



Title	Accurate Point Cloud Alignment for Pose Estimation and Recognition
Author(s)	李, 斐然
Citation	大阪大学, 2023, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/91994
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Accurate Point Cloud Alignment for Pose Estimation and Recognition

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2023

Feiran LI

List of Publications

Journal

1. Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita: “Shuffled Linear Regression with Outliers in Both Covariates and Responses,” International Journal of Computer Vision (IJCV).

Conference

1. Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita: “A Closer Look at Rotation-invariant Deep Point Cloud Analysis,” In Proceedings of International Conference on Computer Vision (ICCV), 2021.
2. Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita: “Generalized Shuffled Linear Regression,” In Proceedings of International Conference on Computer Vision (ICCV), 2021.
3. Feiran Li, Kent Fujiwara, and Yasuyuki Matsushita: “Toward a Unified Framework for Point Set Registration,” In Proceedings of International Conference on Robotics and Automation (ICRA), 2021.
4. Fan Yang, Feiran Li, Yang Wu, Sakriani Sakti, and Satoshi Nakamura: “Using Panoramic Videos for Multi-Person Localization and Tracking In A 3D Panoramic Coordinate,” In Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.
5. Feiran Li, Gustavo Alfonso Garcia Ricardez, Jun Takamatsu, and Tsukasa Ogasawara: “Adaptive Bingham Distribution Based Filter for SE (3) Estimation,” In Proceedings of International Conference on Robotics and Automation (ICRA), 2019.
6. Feiran Li, Gustavo Alfonso Garcia Ricardez, Jun Takamatsu, and Tsukasa Ogasawara: “Multi-view Inpainting for RGB-D Sequence,” In Proceedings of International Conference on 3D Vision (3DV), 2018.
7. Feiran Li, Ding Ming, Jun Takamatsu, and Tsukasa Ogasawara: “Static 3D Map Reconstruction based on Image Semantic Segmentation,” In Proceedings of International Conference on Ubiquitous Robots (UR), 2018.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Feiran LI
December 2022

Acknowledgements

My Ph.D. journey starts right before the breakout of the COVID-19 pandemic, and it has been a tough time since then. Upon the end of this journey, I would like to express my deepest appreciation to those who have spent their greatest efforts on supporting me.

First and foremost, I would like to express my greatest gratitude to my supervisor, Prof. Yasuyuki Matsushita, for his continuous support and encouragement, invaluable advice, and thoughtful comments throughout my Ph.D. life. It is always his insight and experience that pave the way for me to discover ideas, schedule research plans, writing academic papers, and making presentations. I am truly proud and honored to be a student of Prof. Matsushita for the rest of my life.

I am deeply grateful to Prof. Fumio Okura and Dr. Kent Fujiwara, who are always enlightening and supportive during my Ph.D. training. Their rich writing skills dramatically help me to polish paper.

I would also like to express my thanks to the members of the Computer Vision Laboratory at Osaka University, Prof. Hiroaki Santo, Dr. Heng Guo, Dr. Xu Cao, Dr. Kenji Enomoto, Mr. Kensuke Uchida, Mr. Xinpeng Liu, Mr. Tatsuya Ishibashi, Mr. Daichi Iwata, Mr. Takao Mizuno, Mr. Hiroki Nakamura, Ms. Tomoka Takemura, Mr. Haruya Sakashita, Mr. Kota Nakamura, Mr. Itsuki Onishi, Mr. Yuki Konishiike, Ms. Lilika Makabe, Mr. Ryo Nakano, Mr. Kazuma Minami, Mr. Takashi Yamauchi, Mr. Yudai Matsuoka, Mr. Reo Izubuchi, Mr. Naohiro Shimada, Mr. Yuuki Takayama, Mr. Toshiaki Tanaka, Mr. Arashi Fukui, Ms. Chan Weng Ian, Mr. Zhuoyu Yang, Mr. Genma Kosaka, Mr. Naoki Asada, Ms. Mai Ido, Mr. Shota Ueno, Mr. Mutsuki Tomigashi, Mr. Takuto Narumoto, Mr. Kaito Fukui, Ms. Jie Tang, Mr. Kohei Ashida, Mr. Shota Ishii, Mr. Shota Ueno, Mr. Akira Uchida, Mr. Yosuke Kii, Ms. Mone Tanaka, Mr. Ryohei Miyakawa, Mr. Junpei Watanabe, and Mr. Koki Fukai. It was a great pleasure to work together with so many warm-hearted people. I would also like to express my thank to the secretary of our lab, Ms. Mihoko Kaneda, for her support in my daily life at Osaka University.

Last, all my gratitude goes to my parents and Ms. Yanjiao Ao for their patience and unconditional love.

Abstract

Point cloud alignment plays an important role in computation vision and related fields such as robotics and computer graphics. Given a source point cloud and a target one, point cloud alignment aims at recovering the rigid spatial transformation between them to make their poses as similar as possible. After the alignment procedure, the estimated transformation can straightforwardly help to solve the fundamental pose estimation problem encountered in various applications such as autonomous driving and 3D reconstruction. Moreover, the aligned point clouds appear to be an ideal data format for varying high-level recognition tasks such as object classification and detection.

Given the importance of point cloud alignment, various solution methods have been developed in the past decades. Despite such decade-long studies, we realize that there still exist some fundamental problems that are shared by distinct existing algorithms, preventing the alignment accuracy from being further improved. Consequently, the advances of the down-streaming pose estimation and recognition tasks are also hindered. Motivated by this fact, in this thesis, we present detailed analyses and solution methods to mitigate such shared problems, hoping to push the pose estimation and recognition accuracy one step further.

In Chapter 2, we tackle the problem that the relations between the existing geometric and probabilistic alignment algorithms developed for pose estimation are not well studied. These two types of approaches are dominant in solving the point cloud alignment problem but have been developed independently so far, making their respective merits not sharable. As a solution, we present a unified framework by leveraging the knowledge from information geometry to explain them. Our unified explanation makes the respective merits of these two classes of algorithms mutually commutable, leading to more accurate alignment and pose estimation results.

In Chapter 3, we simultaneously solve two problems that existing alignment algorithms may present infeasible point-wise correspondences and suffer from outliers, diminishing the pose estimation accuracy. While the recently raised research topic of shuffled linear regression can perfectly solve the infeasible estimation problem, it still cannot effectively handle outliers in point clouds. As a solution, we present a generalized version of the shuffled linear regression problem. Our proposal can not only theoretically guarantee that

the estimations are always feasible, but also introduces effective robust estimation tools to reject outliers. Various applications show the effectiveness of our proposal.

In Chapter 4, we analyze in detail the ambiguity problem of principal component analysis when applied to intra-category alignment, which is a popular pre-processing step in deep learning-based point cloud recognition tasks. We show that its associated number of ambiguities is not sufficiently recognized in existing works. Moreover, we empirically find that sufficient identification of the ambiguities can lead to conspicuous performance improvements on varying recognition tasks, and such improvements are particularly visible on challenging real-world point clouds.

Table of contents

List of figures	xiii
List of tables	xvii
1 Introduction	1
1.1 Motivation	2
1.2 Thesis organization	5
1.3 Summary of contributions	6
2 Toward a unified framework for point cloud alignment	9
2.1 Introduction	9
2.2 Related works	11
2.3 A unified framework of geometric and probabilistic approaches	13
2.3.1 Unification of generic geometric and probabilistic approaches	13
2.3.2 Unification of point-to-plane and plane-to-plane methods	19
2.4 Bandwidth estimation using an intrinsically contained hint	20
2.5 How unification helps mutual benefit: A case study	21
2.5.1 Problem formulation	21
2.5.2 Experiments	22
2.5.3 Test on synthetic data	24
2.5.4 Test on real-world data	24
2.5.5 Test with varying levels of noise & ratios of outlier	25
2.5.6 Elapsed time of different methods	25
2.6 Performance of the proposed strategy for bandwidth estimation	27
2.6.1 Comparisons of alignment accuracy between deterministic annealing and our proposal	27
2.6.2 Stability of our proposal w.r.t. different trade-off weight λ	28
2.7 Discussion and conclusion	29

3	Point cloud alignment with robust permutational constraint	31
3.1	Introduction	32
3.2	Related works	33
3.2.1	Shuffled linear regression	33
3.2.2	Similar problems in applications	34
3.2.3	Robust estimation	35
3.2.4	Other permutation-constrained problems	36
3.3	Formulation of GSLR	37
3.3.1	Objective function	37
3.3.2	Effective recovery condition	38
3.3.3	NP-hardness	38
3.3.4	Proofs of corollaries, propositions, and lemmas in Chapters 3.3.2 and 3.3.3	40
3.4	Optimization of GSLR	42
3.4.1	A simple optimization algorithm	42
3.4.2	Convergence analyses	45
3.4.3	Proofs of corollaries, propositions, and lemmas in Chapter 3.4.2 . .	47
3.5	Experiments	53
3.5.1	Comparing SLR to GSLR	53
3.5.2	GSLR in point cloud alignment	56
3.5.3	GSLR in isometric shape matching	63
3.5.4	Other experimental studies of GSLR	65
3.6	Discussion and conclusion	69
3.6.1	Limitations	70
3.6.2	Failure cases	70
3.6.3	Future work	70
4	Ambiguity analysis for principal component analysis-based intra-category align- ment	73
4.1	Introduction	73
4.2	Related works	75
4.2.1	Deep learning on point clouds	75
4.2.2	Rotation-robust point cloud analysis	75
4.2.3	Rotation-invariant point cloud analysis	76
4.3	Ambiguities of the PCA-based canonical pose	76
4.3.1	Sign Ambiguity	77
4.3.2	Order ambiguity	78

4.3.3	Summary of ambiguities	79
4.4	Pose selector: Learning to disambiguate the canonical poses	79
4.5	Performance study using the correctly identified ambiguities	81
4.5.1	Implementation details	81
4.5.2	Object classification	81
4.5.3	Object part segmentation	83
4.5.4	Classification on real-world dataset	87
4.5.5	Ablation studies	88
4.6	Further explorations regarding the PCA-based canonical poses	90
4.6.1	How well can the canonical poses perform?	90
4.6.2	Can disambiguation methods help?	92
4.6.3	Do inaccurate ambiguities have any effect?	92
4.6.4	When does PCA perform the best?	93
4.7	Discussion and conclusions	94
5	Discussion and conclusion	97
5.1	Summary	97
5.2	Future works	99
	References	101

List of figures

1.1	An introduction of the concept and applications of point cloud alignment, where we desire to recover the rigid spatial transformation $\mathbf{T}$ between the source and target point clouds. After alignment, the estimated $\mathbf{T}$ and the two aligned point clouds can benefit the pose estimation and recognition tasks, respectively.	2
1.2	Different tasks require aligning varying kinds of point clouds. Top left: Aligning 3D Euclidean point clouds for 3D reconstruction. Bottom left: Aligning 2D Euclidean point clouds for 2D pose estimation. Top right: Aligning image pixels for homography estimation. Bottom right: Aligning mesh features for shape matching.	3
1.3	Definition of intra-category alignment. Different from point cloud alignment for pose estimation purposes. Intra-category alignment aims at aligning point clouds of the same category to conceptually the same poses. For example, we desire all the chairs to stand on the same plane and face toward the same direction.	4
2.1	A conceptual illustration of how geometric and probabilistic methods work. While geometric approaches operate on the Euclidean space with euclidean distance as the metric, probabilistic algorithms work on the probabilistic manifold with the associated Fisher information metric.	17
2.2	Examples of the synthetic point clouds we used.	23
2.3	An example of the ETH Hauptgebaude dataset. Some points are cropped for better visualization.	25
2.4	Alignment accuracy w.r.t. varying levels of noises.	26
2.5	Alignment accuracy w.r.t. varying ratios of outliers.	26
2.6	Comparison of the alignment accuracy w.r.t. different annealing parameters. The orange line is achieved by fixing the tradeoff parameters to $\lambda_1 = \lambda_2 = 50$	27
2.7	Alignment accuracy w.r.t. different tradeoff parameters λ_1 and λ_2	28

3.1	Different strategies for correspondences estimation. GT is short for ground truth. The NN-based method may violate the one-to-one policy by either assigning one point to multiple queries or leaving points unmatched, and SLR cannot reject outliers.	35
3.2	Graphs constructed from the GSLR data. Vertices are labeled by the elements $\mathbf{A}_i$ and $\mathbf{b}_j$, and edges are labeled by the sum of their respective adjacent vertices.	39
3.3	Examples of the image alignment experiment. On the source and target image pairs, outliers detected by GSLR are marked as red and inliers as green. NN denotes the ICP-style algorithm where the correspondences are estimated by nearest-neighbor search, and SLR (ours) is Algorithm 4 with the number of inliers k fixed to $\min(m, n)$. The descriptor matching-based method [18] are omitted as its results are visually the same to the target images.	54
3.4	Example alignment results on the synthetic data. Left: Aligning outlier-contaminated point clouds using our GSLR. The identified outliers are marked as green. Right: Aligning the partially overlapped data. TEASER++, DO, and PointNetLK show either similar or worse visual results than the illustrated ones and thus omitted.	58
3.5	Transformational errors w.r.t. different scales of isotropic Gaussian noise (left) and different ratios of outliers (right).	58
3.6	Transformational errors w.r.t. different initializations. FGR and TEASER++ are omitted as they are global alignment methods.	59
3.7	Some example alignment results achieved by our GSLR on the ETH Hauptgebaude (left) and Office (right) datasets.	63
3.8	Some example alignment results achieved by our GSLR on the 2D laser scans. Left: EUR Pallet dataset. Right: IRL dataset.	64
3.9	An example of the geodesic error maps on the FAUST dataset. ZoomOut-50 is omitted for its inferior performance.	66
3.10	Matching results on the FAUST dataset. Top row: Examples achieved by GSLR. Second and third rows: Intra-class and inter-class geodesic errors and coverage rates.	66
3.11	Matching results on the TOSCA dataset. Top: Examples achieved by GSLR. Bottom: Geodesic errors and coverages.	67
3.12	An example of coverages on the TOSCA dataset. Vertices violating the one-to-one corresponding policy are marked as orange.	67
3.13	Point cloud alignment accuracies w.r.t. varying thresholds of the Huber-skip estimator.	68

3.14	Statistical efficiencies of the trimming operator and the Huber-skip estimator under distinct setups. Left: Varying thresholds in Eq. (3.18). Right: Varying ratios of the trimming operator.	69
3.15	Failure cases of GSLR. Top: Failure due to deficient initialization. Bottom: Failure due to invalid formulation of the robust estimation problem. GT is short for ground truth.	71
4.1	A conceptual explanation of improper rotation, which consists of: 1) rotation along an axis; 2) reflection along the plane perpendicular to this axis. Pure planar reflection can be considered as a special case of improper rotation where the rotation stage vanishes.	78
4.2	An example illustration of the 24 ambiguities of the PCA-based canonical poses. The top row and the left column consist of the 6 order ambiguities and 4 sign ambiguities, respectively.	79
4.3	Structure of the pose selector, which predicts the weights of the 24-dimensional inputs. We implement the module with 256-512 1D convolutional filters followed by max-pooling, 24-dimensional linear mapping, and softmax. . .	80
4.4	Visualization of the ground truth segmentation, results using 1 of the 24 possible poses, and our results from the latter 8 classes in ShapeNet part segmentation dataset. The poses are drawn from the 24 possible PCA-based canonical poses.	84
4.5	Some example point clouds from the <i>OBJ_BG</i> dataset of ScanObjectNN. The data therein is challenging for containing noise, holes, deformations, and incompletely removed backgrounds.	87
4.6	Disambiguation methods may assign objects within the same category to different poses.	91
4.7	Conceptual illustrations of the PCA-based alignments on non-symmetric objects. Planes characterize the three principal axes detected by PCA. . . .	93
4.8	A point cloud can be denoted as $\mathbf{x} + d\mathbf{n}$ and $\mathbf{x} - d\mathbf{n}$ given known symmetry. . .	94

List of tables

2.1	Summary of geometric and probabilistic alignment algorithms	11
2.2	Errors on the synthetic data (both rotational and transformational errors are in $1e-2$ scale).	24
2.3	Errors on the laser scan data.	25
2.4	Average elapsed time records on the synthetic datasets (second).	27
3.1	Notations in Chapter 3.4.2.	46
3.2	Mean and median absolute reprojection errors of image alignment (in pixel unit). The method [18], which employs known correspondences as a prior, is used for reference.	55
3.3	Alignment results on the synthetic data.	57
3.4	Alignment results on the ETH Hauptgebaude and Office datasets. Minimum errors are marked as bold when methods with denser inputs are included in comparison, and underlined when such methods are excluded.	61
3.5	Alignment results on the sparse IRL and EUR Pallet datasets.	62
3.6	Computational cost of Algorithm 4 w.r.t. varying point cloud densities. . . .	70
4.1	Determinants and geometric meanings of the eigenvector matrix w.r.t. different assignment of signs.	78
4.2	Classification accuracy (%) on the ModelNet40 dataset. Point clouds for all the methods are with density 1024. Within the inputs column, xyz denotes the Cartesian coordinates of the point clouds and features stand for handcrafted geometric features. The last column records the differences between $z/\text{SO}(3)$ and $\text{SO}(3)/\text{SO}(3)$	82
4.3	Mean IoU (%) over all instances on ShapeNet. The inputs of each algorithm are the same as mentioned in Table 4.2. All point clouds are with density 2048 except for RI-GCN [70] and Triangle-Net [152], whose inputs are with density 1024.	85

4.4	Per-class mIoU (%) of the SO(3)/SO(3) setup. RI-GCN [70], Triangle-Net [152] and SRI-Net [134] are omitted since the per-class mIoUs are not reported in their respective paper. Also, since the results originally reported in Li <i>et al.</i> [80] and LGR-Net [165] in their papers are the mean values of the mIoU over all categories, we here recalculate their mIoU over all the instances using their reported per-category mIoU and summarize these converted results for a straightforward comparison.	86
4.5	Classification accuracy (%) on the real-world ScanObjectNN dataset. z^* denotes the original pre-aligned dataset without any rotations. All point clouds are with density 1024. Our approach can noticeably outperform LGR-Net, which is the best peer method on synthetic datasets.	87
4.6	Pose selector can increase the accuracy of different tasks compared to the vanilla network, demonstrating its effectiveness in disambiguating the canonical poses.	88
4.7	Classification accuracy (%) w.r.t. distinct networks. Strategy III stands for the disambiguation method described in Chapter 4.6.2, with which all the ambiguities are eliminated.	89
4.8	Classification accuracy on ModelNet40 and mIoU (%) on ShapeNet w.r.t. different input pose on the classification and part segmentation tasks. Results from left to right are obtained with: the original datasets; manually selected canonical pose via Eq. (4.5); and all the 24 ambiguities.	90
4.9	Classification accuracy (%) w.r.t. distinct disambiguation strategies on ModelNet40. The number of ambiguities from left to right are 4, 6, 0, 24, respectively.	92

Chapter 1

Introduction

Point cloud refers to a set of discrete points in Euclidean space that is unordered and unstructured. Owing to such a property, point cloud naturally appears to be a flexible tool to digitalize the 3D world surrounding us. Moreover, owing to the fact that many off-the-shelf 3D perception sensors and devices, such as stereo cameras, structured light systems, and various time-of-flight sensors, will directly generate point cloud when operating, point cloud is broadly employed by applications coming from distinct fields. For example, in autonomous driving [23, 89] and robotics [77, 69], vehicles and robots are commonly equipped with depth-perception sensors to help to locate their positions and detect occlusions in the surroundings. Also, in the 3D reconstruction task in computer vision [168, 71], point cloud is employed as one of the dominant representations to describe and visualize the reconstructed shapes. Furthermore, even in other fields such as medicine [108, 97], devices such as CT and MRI can generate point clouds to aid medical diagnosis and surgery.

Given the power and broad applications of point cloud, there however exists a fundamental problem when using it. Specifically, point clouds are usually obtained in different poses since the sensors employed for data acquisition are commonly placed at different positions with different orientations. Therefore, to facilitate the down-streaming tasks, it is required to estimate the relative poses between the point clouds and align them accordingly.

One technique to solve the aforementioned different-pose problem is point cloud alignment [61]. Formally speaking, as shown in Fig. 1.1, given two point clouds where one is the source and the other is the target, point cloud alignment aims at finding a rigid spatial transformation, with which we desire to rotate and translate the source point cloud to make its pose as similar as possible w.r.t. the target one. After this alignment procedure, the estimated spatial transformation, representing the relative poses of the two point clouds, can be directly used to compute the absolute poses of the two point clouds in the same coordinate system, and hence solving the pose estimation problem encountered in various

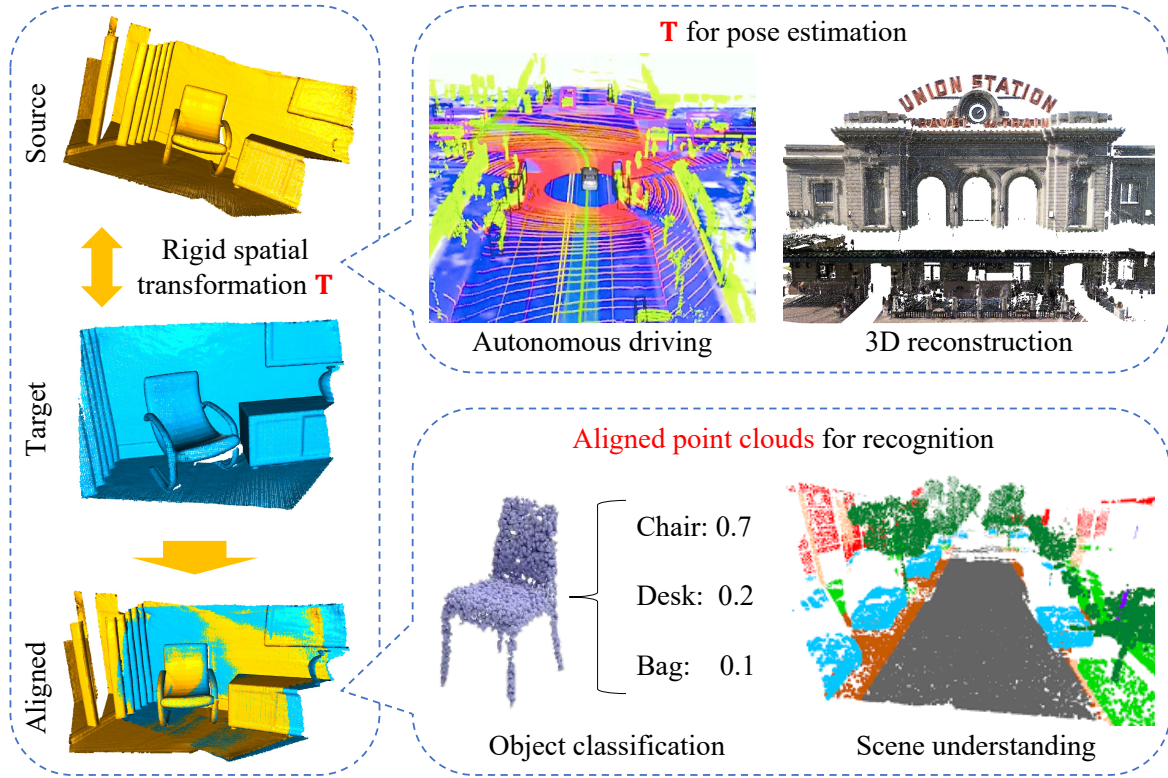


Fig. 1.1 An introduction of the concept and applications of point cloud alignment, where we desire to recover the rigid spatial transformation T between the source and target point clouds. After alignment, the estimated T and the two aligned point clouds can benefit the pose estimation and recognition tasks, respectively.

applications such as autonomous driving and 3D reconstruction [89, 168]. On the other hand, the two aligned point clouds themselves also serve as the ideal data format for many machine learning-based high-level point cloud recognition tasks such as object classification and scene understanding [52].

It is noteworthy that the study range of point cloud is not limited to the 3D Euclidean space. Rather, a point cloud can also be extracted from sources with varying dimensions. For example, as shown in Fig. 1.2, aligning point clouds formulated by pixel coordinates of images leads to the image registration task where the rigid spatial transformation changes to a homography mapping. Also, In shape matching, the two point clouds are formulated by vertex-wise high-dimensional features extracted from meshes, and the rigid spatial transformation also changes to a high-dimensional orthogonal transformation.

1.1 Motivation

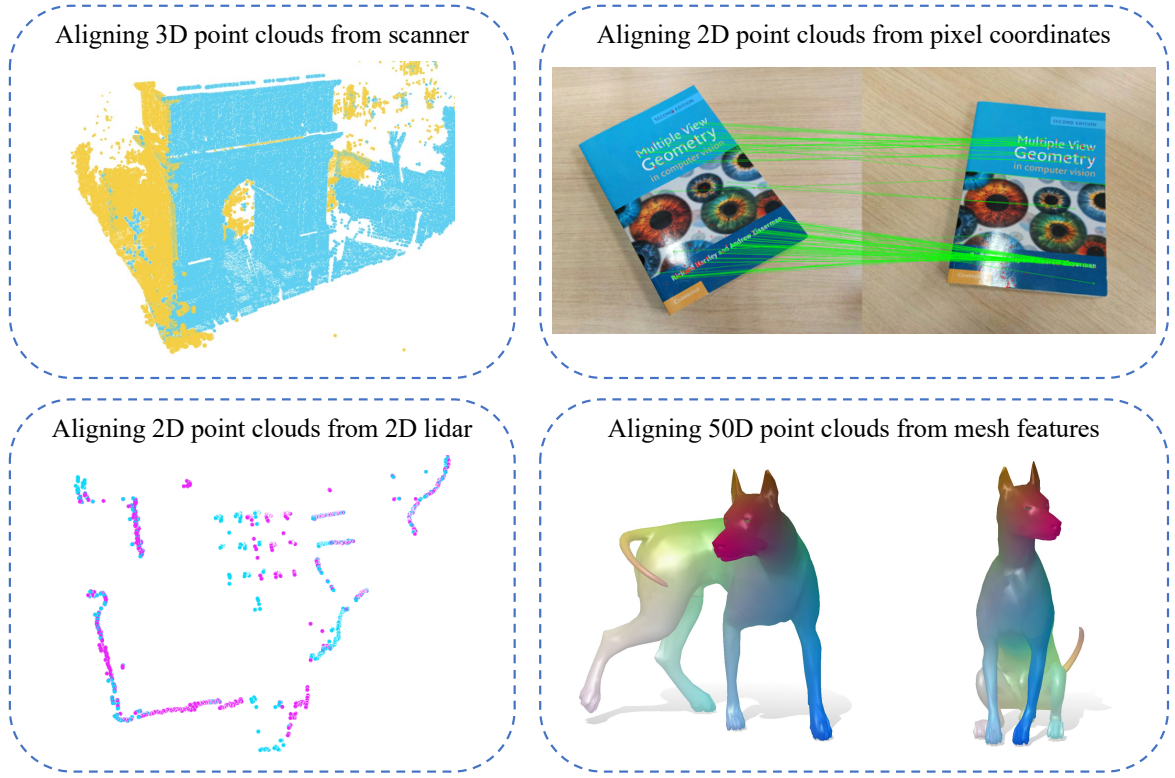


Fig. 1.2 Different tasks require aligning varying kinds of point clouds. Top left: Aligning 3D Euclidean point clouds for 3D reconstruction. Bottom left: Aligning 2D Euclidean point clouds for 2D pose estimation. Top right: Aligning image pixels for homography estimation. Bottom right: Aligning mesh features for shape matching.

Due to its importance, point cloud alignment has already been studied for decades, and many representative solution methods [100, 44, 159, 123, 5] are proposed and widely deployed. Despite such flourishing, we realize that there still exist some problems broadly shared by existing point cloud alignment algorithms, and thus preventing them from achieving more accurate alignment results.

This thesis focuses on increasing the point cloud alignment accuracy by solving some of such shared problems. From many existing problems requiring to solve, we concentrate on three fundamental ones that lie at the bottom level of applications. Briefly, we first study existing alignment algorithms designed for pose estimation and tackle the problem that their relations are unclear. As a solution, we explain a majority of them as repeating the matching-and-pose-estimation process. Then, given the conclusion that many alignment algorithms require to conduct the matching process, we fix a flaw shared by commonly used matching strategies. Finally, we generalize the study range to intra-category alignment, where we present a sufficient analysis of the ambiguity problem encountered therein. Respective details of the three problems are presented in the following.

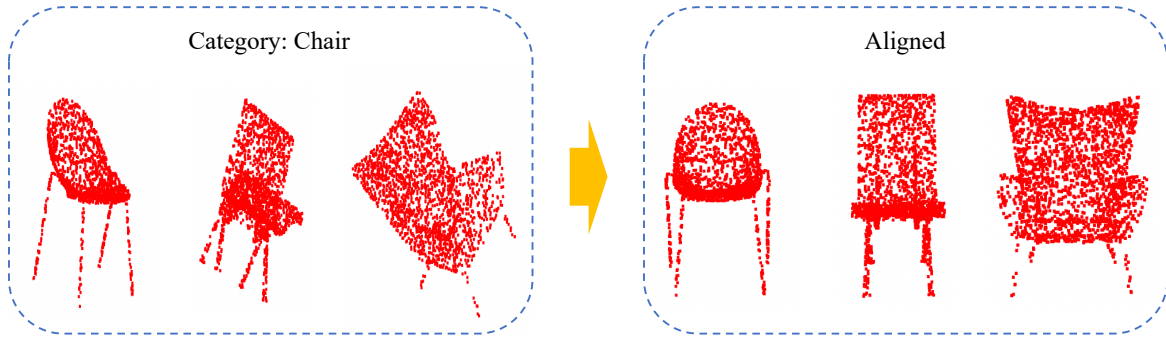


Fig. 1.3 Definition of intra-category alignment. Different from point cloud alignment for pose estimation purposes. Intra-category alignment aims at aligning point clouds of the same category to conceptually the same poses. For example, we desire all the chairs to stand on the same plane and face toward the same direction.

Independent development of existing works Despite the fact that numerous point cloud alignment algorithms for pose estimation have been proposed in the past decades, most of these existing works can be recognized as either a geometric method or a probabilistic one depending on how they parameterize the point clouds. Specifically, while the geometric methods parameterize point clouds as what they appears to be, *i.e.*, a set of points in the Euclidean space represented by their coordinate values, the probabilistic counterparts map each individual point to a component distribution and hence treat the entire point cloud as a mixture distribution. While these two categories of methods respectively possess merits in explicitly constructing point-wise correspondences and enabling the usages of varying distributions to handle noise, they have been developed quite independently so far with vaguely studied relations w.r.t. each other. As a result, we currently cannot develop an algorithm to merge the advantages of both the geometric and the probabilistic methods.

Failure to jointly reject outliers & maintain isometric matching As mentioned above, most geometric alignment methods for pose estimation require establishing the point-wise correspondences between the two input point clouds, and the nearest-neighbor (NN)-based strategy [5] is a popular tool to achieve it. However, such a matching strategy has two main limitations and accordingly diminishes alignment accuracy. Specifically, first, the NN-based strategy lacks effective treatments to identify and reject outliers. Second, per the workflow of NN searching, it may map two distinct points to the same target, and hence violating the isometric mapping property of the rigid spatial transformation. While these two problems can be solved isolatedly by involving robust estimation techniques and permutational constraints, how to solve them simultaneously remains an open problem.

Insufficiently identified number of ambiguities in intra-category alignment Point cloud alignment is commonly conducted in an intra-category manner defined in Fig. 1.3 when it is applied to aid high-level recognition tasks, and the principal component analysis (PCA) algorithm is a popular tool to achieve such alignment. However, we find that the number of ambiguities associated with PCA is not sufficiently counted in existing works. As a result, recognition algorithms such as various neural network-based ones cannot demonstrate their full ability.

1.2 Thesis organization

In this thesis, we aim at analyzing and solving the three commonly-shared problems mentioned above. We organize the thesis to anchor the development timeline of the point cloud alignment techniques. The details of each remainder chapter are described in the following.

Chapter 2: Unified framework for point cloud alignment This chapter presents a unified framework to explain the two dominant yet independently developed classes of existing point cloud alignment algorithms: the geometric methods and the probabilistic ones, making them to be able to mutually benefit from each other. Based on information geometry, which studies statistics by treating probabilistic distributions as points from a special manifold, we derive that the existing geometric and probabilistic alignment methods can be considered as conducting the same operations on different spaces associated with different metrics. Moreover, in order to put the unified framework into practice, we also present a novel bandwidth estimation strategy using intrinsically contained hints to map Euclidean points to probabilistic distributions, which appears to be a good substitution for the broadly used heuristic method of deterministic annealing. Finally, a case study of symmetric registration is conducted to demonstrate that, owing to our unification explanation, we can always combine the advantages of both geometric and probabilistic alignment methods to improve alignment accuracy.

Chapter 3: Aligning with robust permutational constraint This chapter studies a general regression formulation abstracted from the geometric point cloud alignment problem, *i.e.*, shuffled linear regression, aiming to avoid infeasible solutions encountered in existing alignment formulations. Briefly speaking, shuffled linear regression requires estimating a regression variable (*i.e.*, the spatial transformation in point cloud alignment), and the permutational correspondences between the covariates and responses (*i.e.*, the point-wise correspondences in point cloud alignment) simultaneously in a linear regression form. While

the existing solution methods assume the underlying ground-truth correspondences to be an ideal bijection, such an assumption barely holds in point cloud alignment since the data unavoidably contains outliers and partial overlaps. To mitigate such a limitation, we generalize the formulation of shuffled linear regression to a broader range of conditions where only a part of the data should correspond and propose a simple yet effective optimization algorithm. In order to offer a better understanding of our proposal, we also provide detailed theoretical studies w.r.t. the formulation itself and the convergence of the optimization algorithm. Broad experiments validate the effectiveness of our proposal in improving alignment accuracy.

Chapter 4: Intra-category alignment for recognition purpose This chapter studies the ambiguity problem encountered in principal component analysis (PCA) when it is applied for aligning point clouds in an intra-category manner, which is broadly employed as a data pre-processing step in deep learning-based point cloud recognition tasks. We analyze to show that PCA introduces two types of ambiguities: sign ambiguities and order ambiguities, and some existing works have mis-identified the number of the former ambiguities and directly overlooked the latter ones, and hence making the recognition performance substandard. Moreover, in order to effectively handle the ambiguities, we present a simple disambiguation module that can be plugged into most existing neural networks. Experiments show that our sufficient identification of the ambiguities can help to conspicuously improve the accuracy of recognition tasks such as object classification and part segmentation even compared with existing works that developed complicated network structures, and such improvement is particularly significant on real-world point clouds.

Chapter 5: Conclusion and discussion This chapter concludes the thesis by summarizing the theories and algorithms developed in the previous chapters and discussing future works in point cloud alignment for higher accuracy and efficiency.

1.3 Summary of contributions

The main contributions of this thesis can be summarized as follows.

- We present a unified framework by leveraging the concept of information geometry to explain the existing geometric and probabilistic point cloud alignment algorithms, enabling them to be mutually beneficial. We also come up with a novel bandwidth estimation strategy to map Euclidean points to component distributions using the

intrinsic hints provided by the objective function, which appears to be a reasonable substitution of the commonly used heuristic deterministic annealing.

- We generalize the formulation of the shuffled linear regression problem abstracted from point cloud alignment, making it capable of handling outliers. To optimize our formulated objective function, we also come up with a simple yet effective optimization algorithm. Moreover, detailed theoretical analyses, including the effective recovery condition, the NP-hardness, and the convergence property of the optimization algorithm, are presented to help to achieve a deeper understanding.
- We present a detailed study of the ambiguity problem of PCA in intra-category alignment. We show that some ambiguities resulting from PCA have been either mis-used or overlooked by existing works, and the recognition accuracy can be conspicuously improved provided that the ambiguities is sufficiently identified. A simple plug-and-use module for disambiguation is also introduced to facilitate existing neural networks designed for point cloud analysis.

