Faculty of Engineering and Information Technology
University of Technology Sydney

Cross-source point cloud matching by exploring structure property

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Xiaoshui Huang

January 2019

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Production Note:
Signature removed prior to publication.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Associate Prof. Jian Zhang for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I also would like to appreciate my co-supervisor Associate Prof. Qiang Wu for providing me with continuous support throughout my PhD study and research. Without his professional guidance and persistent help, this thesis would not have been possible.

I thank my fellow labmates in Global Big Data and Technology Center: Junjie Zhang and Yifan Zuo. for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last three years.

I place on record, my sense of gratitude to the team leader, Lixin Fan in Nokia Technologies for his expert and sincere help in the project.

Last but not the least, I would like to thank my family: my parents and my girlfriend, for their unconditional support, both financially and emotionally throughout the whole PhD studying.

Xiaoshui Huang
July 2018 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers published

- **Xiaoshui Huang**, Lixin Fan, Qiang Wu, Jian Zhang, Chun Yuan (2017), A coarse-to-fine algorithm for matching and registration in 3D cross-sourced point clouds. *in* 'Transactions on Circuits and Systems for Video Technology (**TCSVT**)', full paper accepted.

- **Xiaoshui Huang**, Jian Zhang, Lixin Fan, Qiang Wu, Chun Yuan (2017), A Systematic Approach for Cross-Source Point Cloud Registration by Preserving Macro and Micro Structures. *in* 'IEEE Transactions on Image Processing (**TIP**)', vol. 26, no. 7, pp. 3261-3276, July 2017.

- **Xiaoshui Huang**, Jian Zhang, Qiang Wu, Lixin Fan, Chun Yuan (2016), A coarse-to-fine algorithm for registration in 3D street-view cross-source point clouds. *in* 'International Conference on Digital Image Computing: Techniques and Applications (**DICTA**)' (pp. 1-6). IEEE.

- **Xiaoshui Huang**, Lixin Fan, Jian Zhang, Qiang Wu, and Chun Yuan (2016), Real Time Complete Dense Depth Reconstruction for a Monocular Camera. *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (**CVPRW**)' (pp. 32-37). IEEE.

- **Xiaoshui Huang**, Chun Yuan, and Jian Zhang (2015), Graph Cuts Stereo Matching Based on Patch-Match and Ground Control Points

Constraint. *in* 'Pacific Rim Conference on Multimedia (**PCM**)' (pp. 14-23). Springer International Publishing.

- **Xiaoshui Huang**, Jian Zhang, Qiang Wu, Chun Yuan, and Lixin Fan (2015), Dense Correspondence Using Non-local DAISY Forest. *in* 'International Conference on Digital Image Computing: Techniques and Applications (**DICTA**)' (pp. 1-8). IEEE.

- Shoujin Wang, Liang Hu, Longbing Cao, **Xiaoshui Huang**, Defu Lian, Wei Liu. (2018), Attention-based Transactional Context Embedding for Next-Item Recommendation. *in* 'The Thirty-Second AAAI Conference on Artificial Intelligence (**AAAI-18**)'.

Papers to be submitted

- **Xiaoshui Huang**, Jian Zhang, Lixin Fan, Qiang Wu, Chun Yuan (2017), Cross-source Point Cloud Registration by using weak regional affinity and pixel-wise refinement. To be submitted to 'IEEE Transactions on Multimedia (**TMM**)'.

- **Xiaoshui Huang**, Jian Zhang, Lixin Fan, Qiang Wu, Chun Yuan (2017), PSTN: learning a rotation-invariant descriptor for cross-source point cloud matching. To be submitted to 'IEEE Transactions on Multimedia (**TMM**)'.

Patents Granted

- Lixin Fan, **Xiaoshui Huang**, Qiang Wu, Jian Zhang. "Point cloud matching process". U.K. Patent: GB2550567. issued date: 2017-11-29.

