POINT CLOUD REGISTRATION USING QUANTILE ASSIGNMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Ecenur Oğuz
July 2023

POINT CLOUD REGISTRATION USING QUANTILE ASSIGNMENT

By Ecenur Oğuz

July 2023

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Mustafa Ç. Pınar(Advisor)

Oya Karaşan(Co-Advisor)

Uğur Güdükbay

Yusuf Sahillioğlu

Approved for the Graduate School of Engineering and Science:

_____

Orhan Arıkan
Director of the Graduate School

ABSTRACT

POINT CLOUD REGISTRATION USING QUANTILE ASSIGNMENT

Ecenur Oğuz
M.S. in Industrial Engineering
Advisors: Mustafa Ç. Pınar and Oya Karaşan
July 2023

Point cloud registration is a fundamental problem in computer vision with a wide range of applications. The problem mainly consists of three parts: feature estimation, correspondence matching and transformation estimation. We introduced the Quantile Assignment problem and proposed a solution algorithm to be used in a point cloud registration framework for establishing the correspondence set between the source and the target point clouds. We analyzed different common feature descriptors and transformation estimation methods to combine with our Quantile Assignment algorithm. The performance of these approaches together with our algorithm are tested with controlled experiments on a dataset we constructed using well-known 3D models. We detected the most suitable methods to combine with our approach and proposed a new end-to-end pairwise point cloud registration framework. Finally, we tested our framework on both indoor and outdoor benchmark datasets and compared our results with state-of-the-art point cloud registration methods in the literature.

Keywords: point cloud registration, Fast Point Feature Histograms (FPFH) descriptor, quantile assignment, iterative closest point algorithm, bipartite graph matching, Hungarian algorithm, Hopcroft-Karp algorithm.

ÖZET

NİCELİKSEL ATAMA YÖNTEMİ İLE NOKTA BULUTU EŞLEŞTİRME PROBLEMİNİN ÇÖZÜLMESİ

Ecenur Oğuz
Endüstri Mühendisliği, Yüksek Lisans
Tez Danışmanları: Mustafa Ç. Pınar ve Oya Karaşan
Temmuz 2023

Nokta bulutu eşleştirme, bilgisayarlı görü alanında geniş uygulama alanları olan temel bir problemdir. Bu problem genel olarak üç kısımdan oluşur: öznitelik belirleme, tekabüliyet eşleştirme ve transformasyon belirleme. Kaynak ve hedef nokta bulutları arasındaki tekabüliyet setinin belirlenmesi ile nokta bulutu eşleştirme probleminde kullanılması için niceliksel atama problemini tanımladık ve bu problem için bir çözüm algoritması önerdik. Niceliksel atama algoritmamızla birlikte kullanmak için yaygınlıkla kullanılan farklı öznitelik tanımlama ve transformasyon belirleme yöntemlerini analiz ettik. Literatürde sıklıkla kullanılan üç boyutlu modeller ile oluşturduğumuz veri kümesini kullanarak bu yöntemlerin algoritmamızla birlikte kullanıldığındaki performansını test ettik. Önermiş olduğumuz niceliksel atama yöntemi ile birlikte kullanılabilecek en uyumlu yöntemleri tespit ettik ve yeni bir baştan-uca ikili nokta bulutu eşleştirme çatısı önerdik. Son olarak, geliştirdiğimiz çatıyı iç mekan ve dış mekan denektaşı veri kümeleri üstünde test ettik ve sonuçlarımızı literatürdeki güncel olan en iyi nokta bulutu eşleştirme yöntemleri ile karşılaştırdık.

Anahtar sözcükler: nokta bulutu eşleştirme, Hızlı Nokta Özellik Histogramları (FPFH) tanımlayıcısı, niceliksel atama, iteratif en yakın nokta algoritması, iki parçalı çizge eşleme, Hungarian algoritması, Hopcroft-Karp algoritması.

Acknowledgement

I would like to express my deepest gratitude to my advisors Prof. Mustafa Ç. Pınar and Prof. Oya Karaşan for the amazing support, mentorship, and kindness they have provided me during the past three years. It has been a great pleasure and privilege to work under their supervision.

I also would like to thank Prof. Uğur Güdükbay. His invaluable wisdom and guidance have inspired me to apply my skills in new fields. Furthermore, I would like to express my appreciation to Yalım Doğan for his assistance and collaboration throughout this project. His expertise and willingness to share his knowledge have been immensely beneficial.

I would like to thank Prof. Yusuf Sahillioğlu for devoting his valuable time to reading and reviewing my thesis as well as for the insightful comments and suggestions he provided. I would also like to thank all the professors and staff in the Department of Industrial Engineering for their help and support.

I am extremely grateful to my amazing friends Deniz Şahin, Göksu Ece Okur, and Deniz Akkaya for their endless support and countless memories. Their companionship has made this journey unforgettable. I am also grateful for my dear friends Ece Ünal and Pelin Erdil, who have been nothing but supportive. I would like to thank Doğaç Eldenk for always being there in times of need and making difficult times bearable.

Finally, I want to express my gratitude to my dear family; Sedef Tanyel, Mehmet Oğuz, and Alihan Oğuz for always supporting and believing in me. I am indebted to them for everything I have achieved and I will achieve in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis introduces a new coarse-to-fine approach to solving the pairwise 3D point cloud registration (PCR) problem. PCR aims to align two or more point clouds in a standard coordinate system by estimating the transformation that maps one point cloud onto another. A point cloud is a set of data points in X, Y, and Z coordinates representing a 3D shape or object. Range sensors such as ultrasonic sensors, Kinect, and LiDAR are widely used technologies to gather point cloud data [2]. Since these sensors have a limited view range, the existing technologies cannot represent a complete scene for larger shapes or objects. PCR combines the point clouds and obtains a complete 3D scene; therefore, it is a fundamental task in computer vision and robotics with many applications such as 3D reconstruction, 3D localization, and pose estimation [3].

The pairwise PCR problem mainly involves detecting the corresponding point pairs between the two clouds (source and target) and calculating the transformation matrix (rotation and translation) that minimizes the distance between the corresponding points. For many applications, the source and the target point cloud do

not match exactly and only partially overlap [4]. To deal with this challenge, we defined the quantile assignment problem to obtain the correspondence set using a bipartite assignment approach where we aim to detect the point pairs that belong to the overlapping region and find an accurate matching for those pairs only.

1.1 Problem Definition

Let $\mathcal{S} = \{p_1, \ldots, p_N\}$ and $\mathcal{T} = \{q_1, \ldots, q_M\}$ be the source and the target point clouds where p_i and q_j are the coordinate vectors of the i^{th} and j^{th} points of their respective point cloud, $\mathcal{S}, \mathcal{T} \subset \mathbb{R}^3$.

The goal of PCR is to find the rotation matrix R and the translation vector t that minimizes the distance between the source and the target point cloud, i.e., solve the following optimization problem:

$$\min_{\substack{R \in \mathbb{R}^{3\times 3} \\ t \in \mathbb{R}^3}} d(R\mathcal{S} + t, \mathcal{T}).$$

We define the correspondence set $\mathcal{C} \subset \mathbb{N}^2$ as the one-to-one mapping between $\mathcal{S}$ and $\mathcal{T}$, i.e. the sets representing the points of the source and the target point clouds where $\mathbb{N}$ denotes the natural numbers. Let (i, j) be any tuple in $\mathcal{C}$, then the points $p_i \in \mathcal{S}$ and $q_j \in \mathcal{T}$ are said to correspond. Assume this $\mathcal{C}$ is known, then $d(\mathcal{S}, \mathcal{T})$, which denotes the distance between $\mathcal{S}$ and $\mathcal{T}$, is defined as

$$d(\mathcal{S}, \mathcal{T}) = \sum_{(i,j)\in\mathcal{C}} \|p_i - q_j\|^2.$$

When $\mathcal{C}$ is known, the optimization problem above has a closed-form solution [5]. However, obtaining an accurate enough correspondence set is a challenging task.

1.2 Challenges

Range sensing technologies have evolved rapidly throughout recent years [6]; however, the point cloud data gathered is still subject to noise and outliers. Moreover, as mentioned earlier, the point clouds to be registered have only partial overlaps. These limitations make it difficult to accurately detect the points in the source point cloud belonging to the overlapping region and the points in the target point cloud they correspond to. Since the initial positions of the two clouds are on different coordinate systems, only using the Euclidean distance between the data points for registration may result in false alignments. Therefore pose-invariant local descriptors are widely utilized [7] to detect the corresponding pairs, and remarkable developments have been made regarding these descriptors. Still, the accuracy of the 3D point cloud registration is limited by the robustness of descriptors, and improvement is needed for better results.

In general, input point clouds may contain up to billions of points, and therefore PCR applications have been limited by high memory footprint and slow speed [8]. Downsampling methods are utilized to deal with more extensive data. However, downsampling the point clouds may result in losing some of the descriptive features of surfaces. Thus, there is a trade-off between accuracy and computational complexity, which makes it challenging to deal with these large point clouds without suffering from inaccurate correspondences. When the correspondence set does not contain enough accurate corresponding pairs, the resulting transformation fails to align the source point cloud with the target point cloud.

1.3 Applications

PCR plays a critical role in various applications in computer vision and robotics. PCR algorithms represent scenes and objects in 3D by registering multiple point cloud data. This process is called 3D reconstruction. 3D reconstruction can be used for medical imaging [9], constructions of buildings, roads and bridges [10], indoor location-based services such as building maintenance and renovation planning [11], autonomous driving [12], monitoring of underground mining [13]. Simultaneous localization and mapping (SLAM) is another process in robotics that utilizes PCR methods for estimating the real-time positions of objects in unknown environments [14]. PCR application areas will likely grow as more accurate and robust techniques are developed.

1.4 Motivation

The interest in PCR has increased rapidly in recent years due to many critical applications of the technique. Many efficient learning and optimization-based algorithms can achieve accurate point cloud registration. However, there is still a need for improvement due to the many challenges of dealing with large and noisy point cloud data. Also, as mentioned earlier, the low overlap ratio between the source and target cloud imposes difficulties in estimating accurate transformations. Our study focused on detecting the overlapping region and adaptively achieving accurate registration by utilizing only the points in this region. We developed an optimization-based algorithm that does not require any initialization.

Chapter 2

Literature Review

There are many different algorithms developed to achieve efficient and accurate PCR. There are two PCR types, which are *global registration* and *local registration*. Global registration methods do not only use the point clouds' positions; hence, they do not require initialization. In contrast, local registration methods utilize the coordinates of points and rely on a rough initial alignment. As mentioned earlier, the initial alignment of the point clouds may affect the registration result adversely. A common strategy to deal with this challenge is to adopt a coarse-to-fine registration approach, which is first estimating an initial approximate transformation with coarse registration (based on pose-invariant features) and refining this transformation with fine registration (based on coordinates) [15]. Typically, global and local registration problems are solved in a coarse-to-fine approach.

The local registration (refinement) is performed generally with the well-known iterative closest point (ICP) method [16] [17] or its variants. ICP matches each point in the source cloud to its geometrically nearest point in the target cloud and transforms the source cloud such that the distance between these corresponding

points is minimized. This procedure is repeated until the distance is below a threshold. There are several standard methods for calculating the transformation: SVD, Lucas-Kanade algorithm, and Procrustes analysis [3]. In addition to the point-point distance metric, the point-plane, and plane-plane distance metrics are also used in other ICP-based methods. ICP is a fast and straightforward method. However, it converges to the nearest local minimum, and therefore its registration performance is highly dependent on the initial position of the point clouds [15]. There are many variants of ICP in the literature that made improvements to the classic method. Scale-adaptive ICP method [18] integrates the scale factor into the optimization process and handles the PCR problem when a scale difference between the input clouds is present. In the Sparse ICP method [19], the registration optimization is formulated using sparsity-inducing norms to become more robust to noise and outliers. Zhang et al. [20] developed a fast converging method using an Anderson acceleration approach. Stechschulte et al. [21] used a Markov random field model to handle outliers well when the overlap ratio between the clouds is low.

As stated earlier, refinement is typically performed with an ICP-based method. Many different approaches can be used to obtain the rough alignment needed for ICP. We classified these registration algorithms as learning-based and optimization-based ones. In the remainder of this chapter, some of the well-known methods in both categories are explained.

2.1 Learning-based Approaches

Deep neural network (DNN) methods are used extensively in the literature to perform PCR. DNN is commonly used for feature extraction and transformation estimation [3]. One common approach is establishing the features with a feature learning model and using a robust estimation tool to obtain the transformation. Random

sample consensus (RANSAC) algorithm [22] is one of the most used methods to estimate the transformation matrix. RANSAC is a search algorithm that uses repeated random sub-sampling to estimate parameters iteratively. An alternative to RANSAC is the optimized sample consensus (OSAC) algorithm [23] which has a similar principle but uses a different error metric. For both algorithms, a different set of correspondences are obtained at each iteration.

Zeng et al. introduced 3DMatch [24], a learning model that uses 3D local volumetric data and extracts 512-dimensional features representing local patches. Instead of learning from volumetric data, the PPFNet model [25] learns local descriptors on pure geometry and extracts 64-dimensional compact descriptors. Yew et al. introduced 3DFeatNet [26], which uses weak supervision for learning and leverages alignment and attention mechanisms to learn feature correspondences from GPS/INS tagged 3D point clouds without explicitly specifying them. SiamesePointNet [27], another learning model for feature extraction, uses a hierarchical encoder-decoder architecture to produce descriptors. PCR can be performed using the extracted features by one of the above models and estimating a transformation via RANSAC.

DNNs are also utilized for transformation estimation with an end-to-end framework. PCR can be solved with end-to-end neural networks by transforming the registration problem into a regression problem [3]. 3DRegNet [28] uses DNN to classify the inliers/outliers among correspondences and perform regression of the transformation parameters. Alternative to DNN, the authors also adopted a Procrustes approach for regression. Similarly, Choy et al. introduced the framework deep global registration [29], which uses a differentiable Weighted Procrustes algorithm for transformation estimation. Deep global registration uses a 6-dimensional convolutional network for correspondence confidence prediction and a robust gradient-based SE(3) optimizer for pose refinement. Gojcic et al. introduced a learning model for solving coarse and fine registration with an end-to-end neural network. Similarly, they use the Procrustes problem to estimate the initial transformation parameters and the

iterative-reweighted least squares (IRLS) problem for refinement [30].

Learning-based methods can perform fast and accurate registration. However, they have a separate training process that requires extensive data, and their performance can drop significantly for unknown scenes much different than the training data [3].

2.2 Optimization-based Approaches

Handcrafted feature descriptors are also commonly used instead of feature learning models. Handcrafted descriptors are based on spatial and geometric attributes or relationships between different points in the cloud [7]. Spin image [31] is generated by accumulating two parameters in a 2D array that describe the position of a point with respect to its neighboring points. Local feature statistics histogram (LFSH) [23] describes local shape geometries using local depth, point density, and angles between normals. Some handcrafted descriptors are built using a local reference frame (LRF). Signature of Histograms of OrienTations (SHOT) [32] descriptor is based on spatial distributions of the local neighborhoods of the key points. Fast Point Feature Histograms (FPFH) [33] is a robust descriptor based on geometric relations within the local neighborhood of a key point.

