to 500.

$$lr_{updated} = lr_{initial} * \lambda^{iteration} \quad (4.2)$$

At the end of each iteration, a stopping criterion is checked, and lastly, the gradients are zeroed out for the next iteration. Our Pytorch implementation of UPGReg opens up new possibilities to test other arbitrary custom loss function terms for registration.

4.2.1 Optimization variables

UPGReg utilizes 7 optimization variables per seed to represent the optimal alignment transformation $\mathbf{T}_{\mathcal{ST}}$. On the one hand, the rotation part of the transformation is represented in the form of a quaternion, $\mathbf{q}$, in the format *wxyz*, and constrained to a norm of 1. On the other hand, the translation is given as a vector $\mathbf{t}$. Therefore, in a multi-seed scheme, we have, $\mathbf{q} \in \mathbb{R}^{s \times 4}$ and $\mathbf{t} \in \mathbb{R}^{s \times 3}$, where s is the number of seeds employed.

All the quaternions are normalized after each iteration to fulfill the unit vector constraint, as follows:

$$\mathbf{q} = \frac{\mathbf{q}_{\mathcal{S}\mathcal{T}}}{\|\mathbf{q}_{\mathcal{S}\mathcal{T}}\|} \quad (4.3)$$

As the optimization parameters are iteratively updated, after each iteration $\mathcal{S}$ is transformed using the updated parameters into $\mathcal{S}'$. The transformed source point cloud is then used in the evaluation of the loss function in the next iteration.

$$\mathcal{S}' = \text{quat_2_mtx}(\mathbf{q}) \times \mathcal{S}^\top + \mathbf{t} \quad (4.4)$$

In the next sections, subsequent steps of the flowchart in Figure 4.1 are covered in greater detail.

4.2.2 Data preprocessing

As the first step for preprocessing, the input point clouds go through a series of operations in preparation for registration, as follows:

1. **Downsampling:** The raw target point cloud is first downsampled using a voxel-based filter of size V and then, 2048 points are randomly drawn using a uniform probability distribution to regularize the point cloud density and fix the number of points. Similarly, 2048 points are uniformly sampled from the 3D CAD model.
2. **Outlier removal:** We utilize a statistical outlier filter on the target point cloud using the `statistical_outlier_removal()` technique presented in Open3D [56].

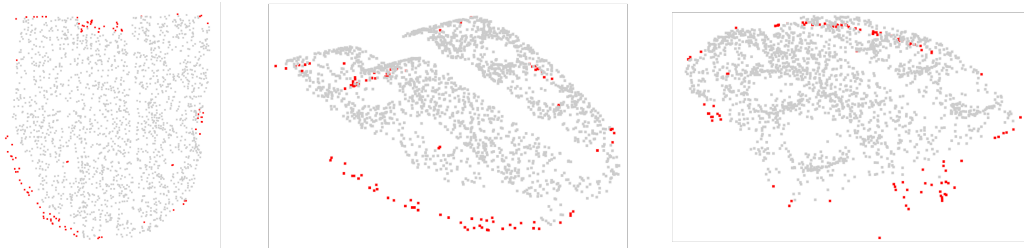


Figure 4.2: Automatic statistical outlier removal. Red points denote outliers in the sample composite part.

3. **Normalization:** The point clouds are zero-centered, and the target is normalized to fit into a unitary radius sphere. For optimization, this aims to enhance convergence to global optima by allowing the gradients to flow in all possible directions within the solution landscape. Additionally, through normalization, we bring objects of arbitrary size to a standard scale alleviating the need for constant parameter fine-tuning. The source point cloud is also normalized using the same scale factor as the target point cloud.

4.2.3 Multi-seed initialization

As indicated previously, one of the key ideas of our optimization method is to harness the vectorization power and accelerated computing with GPUs from existing deep learning machinery.

Hence, our method seeks to find the optimal alignment collectively by utilizing multiple initialization instances called seeds. This way, the optimization, i.e., the transformations associated with the seeds, are initialized at multiple locations within the optimization landscape improving the likelihood to converge to a global optimum. In a single iteration, all the seeds are simultaneously optimized. Alternatively, the optimization can also be executed using one seed. In related studies, the seeds are also known as kernels [16].

While all the translation vectors are initialized with zeros having previously zero-centered the input point clouds, the quaternions, $\mathbf{q} \in \mathbb{R}^{s \times 4}$, take a distinct initial value in each seed. These are initialized using an xyz Euler angle representation, $\mathbf{e} = (e_x, e_y, e_z)$, with rotation variations in the y- and z-axis. This selection is made based on the observation that the variance of the rotation of the workpiece pose is maximized along these two axes. Moreover, the quaternions are initialized considering only two axes to limit the number of seeds and maintain computational efficiency.

Let $\pm A$ be the range of Euler rotation angle in degrees that the initialized quaternions will vary around each selected axis, and n_steps the number of evenly

spaced steps taken in the interval $[-A, +A]$. The quaternions will vary pairing each value of the y- and z-axis Euler rotations as in a 2D-coordinate grid. A visualization representing this arrangement is depicted in Figure 4.3.

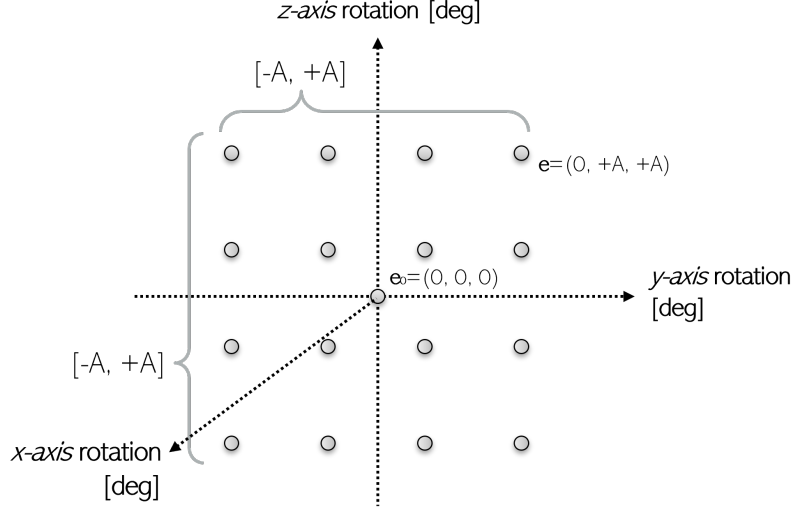


Figure 4.3: Initialization of seeds in the rotation solution space using Euler angle variations around the y- and z-axis.

The number of seeds, s , is then defined, as follows.

$$s = (n_steps + 1)^2 + 1\{n_steps \text{ is odd}\} \quad (4.5)$$

The optimization is always initialized with a seed without any translation or rotation, i.e., $\mathbf{t}_0 = [0, 0, 0]$ and $\mathbf{q}_0 = [1, 0, 0, 0]$. Also, all the x-axis rotations are initialized as 0. Notice that increasing the angle range opens up the possibility to search for the global optima in a larger solution space, although the initialized seeds will be farther away from each other. The definition of A and n_steps comes as a trade-off between computational cost (densely spaced seeds) and the coverage of the solution space.

4.2.4 Loss function

One of the novelties of our method is the choice of the registration loss function to be minimized as presented in Eq. 4.6. We propose a robust registration loss

function as a weighted sum of three distinct subcomponents. Two of them are based on the Chamfer Distance, and the third one is a differentiable metric based on the F1-Score adopted as an overlap metric between two 3D surface point clouds.

$$\mathcal{L}_{hybrid} = \alpha \mathcal{L}_{CD} + \beta \mathcal{L}_{pCD} + \gamma \mathcal{L}_{F1-Overlap} \quad (4.6)$$

These subfunctions are explained below.

Chamfer Distance (CD): The CD takes as input the 3D point clouds and serves the purpose of bringing the point clouds together in the 3D space. See Eq. 4.7.

$$\mathcal{L}_{CD}(\mathcal{S}, \mathcal{T}) = \frac{1}{|\mathcal{S}|} \sum_{p \in \mathcal{S}} \min_{q \in \mathcal{T}} \|p - q\|^2 + \frac{1}{|\mathcal{T}|} \sum_{q \in \mathcal{T}} \min_{p \in \mathcal{S}} \|p - q\|^2 \quad (4.7)$$

where p and q are the points in the source and target point clouds, and $|\mathcal{S}|$ and $|\mathcal{T}|$ are the number of points in each of them, respectively.