Chapter 2

Toward a unified framework for point cloud alignment

Point cloud alignment plays a critical role in computer vision and related fields. Early methods considered alignment as a purely geometric problem, providing various choices for rejecting outliers, and probabilistic methods were later introduced to handle different types of noise. However, the two classes have developed independently, without sharing much in common. In this chapter, we leverage the concept of information geometry to theoretically unify the two classes by interpreting them as operations in different spaces associated with distinct metrics. Based on the unification, we also develop methods to deal with long-standing problems of uncertainty: anisotropic noise and bandwidth estimation. The resulting method possesses solid theoretical foundations as well as intuitive interpretation. Experiments demonstrate the effectiveness of the proposed method on both synthetic and real-world data under various settings.

2.1 Introduction

Point cloud alignment is popularly encountered in application such as 3D reconstruction, shape retrieval, medical imaging, and robotics. It is formulated as finding the optimal spatial transformation $\mathbf{T} = \{\mathbf{R}, \mathbf{t}\}$, where $\mathbf{R}$ is the rotation and $\mathbf{t}$ is the translation, between two point clouds $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ and $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$. The problem has been approached with different perspectives in the existing methods. The story began from the geometric algorithms, such as Iterative Closest Point (ICP) [10], followed by other counterparts based on statistics [100, 66], graph matching [166], feature correspondence [50], and deep learning [36]. Since a vast majority of alignment tasks take geometric and probabilistic approaches,

in this work, we focus on these two classes. While probabilistic methods excel at coping with various types of noise, geometric ones provide more choices in rejecting large outliers. These two approaches possess their own advantages, yet their independent evolutions have prevented them from benefiting each other.

The difficulty of alignment lies in the unknown correspondences between $\mathbf{b}_j$ and $\mathbf{a}_i$. As pioneering approaches, geometric algorithms such as Iterative Closest Point (ICP) [10] first proposed to tackle this problem by alternating between establishing correspondences and optimizing the spatial transformation $\mathbf{T}$. Other methods based on statistics [100, 66], graph matching [166], feature correspondence [50], and deep learning [36] were later introduced to address various challenges. In this work, we focus on the geometric and probabilistic approaches since they are used by a vast majority of alignment tasks.

While probabilistic methods excel at coping with various types of noise, geometric ones provide more choices in various conditions due to their explicit handling of correspondences. Despite the maturity of these methods, there still exist some long-standing problems. First, the two classes of methods evolved independently. As a result, the relationship between the two classes is unclear, preventing them from sharing techniques for mutual benefit. Second, existing probabilistic alignment methods lack effective strategies to estimate bandwidths of probabilistic distributions, and heavily depend on empirical tuning without any theoretical justifications [47]. Consequently, their performance is hindered from further enhancement.

In this work, we develop a unified framework as well as a novel bandwidth estimation strategy to tackle the aforementioned problems. For unification, we show that the relation between geometric and probabilistic alignment algorithms can be explained from an information geometric perspective, where both geometric and probabilistic approaches are regarded as conducting the same operation but in different spaces. For bandwidth estimation, we show that the objective function developed for alignment intrinsically serves as a guidance for estimating bandwidths; hence we do not need to tune them empirically. We summarize our contributions as follows:

- A theoretical unification that bridges the geometric and probabilistic alignment approaches for mutual benefit.
- A bandwidth estimation strategy with theoretical and practical advantages over existing approaches.

It is noteworthy that there already exist some attempts to bridge the geometric approaches with the probabilistic ones via expectation-maximization (EM) [47, 65]. However, the fact that the underlying metric of the EM algorithm is the Kullback-Leibler divergence (KL-divergence) prevents these works from explaining other divergence-minimization-based

Table 2.1 Summary of geometric and probabilistic alignment algorithms

	Problem Formulation	Optimization	Milestone
Geometric	Orthogonal Procrustes problem w/ unknown assignment	Alternating optimization	ICP [10]
Probabilistic	Distribution-fitting of MM/KDE	Expectation maximization	CPD [100]
Probabilistic	Divergence-minimization between MM/KDE	Gradient descent	GMMReg [66]

* MM: mixture model, KDE: kernel density estimation

methods [58, 146]. As a result, such a unification cannot be generalized to all the cases. On the other hand, our explanation suits all cases well without exception.

2.2 Related works

Geometric and probabilistic methods possess their respective merits in solving the point cloud alignment problem. Specifically, owing to the explicit handling of correspondences, geometric methods are more flexible in tasks such as symmetric alignment and outlier rejection. On the other hand, probabilistic ones are more accurate under conditions such as point-wise varying noises or clustering-based alignment. We summarize their common objectives in Table 2.1, where MM and KDE respectively stand for mixture model (*a.k.a.*, mixture family distribution) and kernel density estimation.

Geometric approaches for alignment By formulating the objective as the orthogonal Procrustes problem [46] with unknown correspondences, ICP [10] leads the development of geometric approaches for point cloud alignment. Some variants of it present alternatives on establishing correspondences. For example, EM-ICP [47] relaxes the binary correspondence rule to soft assignment to deal with inaccurate initialization. Gold *et al.* [45] convert the soft assignment matrix into a doubly stochastic one to allow symmetric alignment. Trimmed-ICP [25] robustifies the standard ICP algorithm by trimming out probable mis-correspondences. There also exist other extensions that study global optimization and efficiency. For instance, Yang *et al.* [159] propose a Branch-and-Bound search strategy to guarantee global optimization. Greenspan and Yurick [48] use the approximated k-d tree for reducing the computational cost of matching. The point-to-plane [84, 24] and plane-to-plane [123] variants substitute the point-to-point correspondence for faster convergence and symmetric alignment.

probabilistic approaches for alignment Unlike the geometric methods that represent a point cloud as a group of points in the Euclidean space, the probabilistic methods use a representation based on mixture family distributions to better handle noise. For example, Myronenko *et al.* [100] consider one point cloud as a GMM and another one as samples drawn from it. Consequently, it casts the alignment problem to a semi-parametric estimation one that can be solved via the Expectation-Maximization algorithm. Another thread of work fits distributions to two point clouds, whose optimal transformation are found by minimizing the divergence between them. In specific, Jian and Vemuri [65] recommend minimizing the L_2 -divergence between GMMs. Other distributions and divergences are also explored, such as Student’s t -distribution [170], hybrid mixture model [97], Jensen—Shannon divergence [146] and Cauchy—Schwarz divergence [58], just to name a few.

Uncertainty estimation strategies Covariance and bandwidth are parameters that emerge in probabilistic methods for mapping point clouds to distributions. Some of the previous works estimate the covariance with clustering techniques [11, 19, 132] or via sensor modeling [90], and keep it unchanged during alignment. Others decide to set the covariance as a free shrinkage parameter to perform global-to-local matching [29], which is realized by either a multiplier [47] or deterministic annealing [66, 29, 167]. In contrast with these empirical solutions, we analyze in detail the differences between covariance and bandwidth, and present a simple but effective approach to updating the bandwidth.

Attempts to unify geometric and probabilistic approaches There already exists literature that attempts to unify geometric and probabilistic approaches. For example, Granger and Pennec [47] interpret ICP as minimizing the sum of Mahalanobis distances between Gaussian distributions. Jian and Vemuri [65] explain ICP as an approximation of minimizing KL-divergence between two GMMs. Segal *et al.* [123] use maximum likelihood estimation (MLE) to maximize the sum of probabilities of distances between paired points, which actually leads to the same conclusion drawn by [65] but from a different perspective. The main difference between these existing works and our proposal is that, while they treat ICP as a particular case of their respective objective functions without referring to its variants, we provide thorough theoretical explanations to unify geometric and probabilistic methods without any exception.

2.3 A unified framework of geometric and probabilistic approaches

In this section, we present the details of our unified explanation of the existing geometric and probabilistic point cloud alignment approaches. We also provide a detailed analysis of the two unique geometric methods, *i.e.*, the point-to-plane and the plane-to-plane ICP algorithms, whose problem formulations cannot be directly explained by our unified framework.

2.3.1 Unification of generic geometric and probabilistic approaches

Our work is inspired by the theory of information geometry [3]. It studies statistics with geometric tools by considering distributions as points on Riemannian manifolds, where Fisher information is a natural choice of metric. Before unifying geometric and probabilistic approaches, we first need to bridge the distribution-fitting and divergence-minimization methods within the probabilistic class.

Lemma 1. *Maximizing likelihood is equivalent to minimizing KL-divergence:*

$$\operatorname{argmax}_{\theta} P(x; \theta) = \operatorname{argmin}_{\theta} \operatorname{KL}(P(x; \theta^*) | P(x; \theta)),$$

where θ is the estimated parameter and θ^* is the ground-truth one.

While Lemma 1 broadly raises in literature [148], we present a proof here to make this thesis self-contained.

Proof. Given a ground-truth distribution $P(x; \theta^*)$ and an estimated one $P(x; \theta)$, minimizing the KL-divergence between them can be equivalently written as

$$\begin{aligned} \operatorname{KL}(P(x; \theta^*) | P(x; \theta)) &= \int P(x; \theta^*) \log \left(\frac{P(x; \theta^*)}{P(x; \theta)} \right) dx \\ &= \int P(x; \theta^*) \log(P(x; \theta^*)) - \int P(x; \theta^*) \log(P(x; \theta)). \end{aligned} \quad (2.1)$$

Since the first term on the right side is a constant w.r.t. θ , minimizing the KL-divergence leads to

$$\begin{aligned} \min_{\theta} \operatorname{KL}(P(x; \theta^*) | P(x; \theta)) &= \max_{\theta} \int P(x; \theta^*) \log(P(x; \theta)) \\ &= \max_{\theta} E_{P(x; \theta^*)} [\log(P(x; \theta))] \\ &= \max_{\theta} \lim_{n \rightarrow \infty} \frac{1}{N} \sum \log(P(x_i; \theta)), \end{aligned} \quad (2.2)$$

where x_i are samples drawn from the ground-truth distribution $\log(P(x; \theta^*))$.

One the other hand, we can write MLE into

$$\begin{aligned} \max_{\theta} \prod (P(x_i; \theta)) &= \max_{\theta} \log(\Pi(P(x_i; \theta))) \\ &= \max_{\theta} \sum \log(P(x_i; \theta)). \end{aligned} \quad (2.3)$$

Since Eq. (2.3) and Eq. (2.2) are the same up to a scalar, we can conclude that MLE is the same as minimization of the KL-divergence. $\square$

Per Corollary 1, the objective function of distribution-fitting alignment methods can be rewritten as minimizing divergence:

$$\operatorname{argmax}_{\mathbf{T}} \prod P(\mathbf{b}_j; A, \mathbf{T}, \theta) \equiv \operatorname{argmin}_{\mathbf{T}} \operatorname{KL}(P(B; \theta_b) | P(A; \mathbf{T}, \theta_a)), \quad (2.4)$$

where $P(A; \mathbf{T}, \theta_a) = \sum \alpha_i P(\mathbf{a}_i; \mathbf{T}, \theta_{ai})$ and $P(B; \theta_b) = \sum \beta_j P(\mathbf{b}_j; \theta_{bj})$ are MMs constructed from point clouds $\mathbf{A}$ and $\mathbf{B}$. We hereafter denote them as $P(A) = \sum \alpha_i P(\mathbf{a}_i)$ and $P(B) = \sum \beta_j P(\mathbf{b}_j)$ for notational simplicity. Now, all the probabilistic methods can be seen as a divergence-minimization problem, which lays the foundation for our unification.

KL-divergence is not the only choice for formulating the objective function. In fact, as employed by numerous related works [65, 58, 146], there are many off-the-shelf divergences for selection. However, not all of them are suitable for the alignment problem. As a naive rule, we desire the selected divergence to be invariant when the points are encoded in different ways. Specifically, given a mapping h , we want the divergence D to satisfy

$$D(P|Q) = D(h(P) | h(Q)). \quad (2.5)$$

Equation (2.5) directly leads to the theory of f -divergence, which contains a group of famous divergences such as KL-divergence, total variation, and all of the α -divergences. We present its formal definition in Lemma 2.

Lemma 2. *The f -divergence:*

$$D_f(P|Q) \triangleq \int_{\Omega} p f\left(\frac{q}{p}\right) d\mu,$$

is an invariant measurement, where μ is the reference distribution satisfying $p = \frac{dP}{d\mu}$ and $q = \frac{dQ}{d\mu}$, and f is a smooth convex function. It is convex w.r.t. both P and Q .

A detailed introduction together with proof of Lemma 2 can be found in *P54-P65* of [3]. Although the f -divergence is a reasonable choice to formulate the objective function, it is seldom analytically computable when the incorporated distributions are MMs. Therefore, we need to find a solvable upper bound for optimization purposes.

Given two MMs written in the form of $P = \sum_i^M \alpha_i P_i$ and $Q = \sum_j^N \beta_j Q_j$, where P_i and Q_j are the respective component distributions of P and Q , and α_i and β_j are their weights; we can treat them as having the same number of components

$$\text{where } \left\{ \begin{array}{l} P = \sum_i^{MN} \alpha_{i,n} P_{i,n} \\ Q = \sum_j^{MN} \beta_{j,m} Q_{j,m} \\ \alpha_{i,1} = \cdots = \alpha_{i,n} = \frac{\alpha_i}{N} \\ \beta_{j,1} = \cdots = \beta_{j,m} = \frac{\beta_j}{M} \\ P_{i,1} = \cdots = P_{i,n} = P_i \\ Q_{j,1} = \cdots = Q_{j,m} = Q_j \end{array} \right. \quad (2.6)$$

Owing to its convexity, we can define an upper bound of the f -divergence $D_f(P|Q)$ in the summation form via Jensen's inequality [117]

$$\begin{aligned} D_f(P|Q) &= D_f \left(\sum_{i,n}^{M,N} \alpha_{i,n} P_{i,n} \middle| \sum_{j,m}^{M,N} \beta_{j,m} Q_{j,m} \right) \\ &\leq \sum_{i,n,j,m}^{M,N,M,N} D_f(\alpha_{i,n} P_{i,n} | \beta_{j,m} Q_{j,m}). \end{aligned} \quad (2.7)$$

Moreover, since MMs are invariant to permutation of indices, we can always tighten the upper bounds in the form of

$$\begin{aligned} D_f(P|Q) &\leq \sum_{i,n,j,m}^{M,N,M,N} D_f(\alpha_{i,n} P_{i,n} | \beta_{s(i)} Q_{s(i)}) \\ &\leq \sum_{i,n,j,m}^{M,N,M,N} \pi_{ij} D_f(\alpha_{i,n} P_{i,n} | \beta_{j,m} Q_{j,m}) \\ &= \sum_{i,n,j,m}^{M,N,M,N} \pi_{ij} D_f \left(\frac{\alpha_i}{N} P_{i,n} \middle| \frac{\beta_j}{M} Q_{j,m} \right), \end{aligned} \quad (2.8)$$

Algorithm 1: The unified algorithm for point cloud alignment

Input: point clouds **A** and **B**, initial transformation **T**

While not converged:

- **Step 1:** Find a mapping $\varepsilon : \mathbb{R}^3 \rightarrow \mathcal{M}$ (or $\mathbb{R}^3$) for each point $\mathbf{a}_i$ and $\mathbf{b}_j$ under current **T**.
- **Step 2:** Establish correspondences on $\mathcal{M}$ (or $\mathbb{R}^3$) under current **T**.
- **Step 3:** Minimize the sum of divergences (or Euclidean distances) w.r.t. **T**.

Output: Estimated transformation **T**

where $\beta_{s(i)}Q_{s(i)}$ denotes the nearest component of Q w.r.t. $\alpha_{i,n}P_{j,m}$, and π_{ij} is a weight coefficient s.t. $\sum_j \pi_{ij} = 1$.

Moreover, since points within each point cloud are equally weighted, the MMs generated by the two point clouds obey $\alpha_i = \beta_j = \frac{1}{M}$. Therefore, by applying Eq. (2.8) to the alignment problem, the objective function can be written as

$$J = \sum_{i,n,j,m}^{M,N,M,N} \pi_{ij} D_f \left(\frac{1}{MN} P(\mathbf{a}_i) \middle| \frac{1}{NM} P(\mathbf{b}_j) \right). \quad (2.9)$$

Moreover, according to the invariance property of the f -divergence, we can equally scale each component distribution by MN without changing the objective function. Consequently, the above equation is simplified to

$$J = \sum_{i,j}^{M,N} \pi_{ij} D_f \left(P(\mathbf{a}_i) \middle| P(\mathbf{b}_j) \right). \quad (2.10)$$

Recalling that the objective functions of ICP in its generalized form (*i.e.*, soft assignment) is written as

$$J = \sum \pi_{ij} \left\| \mathbf{T} \circ \mathbf{a}_i - \mathbf{b}_j \right\|_2^2, \quad (2.11)$$

we can see that it is very similar to Eq. (2.10), except that the Euclidean distance is replaced by the f -divergence. In order to present a more comprehensive understanding of what probabilistic methods actually do, we introduce another property of the f -divergence shown in Lemma 3.

Lemma 3. *The f -divergence naturally introduces geometric structures to P and Q , where fisher information metric (FIM) is the associated Riemannian metric. Also, we can always*

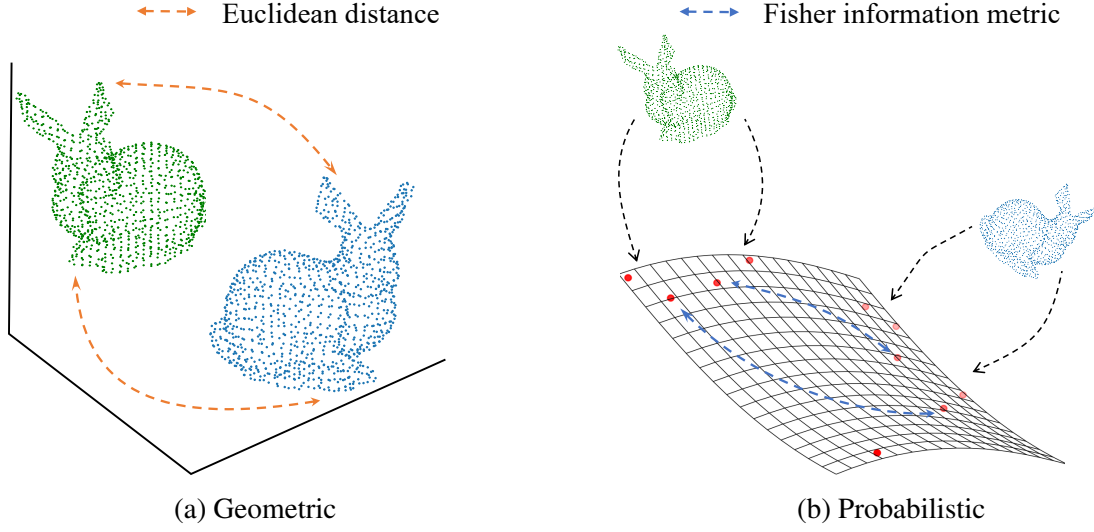


Fig. 2.1 A conceptual illustration of how geometric and probabilistic methods work. While geometric approaches operate on the Euclidean space with euclidean distance as the metric, probabilistic algorithms work on the probabilistic manifold with the associated Fisher information metric.

use a convex function f satisfying $f(1) = 0$, $f'(1) = 0$ and $f''(1) = 1$ to formulate a standard f -divergence.

Lemma 3 in fact leads to the theory of information geometry [3], which studies statistics with geometric tools by treating distributions as points on a Riemannian manifold $\mathcal{M}$. Since existing works mainly treated Lemma 3 as a corollary without any detailed proof, we present a proof of it in the following.

Proof. Given two neighboring points $P(x; \theta)$ and $P(x; \theta + \delta\theta)$ (hereafter denoted as $P(\theta)$ and $P(\theta + \delta\theta)$ for notational simplicity) on $\mathcal{M}$, we can obtain the local form of the f -divergence:

$$D_f(P(\theta) | P(\theta + \delta\theta)) = \int P(\theta) f\left(\frac{P(\theta + \delta\theta)}{P(\theta)}\right). \quad (2.12)$$

By expanding the term $f(\cdot)$ with Taylor expansion to the 2^{nd} order and using the properties $f(1) = 0$, $f'(1) = 0$ and $f''(1) = 1$, Eq. (2.12) can be rewritten into

$$\begin{aligned}
D_f &= \int P(\theta) \left(f\left(\frac{P(\theta)}{P(\theta)}\right) + \frac{\partial f}{\partial \theta} \delta \theta + \frac{1}{2} \delta \theta^T \frac{\partial^2 f}{\partial \theta^2} \delta \theta \right) \\
&= \frac{1}{2} \int P(\theta) \cdot \left(0 + 0 + \delta \theta^T \frac{\partial^2 f}{\partial \theta^2} \delta \theta \right) \\
&= \frac{1}{2} \delta \theta^T \left(\int \cdot \left(\frac{1}{P(\theta)} \right) \cdot \left(\frac{\partial}{\partial \theta} P(\theta) \right) \left(\frac{\partial}{\partial \theta} P(\theta) \right) \right) \delta \theta \\
&= \frac{1}{2} \delta \theta^T \left(\int P(\theta) \frac{\partial \log P(\theta)}{\partial \theta} \frac{\partial \log P(\theta)}{\partial \theta} \right) \delta \theta \\
&= \frac{1}{2} g_{ij} d\theta^i d\theta^j,
\end{aligned} \tag{2.13}$$

where Einstein Summation form is used and $g_{ij} = \int P(\theta) \frac{\partial \log P(\theta)}{\partial \theta_i} \frac{\partial \log P(\theta)}{\partial \theta_j}$ is exactly the FIM. $\square$

Since the last line of Eq. (2.13) is a general notation for distance, we can also express the euclidean distance in the same form by changing g_{ij} to the Kronecker-delta. Therefore, we can now express the objective functions of both probabilistic and geometric methods in the unified form of

$$J = \frac{\pi_k}{2} g_{ij}^k d\theta_k^i d\theta_k^j, \tag{2.14}$$

where k denotes the indices of pairs of points; and g_{ij} encodes distinct metrics for probabilistic and geometric methods.

With this unified objective function, we can conclude that the two classes of algorithms, in fact, conduct the same operations in different spaces associated with different metrics. Specifically, as illustrated in Fig. 2.1, same as the classical ICP algorithm and its variants, both geometric and probabilistic methods can be interpreted as iterating between matching and pose estimation until convergence, but just in distinct spaces.

One merit of the unified explanation is that, we are freed from having to select between geometric and probabilistic alignment algorithms. That is, we can always develop an algorithm that imitates ICP and its variants on handling correspondences while keeping the algorithm probabilistic to handle noise. We summarize the unified algorithm in Algorithm 1, where ε is an endomorphism ($\mathbb{R}^3 \rightarrow \mathbb{R}^3$) for geometric methods and a homomorphism ($\mathbb{R}^3 \rightarrow \mathcal{M}$) for probabilistic counterparts.

It is noteworthy that f -divergence does not exclude the others from consideration. For example, we consider the case of a GMM with isotropic noise. In such a condition, an objective function based on $L2$ -divergence [65] is in fact equivalent to an upper bound of the symmetrized KL-divergence-based one in terms of optimizing $\mathbf{T}$, although it belongs to the

Bregman family. However, whether such a loose upper bound (same weight between any pairs of points) is enough for achieving accurate estimations is another story.

2.3.2 Unification of point-to-plane and plane-to-plane methods

Different from many other variants of ICP, the point-to-plane [84] algorithm developed for faster convergence and the plane-to-plane [123] one for symmetric alignment do not share the same geometric intuitions of ICP. Therefore, in this section, we show how to interpret them under our unified framework.

The point-to-plane method is introduced as minimizing the sum of differences along surface normals. The objective function is in the form of

$$J(\mathbf{T}) = \sum \left\| \mathbf{n}_{s(i)}^T (\mathbf{T} \circ \mathbf{a}_i - \mathbf{b}_{s(i)}) \right\|_2^2, \quad (2.15)$$

where $\mathbf{b}_{s(i)}$ denotes the nearest point to $\mathbf{a}_i$, and $\mathbf{n}_{s(i)}$ is the surface normal at $\mathbf{b}_{s(i)}$. On the other hand, the plane-to-plane approach fits Gaussian distributions on both point clouds and maximizes the likelihood of the residuals $\mathbf{b}_{s(i)} - \mathbf{T} \circ \mathbf{a}_i$ w.r.t. $\mathbf{T}$, which leads to the the following objective function

$$J(\mathbf{T}) = \sum (\mathbf{b}_{s(i)} - \mathbf{T} \circ \mathbf{a}_i)^T \left(\mathbf{\Sigma}_{b_{s(i)}} + \mathbf{R} \mathbf{\Sigma}_{a_i} \mathbf{R}^T \right)^{-1} (\mathbf{b}_{s(i)} - \mathbf{T} \circ \mathbf{a}_i), \quad (2.16)$$

where $\mathbf{\Sigma}_{a_i}$ and $\mathbf{\Sigma}_{b_{s(i)}}$ are the bandwidths determined by the spans of surface normals at points $\mathbf{a}_i$ and $\mathbf{b}_{s(i)}$. Furthermore, if we assume $\mathbf{\Sigma}_{a_i} = \mathbf{0}$ holds for all $\mathbf{a}_i$, Eq. (2.16) can be rewritten as

$$\begin{aligned} J(\mathbf{T}) &= \sum (\mathbf{b}_{s(i)} - \mathbf{T} \circ \mathbf{a}_i)^T \mathbf{\Sigma}_{b_{s(i)}}^{-1} (\mathbf{b}_{s(i)} - \mathbf{T} \circ \mathbf{a}_i) . \\ &= \sum \left\| \eta_{s(i)}^T (\mathbf{T} \circ \mathbf{a}_i - \mathbf{b}_{s(i)}) \right\|_2^2. \end{aligned} \quad (2.17)$$

The equality of Eq. (2.15) and Eq. (2.17) demonstrates the relations between point-to-plane and plan-to-plane approaches. In fact, as analyzed by the plane-to-plane method itself [123], the point-to-plane algorithm can be treated as one of its instance.

Due to the same formulation of objective functions of the two methods, we hereafter only discuss how Eq. (2.16) is related to our framework. Specifically, we can easily obtain the objective of the plane-to-plane algorithm by replacing the f -divergence in our unification with the Mahalanobis distance [95], which belongs to the Bregman family, and the normals serve as a guidance for bandwidth estimation.

2.4 Bandwidth estimation using an intrinsically contained hint

The aforementioned Algorithm 1 is abstract, and we need to concretize the mapping $\varepsilon : \mathbb{R}^3 \rightarrow \mathcal{M}$ therein to make it practically usable. Fortunately, the aforementioned f -divergence intrinsically contains a hint for doing so. To maintain consistency with other alignment literature, and without loss of generality, we hereafter assume that all the points within a given point cloud are contaminated by the same isotropic Gaussian noise.

Since the expectation of each Gaussian distribution can be naturally determined as the point position in the Euclidean space, the only remaining problem related to the mapping pertains to the covariances Σ_a and Σ_b . In the case of clustering-based approaches [11, 132], they are easy to estimate and can be treated as known parameters. However, in general cases, the mapping is considered as a bijection, which leads to the bandwidth estimation problem of KDE.

Empirically, the bandwidth of a general KDE can be estimated beforehand and fixed with rule-of-thumb methods [126]. However, in the context of the alignment problem, a dynamically updated one would be more suitable since the relative position of two point clouds varies according to the transformation $\mathbf{T}$.

The heuristic method of deterministic annealing is a popular tool for setting the bandwidths Σ_a and Σ_b [66, 65, 58]. Its philosophy is based on the asymptotic behaviors as studied by Chui *et al.* [29]. In detail, Σ_a and Σ_b control the correspondence π_{ij} between point clouds. The correspondences are vague when Σ_a and Σ_b are large, and become asymptotically binary when they approach $\mathbf{0}$. Therefore, in order to take advantage of both soft assignment and binary correspondences to avoid local minima and obtain accurate alignment, deterministic annealing gradually decreases Σ_a and Σ_b from large initial values until convergence. Although promising performances have been observed, we hesitate to do so since bandwidth estimation is an independent problem in statistics with more effective solutions.

In this work, we claim that the objective function developed for alignment intrinsically acts as a guidance for dynamically setting the bandwidths. Specifically, it reminds us of the bandwidth estimation technique based on risk functions [116], where the optimal bandwidth is the one that minimizes a risk function L w.r.t. the ground-truth distribution [49].

Remark 1. A risk function L is defined as the expectation of a loss f : $L = E[f]$. Applying to bandwidth estimation of KDE, the optimal bandwidth of a KDE p can be retrieved by minimizing $E[f(p, q)]$, where q is the underlying ground-truth distribution for reference.

According to Remark 1, the f -divergence naturally formulates one risk function in the form of $D_f = E_p \left[f \left(\frac{q}{p} \right) \right]$. Therefore, if we alternately fix the point $P(\mathbf{a}_i)$ or $P(\mathbf{b}_j)$ as the reference, the f -divergence also serves as an objective function for estimating the bandwidth of the other one.

If the reference distribution q is the ground-truth, a plausible bandwidth of the variable distribution p would be the one that minimizes the difference between them. However, for the alignment problem, since q is just temporarily fixed, directly tuning p towards it would unavoidably introduce large variance for smoothing purposes. This problem is known as the bias-variance-tradeoff, where we want to reduce the redundancy of each distribution while keeping it similar to the “ground-truth.” As a solution, we employ the minimizing entropy criterion proposed by Jiang *et al.* [68] as a solution. Specifically, we can limit the variance of a KDE by adding a regularization term of its entropy, which leads to our final objective function:

$$J(\mathbf{T}, \boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_b) = \sum \pi_{ij} D_f(P(\mathbf{a}_i) | P(\mathbf{b}_j)) + \lambda_1 H(P(A)) + \lambda_2 H(P(B)), \quad (2.18)$$

where $H(\cdot)$ is the entropy, and λ_1 and λ_2 are the regularization parameters. As commonly used in other alignment algorithms, Eq. (2.18) can be facily minimized via alternating optimization, which update the transformation $\mathbf{T}$, the bandwidths $\boldsymbol{\Sigma}$ and the correspondences π alternately.

2.5 How unification helps mutual benefit: A case study

Based on the aforementioned unification, we can always design an algorithm that takes advantage of both probabilistic and geometric methods, namely, an algorithm based on divergence-minimization with explicitly maintained correspondences. In this section, we take the symmetric alignment task as an example to demonstrate its effectiveness.

2.5.1 Problem formulation

Assuming that the two point clouds $\mathbf{A}$ and $\mathbf{B}$ consist of a similar amount of points and no preliminary information of the alignment order ($\mathbf{A} \rightarrow \mathbf{B}$ or $\mathbf{B} \rightarrow \mathbf{A}$) is given, symmetric alignment aims at alleviating the effects of alignment order on the final results. That is, we want $\mathbf{A} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{A}$ to return the same estimation of transformations and correspondences.

This problem is easy to solve for geometric methods. For example, as used in [45, 30, 160], it is a common practice to convert the aforementioned soft assignment matrix π to an (asymptotic) doubly stochastic one in each iteration. In detail, the property of doubly

Algorithm 2: A probabilistic method for symmetric alignment

Input: point clouds $\mathbf{A}$ and $\mathbf{B}$, initial transformation $\mathbf{T}$, and initial scales of the isotropic covariance matrices σ_a and σ_b

While not converged:

- **Step 1:** Calculate a soft assignment matrix π under current $\mathbf{T}$, σ_a , and σ_b
 - **Step 2:** Map π into a doubly stochastic matrix
 - **Step 3:** Update $\mathbf{T}$, σ_a , and σ_b by minimizing Eq. (2.18) with current π
-

stochastic matrices that each row and column sums to 1 helps to ignore the normalization axis of π , hence the alignment order is perfectly balanced. On the other hand, despite their superiority on handling noises, dealing with symmetric registration is a weakness of existing probabilistic methods. In detail, the “ $\mathbf{A}$ as model and $\mathbf{B}$ as data” principle of distribution-fitting-based methods [100, 97] naturally requires a pre-defined alignment order. For the methods based on divergence-minimization, although we can rely on some symmetric divergences [146, 58], there still lacks a unified solution when using general divergences.

On the other hand, our unification helps the statistical algorithms to realize symmetric registration. In detail, if we treat each component distribution as a point on $\mathcal{M}$, we can naturally follow the geometric methods and convert the assignment matrix π to doubly stochastic in each iteration. The only difference is that, while geometric methods use the Euclidean distance, the probabilistic ones must replace it with a divergence. Therefore, by further combining with our bandwidth estimation strategy mentioned in Chapter 2.4, we can obtain an algorithm for symmetric alignment summarized in Algorithm 2.

2.5.2 Experiments

We conduct experiments to show how our unification can combine the merits of geometric and probabilistic methods to further improve the alignment accuracy. We select four candidate methods for comparison: ICP [10], RPM [45], CPD [100], and SVR [19]. Among them, ICP solves the alignment problem in a pure geometric manner. RPM converts the correspondences of ICP to doubly stochastic to realize symmetric alignment. CPD and SVR represent the distribution-fitting and divergence-minimization-based probabilistic approaches, respectively. Regarding the implementations used in the experiments, ICP and RPM are implemented by ourselves, CPD is provided by the authors, and SVR is from the *ProbReg* Python library¹. For all of the following experiments, we assume that the datasets are not under extreme

¹ProbReg: <http://problog.readthedocs.io/>, last accessed on Friday 30th December, 2022

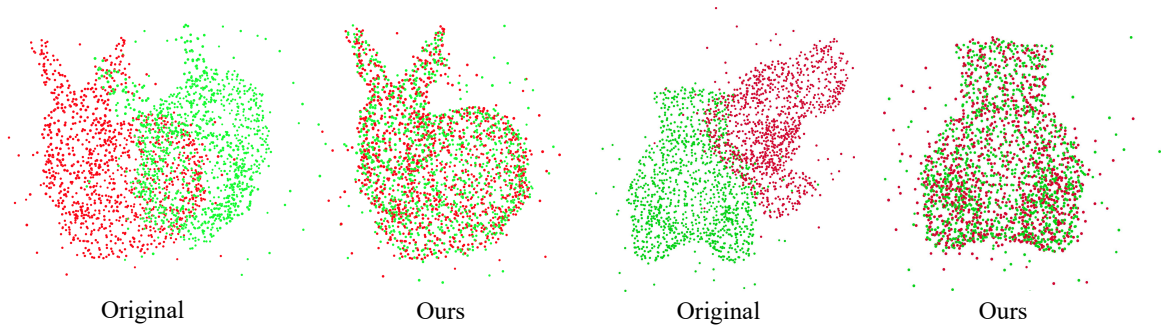


Fig. 2.2 Examples of the synthetic point clouds we used.

conditions (*i.e.*, severe outliers, low overlaps, or significant distinction between numbers of points) since they commonly require additional modification by extending the 4 base approaches with other techniques[25, 107, 171, 87].

We use the symmetrized KL-divergence to specify the f -divergence for maintaining symmetry. Also, in order to demonstrate that the tools developed for geometric methods can be directly applied to probabilistic ones, we follow RPM [47] and use softmax for soft assignment and Sinkhorn iteration [127] for converting the assignment matrix π to doubly stochastic. For each pair of point clouds in both synthetic and realistic datasets, we conduct the alignment from both $A \rightarrow B$ and $B \rightarrow A$. We use $\mathbf{R}\text{-Err} = \|\mathbf{I} - \mathbf{R}_{\text{gt}}^T \mathbf{R}_{\text{esti}}\|_F$, $\mathbf{t}\text{-Err} = \|\mathbf{t}_{\text{esti}} - \mathbf{t}_{\text{gt}}\|_2$ and $\mathbf{T}\text{-Err} = \|\mathbf{I} - \mathbf{T}_{\text{esti}}^{-1} \mathbf{T}_{\text{gt}}\|_F$ as the respective error metrics for rotation, translation and transformation. For parameter setup, the regularization coefficients for bias-variance-tradeoff in our method are set to $\lambda_1 = \lambda_2 = 50$. The annealing parameter is set to 0.95 for RPM and 0.9 for SVR, with which we observe better performances. For CPD, the weight of uniform distribution is set to 0.1. The maximum iteration is set to 100 for all the algorithms except SVR, which is set to 10 for its slow clustering procedure. In fact, we observe that once the SVR algorithm succeeds, the number of iterations shows minor effects on its final accuracy.

The experiments hereafter are arranged in the following manner:

- Chapters 2.5.3 and 2.5.4 compare different methods using synthetic and real-world point clouds, respectively.
- Chapter 2.5.5 explores the tolerance level of different methods w.r.t. varying noise and outlier levels.
- Chapter 2.5.6 reports the elapsed time.