Abstract

Cross-source point cloud are 3D data coming from heterogeneous sensors. The matching of cross-source point cloud is extremely difficult because they contain mixture of different variations, such as missing data, noise and outliers, different viewpoint, density and spatial transformation. In this thesis, cross-source point cloud matching is solved from three aspects, utilizing of structure information, statistical model and learning representation. Chapter 1 introduces the value of cross-source point cloud registration and summarizes the key challenges of cross-source point cloud registration problem. Chapter 2 reviews the existing registration methods and analyse their limitation in solving the cross-source point cloud registration problem. Chapter 3 proposes two algorithms to discuss how to utilize structure information to solve the cross-source point cloud registration problem. In the first part of this chapter, macro and micro structures are extracted based on 3D point cloud segmentation. Then, these macro and micro structure components are integrated into a graph. With novel descriptors generated, the registration problem is successfully converted into graph matching problem. In the second part, weak region affinity and pixel-wise refinement are proposed to solve the cross-source point cloud. These two components are unified represented into a tensor space and the registration problem is converted into tensor optimization problem. In this method, the tensor space is updated when the transformation matrix is updated to get feedback from the recent transformation estimation step. Chapter 4 discusses how to utilize the statistical distribution of cross-source point cloud to solve matching problem.

The goal is to find the potential matching region and estimate the accurate registration relationship. In this chapter, ensemble of shape functions (ESF) is utilized to select potential regions and a novel registration is proposed to solve the matching problem. For the registration, Gaussian mixture models (GMM) is selected as our mathematical tool. However, different to previous GMM-based registration methods, which assume a GMM for each point cloud, the proposed algorithm assumes a virtual GMM and the cross-source point clouds are samples from the virtual GMM. Then, the transformation is optimized to project the samples into a same virtual GMM. When the optimization is convergence, both the parameters of GMM and the transformation matrices are estimated. In Chapter 5, a deep learning method is proposed to represent the local structure information. Because of arbitrary rotation in cross-source point clouds, a rotation-invariant 3D representation method is proposed to robust represent the 3D point cloud although there are arbitrary rotation and translation. Also, there is no robust keypoints in these cross-source point cloud because of they come from heterogenous sensors, train the network is very difficult. A region-based method is proposed to generate regions for each point cloud and synthetic labelled dataset is constructed for training the network. All these algorithms are aimed to solve the cross-source point cloud registration problem. The performance of these algorithms is tested on many datasets, which shows the effective and correctness. These algorithms also provide insightful knowledge for 3D computer vision workers to process 3D point cloud.

Chapter 1

Introduction

1.1 Background

There is currently a wide diversity of sensors and techniques for capturing point clouds, for example, LiDAR, Kinect, Intel@RealSense, range cameras, structure from motion (SFM) and simultaneous localization and mapping (SLAM). To gain high performance in applications, current vision systems usually contain more than one type of sensor, such as LiDAR and the RGB camera on a autonomous driving vision system, and the Kinect and Stereo cameras on the Robotics vision system. Some applications can obtain improved accuracy by using different kinds of point clouds which are captured by different types of sensors. Cross-source point clouds are those point clouds captured by different types of sensors. For example, one scan of a point cloud is from Kinect and the other scan of a point cloud is from SFM. Due to the different sensor modalities, cross-source point clouds are much challenging, the reason being that, given two point clouds acquired by different sensing methods, they will have completely different densities, noise models and uncertain outliers, missing data, partial overlap and scale variation. Registering two point clouds together and recovering the relevant transformation model is extreme challenges. In this thesis, the registration problem of cross-source point clouds is analyzed from three aspects and three kinds of solutions are