The CD is a dissimilarity metric between point clouds. In essence, it quantifies how different two point clouds are by summing the distances between each point in one set and its nearest neighbor in the other set, and it aims to enhance the tolerance for outliers in registration.

Although nearest point queries in the inner loops of registration have been deemed computationally expensive, CD offers the advantage of being differentiable and efficient to compute with vectorized functions (parallelizable on CUDA) for a moderate number of points.

Projected Chamfer Distance (pCD): The projected CD is a variation of the original metric but in a 2D space. In this instance, the input 3D point clouds are projected onto the primary 2D planes xy , xz , and yz . The standard CD is computed separately for each of the three resulting 2D point clouds and then summed up. See Eq. 4.8. This projected point cloud still maintains the contour feature for global registration, and it approximates techniques used in 2D

differential rendering.

$$\mathcal{L}_{pCD}(\mathcal{S}, \mathcal{T}) = \mathcal{L}_{CD_{xy}}(\mathcal{S}, \mathcal{T}) + \mathcal{L}_{CD_{xz}}(\mathcal{S}, \mathcal{T}) + \mathcal{L}_{CD_{yz}}(\mathcal{S}, \mathcal{T}) \quad (4.8)$$

Each CD projected onto the planes is defined as,

$$\mathcal{L}_{CD_{uv}}(\mathcal{S}, \mathcal{T}) = \frac{1}{|\mathcal{S}|} \sum_{p \in \mathcal{S}} \min_{q \in \mathcal{T}} \|p_{uv} - q_{uv}\|^2 + \frac{1}{|\mathcal{T}|} \sum_{q \in \mathcal{T}} \min_{p \in \mathcal{S}} \|p_{uv} - q_{uv}\|^2 \quad (4.9)$$

where uv denotes a projected plane, and p_{uv} and q_{uv} are the projected point vectors.

F1-Overlap Score: In this work, we propose *F1-Overlap Score*, a novel differentiable registration metric that aims to quantify the alignment overlap between two point clouds based on the F1-Score. See Eq. 4.10. This metric in itself is a standard in machine learning classification problems and combines two competing metrics, precision and recall scores. Evidently, the better aligned two points clouds are, the greater our overlap metric will be. Although the F1-Score has been used as a shape comparison metric, previous implementations are not differentiable.

$$\text{F1-Score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (4.10)$$

where precision is the fraction of the target points within a radius (r) of the source point cloud, i.e., the overlap percentage of the target on the source. The recall is the fraction of the source points within r of the target point cloud, in other words, the overlap percentage of the source on target.

However, our F1-Overlap Score is not based on point count, but rather on point distances to enable differentiation with respect to point location using ball queries. We use ball query to find all the points in a point set that are within a radius, r , to query points in another point set with a maximum number of K neighbors.

Therefore, using the same ideas in Eq. 4.10, in our F1-Overlap Score the overlap is defined as the sum of squared distances between the query point of one

point set intersecting the points of the other within the ball queried.

$$\mathcal{L}_{F1-Overlap} = -\frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall} + \epsilon}, \quad (4.11)$$

where $\epsilon = 1 \times 10^{-8}$ is added to prevent numerical instability when dividing by zero. Otherwise, since the F1-Score function may yield indeterminate numeric values, they are replaced with 0, if any.

In our proposed F1-Overlap Score loss, precision is defined as in Eq. 4.12.

$$\text{precision} = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \left(\frac{1}{|K|} \sum_{j=1}^K \text{sq_dist_ball_query}(\mathcal{S}, \mathcal{T}, r^2)_{ij} \right), \quad (4.12)$$

where $\text{dist_ball_query}(\mathcal{S}, \mathcal{T})$ is a function that returns a 2D matrix of size giving the squared distances of each query point in $\mathcal{S}$ to the K neighbors. Similarly, the recall is computed, as follows:

$$\text{recall} = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \left(\frac{1}{|K|} \sum_{j=1}^K \text{sq_dist_ball_query}(\mathcal{T}, \mathcal{S}, r^2)_{ij} \right). \quad (4.13)$$

Figure 4.4 illustrates how the proposed F1-Overlap Score metrics operate on pair-wise overlapping point clouds.

While the traditionally employed point cloud F1-Score operates on the overlap as a discretized metric, our proposed F1-Overlap Score provides a continuous metric given that it operates using neighbor point distances instead of solely a ratio of a number of points. By using the minimization of the distances, we encourage the points to be drawn closer to each other (if already within the overlapping region, i.e., the radius boundaries of their neighboring points), as well as to increase the number of points (increased summation of distances) within such overlapping regions. Although F1-Score functions can be found in the literature [63], our proposed implementation stands as fully differentiable.

F1-Overlap Score is robust to outliers and encourages coarse-to-fine alignment

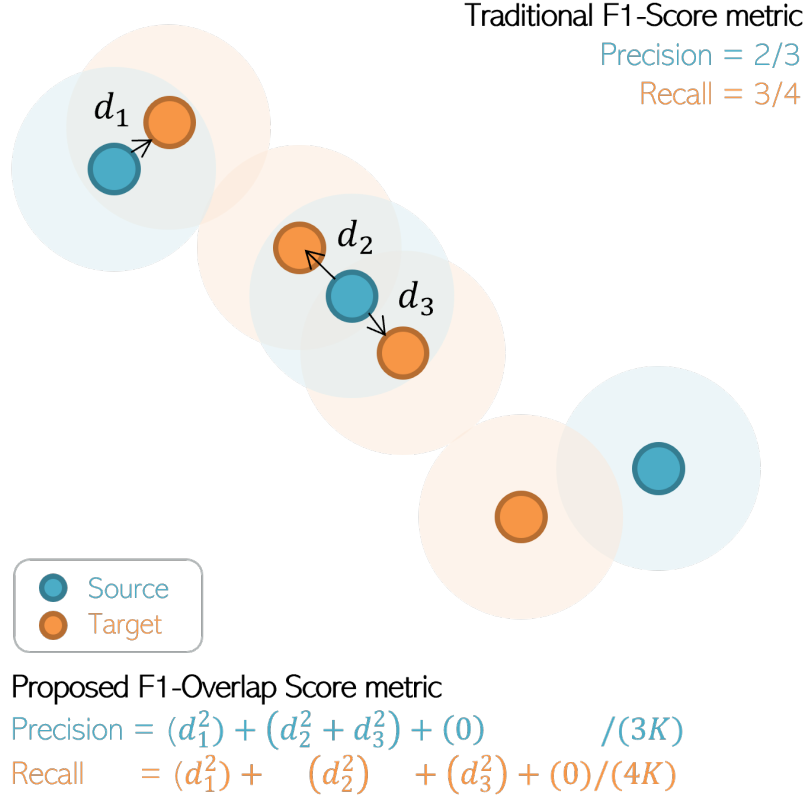


Figure 4.4: Operation of the proposed F1-Overlap Score on point clouds.

between two surface point clouds to maximize the intersecting overlap (as a surface interpenetration measure) and not trespass each other. These are the two key ideas that led to the formulation of this differentiable function. Furthermore, two partially overlapping and fully aligned point clouds present a maximal bidirectional overlap with each other. The strength of our F1-Overlap Score is based on the assumption that cross-source data (Zivid scan data and points sampled from the 3D model) does not have hard one-to-one point correspondences and the points do not align perfectly at the same location in the 3D space. F1-Overlap Score operates within the local intersection region between two point clouds, determined by using a ball query. Hence, it helps reduce sensitivity to differences in the overlap between the two point clouds. This means that even if one point cloud is significantly larger than the other, the metric remains constant.

Given that the loss function is evaluated across all the seeds as a batch, for the

final calculation of the loss function (Eq. 4.6) we reduced it to a single value by summing up each seed’s loss before backpropagation.

4.2.5 Stop criteria

The optimization loop is stopped once a stopping criterion is met. For example, the search can be stopped once the maximum number of iterations has been reached, or an early stop can be triggered when the gradient norm of a seed is below a threshold ϵ .