Table 2.2 Errors on the synthetic data (both rotational and transformational errors are in $1e-2$ scale).

	Mean			Median		
	R err	t err	T err	R err	t err	T err
ICP	39.49	6.93e-4	39.49	2.91	6.71e-4	2.91
RPM	4.92	7.04e-4	4.93	4.29	6.96e-4	4.29
CPD	1.67	3.30e-4	1.67	1.32	3.24e-4	1.32
SVR	11.73	6.02e-4	11.73	4.21	4.08e-4	4.21
Ours	0.93	3.73e-4	0.93	0.96	3.56e-4	0.96

2.5.3 Test on synthetic data

We first test how different algorithms perform on synthetic point clouds. Since noises unavoidably exist in the real world, we follow existing works and add noises to the data to simulate real-world degradations. For the specific experimental setup, we base the tests on the bunny and the dragon point clouds given in [31, 28], and a teddy from Free3D². These point clouds are all down-sampled to around 1000 points for a reasonable runtime. We add isotropic Gaussian noises with standard deviation $\Sigma = 0.0015\mathbf{I}$ and $\Sigma = 0.0025\mathbf{I}$, as well as 10% of outliers to the two point clouds respectively. An example pair of data is presented in Fig. 2.2. The alignment is repeated 20 times on each pair of point clouds. For each trial, a random ground-truth rotation $\mathbf{R}$ drawn from Euler angles in the range of $(0^\circ, 60^\circ)$ along each axis, and a random translation is used. The results are shown in Table 2.2. By unifying probabilistic modeling and geometric corresponding, together with our proposed bandwidth estimation strategy, we can champion both the rotational and the overall transformational estimations, although it performs negligibly inferior to CPD on estimating the translations.

2.5.4 Test on real-world data

We also conduct experiments on a laser scan dataset to study real-world performance. In detail, we base the tests on the first 10 scans of the ETH Hauptgebaude dataset [109]. An example data is presented in Fig. 2.3. This dataset is challenging as it consists of repetitive patterns and partial overlaps. Since some of the original ground-truth rotations are either extremely large that none of the candidates can give reasonable estimations, we manually reset the rotations to the range of $(0^\circ, 60^\circ)$ in Euler angles along each axis. We also change the weight of the uniform distribution of CPD to 0 as we found it barely succeeds with other numbers. Again, each point cloud is down-sampled to around 1000. The results are shown

²Free3D: <http://www.free3d.com>, last accessed on Friday 30th December, 2022

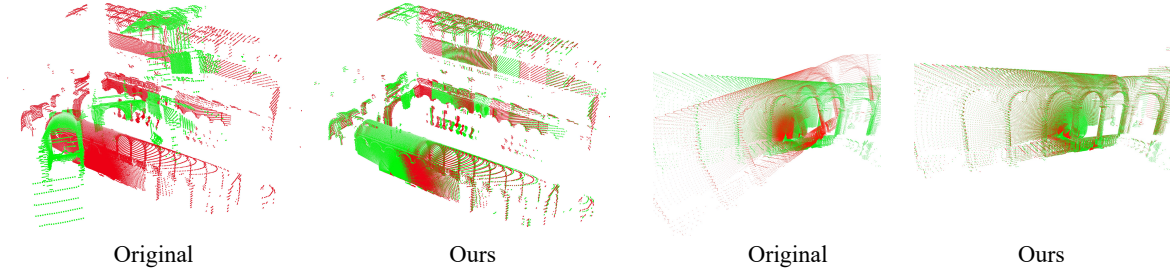


Fig. 2.3 An example of the ETH Hauptgebäude dataset. Some points are cropped for better visualization.

Table 2.3 Errors on the laser scan data.

	Mean			Median		
	R err	t err	T err	R err	t err	T err
ICP	2.11e-2	0.61	0.61	1.90e-2	0.55	0.61
RPM	5.06e-2	0.11	0.11	4.65e-2	0.11	0.11
CPD	2.08e-2	0.15	0.15	1.93e-2	0.15	0.15
SVR	6.69e-2	0.48	0.49	4.21e-2	0.36	0.49
Ours	1.66e-2	0.12	0.12	1.56e-2	0.12	0.12

in Table 2.3. As can be seen, our method outperforms all the others on estimating rotations. For translational and transformational errors, although RPM presents slightly better results, the relative differences between them and ours are small.

2.5.5 Test with varying levels of noise & ratios of outlier

We test the robustness of Algorithm 2 w.r.t. different ratios of outliers and levels of noises. For experimental setup, we use the aforementioned bunny point cloud and fix the ground-truth rotation to 30° in Euler angles. For outliers, we use the 10 ratios uniformly distributed within 5% to 50%. For noises, we vary the covariances of Gaussian noise from $1e - 3\mathbf{I}$ to $5e - 3\mathbf{I}$ with a step size of $5e - 4\mathbf{I}$. We repeat the experiment 10 times with each ratio and each level. The mean errors are plotted in Figs. 2.4 and 2.5. For the case of different levels of noise, our proposal in general performs better than CPD. For the case of distinct ratios of outliers, our method can stably present higher accuracy compared with the others.

2.5.6 Elapsed time of different methods

In this experiment, only ICP, RPM, CPD, and our proposals are considered since all of them are implemented with C++. All the trials are carried ou on a desktop with an Intel i9-9900K

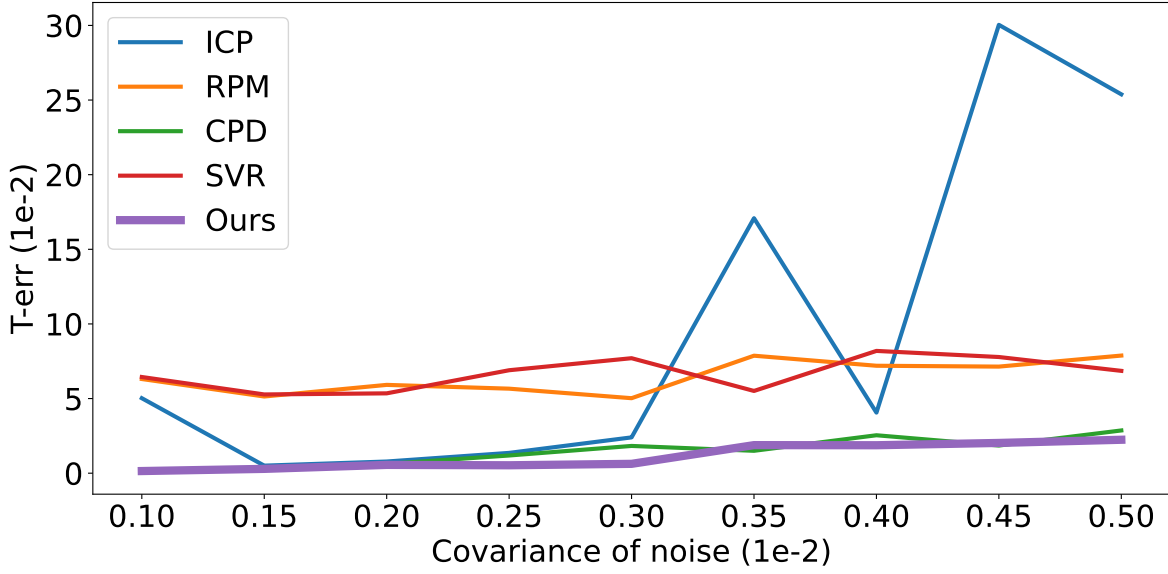


Fig. 2.4 Alignment accuracy w.r.t. varying levels of noises.

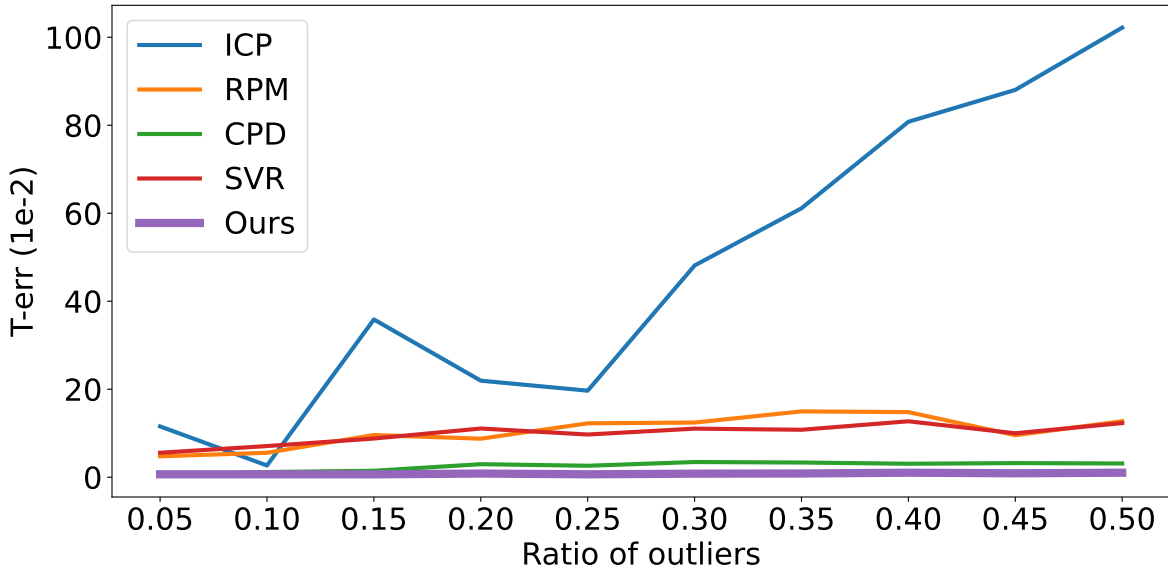


Fig. 2.5 Alignment accuracy w.r.t. varying ratios of outliers.

CPU and 32GB RAM. The values are retrieved from the synthetic experiments mentioned above. We record the time on all of the 4 datasets and their averages are treated as the final values. The results are shown in Table 2.4. Combining geometric and probabilistic algorithm will not significantly enhance the elapsed time.

Table 2.4 Average elapsed time records on the synthetic datasets (second).

	ICP	RPM	CPD	Ours
Avg. elapsed time	0.09	4.13	59.06	7.18

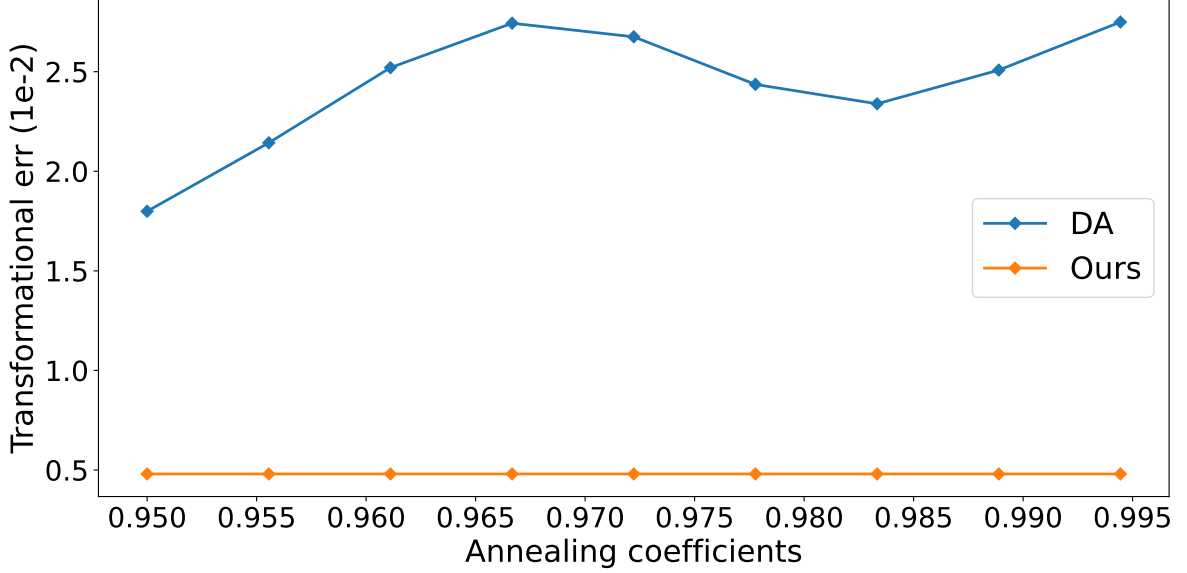


Fig. 2.6 Comparison of the alignment accuracy w.r.t. different annealing parameters. The orange line is achieved by fixing the tradeoff parameters to $\lambda_1 = \lambda_2 = 50$.

2.6 Performance of the proposed strategy for bandwidth estimation

In this section, we study the performance of our proposed bandwidth estimation strategy. For experimental setup, we use the aforementioned synthetic bunny point cloud with a fixed ground-truth rotation of $(30^\circ, 30^\circ, 30^\circ)$ in Euler angles and a random translation. The noises, outliers, and other control variables are maintained the same as mentioned above. All the tests are again based on Algorithm 2 for its good performance, although our unified explanation is applicable to general approaches based on divergence minimization.

2.6.1 Comparisons of alignment accuracy between deterministic annealing and our proposal

We study the relative performance between the commonly used deterministic annealing technique and our proposal to show the superiority of our bandwidth estimation strategy. Specifically, given initial bandwidths, the deterministic annealing technique multiplies it

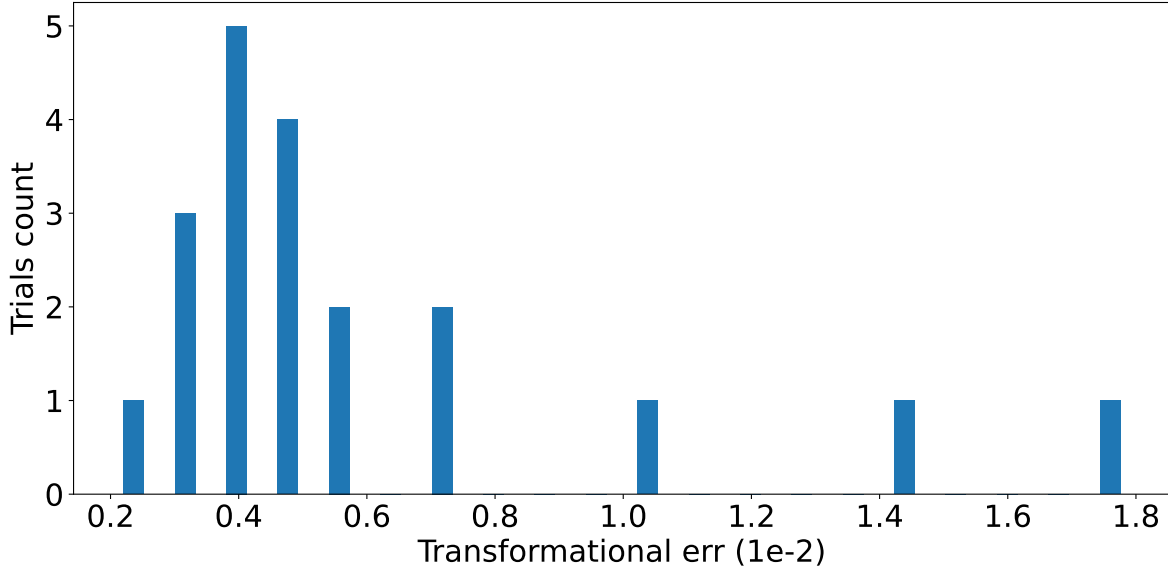


Fig. 2.7 Alignment accuracy w.r.t. different tradeoff parameters λ_1 and λ_2 .

with a shrinkage coefficient s after each iteration. In our implementation, we observe that the alignment performances become significantly unstable when s is within the range of $(0.9, 0.95)$, and it hardly succeeds when $s \in (0, 0.9)$. Therefore, we limit s to be drawn from $(0.95, 1)$ to make the comparison meaningful. The results are reported in Fig. 2.6. It is conspicuous that the performance of deterministic annealing is not stable as minor changes on s led to conspicuous shifts in the estimated results. Moreover, its accuracy is still significantly inferior to our strategy.

2.6.2 Stability of our proposal w.r.t. different trade-off weight λ

As the only free parameters in our proposal, it is meaningful to study how different λ_1 and λ_2 for bias-variance-tradeoff can affect the performance. We again simply assume λ_1 and λ_2 to share the same value: $\lambda_1 = \lambda_2 = \lambda$. For testing purposes, we uniformly pick 20 trials with $\lambda \in [1, 60]$. The final errors are summarized in the histograms presented in Fig. 2.7. As can be seen, our bandwidth estimation strategy is stable w.r.t. varying λ within a broad tuning range. In detail, the apparent dominant error suggests that a large scope of different λ can eventually lead the algorithm to the same estimation. Moreover, compared to the performance of the deterministic annealing mentioned above, the difference between the best and worst estimations of our proposal is only in 10^{-3} magnitude.

2.7 Discussion and conclusion

In this work, we present a unified framework for understanding how geometric and probabilistic point cloud alignment algorithms work. We show that they can be explained as conducting the same operations in different spaces based on the theory of information geometry. We also derive a bandwidth estimation strategy for general probabilistic algorithms based on divergence minimization, which is shown to be a promising substitution of the existing heuristic method, both theoretically and practically. Also, by taking the symmetric alignment task as an example, we show how our unification effectively combines the strengths of both classes, and various experiments demonstrate its effectiveness.

For future works, for practical purposes, one may combine the probabilistic alignment methods with other geometric ones to deal with cases such as partial overlapping, sparse-to-dense alignment and multiple alignment. As another direction of future works, it is interesting to explore how to theoretically select a suitable f -divergence for the alignment problem, which remains an open problem in information science. Also, one may consider about how to design efficient data structures for nearest-neighbor search among distributions, which would play an important role in accelerating the geometric-probabilistic approaches.

Chapter 3

Point cloud alignment with robust permutational constraint

Given the conclusion in Chapter 2 that a majority of point cloud alignment methods, in fact, work by repeating the matching-and-pose-estimation process, in this chapter, we aim to address a flaw that commonly raises in the matching process. *I.e.*, existing matching strategies cannot simultaneously maintain the feasibility of estimation and robustness w.r.t. outliers. To this end, we propose a generalized version of the shuffled linear regression problem, which we call *generalized shuffled linear regression* (GSLR).

As a variant of ordinary linear regression, (G)SLR requires estimating not only the regression variable, but also the permutational correspondences between the covariates and responses. While existing SLR formulations require the underlying ground-truth correspondences to be an ideal bijection such that all pieces of data should match, such a requirement barely holds in real-world applications due to the existence of either missing data or outliers. On the other hand, our GSLR formulation covers a broader range of conditions where only a part of the data should correspond.

In this chapter, besides introducing the detailed formulation of GSLR, we also study its effective recovery condition and NP-hardness. Moreover, we present a simple yet effective optimization algorithm for deriving the solution. Its global convergence property and convergence rate are also analyzed in detail. Distinct tasks validate the effectiveness of our proposed formulation and the solution method.

3.1 Introduction

Linear regression is a powerful tool for various tasks in computer vision. In the least squares form, its objective function is written as

$$\min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2, \quad (3.1)$$

where $\mathbf{A} \in \mathbb{R}^{n \times d}$ ($n \gg d$) and $\mathbf{b} \in \mathbb{R}^n$ respectively stand for the collections of covariates and responses, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ is the regression variable, and $\|\cdot\|$ denotes the Euclidean norm.

Besides the regression variable $\mathbf{x}$, there are cases where the correspondences between $\mathbf{A}$ and $\mathbf{b}$ are also unknown. For example, in computer vision tasks such as point cloud alignment, shape matching, perspective- n -points, and image alignment, the regression variable $\mathbf{x}$ stands for the rigid spatial transformation, spectral mapping, camera projection, and image transformation matrices, respectively. In these cases, correspondences among points, vertices, or pixels are often unknown. Similar situations also appear in other fields, such as data de-anonymization [101] in data science, artifacts dating [42] in archaeology, and sampling with jitter [106] in signal processing.

All these issues have motivated a variant of linear regression recently, namely, the least squares-based *shuffled linear regression* (SLR) problem [106, 163]. Its goal lies in simultaneously recovering both the correspondences between data and the regression variable $\mathbf{x}$. Formally, the objective function of SLR is

$$\min_{\mathbf{P} \in \Pi^\ddagger, \mathbf{x} \in \mathcal{X}} \|\mathbf{P}\mathbf{A}\mathbf{x} - \mathbf{b}\|^2, \quad (3.2)$$

where $\mathbf{P}$ is an n -dimensional square permutation matrix from the discrete and finite set $\Pi^\ddagger$ satisfying

$$\begin{cases} \mathbf{P}\mathbf{1} = \mathbf{1} \\ \mathbf{P}^T\mathbf{1} = \mathbf{1} \\ \mathbf{P}_{ij} \in \{0, 1\} \end{cases}, \quad (3.3)$$

where $\mathbf{1}$ is the all-one vector, and $\mathbf{P}_{ij}$ denotes the ij^{th} entry of $\mathbf{P}$. Intuitively, Constraints (3.3) bring some plausible properties on the relations between each covariate $\mathbf{A}_i$ and response $\mathbf{b}_j$:

- $\mathbf{P}_{ij} \in \{0, 1\}$ ensures $\mathbf{A}_i$ and $\mathbf{b}_j$ are either absolutely matched or completely isolated from each other.
- $\mathbf{P}\mathbf{1} = \mathbf{P}^T\mathbf{1} = \mathbf{1}$ ensures the bijectivity between $\mathbf{A}$ and $\mathbf{b}$ alongside the aforementioned binary property.

This vanilla SLR formulation demands exactly the same cardinalities of $\{\mathbf{A}_i\}$ and $\{\mathbf{b}_j\}$, and all of them to be inliers. However, such requirements are seldom satisfied in practice due to missing data and outliers. For these reasons, the applicability of SLR to real-world tasks is limited.

In this work, we propose a generalized version of SLR: GSLR. In contrast to SLR that requires the underlying ground-truth correspondences to be *all-to-all*, GSLR can also handle *partial-to-all* and *partial-to-partial* cases by constraining the rank of the permutation matrix. Our contributions are summarized as follows:

- We present GSLR, a generalized formulation of SLR. By modifying the conventional permutational constraint to be rank deficient, GSLR is applicable to practical situations where the correspondences should be partial-to-partial (or all). Its effective recovery condition and computational complexity are also analyzed in detail.
- We propose a simple yet effective optimization algorithm to solve the formulated GSLR problem. Its convergence property is also analyzed in detail, appearing to the global convergent with a sub-linear convergence rate.
- We demonstrate how GSLR can benefit computer vision tasks such as point cloud alignment and isometric shape matching in achieving state-of-the-art accuracies.

3.2 Related works

We first review the prior-arts and applications related to (G)SLR. Relevant techniques and other permutation-constrained problems are also discussed.

3.2.1 Shuffled linear regression

Some pioneering works of SLR focus on theoretical analyses of the problem. For example, Unnikrishnan *et al.* [140] prove that for a given set of covariates $\mathbf{A} \in \mathbb{R}^{n \times d}$, $n \geq 2d$ is necessary for obtaining a unique solution of the regression variable $\mathbf{x}$ in noiseless cases. Pananjady *et al.* [106] further extend this condition to noisy data and show that the SLR problem is NP-hard when the regression variable is not a scalar. Tsakiris and Peng [138] establish a theory of homomorphic sensing, which generalizes SLR to any finite sets of linear transformations over vector subspaces. There also exists a work [60] that provides theoretical guidance on the design of algorithms to globally optimize SLR. However, they are still impractical due to the $\mathcal{O}(n^d)$ computational complexity.

Other literature focuses on practically solving the SLR problem. For example, Abid *et al.* [1, 2] establish a Monte-Carlo expectation-maximization framework by treating correspondences as the hidden parameter. Zhang *et al.* [163] reformulate Objective (3.2) by eliminating $\mathbf{x}$ with a maximum likelihood criterion and propose a framework based on the alternating direction method of multipliers (ADMM) for optimization. There also exist some works that focus on solving simplified versions of Objective (3.2). For example, Ashwin *et al.* [105] estimate $\mathbf{P}\mathbf{A}\mathbf{x}$ as a whole and pose it as a denoising problem. Slawski and Ben-David [128] assume only a small amount of data is disordered, and propose to model the sparsely-shuffled property via $L1$ -norm. Based on the same assumption, they later suggest rejecting the mismatched ones as outliers via robust estimation [129].

We are not the first to generalize SLR to deal with outliers. There already exist some works that assume the ground-truth matching to be partial-to-all. For example, Haghighatshoar and Caire [53] propose to find the correspondences via dynamic programming. Elhami *et al.* [37] apply a brute-force strategy over the permutation matrix. Tsakiris and Peng [138] propose a branch-and-bound strategy over the regression variable. Contrary to these works, our GSLR formulation can additionally deal with partial-to-partial cases.

The work most similar to ours is presented by Nejatbakhsh and Varol [102], which also attempts to generalize SLR to partial-to-partial cases. However, it does not propose a formal objective function and instead optimizes using a random sample consensus (RANSAC)-type method by sampling generalized permutations. In contrast, our proposed objective function enables deterministic optimization, which is more efficient for omitting to sample from the space whose size grows exponentially w.r.t. the number of data. Another work targeting at point cloud alignment [82] also realizes partial-to-partial permutational matching. However, it does not contain the binary constraint, and relies on the concavity of their particular objective function to achieve this. In contrast, our proposed GSLR formulation requires the binary constraint as an essential one, and hence applicable to different scenarios regardless of their concavities.

3.2.2 Similar problems in applications

Although it is only recently that SLR was formally formulated, the problem itself has already existed in various applications. For example, the rigid point cloud alignment problem, which aims at recovering the spatial transformation with unknown point-wise correspondences, can perfectly fit in the GSLR formulation. However, existing approaches, such as iterative closest point (ICP) [5] and its variants, are infeasible algorithms for the (G)SLR problem. In detail, as shown in Fig. 3.1, their NN-based matching strategy may either link a point to multiple queries or leave it unmatched, which consequently violates the one-to-one matching

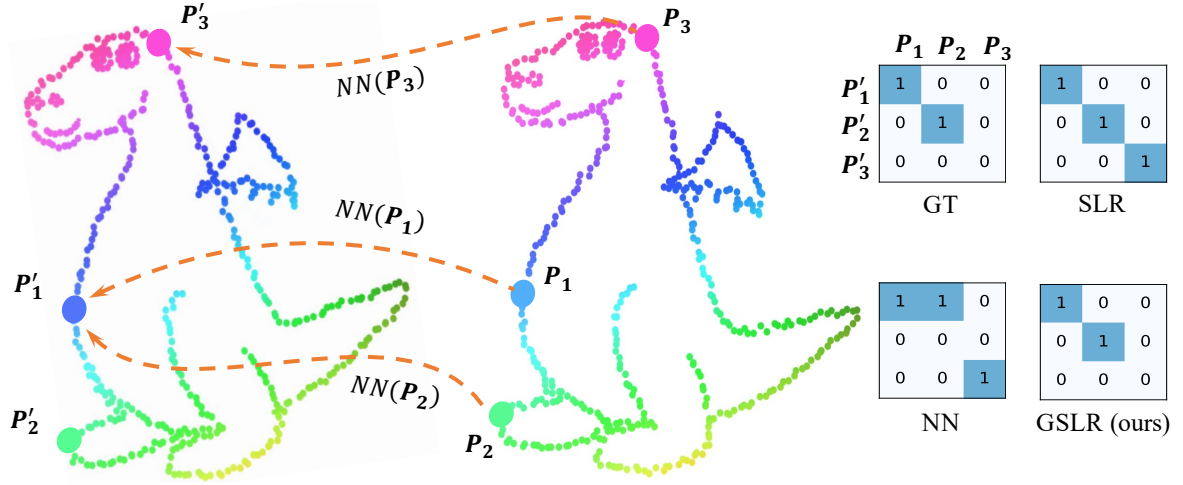


Fig. 3.1 Different strategies for correspondences estimation. GT is short for ground truth. The NN-based method may violate the one-to-one policy by either assigning one point to multiple queries or leaving points unmatched, and SLR cannot reject outliers.

constraint. As we will show in Chapter 3.5, it is this violation that prevents them from achieving stably accurate results.

Some works propose to strengthen the one-to-one correspondence property to solve the aforementioned problem. For example, Jiang *et al.* [67] constrain every row of the assignment matrix to sum to one. Both Jing *et al.* [119] and Szymon [121] propose to symmetrize the alignment process. Gold *et al.* [45] convert the assignment matrix to a doubly stochastic one to balance the weights of correspondences. However, contrary to our GSLR formulation, these attempts still cannot guarantee strict one-to-one correspondences.

There also exist works that encode permutational constraints into the problem formulation. For example, Li and Hartley [78] formulate the alignment problem into SLR by assuming no outliers or missing data. Vestner *et al.* [142] cast the shape matching problem into a rectangular linear assignment problem by maximizing kernel densities in product spaces, though its generalizability is still limited for assuming partial-to-all ground-truth correspondences. Marques *et al.* [91] and Stošić *et al.* [131] show that the permutational matching between some given feature vectors and a prescribed subspace can be determined via convex optimization, benefiting 3D-2D matching tasks.

3.2.3 Robust estimation

Maximum consensus [26] is a popular tool for achieving robust estimation. *E.g.*, the RANSAC-type algorithms [41, 27] propose to repeat sampling a small subset of data to fit the model until it can characterize a majority of the data under a certain tolerance. There

also exists some works [75, 74, 18] that replace the non-deterministic sampling strategies of RANSAC-type algorithms with deterministic ones, making the cardinalities of the consensus sets increase efficiently.

Another class of algorithms for robust estimation is M-estimators [33]. Such methods propose to identify outliers with pre-defined criteria, with which the suspected outliers are either down-weighted (*e.g.*, the Huber loss [62]) or completely rejected (*e.g.*, the trimming operator [157]). In this work, we employ the Huber-skip M-estimator [56] for outlier rejection as it can perfectly preserve the binary constraint of the generalized permutation matrices.

There are two typical criteria to assess the abilities of a robust estimator, namely, the breakdown point and the (statistical) efficiency [120]. Specifically, the breakdown point demonstrates the proportion of outliers an estimator can tolerate before providing an incorrect result. It is theoretically upper-bounded by 0.5 since a robust estimation problem, with no prior knowledge of the outliers, should asymptotically have no fewer inliers than outliers to become valid. The efficiency, calculated as the ratio of the theoretically minimal variance provided by the Cramer-Rao bound w.r.t. the actual one of the estimator, represents the quality of an estimator.

3.2.4 Other permutation-constrained problems

Permutation frequently appears in combinatorial optimization problems. For example, the permutation synchronization problem in computer vision aims at synchronizing the correspondences across multiple sets of objects [104, 92, 12]. Beside permutation itself, some problems require additional constraints to be satisfied. For example, the N-queen puzzle [9] requires every (sub-)diagonal line of the solution to sum to no more than 1. The traveling salesman problem [73] expects every city to be visited only once. Another problem demanding permutational constraint is the quadratic assignment problem (QAP, *e.g.*, second-order graph matching), and there also exist some approaches to robustifying it. For example, both Wang *et al.* [147] and Maciel and Costeira [88] limit the number of matches to a designated number, which inspires our GSLR formulation. Jia *et al.* [64] propose to reject outliers in QAP via sparse optimization. Fiori *et al.* [40] present a sparse coding framework for robust graph matching. However, in contrast to QAP, (G)SLR additionally requires to estimate the regression variable $\mathbf{x}$. Thus, converting (G)SLR to QAP is hardly possible unless the optimal $\mathbf{x}$ can be explicitly expressed by $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{P}$ [163]. *E.g.*, when $\mathbf{x} \in \mathbb{R}^d$, we can eliminate $\mathbf{x}$ from SLR per the relation $\mathbf{x} = (\mathbf{A}^T \mathbf{P}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P}^T \mathbf{b}$.

3.3 Formulation of GSLR

We first present the formal objective function of the GSLR problem. Its effective recovery condition and computational complexity are then analyzed.

3.3.1 Objective function

Given the data $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^n$, a plausible routine of robust estimation is to set zero weights to outlier pairs. For the GSLR problem, we propose to achieve this by reformulating the constraints on the generalized permutation matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$ since it can directly characterize whether a matched pair should be rejected as an outlier.

Let us first explore the properties that $\mathbf{P}$ should satisfy. Apart from the binary and one-to-one matching properties inherited from the square and full-rank counterpart, we also desire it to be capable of characterizing outliers. That is,

- $\mathbf{P}_{ji} = 1$ if $\mathbf{A}_i$ and $\mathbf{b}_j$ should match, otherwise 0.
- $\mathbf{P}_{ji} = 0$ if either $\mathbf{A}_i$ or $\mathbf{b}_j$ is an outlier.
- $\sum_i \mathbf{P}_{ij}, \sum_j \mathbf{P}_{ij} \in \{0, 1\}$ to guarantee that each $\mathbf{A}_i$ and $\mathbf{b}_j$ can have at most one correspondence.

These requirements imply that

$$\begin{cases} \mathbf{P}\mathbf{1} \leq \mathbf{1} \\ \mathbf{P}^T\mathbf{1} \leq \mathbf{1} \\ \mathbf{P}_{ij} \in \{0, 1\} \end{cases}, \quad (3.4)$$

However, Constraints (3.4) are not well-established for optimization since the minimum objective value is anchored to 0 with the associated trivial solution $\mathbf{P} = \mathbf{0}$. Therefore, other constraints are required to guarantee the existence of matches.

A simple yet effective solution is to manually specify the number of matches [147, 88]. In detail, we introduce a positive integer k and force $\text{rank}(\mathbf{P}) = k$, indicating that there must exactly exist k matches. Consequently, the set of the k -ranked generalized permutation matrices is

$$\Pi_k = \{\mathbf{P} : \mathbf{P}\mathbf{1} \leq \mathbf{1}, \mathbf{P}^T\mathbf{1} \leq \mathbf{1}, \mathbf{P}_{ij} \in \{0, 1\}, \text{rank}(\mathbf{P}) = k\}. \quad (3.5)$$

Equation (3.5) implies that Π_k is discrete and finite, and degenerates to the set of full-rank square permutations $\Pi^\ddagger$ when $k = m = n$.

Now we are ready to introduce the objective function of GSLR. Beside using the permutation $\mathbf{P}$ to reorder $\mathbf{A}$ to match $\mathbf{b}$ as conducted in Objective (3.2), GSLR additionally requires

to reject outliers from $\mathbf{b}$. Therefore, we formulate the GSLR problem as

$$\begin{aligned} & \min_{\mathbf{P}, \mathbf{x} \in \mathcal{X}} \|\mathbf{P}\mathbf{A}\mathbf{x} - \mathbf{P}\mathbf{P}^T\mathbf{b}\|^2, \\ & \text{s.t. } \mathbf{P}\mathbf{1} \leq \mathbf{1}, \mathbf{P}^T\mathbf{1} \leq \mathbf{1}, \mathbf{P}_{ij} \in \{0, 1\}, \text{rank}(\mathbf{P}) = k. \end{aligned} \quad (3.6)$$

In Objective (3.6), $\mathbf{P}\mathbf{P}^T$ forms a binary diagonal matrix to identify outliers from $\mathbf{b}$ while maintaining the relative order of its elements unchanged, and the $\mathbf{P}$ operating on $\mathbf{A}$ simultaneously rejects outliers from $\mathbf{A}$ and reorders it to match $\mathbf{b}$. In case GSLR degenerates to SLR, Objectives (3.6) and (3.2) coincide since $\mathbf{P}\mathbf{P}^T$ becomes an identity matrix.

3.3.2 Effective recovery condition

There are some conditions to satisfy to guarantee a meaningful GSLR problem. Specifically, we desire Objective (3.6) to be able to provide a correct and unique estimation of $\mathbf{x}$.