Fast global registration (FGR) [34] is a state-of-the-art optimization-based global registration method. FGR uses FPFH features for correspondence search and estimates the transformation with an alternating optimization algorithm that utilizes the Jacobian of the feature differences and the Gauss-Newton method. Probabilistic approaches are also adopted for optimization-based registration algorithms, such as the coherent point drift (CPD) algorithm. CPD [35] transforms the registration problem into a likelihood maximization problem using Gaussian mixture models (GMM).

Another popular approach is to use graph matching to establish the correspondence set. The iterative global similarity point (IGSP) algorithm [36] is a variant of the ICP algorithm where correspondences are obtained with the Hungarian algorithm using a hybrid distance metric. The metric used in IGSP utilizes the points' local and geometric features. The singular value decomposition (SVD) method calculates the transformation matrix.

Since optimization-based registration methods do not need training, they can be generalized well to unknown scenes. However, the performance of these methods might suffer from high computation costs.

Significant advancements in point cloud registration have been made in recent years, with a predominant emphasis on feature extraction and transformation estimation. However, we have observed a relative lack of attention given to the crucial task of feature-matching or correspondence estimation by computer vision researchers. Typically, existing techniques such as maximum matching or nearest neighbor approaches are employed for this task, but they might fall short in terms of accuracy in the presence of noise and partial overlap. The correspondence set directly affects the end transformation and, thus, is a crucial part of the registration process. Therefore, integrating more sophisticated novel feature-matching techniques to point cloud registration frameworks would help to achieve higher accuracy and precision. The main focus of this thesis is developing and applying the quantile assignment algorithm for precise correspondence estimation. We defined the quantile assignment problem specifically for the registration task considering the limitations such as noise, outliers, and partial overlap between the input point clouds. We analyzed different feature descriptors and transformation estimation methods to combine with our correspondence estimation algorithm and proposed a new optimization-based framework.

Chapter 3

The Quantile Assignment Problem

We used the previously reported quantile assignment algorithm [4][1] to obtain the correspondence set needed for estimating a transformation that aligns the source cloud with the target cloud. The assignment problem was shown as polynomially solvable [37], and the quantile assignment problem is a variant of the maximum bipartite assignment problem.

3.1 Problem Definition

Let the sets $S = \{p_1, \ldots, p_N\}$ and $T = \{q_1, \ldots, q_M\}$ represent points in the source and target cloud respectively. Without loss of generality, let $M \geq N$. The feature of each point is calculated using a local descriptor. Note that these features can be a scalar or a vector depending on the descriptor. Let p_i be the feature value/vector of i^{th} point in S and q_j be the feature value/vector of j^{th} point in T. Using these

[1] Part of this chapter was previously published elsewhere.

features, we construct the affinity matrix $A_{N\times M}$ where $A_{ij} = -\rho e^{\|p_i - q_j\|_2}$ and ρ is the penalty coefficient.

The standard maximum weight assignment problem in graph $G = (S \cup T, A)$ where S, T, and A are defined as above solves the following mathematical model.

$$\begin{aligned}
\max \quad & \sum_{n=1}^{N}\sum_{m=1}^{M} A_{nm}X_{nm} \\
\text{s.t.} \quad & \sum_{n=1}^{N} X_{nm} = 1, && m \in \{1,\dots,M\} \\
& \sum_{m=1}^{M} X_{nm} = 1, && n \in \{1,\dots,N\} \\
& X \in \{0,1\}^{N\times M}
\end{aligned}$$

The following relaxation is defined over the polytope of doubly stochastic matrices is exact for this problem [38, 39].

$$\begin{aligned}
\max \quad & \sum_{n=1}^{N}\sum_{m=1}^{M} A_{nm}X_{nm} \\
\text{s.t.} \quad & \sum_{n=1}^{N} X_{nm} = 1, && m \in \{1,\dots,M\} \\
& \sum_{m=1}^{M} X_{nm} = 1, && n \in \{1,\dots,N\} \\
& X_{nm} \geq 0 \text{ for } n \in \{1,\dots,N\}, m \in \{1,\dots,M\}.
\end{aligned}$$

We defined $\alpha \in [0,1]$ as the overlap ratio between the source and the target clouds. In practical applications, noise makes the overlap ratio difficult to estimate. Let us

assume there is no noise, and $\mathcal{K} \subseteq \mathbb{N}$ is the set of points in the source cloud (smaller cloud without loss of generality) that matches exactly to the points in the target cloud where $\mathbb{N}$ denotes the natural numbers. Then we set $\alpha = \dfrac{|\mathcal{K}|}{N}$, representing the ratio of expected matches for the smaller point cloud. Using this value, we define $k_\alpha = \max\left(1, \lceil (1-\alpha)N \rceil\right)$. We denote the distinct entries of the matrix A as q values. The k_α^{th} smallest q value is called the α-quantile, q_α.

Given an affinity matrix A, let

$$\mathcal{X} = \left\{ X \in \{0,1\}^{N \times M} : \sum_{n=1}^{N} X_{nm} = 1, m \in \{1,\ldots,M\} \text{ and } \sum_{m=1}^{M} X_{nm} = 1, n \in \{1,\ldots,N\} \right\}$$

be the set of all bipartite matchings, i.e., the set of permutation matrices. Given α, the objective of the quantile assignment problem is to find the bipartite matching on the affinity matrix A that maximizes the α-quantile of the weights associated with a bipartite matching problem, i.e., solve the following optimization problem.

$$\max_{X \in \mathcal{X}} \quad q_\alpha\left(\{A_{nm}X_{nm},\ n \in \{1,\ldots,N\}, m \in \{1,\ldots,M\}\right).$$

It was shown that the problem is polynomially solvable with the proposed algorithm, and an integer programming formulation is reported for completeness [4]. After the correspondence set is established using the quantile assignment problem, the transformation matrix is estimated using only the matches with larger affinity values than the optimal k_α^{th} smallest affinity value since we assume that the matches with smaller affinity values belong to the non-overlapping region. Hence, we avoid using faulty matches that could disrupt the calculation of our transformation matrix.

3.2 Solution Method

3.2.1 Algorithms

We can solve the quantile assignment problem by conducting a binary search on the affinity matrix's distinct entries called the q values. The q values are sequentially tested at each binary search step for $\alpha-feasibility$. A particular q value is considered $\alpha - feasible$ if there exists $X \in \mathcal{X}$ such that the k_α^{th} smallest affinity value in $\{A_{nm}X_{nm},\ n \in \{1,\ldots,N\}, m \in \{1,\ldots,M\}\}$ is at least as high as q [4]. We aim to find the largest α-*feasible* q value.

This property is tested by constructing a binary cost matrix C depending on the q value and solving the assignment problem on this C. A Hungarian-based method is proposed to test whether a q value is $\alpha-feasible$. This method first constructs C by assigning each entry of A the value one if that entry is smaller than q and zero if not. Then, the minimum cost matching problem can be solved on C using the Hungarian algorithm [40]. The current q value is considered $\alpha - feasible$ if the minimum cost found with the Hungarian algorithm is less than or equal to $k_\alpha - 1$. Note that the source and the target clouds are generally not the same sizes; hence, our matrix C may not be square. Since the Hungarian algorithm can only be performed in square matrices, we complement C to a square one by adding appropriate dummy rows or columns [4].

Alternatively, a Hopcroft-Karp-based method is proposed. The Hopcroft-Karp maximum cardinality matching algorithm [41] is applied in a bipartite graph that is not necessarily complete or balanced to test the α-*feasibility* of a q value. The edges of this bipartite graph will correspond to pairs in A that have zero corresponding costs in C. Augmenting the maximum cardinality matching edges in this graph with enough edges with cost values as one will result in a minimum cost matching.

Similarly, if the cost computed for C is less than or equal to $k_\alpha - 1$, one can conclude that the current q value is $\alpha - feasible$. A third option for testing the $\alpha - feasibility$ is to adopt the method reported by Itai et al. [42]. We will refer to this method as the Bellman-Ford-based method. In [42], the maximum cardinality problem on a bipartite graph can be solved by adding source and sink nodes to the graph and solving the minimum cost maximum flow problem similar to the second alternative. Ford and Fulkerson's augmenting path algorithm [43] can be applied through the shortest paths in the residual network. Since the residual network will have negative costs, the Bellman-Ford algorithm [24] is the proper algorithm to find the shortest paths [4].

When the largest $\alpha - feasible$ q value is obtained, the matching found for that q is utilized for constructing the correspondence set. Matched points with an affinity value smaller than q are excluded from the matching, and the remaining pairs are used for estimating the transformation matrix [4]. However, recall that the matching is calculated on the binary cost matrix C; therefore, each potential pair with an affinity value higher than q is treated the same. To further differentiate between the potential point pairs, one can use the cost matrix C_q with the following entries instead to obtain the correspondence set:

$$(C_q)_{ij} := \begin{cases} A_{ij}, & \text{if } A_{ij} \geq q, \\ 0, & \text{otherwise.} \end{cases}$$

Once the largest $\alpha - feasible$ q value is found, the maximum cost matching problem is solved on this matrix one last time with the Hungarian algorithm to obtain the final matching. With this modification, we aim to detect the pairs belonging to the overlapping region of the two point clouds and search for the best matching among the overlapping pairs. We will refer to this method as *Hungarian cost sensitive* in the remainder of the sequel.

Lemma. *Given α, q, and A, deciding whether there exists a matching such that the k_α^{th} smallest entry in this matching is at least q can be done in $O(M^{1.5}N)$ time.*

Proof. It has been shown in [42] that given a bipartite graph, finding a matching with the additional restriction that no more than a given number of edges from a specified subset are in the matching can be done in $O(N^3)$ time. We can adapt their algorithm to solve the problem at hand. Let the subset of edges be those with affinity values strictly less than the given q value, and the restriction is that no more than $k_\alpha - 1$ edges should be used from this set. Then, any perfect cardinality matching in this graph will correspond to a perfect matching in the underlying weighted bipartite graph with the k_α^{th} smallest entry at least as large as the given q value. This bound can further be improved by eliminating the edges with affinity values less than q from the bipartite graph and solving for a maximum cardinality matching via the Hopcropt-Karp algorithm [41]. If the number of edges to complement the matching to a perfect one is no more than $k_\alpha - 1$, then the answer is yes. Since the complexity of the Hopcropt-Karp maximum cardinality finding algorithm is $O((\#edges + \#nodes)\sqrt{\#nodes})$ [41], our result follows. □

Our main result follows from a binary search on possible q values, i.e., the distinct entries in the affinity matrix.

Theorem. *Finding a perfect matching that maximizes the α-quantile can be done in $O(M^{1.5}N \log M)$ time.*

Proof. MN elements can be sorted in $O(MN \log MN)$ time. After applying a binary search to these elements, the largest value will be determined after at most $\log MN$ attempts. Since each requires $O(M^{1.5}N)$ time, the result follows. □

The above result can also be alternatively reached. In [44], the authors show that the problem of maximizing the k^{th} smallest entry in a combinatorial optimization

problem can be achieved in time complexity $O(T \log p)$ where p is the number of variables in the optimization problem and $O(T)$ corresponds to solving the underlying optimization problem with unitary data. In the special case when the combinatorial optimization problem is finding a perfect cardinality matching, the same complexity is attained.

The pseudocode for solving the quantile assignment problem [4] is depicted in Algorithm 1.

Algorithm 1 Quantile assignment

Input: $N \times M$ affinity matrix A and $\alpha \in [0, 1]$
Output: ${q_\alpha}^*$

Initialize:
$q^* \leftarrow 0$
$k_\alpha \leftarrow \max(1, \lceil (1-\alpha)N \rceil)$
Sort distinct entries of $(\{A_{ij}, i \in \mathcal{N}, j \in \mathcal{M}\})$ into $q_1 < q_2, \ldots, < q_s$
$left \leftarrow 1$
$right \leftarrow s$
while $left \leq right$ **do**
  $mid \leftarrow \lfloor \frac{left+right}{2} \rfloor$
  $C_{ij} = \begin{cases} 1 \text{ if } A_{ij} < q_{mid} \\ 0 \text{ otherwise} \end{cases} \quad i \in \mathcal{N}, j \in \mathcal{M}.$
  $cost \leftarrow$ minimum cost perfect matching value for data $C = [C_{ij}]$
  if $cost \leq k_\alpha - 1$ **then**
    $q^* \leftarrow q_{mid}$
    $left \leftarrow mid + 1$
  else
    $right \leftarrow mid - 1$
  end if
end while
return ${q_\alpha}^*$

3.2.2 Integer Linear Programming Formulation

Even though we have established above that the problem is solvable in polynomial time, the problem is immediately amenable to a mixed integer linear programming formulation that off-the-shelf solvers can process. For the interested reader, we provide this formulation below. Let $A* = \max\{A_{mn} : m, n \in \mathcal{N}\}$ be the largest affinity value. Let us partition the matched edge variables defined for all $m, n \in \mathcal{N}$ into two, namely,

$$x_{mn} = \begin{cases} 1 \text{ if matched edge } \{m, n\} \text{ has } A_{mn} \geq q \\ 0 \text{ otherwise} \end{cases}$$

and

$$y_{mn} = \begin{cases} 1 \text{ if matched edge } \{m, n\} \text{ has } A_{mn} < q \\ 0 \text{ otherwise} \end{cases} .$$

Then, the following model will solve the quantile bipartite perfect matching problem.

$$\max \quad q \tag{3.1a}$$

$$\text{s.t.} \sum_{m=1}^{N}(x_{mn} + y_{mn}) = 1 \qquad n \in \mathcal{N}, \tag{3.1b}$$

$$\sum_{n=1}^{N}(x_{mn} + y_{mn}) = 1 \qquad m \in \mathcal{N}, \tag{3.1c}$$

$$A_{mn}x_{mn} \geq q - \mathrm{A}^*(1 - x_{mn}) \qquad m, n \in \mathcal{N}, \tag{3.1d}$$

$$A_{mn}y_{mn} \leq q - 1 \qquad m, n \in \mathcal{N}, \tag{3.1e}$$

$$\sum_{m=1}^{N}\sum_{n=1}^{N} y_{mn} \leq k_\alpha - 1, \tag{3.1f}$$

$$x_{mn}, y_{mn} \in \{0, 1\} \qquad m, n \in \mathcal{N}. \tag{3.1g}$$

Constraints (3.1b) and (3.1c) ensure that there will be a perfect matching. The matching edges indicated with x and y variables are forced to have corresponding affinity values at least and strictly less than q values, respectively, through constraints (3.1d) and (3.1e). With constraint (3.1f), the number of matched edges with small values is limited to the maximum allowed value for the given α. Finally, the objective function finds the maximum possible q, i.e., the maximum α-quantile value.