4.2.6 Selection of the best alignment candidate

Upon termination of the optimization loop, the transformation associated with the seed that has the maximum F1-Overlap score is chosen for the best alignment candidate, and this transformation is the final output of the registration algorithm. Our F1-Overlap Score is deemed a surface interpenetration measure and provides a robust relative indication of better point cloud matching compared to other inaccurate alignments. However, relying solely on the F1-Overlap Score for the binary alignment classification of two point clouds is not advised as it still requires an empirically estimated threshold.

4.3 Datasets

Three datasets have been used to test our proposed optimization-based registration method. Each dataset is aimed at evaluating specific criteria. The sample composite part introduced in the previous chapter is used as a benchmark for real-world data. Secondly, ModelNet40 [40] was chosen as a standard to evaluate our method and provide comparable results to existing methods in the literature. 2000 models were randomly selected from this dataset. Finally, using our synthetic point cloud generation engine we created 2,000 samples for testing on industrial-looking diversity of objects under corruptions. These datasets are described in Table 4.1.

Table 4.1: Datasets for testing our optimization-based method.

Dataset	Real data	Num of samples	Corruptions/ partiality	Ground truth
Composite part dataset	✓	40	✓	Manual+GICP
ModelNet40	No	2,000	Synthetic	Synthetic
Synthetically generated	No	2,000	Synthetic	Synthetic

4.4 Results and discussion

We performed a series of comprehensive tests to evaluate the performance of our registration method. In all of our experiments, we adhere to a standard set of optimization hyperparameters, θ , unless otherwise stated with the purpose of testing specific performance under distinct conditions and parameter settings. The parameters used are detailed below.

- Outlier removal filter settings, `nb_neighbors` = 30 and `std_ratio` = 1.65.
- For the ball query search, we use a search squared radius of $r^2 = 0.15$, and $K = 5$.
- The maximum number of iterations is `max_iter` = 500, and an early stop threshold of $\epsilon = 5 \times 10^{-5}$.
- Initial learning rate of 0.9989 and decay of 0.9989.
- The overall loss function weights are $\alpha = 1$, $\beta = 6$, and $\gamma = 2$.
- All the point clouds are constrained to have 2048 points or less if outliers are removed.
- For the initialization strategy, we populate the seeds using $A = \pm 20deg$ and $n_{steps} = 4$ as our parameters.
- Based on the previous statement, the standard number of seeds used is $s = 17$, unless otherwise specified.

For evaluation, we utilize the registration metrics as presented in Chapter 3. Also, given the ability of this registration method to find the global alignment, we are not providing an initial guess from our keypoint matching module. Thus, we report our results solely based on our optimization algorithm.

In all our experiments, we assumed a maximum initial deviation of ± 20 deg around all the rotation axes in an Euler angle representation, and ± 0.5 translation deviation in the normalized scale. The tests were performed on an NVIDIA GeForce GTX 1650 GPU. A Google Colab Tesla V100 GPU was additionally employed for runtime performance comparison based on the choice of hardware.

4.4.1 Testing on composite part dataset

We report our experimental results on the sample composite part dataset. This dataset is targeted to test performance on real-world data, and it includes samples of the workpiece with virtually full visibility ($\geq 90\%$ overlap) and samples with lower partial visibility of the part (approximately 30%) for challenging scenarios.

Since we only operate on point cloud data, a random misalignment on the source point cloud can be applied to test our registration method. Before performing registration, all the part scans have been preprocessed with our workpiece segmentation module to remove background and clamps.

First, we investigate the performance of UPGReg on the initial misalignment in rotation and translation components separately. In each test, the rotation and translation misalignment are randomly chosen from the specified maximum range. Tables 4.2 and 4.3 summarize the results. To obtain these results, 780 tests were performed, and the metrics were averaged and reported as a single value.

Table 4.2: Test results on random rotation misalignment.

Rot	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
± 10	0.432	0.313	0.132	0.101	0.671	0.196	0.994
± 20	0.711	0.328	0.130	0.100	0.698	0.194	0.995
± 20	0.660	0.361	0.134	0.103	0.747	0.198	0.987
± 45	9.061	1.522	0.125	0.096	2.750	0.188	0.910
± 90	62.419	32.193	0.145	0.108	48.955	0.219	0.449

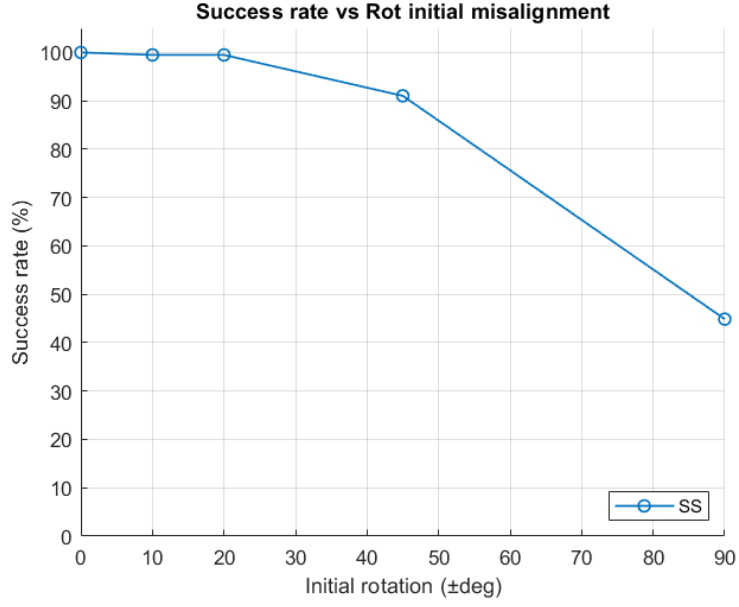
Table 4.3: Test results on random translation misalignments.

trans.	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
± 0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
± 0.1	0.428	0.310	0.131	0.101	0.648	0.195	0.994
± 0.2	0.494	0.329	0.130	0.100	0.679	0.193	0.987
± 0.3	0.412	0.308	0.131	0.101	0.641	0.196	1.000
± 0.4	0.416	0.315	0.131	0.101	0.658	0.196	1.000
± 0.5	0.711	0.328	0.130	0.100	0.698	0.194	0.995
± 0.7	0.490	0.325	0.131	0.100	0.677	0.194	0.994
± 1.0	1.144	0.380	0.132	0.102	0.783	0.196	0.994

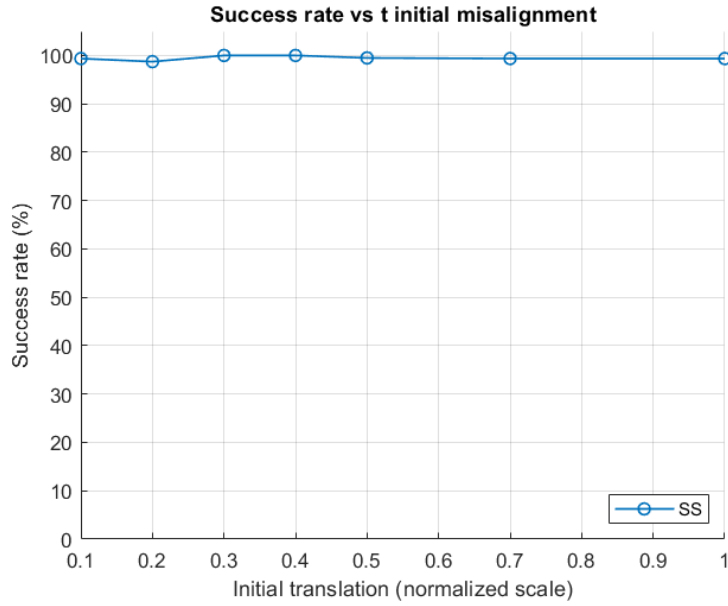
Figure 4.5 illustrates our results based on the success rate as our main metric in this study. It is evident that our method can handle rotation misalignments of up to 20 deg but the performance degrades for larger rotation misalignments. The registration success rate remains virtually invariant under translation variations due to the normalization step. Hence, UPGReg can yield a high success rate on real-world point cloud data.