For the capability of estimating correctly, as mentioned in Chapter 3.2.3, since a valid robust estimation problem requires there to be no fewer inliers than outliers, the number of inliers k should satisfy

$$k \geq \frac{\min(m, n)}{2}. \quad (3.7)$$

For the unique recovery property, we claim that our GSLR formulation can perfectly fit in the homomorphic sensing [138] framework, from which the unique recovery condition for the all-to-all and partial-to-all SLR problems can also be derived. Specifically, we introduce the following corollary by combining and instantiating Theorem 1 and Proposition 3 of [138]:

Corollary 1 (Unique recovery condition). *Given the d -dimensional regression variable $\mathbf{x}$, a sufficient condition for GSLR to uniquely recover it is $k \geq 2d$.*

More details of Corollary 1 can be found in Chapter 3.3.4. Furthermore, by combining it with Constraint (3.7), we can conclude the effective recovery conditions of our GSLR formulation as

$$k \in \left[\max \left(2d, \frac{\min(m, n)}{2} \right), \min(m, n) \right]. \quad (3.8)$$

3.3.3 NP-hardness

It is straightforward to reduce from SLR to prove the NP-hardness of generic GSLR problems. However, since SLR is a P problem when $\mathbf{x}$ is a scalar [106, 60], such a proof becomes invalid, making the complexity of GSLR unknown for this particular case. Therefore, in this

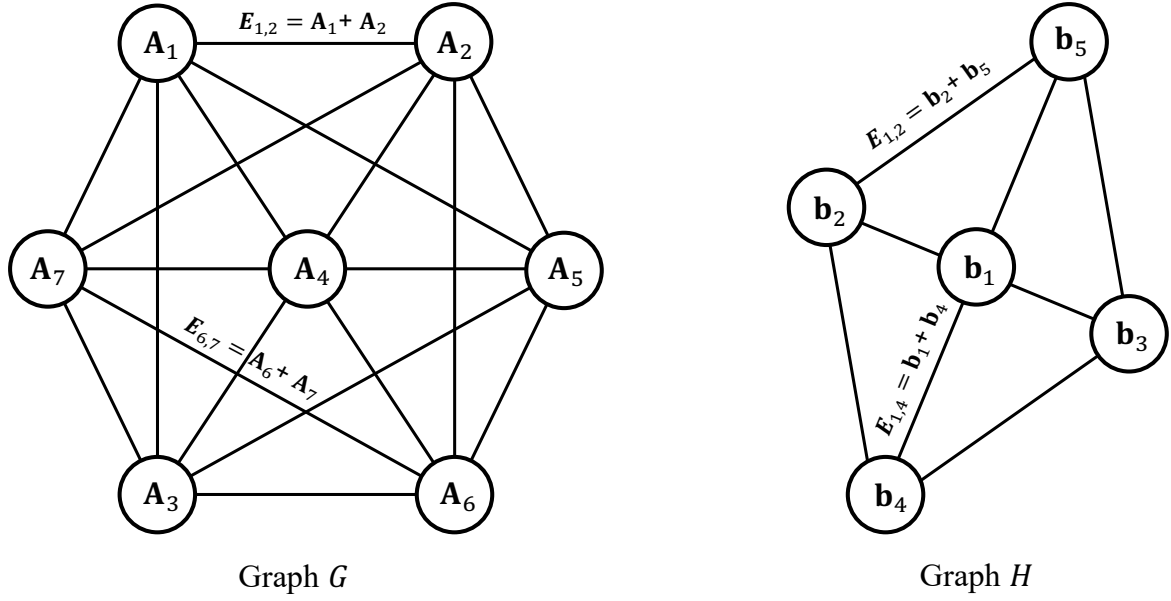


Fig. 3.2 Graphs constructed from the GSLR data. Vertices are labeled by the elements A_i and b_j , and edges are labeled by the sum of their respective adjacent vertices.

work, we base our analysis on the well-known NP-complete subgraph matching problem [86] and demonstrate that GSLR is NP-hard regardless of the dimension of $\mathbf{x}$.

To facilitate the proof, we start from introducing the following subproblem of GSLR:

Problem 1 (A subproblem of GSLR). *Solving the GSLR problem where only $\mathbf{A}$ contains outliers:*

$$\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{b}, \text{ s.t. } \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{P}^T \mathbf{1} \leq \mathbf{1}, \mathbf{P} \in \{0, 1\}.$$

Remark 2 (Trivial reduction from Problem 1 to GSLR). *Since the reduction from Problem 1 to Objective (3.6) is simply the identity mapping, the latter one is NP-hard as long as the former one is.*

Let us also remark the definition of the labeled subgraph matching problem:

Problem 2 (Labeled subgraph matching). *Given two vertex and edge-labeled graphs $G = (V, E)$ and $H = (V', E')$, can we find a subgraph $\tilde{G} = (\tilde{V}, \tilde{E})$ of G that isomorphically matches H ? That is, can we find a $\tilde{G}$ and a bijective mapping $f : \tilde{V} \rightarrow V', \tilde{E} \rightarrow E'$ such that*

$$f(\tilde{E}_{(\tilde{v}_1, \tilde{v}_2)}) = E'_{(f(\tilde{v}_1), f(\tilde{v}_2))} \quad (3.9)$$

holds for any $\tilde{E}_{(\tilde{v}_1, \tilde{v}_2)} \in \tilde{E}$ and $E'_{(f(\tilde{v}_1), f(\tilde{v}_2))} \in E'$?

Now we are ready to demonstrate the NP-hardness of Problem 1 by reducing it from Problem 2. The reduction method is presented in Proposition 1 and its proof is given in Chapter 3.3.4.

Proposition 1 (Problem 2 can reduce to Problem 1). *Given data $(\mathbf{A}, \mathbf{b})$, Problem 1 can be reduced from Problem 2 via constructing the graphs G and H in the following manner (see Fig. 3.2 for illustration):*

- $G : V_i = \mathbf{A}_i, E_{ij} = \mathbf{A}_i + \mathbf{A}_j,$
- $H : V'_i = \mathbf{b}_i, E'_{ij} = \mathbf{b}_i + \mathbf{b}_j.$

An efficient algorithm for noiseless GSLR

The NP-hardness of GSLR does not mean all its instances are infeasible to solve. For example, for the *noiseless but outlier-contaminated* GSLR problem where $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{b}, \text{ s.t. } \mathbf{P}\mathbf{1} \leq \mathbf{1}, \mathbf{P}^T\mathbf{1} \leq \mathbf{1}, \mathbf{P} \in \{0, 1\}, \text{rank}(\mathbf{P}) = k, \quad (3.10)$$

we can design an efficient algorithm by searching over all possible $\mathbf{x}$. Specifically, since the equality in Eq. (3.10) strictly holds true, we know that the underlying ground-truth $\mathbf{x}_{\text{gt}}$ must fall in the set $\mathcal{S}$ consisting of all possible solutions generated by all possible combinations of $\mathbf{A}_i$ and $\mathbf{b}_j$:

$$\mathcal{S} = \{ \mathbf{A}_{\{1, \dots, d\}} / \mathbf{b}_{\{1, \dots, d\}} : \mathbf{A}_{\{1, \dots, d\}} \subseteq \mathbf{A}, \mathbf{b}_{\{1, \dots, d\}} \subseteq \mathbf{b} \}. \quad (3.11)$$

where $\mathbf{A}_{\{1, \dots, d\}}$ and $\mathbf{b}_{\{1, \dots, d\}}$ are d -element subsets of $\mathbf{A}$ and $\mathbf{b}$, respectively, and $\mathbf{A}_{\{1, \dots, d\}} / \mathbf{b}_{\{1, \dots, d\}}$ denotes the solution of the linear equation $\mathbf{A}_{\{1, \dots, d\}}\mathbf{x} = \mathbf{b}_{\{1, \dots, d\}}$ in terms of $\mathbf{x}$. Therefore, we can facilely pass through all possible $\mathbf{x} \in \mathcal{S}$ and take the one that preserves k matches between $\mathbf{A}_i$ and $\mathbf{b}_j$. Algorithm 3 summarizes the entire procedure.

For the computational complexity of Algorithm 3, passing through all possible $\mathbf{x} \in \mathcal{S}$ is of complexity $\mathcal{O}(C_m^d C_n^d)$, and computing the set intersection for each $\mathbf{x}$ is $\mathcal{O}(n)$ (assuming $m \geq n$). Therefore, the overall computational complexity of Algorithm 3 is $\mathcal{O}(nC_m^d C_n^d)$.

3.3.4 Proofs of corollaries, propositions, and lemmas in Chapters 3.3.2 and 3.3.3

We here present the proofs of the corollaries, propositions, and lemmas in the derivation of the unique recover condition and the NP-hardness mentioned above.

Algorithm 3: An efficient exhaustive search algorithm for noiseless but outlier-contaminated GSLR

Input: $\mathbf{A} \in \mathbb{R}^{m \times d}$, $\mathbf{b} \in \mathbb{R}^n$

- Generate $\mathcal{S}$ as defined in Eq. (3.11).
- Compute the number of consensus w.r.t. every $\mathbf{x} \in \mathcal{S}$:
 $\#\text{consensus}_{\mathbf{x}} = \# \{ \{ \mathbf{A}_i \mathbf{x} \} \cap \{ \mathbf{b}_j \} \}$.
- Set $\hat{\mathbf{x}}$ to $\mathbf{x}$ where $\#\text{consensus}_{\mathbf{x}} = k$ and $\hat{\mathbf{P}}$ to the corresponding matches where $\mathbf{A}_i \mathbf{x}$ and $\mathbf{b}_j$ coincide.

Output: $\hat{\mathbf{x}}, \hat{\mathbf{P}}$

Derivation of Corollary 1

Given a generic linear subspace $\mathcal{V} \subset \mathbb{R}^m$ of dimension d (hereafter denoted as $\dim(\mathcal{V}) = d$), and a finite set of endomorphisms $\mathcal{T}$ of $\mathbb{R}^m$, homomorphic sensing [138] presents the unique recovery condition in $\mathcal{V}$ under $\mathcal{T}$. That is, $\forall v_1, v_2 \in \mathcal{V}, \tau_1, \tau_2 \in \mathcal{T}$, when can $\tau_1(v_1) = \tau_2(v_2)$ imply $v_1 = v_2$.

For self-containment, let us first remark Theorem 1 and Proposition 3 of [138], which can be jointly described by the following corollary:

Corollary 2 (Unique recovery condition of homomorphic sensing). *Assuming that $\forall \tau \in \mathcal{T}, \text{rank}(\tau) \geq 2d$, then, we have unique recovery in $\mathcal{V}$ under $\mathcal{T}$ as long as*

$$\dim(\mathcal{V}) \leq \mathcal{Q}_{\tau_1, \tau_2}, \forall \tau_1, \tau_2 \in \mathcal{T}, \quad (3.12)$$

where $\mathcal{Q}_{\tau_1, \tau_2}$ is an intermediate technical term dependent to τ_1 and τ_2 such that, in case $\pi_1, \pi_2 \in \Pi^\ddagger$ are full-rank square permutation matrices, we have

$$\mathcal{Q}_{\rho \pi_1, \rho \pi_2} \geq \frac{\text{rank}(\rho)}{2}, \forall \pi_1, \pi_2 \in \Pi^\ddagger, \quad (3.13)$$

where ρ denotes a coordinate projection matrix.

To apply Corollary 2 to GSLR, we shall consider $\mathbf{A}\mathbf{x}$ as a whole for some d -dimensional $\mathbf{x}$ and set $\mathcal{V} = \{\mathbf{A}\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$. Since linear mapping does not increase dimension, we have $\dim(\mathcal{V}) \leq d$. Furthermore, by setting $\mathcal{T} = \Pi_k$ and the ρ in Eq. (3.13) to the projections that map from $\Pi^\ddagger$ to Π_k , we obtain

$$\mathcal{Q}_{\tau_1, \tau_2} \geq \frac{\text{rank}(\rho)}{2} = \frac{k}{2}, \forall \tau_1, \tau_2 \in \Pi_k. \quad (3.14)$$

Using Eq. (3.14) as a lower bound for Eq. (3.12), we can immediately recognize $k \geq 2d$ as a sufficient condition for unique recovery.

Proof of Proposition 1

Proof of Proposition 1. Since the bijectivity will always be satisfied as long as Problem 2 is solved, we here show that if there exists an f to solve Problem 2 that takes the graphs G and H defined in Proposition 1 as inputs, then this f is a linear mapping from a subset of $\mathbf{A}$ to $\mathbf{b}$. Specifically, since $\text{dom}(f)$ is a finitely discrete set, an f that solves Problem 2 would be a continuous mapping. Furthermore, assuming that vertices are mapped in the form of $\mathbf{A}_i$ to $\mathbf{b}_i$ and $\mathbf{A}_j$ to $\mathbf{b}_j$, we have

$$\underbrace{f(\mathbf{A}_i + \mathbf{A}_j)}_{\text{edge in } \tilde{G}} = \underbrace{\mathbf{b}_i + \mathbf{b}_j}_{\text{edge in } H} = \underbrace{f(\mathbf{A}_i) + f(\mathbf{A}_j)}_{\text{vertices in } H} \quad (3.15)$$

always holds w.r.t. the preimage of $\mathbf{b}$, indicating that f is additive. By combining the continuity and the additivity we can obtain that f is a linear mapping [93]. $\square$

3.4 Optimization of GSLR

In this section, we present a simple yet effective algorithm to feasibly solve the GSLR problem. Its convergence properties are also analyzed in detail.

3.4.1 A simple optimization algorithm

Unlike some existing problems that care little about the recovered correspondences, such as point cloud alignment, (G)SLR considers the permutational constraint an essential condition to satisfy, leading to difficulties of twofold: First, the combinatorial property over the generalized permutation $\mathbf{P}$ makes the objective non-smooth. Second, the number of matches k is hard to determine. As a solution, we propose to seek benefits from the specific properties of GSLR to make the sub-problem of estimating $\mathbf{P}$ under a fixed regression variable $\mathbf{x}$ globally solvable. For clarification, we first introduce the optimization framework by temporarily assuming k as given, and provide the method for determining k .

Solving GSLR with known k

Our proposal is based on a pivotal observation that slight reformulation of the objective function of GSLR would make $\mathbf{P}$ globally optimizable within each iteration. Specifically,

Algorithm 4: The optimization algorithm for GSLR**Input:** $\mathbf{A} \in \mathbb{R}^{m \times d}$, $\mathbf{b} \in \mathbb{R}^n$ **Initialization:** $\mathbf{x} = \mathbf{x}_0 \in \mathcal{X}$ **while** *not converged* **do**

- Calculate $\mathbf{P}_t$ by solving the standard LAP with $k_t = \min(m, n)$ and $\mathbf{x}_{t-1}$.
- Compute residuals under $\mathbf{P}_t$ and $\mathbf{x}_{t-1}$ and apply the Huber-skip estimator to get k_t .
- Set $\hat{k}_t = \min(k_t, k_{t-1})$. Also make sure it is within the range shown in Eq. (3.8).
- Calculate $\hat{\mathbf{P}}_t$ by solving the k -LAP with the updated k_t and $\mathbf{x}_{t-1}$.
- Get $\mathbf{x}_t \in \mathcal{X}$ by solving the weighted linear regression problem under $\hat{\mathbf{P}}_t$.

Output: $\mathbf{x}, \mathbf{P}$

owing to the permutational property of $\mathbf{P}$, we can equivalently rewrite Objective (3.6) into

$$\min_{\mathbf{P} \in \Pi_k, \mathbf{x} \in \mathcal{X}} \sum_{i,j}^{m,n} \mathbf{P}_{ji} \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_j\|^2. \quad (3.16)$$

Applying alternative optimization to Objective (3.16) and treating the correspondences $\mathbf{P}_{ji}$ as known, updating $\mathbf{x}$ leads to a weighted linear least square problem, which can be globally solved in many cases. On the other hand, if $\mathbf{x}$ is fixed, the objective function for updating $\mathbf{P}$ is in the form of

$$\min_{\mathbf{P} \in \Pi_k} \sum_{i,j}^{m,n} \mathbf{D}_{ij} \mathbf{P}_{ji}, \quad (3.17)$$

where $\mathbf{D} = \|\mathbf{A}_i \mathbf{x} - \mathbf{b}_j\|^2$ is the cost matrix.

In cases where the number of correspondences satisfies $k = \min(m, n)$, Objective (3.17) denotes a standard linear assignment problem (LAP) globally optimizable via off-the-shelf approaches such as the Hungarian algorithm [72]. This problem remains solvable even if k makes $\mathbf{P}$ rank-deficient. Specifically, Objective (3.17) forms a k -cardinality LAP (k -LAP) in such cases, which can be transformed into a standard one via padding the cost matrix and thus also solvable by off-the-shelf LAP solvers [144]. Therefore, alternating between solving the k -LAP and the weighted least square can facily address the GSLR problem.

Characterizing k as an M-estimator

The number of matches k characterizes the ability of GSLR in rejecting outliers. A naive strategy is to fix its value by manual specification, which is akin to the trimming operator that discards a specific proportion of data. Although such a technique has been broadly applied in various applications [25, 147], we hesitate to use it for its well-studied limitation on simultaneously maintaining a high breakdown point and high efficiency. That is, although a breakdown point of 0.5 can be facily achieved by ignoring 50% of the data, the associated statistical efficiency would only be 7% [34], representing a large variance in the estimated result.

In this work, we employ another M-estimator to determine k on the fly, namely, the Huber-skip estimator [56]. Apart from maintaining the necessary properties of a binary $\mathbf{P}$ and an integer k , it also achieves significantly higher efficiency while maintaining a breakdown point of 0.5. Specifically, given some data $(r_1, r_2, \dots, r_n)$, it rejects r_i that violates

$$\left| \frac{r_i - \text{median}(r)}{\text{MAD}(r)} \right| \leq \text{threshold}, \quad (3.18)$$

where MAD is the median absolute deviation. The smaller the threshold is, the more data is rejected.

To embed the Huber-skip estimator into the aforementioned optimization framework, we directly follow the formulation of the trimming operator [59] and compute k as the number of pairs whose residuals satisfy Eq. (3.18). However, such a naive formulation would break the monotonicity of the alternative optimization scheme and make its convergence non-guaranteed. Therefore, we set $\hat{k}_t = \min(k_{t-1}, k_t)$ in the t^{th} iteration to ensure convergence. Algorithm 4 summarizes the entire procedure of our proposed algorithm.

Computational complexity

We analyze the space and time complexities of Algorithm 4 under the assumption that $\#\{\mathbf{A}_1, \dots, \mathbf{A}_m\} \geq \#\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ (*i.e.*, $m \geq n$) without loss of generality. Specifically, the space complexity follows the size of the cost matrix, which is $\mathcal{O}(mn)$. When cast to the k -LAP, it changes to $\mathcal{O}(n(m+n-k))$ to suit the Hungarian algorithm. For the time complexity in each iteration, the Hungarian algorithm is $\mathcal{O}(n(m+n-k)^2)$. The Huber-skipping procedure requires sorting and traversing operations; hence the complexity is $\mathcal{O}(n \log n)$. As for the linear regression step, it depends on the feasible space $\mathcal{X}$ and thus varies from case to case.

3.4.2 Convergence analyses

Our proposed Algorithm 4 is globally convergent with sub-linear convergence rate. Since Objective (3.6) is non-smooth, throughout this section, we denote the critical points in the scene of the limiting-subdifferential [79]. Moreover, we create the following hypotheses on the set $\mathcal{X}$ of the regression variable $\mathbf{x}$:

$$\begin{cases} H1: & \mathcal{X} \subset \mathbb{R}^d \text{ is compact and sub-analytic.} \\ H2: & \text{Objective (3.6) is } C^1\text{-smooth on } \mathcal{X}. \end{cases} \quad (\text{H})$$

Remark 3 (Aforementioned Hypotheses are broadly satisfied). *Hypotheses H are rather weak and satisfied by most applications that admit to the GSLR formulation. For example, the (special) orthogonal groups that frequently appear in geometric computer vision can satisfy both conditions. For the space $\mathbb{R}^d$, despite not being compact by itself, we can reasonably assume that $\mathbf{x}$ lies in a closed interval of $\mathbb{R}^d$ since values that tend to infinity are rarely meaningful in practice. Moreover, the closed bounds also imply the sub-analyticity.*

Notations

Throughout Chapters 3.4.2 and 3.4.2, for simplicity, we use Π to denote the set of all the generalized permutation matrices under all possible ranks k : $\Pi = \Pi_{k_1} \cup \dots \cup \Pi_{k_i}$. We also use $\mathcal{Z} = \Pi \times \mathcal{X}$ to denote the feasible region, and $J_t = J(\mathbf{z}_t) = J(\mathbf{P}_t, \mathbf{x}_t)$ in the form of Objective (3.6) to represent the energy value at the i^{th} step. Accordingly, the sequences of updated z and corresponding energy value J are written as $\{\mathbf{z}_t : t \in \mathbb{N}\}$ and $\{J_t : t \in \mathbb{N}\}$, respectively. Table 3.1 provides a summary for reference.

Global convergence

We establish the proof of the global convergence of Algorithm 4 upon the famous Zangwill's convergence theorem [162], which is defined as:

Theorem 1 (Global convergence theorem). *Let $\mathcal{H}$ be a set-valued mapping on $\mathcal{S}$ that, given $s_1 \in \mathcal{S}$, generates a sequence $\{s_k\}_k^\infty$ via iterating $s_{k+1} \in \mathcal{H}(s_k)$. Also let a solution set $\Gamma \subset \mathcal{S}$ be given. Suppose that*

1. *The sequence $\{s_k\}_k^\infty \subset \mathcal{S}'$ for $\mathcal{S}' \subset \mathcal{S}$ is compact.*
2. *The set-valued mapping $\mathcal{H}$ is closed at all $s \in \mathcal{S} \setminus \Gamma$.*
3. *There exists a continuous function Ψ on $\mathcal{S}$ that*

Table 3.1 Notations in Chapter 3.4.2.

Notation	Meaning
Π	The set of all generalized permutation matrices with all possible ranks.
$\mathcal{X}$	The space of the regression variable.
$\mathbf{P}$	The generalized permutation matrix.
$\mathbf{x}$	The regression variable.
$\mathcal{Z}$	$\mathcal{Z} = \Pi \times \mathcal{X}$, the feasible space.
$\mathbf{z}$	$\mathbf{z} = (\mathbf{P}, \mathbf{x})$, a point within the feasible space.
J	$J = J(\mathbf{z}) = J(\mathbf{P}, \mathbf{x})$, the GSLR objective function shown in Eqs 3.6 and 3.16.
J_t	Value of J at the t^{th} iteration of Algorithm 4, similar notation also applies to $\mathbf{z}$.
$\{\mathbf{z}_t : t \in \mathbb{N}\}$	The entire sequence of $\mathbf{z}$ generated by Algorithm 4 until convergence.

- if $s \notin \Gamma$: $\Psi(s') < \Psi(s)$, $\forall s' \in \mathcal{H}(s)$.
- if $s \in \Gamma$: $\Psi(s') \leq \Psi(s)$, $\forall s' \in \mathcal{H}(s)$.

Then the limit of any convergent subsequence of $\{s_k\}_k^\infty$ is in the solution set Γ . Furthermore, $\lim_{k \rightarrow \infty} \Psi(s_k) = \Psi(s^*)$ holds for all limit points s^* .

In the terminology of Theorem 1, an iterative algorithm can be described as a triplet $\{\Psi, \mathcal{H}, \Gamma\}$. For GSLR, we can straightforwardly set Ψ as Objective (3.6), and $\mathcal{H} : \mathcal{Z} \rightarrow \mathcal{Z}$ as one iteration of Algorithm 4. Also, we can formulate Γ with all the critical points:

$$\Gamma = \{(\mathbf{P}, \mathbf{x}) \in \mathcal{Z} : \mathbf{P} \in \underset{\mathbf{P}'}{\operatorname{argmin}} \Psi(\mathbf{P}', \mathbf{x}) \text{ and } \mathbf{x} \in \underset{\mathbf{x}'}{\operatorname{argmin}} \Psi(\mathbf{P}, \mathbf{x}')\}.$$

We claim that all the three conditions in Theorem 1 are satisfied by the GSLR triplet mentioned above, and thus it is globally convergent to a critical point. The detailed proofs are presented in Chapter 3.4.3.

Convergence rate

Kurdyka-Łojasiewicz inequality (KL-inequality) is a powerful tool for studying the convergence rates of non-convex optimization algorithms. When applied to alternative optimization, existing works [155, 7, 15, 154] commonly change the standard update rule to a proximal one to facilitate the analyses. In contrast to these works, to avoid complicating the easy-to-solve

LAP in Algorithm 4 to an NP-hard QAP, we stick to using the standard update rule and seek benefits purely from the properties of GSLR itself.

We present the following two propositions, which serve as the primary components to apply the KL-inequality-based theorem of convergence rate. Their proofs can be found in Chapter 3.4.3.

Proposition 2 (Properties of the GSLR problem). *Under Hypotheses H, the following properties of the GSLR objective function J hold true:*

1. *The gradient of J w.r.t. $\mathbf{x}$ (hereafter denoted as $\nabla_{\mathbf{x}}J$) is Lipschitz continuous.*
2. *J satisfies the KL-inequality w.r.t. $\mathcal{Z}$.*

Proposition 3 (Bounded solution sequence). *Under Proposition 2 and the global convergence described in Chapter 3.4.2, $\forall p \geq 1$, the following equation holds true:*

$$\sum_{t=p}^{+\infty} \|\mathbf{z}_{t+1} - \mathbf{z}_t\| < +\infty.$$

Given the two propositions mentioned above, we are ready to introduce the following theorem of the convergence rate. This theorem is already well-studied by various works[155, 6, 7, 154], and we refer interested readers to the details therein.

Theorem 2 (Convergence rate based on KL-inequality). *Given an objective function satisfying the KL-inequality, and a solution sequence of it satisfying Proposition 3, then the following statements in terms of the Łojasiewicz exponent θ hold true:*

1. *If $\theta = 0$, $\{\mathbf{z}_t : t \in \mathbb{N}\}$ converges in finite steps.*
2. *If $\theta \in (0, \frac{1}{2}]$, $\{\mathbf{z}_t : t \in \mathbb{N}\}$ converges linearly.*
3. *if $\theta \in (\frac{1}{2}, 1)$, $\{\mathbf{z}_t : t \in \mathbb{N}\}$ converges sub-linearly.*

According to Propositions 2 and 3 and Theorem 2, given the fact that θ may vary from case to case, we can conclude that Algorithm 4 is sub-linearly convergent in general.

3.4.3 Proofs of corollaries, propositions, and lemmas in Chapter 3.4.2

In this section, we present proofs of the corollaries, propositions, and lemmas raised in the convergence analysis mentioned above.

Proofs of the global convergence

We here show that the GSLR triplet mentioned in Chapter 3.4.2 satisfies all the conditions of Theorem 1. For self-containment, we first remark the definition of closedness of set-valued mapping:

Definition 1 (Closedness of set-valued mapping [51]). *The concept of closedness of set-valued mapping $\mathcal{F} : \mathcal{U} \rightarrow \mathcal{V}$ generalizes the notion of continuity of point-to-point mappings: $\mathcal{F}$ is closed at a point $u \in \mathcal{U}$ if*

- $u_k \rightarrow u$, where $u_k \in \mathcal{U}$,
- $v_k \rightarrow v$, where $v, v_k \in \mathcal{V}$,
- $v_k \in \mathcal{F}(u_k)$,

imply $v \in \mathcal{F}(u)$. Based on this, $\mathcal{F}$ is called closed if it is closed at any points of $\mathcal{U}$.

Now we are ready to introduce the proofs:

Proof of the compactness of $\mathcal{Z}$. Since Π is discrete and finite, Π is compact. Since $\mathcal{X}$ is compact per Hypotheses H and unions of compact sets are still compact, $\mathcal{Z}$ is compact. $\square$

Proof of the closedness of $\mathcal{H}$. For any $(\mathbf{P}_t, \mathbf{x}_t)$, there exist some $\delta_1, \delta_2 \in \mathbb{R}^+$ that satisfy

$$\begin{aligned} \forall \mathbf{P}', \mathbf{P}' \rightarrow \mathbf{P} &:= \{\mathbf{P}' : \mathbf{P}' \in \mathcal{N}_{\delta_1}(\mathbf{P}_t)\} = \{\mathbf{P}_t\}, \\ \forall \mathbf{x}', \mathbf{x}' \rightarrow \mathbf{x} &:= |\mathbf{x}', \mathbf{x}_t|_{\mathbf{x}} < \delta_2, \end{aligned} \quad (3.19)$$

where $\mathcal{N}_{\delta_1}(\mathbf{P}_t)$ is the neighborhood of $\mathbf{P}_t$. Therefore, we can derive the differences of mapped $\mathbf{x}$ as

$$\begin{aligned} \mathcal{H}(\mathbf{P}', \mathbf{x}') \setminus \mathcal{H}(\mathbf{P}_t, \mathbf{x}_t) &= \mathcal{H}(\mathbf{P}_t, \mathbf{x}_t) \setminus \mathcal{H}(\mathbf{P}', \mathbf{x}') \\ &= \mathcal{H}(\mathbf{P}_t, \mathbf{x}') \setminus \mathcal{H}(\mathbf{P}_t, \mathbf{x}_t) \\ &= \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \Psi(\mathbf{P}_t, \mathbf{x}) \setminus \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \Psi(\mathbf{P}_t, \mathbf{x}) = \emptyset. \end{aligned} \quad (3.20)$$

Since Eq. (3.20) implies that the cost matrix for the k -LAP generated by the regression variable $\mathbf{x} \in \mathcal{H}(\mathbf{P}, \mathbf{x})$ remains unchanged when the input of $\mathcal{H}$ switches from $(\mathbf{P}_t, \mathbf{x}_t)$ to $(\mathbf{P}', \mathbf{x}')$, so will the mapped permutation $\mathbf{P}$. Therefore, we have $\mathcal{H}(\mathbf{P}_t, \mathbf{x}_t)$ closed by further joining Eqs. (3.19) and (3.20). $\square$

Proof of the continuity and monotonicity of Ψ . For the continuity, since compositions of continuous mappings are still continuous, we only need to prove the continuity of $f(\mathbf{P}, \mathbf{x}) = \mathbf{P}\mathbf{A}\mathbf{x}$ and $g(\mathbf{P}) = \mathbf{P}\mathbf{P}^T\mathbf{b}$. Since the proof roadmap for f and g is similar, we only take f as an example. Specifically, given the fact that such a function is a bilinear form over the finite-dimensional space $(\mathbf{P}, \mathbf{x})$, we only need to show that it is separately continuous w.r.t. both $\mathbf{P}$ and $\mathbf{x}$. For proof, per Hypotheses H, we have $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ continuous w.r.t. $\mathbf{x} \in \mathcal{X}$. Since Π is finitely discrete, we also have $f(\mathbf{P}) = \mathbf{P}\mathbf{A}$ continuous w.r.t. any $\mathbf{P} \in \Pi$. Therefore, we can obtain that $f(\mathbf{P}, \mathbf{x}) = \mathbf{P}\mathbf{A}\mathbf{x}$ is continuous and hence so is Ψ .

For the monotonicity, by conducting the linear regression step w.r.t. $\mathbf{x}$ we have

$$J(\mathbf{P}_t, \mathbf{x}_{t+1}) \leq J(\mathbf{P}_t, \mathbf{x}_t), \quad (3.21)$$

where the equality holds if and only if $\mathbf{x}_t$ is a critical point. Furthermore, assuming there is a temporary permutation matrix $\mathbf{P}_{\text{tmp}}$ that randomly removes $k_{t+1} - k_t$ non-zero assignments from $\mathbf{P}_t$, since the cost matrix $\mathbf{D}_{ij} = (\mathbf{A}_i\mathbf{x} - \mathbf{b}_j)^2$ is non-negative, we can obtain that

$$J(\mathbf{P}_{\text{tmp}}, \mathbf{x}_{t+1}) \leq J(\mathbf{P}_t, \mathbf{x}_{t+1}). \quad (3.22)$$

Since $\mathbf{P}_{t+1}$ is a global optimum of the objective function w.r.t. $\mathbf{P}$, and it preserves the same number of inliers to $\mathbf{P}_{\text{tmp}}$, we can further derive that

$$J(\mathbf{P}_{t+1}, \mathbf{x}_{t+1}) \leq J(\mathbf{P}_{\text{tmp}}, \mathbf{x}_{t+1}), \quad (3.23)$$

where the equality holds if and only if $\mathbf{P}_{\text{tmp}}$ is a critical point. Summarizing Eqs. (3.21)-(3.23) leads to:

$$J(\mathbf{P}_{t+1}, \mathbf{x}_{t+1}) \leq J(\mathbf{P}_{\text{tmp}}, \mathbf{x}_{t+1}) \leq J(\mathbf{P}_t, \mathbf{x}_t). \quad (3.24)$$

Therefore, we can conclude that Algorithm 4 is monotonically decreasing and the equality in Eq. (3.24) holds if and only if $(\mathbf{P}_t, \mathbf{x}_t)$ is already a critical point. $\square$

Proofs of the convergence rate

We here present the proofs of Propositions 2 and 3. Throughout the proofs provided hereafter, apart from the notations described in Table 3.1, we also use C to denote a constant whose value may vary from line to line.

To prove Proposition 2, due to the non-smoothness of our Objective (3.6), we following the concept of the KL-inequality generalized to sub-analytic set [14]. Let us first remark the definitions for self-containment:

Definition 2 (Kurdyka-Łojasiewicz inequality). *Let f be a continuous sub-analytic function (i.e., a function whose graph is a sub-analytic set) with closed domain that maps to $\mathbb{R}$, and $\bar{x} \in \text{dom}(f)$ be a critical point, there exists a concave function $\phi(s) = Cs^{1-\theta}$ with $\theta \in [0, 1)$ and $C > 0$ such that the following inequality holds for any x within the neighborhood of $\bar{x}$ under the conventions $0^0 = 1$ and $0/0 = \infty/\infty = 0$:*

$$\phi'(f(x) - f(\bar{x})) \text{dis}(0, \partial f(x)) \geq 1, \quad (3.25)$$

where $\text{dis}(0, \partial f(x)) = \inf\{\|x^*\| : x^* \in \partial f(x)\}$ denotes the point-to-set distance.

Definition 3 (Semi-X & sub-analytic sets). *A subset $\mathcal{S}$ of $\mathbb{R}^d$ is called semi-algebraic if every $\mathbf{x} \in \mathbb{R}^d$ admits a neighborhood $\mathcal{N}$ such that*

$$\mathcal{S} \cap \mathcal{N} = \bigcap_i^m \bigcup_j^n \{\mathbf{x} \in \mathcal{N} : f_{ij}(\mathbf{x}) = 0, g_{ij}(\mathbf{x}) \geq 0\},$$

where f_{ij} and g_{ij} are polynomials and $m, n < +\infty$.

A semi-analytic set generalizes the above definition by relaxing f_{ij} and g_{ij} to real analytic functions.

A subset $\mathcal{S}$ of $\mathbb{R}^d$ is called sub-analytic if every $\mathbf{x} \in \mathbb{R}^d$ admits a neighborhood $\mathcal{N}$, and there exists a bounded semi-analytic set $\mathcal{Y} \subset \mathbb{R}^{d+o}$ such that $\mathcal{S} \cap \mathcal{N}$ is the projection from $\mathcal{Y}$ to $\mathcal{N}$:

$$\mathcal{S} \cap \mathcal{N} = \pi(\mathcal{Y}),$$

where $\pi : \mathbb{R}^{d+o} \rightarrow \mathbb{R}^d$ is the projection. In summary, sub-analytic set broadens the semi-analytic one, and the semi-analytic set broadens the semi-algebraic one.

We also introduce the following lemma:

Lemma 4. *The set of generalized permutation matrices Π can be characterized by the following polynomials:*

$$\begin{aligned} \mathbf{P}\mathbf{1} &\leq \mathbf{1} \text{ and } \mathbf{P}^T\mathbf{1} \leq \mathbf{1}, \\ \mathbf{e}_i^T \mathbf{D} \mathbf{e}_i (\mathbf{e}_i^T \mathbf{D} \mathbf{e}_i - 1) &= 0, \forall i \in [n], \\ \mathbf{e}_i^T \mathbf{D} \mathbf{e}_j &= 0, \forall i, j \in [n], i \neq j, \end{aligned}$$

where $\mathbf{e}_i$ is the one-hot vector with 1 at position i , $\mathbf{D} = \{\mathbf{P}\mathbf{P}^T, \mathbf{P}^T\mathbf{P}\}$, and $[n]$ is the set of positive integers that are no greater than the size of $\mathbf{D}$.