3.3 Numerical Example

Consider the following instance with $N = 5$, the affinity matrix

$$A = \begin{bmatrix} 19 & 13 & 8 & 1 & 14 \\ 9 & 3 & 18 & 2 & 18 \\ 17 & 15 & 7 & 14 & 19 \\ 2 & 1 & 9 & 6 & 13 \\ 17 & 20 & 13 & 14 & 15 \end{bmatrix}.$$

and $\alpha = 0.55$. One can easily find that $k_\alpha = 3$. Consider the following perfect matching.

$$A = \begin{bmatrix} \boxed{19} & 13 & 8 & 1 & 14 \\ 9 & 3 & 18 & 2 & \boxed{18} \\ 17 & 15 & 7 & \boxed{14} & 19 \\ 2 & 1 & \boxed{9} & 6 & 13 \\ 17 & \boxed{20} & 13 & 14 & 15 \end{bmatrix}.$$

The q value in this matching is the third smallest value in the set of matched elements and is equal to 18. However, the following assignment

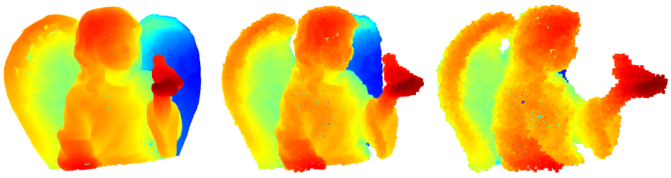
$$A = \begin{bmatrix} \boxed{19} & 13 & 8 & 1 & 14 \\ 9 & 3 & 18 & \boxed{2} & 18 \\ 17 & 15 & 7 & 14 & \boxed{19} \\ 2 & 1 & \boxed{9} & 6 & 13 \\ 17 & \boxed{20} & 13 & 14 & 15 \end{bmatrix}.$$

is preferable since it gives a q value equal to 19. The latter is the optimal value.

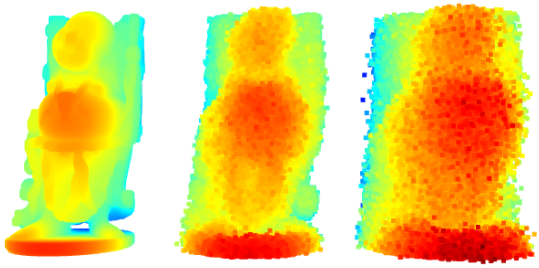
Chapter 4

Computational Experiments

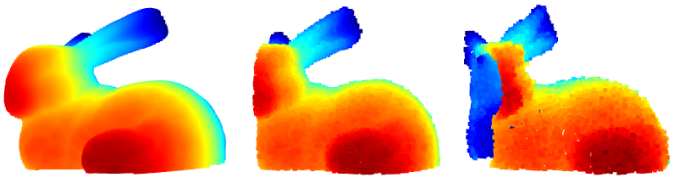
The performance of our approach explained in Chapter 3 is tested by conducting a series of PCR experiments. We constructed a synthetic dataset inspired by the experiments conducted by Zhou et al. [34]. Our dataset contains five models: the Angel, Bunny, Happy Buddha, Dragon, and Horse point clouds [45]. We generated five partially overlapping clouds for each model by cropping the models, and we added Gaussian noise to each partial cloud using three noise levels. We set the Gaussian standard deviation parameter σ equal to 0 (no noise), 0.0025 (noise level 1), and 0.005 (noise level 2) for each noise level, respectively, and multiplied this parameter with the diameter of the partial cloud. Our dataset contains 75 different point clouds, three noise levels for each model, and five point clouds for each noise level. The overlap ratio varies between 51% and 94%. Visuals of an example partial cloud from the three noise levels for each model can be seen in 4.1.



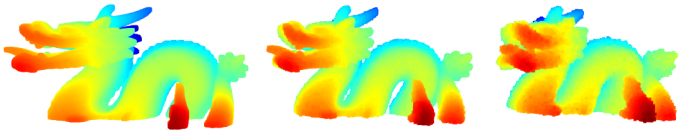
(a) Angel



(b) Happy Buddha



(c) Bunny

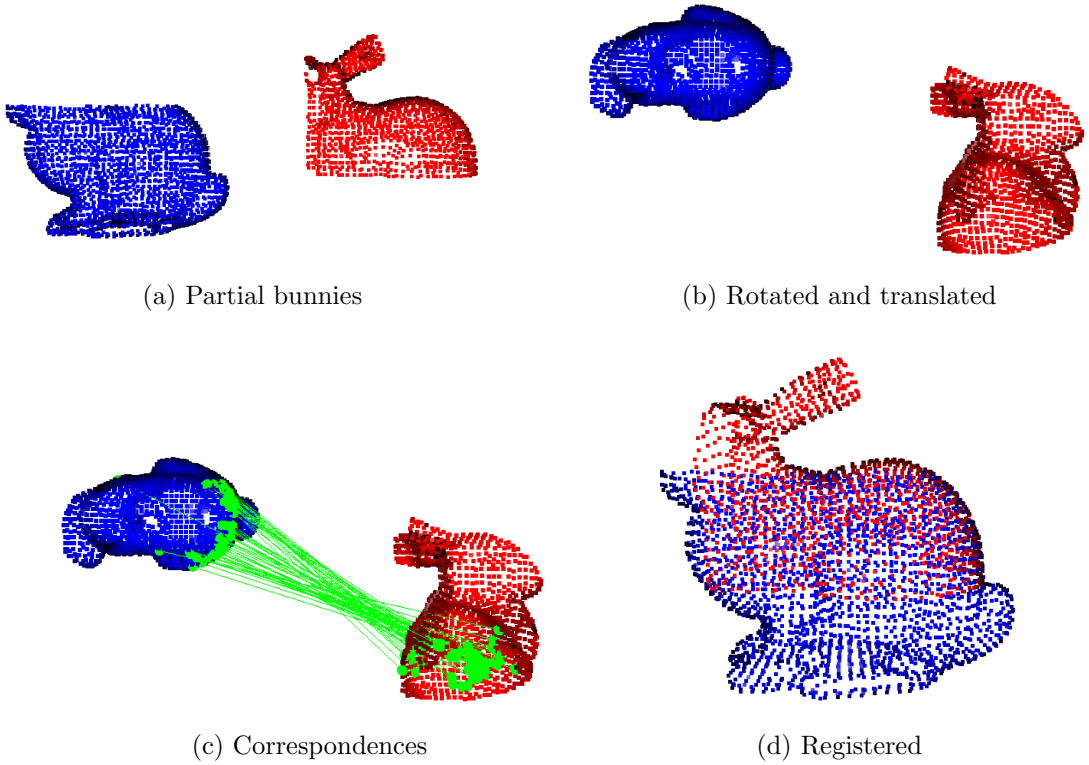


(d) Dragon

(e) Horse

Figure 4.1: Noise added partial point clouds in the synthetic dataset

An example of the PCR process for our experiments is visualized in 4.2.



(a) Partial bunnies

(b) Rotated and translated

(c) Correspondences

(d) Registered

Figure 4.2: (a) shows the initial position of the two partially overlapping clouds. Then, the source cloud (blue) is rotated by the matrix R and translated by the vector t as shown in (b) where R is constructed from the XYZ Euler angles [1] $\left[\frac{\pi}{3}, \frac{\pi}{2}, \pi\right]$, and $t = [0.5, 0.5, 0.5]$. The correspondence set for the point clouds is constructed using our quantile assignment algorithm. The correspondence lines between the point clouds for the top 50 affinity values are visualized in (c). Then the transformation matrix is calculated using the correspondence set and applied to the source cloud. The final registration is shown in (d)

Ground truth rotation and translation for the experiments conducted with synthetic data can be found by calculating the inverse of R and t. The registration

performance is evaluated using the ground truth rotation and translation. The registration is successful if the difference between the transformation calculated and the ground truth transformation is below a threshold. For the synthetic dataset, the rotation threshold is 5°, and the translation threshold is 2 cm. We used the evaluation metric *recall* in our experiments, representing the ratio of successfully registered point cloud pairs to all pairs [3].

4.1 Framework

We adopted a coarse-to-fine approach to perform PCR, consisting of global and local registration as explained in Chapter 2. The workflow of our registration process can be found in Figure 4.3.

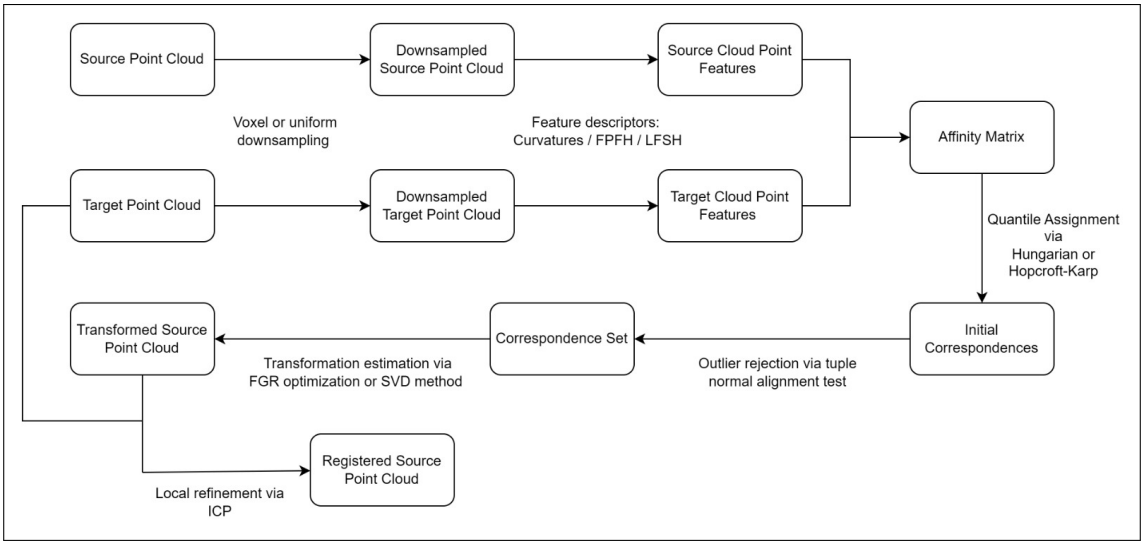


Figure 4.3: Our workflow

The inputs of our algorithm are the source and the target point clouds. The point clouds are simplified using voxel or uniform downsampling methods to deal

with large data. In voxel downsampling, points in the cloud are bucketed into voxels for a given measure, and each occupied voxel generates one point by averaging all points inside [46]. An alternative method is uniform downsampling which samples the point cloud in the order of the points. The selected point indices are $[0, k, 2k, ...]$ for a given parameter k [46].

Features of the simplified point clouds are extracted with the chosen descriptor. We tested our algorithm with three different feature descriptors: curvatures [47], FPFH [33], and LFSH [23]. The computation of each descriptor is explained, and their performances are analyzed in Section 4.2.

The affinity matrix is constructed using the chosen descriptor, as explained in Section 3.1, and the quantile assignment problem is solved on this affinity matrix with one of the algorithms explained in 3.2.1. The registration performances of different choices of algorithms are analyzed in Section 4.3. As stated earlier, the quantile assignment algorithm requires an α value as input representing the overlap ratio between the source and the target point cloud. This ratio must be estimated visually for point clouds obtained by different scans. We manually synthesized the partial point cloud pairs from a single model for our synthetic dataset. Therefore, the overlap ratio is easily computed and used as an input.

The resulting matching of our quantile assignment algorithm is the initial correspondence set. We apply a test called *tuple normal alignment test* to the initial correspondence set to eliminate false correspondences. The tuple normal alignment test process details are given in Section 4.4. After applying the test, we obtain the final correspondence set to estimate the transformation.

Given the correspondence set, we implemented two methods to estimate the transformation matrix. The first option is to perform singular value decomposition (SVD), a well-known method for estimating the rotation matrix. After the rotation matrix

is calculated with SVD and applied to the source point cloud, the translation vector is calculated by simply using the difference between the mean coordinates of the matched points among the target cloud and the rotated source cloud. We also implemented the algorithm developed by Zhou et al. [34], which we refer to as the *FGR optimization method* where FGR stands for *Fast Global Registration*. Both methods are explained, and their registration performances are evaluated in Section 4.5.

The process so far is the global registration part of our framework. Finally, we perform local registration after the found transformation is applied to the source cloud. The point-to-plane ICP algorithm (cf. Section 4.6) performs fast and accurate registration if the initial alignment of the input point clouds is close enough. Therefore, the original target cloud and the previously transformed source cloud are taken as input for the ICP algorithm to perform local refinement, and the final registration is obtained.

4.2 Feature Investigation

Feature descriptor choice is crucial for our work. The main idea is that one should associate with each point in the cloud a number or a vector that does not change with the set of possible transformations that the point cloud may have to go through to achieve an optimal registration with another point cloud. We implemented three pose-invariant local descriptors for our PCR algorithm: curvatures [47], Fast Point Feature Histograms (FPFH) [33], and Local Feature Statistics Histogram (LFSH) [23]. In this chapter, computations of each of the implemented local descriptors are explained, and the performances of the features are compared according to their registration performance and descriptiveness.

Our first choice as a descriptor was curvatures. *Curvature* is a quantity preserved

under rigid transformation. Histogram shapes of local curvature for K-nearest neighbors are invariant under, e.g., affine transformations. Curvature computation for point clouds is performed as follows. For each point x_i, $i = 1, \ldots, n$ in the point cloud, let M_i be the associated unit normal vector. Fix one point $P = x_{i_0}$ and let $N = M_{i_0}$ denote its normal vector. Let another point $Q_i = x_i$ be chosen in a close neighborhood of P. The normal curvature τ_i can be estimated at any point by

$$\tau_i = \frac{\sin(\beta)}{|PQ_i| \sin(\alpha)},$$

where α denotes the angle between $-N$ and PQ_i, and β represents the angle between N and M_i (see [47] for details).

Alternative pose-invariant local features are Fast Point Feature Histograms (FPFH). FPFH is a modification of previously reported Point Feature Histograms (PFH) to be more robust. PFH is computed as follows. For a given point p, a radius r, and neighbor count k, all the points in the k-neighborhood of p are selected. These points must be enclosed in the sphere where p is the center and the radius is r. For any point p_i in the k-neighborhood of p, the estimated normal of p_i is denoted as n_i. Let p_i and p_j be any point pair in the k-neighborhood where p_i has a smaller angle between its associated normal and the line connecting the points, without loss of generality. Let $u = n_i, v = (p_j - p_i) \times u, w = u \times v$. The PFH of the point p is computed using the following values α, ϕ, θ for each such pair (p_i, p_j) in the k-neighborhood of p:

$$\begin{aligned}
&\alpha = v \cdot n_j, \\
&\phi = (u \cdot (p_j - p_i))/||p_j - p_i||, \text{and} \\
&\theta = \arctan(w \cdot n_j, u \cdot n_j).
\end{aligned}$$

The default implementation uses 11 binning subdivisions, which results in a 33-dimensional feature for each key point. FPFH is a modification of PFH where all

neighbors of p are not fully interconnected, and thus the process is simplified. After the set of tuples α, ϕ, θ between itself and some of its neighbors are computed for each point p (called Simplified Point Feature Histograms (SPFH)), k neighbors of p are re-determined and the final histogram (FPFH) is calculated as follows:

$$FPFH(p) = SPFH(p) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_i}SPFH(p_i),$$

where w_i is the weight that represents the distance between query point p and a neighbor point p_i in a given metric space. The computation times are reduced drastically for the implementation of FPFH compared to PFH, and most of the discriminative power of PFH is retained (see [33] for details).