Secondly, we perform a comparison study of our method against selected baselines utilized in the literature. In this comparison, we considered local and global optimization-based methods including ICP and its variants, RANSAC+ICP, Go-ICP, ICP Generalized, and Fast Global Registration. These traditional methods were utilized as implemented and packed in [56]. On the learning-based side, we selected the baselines [16, 20, 24, 28] using the official pre-trained model and hyperparameter selection by their authors. Table 4.4 shows the comparison results. We observed that our method outperforms traditional and learning-based methods on real-world data yielding a success rate of 99.8%. Given that not all these methods support partial overlapping scenarios, only part scans with an overlap greater than 60% were utilized. Moreover, the input point clouds were also normalized to keep consistent translation metrics across all the selected methods. Figure 4.6 shows a bar plot demonstrating the comparison results based on the success rate.

3D reconstruction of sample composite part: Same-source PCR is instrumental for 3D scene reconstruction, e.g., to assemble point clouds of the same object from different viewpoints to obtain a more comprehensive representation of



(a) Rotation misalignment



(b) Translation misalignment

Figure 4.5: Success rate as a function of the initial misalignment.

the workpiece and increase overlap surfaces with the CAD model. Due to its significance, we also report registration metrics using two scans of the sample composite part dataset as the target and source point clouds.

Table 4.4: Comparison of our proposed optimization-based method against existing baselines.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
ICP P2P	15.496	6.283	0.363	0.185	12.188	0.355	0.099
ICP P2L	31.971	14.159	1.924	0.308	28.805	0.662	0.233
RanICP	67.638	33.994	0.141	0.080	75.140	0.171	0.335
GO-ICP	23.450	6.564	0.059	0.039	12.962	0.078	0.497
GICP	3.538	1.592	0.097	0.034	3.310	0.070	0.622
FGR	13.429	3.994	0.059	0.029	7.798	0.058	0.656
iPCRNet	31.928	24.776	0.662	0.566	45.999	1.101	0.000
PRNet	5.728	3.581	0.115	0.082	8.296	0.165	0.109
PRMNet	23.311	8.730	0.331	0.160	18.609	0.322	0.067
HOUV	88.070	64.488	0.538	0.450	118.977	0.879	0.043
Ours	0.711	0.328	0.130	0.100	0.698	0.194	0.991

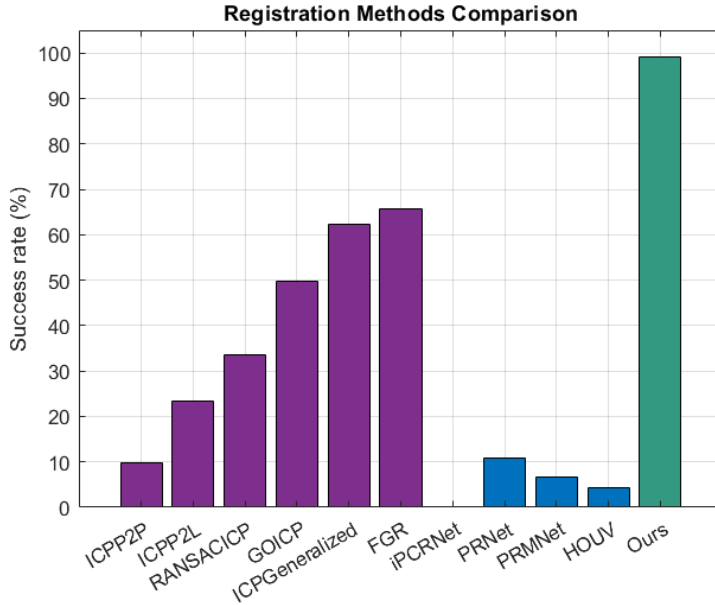


Figure 4.6: Success rate comparison of our method (green) against existing traditional (purple) and learning-based (blue) baselines on real-world data.

Single vs multi-seed optimization Tests were performed to observe the influence of the number of parallel seeds on the registration performance with the sample composite part dataset. Table 4.5 shows the obtained metrics from this study.

An average performance improvement of 21.53% is noticed when using multi-seed (17 seeds) at the expense of a minimal computation time increase as shown in Figure 4.7. On the same plot, we show the success rate as a function of the

Table 4.5: Test results on increasing the number of parallel optimization seeds.

Seeds	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	E(R)	E(t)	SS	Time (s)
1	7.207	2.002	0.122	0.093	4.170	0.184	0.782	3.53
5	3.940	0.579	0.129	0.100	1.228	0.193	0.967	3.82
9	0.919	0.376	0.203	0.172	0.739	0.341	0.979	3.56
17	0.654	0.325	0.131	0.101	0.679	0.195	0.997	3.64
25	2.839	0.375	0.209	0.175	0.820	0.351	0.997	3.83
37	0.419	0.307	0.131	0.101	0.645	0.195	0.997	5.23
49	0.446	0.295	0.208	0.175	0.583	0.349	0.995	6.20
65	0.411	0.308	0.132	0.101	0.645	0.196	1.000	7.17

number of seeds. The parameter A remained constant for these tests, thus the seed density grew in the same solution subspace. Interestingly, it is noted that the success rate remains nearly consistent above 17 seeds. This experiment shows the significance of combining optimization techniques with parallel seeds to solve the registration problem.

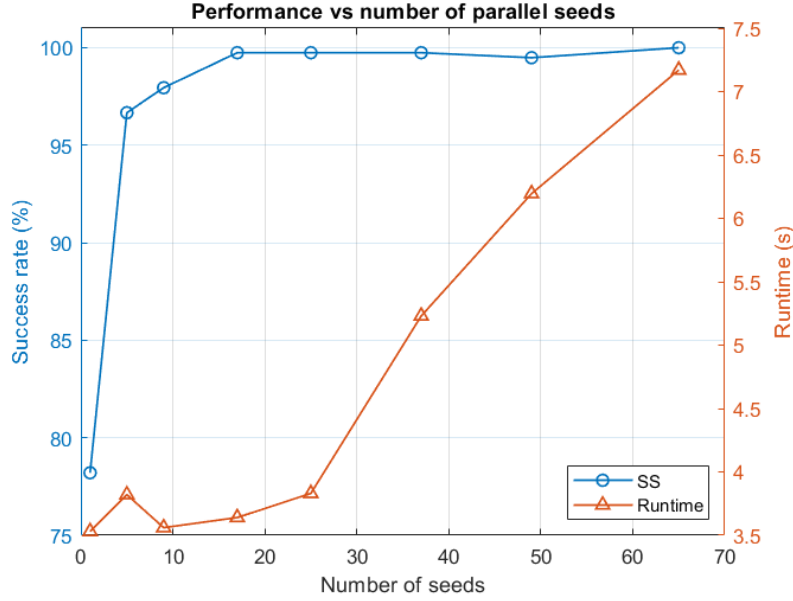


Figure 4.7: Success rate and runtime (per pairwise alignment) as a function of the number of seeds.

The average runtime for 17 seeds is 3.64 seconds on a Tesla V100 per pairwise registration. Running the same optimization algorithm on the GTX 1650 would increase the average runtime by a factor of 2. Still, these processing times remain competitive for quasi-real time processing on the shop floor.

Robustness on real-world data To further investigate the effects of partially overlapping point clouds on success rates, in the following test we utilized the scans of the sample composite part with low overlap. We also progressively applied cut-out adversarial corruptions to synthetically reduce the overlap region with respect to the nominal CAD model on the part scan with the original high overlap. The results of the success rate are shown in Figure 4.8.