Proof of Lemma 4. This lemma is an extension of the properties of the full-rank permutation matrices, which states that a matrix is permutational if and only if it is orthogonal and doubly

stochastic [42]. The latter two equations imply that the diagonal entries of $\mathbf{D}$ should be within $\{0, 1\}$ and the off-diagonal ones 0. While the sufficiency is guaranteed by the definition of generalized permutation matrices, we here demonstrate that these polynomials are necessary for a matrix to be permutational by forcing $\mathbf{P}$ to be binary. Specifically, assuming there is a matrix $\mathbf{P} \in \Pi$ with entries $\mathbf{P}_{ip}, \dots, \mathbf{P}_{iq} \in (0, 1)$ with $1 \leq p < q \leq n$ in the i^{th} row, then the pq^{th} entry of $\mathbf{D} = \mathbf{P}^T \mathbf{P}$ can be denoted as

$$\mathbf{D}_{pq} = \sum_k \mathbf{P}_{pk}^T \mathbf{P}_{kq} = \sum_k \mathbf{P}_{kp} \mathbf{P}_{kq} \neq 0, \quad (3.26)$$

which contradicts the requirement that the off-diagonal entries of $\mathbf{D}$ should be 0. $\mathbf{D} = \mathbf{P}\mathbf{P}^T$ follows the same proof. $\square$

Now we are ready to prove Proposition 2:

Proof of Proposition 2. The first property is a direct result of the fact that

$$\nabla_{\mathbf{x}} J = 2 (\mathbf{A}^T \mathbf{P}^T \mathbf{P} \mathbf{A} \mathbf{x} - \mathbf{P} \mathbf{P}^T \mathbf{b})$$

is a continuous linear mapping.

For the second property, from Lemma 4 we know that Π is semi-algebraic (hence sub-analytic). Moreover, given the facts that $\mathcal{X}$ is sub-analytic per the assumption, Objective (3.6) is polynomial, and unions and intersections preserve sub-analyticity per Definition 3; we have the graph of Objective (3.6) sub-analytic. Joining the sub-analyticity together with the continuity mentioned in Chapter 3.4.2 implies the KL-inequality. $\square$

To prove Proposition 3, we introduce the following two lemmas:

Lemma 5. *Suppose that Hypotheses H are satisfied, then for any t^{th} step of Algorithm 4, the following equation holds true:*

$$(\nabla_{\mathbf{x}} J(\mathbf{P}_t, \mathbf{x}_t) - \nabla_{\mathbf{x}} J(\mathbf{P}_{t-1}, \mathbf{x}_t), \mathbf{0}) \in \partial J(\mathbf{P}_t, \mathbf{x}_t).$$

Proof of Lemma 5. Per the alternative update rule of Algorithm 4 we have

$$\mathbf{0} = \nabla_{\mathbf{x}} J(\mathbf{P}_{t-1}, \mathbf{x}_t), \quad (3.27)$$

$$\mathbf{0} \in \partial_{\mathbf{P}} J(\mathbf{P}_t, \mathbf{x}_t). \quad (3.28)$$

From Eq. (3.27) we can obtain

$$\nabla_{\mathbf{x}} J(\mathbf{P}_t, \mathbf{x}_t) - \nabla_{\mathbf{x}} J(\mathbf{P}_{t-1}, \mathbf{x}_t) \in \partial_{\mathbf{x}} J(\mathbf{P}_t, \mathbf{x}_t), \quad (3.29)$$

which together with Eq. (3.28) concludes the proof. $\square$

Lemma 6. *Suppose that Hypotheses H are satisfied, then for any t^{th} step of Algorithm 4, there exists $C \in \mathbb{R}^+$ such that the following equation holds true:*

$$J_t - J_{t+1} \geq C \|\mathbf{z}_{t+1} - \mathbf{z}_t\|^2.$$

Proof of Lemma 6. We base the proof on the quotient law of convergent sequences. In detail, let us define the sequences:

1. Sequence of difference between energy values: $P = \{J_t - J_{t+1} : t \in \mathbb{N}\}$.
2. Sequence of squared difference between solutions: $Q = \{\|\mathbf{z}_t - \mathbf{z}_{t+1}\|^2 : t \in \mathbb{N}\}$.
3. Sequence of quotient: $S = \left\{\frac{P_t}{Q_t} : t \in \mathbb{N}\right\}$.

Per the compactness of $\mathbf{z}$ and continuity of J , we know that both $\mathbf{z}$ and J are bounded. Moreover, per the global convergences mentioned in Chapter 3.4.2, for any $t \in \mathbb{N}$ we have $P_t, Q_t \geq 0$ and $P_t = 0 \Leftrightarrow Q_t = 0$. Therefore, with the convention $\frac{0}{0} = 1$, for any $s \in S$ we have $s \in \mathbb{R}_+$, which implies that there always exist some $C \in \mathbb{R}_+$ such that $\inf S \geq C$ holds. $\square$

With Lemmas 5 and 6 in hand, we now can prove Proposition 3:

Proof of Proposition 3. According to Lemma 5 and the Lipschitz continuity of $\nabla_{\mathbf{x}}J$, we have

$$\begin{aligned} \text{dis}(\mathbf{0}, \partial J(\mathbf{z}_t)) &\leq \|\nabla_{\mathbf{x}}J(\mathbf{P}_{t-1}, \mathbf{x}_t) - \nabla_{\mathbf{x}}J(\mathbf{P}_t, \mathbf{x}_t)\| \\ &\leq L \|\mathbf{P}_{t-1} - \mathbf{P}_t\| \leq L \|\mathbf{z}_{t-1} - \mathbf{z}_t\|, \end{aligned} \quad (3.30)$$

where $L \in \mathbb{R}^+$ is the Lipschitz constant.

Assuming the energy value $\bar{J}$ at the critical point satisfies $\bar{J} = 0$ without loss of generality, combining Eq. (3.30) and Eq. (3.25) leads to

$$\phi'(J_t) = \phi'(J_t - \bar{J}) \geq \frac{1}{L} \|\mathbf{z}_{t-1} - \mathbf{z}_t\|^{-1}. \quad (3.31)$$

On the other hand, by using the concavity of ϕ and Lemma 6, we can obtain

$$\begin{aligned} \phi(J_t) - \phi(J_{t+1}) &\geq \phi'(J_t) (J_t - J_{t+1}) \\ &\geq C \phi'(J_t) \|\mathbf{z}_{t+1} - \mathbf{z}_t\|^2. \end{aligned} \quad (3.32)$$

Taking Eq. (3.31) into Eq. (3.32) yields

$$\|\mathbf{z}_{t-1} - \mathbf{z}_t\| \cdot C (\phi(J_t) - \phi(J_{t+1})) \geq \|\mathbf{z}_{t+1} - \mathbf{z}_t\|^2. \quad (3.33)$$

Taking square root on both sides of Eq. (3.33) and applying the inequality $a^2 + b^2 \geq 2ab$ result in

$$\|\mathbf{z}_{t-1} - \mathbf{z}_t\| + C(\phi(J_t) - \phi(J_{t+1})) \geq 2\|\mathbf{z}_{t+1} - \mathbf{z}_t\|. \quad (3.34)$$

Summing Eq. (3.34) on both sides for $t \in [p, +\infty)$ leads to

$$\sum_{t=p}^{+\infty} \|\mathbf{z}_{t+1} - \mathbf{z}_t\| \leq \|\mathbf{z}_p - \mathbf{z}_{p-1}\| + C\phi(J_p), \quad (3.35)$$

Since $\mathcal{Z}$ is compact as mentioned in Chapter 3.4.2, there exists a B such that $\sup\{\|\mathbf{z}_i - \mathbf{z}_j\| : \mathbf{z}_i, \mathbf{z}_j \in \mathcal{Z}\} = B < +\infty$. Hence Eq. (3.35) yields

$$\sum_{t=p}^{+\infty} \|\mathbf{z}_{t+1} - \mathbf{z}_t\| \leq C\phi(J_p) + B < +\infty, \quad (3.36)$$

which concludes the proof. $\square$

3.5 Experiments

We first compare GSLR to some existing SLR algorithms using the image alignment task. Then, we explore how the GSLR formulation together with Algorithm 4 can benefit point cloud alignment and isometric shape matching. Finally, other experiments w.r.t. the properties of GSLR is presented. As noted in Remark 3, all the problems mentioned hereafter satisfy Hypotheses H, and thus are subject to the analyses presented above. For all the experiments, we set the threshold mentioned in Eq. (3.18) to 3.5 as suggested in [56] unless explicitly noted otherwise.

3.5.1 Comparing SLR to GSLR

Image alignment is an excellent example to demonstrate the differences between SLR and GSLR since even a small portion of mismatches may extremely distort the results. Given a pair of images portraying the same plane, or a non-planar scene subject to $\mathbf{0}$ translation in terms of the camera pose, homography describes the relation of $\mathbf{p}' = \alpha \mathbf{H} \mathbf{p}$, where $\mathbf{p}'$ and $\mathbf{p}$ are matched pixels on the two images in homogeneous coordinates, α is a scale factor, and $\mathbf{H}$ is the homography matrix. Furthermore, if images are captured with long focal lengths, $\mathbf{H}$ can be better modeled by affine transformations [57]. Since such a modeling implies that the

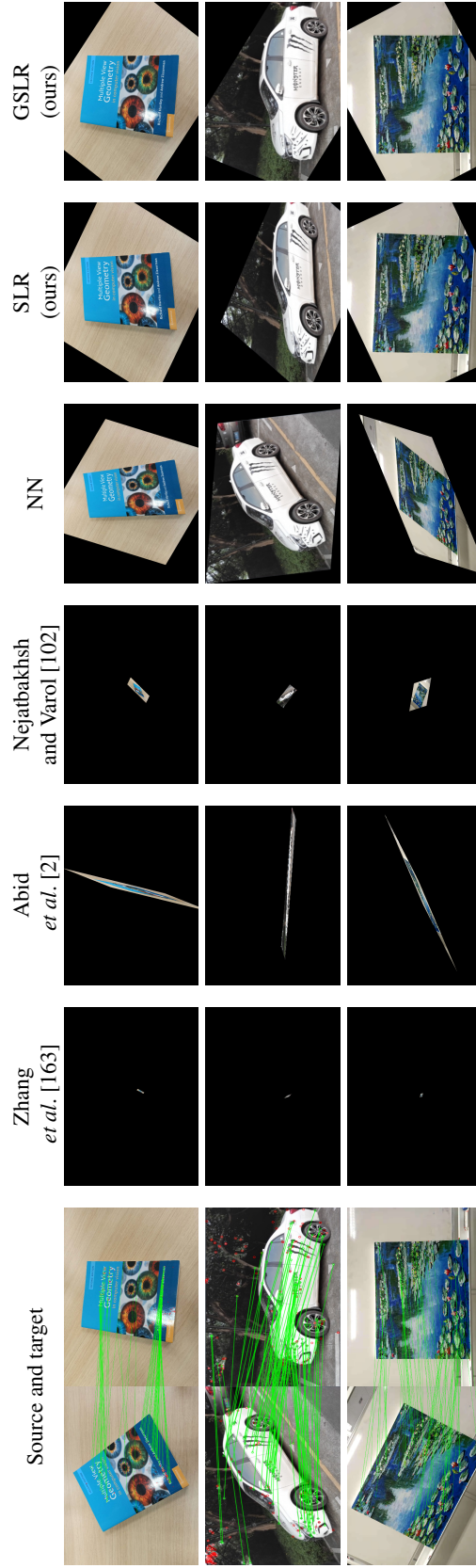


Fig. 3.3 Examples of the image alignment experiment. On the source and target image pairs, outliers detected by GSLR are marked as red and inliers as green. NN denotes the ICP-style algorithm where the correspondences are estimated by nearest-neighbor search, and SLR (ours) is Algorithm 4 with the number of inliers k fixed to $\min(m, n)$. The descriptor matching-based method [18] are omitted as its results are visually the same to the target images.

Table 3.2 Mean and median absolute reprojection errors of image alignment (in pixel unit). The method [18], which employs known correspondences as a prior, is used for reference.

	Zhang <i>et al.</i> [163]	Abid <i>et al.</i> [2]	Nejatbakhsh and Varol [102]	NN	SLR (ours)	GSLR (ours)	Cai <i>et al.</i> [18]
Mean	160.82	122.03	160.54	98.42	36.56	14.16	6.42
Median	182.84	116.71	181.89	99.33	32.93	4.30	3.16

scaling $\alpha = 1$, we can formulate the problem as

$$\min_{\mathbf{H}, \mathbf{P} \in \Pi} \|\mathbf{PAH}^T - \mathbf{PP}^T \mathbf{B}\|^2, \quad (3.37)$$

where $\mathbf{A}$ and $\mathbf{B}$ stand for the collections of pixels from the two images in homogeneous coordinates.

We collect 30 pairs of images and extract 500 SIFT feature points [85] from each image to formulate the data $\mathbf{A}$ and $\mathbf{B}$. All the algorithms are initialized with $\mathbf{H}$ being the identity matrix and iterated 100 times except the RANSAC-based one [102], which is looped 1000 times for better sampling. We evaluate the results with the absolute reprojection error:

$$\text{err} = \frac{\|(\mathbf{A}_{\text{in}} \mathbf{H}_{\text{esti}}^T - \mathbf{B}_{\text{in}}) \mathbf{M}\|_1}{2N_{\text{in}}}, \quad \mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T,$$

where $\|\cdot\|_1$ denotes the l_1 -norm, $\mathbf{A}_{\text{in}}$ and $\mathbf{B}_{\text{in}}$ are the ground-truth inlier matchings sieved from $\mathbf{A}$ and $\mathbf{B}$ using the mature descriptor-matching, ratio test, and RANSAC homography estimation pipeline [57], and N_{in} is the cardinality of $\mathbf{A}_{\text{in}}$.

The results are summarized in Table 3.2. All methods are correspondence-free except the one proposed by Cai *et al.* [18], which takes in known correspondences as a prior, and hence used as a reference. The Monte Carlo-based method [2] and the RANSAC-based one [102] fail to present reasonable estimations since both of their sample spaces are too large to achieve sufficient sampling. The QAP solver proposed in [163] also cannot perform well due to the enlarged dimension of the solution space (*i.e.*, due to its ADMM-based solver, replacing the homography matrix by a permutation raises the solution space from $(\mathbb{R}^{3 \times 3}, \mathbb{R}^{500 \times 500})$ to $(\mathbb{R}^{500 \times 500}, \mathbb{R}^{500 \times 500})$). For the NN-based method and the SLR version of Algorithm 4, compared to GSLR, their errors are still large, as outliers are not rejected. Examples of the warped images are illustrated in Fig. 3.3.

3.5.2 GSLR in point cloud alignment

Point cloud alignment aims at estimating the rigid transformation $\mathbf{T}$ between two point clouds under unknown correspondences. Its objective function is

$$\min_{\mathbf{T} \in \text{SE}(n), \mathbf{P} \in \Pi} \|\mathbf{P}\mathbf{A}\mathbf{T}^T - \mathbf{P}\mathbf{P}^T\mathbf{B}\|^2, \quad (3.38)$$

where $\mathbf{A}$ and $\mathbf{B}$ represent the two point clouds in homogeneous coordinates. Usually, both of them contain outliers that should be rejected for estimating $\mathbf{T}$.

Peer methods and experimental setups

We use ICP [5], Trim-ICP [25], SVR [19], CPD [100], FGR [169], TEASER++ [158], FilterReg [44], DO [145], and PointNetLK [4] for comparison¹. Among them, ICP works as the basic baseline, Trim-ICP shares a similar robust estimation technique with GSLR but employs the NN-based matching strategy, SVR uses the robust L_2 -divergence to formulate the objective function; FGR and TEASER++ are feature-based alignment methods that respectively use the scaled Geman-McClure estimator and truncated least square for robust estimation; CPD and FilterReg use a uniform distribution to model outliers; and DO and PointNetLK solve the alignment problem in self-supervised and supervised manners, respectively. For the hyper-parameters of the competing approaches, we equip both FGR and TEASER++ with the FPFH feature [122]. We also set the trimming ratio of Trim-ICP, as well as the weights of the uniform distributions in CPD and FilterReg to 0.5, representing a breakdown point of 0.5. For DO, we down-sample each point cloud to approximately 600 points for a tractable elapsed time, set the number of update maps to 50, and generate 10000 training data for each trial with 50% of points being outliers. For PointNetLK, since all the datasets used hereafter do not provide data for training, we employ the model pre-trained on the ModelNet40 dataset [151] in the tests. All the methods are initialized with an identity transformation matrix if required.

We conduct experiments on both synthetic and real-world datasets. All pairs of point clouds are aligned bidirectionally. We use the metrics $\|\mathbf{R}_{\text{gt}}\mathbf{R}_{\text{esti}}^{-1} - \mathbf{I}\|$, $\|\mathbf{t}_{\text{gt}} - \mathbf{t}_{\text{esti}}\|$, and $\|\mathbf{T}_{\text{gt}}\mathbf{T}_{\text{esti}}^{-1} - \mathbf{I}\|$ to assess the rotation $\mathbf{R}$, the translation $\mathbf{t}$, and the overall transformation $\mathbf{T}$, respectively.

¹Some implementations are from ProbReg: <http://probreg.readthedocs.io/>, last accessed on Friday 30th December, 2022

	Outlier-contaminated						Partially-overlapped					
	Mean			Median			Mean			Median		
	R	t	T	R	t	T	R	t	T	R	t	T
ICP	1.05	4.80e-3	1.05	1.04	4.67e-3	1.04	1.69e-1	1.04e-2	1.70e-1	1.19e-1	8.43e-3	1.19e-1
Trim-ICP	1.07	3.69e-3	1.07	1.09	3.49e-3	1.09	8.56e-1	3.81e-2	8.61e-1	8.54e-1	4.83e-2	8.56e-1
SVR	1.05	3.00e-3	1.05	0.99	2.67e-3	1.00	2.15e-1	7.14e-3	2.16e-1	1.71e-1	4.94e-3	1.71e-1
CPD	1.58e-2	4.22e-4	1.58e-2	1.52e-2	4.00e-4	1.52e-2	8.99e-2	5.66e-3	9.03e-2	9.06e-2	3.75e-3	9.09e-2
FilterReg	2.19e-2	4.84e-4	2.20e-2	1.69e-2	4.99e-4	1.70e-2	1.11e-1	6.22e-3	1.12e-1	9.88e-2	3.89e-3	9.95e-2
FGR	8.53e-2	1.57e-3	8.55e-2	7.21e-2	1.10e-3	7.22e-2	1.07	3.74e-2	1.07	1.12	3.18e-2	1.12
TEASER++	5.39e-2	1.53e-3	5.40e-2	5.49e-2	1.04e-3	5.50e-2	1.65	7.29e-2	1.66	1.94	5.14e-2	1.95
DO	4.13e-1	6.82e-3	4.14e-1	2.98e-1	7.06e-3	2.99e-1	8.69e-2	8.29e-2	1.31e-1	6.83e-2	1.08e-1	1.27e-1
PointNetLK	1.12	9.05e-3	1.12	1.20	8.23e-3	1.21	3.81e-1	1.91e-2	3.83e-1	3.53e-1	1.52e-2	3.54e-1
GSLR (ours)	6.22e-3	1.63e-4	6.24e-3	5.64e-3	1.18e-4	5.65e-3	6.49e-2	4.01e-3	6.52e-2	3.76e-2	1.93e-3	3.78e-2

Table 3.3 Alignment results on the synthetic data.

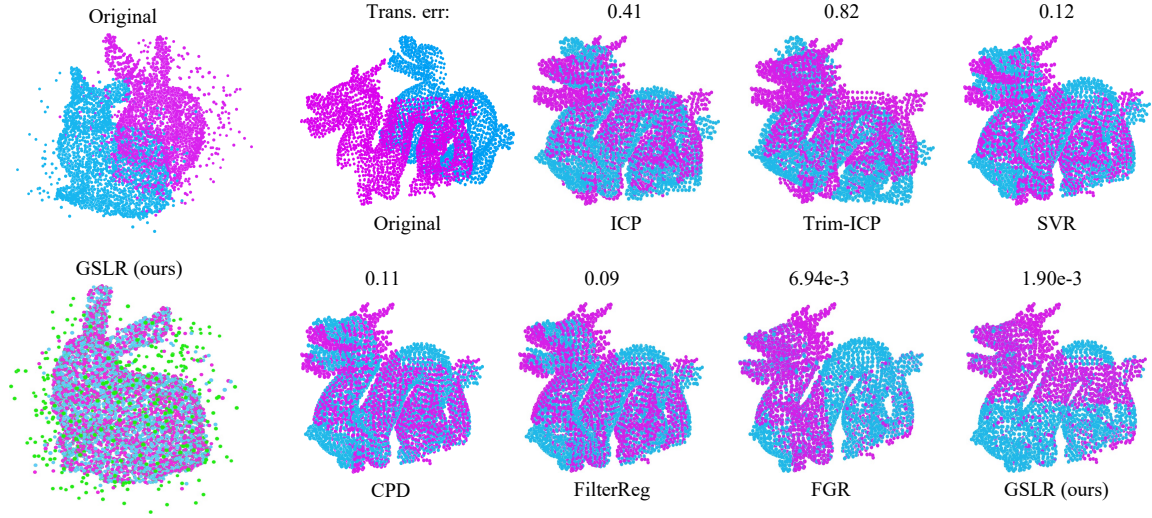


Fig. 3.4 Example alignment results on the synthetic data. Left: Aligning outlier-contaminated point clouds using our GSLR. The identified outliers are marked as green. Right: Aligning the partially overlapped data. TEASER++, DO, and PointNetLK show either similar or worse visual results than the illustrated ones and thus omitted.

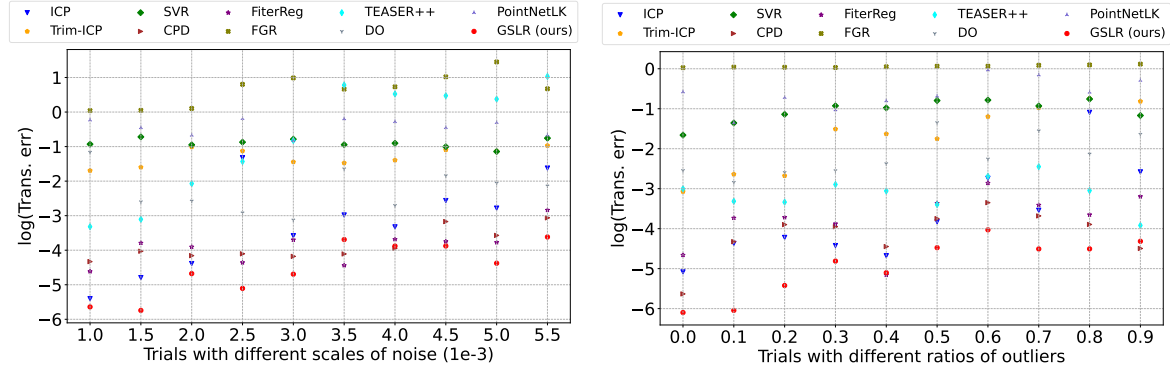


Fig. 3.5 Transformational errors w.r.t. different scales of isotropic Gaussian noise (left) and different ratios of outliers (right).

Alignment with noises, outliers, and partial overlaps

We use the synthetic point clouds of bunny, dragon, and armadillo [31] for tests. For the outlier-contaminated cases, we first down-sample the point clouds to approximately 2500 points, scale them to fit in the sphere with radius around 0.1, and corrupt them with isotropic Gaussian noises with covariance scales $1.5e-3$ and $2.5e-3$. We then insert outliers within the enclosing sphere with ratios sampled from $(0.25, 1)$, representing that 20% to 50% of points are outliers. The above procedures are repeated 10 times independently on each shape to obtain 30 trials in total. For the partial-overlapping setup, we down-sample the point clouds to around 3000 points each, randomly crop some parts from them, and add in

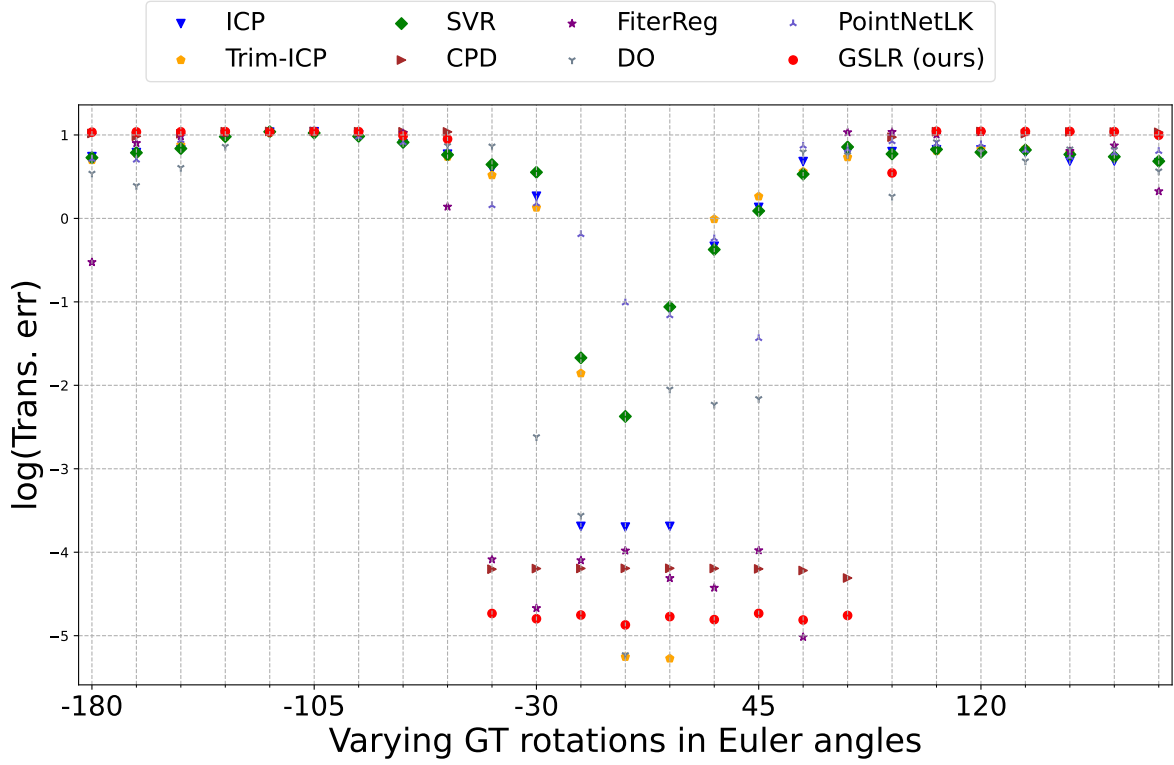


Fig. 3.6 Transformational errors w.r.t. different initializations. FGR and TEASER++ are omitted as they are global alignment methods.

isotropic Gaussian noises with covariance scales $1e-3$, $1.5e-3$, $2e-3$, and $2.5e-3$. We retrieve 4 cropped point clouds from each shape, leading to 36 trials in total. For all the tests, we set the ground-truth rotation to 30° in Euler angles together with a random translation within $[0, 0.1]$ along each axis. Table 3.3 records the results of aligning outlier-contaminated and partial-overlapping point clouds, and some examples are presented in Fig. 3.4. SVR fails to provide accurate results, as dense clusters of outliers are often similarly weighted as inliers. FGR and TEASER++ highly rely on the descriptiveness of the features, and hence are unstable compared to the others. This inconsistency can be seen from the good result in Fig. 3.4 (b) and the large overall errors on all trials. DO and PointNetLK cannot stably present plausible results due to the gaps between their training and test data. Moreover, compared to CPD, SVR, and FilterReg, which can only down-weight the outliers, GSLR presents more accurate results by completely rejecting them. We also quantify how different scales of noise and ratios of outliers can affect the alignment accuracies. For both experiments, we employ the bunny point cloud and fix the ground-truth rotation to 15° in Euler angles to let peer approaches perform reasonably. As shown in Fig. 3.5, GSLR can consistently provide more accurate estimations compared to the other approaches.

Alignment with distinct initializations

We employ the aforementioned bunny point clouds with 1/3 of points as outliers to investigate the sensitivity of Algorithm 4 w.r.t. initializations. The experimental setups are the same to the outlier-rejection experiment mentioned in Chapter 3.5.2 except for the ground-truth rotations, which are set to $[x, x, x/2]$ in Euler angles with x varying from -180° to 180° with a step size of 15° . As shown in Fig. 3.6, GSLR is comparably robust to other initialization-robust methods such as CPD and FilterReg, and can achieve more accurate results given reasonable initializations. Furthermore, given the fact that Trim-ICP is excessively similar to GSLR except for the matching policy, and appears to be sensitive to initializations; we can conclude that the one-to-one matching strategy can effectively alleviate the initialization problem.

Real-world alignment

We examine the performance of GSLR in aligning real-world point clouds. In the first experiment, we employ the ETH Hauptgebaude dataset [110], which is challenging due to the existence of repeated patterns and a minimum overlap ratio of 63.9%, and the Office dataset [135] for tests. For the former dataset, we divide it into 7 groups with 5 consecutive frames in each group, resulting in 140 trials for pairwise alignment. For the latter one, we pairwise align all the scans, leading to 20 trials altogether. In all trials, we fix the ground-truth rotations to 15° in Euler angles and translations within $[0, 1]$ along each axis to let locally-optimal methods present reasonable results, and down-sample the point clouds to approximately 3500 points for all candidate methods. Furthermore, to extensively study GSLR, we add an additional setup by lowering the down-sampling rate to provide ICP, trim-ICP, and FilterReg 40k points, and the feature-based methods of FGR and TEASER++ 60k points (the densities are selected to fit the memory limitation or maintain a reasonable elapsed time).

Table 3.4 records the quantitative results and Fig. 3.7 provides a visualization. GSLR consistently presents the best overall transformational accuracy when candidate methods are inputted the same sparsely down-sampled point clouds. Its mean error even remains to be the smallest on ETH Hauptgebaude when denser point clouds are inputted to the efficient methods. For algorithms with dense inputs, ICP and trim-ICP still cannot achieve the best accuracy due to their sensitiveness to outliers and initializations, respectively; FilterReg becomes unstable, evidenced by its largely increased mean error on ETH Hauptgebaude. While TEASER++ with denser inputs presents the best results on Office, performance of such feature-based methods on ETH Hauptgebaude decreases significantly when the inputs are

	#Points	ETH Hauptgebaude						Office					
		Mean			Median			Mean			Median		
		R	t	T	R	t	T	R	t	T	R	t	T
ICP	~3.5k	5.18e-2	5.64e-1	5.86e-1	1.20e-2	1.64e-1	1.63e-1	1.50e-2	1.83e-1	1.83e-1	9.33e-3	1.53e-1	1.54e-1
Trim-ICP	~3.5k	5.27e-3	2.04e-1	2.05e-1	2.89e-3	1.48e-1	1.48e-1	2.56e-1	5.47e-1	5.92e-1	2.52e-1	5.52e-1	5.46e-1
SVR	~3.5k	6.54e-2	7.46e-1	7.59e-1	6.30e-2	6.58e-1	6.62e-1	1.00e-1	5.88e-1	5.87e-1	1.12e-1	5.95e-1	5.96e-1
CPD	~3.5k	3.22e-3	4.97e-1	4.97e-1	2.65e-3	1.22e-1	1.22e-1	<u>3.22e-3</u>	1.50e-1	1.50e-1	2.59e-3	1.44e-1	1.43e-1
FilterReg	~3.5k	1.02e-2	3.01e-1	3.03e-1	2.26e-3	1.97e-1	2.00e-1	4.00e-3	1.88e-1	1.86e-1	<u>1.66e-3</u>	1.82e-1	1.81e-1
FGR	~3.5k	2.81e-2	6.00e-1	6.05e-1	2.41e-2	4.41e-1	4.46e-1	6.51e-2	4.11e-1	4.04e-1	4.92e-2	2.87e-1	1.68e-1
TEASER++	~3.5k	1.38e-2	4.36e-1	4.41e-1	1.27e-2	1.28e-1	1.33e-1	3.14e-2	1.74e-1	2.68e-1	2.64e-2	1.59e-1	1.58e-1
DO	~3.5k	1.92e-1	16.51	16.5	3.75e-2	16.20	16.23	5.91e-1	6.69	6.75	4.02e-1	6.67	6.67
PointNetLK	~3.5k	3.22e-1	5.06	5.19	1.01e-1	2.01	2.09	1.25	6.94	7.12	4.69e-1	1.49	1.44
ICP	~40k	5.73e-2	7.47e-1	7.70e-1	1.73e-2	1.44e-1	1.53e-1	1.46e-2	9.13e-2	9.27e-2	6.49e-3	6.14e-2	6.05e-2
Trim-ICP	~40k	6.10e-2	6.64e-1	6.76e-1	6.09e-2	6.56e-1	6.81e-1	2.51e-1	5.66e-1	5.99e-1	2.58e-1	6.40e-1	6.56e-1
FilterReg	~40k	4.40e-3	7.85e-1	7.86e-1	3.62e-3	3.85e-2	4.14e-2	2.11e-3	4.87e-2	4.8e-2	1.34e-3	4.68e-2	4.64e-2
FGR	~60k	1.31e-1	1.71	1.75	1.29e-1	1.73	1.75	3.75e-1	1.97	1.92	9.76e-2	9.58e-1	9.52e-1
TEASER++	~60k	8.23e-3	1.27	1.27	7.29e-3	1.29	1.29	4.83e-3	1.78e-2	1.80e-2	4.31e-3	1.68e-2	1.69e-2
GSLR (ours)	~3.5k	7.05e-3	1.38e-1	1.40e-1	4.23e-3	1.12e-1	1.16e-1	7.62e-3	1.43e-1	1.42e-1	2.46e-3	1.36e-1	1.32e-1

Table 3.4 Alignment results on the ETH Hauptgebaude and Office datasets. Minimum errors are marked as bold when methods with denser inputs are included in comparison, and underlined when such methods are excluded.

	IRL			EUR Pallet								
	Mean		Median	Mean		Median						
	R	t	T	R	t	T						
ICP	1.29e-1	1.47e-2	1.32e-1	5.30e-2	1.12e-2	5.43e-2	3.60e-2	1.45e-1	1.58e-1	1.88e-2	9.60e-2	1.01e-1
Trim-ICP	1.26e-1	1.54e-2	1.30e-1	5.78e-2	1.75e-2	6.03e-2	1.66e-1	2.72e-1	3.63e-1	1.76e-1	2.88e-1	3.42e-1
SVR	2.07e-1	2.40e-2	2.10e-1	1.94e-1	2.22e-2	2.00e-1	3.69e-1	7.77e-1	8.78e-1	3.69e-1	8.40e-1	9.18e-1
CPD	1.88e-1	1.47e-2	1.90e-1	4.51e-2	1.26e-2	4.72e-2	1.82e-2	8.77e-2	9.47e-2	1.74e-3	1.81e-2	1.84e-2
FilterReg	1.73e-1	1.24e-2	2.06e-1	4.96e-2	9.15e-3	5.67e-2	1.19e-2	3.12e-2	3.95e-2	2.40e-4	1.56e-3	1.61e-3
FGR	2.60e-1	1.80e-2	2.62e-1	3.69e-1	1.78e-2	3.69e-1	2.14e-1	3.64e-1	4.98e-1	7.01e-2	1.98e-1	2.24e-1
TEASER++	6.81e-1	2.68e-2	6.84e-1	1.12e-1	1.49e-2	1.13e-1	3.92e-1	3.98e-1	6.68e-1	4.22e-3	9.67e-3	1.23e-2
DO	7.90e-1	9.06e-2	2.05	4.08e-1	8.71e-2	4.38e-1	2.03e-1	2.50	3.70	6.28e-2	2.44	2.46
PointNetLK	4.52e-1	2.90e-1	9.34e-1	1.99e-1	2.08e-1	3.72e-1	3.56e-2	1.68e-1	1.82e-1	1.66e-2	7.67e-2	8.33e-2
GSLR (ours)	7.76e-2	1.22e-2	9.26e-2	2.45e-2	1.42e-2	3.37e-2	1.16e-2	3.03e-2	3.74e-2	1.70e-4	1.33e-3	1.43e-3

Table 3.5 Alignment results on the sparse IRL and EUR Pallet datasets.