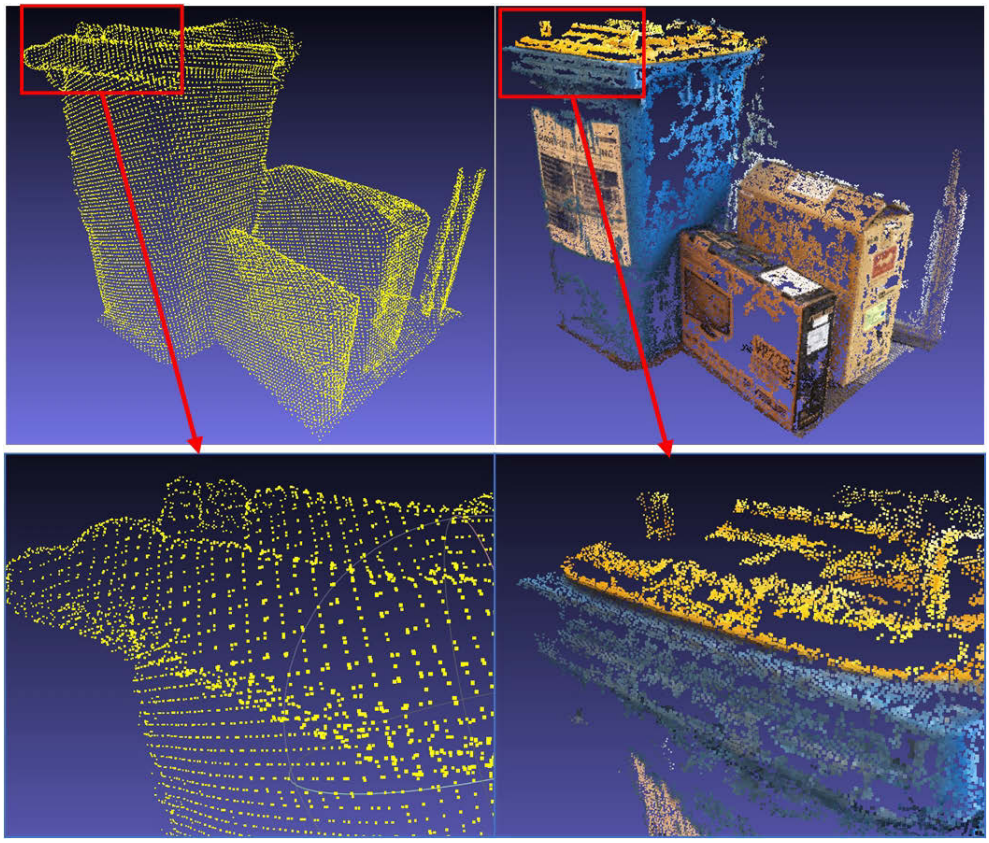

Figure 1.1: The top row is two cross-source point clouds captured by Kinect and RGB camera. The bottom row is the detail of one part.

proposed.

For point cloud registration, most of the previous methods mainly focus on the same-source point clouds (e.g. the registration data are all from Kinect), which achieve local and global registration results. Local registration methods are a method which only considers part of the points or constraints, it is usually efficient while suffers suboptimal. Global registration methods consider the overall structure or upper and lower bounds, and they usually obtain global registration results. However, neither the existing local matching-based method nor the global matching-based method can sort out the case of cross-source point cloud registration. Either existing local registration or global registration methods assume strong structure consistency

between the two point-cloud sets being registered. However, this is not the case for cross-source point cloud registration in which structure consistency is weak. The reliable correspondences which can be located between two sets is sparse. Larger inconsistencies caused by different point cloud density, noise models and various outliers significantly degrade the estimation on rigid transformation.

Cross-source point cloud registration is a new research topic and only two methods are currently available(Peng, Wu, Fan, Zhang, You, Lu & Yang 2014*a*, Mellado, Dellepiane & Scopigno 2015). (Peng et al. 2014*a*) uses an ensemble of signal function (ESF) descriptor to coarsely select potential region and use iterative closet point(ICP) to refine the ranking of the potential regions. The registration part is done using ICP. In (Mellado et al. 2015, Lin, Tamaki, Zhao, Raytchev, Kaneda & Ichii 2014), they use a continuous descriptor to estimate the scale and the registration is done using RANSAC. However, ICP is sensitive to initialization and RANSAC is sensitive to noise and high computation complexity. These methods all fail in our datasets. In this thesis, we aim to solve the cross-source point cloud registration problem.

Figure 1.1 shows two cross-source point clouds with two monitors and an audio equipment, which illustrates the challenges confronting a robust registration method. We will talk about these challenges in details in the below part. As demonstrated in our experiments, these combined challenges often give rise to adverse scenarios in which many existing registration methods fail miserably.

Despite the large variations in cross-source point clouds, our human vision system seems able to align them effortlessly with high accuracy. This is probably due to the fact that humans exploit the similarities between the *structures of two cross-source point clouds* instead of the detailed points. Motivated by this insight, in this thesis, we are going to solve the cross-source point cloud registration problem by exploring the structure property. We are going to exploring the structures from three aspects: directly utilize

macro and micro structures, use statistic models to describe the structures and use deep learning to generate descriptor for the structures.

At the same time, point cloud is a type of data to describe the 3D world in the computer. It has only position information and is a typical irregular data. When the point clouds are captured from different types of 3D sensors (e.g. stereo camera and LiDAR), they are cross-source point clouds. Because of different sensor modalities, cross-source point clouds contain large variations including noise, outliers, density, missing data, partial overlap, viewpoint changing and scale. We name these variations in cross-source point clouds as the cross-source problem. These variations make the computer vision problem in cross-source point clouds very challenging. Registration (also known as alignment) is a basic step in point cloud generation and it can generate more comprehensive and larger point clouds. Registration of cross-source point clouds is inevitable to confront the cross-source problem. We argue that cross-source point clouds can be aligned automatically with high accuracy and efficiency.