The third local descriptor we implement for our PCR algorithm is Local Feature Statistics Histograms (LFSH). LFSH describes local shape geometries by encoding their statistical properties on local depth, point density, and angles between normals. The eigenvector corresponding to the minimum value of $Cov(p)$ is used to estimate the normal of a point p where

$$Cov(p) = \begin{bmatrix} p_1 - \bar{p} \\ \dots \\ p_k - \bar{p} \end{bmatrix}^{\top} \cdot \begin{bmatrix} p_1 - \bar{p} \\ \dots \\ p_k - \bar{p} \end{bmatrix}$$

and $\bar{p}$ is the centroid of the k-neighborhood of p.

The first aspect, local depth, is computed as follows. For a key point p, the k-neighborhood of p is selected as described before. Along the positive direction of n (normal vector of p), a 2D plane L is defined as a tangent plane of the sphere with radius r. The projection of any neighbor p_i on L is denoted as p_i'. The local depth is defined as the distance between p_i and p_i', which is:

$$d_i = r - n \cdot (q_i - q).$$

A circle of radius r is placed at the crossing point of n and L to calculate point density. This circle is divided into several parts with the same width, and the ratio of points distributed in each part is calculated using the distance of p'_i from the crossing point p'. This distance is denoted as ρ where

$$\rho_i = \sqrt{||p - p_i||^2 - (n \cdot (p - p_i))^2}.$$

The deviation angle between normals is represented as

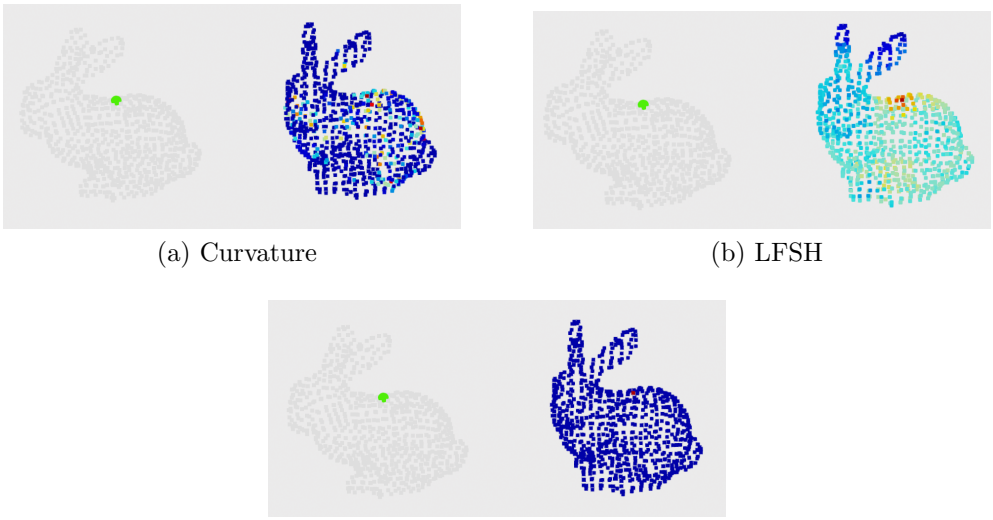
$$\theta_i = \arccos(n \cdot n_i).$$

Three sub-histograms are obtained using these feature statistics containing 10, 15, and 5 bins, respectively. The LFSH descriptor is obtained by concatenating these three sub-histograms into one histogram (see [23] for details).

The performances of the three local descriptors curvatures, FPFH and LFSH, are first tested using the Feature Visualizer tool we designed using the Open3D library [46]. For any given point in the source cloud, the location of the closest point in the target cloud regarding its feature value/vector for the given local descriptor is shown via the Feature Visualizer.

The given point for the point cloud on the left side of Figure 4.4 is shown as green. The points in the point cloud on the right are colored from red to blue using the heat map technique depending on their closeness (in terms of the used feature) to the green point. The colors are normalized, the closest points are colored red, and the furthest points are colored blue accordingly. According to our Feature Visualizer tool, the FPFH feature seems more discriminative than the other two.

We used the Bunny model from our synthetic dataset to test the registration performances of the descriptors. For each descriptor, our PCR framework is used on the ten partially overlapping pairs from each noise level, with voxel downsampling and



(a) Curvature

(b) LFSH

(c) FPFH

Figure 4.4: Images obtained from the Feature Visualizer tool

uniform downsampling for simplification. The recall values are plotted for the cases of no noise, noise level 1, and noise level 2 in Figures 4.5, 4.6, and 4.7, respectively.

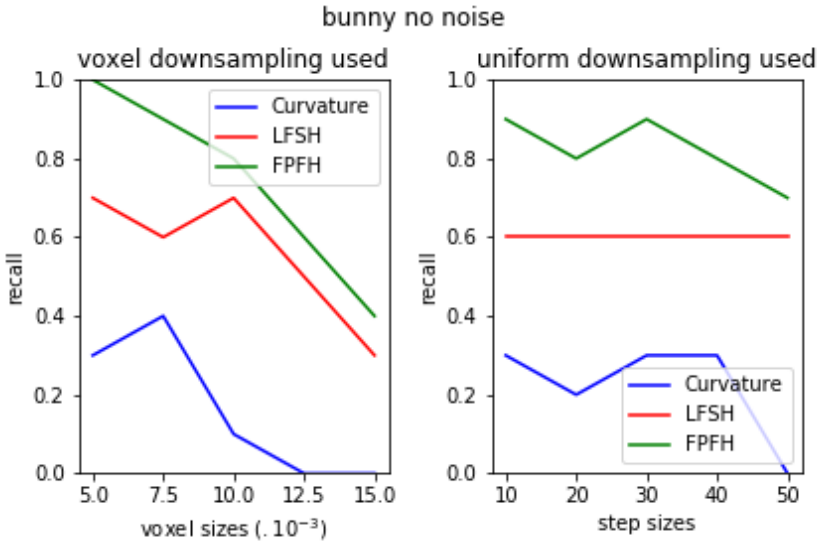


Figure 4.5: Feature recalls with no noise

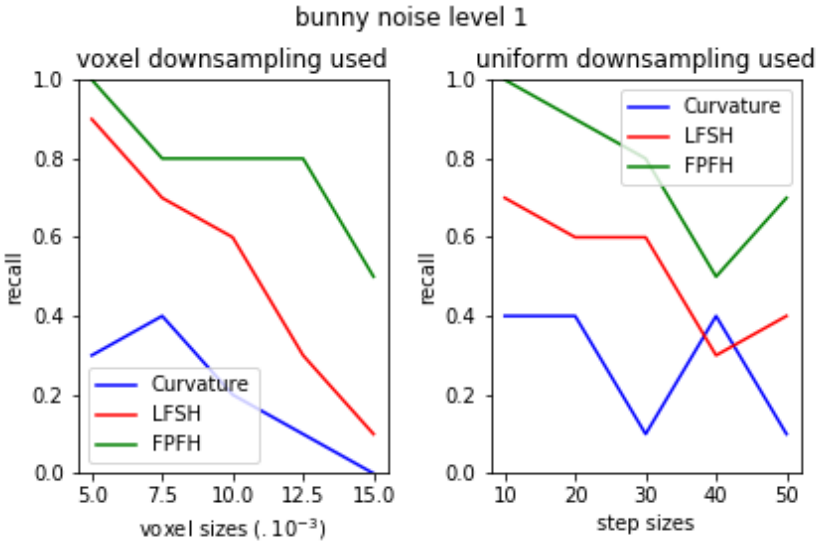


Figure 4.6: Feature recalls with noise Level 1

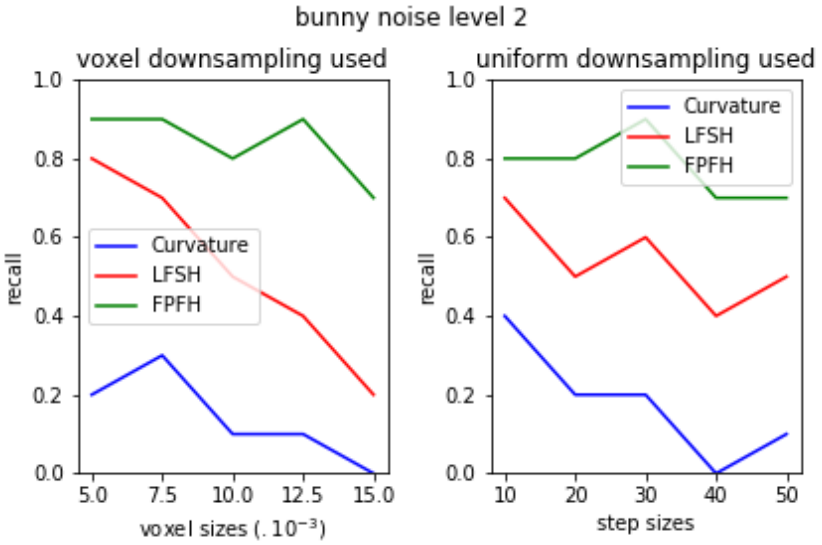


Figure 4.7: Feature recalls with noise Level 2

As a result of our experiments, the FPFH feature achieved better registration results in our PCR framework for each noise level and downsampling method compared to curvatures and LFSH.

4.3 Correspondence Set

The initial correspondence set is obtained by solving the quantile assignment problem. Several solution methods are proposed and explained in Chapter 3. Integer linear programming formulation is not preferred for implementation since it has a much higher computation cost than the algorithms. The Hungarian-based, Hopcroft-Karp-based, and Bellman-Ford-based methods give the same solution among the proposed algorithms. The Bellman-Ford-based method is much slower than the other two. The Hungarian-based and the Hopcroft-Carp-based methods have similar computation times. However, the Hopcroft-Karp-based method is preferable since the affinity matrix is not generally square (see [4] for more details regarding this comparison). The results were expected to differ for the Hungarian cost-sensitive method since the cost matrix construction was modified. Therefore, among the proposed solution methods, we used the Hopcroft-Karp-based method and the Hungarian cost-sensitive method for our computational PCR experiments since these two methods perform better than their alternatives in terms of both accuracy and computation time.

In this chapter, the registration performances of the two methods are compared. The bunny model in our synthetic dataset is again used for the experiments. The recall values and the computation times are plotted below for the two methods.

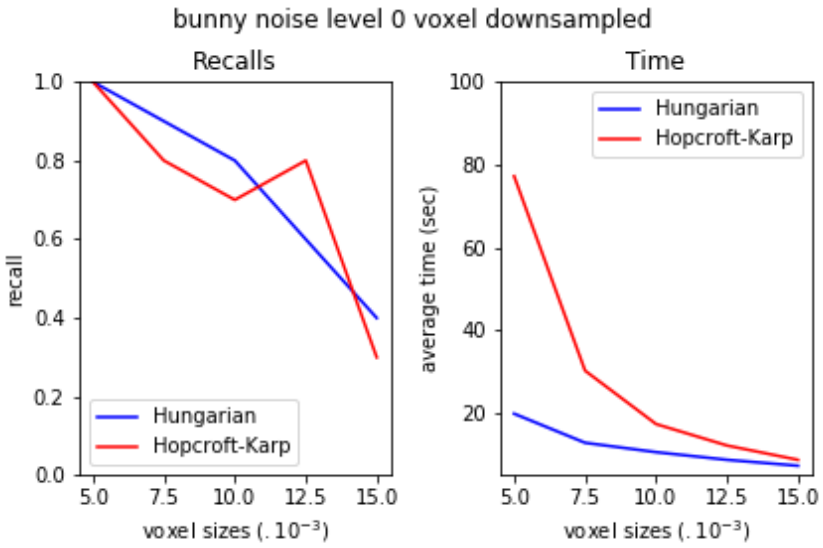


Figure 4.8: Hungarian cost-sensitive and Hopcroft-Karp-based recalls with voxel downsampling

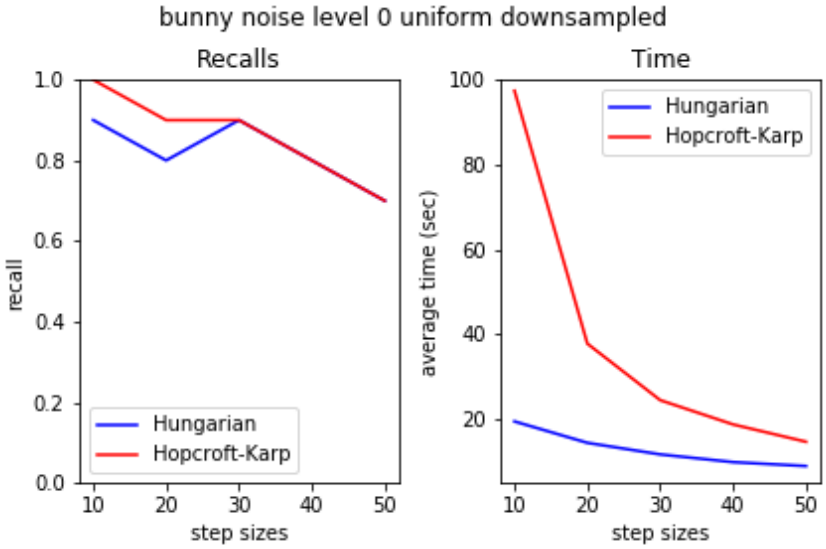


Figure 4.9: Hungarian cost-sensitive and Hopcroft-Karp-based recalls with uniform downsampling

As shown in Figures 4.8 and 4.9, the recall values resulting from the Hopcroft-Karp-based and the Hungarian cost-sensitive methods are similar. However, there is a significant difference between the computation times of the compared methods. Therefore, we prefer the Hungarian cost-sensitive method in our implementation.

4.4 Tuple Normal Alignment Test

We adopted the tuple test used in [34] and used it on our initial correspondence set to eliminate the faulty matches. The tuple test is applied as follows. Three correspondence pairs (p_1, q_1), (p_2, q_2), (p_3, q_3) are randomly selected from the set. The tuples (p_1, p_2, p_3) and (q_1, q_2, q_3) are considered compatible and pass the test if the condition below is satisfied:

$$\tau < \frac{||p_i - p_j||}{||q_i - q_j||} < \frac{1}{\tau}, \qquad \forall \quad i, j = 1, 2, 3 \quad \text{and} \quad i \neq j$$

where τ is selected as 0.9. For any two corresponding pairs (p_i, q_i), (p_j, q_j), the ratio of their distances must be close to one if any of the correspondences is not false.

The source cloud is transformed by utilizing the correspondences that pass the tuple test. Then, to further eliminate the faulty correspondences, we applied one more test using the normal alignments of the triangles formed by the tuples. We will refer to this test as the tuple normal alignment test, which is applied as follows. Three correspondence pairs (p_1, q_1), (p_2, q_2), (p_3, q_3) are randomly selected from the set. Let n_p be the surface normal of the triangle formed by the tuple (p'_1, p'_2, p'_3) and n_q be the surface normal of the triangle formed by the tuple (q_1, q_2, q_3), where (p'_1, p'_2, p'_3) is the points in the source cloud after the transformation is applied. The correspondence pairs (p_1, q_1), (p_2, q_2), (p_3, q_3) remain in the final correspondence set

if the angle between n_p and n_q is less than or equal to our threshold which is set to 15°. The final correspondence set is established in our implementation after both the tuple test and the tuple normal test is applied to the initial correspondence set.