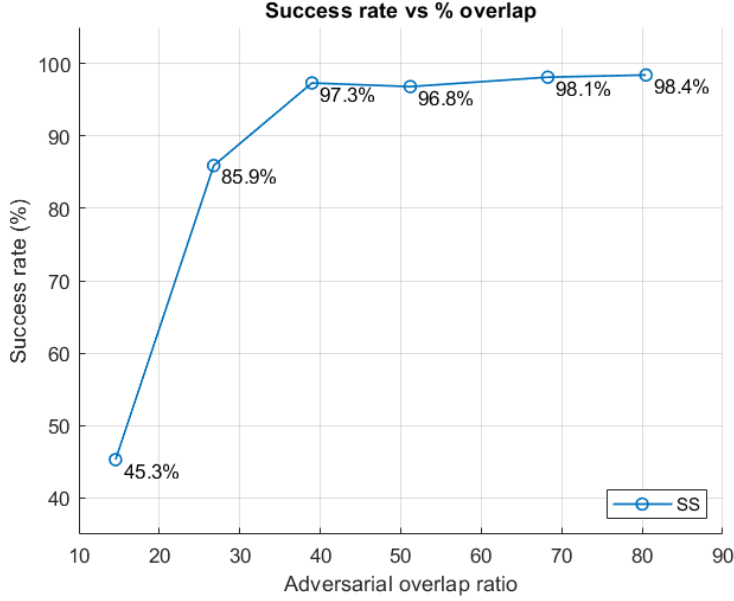
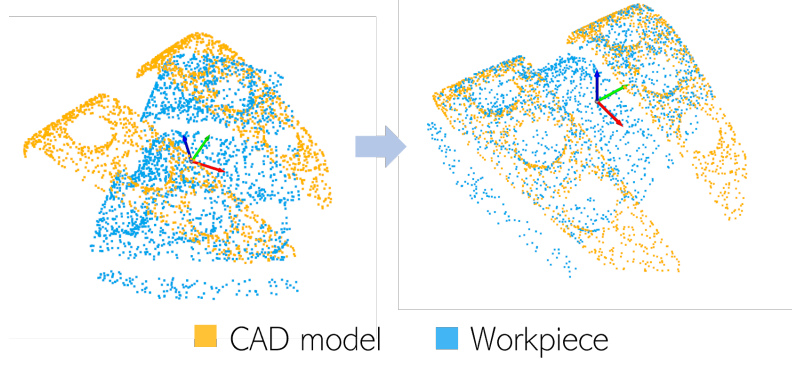


Figure 4.8: Success rate as a function of percentage overlap on the sample composite part dataset.

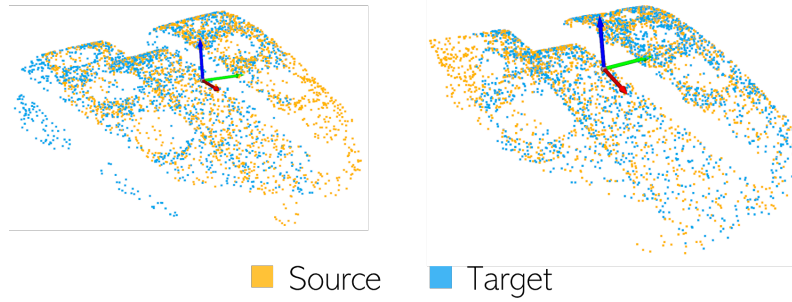
These results demonstrate that even though our method degrades in the case of partially overlapping point clouds, especially in the low percentage regime, it shows a consistent robust performance for overlaps greater than 40%. This performance is satisfactory for most practical scenarios.

Qualitative results Figure 4.9 depicts some successful registration cases from the composite part dataset. These cases showcase the tight alignments obtained in cross-source and same-source registration.

The presented test results show that our loss function, and more precisely, F1-Overlap score has the ability to finely align two point clouds and eliminate the need for a refinement stage typically carried out with ICP as the last step.



(a) CAD-based registration example



(b) 3D reconstruction examples

Figure 4.9: Qualitative results on the composite part dataset.

Optimization landscape analysis of the loss function In this section, we present an analysis and visualization of the convergence of the proposed hybrid loss function (Eq. 4.6) and the optimization landscape. The following plots were obtained by running the optimization algorithm with a single seed using a sample point cloud from the composite part dataset with the standard parameters.

An example is shown in Figure 4.10. We observe that the loss function converges to a minimum value over the number of iterations. In this example, a convergence was attained within the first 100 iterations.

Furthermore, the norm of the gradients (see Figure 4.11) shows that the optimization variables decrease with the number of iterations, showing agreement with the behavior of the loss function.

To observe the optimization landscape of the loss function, 3D surface plots

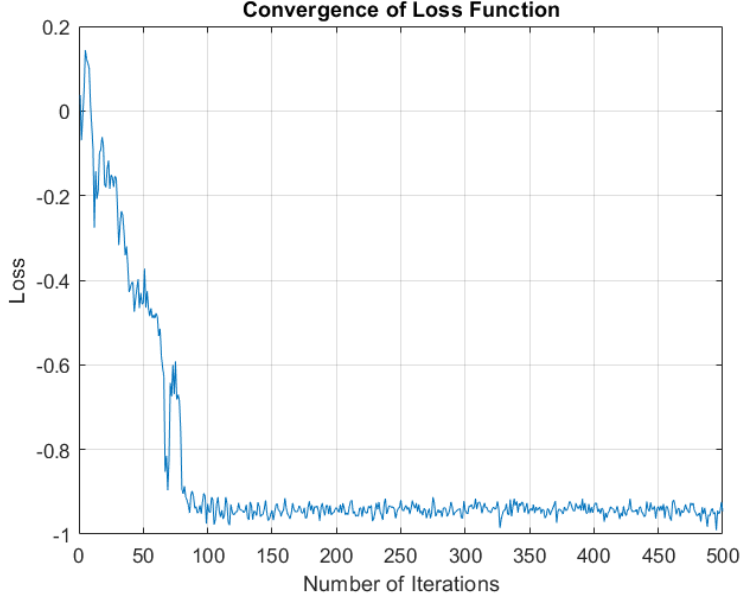


Figure 4.10: Convergence of loss function.

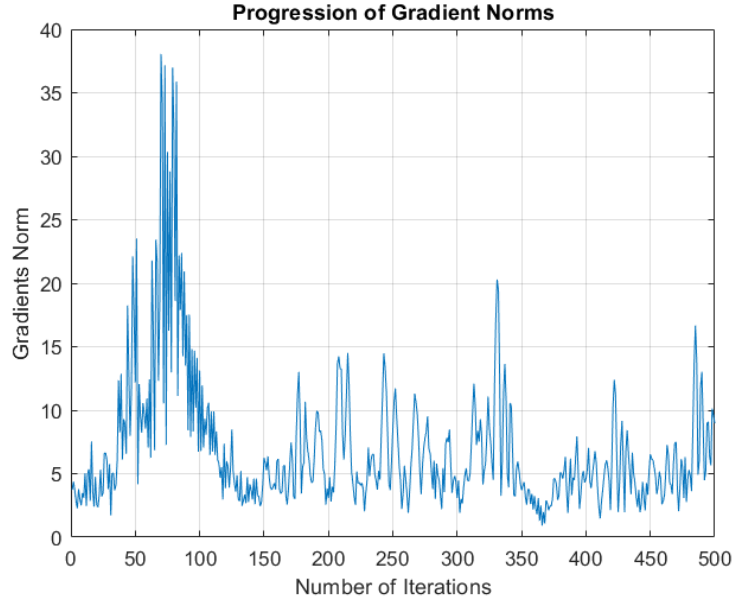


Figure 4.11: Gradients as a function of number of iterations.

were obtained by solely translating the source point cloud position in the x-, y- and z-axis. These plots are shown in Figures 4.12, 4.13, and 4.14. It is noted that the shape of the optimization solution surface possesses convex properties ideal for fast convergence to a global solution. At the center of the surface, we observe the effect of the F1-Overlap Score loss converging to the global minimum, i.e., the

correct transformation with tight alignment.

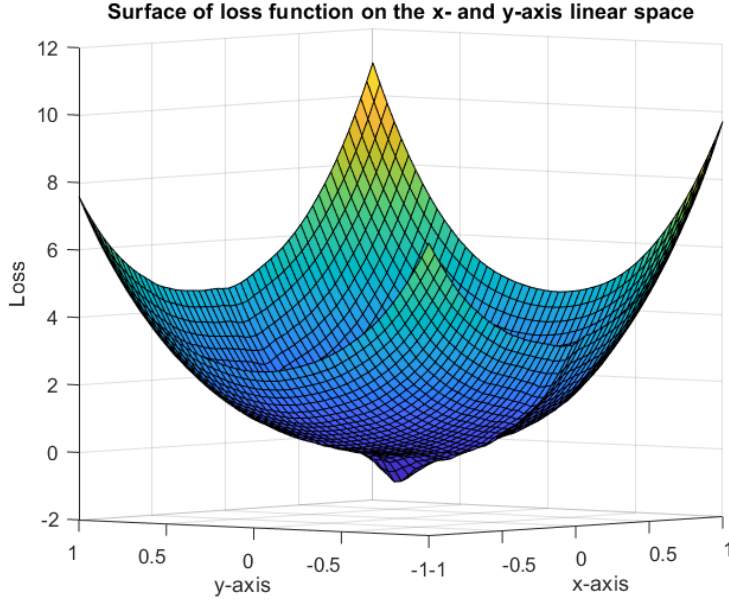


Figure 4.12: Optimization landscape translating the source in the x- and y-axis.