1.2 Key challenges

As previously discussed, the key challenges of cross-source point cloud registration are as follows:

1.2.1 Noise, outliers, density

Because of different sensor modalities, different sensors can capture different 3D point signals at different position. This theory produces the noise, outliers and density variations. For example, cross-source point clouds come from LiDAR and stereo camera. LiDAR is a proactive sensor which usually captures 3D points evenly; while a stereo camera captures 3D points from a feature-based 3D reconstruction. It is very hard to capture the same 3D point in the same position and they always capture a different amount of points at a region of the same size.

1.2.2 Missing data

Due to the different limitations of different types of sensors, they face the missing data problem in different regions. For example, LIDAR works well in all light conditions, but starts failing with increases in snow, fog, rain, and dust particles in the air due to its use of light spectrum wavelengths. In contrast, a stereo camera starts to fail in texture-less regions. The different sensor limitations in capturing 3D cause variations in the missing data.

1.2.3 Partial overlap and viewpoint changing

Because it is difficult to put different 3D sensors in the same place at two acquisition processes, the sensors may capture the 3D scenes from different viewpoints. Therefore, there is partial overlap and changing viewpoints.

1.2.4 Scale variation

Because a cross-source point cloud comes from different types of 3D sensors, the coordinate system between the 3D sensors is totally different. There- fore, there is usually scale variation between cross-source point clouds. However, because of the other variations of the cross-source problem, it is very hard to calibrate and the scale estimation becomes extremely difficulty.

1.3 Applications

Cross-source point cloud registration is a basic technology in the 3D field. It facilitates 3D data production and its accuracy is higher than the accuracy of the applications discussed in the following sub-sections. In addition to the general importance of basic technology research, cross-source point cloud registration can be directly applied to the following topics:

1.3.1 Computer vision

Cross-source point cloud searching and localization: the advances in the area of 3D sensors has made them popular in daily life. Cross-source point cloud registration is an easy and accurate way to obtain a final searching and localization result. Taking, for example, a city scale LiDAR database 3D model, we first capture a 3D scene using our mobile stereo camera and we can search what we are seeing and know accurately within 1cm where we are in the city-scale model.

3D completion: The missing data problem occurs in some regions when using a certain type of sensor however other types of sensors can produce 3D points in these regions. We can conduct registration to overcome the problem. After the registration, a completed point cloud of this region is produced.

3D cross-source template matching: Sometimes, we need to conduct template matching to detect a perfect matching region with a given point cloud. Cross-source point cloud registration is a technique to tackle this problem.

3D reconstruction: Using different sensors, we can build more accurate and comprehensive 3D models. For example, LiDAR is known for its accuracy but it is very expensive to use to increase the resolution. Usually, we use a fused solution and obtain both accurate and cheap 3D point clouds. To do this, cross-source point cloud registration is a necessary technique.

1.3.2 Medical assistance

We capture many different 3D scans using different sensors and obtain different properties of the human body. With cross-source point cloud registration techniques, we can easily view the scanned data, easily communicate with patients and easily visualize for device implantation.

1.3.3 Remote sensing

We can use different kinds of sensors to sense different properties of an area and apply the cross-source point cloud registration method to fuse these data into a more comprehensive point cloud.

1.4 Issues

Based on the aforementioned current research limitations, we present these following research issues:

1.4.1 Cross-source point cloud acquisition

Cross-source point clouds are 3D data from different types of sensors. There are many available 3D techniques. To conduct research on 3D cross- source point cloud registration, the challenges need to be clarified firstly. To achieve this goal, we conduct several research works on 3D data generation from recent commonly used sensors (RGB camera and Kinect). With these 3D point clouds on hand, the problem of cross-source point clouds registration can be analyzed. In this section, we discuss point clouds generated by stereo matching, SLAM and KinectFusion. Stereo matching is the key technique used by many recent stereo cameras and SLAM is the key techniques used for large- scale outdoor scene reconstruction. KinectFusion is a popular sensor for indoor 3D reconstruction. Based on the cross-source point cloud production work, the cross-source problem is summarized as a combination of noise, outliers, density, missing data, partial overlap and scale.