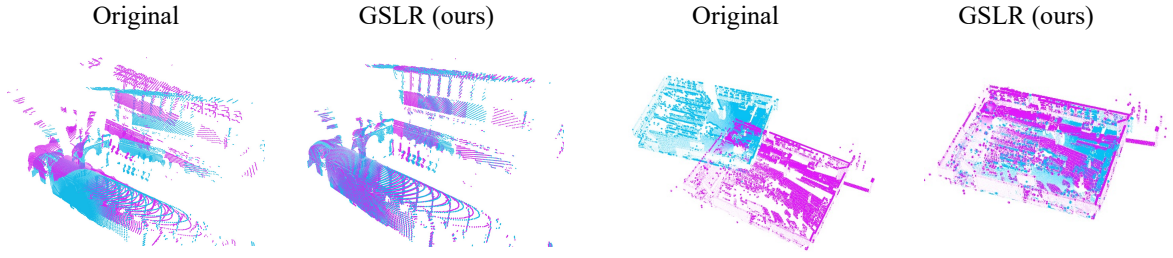


Fig. 3.7 Some example alignment results achieved by our GSLR on the ETH Hauptgebäude (left) and Office (right) datasets.

dense. This is because dense point clouds preserve repeated patterns well, making the local features similar to each other and thus increasing the number of mismatches. Consequently, the performance of the underlying robust estimator decreases.

We also conduct experiments to study the performance of GSLR in aligning sparse point clouds directly acquired from sensors. For experimental setup, we employ the Intel Research Lab (IRL) dataset [54] and the EUR pallet dataset [99], which are captured by 2D laser rangefinders with 180 and 761 points in each scan, respectively. The IRL dataset is captured by a mobile robot scanning an indoor scene, and the EUR pallet dataset is collected in a dynamic environment with the sensor fixed to several positions. To prepare the experimental data, for the IRL dataset, we bidirectionally pair scans with consecutive timestamps from the first 100 scans, leading to 200 trials in total. For the EUR pallet dataset, we similarly pair scans with an index interval of 5 among all the 565 scans as long as their sensor positions are the same, resulting 192 trials. For some candidate methods such as TEASER++ and PointNetLK that can only work with 3D point clouds, we lift the 2D points to 3D by setting their z -coordinates to 1. Other experimental setups are the same as those used in aligning the aforementioned ETH Hauptgebäude and Office datasets except for the threshold of the Huber-skip estimator of GSLR, which is tuned to 1.5 to better reject outliers. Table 3.5 records the results and Fig. 3.8 illustrates some examples. GSLR can present more accurate estimations than the other methods, demonstrating its practical value.

3.5.3 GSLR in isometric shape matching

Shape matching looks for vertex-wise correspondences between meshes. Following functional map [103], its objective function can be written as

$$\min_{\mathbf{C} \in \mathcal{O}(s), \mathbf{P} \in \Pi} \|\mathbf{P}\mathbf{F}\mathbf{C}^T - \mathbf{P}\mathbf{P}^T\mathbf{G}\|^2, \quad (3.39)$$

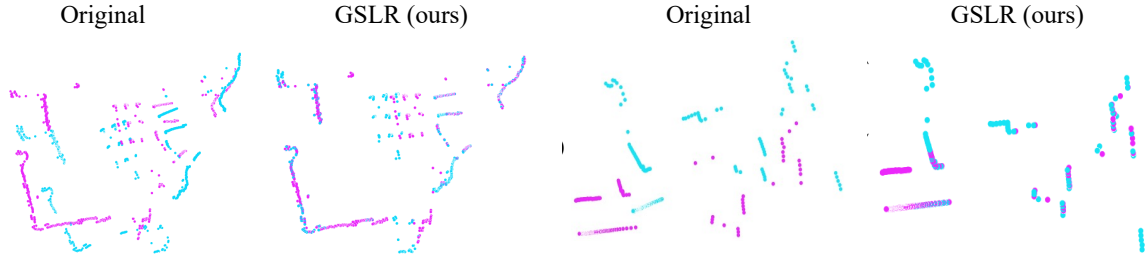


Fig. 3.8 Some example alignment results achieved by our GSLR on the 2D laser scans. Left: EUR Pallet dataset. Right: IRL dataset.

where $O(\cdot)$ is the orthogonal group, and $\mathbf{F}$ and $\mathbf{G}$ are the vertex-wise Laplacian eigen-functions of the two meshes. In this work, we limit the discussion within the near-isometric cases for which the vanilla functional map is designed. Currently, Applying Algorithm 4 to Objective (3.39) does not perform well on other scenarios such as non-isometric and partial shape matching. This is because these setups commonly require additional regularization terms or constraints on $\mathbf{C}$, and designing such elements is out of scope for GSLR.

Experimental setups and peer methods

We conduct both intra-class and inter-class matching on the FAUST dataset [13], and intra-class matching on the TOSCA one [17]. Inter-class tests on TOSCA are omitted due to their high non-isometry. For each task, we randomly select 40 pairs of shapes and report the mean values as the final results. Since the underlying ground-truth matches are full-rank permutations, we set the threshold of the Huber-skip estimator to 10 to only reject data with severe residuals. After the GSLR process, we solve one more step of LAP to deal with the vertices that are labeled as outliers.

We employ the commonly used geodesic metric for evaluation. We also use the coverage metric presented in [119] to measure the bijectivity preserved by the raw GSLR procedure, which is defined as the ratio of vertices that possess at least one correspondence to their total amount. We select ICP [103], BCICP [119] and ZoomOut [96] as the competing methods. For ICP, BCICP, and GSLR, we use 50-dimensional eigen-functions of the Laplace-Beltrami operator to formulate the sets of functions $\mathbf{F}$ and $\mathbf{G}$. For ZoomOut, we use both 50 and 100-dimensional counterparts to extensively explore its potential. All methods are initialized by the wave kernel signature [8] and the functional map with direct operator [119], and iterated 5 times.

Results

Figure 3.9 illustrates an example of the estimation errors on the FAUST dataset and Fig. 3.10 shows the overall results. Compared to the peer methods, GSLR can provide much higher percentages of correspondences that coincide with the ground truths. Furthermore, it can also effectively maintain the one-to-one correspondences by covering more than 90% vertices on all the pairs. For the TOSCA dataset, we decimate the meshes to approximately 3400 vertices each for a tractable elapsed time and memory consumption. The final error curves are obtained via NN-composition following [119]. Figure 3.11 records the results. As presented, GSLR can champion both the matching accuracy and the preservation of bijectivity. An illustration of coverage is given in Fig. 3.12.

3.5.4 Other experimental studies of GSLR

We study the stability of our employed Huber-skip robust estimator w.r.t. varying thresholds and analyze its efficiency against the trimming operator.

Threshold of the Huber-skip estimator

The threshold mentioned in Eq. (3.18) is the only hyper-parameter in our proposal. In general, Algorithm 4 performs stably under varying thresholds. For demonstration, we conduct tests on the point cloud alignment task. The experimental setup is the same as mentioned in Chapter 3.5.2 except that the ground-truth rotation is set to 15° in Euler angles. The two point clouds are aligned with a varying threshold ranging from 1 to 10 with a step size 0.5. Figure 3.13 shows the results; the peer methods are also reported for reference. GSLR can consistently provide more accurate estimations as long as the threshold is within a reasonable range.

Statistical efficiencies of the trimming operator and the Huber-skip estimator

We conduct Monte Carlo experiments to study the statistical efficiencies of the trimming operator and the Huber-skip estimator. We follow the common settings used in [56, 55] to test their respective efficiencies under the following conditions:

- Different dimensions of the regression variable $\mathbf{x}$: We increase the size of $\mathbf{x}$ from 3 to 43 with a step size of 2.
- Different numbers of data: We increase the number of data from 100 to 2000 with a step size of 100.

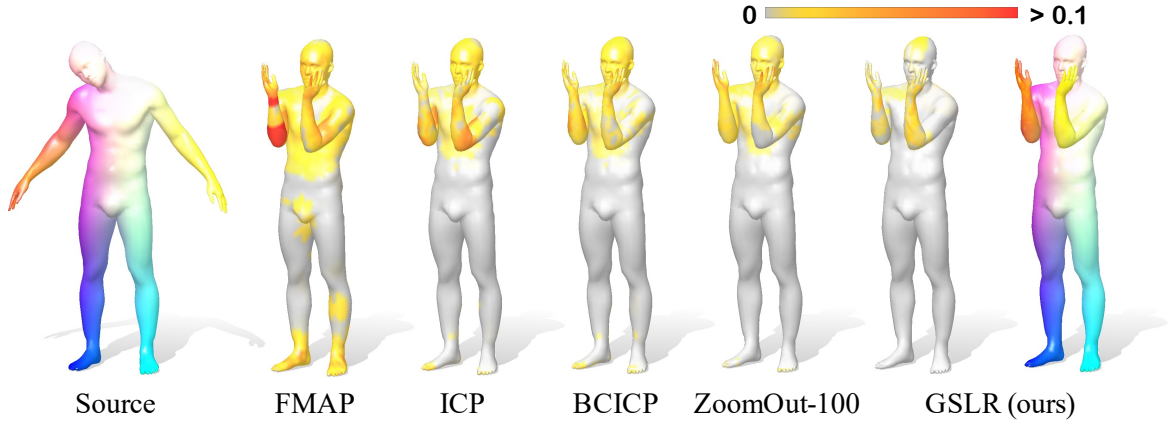


Fig. 3.9 An example of the geodesic error maps on the FAUST dataset. ZoomOut-50 is omitted for its inferior performance.

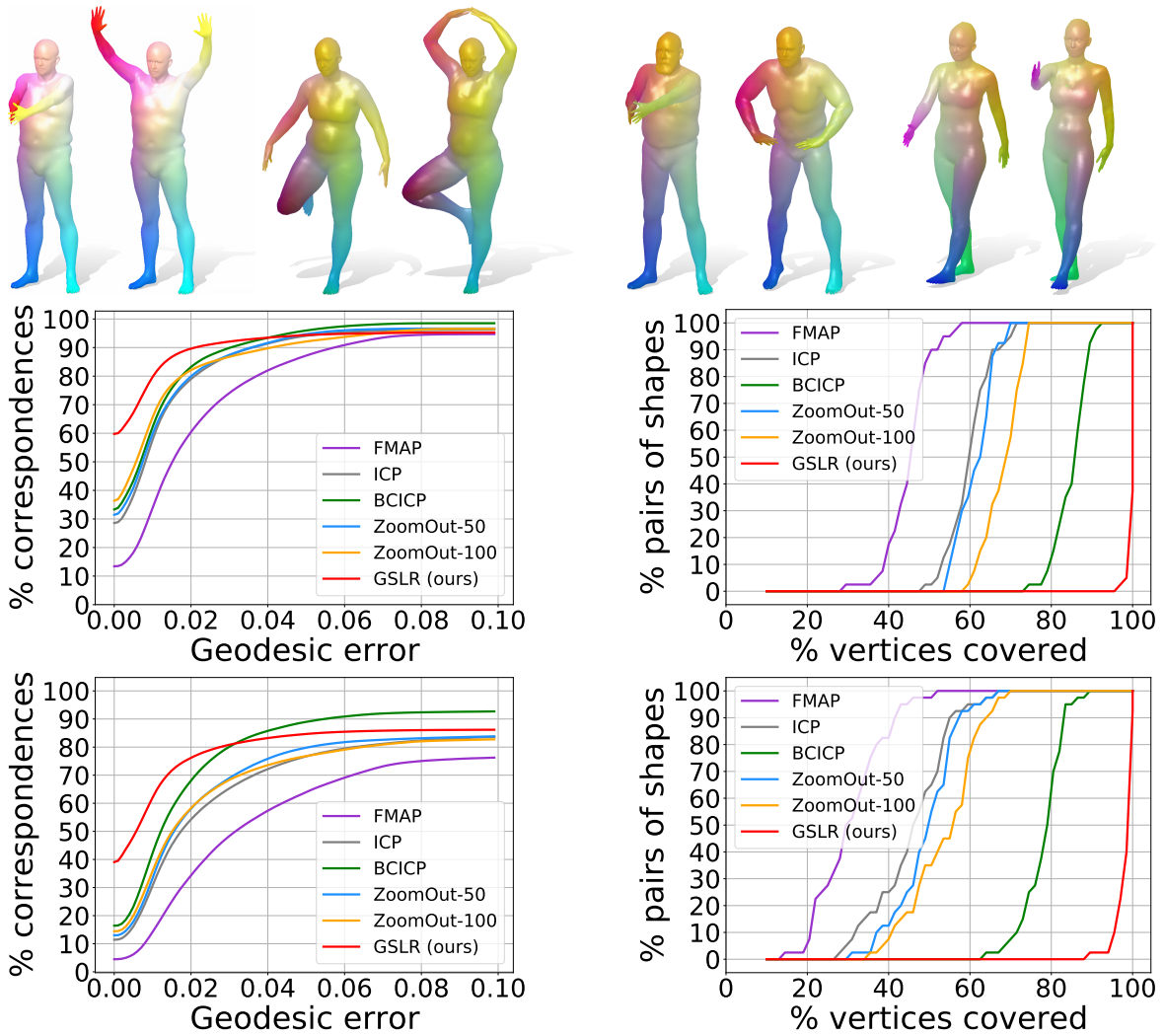


Fig. 3.10 Matching results on the FAUST dataset. Top row: Examples achieved by GSLR. Second and third rows: Intra-class and inter-class geodesic errors and coverage rates.

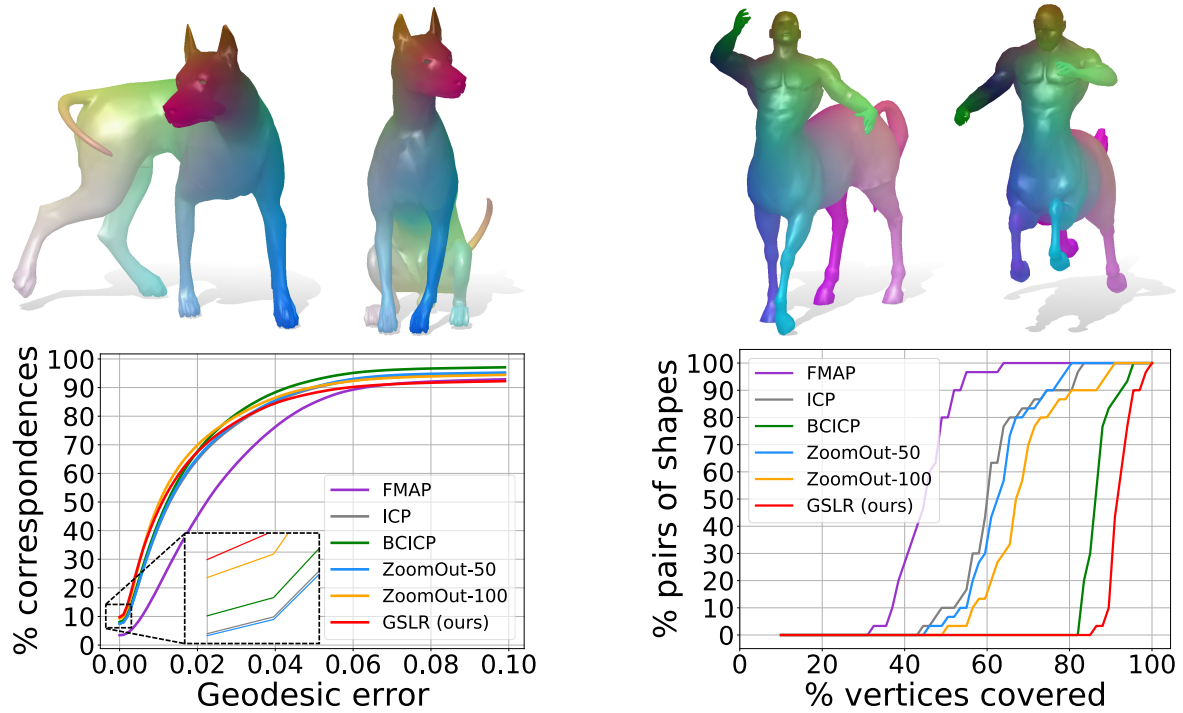


Fig. 3.11 Matching results on the TOSCA dataset. Top: Examples achieved by GSLR. Bottom: Geodesic errors and coverages.

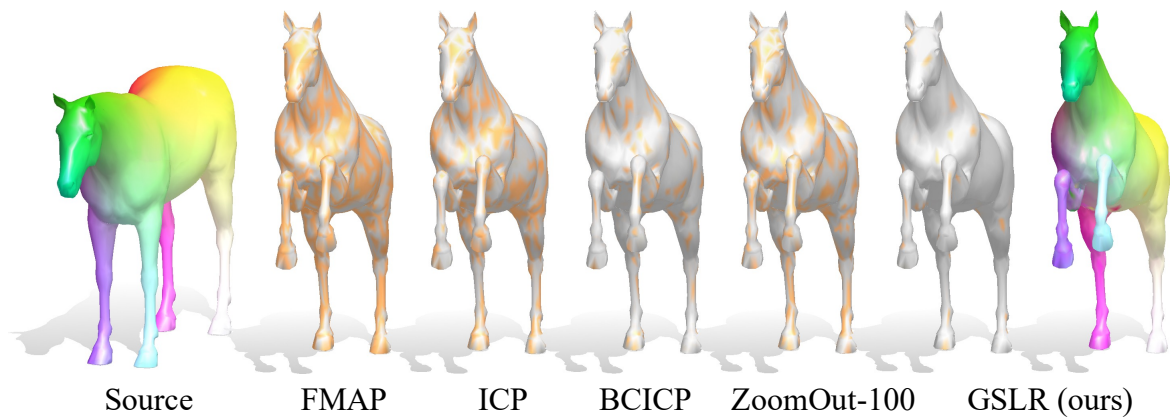


Fig. 3.12 An example of coverages on the TOSCA dataset. Vertices violating the one-to-one corresponding policy are marked as orange.

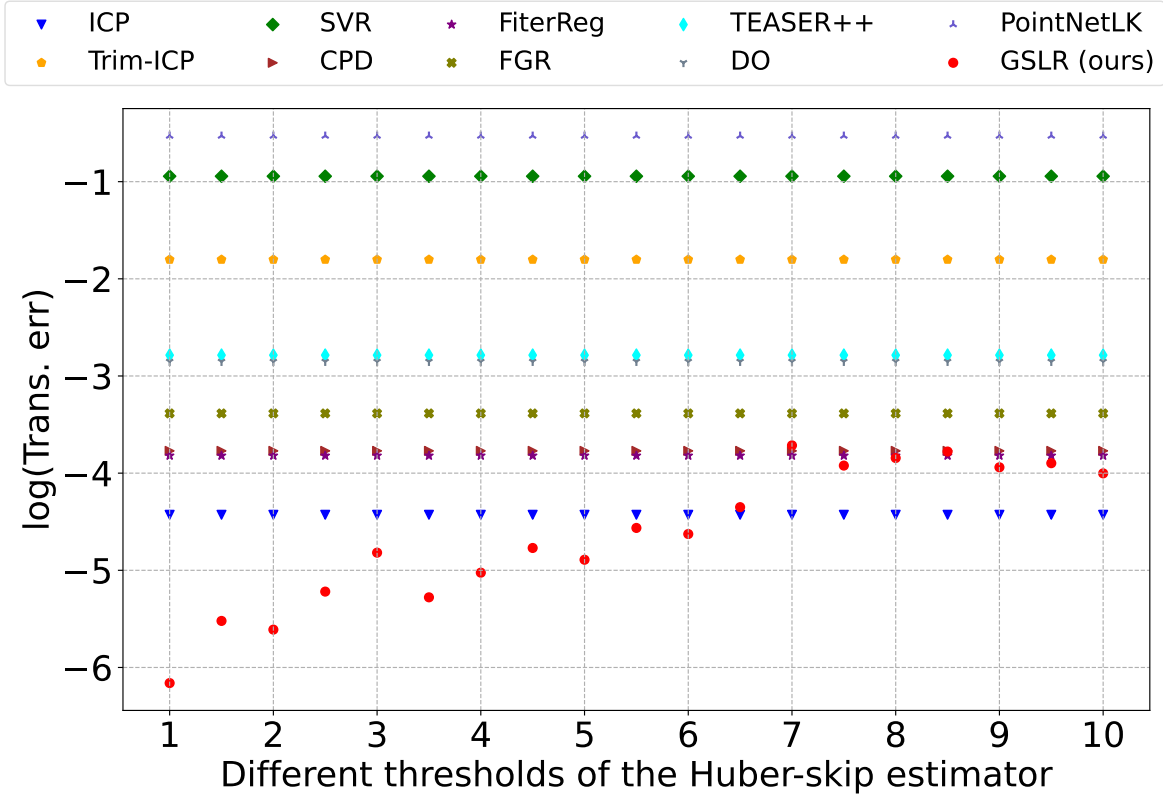


Fig. 3.13 Point cloud alignment accuracies w.r.t. varying thresholds of the Huber-skip estimator.

- Different ratios of outliers: We increase the ratio of outliers from 0 to 1 with a step size of 0.05.

All the data are randomly generated and corrupted by the Gaussian noise with covariance of 0.1. The control variables in each trial are set to 20 for the dimension of the regression variable, 2000 for the number of data, and 0.4 for the ratio of outliers. Due to the existence of outliers, we follow [56] to limit the maximum threshold of the Huber-skip estimator to 3.5 for reasonable performance. Each sub-setup (*e.g.*, efficiency w.r.t. a specific dimension of regression variable) is repeated 100 times, and their mean value is used as the final result. We first compare the efficiencies between the Huber-skip estimator with different thresholds and the 50%-trimming operator. As shown in the left column of Fig. 3.14, the Huber-skip estimator is more efficient than the trimming operator in most of the setups. Moreover, the right column of Fig. 3.14 presents the efficiencies of the trimming operator under varying trimming ratios. The Huber-skip estimator with the lowest and highest efficiencies are also plotted for comparison. As shown, even with careful tuning, only one setup of the trimming operator can surpass the least efficient setup of the Huber-skip estimator.

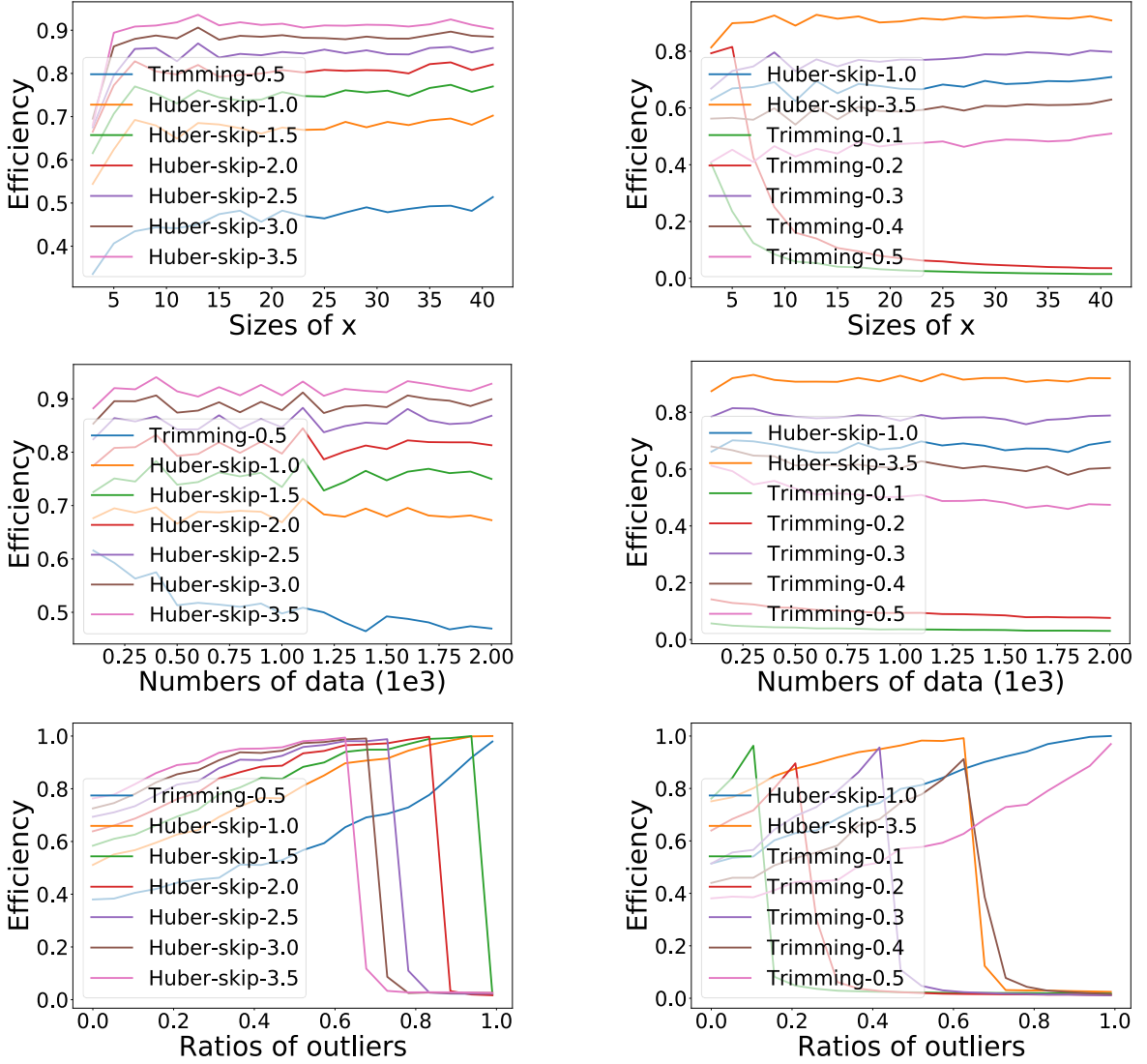


Fig. 3.14 Statistical efficiencies of the trimming operator and the Huber-skip estimator under distinct setups. Left: Varying thresholds in Eq. (3.18). Right: Varying ratios of the trimming operator.

3.6 Discussion and conclusion

In this work, we propose the GSLR formulation to generalize SLR to broader-ranging conditions where only a part of the data should match. The effective recovery condition and NP-hardness of the proposed formulation are also studied. To optimize GSLR, we also present a simple yet effective algorithm together with detailed convergence analyses. Distinct applications demonstrate the effectiveness of our proposals.

Table 3.6 Computational cost of Algorithm 4 w.r.t. varying point cloud densities.

#Points	Elapsed time (s)	Memory cost (MB)*	#Points	Elapsed time (s)	Memory cost (MB)*
~1000	1.36e1	12	~6000	1.53e3	432
~2000	6.76e1	48	~7000	1.89e3	588
~3000	2.61e2	108	~8000	3.10e3	768
~4000	4.36e2	192	~9000	5.85e3	972
~5000	9.36e2	300	~10000	7.08e3	1200

* The memory cost is computed using the maximum possible size of the cost matrix mentioned in Objective (3.17) (*i.e.*, 50% of the data is outliers) in 64-bit precision.

3.6.1 Limitations

The main limitation of our proposed Algorithm 4 lies in the computational cost, which is dominated by the LAP solver. To present a conceptual understanding, we set up an experiment same as the one mentioned in Chapter 3.5.2 except for the point cloud densities, which are varied from 1000 to 10000 with a step size of 1000. All the trials are carried out using an Intel Xeon E5-2680 CPU. As shown in Table 3.6, the memory consumption of Algorithm 4 increases quadratically w.r.t. the number of data due to the dense cost matrix constructed following Objective (3.17), and the elapsed time also increases rapidly as the data size grows. As a result, our current implementation will become infeasible given large-scale data.

3.6.2 Failure cases

As a local optimization algorithm, a typical failure case of Algorithm 4 is triggered by deficient initialization. For example, as the top part of Fig. 3.15 shows, two point clouds will be incorrectly aligned if the initial rotation is far biased from the latent ground truth. Another type of failure is caused by the invalid formulation of the robust estimation problem that makes Constraint (3.7) never satisfied. For example, as shown at the bottom of Fig. 3.15, GSLR may fail when directly applied to align two point clouds with an overlap ratio less than 50%. In such a case, some prior knowledge of the outliers, such as the non-overlap area identified by overlapping detection [35], is necessary to ensure the validity of the problem formulation itself.

3.6.3 Future work

For future work, we wish to further generalize the formulation of GSLR to deal with other tasks such as partial shape matching and alignment with other metrics. Also, to deal with the

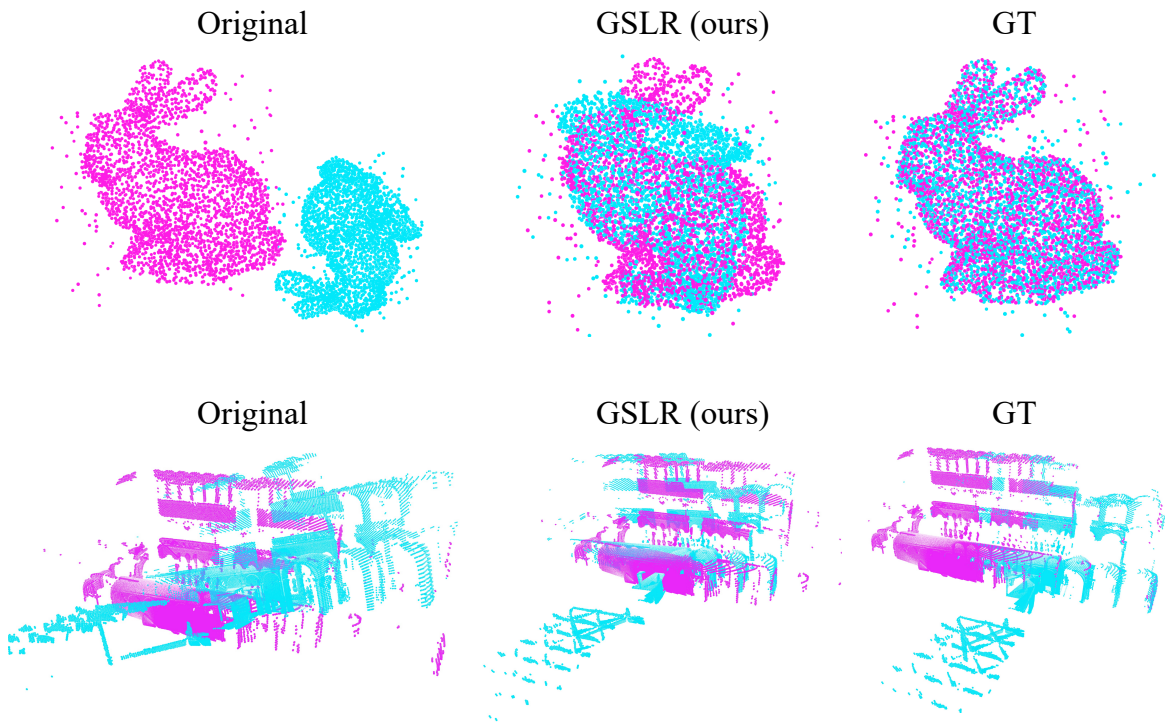


Fig. 3.15 Failure cases of GSLR. Top: Failure due to deficient initialization. Bottom: Failure due to invalid formulation of the robust estimation problem. GT is short for ground truth.

scalability problem, one may replace our currently used CPU-based Hungarian algorithm with the GPU-based one [32, 156], which appears to be much faster. Another possible remedy is combining GSLR with data partition techniques such as those developed for large-scale point cloud alignment [125, 112, 71], which functions well when the data are too large to be treated as a whole.

Chapter 4

Ambiguity analysis for principal component analysis-based intra-category alignment

While Chapters 2 and 3 study point cloud alignment for the same object, in the chapter, we focus on a broader range of alignment problems. *I.e.*, aligning point clouds of intra-category objects. This study is motivated by the fact that high-level recognition tasks prefer the input point clouds to be pre-aligned in an intra-category manner. To solve intra-category alignment, the canonical poses computed with PCA is a simple yet effective solution method to achieve such alignment. However, since the alignment presented by PCA is not entirely perfect, most of the recent works are devoted to developing powerful neural networks and features to overcome this deficiency, without thoroughly analyzing the PCA-based canonical poses themselves. In this chapter, we present a detailed study w.r.t. the PCA-based canonical poses of point clouds. Our investigation reveals that the ambiguity problem associated with the PCA-based canonical poses is handled insufficiently in some recent works. To this end, we also develop a simple pose selector module for disambiguation to aid existing neural networks designed for point cloud analysis. Experiments show that the sufficient identification of ambiguities can help to present conspicuous enhancement over state-of-the-art approaches on the challenging real-world dataset.

4.1 Introduction

As a primitive representation of 3D data, point clouds are widely employed as the direct data format for many high-level recognition tasks such as classification, retrieval, and segmen-

tation. However, there exists a fundamental problem when point clouds are used therein. Specifically, in such applications, we expect the recognition algorithms to present consistent inferences w.r.t. varying affine transformations on the point clouds. While the effects of scaling and translation can be eliminated effectively by normalization and centralization [113], achieving rotational invariance remains an open problem.

Various methods have been proposed to tackle this issue. There are attempts to learn to align shapes in an optimal pose [63, 39], or robustify the networks via equivariant properties [118, 124, 38]. However, these methods are not strictly rotation-invariant and require the rotational space to be densely sampled for data augmentation. Some works also attempt to handcraft rotation-invariant geometric features [22, 164, 134], although they inevitably suffer from information loss compared to directly using the Cartesian coordinates of the point clouds.

Some recent works propose to use intrinsically determined canonical poses to avoid information loss. As a practical tool, PCA enables us to intrinsically determine 3 orthogonal bases (*i.e.*, the principal axes) of a given point cloud and align them to the world Cartesian coordinate. Despite the effectiveness of PCA-based canonical poses, we observe that some recent works are devoted to developing more powerful networks without clearly studying the canonical poses themselves [70, 161, 165, 153]. Consequently, these works have not explicitly tackled the ambiguity problem of the PCA-based canonical poses, which hinders their performance.

In this work, we analyze in detail the PCA-based canonical poses in point cloud analysis. We study the actual number of ambiguities of the PCA-based canonical poses, explore the effects of PCA in point cloud analysis, and propose a pose selector module for disambiguation. Our investigation reveals that the accurate identification of ambiguities can lead to noticeable performance boosts than the recently developed networks and features. *I.e.*, state-of-the-art approaches would be outperformed by a large margin on the challenging real-world dataset if ambiguities are properly identified. We summarize our contributions as follows:

- We study the actual number of ambiguities of the PCA-based canonical poses, showing that it has not been fully addressed in recent works.
- We demonstrate the actual ability of PCA-based canonical poses by distinct experiments and detailed analysis, hoping to prompt future works in the context of rotation-invariant point cloud analysis.
- We propose a pose selector module that can be embedded to off-the-shelf neural networks proposed for point cloud analysis, which can effectively disambiguate the canonical poses and boost performance.

4.2 Related works

In this section, we briefly review the developments of point cloud analysis. We here concentrate on deep learning-based methods since they are currently dominating the point cloud analysis research for its accuracy.

4.2.1 Deep learning on point clouds

Point clouds are difficult for neural networks to handle due to their irregularity. As solutions, conventional methods propose to render them to images [133, 23] or conduct quantization to obtain volumetric grids [94, 114] to facilitate convolution. However, such approaches either cannot handle tasks, such as segmentation, that require point-wise labeling, or generalize poorly to dense point clouds due to high memory consumption.

Some recent research explores inputting point clouds directly to networks. Pioneered by PointNet [113], which uses point-wise convolution followed by a max-pooling layer to achieve permutation-invariant representations, the following works explore aggregating local information to boost performance. For example, DGCNN [149] employs graph convolution by mapping point clouds to k-nearest-neighbor graphs. PointConv [150] propose to approximate the convolution kernel with a multi-layer perceptron network. KPConv [136] uses a set of kernel points to define the convolution area. However, all these methods assume intra-category point clouds to be already aligned to the same pose, without which the performance would decline significantly.