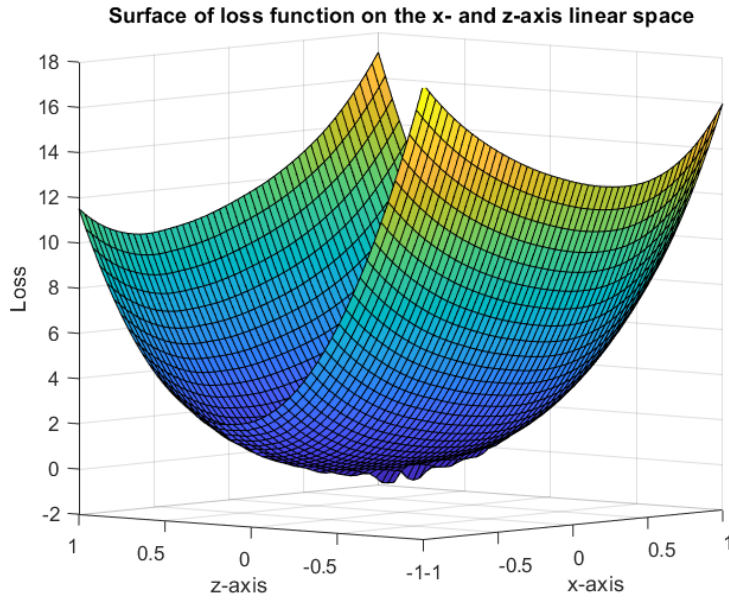


Figure 4.13: Optimization landscape translating the source in the x- and z-axis.

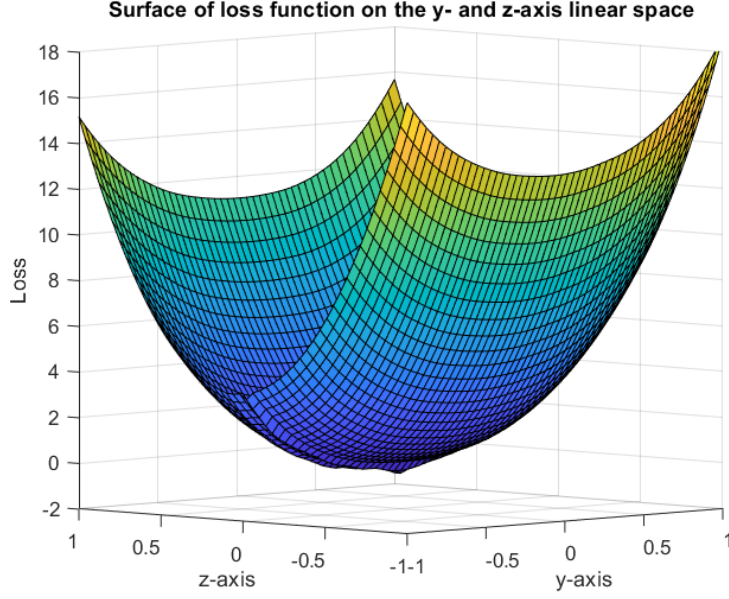


Figure 4.14: Optimization landscape translating the source in the y- and z-axis.

4.4.2 Testing on synthetic dataset

We carried out a series of comprehensive tests on our synthetically generated industrial dataset to evaluate the generalization capability of our method and assess its robustness to adversarial corruptions on the diversity of objects. For this evaluation, we randomly took 2,000 models from this dataset.

In our first analysis, we investigate the effect of the severity factor, introduced in the previous chapter, on the registration metrics. The results are detailed in Table 4.6.

Table 4.6: Test results of the severity factor on the industrial synthetic dataset.

Severity	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
1	0.762	0.125	0.068	0.057	0.272	0.113	0.983
2	0.652	0.162	0.140	0.119	0.324	0.237	0.993
3	0.721	0.337	0.217	0.182	0.687	0.365	0.967
4	6.366	1.106	0.366	0.303	2.215	0.615	0.847
5	5.597	2.215	0.409	0.335	4.563	0.685	0.533

Although our results show that our method is sensitive to corruption, it shows robustness for common point cloud corruptions. A registration success rate of 96.7% is attained for a severity factor of 3, which represents the worst level of

corruption observed in real-world applications. Figure 4.15 illustrates the performance variation as more aggressive corruption levels are applied.

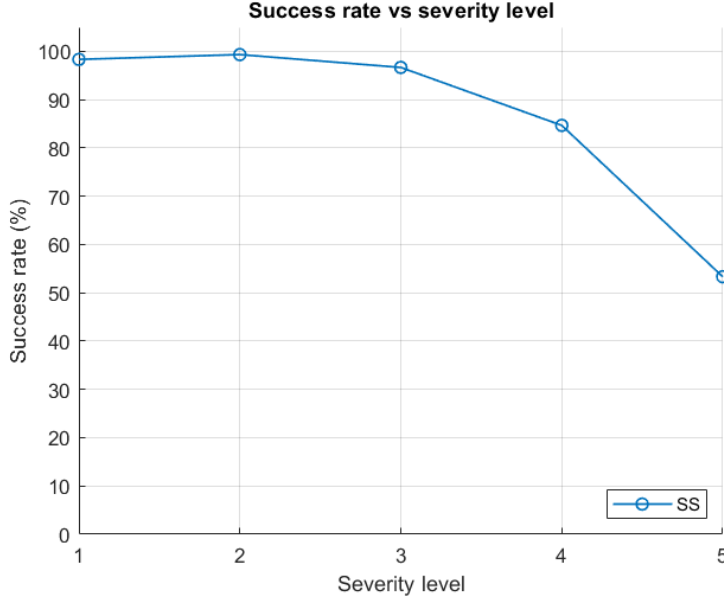


Figure 4.15: Success rate as a function of severity factor on the synthetic dataset.

Furthermore, we also evaluated this dataset subject to random variations in the initial rotation misalignment. A random ± 0.5 translation deviation is also applied. A constant severity factor of 3 was used for this analysis. As the maximum angle range increases, we also augmented the kernel spacing (A). The results are shown in Table 4.7.

Table 4.7: Test results on the rotational misalignment on the industrial synthetic dataset.

A	Rot	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	R(R)	E(t)	SS
20.000	± 20	0.272	0.050	0.220	0.184	0.103	0.370	0.994
40.000	± 45	0.645	0.096	0.216	0.185	0.181	0.367	0.987
80.000	± 90	33.797	8.931	0.195	0.170	12.443	0.329	0.850

As a qualitative test result, Figure 4.16 shows two objects from this dataset accurately aligned. Again, tight alignment is ensured without extraneous refinement steps.

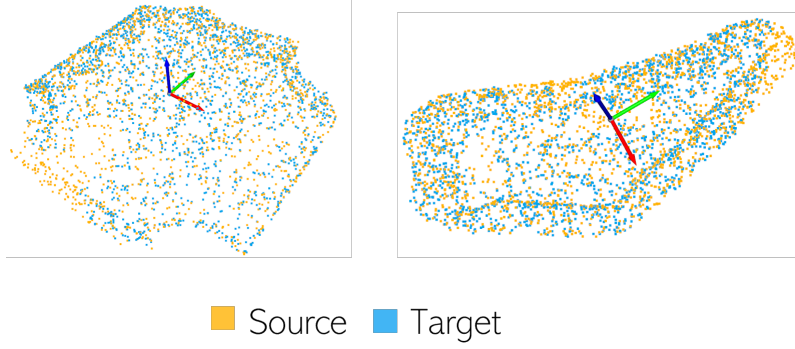


Figure 4.16: Qualitative results on our custom synthetic dataset.