4.5 Transformation Estimation

The well-known singular value decomposition (SVD) method can estimate the transformation matrix. After the final correspondence set is established, our aim is to compute the rotation matrix R and the translation vector t that minimizes the weighted sum of squared errors (the distance between the corresponding pairs), i.e., the following optimization problem is solved:

$$\min_{\substack{R\in\mathbb{R}^{3\times 3}\\ t\in\mathbb{R}^3}} \sum_{(i,j)\in\mathcal{C}} \|(Rp_i + t) - q_j\|^2 w_{ij}.$$

where w_{ij} weight of the corresponding pair (p_i, q_j). We used the affinity matrix values, which show the closeness of two points in terms of their features as weights. A direct optimal solution for the rotation matrix can be found via SVD. First, the weighted mean values p_0 and q_0 for the source and the target clouds, respectively, are calculated as follows:

$$p_0 = \frac{\sum_{(i,j)\in\mathcal{C}} p_i w_{ij}}{\sum_{(i,j)\in\mathcal{C}} w_{ij}}, \qquad q_0 = \frac{\sum_{(i,j)\in\mathcal{C}} q_j w_{ij}}{\sum_{(i,j)\in\mathcal{C}} w_{ij}}.$$

Next, the covariance matrix H is calculated as shown below:

$$H = \sum_{(i,j)\in\mathcal{C}} (q_j - q_o)(p_i - p_0)^\top w_{ij}$$

Finally, we apply SVD to H to decompose H into matrices U, D and V:

$$SVD(H) = UDV^\top.$$

Using the SVD decomposition, the rotation matrix R can be calculated:

$$R = VU^\top.$$

Once R is computed, the translation vector is calculated as a shift between the means of the matched points from the two point clouds:

$$t = q_0 - Rp_0.$$

It was shown that this (R, t) is the optimal solution to the optimization problem defined above (see [48] for details).

Our second option to estimate the transformation matrix is to use the FGR optimization method proposed by Zhou et al. [34]. The following objective function is used in this method:

$$E(R, t) = \sum_{(i,j)\in\mathcal{C}} \rho(||(Rp_i + t) - q_j||).$$

where a scaled Geman-McClure estimator is used as the robust penalty function $\rho(.)$:

$$\rho(x) = \frac{\mu x^2}{\mu + x^2}.$$

Since optimizing $E(R, t)$ is difficult, the authors proposed an equivalent joint objective using Black-Rangajaran duality. This equivalent objective is optimized with an alternate algorithm (see [34] for details).

The registration performances of the two methods are compared using the bunny model in our synthetic dataset. The recall values are plotted below.

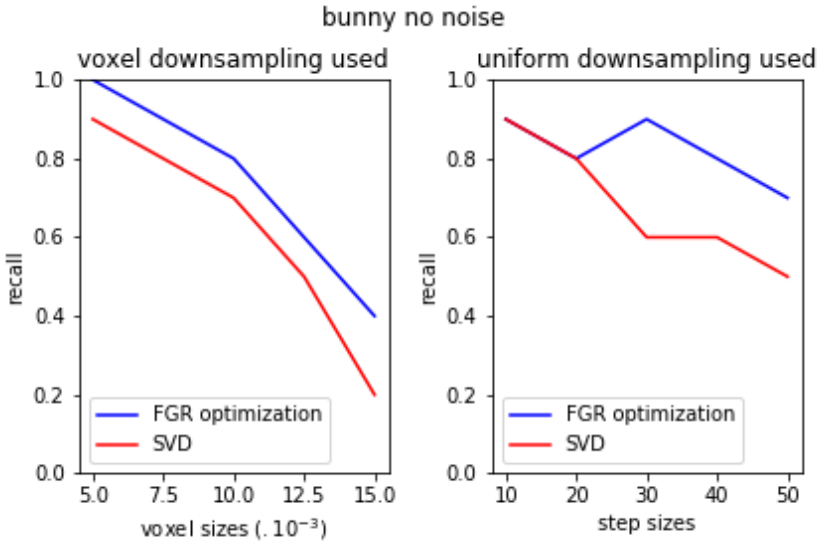


Figure 4.10: FGR optimization method and SVD recalls with no noise

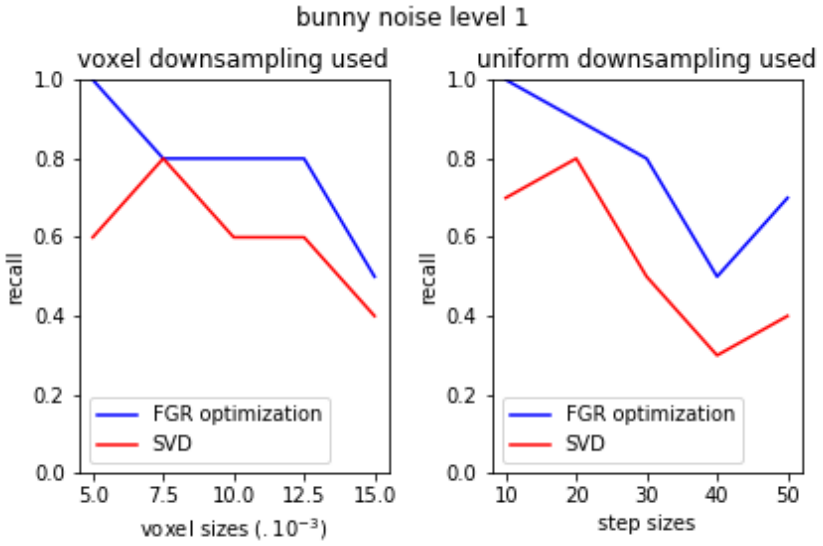


Figure 4.11: FGR optimization method and SVD recalls with noise Level 1

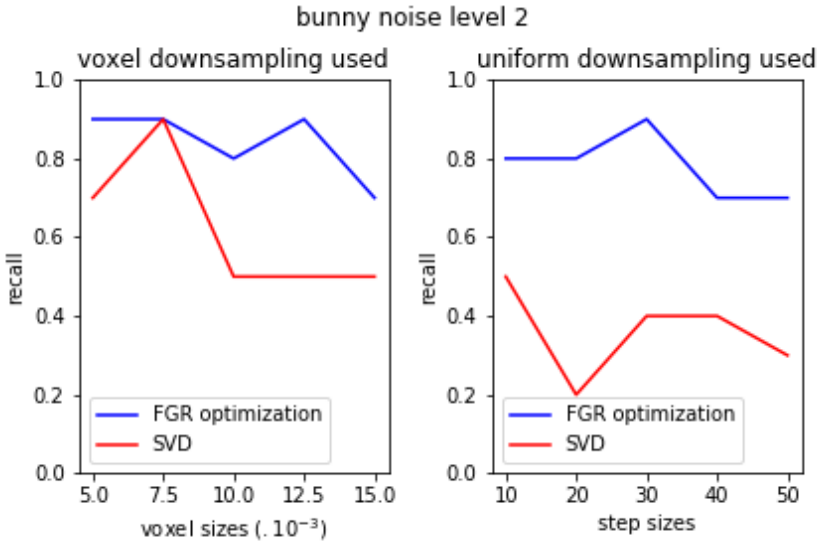


Figure 4.12: FGR optimization method and SVD recalls with noise Level 2

As shown in Figures 4.10, 4.11 and 4.12, the FGR optimization method achieves

more accurate PCR results. The parameter μ used in the objective in the FGR optimization method controls which correspondences significantly affect the objective. Therefore it is more successful in dealing with outlier correspondences.

4.6 Local Refinement via ICP

The resulting alignment of the source and the target point clouds from our global registration algorithm requires further refinement to achieve a more accurate registration result. The point-to-plane ICP method [16] is used in our implementation for local refinement. We will use the same notation in Section 1.1 to explain the point-to-plane ICP method. First, the correspondence set is established by matching each point in the source cloud to its closest neighbor point in the target point cloud. The point-to-plane ICP method considers the surface normal of the target scan. The following optimization problem is solved:

$$\min_{\substack{R\in\mathbb{R}^{3\times 3}\\ t\in\mathbb{R}^3}} \sum_{(i,j)\in\mathcal{C}} ||((Rp_i + t) - q_j) \cdot n_j||^2,$$

where n_j is the surface normal of the point q_j. This error function is minimized using the least squares approach. The Gauss-Newton method is used to compute the least squares solution, and the resulting transformation is applied to the source cloud. This process is iterated until the error function is below a certain threshold (see [16] for details).

4.7 PCR Tests

The registration performance of our algorithm is tested by running our algorithm on three different datasets and comparing our results with some of the state-of-the-art PCR methods. We used a synthetic dataset containing commonly used 3D models, one indoor benchmark dataset (3DMatch), and one outdoor benchmark dataset (KITTI). The evaluation of our method for each dataset is analyzed in this section. All tests were performed on a PC with 64-bit Operating System, x64-based Intel(R) E5-2620 v4 processor, CPU @2.10GHz, and with installed memory (RAM) equal to 64.00 GB.

4.7.1 Synthetic Dataset

We ran our point cloud registration algorithm on 150 pairs in our synthetic dataset described earlier in this chapter and compared our results with a state-of-the-art optimization-based method, Fast Global Registration algorithm [34]. We performed the registration for each point cloud pair after using both voxel and uniform downsampling with several different downsample parameters. The overlap ratio is computed easily for each cloud pair in our experiments since the point clouds are manually created, and our α parameter is set accordingly. The recall plots for different downsampling sizes for each model in our synthetic dataset and for each noise level can be found in Figures 4.13, 4.14, 4.14, 4.16, 4.17. The average recall values of the two methods are compared in Table 4.1.

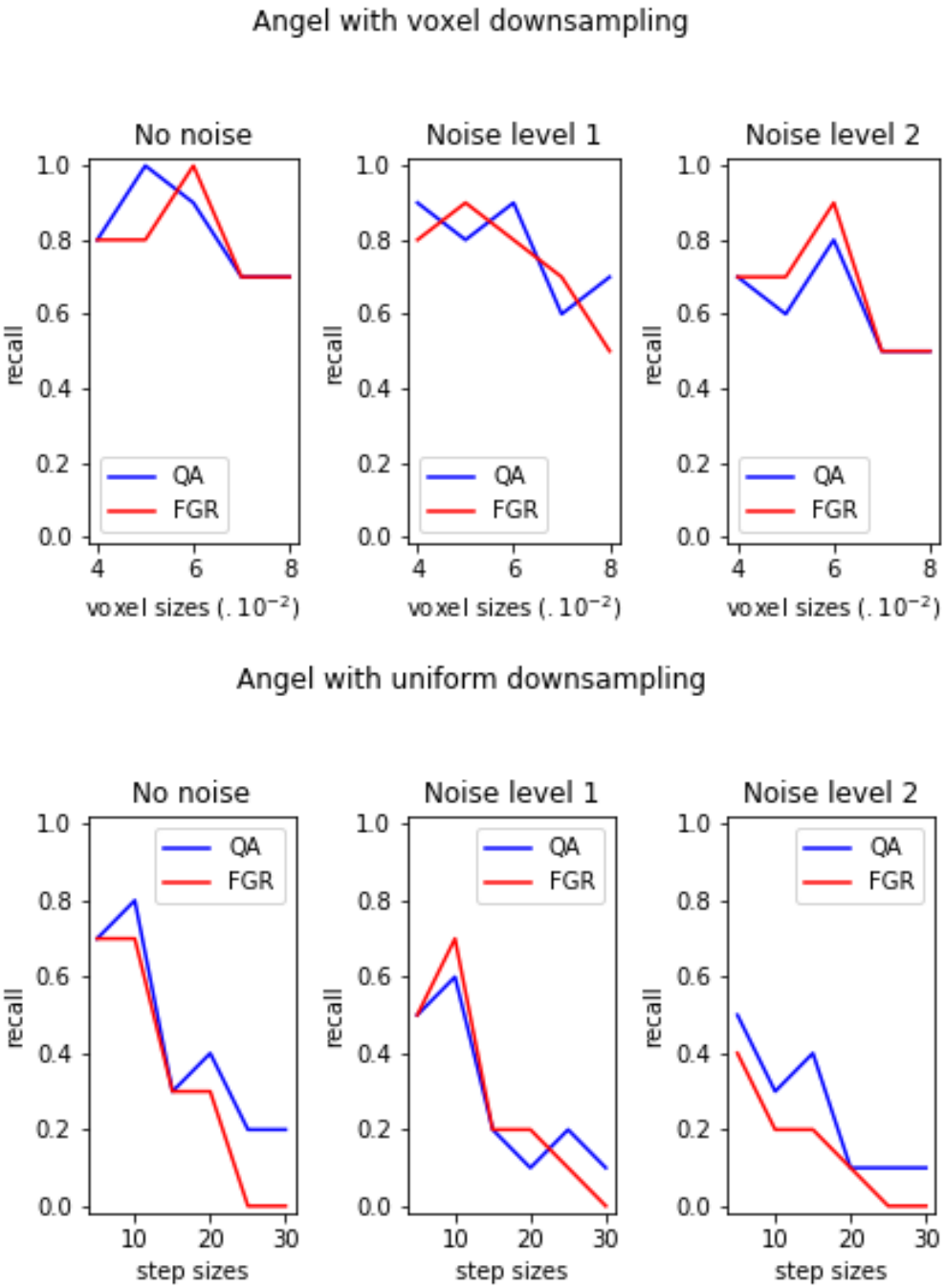


Figure 4.13: Registration results of Quantile Assignment and Fast Global Registration on Angel