4.2.2 Rotation-robust point cloud analysis

Some works aim at robustifying the networks w.r.t. randomly rotated point clouds. The primary philosophy lies in designing modules that are equivariant to rotations. For example, the pioneering work of Esteves *et al.* [38] propose to map the point clouds into spherical functions and introduces a spherical convolution operator equivariant to rotations. SFCNN [118] replaces the manually designed mapping with a trainable neural network to learn to project the point clouds onto discretized spheres. Spezialetti *et al.* [130] propose a self-supervised strategy to learn canonical poses via spherical convolution. Shen *et al.* [124] define a transformation that maps both the inputs and the intermediate-layer features into unit quaternions to realize rotational equivariance. RotPredictor [39] proposes to transform the Cartesian coordinate system into a cylindrical one, on which rotations are represented as translations. Consequently, rotational equivariance can be achieved due to the translation-equivariant property of convolutional neural networks [150]. Although these works present significant

robustness w.r.t. rotations, they are still not strictly rotation-invariant. Moreover, since most approaches in this thread require test-time augmentation (*a.k.a.*, voting) and such augmentations are often conducted by randomly sampling rotations from $SO(3)$, their performances are not entirely stable and heavily depend on the sampling efficiency.

4.2.3 Rotation-invariant point cloud analysis

Another thread of works explores achieving strict rotational invariance. For example, SPH-Net [111] proposes to extend the point clouds to volumetric functions and designs a rotation-invariant convolution kernel based on spherical harmonics. Its expression ability is, however, limited due to the calculation of feature norms. There also exist methods that focus on handcrafting rotation-invariant features based on intrinsic geometries. For example, based on the relative locations, distances, and angles among points, RIConv [164], Triangle-Net [152], Li *et al.* [80], and SRI-Net [134] manually design different forms of rotation-invariant features and propose their respective network structures. Despite the well-designed networks, these handcrafted geometric features inevitably lead to information loss.

On the other hand, some approaches propose to convert rotated point clouds to their PCA-based canonical poses to achieve rotational invariance. Compared to the feature-based methods, these canonical poses can ultimately preserve the shape information of the input point clouds. For example, Fujiwara and Hashimoto [43] leverage the distance field to disambiguate the signs of canonical poses and concatenate them to formulate the inputs. Xiao *et al.* [153] and Yu *et al.* [161] claim that there are 8 sign ambiguities of the canonical poses and employ trainable attention-based selection modules for disambiguation. Kim *et al.* [70] propose to conduct PCA from local patches to the entire point clouds. Zhao *et al.* [165] combine the canonical poses with handcrafted features to formulate the inputs. In summary, although these works have developed varying network structures to suit the canonical poses better, the associated ambiguity problem is unfortunately not clearly studied.

4.3 Ambiguities of the PCA-based canonical pose

In this section, we explore the exact number of ambiguities of the PCA-based canonical poses. To make this thesis self-contained, we first briefly review how the canonical pose is calculated. Specifically, for a given point cloud $\mathbf{P} \in \mathbb{R}^{n \times 3}$, PCA is performed by:

$$\frac{\sum (\mathbf{P}_i - \bar{\mathbf{P}}) (\mathbf{P}_i - \bar{\mathbf{P}})^T}{n} = \mathbf{E} \mathbf{\Lambda} \mathbf{E}^T, \quad (4.1)$$

where $\mathbf{P}_i \in \mathbb{R}^3$ is the i^{th} point of $\mathbf{P}$, $\bar{\mathbf{P}} \in \mathbb{R}^3$ is the center of $\mathbf{P}$, $\mathbf{E}$ is the eigenvector matrix composed of eigenvectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ (*a.k.a.*, principal axes), and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ are the corresponding eigenvalues (*a.k.a.*, principal values). By aligning the principal axes to the three axes of the world coordinate, we obtain the canonical pose as $\mathbf{P}_{\text{can}} = \mathbf{PE}$. The rotation-invariant property of $\mathbf{P}_{\text{can}}$ can be facilely derived as follows:

Proof. By applying a random rotation matrix $\mathbf{R} \in \text{SO}(3)$ on the point cloud $\mathbf{P}$, we can obtain its rotated version $\mathbf{PR}^T$. In the same manner as shown in Eq. (4.1), we can conduct PCA on it in the form of

$$\begin{aligned} & \frac{\sum (\mathbf{RP}_i - \mathbf{R}\bar{\mathbf{P}}) (\mathbf{RP}_i - \mathbf{R}\bar{\mathbf{P}})^T}{n} \\ &= \mathbf{R} \left(\frac{\sum (\mathbf{P}_i - \bar{\mathbf{P}}) (\mathbf{P}_i - \bar{\mathbf{P}})^T}{n} \right) \mathbf{R}^T \\ &= (\mathbf{RE}) \mathbf{\Lambda} (\mathbf{RE})^T, \end{aligned} \quad (4.2)$$

where $\mathbf{RE}$ becomes the new principal axes. Therefore, as mentioned above, the canonical pose can be computed as

$$(\mathbf{PR}^T)_{\text{can}} = \mathbf{PR}^T \cdot \mathbf{RE} = \mathbf{PE}, \quad (4.3)$$

on which the rotation $\mathbf{R}$ has no effect. $\square$

4.3.1 Sign Ambiguity

As pointed out by some previous works [43, 161, 153], the PCA-based canonical pose contains sign ambiguities. Specifically, for a certain eigenvector $\mathbf{e}$, both $+\mathbf{e}$ and $-\mathbf{e}$ can satisfy the rule of eigen-decompositions. Consequently, by assigning different signs to each of the three eigenvectors, there are 8 possible poses when calculating the canonical pose of a given point cloud.

While this ambiguity problem is ignored in some existing works [70, 165], it is also explicitly pointed out and handled via trainable modules [153, 161] or analytical methods [43]. However, we argue that this 8-ambiguity declaration contains non-rotational transformations. In detail, assume that a specific combination of eigenvectors $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ is with determinant 1. Consequently, only 4 out of the 8 ambiguities are with determinant 1, representing rotations, and the others are with -1 , characterizing improper rotations (*i.e.*, the combination of rotation and reflection, as illustrated in Fig. 4.1). A detailed justification is presented in Table 4.1.

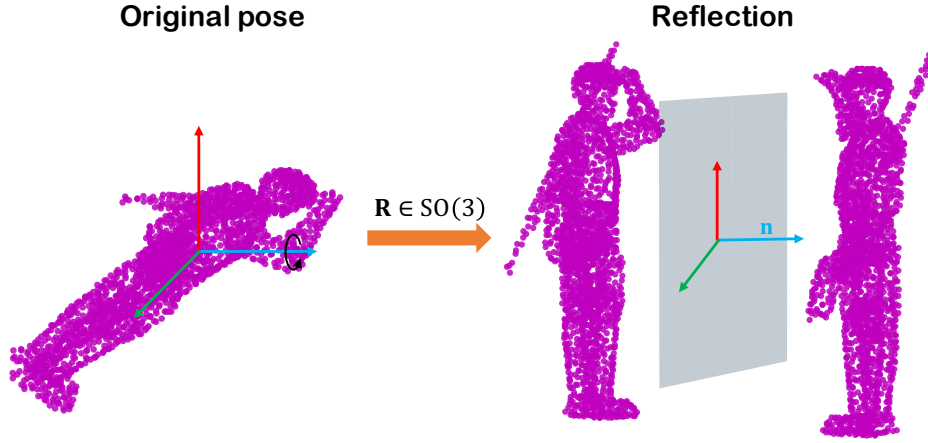


Fig. 4.1 A conceptual explanation of improper rotation, which consists of: 1) rotation along an axis; 2) reflection along the plane perpendicular to this axis. Pure planar reflection can be considered as a special case of improper rotation where the rotation stage vanishes.

	Determinant	Geometric meaning
$[+\mathbf{e}_1, +\mathbf{e}_2, +\mathbf{e}_3]$	1	Rotation
$[-\mathbf{e}_1, -\mathbf{e}_2, +\mathbf{e}_3]$	1	Rotation
$[+\mathbf{e}_1, -\mathbf{e}_2, -\mathbf{e}_3]$	1	Rotation
$[-\mathbf{e}_1, +\mathbf{e}_2, -\mathbf{e}_3]$	1	Rotation
$[-\mathbf{e}_1, +\mathbf{e}_2, +\mathbf{e}_3]$	-1	Improper rotation
$[+\mathbf{e}_1, -\mathbf{e}_2, +\mathbf{e}_3]$	-1	Improper rotation
$[+\mathbf{e}_1, +\mathbf{e}_2, -\mathbf{e}_3]$	-1	Improper rotation
$[-\mathbf{e}_1, -\mathbf{e}_2, -\mathbf{e}_3]$	-1	Improper rotation

Table 4.1 Determinants and geometric meanings of the eigenvector matrix w.r.t. different assignment of signs.

4.3.2 Order ambiguity

The aforementioned claim of 8 ambiguities is also insufficient. Specifically, this claim takes for granted that the eigenvectors are sorted according to some rules (*e.g.*, by arranging the eigenvectors in the same order w.r.t. ascending/descending eigenvalues). However, we argue that such sorting rules are just manually defined for convenience. *I.e.*, there is no mathematical support for taking the eigenvector corresponding to the largest/smallest eigenvalue as the x -axis and the smallest/largest one as the z -axis. As we will show later in Chapter 4.6.2, this sorting-dependent order in fact obviously hinders the performances for aligning intra-category objects to different poses. Consequently, apart from the sign ambiguities, we claim that there also exist 6 order ambiguities by permuting the 3 eigenvectors.

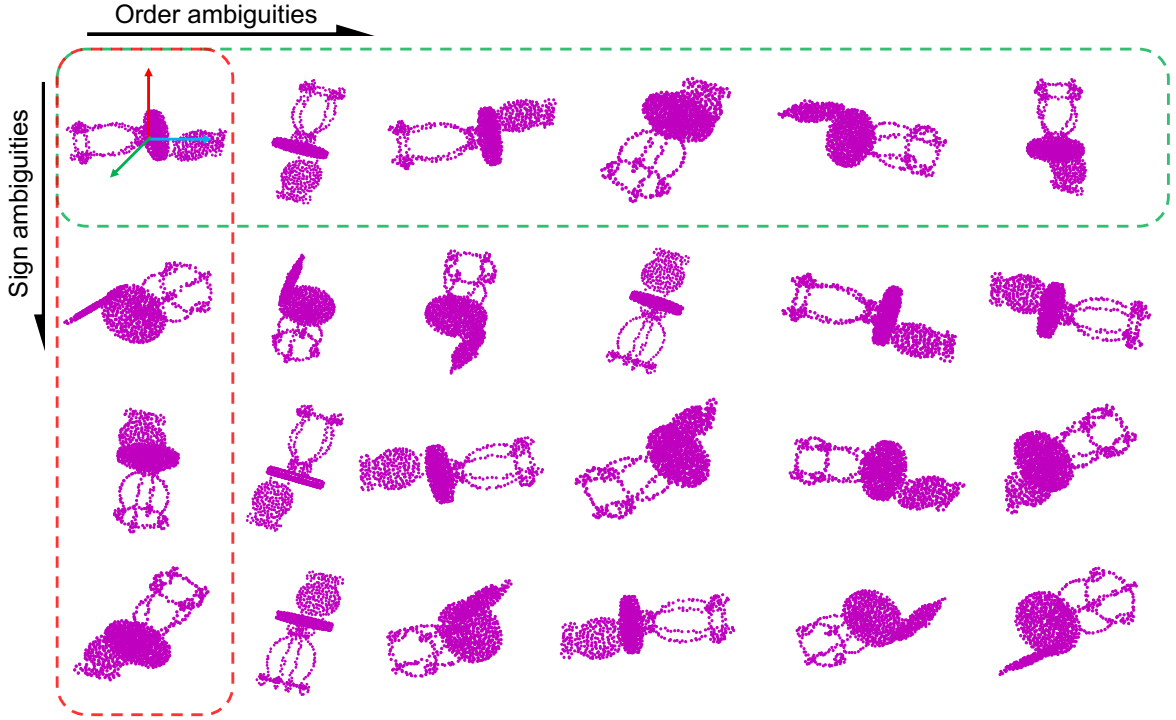


Fig. 4.2 An example illustration of the 24 ambiguities of the PCA-based canonical poses. The top row and the left column consist of the 6 order ambiguities and 4 sign ambiguities, respectively.

4.3.3 Summary of ambiguities

By summarizing the above discussions of ambiguities together, we can conclude that there in fact should be

$$4(\text{sign ambiguities}) \times 6(\text{order ambiguities}) = 24$$

ambiguities in total speaking of the PCA-based canonical poses of point clouds. A detailed visualization of the 24 ambiguities of the PCA-based canonical poses as well as the 4 and 6 counterparts obtained with order-disambiguation and sign-disambiguation is presented in Fig. 4.2.

4.4 Pose selector: Learning to disambiguate the canonical poses

In this section, we propose a novel function to cope with the ambiguities, which can effectively boost the performances of point cloud analysis tasks.

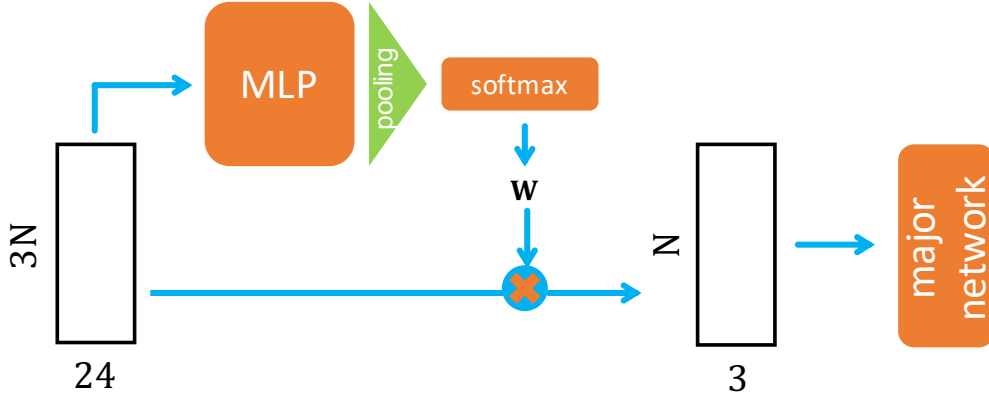


Fig. 4.3 Structure of the pose selector, which predicts the weights of the 24-dimensional inputs. We implement the module with 256-512 1D convolutional filters followed by max-pooling, 24-dimensional linear mapping, and softmax.

A straightforward strategy to incorporate the aforementioned ambiguities for rotation-invariant analysis is to consider every possible pose as an independent training instance, so that the network would learn to associate each possible pose to the label of the corresponding shape. Although this strategy works well, as we will show in the following sections, it is more effective to let the model learn to select the optimal canonical pose by incorporate pose selection into the training process. To this end, we design a trainable module to learn to use the 24 poses.

We denote this module as “pose selector.” As illustrated in Fig. 4.3, it is a lightweight trainable network that can be incorporated into any network developed for point cloud analysis. Specifically, by flattening the point cloud $\mathbf{P} \in \mathbb{R}^{n \times 3}$ to a vector and concatenating all the 24 ambiguities together, we obtain a $3n \times 24$ representation of all the canonical poses. Pose selector then takes this representation as input and axis-wise converts it into high dimensional features, which are pooled among the $3n$ dimensions to reach a single feature vector. This feature is further linearly mapped back to 24 dimensions and activated by softmax, resulting in a 24-dimensional weight vector $\mathbf{w}$. To obtain the selected point cloud $\mathbf{P}_{\text{sel}} \in \mathbb{R}^{n \times 3}$ in the optimal pose, we take the weighted sum:

$$\mathbf{P}_{\text{sel}} = \sum_{d=1}^{24} \mathbf{w}_d \mathbf{P}_d, \quad (4.4)$$

where $\mathbf{w}_d$ and $\mathbf{P}_d$ are the weight and corresponding canonical pose, respectively.

There remain ambiguities in the orders of the 24 poses during concatenation (*i.e.*, 24! possible orders for concatenation). However, owing to the closedness of $\text{SO}(3)$, we can always generate the remaining 23 poses in a deterministic manner from an arbitrary canonical

pose. Therefore, the order ambiguity that arises from concatenation can in fact be reduced to 24.

4.5 Performance study using the correctly identified ambiguities

We claim that the proper identification of ambiguities can lead to more significant accuracy enhancements than well-designed networks or handcrafted rotation-invariant features. For verification, we conduct experiments on popularly used classification and segmentation benchmarks to compare with the performance of related works.

4.5.1 Implementation details

We employ the well-known DGCNN [149] network structure (with the spatial transformer network [63]) as our backbone without any modification. All the hyper-parameters and training setups remain the same as suggested in [149], except that the initial learning rate of the SGD optimizer is reduced from 0.1 to 0.01 for faster convergence and fine-tuning purpose. In the training phase, we randomly sample from all the possible $24!$ different concatenation patterns to robustify the model. For testing, we report two kinds of results: with and without test-time augmentation (TTA), which takes all the 24 concatenation patterns generated along a specific root as inputs and uses their mean value as the final prediction.

For experimental setup, we follow the related works [164, 134, 161, 80] and compare different algorithms in 3 modes:

- Both training and testing sets are rotated around the z -axis: z/z .
- Training set is rotated around the z -axis and testing set is randomly rotated: $z/\text{SO}(3)$.
- Both training and testing sets are randomly rotated: $\text{SO}(3)/\text{SO}(3)$.

In our implementation, rotation matrices are randomly sampled with the *special_ortho_group* function of *Scipy* [143].

4.5.2 Object classification

We first carry out comparisons on the synthetic ModelNet40 dataset [151], which consists of 12311 meshes from 40 categories with 9843 for training and 2468 for testing. In practice, we use the data released by Qi *et al.* [113], where the point clouds are already pre-processed.

	Method	Inputs	z/z	$z/\text{SO}(3)$	$\text{SO}(3)/\text{SO}(3)$	Acc. drop
Rotation-sensitive	PointNet [113]	xyz	88.5	16.4	70.5	54.1
	DGCNN [149]	xyz	92.2	20.6	81.1	60.5
	PointNet++ [115]	xyz	89.3	28.6	85.0	56.4
	PointConv [150]	xyz	91.6	-	85.6	-
Rotation-robust	Shen <i>et al.</i> [124]	xyz	83.0	83.0	83.0	0.0
	Spherical CNN [38]	voxel	88.9	76.9	86.9	10.0
	a^3 SCNN [83]	voxel	89.6	87.9	88.7	0.8
	SFCNN [118]	xyz	91.4	84.8	90.1	5.3
	SFCNN [118]	xyz + normal	92.3	85.3	91.0	5.7
	RotPredictor [39]	xyz	92.1	-	90.8	-
	RIConv [164]	feature	86.5	86.4	86.4	0.0
Rotation-invariant	Triangle-Net [152]	feature	-	-	86.7	-
	SRI-Net [134]	feature	87.0	87.0	87.0	0.0
	SPH-Net [111]	xyz	87.7	86.6	87.6	1.0
	Yu <i>et al.</i> [161]	xyz + feature	89.2	89.2	89.2	0.0
	Li <i>et al.</i> [80]	feature	89.4	89.4	89.3	0.1
	RI-GCN [70]	xyz	89.5	89.5	89.5	0.0
	RI-GCN [70]	xyz + normal	91.0	91.0	91.0	0.0
	LGR-Net [165]	xyz + normal + feature	90.9	90.9	91.1	0.2
	Ours (w/o TTA)	xyz	90.2	90.2	90.2	0.0
	Ours (w/ TTA)	xyz	91.6	91.6	91.6	0.0

Table 4.2 Classification accuracy (%) on the ModelNet40 dataset. Point clouds for all the methods are with density 1024. Within the inputs column, xyz denotes the Cartesian coordinates of the point clouds and features stand for handcrafted geometric features. The last column records the differences between $z/\text{SO}(3)$ and $\text{SO}(3)/\text{SO}(3)$.

We employ both rotation-robust and rotation-invariant approaches for comparison. Peer methods characterized as rotation-robust all aim to learn rotation-equivariant representations by converting the Cartesian coordinates to either spherical or cylindrical ones. For the rotation-invariant counterparts, they mainly use the PCA-based canonical poses [70] or handcrafted geometric features [164, 152, 134, 80], or both of them [161, 165] as the inputs. Within our knowledge, LGR-Net [165] is currently the most competitive algorithm that leads the scoreboards of various tasks.

The results are presented in Table 4.2. It is facile to champion the accuracy providing correctly identified ambiguities. Among the peer algorithms, RI-GCN (xyz only) is especially meaningful in comparison since it also uses the PCA-based canonical poses as inputs and develops powerful networks to better aggregate local features. However, it did not address the ambiguity problem, resulting in a lower accuracy than our method.

4.5.3 Object part segmentation

We also evaluate different methods on the part segmentation task, which aims to predict point-wise segmentation labels for the input point clouds. For experimental setup, we employ the ShapeNet part segmentation dataset [20] for benchmarking and use the pre-processed data released by Qi *et al.* [113], which consists of 16881 point clouds from 16 categories partitioned with 50 part labels in total. We follow the standard train-test split with 14007 objects for training and 2874 for testing. The experiments are again conducted in 3 modes as mentioned above. For evaluation, we calculate the mean intersection of union (mIoU, %) for each shape and report the mean value over all instances as the final results.

We first show some qualitative segmentation results from each class in Fig. 4.4, where our results are compared with the ground truth and the results obtained by using only one of the poses that is randomly selected. As shown in the figures, some of the shapes can be correctly segmented just by using one of the poses. However, for some shapes, chairs for example (right side of the first row), wrong classes are assigned. For other shapes such as laptops and tables (left side of the second row, right side of the last row, respectively), the labels are reversed, showing the negative effects of random orientation. On the other hand, these shapes are successfully segmented into correct parts using our pose selector, despite their varying poses. Our method especially excels at shapes whose self-symmetry is prominent.

Table 4.3 presents the quantitative results. Some of the methods mentioned in the classification experiments are omitted since their networks are not modified for the segmentation task in their respective proposals. We emphasize that the enhancement over LGR-Net is achieved without including normals and geometric features in the input as it does. In fact, such complementary inputs are known to boost the performances (*i.e.*, as shown in Table 4.2,

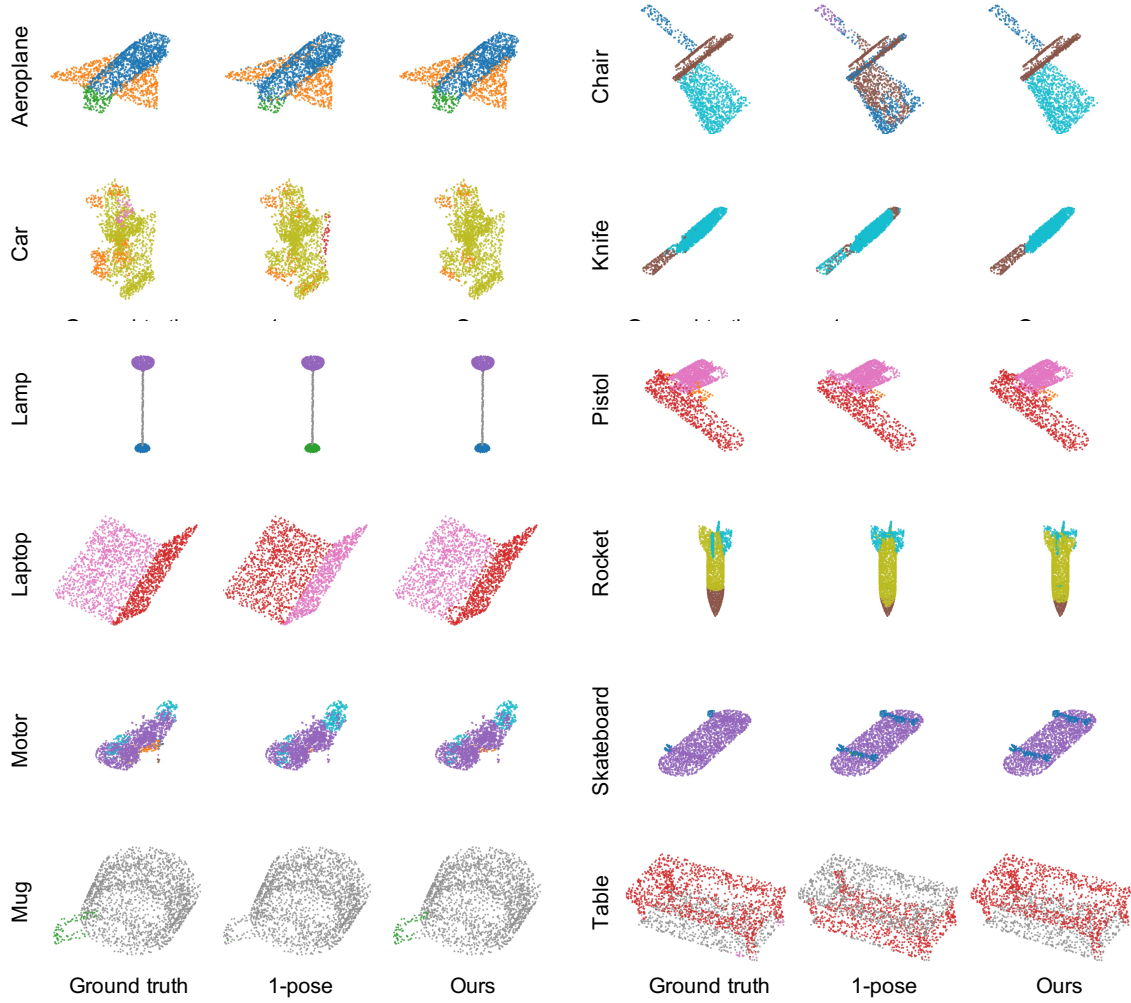


Fig. 4.4 Visualization of the ground truth segmentation, results using 1 of the 24 possible poses, and our results from the latter 8 classes in ShapeNet part segmentation dataset. The poses are drawn from the 24 possible PCA-based canonical poses.

even just adding normals can lead to approximately 1% enhancements for both SFCNN and RI-GCN). Moreover, by comparing the performances on both classification and segmentation tasks, we can observe that the abilities of some specifically designed features and network structures [70, 80] are not very consistent among different point cloud analysis tasks, showing as the competitive performances on a particular application but limit results on the other one. The per-class mIoU is reported in Table 4.4.

	Method	Inputs	$z/\text{SO}(3)$	$\text{SO}(3)/\text{SO}(3)$	Drop of mIoU
Rotation-sensitive	PointCNN [81]	xyz	34.7	71.4	36.7
	DGCNN [149]	xyz	37.4	73.3	35.9
	PointNet [113]	xyz	37.8	74.4	36.6
	PointNet++ [115]	xyz	48.2	76.7	28.5
Rotation-robust & invariant	Triangle-Net [152]	feature	-	72.5	-
	RIConv [164]	feature	75.3	75.5	0.2
	RI-GCN [70]	xyz	77.2	77.3	0.1
	SRI-Net [134]	feature	80.0	80.0	0.0
	Li <i>et al.</i> [80]	feature	82.2	82.5	0.3
	LGR-Net [165]	xyz + normal + feature	-	82.8	-
	Ours (w/o TTA)	xyz	81.7	81.7	0.0
	Ours (w/ TTA)	xyz	83.1	83.1	0.0

Table 4.3 Mean IoU (%) over all instances on ShapeNet. The inputs of each algorithm are the same as mentioned in Table 4.2. All point clouds are with density 2048 except for RI-GCN [70] and Triangle-Net [152], whose inputs are with density 1024.

Method	Overall mIoU	aero	bag	cap	car	cha.	earph.	guit.	knif.	lam.	lap.	mot.	mug	pist.	rock.	ska.	tab.
PointNet [113]	74.4	81.6	68.7	74.0	70.3	87.6	68.5	88.9	80.0	74.9	83.6	56.5	77.6	75.2	53.9	69.4	79.9
PointNet++ [115]	76.7	79.5	71.6	87.7	70.7	88.8	64.9	88.8	78.1	79.2	94.9	54.3	92.0	76.4	50.3	68.4	81.0
PointCNN [81]	71.4	78.0	80.1	78.2	68.2	81.2	70.2	82.0	70.6	68.9	80.8	48.6	77.3	63.2	50.6	63.2	82.0
DGCNN [149]	73.3	77.7	71.8	77.7	55.2	87.3	68.7	88.7	85.5	81.8	81.3	36.2	86.0	77.3	51.6	65.3	80.2
RIConv [164]	75.5	80.6	80.2	70.7	68.8	86.8	70.4	87.2	84.3	78.0	80.1	57.3	91.2	71.3	52.1	66.6	78.5
Li <i>et al.</i> [80]	82.5	81.4	84.5	85.1	75.0	88.2	72.4	90.7	84.4	80.3	84.0	68.8	92.6	76.1	52.1	74.1	80.0
LGR-Net [165]	82.8	81.7	78.1	82.5	75.1	87.6	74.5	89.4	86.1	83.0	86.4	65.3	92.6	75.2	64.1	79.8	80.5
Ours (w/o TTA)	81.7	81.9	58.2	77.0	71.8	89.6	64.2	89.1	85.9	80.7	84.7	46.8	89.1	73.2	45.6	66.5	81.0
Ours (w/ TTA)	83.1	83.7	62.9	79.1	73.4	90.1	64.2	90.3	86.4	82.5	87.3	46.5	89.1	75.4	46.1	66.6	81.3

Table 4.4 Per-class mIoU (%) of the SO(3)/SO(3) setup. RI-GCN [70], Triangle-Net [152] and SRI-Net [134] are omitted since the per-class mIoUs are not reported in their respective paper. Also, since the results originally reported in Li *et al.* [80] and LGR-Net [165] in their papers are the mean values of the mIoU over all categories, we here recalculate their mIoU over all the instances using their reported per-category mIoU and summarize these converted results for a straightforward comparison.

	z^*/z^*	$z^*/\text{SO}(3)$	$\text{SO}(3)/\text{SO}(3)$
PointNet [113]	79.4	16.7	54.7
PointNet++ [115]	87.8	15.0	47.4
PointCNN [81]	89.9	14.6	63.7
DGCNN [149]	87.3	17.7	71.8
RIConv [164]	-	78.4	78.1
LGR-Net [165]	-	81.2	81.4
Ours (w/o TTA)	84.3	84.3	84.3
Ours (w/ TTA)	86.7	86.7	86.7

Table 4.5 Classification accuracy (%) on the real-world ScanObjectNN dataset. z^* denotes the original pre-aligned dataset without any rotations. All point clouds are with density 1024. Our approach can noticeably outperform LGR-Net, which is the best peer method on synthetic datasets.



Fig. 4.5 Some example point clouds from the *OBJ_BG* dataset of ScanObjectNN. The data therein is challenging for containing noise, holes, deformations, and incompletely removed backgrounds.

4.5.4 Classification on real-world dataset

A major concern of applying PCA to real-world point clouds lies in its sensitiveness w.r.t. various nuisances such as noise, incompleteness, and deformations. Therefore, in this section, we study how the PCA-based canonical poses perform in such scenarios. For experimental setup, we follow LGR-Net [165] and employ the *OBJ_BG* dataset of ScanObjectNN [141], which contains scans of 2890 indoor objects classified into 15 categories divided into 2312 for training and 578 for testing¹. In this dataset, all objects are manually aligned to the closest poses w.r.t. their CAD models, leading to analogously the same pose within each category. In general, this dataset is much more challenging than the synthetic ModelNet40 for containing noise, holes, deformations, and backgrounds. Some example point clouds are illustrated in Fig. 4.5.

We report the results in Table 4.5. Compared to LGR-Net, which is the best peer method on both ModelNet40 and ShapeNet, our method presents a noticeable outperformance on the

¹The author-released dataset contains 2890 point clouds although it is claimed as 2902 in [141].

	ModelNet40	ShapeNet
w/o pose selector	91.3	82.8
w/ pose selector	91.6	83.1

Table 4.6 Pose selector can increase the accuracy of different tasks compared to the vanilla network, demonstrating its effectiveness in disambiguating the canonical poses.

challenging real-world dataset (*i.e.*, 5.3% enhancement in accuracy). This result indicates that the PCA-based canonical poses with properly identified ambiguities are in fact more effective than some handcrafted features and related network structures. Furthermore, by comparing the results under the z/z setup, we can observe that the result presented by PCA-based canonical poses is only slightly worse than the one obtained with the manual alignment, demonstrating the robustness of PCA-based canonical forms w.r.t. various nuisances in the real world.

4.5.5 Ablation studies

In this section, we present various ablation studies to demonstrate the effectiveness and time consumption of our proposed pose selector module, as well as the effects of distinct backbone network structures.

Ablation study of the pose selector

In this section, we study the effectiveness of our proposed pose selector module. For comparison, in the training phase, we randomly select 1 from the 24 ambiguities and insert it into the vanilla DGCNN network. For testing, we conduct TTA over all the ambiguities. Experiments are carried out on both classification and segmentation tasks for clear comparison. Results are reported in Table 4.6. Our pose selector can effectively enhance the performance on different tasks by merging the information of all the possible canonical poses. Furthermore, we observe that the pose selector can also accelerate the convergence compared to using the vanilla DGCNN network.

Time consumption of the pose selector

As mentioned in Chapter 4.4, our proposed pose selector is a lightweight module that barely increases the computation time at the inference phase. For demonstration, we record the time required for classifying a point cloud of shape (1024×3) with both the vanilla DGCNN network structure and the one that consists of the pose selector. In our test, the former version takes 6.03 milliseconds to infer a certain input pose and the latter requires 6.51 milliseconds

	Pre-aligned pose	Data augmentation with random rotations	Disambiguation with Strategy III	All possible canonical poses
PointNet [113]	88.5	70.5	82.8	86.7
PointConv [150]	92.4	85.6	86.7	89.1
DGCNN [149]	92.9	81.1	87.2	91.6

Table 4.7 Classification accuracy (%) w.r.t. distinct networks. Strategy III stands for the disambiguation method described in Chapter 4.6.2, with which all the ambiguities are eliminated.

on a Tesla V100 GPU. The bottleneck of time consumption lies in the data exchange between CPU and GPU.

Effect of different backbone networks

We carry out experiments to study how different network structures can affect performance. For experimental setup, we select PointNet [113], PointConv [150], and DGCNN [149] as the backbones and conduct classification on ModelNet40. The results are presented in Table 4.7. As shown in the table, the performance of all the networks can be boosted by providing PCA-based canonical poses with correctly handled ambiguities.

Effect of the spatial transform networks

Spatial transform networks (STN) [63] are self-attention modules designed to take the point cloud as input and learn to apply a spatial transformations on it. In general, such a module can effectively handle small rotational perturbations but is weak for large ones. Therefore, although STN is commonly mentioned as a component for various existing network structures [113, 149, 115], it is often omitted in realistic implementation for being either redundant (on pre-aligned data) or ineffective (on randomly rotated data). However, we find that STN does contribute when PCA-based canonical poses are used as the inputs. Specifically, the classification accuracy declines from 91.6% to 89.7% if STN is omitted. This result agrees with our observation that PCA can achieve rough alignments with small rotational disorders. Therefore, we consider STN as a necessary module for complementing PCA-based canonical poses.

	Pre-aligned pose	Selected canonical pose	All possible canonical poses
Acc.	92.9	92.0	91.6
mIoU	85.2	84.7	83.1

Table 4.8 Classification accuracy on ModelNet40 and mIoU (%) on ShapeNet w.r.t. different input pose on the classification and part segmentation tasks. Results from left to right are obtained with: the original datasets; manually selected canonical pose via Eq. (4.5); and all the 24 ambiguities.

4.6 Further explorations regarding the PCA-based canonical poses

Despite its effectiveness, the capacities of PCA-based canonical poses are not clearly studied. Therefore, we desire to explore the following questions with the hope to prompt future research in the context of rotation-invariance point cloud analysis:

- How well can the PCA-based canonical poses perform?
- Can we reduce the number of ambiguities in the light of disambiguation methods?
- To what extent do the inaccurate ambiguities hinder the performance?
- What does PCA conduct on the point clouds? In what conditions does it perform the best?

All the experiments mentioned hereafter are with exactly the same setup as mentioned in Chapter 4.5. TTA is conducted for all the experiments containing ambiguities. For experiments that do not contain ambiguities, we switch to use the vanilla DGCNN without the pose selector.

4.6.1 How well can the canonical poses perform?

Regardless of the ambiguity problem, PCA also cannot perfectly align intra-category objects to the same canonical pose due to their varying shapes. Consequently, some existing works [70, 161, 165, 39] consider these imperfect alignments as the primary reason that hampers the performance when canonical poses are used for rotation-invariant point cloud analysis. However, we argue that such a declaration is biased due to the improperly identified ambiguities.