1.4.2 Structure-based methods

3D point clouds only record position information describing the surface of an object or scene. Based on our observations, the overall structure information is similar although there are large variation in cross-source problems. Figure 1.2 shows an example of cross-source point clouds and the points describing

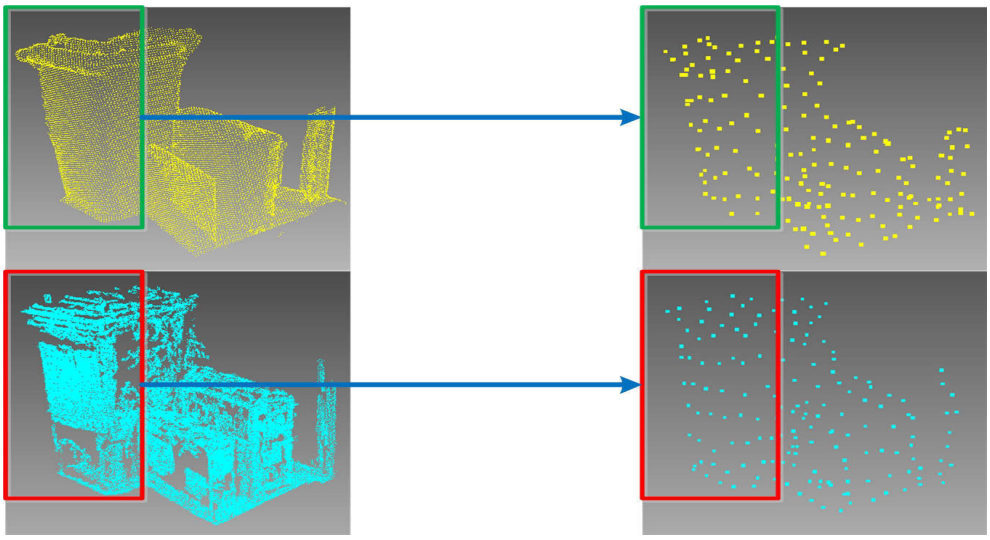

Figure 1.2: The left column is a set of cross-source point clouds captured by KinectFusion and VSFM. The right column is the structure points of the corresponding point clouds.

their structures. Figure 1.2 shows the structure is similar even though their original point clouds contain large variations, as indicated by the rectangular regions. In this section, we introduce how to extract the structure points and how to use the structure information to solve the registration problem. First, a novel structure-based method is proposed; and second, the structure information is ensembled into tensors and the registration problem is converted into the rank-1 tensor optimization problem. The latter is a faster version of the structure-based method.

1.4.3 Statistic model methods

There are large challenges when undertaking research on cross-source point clouds, such as how to solve the cross-source problem in other domains. For example, if there is large amount of missing data in cross-source point clouds, we can build their 3D models and use these models to conduct the registration. Therefore, the point cloud registration problem is transferred

to model alignment. In this thesis, we argue that the statistic models of cross-source point clouds are similar although they contain large cross-source variations. Particularly, Gaussian mixture models (GMM) are built for each point cloud. Then, the registration problem is solved by optimizing an energy function. When the energy function is convergent, both the parameters of GMM and the transformation matrix are estimated.

1.4.4 3D rotation-invariant descriptor based on deep learning

Recently, deep learning has shown great potential in representation. What is the representation ability of deep learning in cross-source point clouds Moreover, more attention should be paid to rotation ability because large rotation variation is often existed in registration problems. In this thesis, a deep learning strategy is applied to learn the representation for 3D cross-source point clouds. Specifically, 3D point cloud is converted to volumetric data and 3D CNN is applied to extract deep features.

1.5 Research Contributions

- This thesis proposes an algorithm to extract the structure information of cross-source point clouds. The algorithm can deal with density and part of the missing data problem. (chapter 3);
- This thesis proposes an algorithm to integrate the macro/micro structures into graph and propose a graph matching algorithm to achieve the initial structure registration results. Then, the cross-source point cloud registration is obtained by simply using RANSAC and ICP to conduct the further refinement. (chapter 3);
- This thesis proposes a tensor-based framework to convert the registration problem into the tensor optimization problem. Then, the optimization problem can be efficiently solved by rank-1 tensor optimiza-

tion algorithm. This algorithm is also used the structure concept, but with fast efficiency and comparable accuracy. (chapter 3);

- This thesis proposes an algorithm to use clustering and Gaussian mixture models(GMM) to simultaneously estimate the transformation matrix and the GMM of the point clouds. The theory of this algorithm is that the statistic model of the cross-source point cloud are similar. (chapter 5);
- This thesis proposes an enhanced algorithm to estimate the scale by using GMM(chapter 5);
- This thesis proposes an algorithm to generate rotation-invariant descriptor by using deep learning network. (chapter 6).