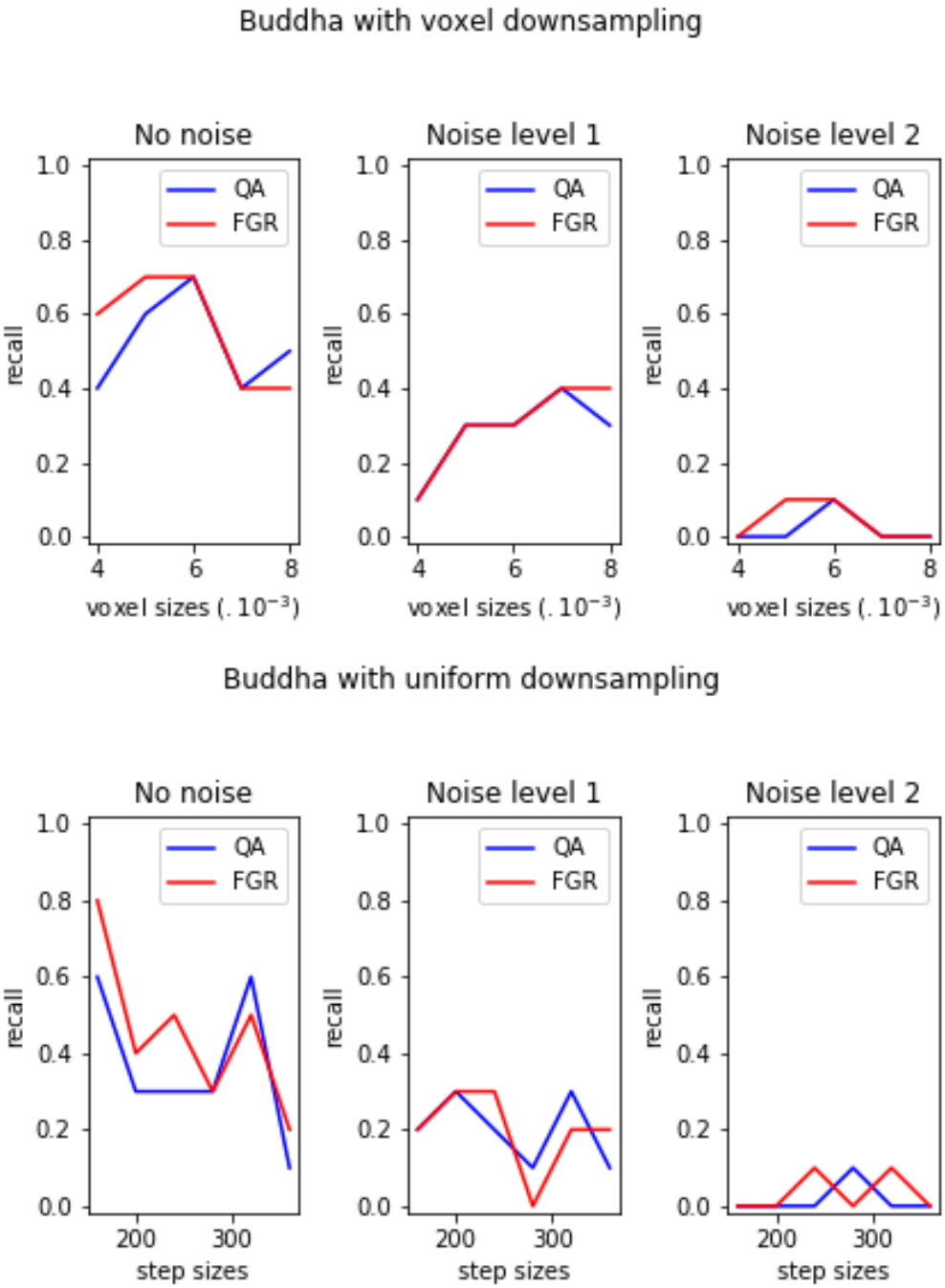


Figure 4.14: Registration results of Quantile Assignment and Fast Global Registration on Buddha

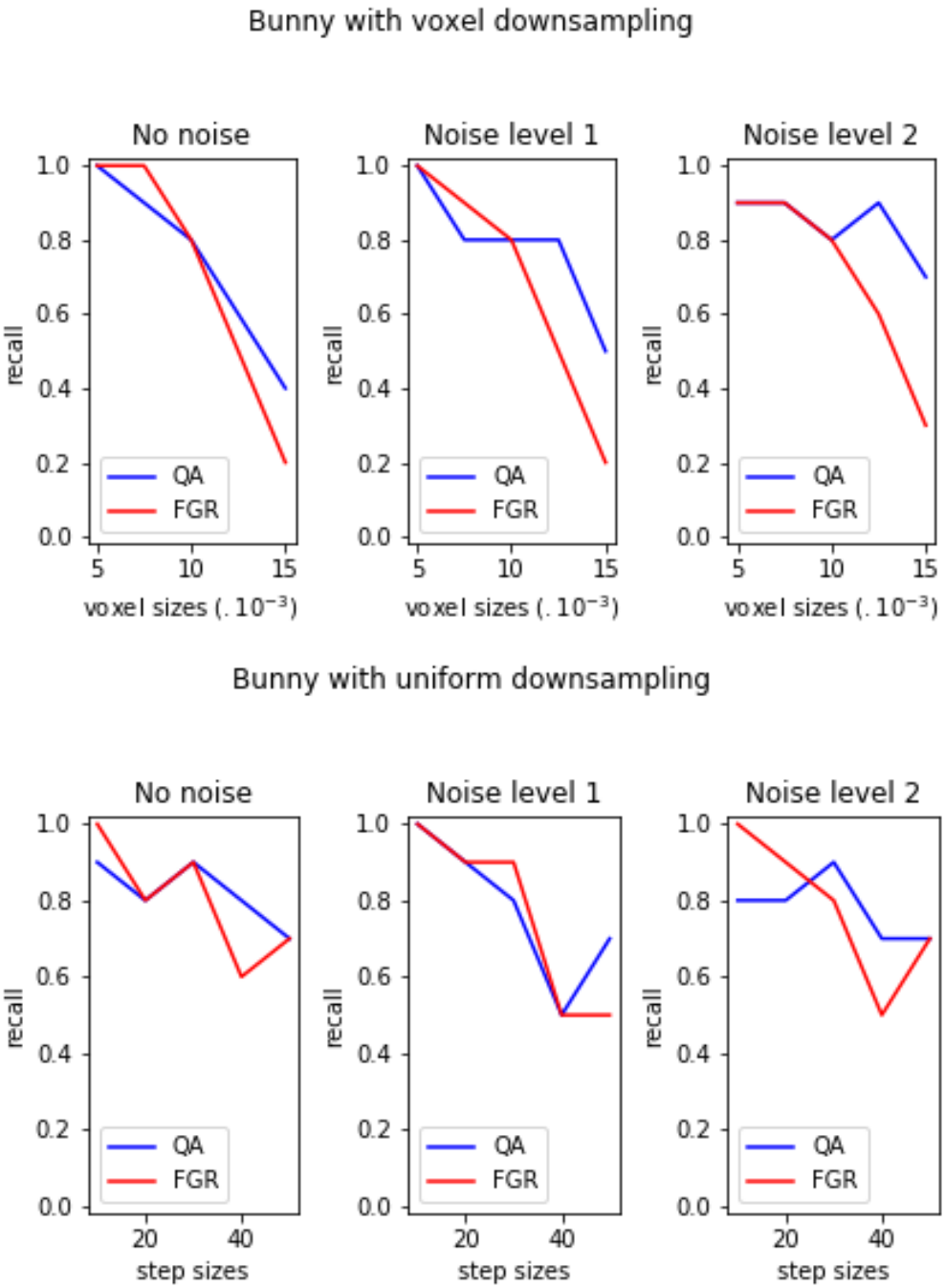


Figure 4.15: Registration results of Quantile Assignment and Fast Global Registration on Bunny

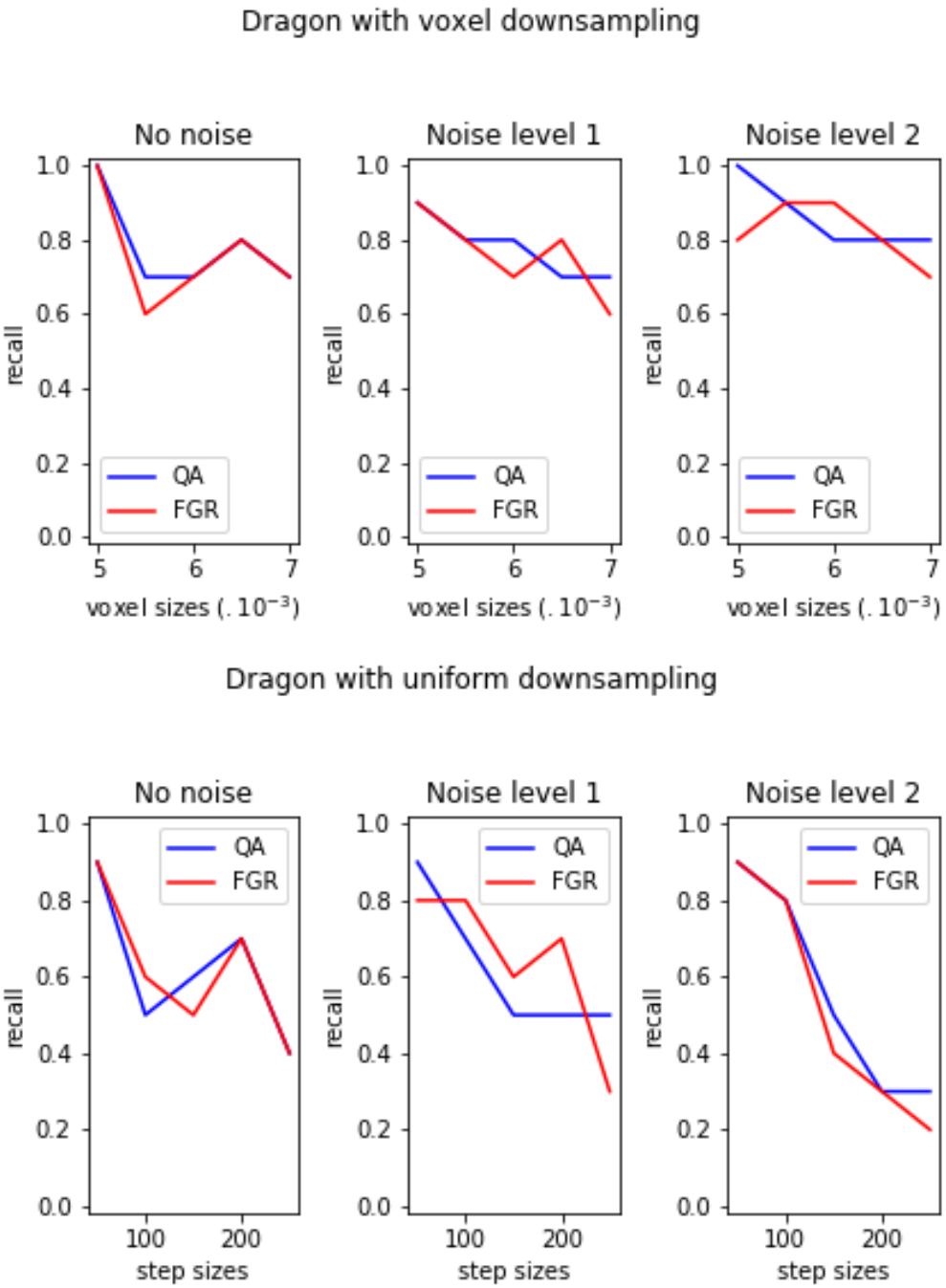


Figure 4.16: Registration results of Quantile Assignment and Fast Global Registration on Dragon

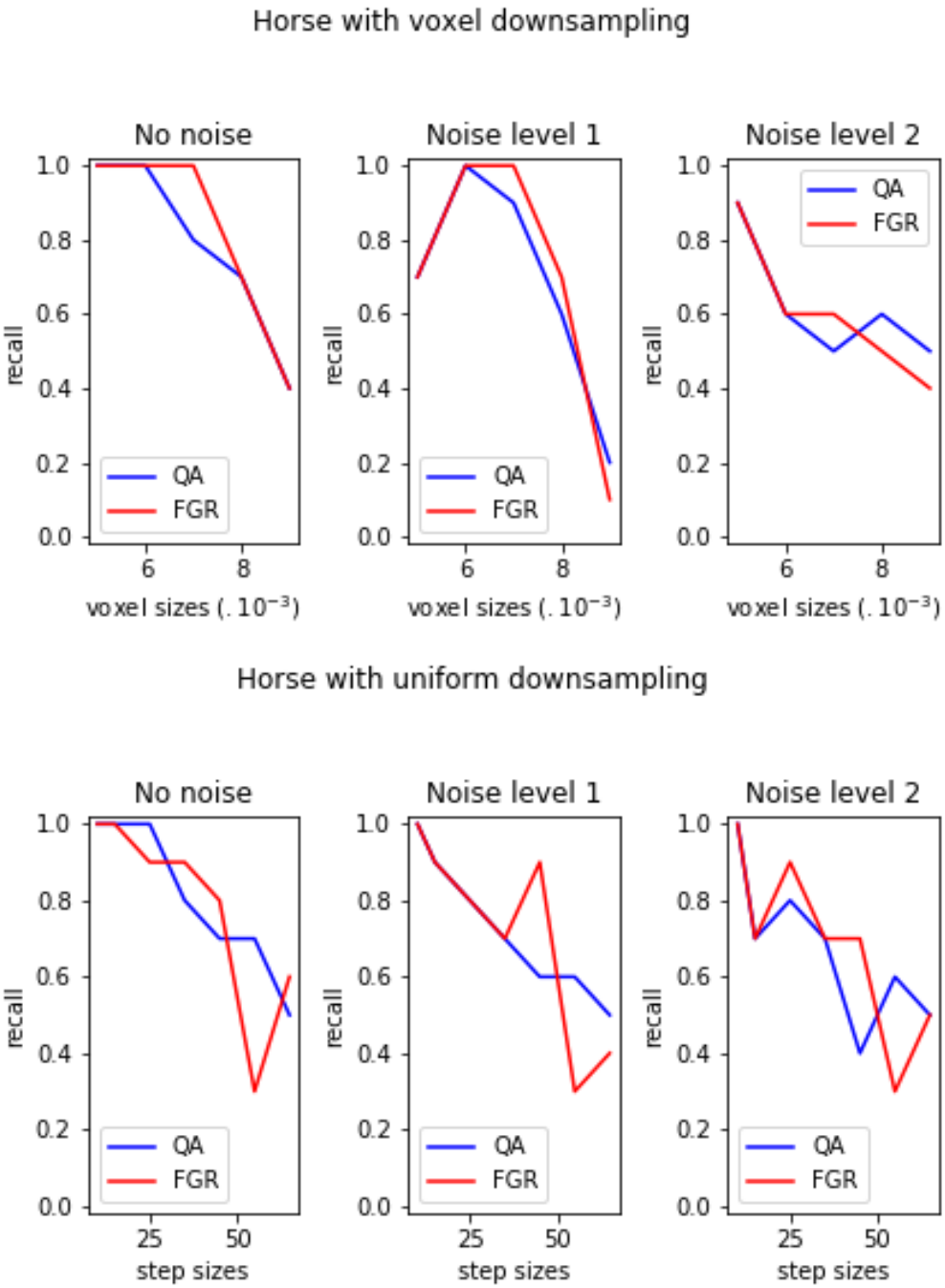


Figure 4.17: Registration results of Quantile Assignment and Fast Global Registration on Horse

Table 4.1: Average recall values of Quantile Assignment (QA) and Fast Global Registration (FGR) on the synthetic dataset

		Average Recall (%)			
		With voxel downsampling		With uniform downsampling	
		QA	FGR	QA	FGR
Angel	$\sigma = 0$	**82**	80	**43**	33
	$\sigma = 0.0025$	**78**	74	**28**	**28**
	$\sigma = 0.0050$	62	**66**	**25**	15
Buddha	$\sigma = 0$	52	**56**	42	**50**
	$\sigma = 0.0025$	28	**30**	**22**	20
	$\sigma = 0.0050$	2	**4**	2	**4**
Bunny	$\sigma = 0$	**74**	70	**82**	80
	$\sigma = 0.0025$	**78**	68	**78**	76
	$\sigma = 0.0050$	**84**	70	**78**	**78**
Dragon	$\sigma = 0$	**78**	76	**62**	**62**
	$\sigma = 0.0025$	**78**	76	62	**64**
	$\sigma = 0.0050$	**86**	82	**56**	52
Horse	$\sigma = 0$	78	**82**	**81**	79
	$\sigma = 0.0025$	68	**70**	**73**	71
	$\sigma = 0.0050$	**62**	60	67	**69**

For the synthetic dataset, the Fast Global Registration method is selected for comparison since it is one of the state-of-the-art optimization-based algorithms like our approach, and the implementation was available online [46]. The average recall values are computed for a total of 30 cases where registrations are performed with different 3D models, noise levels, and downsampling method combinations for each case. Out of these 30 cases, our algorithm outperforms the Fast Global Registration method in 17 of them and results in a tie in 3 of them. Overall, average recall values

for the two methods are comparable.