In this section, we test the maximum capability of the canonical poses. Specifically, we desire to explore how well the canonical poses can perform if the ambiguity problem is

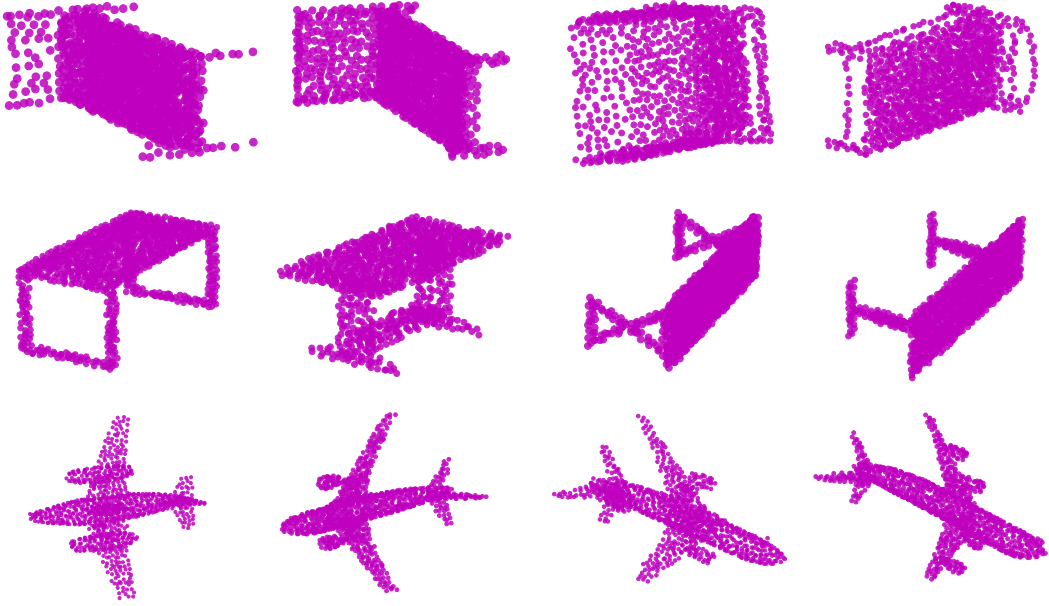


Fig. 4.6 Disambiguation methods may assign objects within the same category to different poses.

ideally solved. For experimental setup, we manually select 1 pose from the 24 ambiguities to make the poses of intra-category objects as similar as possible. In the implementation, since the original point clouds within the ModelNet40 dataset are already precisely aligned, we use them as reference and select 1 from all the 24 possible canonical poses that minimizes the rotational residual:

$$\mathbf{P}_{\text{selected}} = \underset{\mathbf{P} \in \mathcal{A}}{\operatorname{argmin}} \|\mathbf{R}_{\mathbf{P}} - \mathbf{I}\|_F, \quad (4.5)$$

where $\mathcal{A}$ consists of the 24 canonical poses, $\mathbf{R}_{\mathbf{P}}$ is the relative rotation from the current candidate canonical pose $\mathbf{P}$ to the reference calculated by the method of Umeyama [139], and $\mathbf{I}$ is the 3D identity matrix.

We conduct experiments on both classification and part segmentation tasks to extensively demonstrate the potential of the PCA-based canonical poses. Results are reported in Table 4.8. As presented, both classification and segmentation accuracies can be further enhanced if the ambiguities are ideally handled. For the segmentation task, the accuracy even approaches the full capacity of the network structure. Consequently, we can conclude that contrary to the common blame on the imperfect alignment, it is in fact the incorrect number of ambiguities that primarily lowers the accuracy.

	Strat. I	Strat. II	Strat. III	All possible canonical poses
Acc. (%)	89.5	90.3	87.2	91.6

Table 4.9 Classification accuracy (%) w.r.t. distinct disambiguation strategies on ModelNet40. The number of ambiguities from left to right are 4, 6, 0, 24, respectively.

4.6.2 Can disambiguation methods help?

Although the ambiguity cannot be addressed mathematically, there do exist some conventions on disambiguation. In this section, we explore whether these methods can benefit point cloud analysis tasks. Specifically, we conduct experiments with 3 disambiguation strategies:

- Strategy I: Determine the order of eigenvectors by sorting the corresponding eigenvalues in ascending order. This operation reduces the number of ambiguities to 4.
- Strategy II: Determine the sign of eigenvectors by letting more data lie on the positive half-axes. This strategy states that the principal axes should keep the same signs with the majority of the data vectors [16, 137]. Accordingly, the number of ambiguities is reduced to 6.
- Strategy III: Combine the aforementioned Strategies I and II together to eliminate all the ambiguities.

As shown in Table 4.9, these disambiguation methods are not ideal for point cloud analysis tasks, as there is a noticeable drop in accuracy compared to the case where all the ambiguities are considered. This is because objects within the same category may still lie in different poses after such disambiguation procedures. A detailed illustration is presented in Fig. 4.6, where poses are obtained with Strategy III.

4.6.3 Do inaccurate ambiguities have any effect?

We use Strategy I from Chapter 4.6.2 to determine the order of eigenvectors and assign all the 8 possible combinations of signs to it. By doing so, we obtain an accuracy of 89.2%, which is 2.4% lower than the one obtained with correctly identified ambiguities. Moreover, this result is also worse than the one obtained with correct but insufficient 4 sign ambiguities (Strategy I of Table 4.9), demonstrating that the incorrectly identified ambiguities do harm the performance.

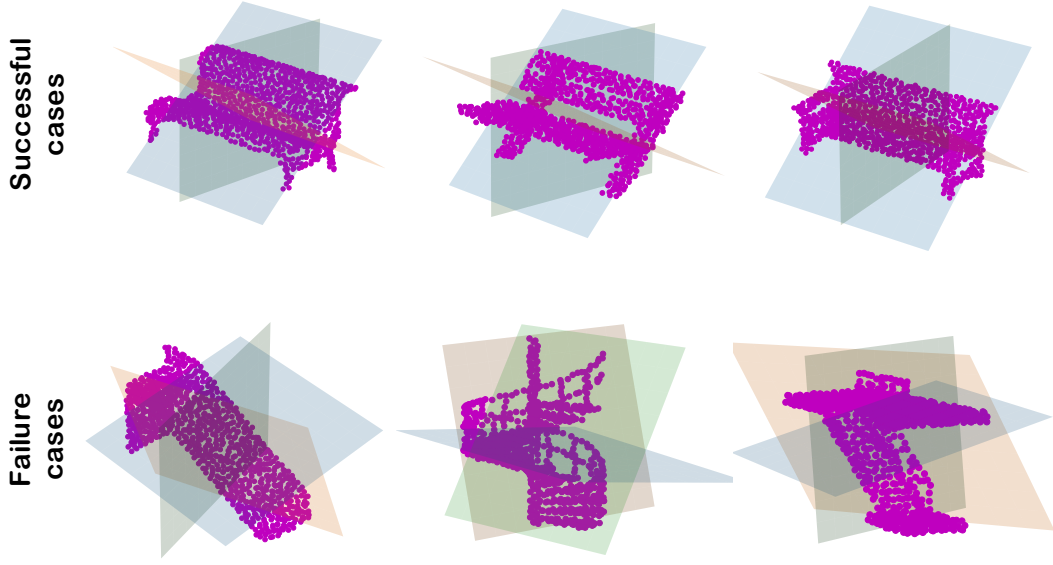


Fig. 4.7 Conceptual illustrations of the PCA-based alignments on non-symmetric objects. Planes characterize the three principal axes detected by PCA.

4.6.4 When does PCA perform the best?

We claim that PCA is effective for aligning objects that contain orthogonal plane-reflective symmetries. Specifically, for a symmetric object with orthogonal reflection planes, (part of) its principal axes would always be parallel to the normals of these planes [98, 21]. As simple proof, let us assume that a given centralized point cloud $\mathbf{P}$ is symmetric along a plane with unit normal $\mathbf{n}$. Then, as shown in Fig. 4.8, we can denote each pair of symmetric points as $\mathbf{x} + d\mathbf{n}$ and $\mathbf{x} - d\mathbf{n}$, where $\mathbf{x}$ belongs to the reflection plane and d is the point-to-plane distance. Thus the covariance matrix of $\mathbf{P}$ can be calculated as:

$$\begin{aligned} \mathbf{C} &= \sum_i (\mathbf{x}_i + d_i \mathbf{n}) (\mathbf{x}_i + d_i \mathbf{n})^T + (\mathbf{x}_i - d_i \mathbf{n}) (\mathbf{x}_i - d_i \mathbf{n})^T \\ &= \sum_i \mathbf{x}_i \mathbf{x}_i^T + 2d_i^2 \mathbf{n} \mathbf{n}^T. \end{aligned} \quad (4.6)$$

By multiplying $\mathbf{n}$ on both side of Eq. (4.6), we can obtain the following formulation with the property that $\mathbf{x}$ is always perpendicular to $\mathbf{n}$:

$$\mathbf{C} \mathbf{n} = \left(\sum_i \mathbf{x}_i \mathbf{x}_i^T + 2d_i^2 \mathbf{n} \mathbf{n}^T \right) \mathbf{n} = 2 \sum_i d_i^2 \mathbf{n}, \quad (4.7)$$

which is an eigen-problem identical to the one in PCA.

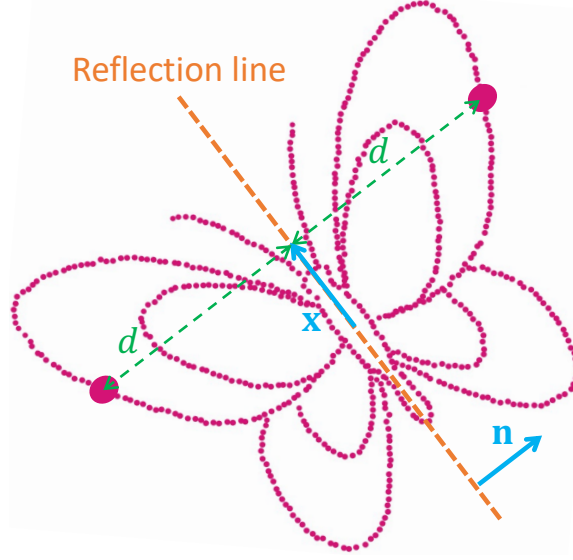


Fig. 4.8 A point cloud can be denoted as $\mathbf{x} + d\mathbf{n}$ and $\mathbf{x} - d\mathbf{n}$ given known symmetry.

Since Eq. (4.7) holds for all reflection planes, a point cloud with more than 2 orthogonal plane-reflections would have its principal axes uniquely determined up to the 24 sign and order ambiguities. Therefore, it is facile for PCA to perform excellent intra-class alignment on objects that consist of orthogonal plane-symmetries, such as desks, tables, and bathtubs. Owing to their man-made properties, these alignments are often presented as upright on the tabletop. Moreover, we observe that PCA can still achieve acceptable alignment on objects where the symmetries are only approximately satisfied, such as guitars, cups, and cars. However, this does not imply PCA would fail if orthogonal symmetries do not exist. As shown in the top row of Fig. 4.7, intra-class objects can still be aligned to similar poses owing to their similar structures. However, the alignments would be arbitrary if the intra-class shapes are not even similar in structure, as shown in the bottom row of Fig. 4.7, the aligned poses of the stairs are random due to their completely different structures.

4.7 Discussion and conclusions

This work explores the identification of proper pose ambiguities and its effects when PCA is used to achieve rotation-invariant point cloud analysis. A pose selector module is also developed for disambiguation. Although many efforts had been spent on developing powerful network structures or handcrafting descriptive features to complement the PCA-based canonical poses, our experiments indicate that it is the correctly identified ambiguities that can lead

to more significant performance boosts, evident from the noticeable accuracy enhancements on various datasets.

For future works, given the fact that our current disambiguation module does not take the number of symmetries of different point clouds into consideration, designing a new one that uses such information may help to push the accuracy one step further. Another thread of work lies in achieving more precise alignments by robustifying the vanilla PCA.

Chapter 5

Discussion and conclusion

This chapter presents a brief summary of the entire thesis and potential future works to further push forward the point cloud alignment accuracy.

5.1 Summary

This thesis aims at increasing point cloud alignment accuracy, wishing to benefit the downstream tasks of pose estimation and recognition. To achieve this goal, we propose to solve three problems, which are fundamental but commonly shared by existing methods. They are summarized in the following.

- The unknown relation between existing geometric and probabilistic alignment methods for pose estimation prevents them from benefiting from each other (Chapter 2).
- Existing alignment methods for pose estimation cannot present jointly feasible and outlier-robust estimations (Chapter 3).
- The number of ambiguities of the PCA-based intra-category alignment algorithm has not been sufficiently identified (Chapter 4).

A summary of the introduced algorithms and techniques in each chapter is presented in the following.

Chapter 2 We present a unified framework to explain the existing geometric and probabilistic alignment methods from an information-geometric perspective, making the advantages of these two dominant types of solution methods commutable. By leveraging the concept of information geometry, we demonstrate that both the geometric and probabilistic alignment

approaches can be explained in a geometric manner. *I.e.*, They can be seen as conducting the same iteratively matching and pose estimation procedure, but just on different spaces associated with different metrics. Moreover, we also propose to add an entropy term to the aligning objective function to map from Euclidean points to probabilistic distributions, which is required by varying probabilistic alignment methods for bandwidth estimation. Compared with the popularly used heuristic deterministic annealing algorithm, our bandwidth estimation strategy appears to be much more robust to hyper-parameters.

Chapter 3 We propose GSLR, a generalized formulation of the shuffled linear regression problem, making it robust to outliers. Specifically, we first extend the definition of permutation matrix to allow it to be rank deficient, and then add a rank constraint on it to avoid the trivial all-zero solution. With our modification, our proposal can not only theoretically guarantee the estimated correspondences between the two point clouds to be one-to-at-most-one (hence feasible), but also completely reject a certain pair of matching if it is suspected to be an outlier. To solve the GSLR problem, we technically rewrite the object function of GSLR and propose a simple yet effective alternative optimization algorithm based on it, which can achieve bi-optimal estimation within each iteration. Moreover, we also present detailed theoretical analyses from multiple perspectives to help to understand the GSLR problem and our optimization algorithm better. *I.e.*, we present the effective recovery condition of GSLR, prove its NP-hardness, and show that our optimization algorithm can converge globally within a sub-linear convergence rate.

Chapter 4 We analyze to show that the number of ambiguities of PCA has not been sufficiently identified so far when applied for intra-category point cloud alignment, and such insufficient identification of ambiguities diminishes the accuracy of the down-streaming deep learning-based recognition applications. Specifically, we classify the ambiguities into two parts: sign ambiguity and order ambiguity. For the former one, we show that half of them are in fact unnecessary in intra-category alignment for leading to improper rotations. For the latter, we claim that it has unfortunately been overlooked in existing works. Furthermore, to effectively handle such alignment ambiguities in recognition tasks, we also propose a pose selector module for disambiguation, which can be facilely mounted on most existing neural network structures. Experiments show that our sufficient identification of ambiguities can conspicuously improve the recognition accuracy, particularly on challenging real-world data.

5.2 Future works

Although the respective solutions methods we proposed can solve the three problems well, there exist rooms for improvement:

- **Unified explanation of existing methods** While our current framework can explain the geometric and probabilistic alignment algorithms well, there also exist some approaches that parameterize point clouds as graphs and consequently formulate the alignment problem into a graph matching problem [76]. It is interesting and beneficial to study how to include such methods in the unification framework.
- **Generalized shuffled linear regression** Our current optimization algorithm for GSLR sacrifices time efficiency to enhance the estimation accuracy. As a result, it cannot be applied to large-scale point clouds and hence cannot benefit some tasks such as dense 3D reconstruction. Therefore, improving time efficiency appears to be an important direction for future work.
- **Ambiguity analysis of intra-category alignment** As we analyzed in Chapter 4, PCA in fact acts as a symmetry detection algorithm in intra-category alignment. Therefore, its alignment results may be substandard if there is no symmetric part in the point cloud. As a solution, in the future, one may explore replacing PCA with some robust symmetry detection algorithms to improve alignment accuracy.

Apart from what we have addressed in this thesis, some other problems are waiting to be solved. For example, one commonly shared problem lies in the ability to handle huge point clouds. While down-sampling or dividing the point cloud are viable solutions, they may lead to either information loss or inconsistent estimations and diminish the alignment accuracy. Another common problem is the initialization of alignment algorithms. Specifically, given that most fine-alignment algorithms can only present a local minimum, feature-based global alignment methods are popularly conducted at the beginning to obtain a rough initialization. However, since the point cloud features are not convincingly stable and descriptive, feature-based alignment may also fail from case to case. Therefore, obtaining a reasonable initialization appears to be an important problem to solve.

References

- [1] Abubakar Abid, Ada Poon, and James Zou. Linear regression with shuffled labels. *ArXiv Preprint ArXiv:1705.01342*, 2017.
- [2] Abubakar Abid and James Zou. A stochastic expectation-maximization approach to shuffled linear regression. In *Proceedings of Annual Allerton Conference on Communication, Control, and Computing*, 2018.
- [3] Shunichi Amari. *Information geometry and its applications (Applied Mathematical Sciences Book)*, volume 194. 2016.
- [4] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7163–7172, 2019.
- [5] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *Transactions on Pattern Analysis and Machine Intelligence*, pages 698–700, 1987.
- [6] Hedy Attouch and Jérôme Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1):5–16, 2009.
- [7] Hedy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.
- [8] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proceedings of International conference on computer vision workshops (ICCV workshops)*, 2011.
- [9] Jordan Bell and Brett Stevens. A survey of known results and research areas for n-queens. *Discrete Mathematics*, 309(1):1–31, 2009.
- [10] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Transactions on Pattern Analysis and Machine Intelligence*, pages 239–256, 1992.
- [11] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748, 2003.
- [12] Tolga Birdal and Umut Simsekli. Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11105–11116, 2019.
- [13] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [14] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems.

- SIAM Journal on Optimization*, 17(4):1205–1223, 2007.
- [15] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.
 - [16] Rasmus Bro, Evrim Acar, and Tamara G Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008.
 - [17] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. 2008.
 - [18] Zhipeng Cai, Tat-Jun Chin, Huu Le, and David Suter. Deterministic consensus maximization with biconvex programming. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 685–700, 2018.
 - [19] Dylan Campbell and Lars Petersson. An adaptive data representation for robust point-set registration and merging. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 4292–4300, 2015.
 - [20] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
 - [21] Mohamed Chaouch and Anne Verroust-Blondet. Alignment of 3d models. *Graphical Models*, 71(2):63–76, 2009.
 - [22] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
 - [23] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
 - [24] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.
 - [25] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. *Object Recognition Supported by User Interaction for Service Robots*, 3:545–548, 2002.
 - [26] Tat-Jun Chin and David Suter. The maximum consensus problem: recent algorithmic advances. *Synthesis Lectures on Computer Vision*, 7(2):1–194, 2017.
 - [27] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of RANSAC family. In *Proceedings of British Machine Vision Conference (BMVC)*, 2009.
 - [28] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016.
 - [29] Haili Chui and Anand Rangarajan. A feature registration framework using mixture models. In *Proceedings of Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000.
 - [30] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Journal on Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.
 - [31] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH*, pages 303–312, 1996.
 - [32] Ketan Date and Rakesh Nagi. Gpu-accelerated hungarian algorithms for the linear assignment problem. *Parallel Computing*, 57:52–72, 2016.
 - [33] DQF De Menezes, Diego M Prata, Argimiro R Secchi, and José Carlos Pinto. A review on robust m-estimators for regression analysis. *Computers & Chemical Engineering*, 147:107254, 2021.

- [34] Jurgen A Doornik. Robust estimation using least trimmed squares. Technical report, Institute for Economic Modelling, Oxford Martin School, and Economics Department, University of Oxford, UK, 2011.
- [35] Benjamin Eckart, Kihwan Kim, and Kautz Jan. Eoe: Expected overlap estimation over unstructured point cloud data. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 747–755, 2018.
- [36] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640, 2017.
- [37] Golnoosh Elhami, Adam Scholefield, Benjamin Bejar Haro, and Martin Vetterli. Unlabeled sensing: Reconstruction algorithm and theoretical guarantees. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4566–4570, 2017.
- [38] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.
- [39] Jin Fang, Dingfu Zhou, Xibin Song, Shengze Jin, Ruigang Yang, and Liangjun Zhang. Rotpredictor: Unsupervised canonical viewpoint learning for point cloud classification. In *Proceedings of International Conference on 3D Vision (3DV)*, 2020.
- [40] Marcelo Fiori, Pablo Sprechmann, Joshua Vogelstein, Pablo Musé, and Guillermo Sapiro. Robust multimodal graph matching: Sparse coding meets graph matching. *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2013.
- [41] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [42] Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d’Aspremont. Convex relaxations for permutation problems. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2013.
- [43] Kent Fujiwara and Taiichi Hashimoto. Neural implicit embedding for point cloud analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [44] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11095–11104, 2019.
- [45] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.
- [46] John C. Gower and Garmt B. Dijkstra. *Procrustes problems*, volume 30. 2004.
- [47] Sébastien Granger and Xavier Pennec. Multi-scale em-icp: A fast and robust approach for surface registration. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 418–432, 2002.
- [48] Michael Greenspan and Mike Yurick. Approximate kd tree search for efficient icp. In *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, pages 442–448, 2003.
- [49] Aarsalane Chouaib Guidoum. Kernel estimator and bandwidth selection for density and its derivatives. *The kedd package, version*, 1, 2015.
- [50] Stefan Gumhold, Xinlong Wang, and Rob S MacLeod. Feature extraction from point clouds. In *Proceedings of International Meshing Roundtable, Sandia National*

- Laboratories*, 2001.
- [51] Asela Gunawardana and William Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.
 - [52] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Benamoun. Deep learning for 3d point clouds: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2020.
 - [53] Saeid Haghighatshoar and Giuseppe Caire. Signal recovery from unlabeled samples. *Transactions on Signal Processing*, 66(5):1242–1257, 2017.
 - [54] Dirk Hahnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with rfid technology. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, volume 1, pages 1015–1020, 2004.
 - [55] Frank Hampel. Robust inference. *Statistics Reference Online*, 2014.
 - [56] Frank R Hampel. The breakdown points of the mean combined with some rejection rules. *Technometrics*, 27:95–107, 1985.
 - [57] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2003.
 - [58] Erion Hasanbelliu, Luis Sanchez Giraldo, and Jose C Principe. Information theoretic shape matching. *Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2436–2451, 2014.
 - [59] Douglas M Hawkins. The feasible solution algorithm for least trimmed squares regression. *Computational Statistics & Data Analysis*, 17(2):185–196, 1994.
 - [60] Daniel J Hsu, Kevin Shi, and Xiaorui Sun. Linear regression without correspondence. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2017.
 - [61] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690*, 2021.
 - [62] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in Statistics*, pages 492–518. 1992.
 - [63] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.
 - [64] Kui Jia, Tsung-Han Chan, Zinan Zeng, Shenghua Gao, Gang Wang, Tianzhu Zhang, and Yi Ma. Roml: A robust feature correspondence approach for matching objects in a set of images. *International Journal of Computer Vision*, 117(2):173–197, 2016.
 - [65] Bing Jian and Baba C Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 2, pages 1246–1251, 2005.
 - [66] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2010.
 - [67] Hao Jiang, X Yu Stella, and David R Martin. Linear scale and rotation invariant matching. *Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1339–1355, 2010.
 - [68] Jie Jiang, Yu-Lin He, De-Xin Dai, and Joshua Zhexue Huang. A new kernel density estimator based on the minimum entropy of data set. *Information Sciences*, 491:223–231, 2019.
 - [69] Pileun Kim, Jingdao Chen, and Yong K Cho. Slam-driven robotic mapping and registration of 3d point clouds. *Automation in Construction*, 89:38–48, 2018.
 - [70] SeoHyun Kim, JaeYoo Park, and Bohyung Han. Rotation-invariant local-to-global representation learning for 3d point cloud. In *Proceedings of Conference on Neural*

- Information Processing Systems (NIPS)*, 2020.
- [71] Andreas Kuhn and Helmut Mayer. Incremental division of very large point clouds for scalable 3d surface reconstruction. In *Proceedings of International conference on computer vision workshops (ICCV workshops)*, 2015.
 - [72] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
 - [73] Pedro Larranaga, Cindy M. H. Kuijpers, Roberto H. Murga, Inaki Inza, and Sejla Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.
 - [74] Huu Le, Tat-Jun Chin, Anders Eriksson, Thanh-Toan Do, and David Suter. Deterministic approximate methods for maximum consensus robust fitting. *Transactions on Pattern Analysis and Machine Intelligence*, 43(3):842–857, 2019.
 - [75] Huu Le, Tat-Jun Chin, and David Suter. An exact penalty method for locally convergent maximum consensus. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1888–1896, 2017.
 - [76] Huu M Le, Thanh-Toan Do, Tuan Hoang, and Ngai-Man Cheung. Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
 - [77] Bo Li, Yingqiang Wang, Yu Zhang, Wenjie Zhao, Jianyuan Ruan, and Ping Li. Gp-slam: laser-based slam approach based on regionalized gaussian process map reconstruction. *Autonomous Robots*, 44(6):947–967, 2020.
 - [78] Hongdong Li and Richard Hartley. The 3d-3d registration problem revisited. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2007.
 - [79] Jiajin Li, Anthony Man-Cho So, and Wing-Kin Ma. Understanding notions of stationarity in nonsmooth optimization: A guided tour of various constructions of sub-differential for nonsmooth functions. *Signal Processing Magazine*, 37(5):18–31, 2020.
 - [80] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. A rotation-invariant framework for deep point cloud analysis. *arXiv preprint arXiv:2003.07238*, 2020.
 - [81] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on χ -transformed points. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2018.
 - [82] Wei Lian and Lei Zhang. Point matching in the presence of outliers in both point sets: A concave optimization approach. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
 - [83] Min Liu, Fupin Yao, Chiho Choi, Ayan Sinha, and Karthik Ramani. Deep learning 3d shapes using alt-az anisotropic 2-sphere convolution. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
 - [84] Kok-Lim Low. Linear least-squares optimization for point-to-plane icp surface registration. Technical Report 10, Chapel Hill, University of North Carolina, 2004.
 - [85] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of International Conference on Computer Vision (ICCV)*, 1999.
 - [86] Anna Lubiw. Some np-complete problems similar to graph isomorphism. *SIAM Journal on Computing*, 10(1):11–21, 1981.
 - [87] Jiayi Ma, Ji Zhao, Junjun Jiang, and Huabing Zhou. Non-rigid point set registration with robust transformation estimation under manifold regularization. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

- [88] Joao Maciel and Joao Paulo Costeira. A global solution to sparse correspondence problems. *Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.
- [89] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007.
- [90] Lena Maier-Hein, Alfred Michael Franz, Thiago R Dos Santos, Mirko Schmidt, Markus Fangerau, Hans-Peter Meinzer, and J Michael Fitzpatrick. Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error. *Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1520–1532, 2011.
- [91] Manuel Marques, Marko Stošić, and João Costeira. Subspace matching: Unique solution to point matching with geometric constraints. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1288–1294, 2009.
- [92] Eleonora Maset, Federica Arrigoni, and Andrea Fusiello. Practical and efficient multi-view matching. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 4568–4576, 2017.
- [93] Roy Mathias. *The Linear Algebra a Beginning Graduate Student Ought to Know*. 2006.
- [94] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [95] Goeffrey J McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.
- [96] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *Transactions on Graphics*, 38(6), 2019.
- [97] Zhe Min, Jiaole Wang, and Max Q-H Meng. Robust generalized point cloud registration using hybrid mixture model. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pages 4812–4818, 2018.
- [98] Predrag Minovic, Seiji Ishikawa, and Kiyoshi Kato. Symmetry identification of a 3-d object represented by octree. *Transactions on Pattern Analysis and Machine Intelligence*, 15(5):507–514, 1993.
- [99] Ihab S Mohamed, Alessio Capitanelli, Fulvio Mastrogiovanni, Stefano Rovetta, and Renato Zaccaria. A 2d laser rangefinder scans dataset of standard eur pallets. *Data in brief*, 24:103837, 2019.
- [100] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.
- [101] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of Symposium on Security and Privacy*, 2008.
- [102] Amin Nejatbakhsh and Erdem Varol. Neuron matching in c. elegans with robust approximate linear regression without correspondence. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2021.
- [103] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *Transactions on Graphics*, 31(4), 2012.
- [104] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 26, 2013.

- [105] Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Denoising linear models with permuted data. In *Proceedings of International Symposium on Information Theory (ISIT)*, 2017.
- [106] Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *Transactions on Information Theory*, 64(5):3286–3300, 2017.
- [107] Lei Peng, Guangyao Li, Mang Xiao, and Li Xie. Robust cpd algorithm for non-rigid point set registration based on structure information. *PloS one*, 11(2), 2016.
- [108] Rosalie Plantefeve, Igor Peterlik, Nazim Haouchine, and Stéphane Cotin. Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery. *Annals of biomedical engineering*, 44(1):139–153, 2016.
- [109] François Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *International Journal of Robotics Research*, 31(14):1705–1711, 2012.
- [110] François Pomerleau, M. Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *International Journal of Robotics Research*, 31(14):1705–1711, 2012.
- [111] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point cnn with spherical harmonics kernels. In *Proceedings of International Conference on 3D Vision (3DV)*, 2019.
- [112] Timo Pylvänäinen, Jérôme Berclaz, Thommen Korah, Varsha Hedau, Mridul Aanjaneya, and Radek Grzeszczuk. 3d city modeling from street-level data for augmented reality applications. In *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 238–245, 2012.
- [113] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [114] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [115] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.
- [116] Wanli Qiao et al. Asymptotics and optimal bandwidth for nonparametric estimation of density level sets. *Electronic Journal of Statistics*, 14(1):302–344, 2020.
- [117] Yu Qiao and Nobuaki Minematsu. A study on invariance of f -divergence and its application to speech recognition. *Transactions on Signal Processing*, 58(7):3884–3890, 2010.
- [118] Yongming Rao, Jiwen Lu, and Jie Zhou. Spherical fractal convolutional neural networks for point cloud recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [119] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *Transactions on Graphics*, 37(6), 2018.
- [120] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. 2005.
- [121] Szymon Rusinkiewicz. A symmetric objective function for icp. *Transactions on Graphics*, 38(4), 2019.

- [122] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2009.
- [123] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Proceedings of Robotics: Science and Systems (RSS)*, page 435, 2009.
- [124] Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 3d-rotation-equivariant quaternion neural networks. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.
- [125] Takaaki Shiratori, Jérôme Berclaz, Michael Harville, Chintan Shah, Taoyu Li, Yasuyuki Matsushita, and Stephen Shiller. Efficient large-scale point cloud registration using loop closures. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 232–240, 2015.
- [126] Bernard W Silverman. *Density estimation for statistics and data analysis*. 2018.
- [127] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35(2):876–879, 1964.
- [128] Martin Slawski, Emanuel Ben-David, et al. Linear regression with sparsely permuted data. *Electronic Journal of Statistics*, 13(1), 2019.
- [129] Martin Slawski, Emanuel Ben-David, and Ping Li. A two-stage approach to multivariate linear regression with sparsely mismatched data. *Journal of Machine Learning Research*, 2019.
- [130] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to orient surfaces by self-supervised spherical cnns. *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2020.
- [131] Marko Stošić, Manuel Marques, and João Paulo Costeira. Convex solution of a permutation problem. *Linear Algebra and its Applications*, 434(1):361–369, 2011.
- [132] Todor Stoyanov, Martin Magnusson, and Achim J Lilienthal. Point set registration through minimization of the l2 distance between 3d-ndt models. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pages 5196–5201, 2012.
- [133] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [134] Xiao Sun, Zhouhui Lian, and Jianguo Xiao. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In *Proceedings of International Conference on Multimedia (ACMMM)*, 2019.
- [135] PW Theiler and K. Schindler. Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. *Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:173–178, 2012.
- [136] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotequi, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.
- [137] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.
- [138] Manolis Tsakiris and Liangzu Peng. Homomorphic sensing. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.
- [139] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *Transactions on Pattern Analysis and Machine Intelligence*, pages

- 376–380, 1991.
- [140] Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing with random linear measurements. *Transactions on Information Theory*, 64(5):3237–3253, 2018.
 - [141] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.
 - [142] Matthias Vestner, Zorah Löhner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, et al. Efficient deformable shape correspondence via kernel matching. In *Proceedings of International Conference on 3D Vision (3DV)*, 2017.
 - [143] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
 - [144] Anton Volgenant. Solving the k-cardinality assignment problem by transformation. *European Journal of Operational Research*, 157(2):322–331, 2004.
 - [145] Jayakorn Vongkulbhisal, Fernando De la Torre, and Joao Paulo Costeira. Discriminative optimization: Theory and applications to computer vision. *Transactions on Pattern Analysis and Machine Intelligence*, 41(4):829–843, 2018.
 - [146] Fei Wang, Baba C Vemuri, and Anand Rangarajan. Groupwise point pattern registration using a novel cdf-based jensen-shannon divergence. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1283–1288, 2006.
 - [147] Fudong Wang, Nan Xue, Jin-Gang Yu, and Gui-Song Xia. Zero-assignment constraint for graph matching with outliers. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
 - [148] Xiaogang Wang and James V Zidek. Derivation of mixture distributions and weighted likelihood function as minimizers of kl-divergence subject to constraints. *Annals of the Institute of Statistical Mathematics*, 57(4):687–701, 2005.
 - [149] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Transactions on Graphics*, 38(5):1–12, 2019.
 - [150] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
 - [151] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.
 - [152] Chenxi Xiao and Juan Wachs. Triangle-net: Towards robustness in point cloud learning. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2021.

- [153] Zelin Xiao, Hongxin Lin, Renjie Li, Lishuai Geng, Hongyang Chao, and Shengyong Ding. Endowing deep 3d models with rotation invariance based on principal component analysis. In *Proceedings of International Conference on Multimedia and Expo (ICME)*, 2020.
- [154] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.
- [155] Yangyang Xu and Wotao Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.
- [156] Satyendra Singh Yadav, Paulo Alexandre Crisóstomo Lopes, Aleksandar Ilic, and Sarat Kumar Patra. Hungarian algorithm for subcarrier assignment problem using gpu and cuda. *International Journal of Communication Systems*, 32(4), 2019.
- [157] Eunho Yang, Aurélie C Lozano, and Aleksandr Aravkin. A general family of trimmed estimators for robust high-dimensional data analysis. *Electronic Journal of Statistics*, 12(2):3519–3553, 2018.
- [158] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *Transactions on Robotics*, 37(2):314–333, 2020.
- [159] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, 2015.
- [160] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [161] Ruixuan Yu, Xin Wei, Federico Tombari, and Jian Sun. Deep positional and relational feature learning for rotation-invariant point cloud analysis. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.
- [162] Willard I Zangwill. *Nonlinear programming: a unified approach*. 1969.
- [163] Hang Zhang, Martin Slawski, and Ping Li. Permutation recovery from multiple measurement vectors in unlabeled sensing. In *Proceedings of International Symposium on Information Theory (ISIT)*, 2019.
- [164] Zhiyuan Zhang, Binh-Son Hua, David W Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *Proceedings of International Conference on 3D Vision (3DV)*, 2019.
- [165] Chen Zhao, Jiaqi Yang, Xin Xiong, Angfan Zhu, Zhiguo Cao, and Xin Li. Rotation invariant point cloud classification: Where local geometry meets global topology. *arXiv preprint arXiv:1911.00195*, 2019.
- [166] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 127–134, 2012.
- [167] Jie Zhou, Xinke Ma, Li Liang, Yang Yang, Shijin Xu, Yuhe Liu, and Sim-Heng Ong. Robust variational bayesian point set registration. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 9905–9914, 2019.
- [168] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *Transactions on Graphics*, 32(4):112, 2013.
- [169] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.
- [170] Zhiyong Zhou, Jian Zheng, Yakang Dai, Zhe Zhou, and Shi Chen. Robust non-rigid point set registration using student’s-t mixture model. *PloS one*, 9(3):e91381, 2014.

-
- [171] Jihua Zhu, Congcong Jin, Zutao Jiang, Siyu Xu, Minmin Xu, and Shanmin Pang. Robust point cloud registration based on both hard and soft assignments. *Optics & Laser Technology*, 110:202–208, 2019.