1.6 Thesis Structure

The thesis is structured as follow:

Chapter 2 provides a literature review on the definition of registration and its bivariate families and various models. The direct method and its variations are reviewed. We first review point-point based registration methods, then we review pair-constraint based and high-order constraint registration methods. The relationship between these different methods is also reviewed. In addition, we review stereo matching, SLAM, graph matching and 3D descriptors. Lastly, the applications of cross-source point cloud registration are reviewed.

Chapter 3 proposes to use micro and macro structures to conduct cross-source point cloud registration. Firstly, the 3D segmentation method is developed to extract the structure elements, then a graph is used to integrate these structures into an unified mathematical format. Then, the cross-source point cloud registration problem is converted into the graph matching problem. We then develop a graph matching method considering rigid geometric constraints to conduct the graph matching. The results show that the

proposed method using the micro and macro structures can conduct the registration successfully and outperforms other state-of-the-art point cloud registration methods.

Chapter 5 presents a registration method using statistic models, specifically, Gaussian mixture models (GMM). This model uses the cluster concept to align two different types of point clouds into a virtual GMM. The method optimizes an energy function which contains both model parameters and transformation parameters. When the energy function converges, both the parameters are estimated. Our method can successfully achieve cross-source point cloud registration. The results show that our method is better than other point cloud registration methods.

Chapter 6 proposes novel 3D descriptors using deep learning. Recently, although convolution neural networks have been shown to be translation invariant through the pooling layer, they show no ability to handle rotation variations. In 3D cross-source point cloud registration, large rotation is a common existing problem. In this thesis, we develop a learning method to extract 3D rotation-invariant descriptors to embed into the registration method. The new 3D descriptor is robust to outliers, missing data and noise. The results show that our descriptor is much better than other 3D descriptors. Also, our novel descriptor achieves high registration accuracy even with a very simple registration method (RANSAC).

Chapter 2

Literature review and Mathematical background

As discussed in Chapter 1, cross-source point cloud registration is a challenging problem and an emerging new topic. The existing works related to point cloud registration are reviewed in this chapter. Firstly, the existing methods are reviewed. Secondly, several mathematical tools are reviewed and the connection between these mathematical and registration problems is discussed.

2.1 Related works

3D Point sets matching on cross source has endured much development for a long period. There are many ways to tackle this problem. The preview work can be classified into the following types:

2.1.1 Direct methods

Direct point set registration methods usually minimize the Euclidean distance between nearby points. The most popular approach is the Iterative closest point (ICP) (Best & McKay 1992) algorithm, which alternates between estimating the point correspondence and estimating the transformation

matrix for a given correspondence (Newcombe, Izadi, Hilliges, Molyneaux, Kim, Davison, Kohi, Shotton, Hodges & Fitzgibbon 2011, Huber & Hebert 2003, Torsello, Rodola & Albarelli 2011). The vanilla ICP method (Best & McKay 1992) relies on the assumption that all points have pairwise counterparts between two sets and are very sensitive to a given initialization. The method is widely used in same-source and cross-source registration. Iterative non-rigid point set matching has improved ICP by incorporating outlier detection in the iterative correspondence estimation steps (Chui & Rangarajan 2000). The above methods are all heuristic methods, hence they cannot guarantee the global optimality of the solutions. Go-ICP (Yang, Li & Jia 2013) provides a globally optimal solution to ICP in 3D Euclidean registration, which combines ICP with a branch-and-bound (BnB) scheme. Similarly, GOGMA (Campbell & Petersson 2016) combines Gaussian mixture model (GMM) with a BnB scheme. These global optimal methods are sensitive to scale problem. A roots-finding technique was used in (Ho, Peter, Rangarajan & Yang 2009) for affine invariant point set registration. The method is sensitive to outliers due to use of moments. To tackle outlier sensitive problem in (Ho et al. 2009), (Ma, Zhao, Tian, Tu & Yuille 2013) proposes a method that uses an L2E-estimator and ICP. The method of (Ma et al. 2013) is suitable both for 2D and 3D situations. In the 2D instance, shape context is used as descriptor and the Hungarian method is used for matching with the χ^2 test statistic as the cost measure. In the 3D instance, the spin image can be used as a feature descriptor, where the local similarity is measured by an improved correlation coefficient. (Ma et al. 2013) uses L2E, which is particularly appropriate for analyzing massive data sets when data cleaning is impractical. With the ICP refinement algorithm, this algorithm robustly estimates transformation f with noise and outlier points. Also, the initial correspondences do not need to be highly accurate. The experimental results show that this algorithm has good performance under deformation, occlusion, rotation, noise and outliers. However, experiments have only been conducted on the algorithm in same-source situation; in the cross-source sit-

uation, the L2E estimator may face the problem of large outliers. Despite these improvements to the ICP method, the direct registration approaches above are intrinsically sensitive to missing data, large variations in point density, and scale differences, thus rendering them useless for cross-source point cloud registration.