4.7.2 3DMatch Dataset

3DMatch benchmark [24] is a large-scale real-world indoor dataset that contains eight sets of indoor scene fragments captured by the RGBD sensor. Sample fragment visuals of each set can be found in Figure 4.18. Each set contains 37 to 66 fragments; each fragment is a 3D point cloud of a surface. The point clouds are integrated using the volumetric representation method Truncated Signed Distance Function (TSDF) [49].

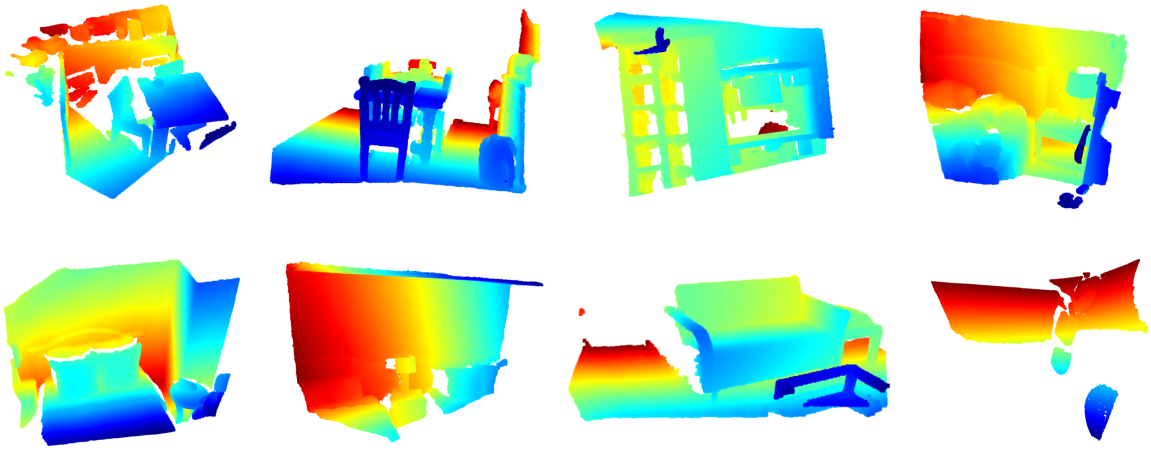


Figure 4.18: Some scene fragments from the 3DMatch dataset

We followed the evaluation protocol described in [50] to obtain the recall and precision results of our method where

$$\text{Recall} = \frac{\text{\# of true positives}}{\text{\# of ground truth loop closures}},$$

$$\text{Precision} = \frac{\#\text{ of true positives}}{\#\text{ of detected loop closures}}$$

(see [50] for details).

Unfortunately, computing the overlap ratio of the point cloud pairs in 3DMatch was not possible, however, it is known that the pairs in the dataset have $> 30\%$ overlap [51]. Therefore, we set our parameter α as 0.3 in our experiments for all point cloud pairs.

We first compare our average and maximal recall and precision results with Fast Global Registration (FGR) [34] in Table 4.2. We obtained the results for FGR using the implementation in [46]. The registration result for one sample point cloud pair in the 3DMatch dataset is shown in Figure 4.19.

Table 4.2: Our evaluation results of Quantile Assignment and Fast Global Registration on 3DMatch

Method	Average Recall (%)	Maximal Recall (%)	Average Precision (%)	Maximal Precision (%)
Quantile Assignment	9.25	15.56	11.62	34.85
Fast Global Registration	35.02	47.80	27.30	53.37

(a) Source (b) Target (c) Transformed source and target

Figure 4.19: Registration result of Quantile Assignment on a point cloud pair in the 3DMatch dataset

Registration recall values for some of the state-of-the-art learning-based PCR methods on 3DMatch can be found in Table 4.3.

Table 4.3: Feature Matching Recall (FMR) values of some learning-based PCR methods on 3DMatch

Method	Feature Matching Recall (%)
IMFNet [52]	98.6
MS-SVConv [53]	98.4
GeDi [54]	97.9
SpinNet [55]	97.6

We observed that our registration results were not comparable with the state-of-the-art methods. The overlap ratio is a piece of critical information for our approach since we use it to set our parameter α, and our algorithm is highly sensitive to this parameter. The overlap ratio for each pair in the dataset we performed the registration on varies a lot, and we could not set the α parameter differently for

different pairs. Therefore, the 3DMatch dataset is a particularly challenging dataset for our Quantile Assignment algorithm.

4.7.3 KITTI Dataset

Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) benchmark is a large-scale real-world outdoor dataset containing 555 point cloud pairs. The scenes in the dataset are captured by driving around the city of Karlsruhe with two high-resolution color and grayscale video cameras using the autonomous driving platform Annieway [56]. Figure 4.20 shows some of the point clouds in the dataset.

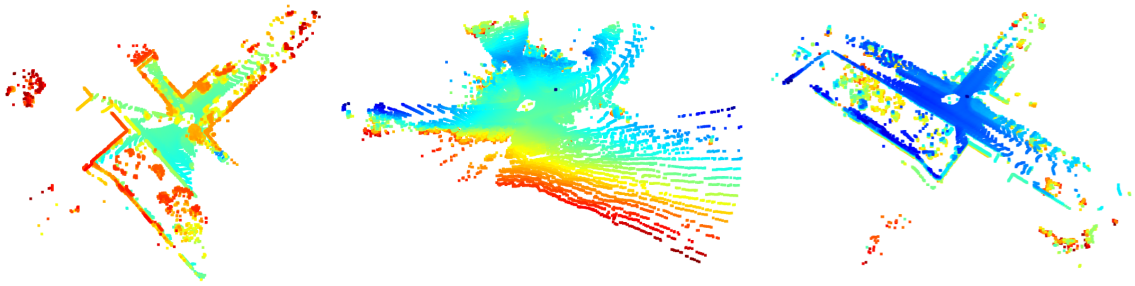


Figure 4.20: Some examples from the KITTI dataset

We used the evaluation metrics *relative rotation error* (RRE), *relative translation error* (RTE), and *registration recall* (RR) in our experiments on KITTI where RRE is the geodesic distance between estimated and ground-truth rotation matrices and RTE is the Euclidean distance between the estimated and ground-truth translation vectors. A registration result is considered successful if RRE is below 5° and RTE is below 2m.

Computing the exact overlap ratio between the source and the target cloud again was not possible for the KITTI dataset. In our experiments, there is at least 10

meters of distance between each point cloud pair, and the range of the sensor used for collecting the point clouds is 120 meters [57]. Therefore, we expected the overlap ratio to be high for any two cloud pairs in our experiments and set our parameter α equal to 0.95.

The comparison of our algorithm's registration results and some state-of-the-art methods can be found in Table 4.4.

Table 4.4: Registration results on KITTI

Method	RTE (cm)	RRE (°)	RR (%)
GeoTransformer [51]	6.8	0.24	99.8
DGR [29]	~ 32	0.37	98.7
FCGF [58]	9.5	0.30	96.6
FMR [59]	~ 66	1.49	90.6
Quantile Assignment	84.7	1.63	60.36

Even though our framework could not outperform the state-of-the-art methods, our results for the KITTI dataset were much better than the 3DMatch dataset. We believe this may be due to the overlap ratio between the source and the target point cloud pairs in KITTI having much less variation than 3DMatch. Thus, we were able to set our parameter α in a more precise manner. The state-of-the-art methods in Table 4.4 are all learning-based registration methods except ours. We recognize that our results can be improved by combining our Quantile Assignment algorithm with a feature learning method.

Chapter 5

Conclusion

We propose a new pairwise point cloud registration framework. Our main contribution is introducing a new algorithm for establishing the correspondence between the source and the target point clouds. The performance of our framework is evaluated using three datasets, and we observed that our registration results were most successful on the synthetic dataset.

A possible reason for achieving better results with the synthetic dataset compared to the other two real-world datasets is that we could accurately estimate the overlap ratio between each point cloud pair in the synthetic dataset. As explained in Section 3.1, we set our parameter α as the overlap ratio, and during our experiments, we observed that the Quantile Assignment algorithm is susceptible to the α parameter. Estimating the overlap ratio is difficult for real-world datasets like 3DMatch and KITTI. Thus, setting the parameters that achieve the best possible results is challenging. Developing a method for overlap ratio estimation given any two input point clouds using a learning model may be a possible future research direction.

One key observation from our experiments is that our framework has better results

with smaller datasets. The reason for this is the reduced need for downsampling, allowing us to preserve crucial features during registration. Large real-world datasets like 3DMatch and KITTI require substantial downsampling to perform the registration reasonably, reducing registration accuracy. Therefore, the computational time is a limitation for our method in practical applications.

Our Quantile Assignment algorithm is computationally costly since the affinity matrix constructed is generally dense for real applications. One possible future research direction is to explore sparser affinity matrix construction techniques to reduce computational complexity while maintaining accuracy.

Our registration performance for real-world datasets needs to improve compared to learning-based point cloud registration frameworks. We recognize that combining it with a feature learning method could enhance our algorithm's performance. Using the Quantile Assignment algorithm, we could extract more informative and discriminative features by incorporating feature learning techniques, leading to more accurate correspondence matching.

Bibliography

[1] R. Pio, "Euler angle transformations," IEEE Transactions on Automatic Control, vol. 11, no. 4, pp. 707–715, 1966.

[2] J. Choi, "Range Sensors: Ultrasonic Sensors, Kinect, and LiDAR," in Humanoid Robotics: A Reference (A. Goswami and P. Vadakkepat, eds.), pp. 2521–2538, Switzerland: Springer Nature, 2018.

[3] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," 2021.

[4] S. Chrétien, O. E. Karaşan, E. Oğuz, and M. Ç. Pınar, "The quantile matching problem and point cloud registration," in Proceedings of the SIAM Conference on Applied and Computational Discrete Algorithms, ACDA '21, pp. 13–20, 2021.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, pp. 239–256, 1992.

[6] F. De Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," Sensors, vol. 17, no. 2, 2017.

[7] X.-F. Han, S.-J. Sun, X.-Y. Song, and G.-Q. Xiao, "3D point cloud descriptors in hand-crafted and deep learning age: State-of-the-art," 2020.

[8] L. Zhao, Z. Xiang, M. Chen, X. Ma, Y. Zhou, S. Zhang, C. Hu, and K. Hu, "Establishment and extension of a fast descriptor for point cloud registration," Remote Sensing, vol. 14, no. 17, 2022.

[9] H. Quan, J. Dong, and X. Qian, "Med-3d: 3d reconstruction of medical images based on structure-from-motion via transfer learning," in Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, BIBM '21, pp. 1051–1056, 2021.

[10] Z. Ma and S. Liu, "A review of 3D reconstruction techniques in civil engineering and their applications," Advanced Engineering Informatics, vol. 37, pp. 163–174, 2018.

[11] Z. Kang, J. Yang, Z. Yang, and S. Cheng, "A review of techniques for 3D reconstruction of indoor environments," ISPRS International Journal of Geo-Information, vol. 9, no. 5, 2020.

[12] J. Mao, S. Shi, X. Wang, and H. Li, "3D object detection for autonomous driving: A review and new outlooks," 2022.

[13] A. Arles, P. Clerc, G. Sarah, F. Téreygeol, G. Bonnamour, J. Heckes, and A. Klein, "3D reconstruction and modeling of subterranean landscapes in collaborative mining archeology projects: techniques, applications and experiences," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 5, p. W2, 2013.

[14] A. Vokhmintsev, M. Timchenko, and K. Yakovlev, "Simultaneous localization and mapping in unknown environment using dynamic matching of images and registration of point clouds," in Proceedings of the 2nd International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM '16, pp. 1–6, 2016.

[15] N. Brightman, L. Fan, and Y. Zhao, "Point cloud registration: a mini-review of current state, challenging issues and future directions," AIMS Geosciences, vol. 9, no. 1, pp. 68–85, 2023.

[16] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," Image and Vision Computing, vol. 10, no. 3, pp. 145–155, 1992. Range Image Understanding.

[17] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, 1992.

[18] Y. Sahillioğlu and L. Kavan, "Scale-adaptive ICP," Graphical Models, vol. 116, p. 101113, 2021.

[19] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing, SGP '13, (Goslar, DEU), p. 113–123, Eurographics Association, 2013.

[20] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2021.

[21] J. Stechschulte, N. Ahmed, and C. Heckman, "Robust low-overlap 3-D point cloud registration for outlier rejection," in Proceedings of the International Conference on Robotics and Automation, ICRA '19, pp. 7143–7149, 2019.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of ACM, vol. 24, p. 381–395, jun 1981.

[23] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3D point cloud registration," Information Sciences, vol. 346-347, pp. 163–179, 2016.

[24] A. Zeng, S. Song, M. Nießner, M. Fisher, and J. Xiao, "3DMatch: Learning the matching of local 3D geometry in range scans," CoRR, vol. abs/1603.08182, 2016.

[25] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," 2018.

[26] Z. J. Yew and G. H. Lee, "3DFeat-Net: Weakly supervised local 3D features for point cloud registration," in Computer Vision – ECCV 2018, pp. 630–646, Springer International Publishing, 2018.

[27] J. Zhou, M. J. Wang, W. D. Mao, M. L. Gong, and X. P. Liu, "SiamesePoint-Net: A Siamese point network architecture for learning 3D shape descriptor," Computer Graphics Forum, vol. 39, no. 1, pp. 309–321, 2020.

[28] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3DRegNet: A deep neural network for 3D point registration," 2020.

[29] C. Choy, W. Dong, and V. Koltun, "Deep global registration," 2020.

[30] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, "Learning multiview 3D point cloud registration," 2020.

[31] A. Johnson and M. Hebert, "Surface matching for object recognition in complex three-dimensional scenes," Image and Vision Computing, vol. 16, no. 9, pp. 635–651, 1998.

[32] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in Computer Vision – ECCV 2010 (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 356–369, Springer Berlin Heidelberg, 2010.

[33] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '09, pp. 3212–3217, 2009.

[34] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in Computer Vision – ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 766–782, Springer International Publishing, 2016.

[35] A. Myronenko and X. Song, "Point set registration: Coherent point drift," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2262–2275, 2010.

[36] P. Yue, Y. Bisheng, L. Fuxun, and D. Zhen, "Iterative global similarity points: A robust coarse-to-fine integration solution for pairwise 3D point cloud registration," in Proceedings of the International Conference on 3D Vision, 3DV '08, 2018.