4.4.3 Testing on ModelNet40 dataset

For standard benchmarking purposes and consistent evaluation with related works, we tested our method on the ModelNet40 dataset. Two main sets of tests were performed, with clean and noisy models. For the latter, we applied a Gaussian noise with a severity factor of 1, i.e., $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$, as employed in previous studies [8, 21, 24, 28]. Following [26], all categories with axisymmetric objects were dismissed to avoid false negatives due to arbitrary ground truth angles on the symmetrical axis. Moreover, points from the CAD model are randomly sampled twice for the source and target separately. This technique also helps to improve the convergence of our F1-Overlap Score function. Perfect point-to-point correspondences may cause a low F1-Overlap Score for the best point cloud alignment, causing our method to pick the wrong alignment. However, this exceptional condition does not occur in practical scenarios with real-world data. The metrics from our experiments are reported in Table 4.8.

Table 4.8: Test results on clean and noisy samples from the ModelNet40 dataset.

	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	Error(R)	Error(t)	SS
Clean	9.836	1.033	0.229	0.198	1.949	0.381	0.961
Noisy	7.793	1.329	0.230	0.199	2.841	0.385	0.940

Our method shows competitive performance on the clean and noisy dataset with a registration success rate as high as 94.0% to 96.1%. Figure 4.17 depicts four models accurately aligned using UPGReg.

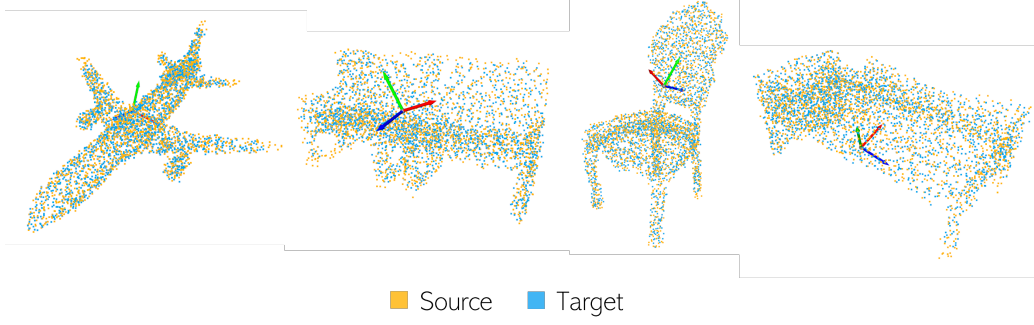


Figure 4.17: Qualitative results on the ModelNet40 dataset.

Analysis of failed cases: To further gain an understanding of the causes of failed registration outcomes in our optimization pipeline, we conducted an analysis of 20 representative failed cases from the composite part dataset and ModelNet40. From this examination, we observed that the following cases are more prone to registration failure.

- **Partial similarity:** Repetitive surface patterns and structures across different sections of the object.
- **Symmetric objects by two or three planes.**
- **Lack of prominent features.** For instance, objects with predominantly planar surfaces and small distinctive structures.
- **Non-optimized registration hyperparameters:** Although our experiments show robustness on the selection of the optimization parameters, if the point clouds present a fairly small initial misalignment, a lower learning rate should be utilized. We selected the learning rate based on an exploration approach; hence, the optimization seeks out a solution more aggressively further away from the initialization in the early iterations.

4.4.4 Loss ablation study

In this section, we aim to assess our method based on the choice of the loss function for optimization. Variants of our proposed hybrid loss functions (see Eq. 4.6) were

tested for comparison. In each variant, we dismiss one of the subfunction losses, as described in Table 4.9. The obtained results are shown below.

Table 4.9: Test results on the loss ablation study.

Loss variant	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	E(R)	E(t)	SS
No $\mathcal{L}_{F1-Ov.}$	4.136	1.990	0.211	0.178	4.258	0.345	0.718
No $\mathcal{L}_{pCD}$	17.306	2.550	0.218	0.187	5.835	0.359	0.897
No $\mathcal{L}_{CDS}$	0.859	0.397	0.211	0.181	0.800	0.347	0.974
All, $\mathcal{L}_{hybrid}$	0.416	0.307	0.211	0.182	0.657	0.347	1.000

These results demonstrate that the absence of our novel F1-Overlap Score causes the most detrimental performance on the registration success rate. Furthermore, $\mathcal{L}_{hybrid}$ yields the best-performing results with a perfect score of 100% SS in these tests.

Other unsupervised loss functions were tested including the Rigid Motion Loss [24] and the Earth Mover’s Distance [64] as the objective function for gradient descent. Similarly, different optimizers such as SGD and LBFGS were put into test, however, neither of the above methods yielded promising converging results compared to our selected Adam optimizer.

Choice of loss function for best seed transformation: Throughout our experiments, our F1-Overlap Score demonstrated to be a robust metric for selecting the best alignment seed as compared to the Chamfer Distance, especially in the low overlap regime. Figure 4.18 shows the transformations of two different seeds upon completion of the optimization. The alignment on the left shows two registered point clouds based on the best hybrid F1-Overlap Score ($CD = 0.186, \mathcal{L}_{hybrid} = 0.279$), whereas the alignment on the right depicts the alignment using the minimum CD as the selection criterion ($CD = 0.075, \mathcal{L}_{hybrid} = 0.153$).

While the seed yielding the minimum CD is completely misaligned, the best F1-Overlap Score is tightly aligned and in the correct global location. This evidently shows that CD is not a reliable metric for the selection of the best seed and that our proposed loss shows resilience even in the low overlap scenarios.

The selection of the point cloud sampling strategy was also a design factor.

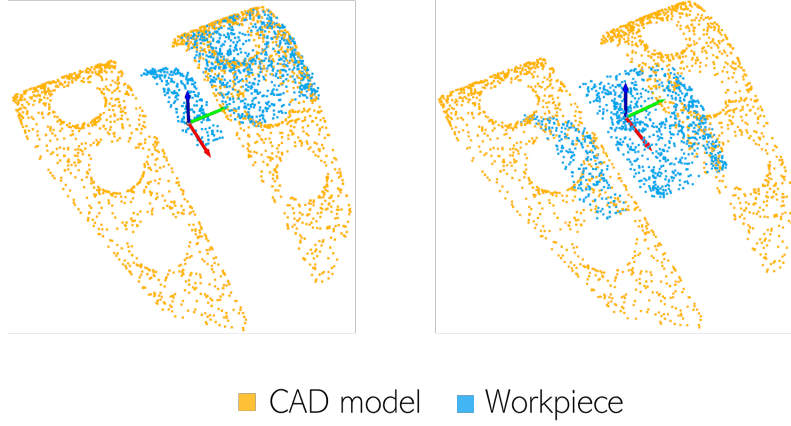


Figure 4.18: Qualitative results of best alignment based on two registration metrics.

In our ablation experiments, it was observed that using the farthest point sampling decreased the registration performance of our optimization strategy. Hence, sampling the point clouds with a uniform distribution is encouraged.

4.5 Summary

In this chapter, we proposed a novel optimization-based registration strategy to align point clouds under partiality and point cloud corruptions. The method was developed leveraging deep learning machinery using a gradient descent as our main optimization algorithm with multiple initializations in the solution space that are solved efficiently by harnessing the power of parallel GPU processing. Three datasets were chosen for benchmarking of our registration results. The composite part dataset for real-world data, ModelNet40 for comparison with existing models, and our synthetically generated dataset of industrial objects. Our results show that our method can attain a 99% registration success rate on real-world data and shows generalizability to different workpieces when tested on our synthetically generated dataset with diverse objects with a success rate of up to 98.3%. While this method was initially developed to solve the tight alignment issues posed by learning-based registration methods, our results show that it has the ability to solve the alignment from a global manner. Additionally, our results demonstrate

that optimization-based approaches still hold the potential to solve challenging point cloud scenarios even under partiality and common point cloud corruptions. Solving registration with a combination of learning- and optimization-based methods [15, 16] is a promising venue for generalized workpiece localization in robotic manufacturing applications.

Chapter 5

Conclusions

5.1 Completed work and key contributions

In conclusion, this research addresses the current challenges of point cloud registration under partial overlaps and corruption patterns. This is a crucial step for automated part localization in fully autonomous robot cells used in high-mix manufacturing.