In contrast to these ICP-based methods, registration amounts to solving a global problem to find the best aligning rigid transform over the 6DOF space of all possible rigid transforms comprised of translations and rotations when scan pairs start in arbitrary initial poses. Since aligning rigid transforms are uniquely determined by three pairs of (non-degenerate) corresponding points, one popular strategy is to invoke RANSAC (Fischler & Bolles 1981) to find the aligning triplets of point pairs (Chen, Hung & Cheng 1999). This approach, however, regularly degrades to its worst case $O(n^3)$ complexity in the number n of data samples in presence of partial matching with low overlap. Various alternatives to RANSAC have been proposed to counter the cubic complexity, such as hierarchical representation in the normal space (Diez, Martí & Salvi 2012); super-symmetric tensors to represent the constraints between the tuples (Cheng, Chen, Martin, Lai & Wang 2013) ; stochastic non-linear optimization to reduce the distance between scan pairs (Papazov & Burschka 2011); branch-and-bound using pairwise distance invariants (Gelfand, Mitra, Guibas & Pottmann 2005); or evolutionary game theoretic matching (Albarelli, Rodola & Torsello 2010, Rodolà, Albarelli, Bergamasco & Torsello 2013). However, these methods are all sensitive to missing data.

Following the concept of random sample consensus (RANSAC), another kind of method is 4-points Congruent Sets (4PCS) (Aiger, Mitra & Cohen-Or 2008), which uses a randomized alignment approach and the idea of planar congruent sets to compute optimal global rigid transformation. The 4PCS method is widely used and has been extended to take into account uniform scale variations (Corsini, Dellepiane, Ganovelli, Gherardi, Fusiello & Scopigno 2013). However, these methods have a complexity of $O(n^2 + k)$

where n denotes the size of the point clouds and k is the set of candidate congruent 4-points. It has great limitations when point numbers are large. To remove the quadratic complexity of the original 4PCS, (Mellado, Aiger & Mitra 2014) extends it to a fast algorithm with only linear computation time needed. This method reports the points or spheres in R^3 and uses a smart index to quickly find the matched plane in all candidate congruent 4-points planes. One cross-source point cloud registration experiment is reported in (Mellado et al. 2014). However, these methods have many limitations due to their point-level operation. They may easily be sub-optimal when computing their transformation relations. The varying density of the cross-source problem makes the performance of the 4PCS-based method even worse.

Although these direct methods show some ability in addressing elements of the cross-source problem, none of them can deal with the complete cross-source problem. In this thesis, a novel method is proposed to robustly deal with the entire cross-source problem. The method extracting and combining macro and micro structures is robust to large variations in density, noise and outliers. In addition, the enhanced graph matching globally registers two structures. Lastly, a scale normalization step is used to eliminate most of the scale variation.

2.1.2 Transformed methods

One of the mathematical tools typically used for registration is Mutual Information (MI), which catches the non-linear correlations between the point clouds and the geometric properties of the target surface. The authors in (Sinha, Cash, Weil, Galloway & Miga 2002) use ICP and mutual information (MI) to build one-to-one correspondence between an magnetic resonance (MR) surface and laser-scanned cortical surface; however, this method is highly dependent on initialization and overlap rate. The work in (Pandey, McBride, Savarese & Eustice 2012) registers unstructured 3D point clouds by using K-means to form a set of codewords and using an estimator to optimize the MI value to obtain the final rigid relations. Cross correlation of the

horizontal cross section images of the two point clouds is used in (Moussa & Elsheimy 2015) to coarsely register the point clouds, and ICP is then used to refine the coarse results. These MI-based methods perform poorly when data is missing because it make the MI of two point clouds originally not the same.