[37] M. Akgül, "A genuinely polynomial primal simplex algorithm for the assignment problem," Discrete Applied Mathematics, vol. 45, no. 2, pp. 93–115, 1993.

[38] R. Burkard, M. Dell'Amico, and S. Martello, Assignment Problems. SIAM - Society of Industrial and Applied Mathematics, 1 ed., 2009. 382 Seiten.

[39] G. Dantzig, Linear Programming and Extensions. Princeton: Princeton University Press, 1963.

[40] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," Naval Research Logistics Quarterly, vol. 2, pp. 83–97, March 1955.

[41] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," SIAM Journal on Computing, vol. 2, no. 4, pp. 225–231, 1973.

[42] S. L. Tanimoto, A. Itai, and M. Rodeh, "Some matching problems for bipartite graphs," J. ACM, vol. 25, p. 517–525, oct 1978.

[43] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," Canadian Journal of Mathematics, vol. 8, p. 399–404, 1956.

[44] J. Gorski and S. Ruzika, "On -max-optimization," Operations Research Letters, vol. 37, pp. 23–26, 01 2009.

[45] "The Stanford 3D Scanning Repository." http://graphics.stanford.edu/data/3Dscanrep/.

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

[47] X. Zhang, H. Li, and Z. Cheng, "Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures," in Proceedings of ASIAGRAPH, 2008.

[48] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[49] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, (New York, NY, USA), p. 303–312, Association for Computing Machinery, 1996.

[50] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '15, pp. 5556–5565, 2015.

[51] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, CVPR '22, pp. 11133–11142, 2022.

[52] X. Huang, W. Qu, Y. Zuo, Y. Fang, and X. Zhao, "IMFNet: Interpretable multimodal fusion for point cloud registration," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 12323–12330, 2022.

[53] S. Horache, J. Deschaud, and F. Goulette, "3D point cloud registration with multi-scale architecture and unsupervised transfer learning," in Proceedings of the International Conference on 3D Vision, 3DV '21, (Los Alamitos, CA, USA), pp. 1351–1361, IEEE Computer Society, dec 2021.

[54] F. Poiesi and D. Boscaini, "Learning general and distinctive 3D local deep descriptors for point cloud registration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3979–3985, 2023.

[55] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "SpinNet: Learning a general surface descriptor for 3D point cloud registration," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR '21, pp. 11753–11762, 2021.

[56] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361, 2012.

[57] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.

[58] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in Proceedings of the IEEE/CVF International Conference on Computer Vision, CVPR '21, October 2019.

[59] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," 2020.

Appendix A

Comparison With Standard Assignment

To show the contribution of the quantile assignment approach in point cloud registration, the standard maximum cost assignment is integrated into our framework for correspondence search. Registration experiments are conducted on the synthetic dataset using both quantile assignment and standard assignment, and recall values are compared for a series of different downsampling sizes.

A.1 Accuracy Comparison

Recall plots for each model and noise level are presented in Figures A.1, A.2, A.3, A.4, A.5. Some example visuals of registration results can be seen in Figure A.6.

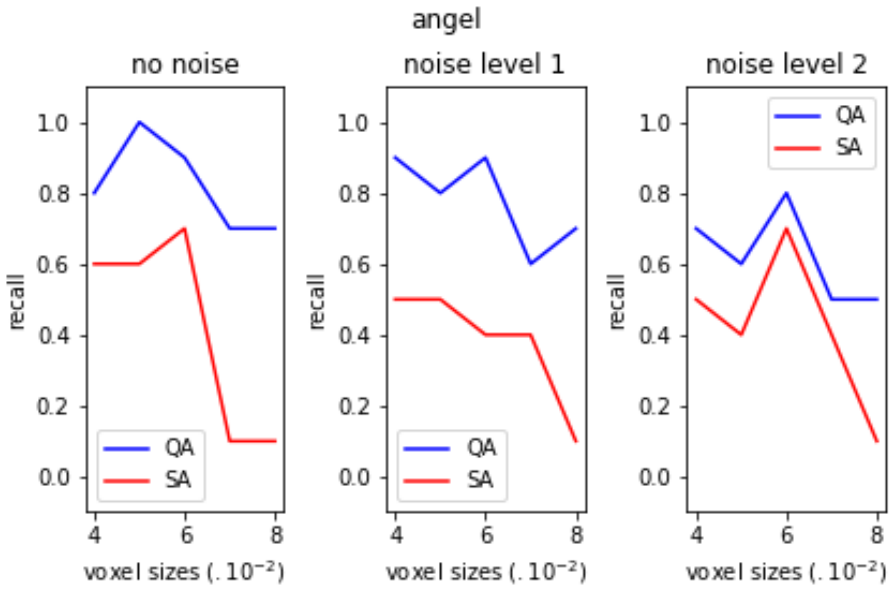


Figure A.1: Registration results of Quantile Assignment (QA) and standard assignment (SA) on Angel

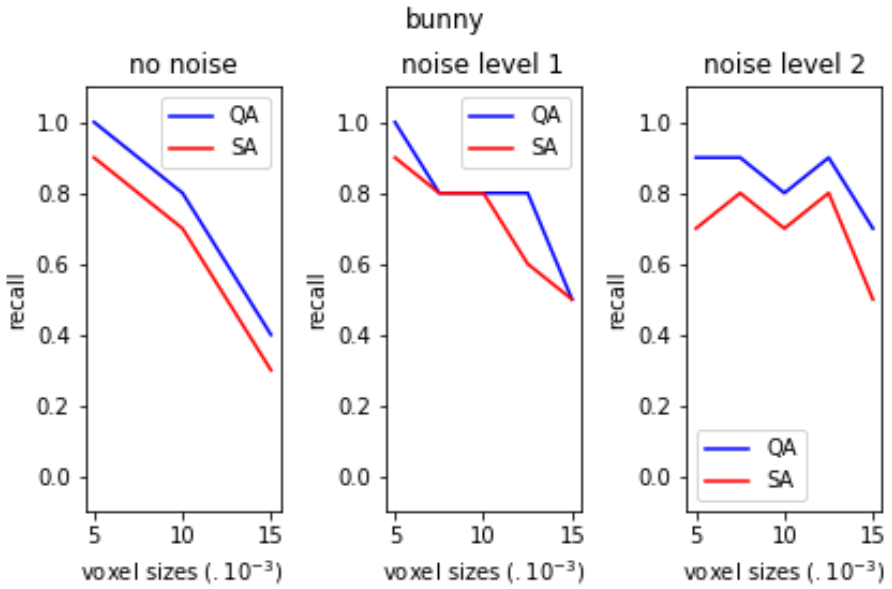


Figure A.3: Registration results of Quantile Assignment (QA) and standard assignment (SA) on Bunny

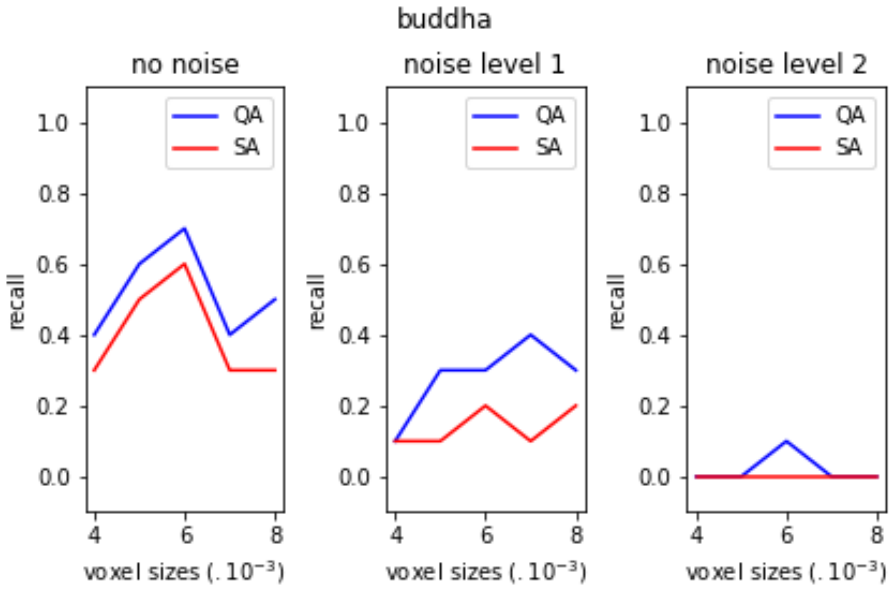


Figure A.2: Registration results of Quantile Assignment (QA) and standard assignment (SA) on Buddha

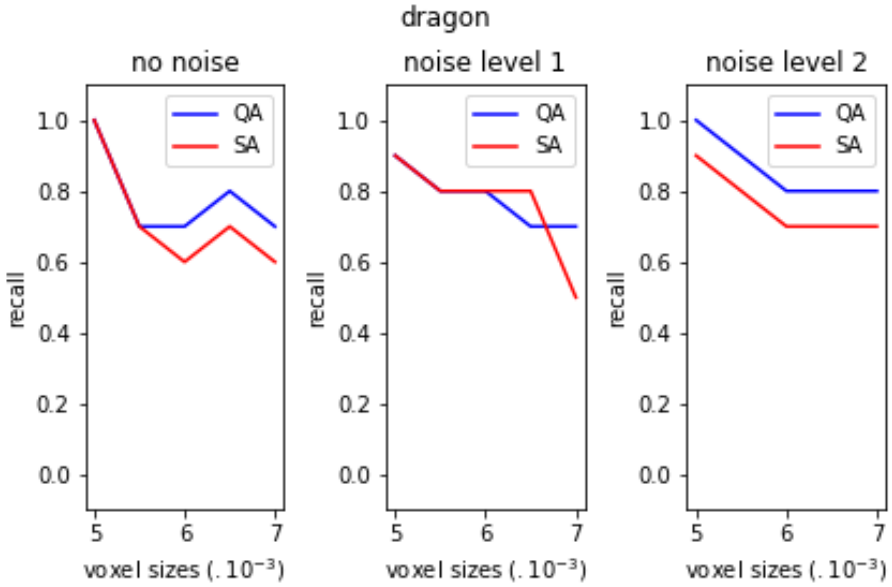


Figure A.4: Registration results of Quantile Assignment (QA) and standard assignment (SA) on Dragon

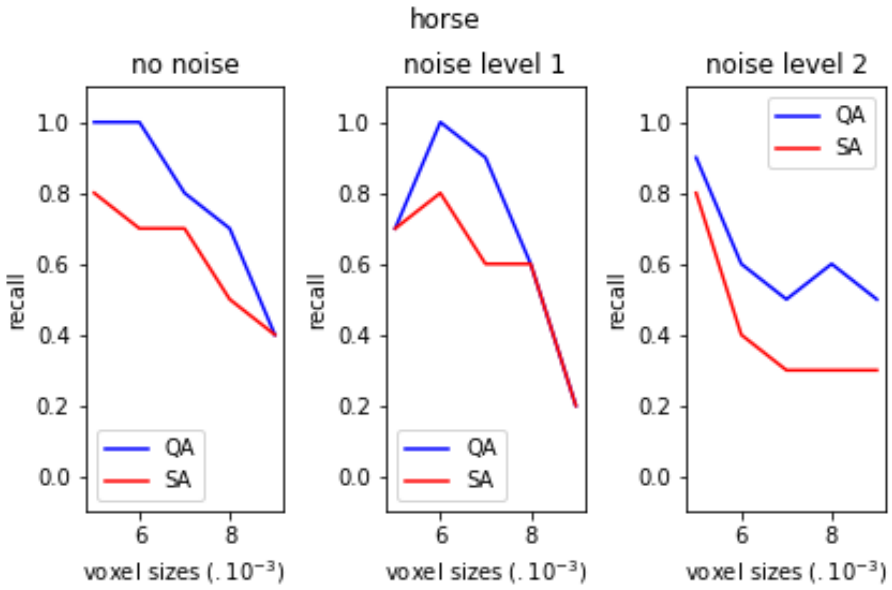
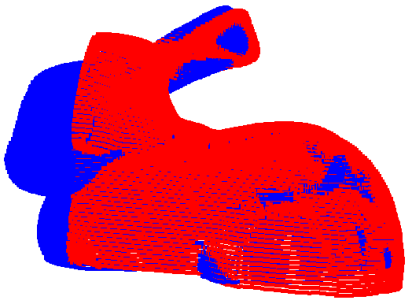


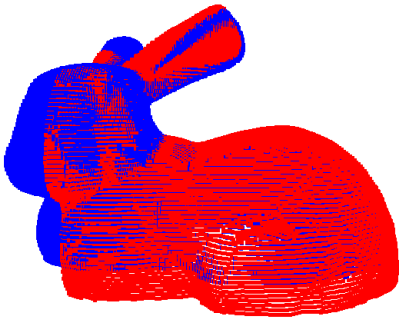
Figure A.5: Registration results of Quantile Assignment (QA) and standard assignment (SA) on Horse

(a) Standard assignment on Angel

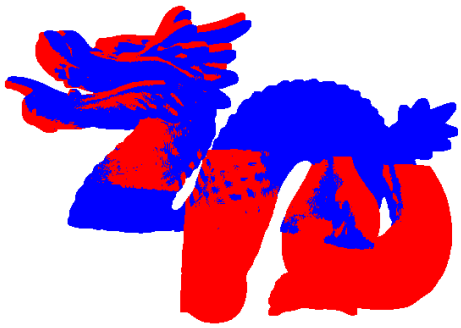
(b) Quantile assignment on Angel



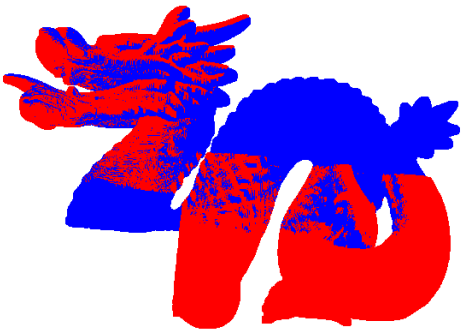
(c) Standard assignment on Bunny



(d) Quantile assignment on Bunny



(e) Standard assignment on Dragon



(f) Quantile assignment on Dragon

Figure A.6: Comparison of registration results of Quantile Assignment (QA) and standard assignment (SA)

It can be observed that significant improvement in registration accuracy can be achieved by using the quantile assignment approach instead of the standard assignment for correspondence estimation.

A.2 Time Comparison

Average computation times in terms of seconds of our framework, our framework with standard assignment instead of quantile assignment, and the Fast Global Registration framework on the Bunny model for different downsampling sizes are compared in Table A.1.

Table A.1: Average computation times of Quantile Assignment (QA), standard assignment (SA) and Fast Global Registration (FGR) on Bunny

		Average Computation Time (seconds)		
Voxel size ($.10^{-3}$)	Average number of points	QA	SA	FGR
5	1778.8	19.9	10.9	5.6
7.5	830.0	12.9	7.9	4.9
10	472.2	10.6	6.6	4.5
12.5	312.4	8.8	5.8	4.3
15	216.6	7.3	5.4	4.1

It can be observed that our algorithm runs slower on the synthetic dataset compared to standard assignment and Fast Global Registration, however, achieves registration results with higher accuracy.