In Chapter 3, we propose a learning-based pipeline comprising 3 modules targeting workpiece agnostics localization. The first module leverages a pre-trained segmentation model for accurate selection of the part neglecting noise in the surroundings such as fixturing devices. The second module integrates an image-guided registration strategy in which a second pre-trained computer vision model is used to extract accurate pixel correspondences that are then projected into the 3D space to estimate the alignment transformation. In our third and last module, we have introduced the design of an iterative neural network with the novelty of adding a joint point cloud global feature embedding followed by a multi-head cross attention later. To overcome the limitations observed in the third learning-based module to reach tight alignment, we proposed a novel optimization-based registration method based on gradient descent and parallel GPU processing. This method incorporates a novel loss overlap loss function that is robust to noise and partial overlaps.

Our optimization-based solution achieves a 99.8% registration success rate on sample real data. The registration robustness is demonstrated across three distinct datasets under different levels of corruption and overlap. Furthermore, we

introduced a synthetic point cloud engine to generate industrial objects for robust benchmarking on the diversity of objects with corruption patterns to mimic real-world data.

We demonstrated that optimization-based approaches hold research potential in tackling point cloud alignment with partial overlaps. It was also shown that a hybrid combination of learning- and optimization-based methods is a promising avenue to achieve fully automated workpiece localization.

5.2 Future research direction

Several research areas can be considered to continue and extend the work described in the thesis.

First, the proposed learning architecture currently requires extensive computational resources and memory. A promising research direction would be to develop more efficient neural networks to address the partial overlap scenarios without the need to downsample point clouds. This approach would allow for the creation of more robust and accessible architectures with better generalization capabilities. With more efficient architecture designs, raw image data could be input into the same registration network to learn from the intrinsics of this data.

Second, further extensions on our optimization-based approach could increase the computational efficiency. For instance, implementing a seed pruning strategy if the seeds diverge from a collective global minimum could reduce registration runtime.

Third, addressing the workpiece localization problem using non-rigid registration is a promising area for future research. In real-world high-mix manufacturing scenarios, there could be a mismatch between the nominal CAD model and the actual workpiece. Rigid CAD-based matching assumes the workpieces are repetitive and accurate with respect to their nominal CAD model which is not always the case. Non-rigid registration would help address the differences in object scale and part deformations. Our proposed approach allows for adjusting the loss function

by incorporating a scale factor enabling us to solve the registration problem in the non-rigid regime. This extension would pave the way for CAD-based scale-invariant registration, providing more robust solutions for real-world applications on the shop floor.

Bibliography

- [1] OpenAI. Chatgpt: An ai language model, 2024. Accessed: 2024-07-03.
- [2] VTC Manitoba. Manitoba’s vehicle technology centre, 2024. <https://vtci.ca/about-us> [Accessed: March 13, 2024].
- [3] Zhiyuan Zhang, Jiadai Sun, Yuchao Dai, Dingfu Zhou, Xibin Song, and Mingyi He. A representation separation perspective to correspondence-free unsupervised 3-d point cloud registration. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.
- [4] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration, 2021.
- [5] Aleksandr Segal, Dirk Hähnel, and Sebastian Thrun. Generalized-icp. 06 2009.
- [6] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, November 2016.
- [7] Zhi Deng, Yuxin Yao, Bailin Deng, and Juyong Zhang. A robust loss for point cloud registration, 2021.
- [8] Yousef Alborzi. Automated vision-based part localization and path generation in robotic fiberglass trimming. Master’s thesis, University of Manitoba, Department of Mechanical Engineering, 2023.

- [9] Shan Xue, Zhen Zhang, Qiongying Lv, Xianyu Meng, and Xin Tu. Point cloud registration method for pipeline workpieces based on pca and improved icp algorithms. *IOP Conference Series: Materials Science and Engineering*, 612:032188, 10 2019.
- [10] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.
- [11] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. volume 9906, 10 2016.
- [12] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration, 2020.
- [13] Nicolas Mellado, Niloy Mitra, and Dror Aiger. Super 4pcs fast global point-cloud registration via smart indexing. *Computer Graphics Forum*, 33, 08 2014.
- [14] Kenta-Tanaka et al. probreg.
- [15] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration, 2020.
- [16] Yuanjie Yan, Junyi An, Jian Zhao, and Furao Shen. Hybrid optimization with unconstrained variables on partial point cloud registration. *Pattern Recognition*, 136:109267, 2023.
- [17] Liang Pan, Zhongang Cai, and Ziwei Liu. Robust partial-to-partial point cloud registration in a full range. *arXiv preprint arXiv:2111.15606*, 2021.
- [18] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.
- [19] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon

- Lucey. Pointnetlk: Robust and efficient point cloud registration using pointnet, 2019.
- [20] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. Pernet: Point cloud registration network using pointnet encoding, 2019.
- [21] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration, 2019.
- [22] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019.
- [23] Wentao Yuan, Ben Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration, 2020.
- [24] Yue Wang and Justin M. Solomon. Prnet: Self-supervised learning for partial-to-partial registration, 2019.
- [25] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration, 2020.
- [26] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration, 2021.
- [27] Hao Xu, Nianjin Ye, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Finet: Dual branches feature interaction for partial-to-partial point cloud registration, 2022.
- [28] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned

- p>features. In
- 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*
- . IEEE, June 2020.
- [29] Vinit Sarode, Animesh Dhagat, Rangaprasad Arun Srivatsan, Nicolas Zevallos, Simon Lucey, and Howie Choset. Masknet: A fully-convolutional network to estimate inlier points, 2020.
 - [30] Ruqin Zhou, Hanyun Wang, Xixing Li, Yulan Guo, Chenguang Dai, and Wanshou Jiang. Masknet++: Inlier/outlier identification for two point clouds. *Computers and Graphics*, 103:90–100, 2022.
 - [31] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap, 2021.
 - [32] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration, 2022.
 - [33] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration, 2021.
 - [34] Zi Jian Yew and Gim Hee Lee. *3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration*, page 630–646. Springer International Publishing, 2018.
 - [35] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019.
 - [36] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features, 2020.
 - [37] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration, 2021.

- [38] Ahmed Hatem, Yiming Qian, and Yang Wang. Point-tta: Test-time adaptation for point cloud registration using multitask meta-auxiliary learning, 2023.
- [39] Jiawei Ren, Lingdong Kong, Liang Pan, and Ziwei Liu. Pointcloud-c: Benchmarking and analyzing point cloud perception robustness under corruptions, 11 2022.
- [40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.
- [41] Xiaoqian Zhang, Junlin Li, Wei Zhang, Yansong Xu, and Feng Li. Two-stage point cloud registration framework based on graph neural network and attention. *Electronics*, 13(3), 2024.
- [42] Yong-Deuk Shin, Jae-Han Park, and Moon-Hong Baeg. 6dof pose estimation using 2d-3d sensor fusion. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 714–717, 2012.
- [43] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.
- [44] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.
- [45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick La-

- batut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.
- [46] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions, 2022.
- [47] Zheng Dang, Lizhou Wang, Junning Qiu, Minglei Lu, and Mathieu Salzmann. What stops learning-based 3d registration from working in the real world?, 2021.
- [48] Håkan Almqvist, Martin Magnusson, Tomasz Kucner, and Achim Lilienthal. Learning to detect misaligned point clouds. *Journal of Field Robotics*, 35, 12 2017.
- [49] Daniel Adolfsson, Martin Magnusson, Qianfang Liao, Achim J. Lilienthal, and Henrik Andreasson. Coral – are the point clouds correctly aligned?, 2021.
- [50] Ludvig Dillén. Feature-aware point transformer for point cloud alignment classification : Pose your pose to fact. Master’s thesis, Linköping University, Department of Electrical Engineering, 2023.
- [51] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Härtinger, and Carsten Steger. Introducing mvtec itodd — a dataset for 3d object recognition in industry. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2200–2208, 2017.
- [52] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning, 2019.
- [53] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models, 2021.

- [54] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.
- [55] RoboDK. Computer vision - simulated camera, 2024. <https://robodk.com/doc/en/Computer-Vision-Simulated-Camera.html> [Accessed: March 19, 2024].
- [56] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
- [57] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence, 2023.
- [58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [59] Trevor Laughlin. pyocct – python bindings for opencascade via pybind11, 2020. <https://github.com/trelau/pyOCCT>.
- [60] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. The MIT Press, 2019.
- [61] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.
- [62] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

- [63] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebedev. Kaolin: A pytorch library for accelerating 3d deep learning research. <https://github.com/NVIDIAGameWorks/kaolin>, 2022.
- [64] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image, 2016.