Another type of transformed method is the feature-based method, which extracts features from 3D point clouds and transforms the point cloud registration Euclidean space into feature space. Typical 3D feature extraction methods are Fast Point Feature Histograms (FPFH) (Rusu, Blodow & Beetz 2009a), Ensemble of Shape Functions (ESF) (Wohlkinger & Vincze 2011), Spin image (Johnson 1997) and Signature of Histograms of OrienTations (SHOT) (Tombari, Salti & Di Stefano 2010a). These feature-based methods produce exciting results on same-source point clouds. However, it is very difficult to reliably extract similar features from cross-source point clouds, and these methods always fail in this situation. This is because these features may originally process large discrepancy and cannot used for registration.

Torki and Elgammal (Torki & Elgammal 2010) use local features in images to learn manifold symbol. The authors first learn a feature embedding representation that contains the spatial structure of the features as well as the local appearance similarity. The out-of-sample method is then used to embed the features from new images. Similarly, Yuan (Deng, Rangarajan, Eisenschenk & Vemuri 2014) transforms every point in a point clouds into a shape representation, in order to cast the problem of point sets matching as a shape registration problem, which is the Schrodinger distance transform (SDT) representation. The problem is then transformed into solving a static Schrodinger equation in place of the consistent static Hamilton-Jacobi equation in the setting. The SDT representation is an analytic expression which can be normalized to have unit L2 norm in accordance with theoretical physics literature. The outline of this method is "points set"$->$ "SDTs"$->$"minimize the geodesic distance".

Related to point cloud registration, another kind of methods is Gaus-

sian mixture models (GMM)-based methods. To deal with the noise and outliers existing in the point sets registration problem, Bing et al. (Jian & Vemuri 2011*a*) proposed a method in which point clouds were represented as GMM and, went on to solve the registration problem by minimizing the statistical discrepancies between corresponding GMMs. This approach can be used for both rigid and non-rigid point cloud registration, and has demonstrated its ability to deal with noise and outliers to some extent. Georgios et al. (Myronenko & Song 2010) introduced a motion drift idea into the GMM framework and achieved good results on rigid and non-rigid point set registration. A solution to the GMM-based approach by recasting registration as a clustering problem was proposed in (Evangelidis & Horaud 2018). However, there are an increasing number of GMM models to robustly represent point clouds. When the point number increases to tens of thousands or millions, these methods are impractical in terms of both computational and memory cost. On the other hand, the GMMs depicting two point clouds are shown a lot of difference when there is missing data and large noise and outliers variations in cross-source point clouds, which makes the registration inaccurate or it may even fail. The experiments in Section 3.4 demonstrate these approaches do not lead to satisfactory results for cross-source point cloud registration.

The aforementioned transformed methods demonstrate ability in dealing with parts of noise and outliers or density variation, but none of them can successfully address the cross-source registration problem, which comprises issues of scale, density variation, noise and outliers and missing data. In this thesis, we aim to address this difficult cross-source problem. Motivated by our human registration process, a structure-based framework is proposed to robustly register two cross-source point clouds.

In summary, none of the existing works are designed to deal with cross-source point cloud registration. The existing algorithms can solve part of the challenges relating to the cross-source problem, such as noise, outliers, density, different viewpoints and missing data, but none of them are able

to solve all of these challenges. In this thesis, the structure property of the cross-source point cloud is explored and several algorithms are proposed to solve the cross-source point cloud registration problem.

To solve the cross-source point cloud registration problem, apart from traditional registration processes such as ICP, several new mathematical tools are introduced. At the same time, connections between these mathematical tools and the registration problem are proposed.

2.2 Graph matching

In this section, we review graph matching. As it has a long history and a large body of work has been dedicated to this issue, we only review the work which is related to our proposed methods. Firstly, factorized graph matching is reviewed. Secondly, high-order graph matching is detailed. Thirdly, the connection between graph matching and registration is discussed.

2.2.1 Graph matching problem

A graph is a mathematical tool to describe the structural information and the relations between this information. Nodes and edges are usually defined in a graph. Graph matching is a problem to find the similarity of graphs by comparing the similarity of nodes and edges. A generic formulation of the graph matching problem consists of finding the optimal matching matrix X given by the solution of the following (NP-hard) quadratic assignment problem,

$$arg\max_{X} J(X) = \Sigma_{i_1 i_2} x_{i_1 i_2} k^p_{i_1 i_2} + \Sigma_{\substack{i_1 \neq i_2, j_1 \neq j_2 \\ h^1_{i_1 c_1} g^1_{j_1 c_1}=1 \\ h^2_{i_2 c_2} g^2_{j_2 c_2}=1}} x_{i_1 i_2} x_{j_1 j_2} k^q_{c_1 c_2} \tag{2.1}$$

where $k^p_{i_1 i_2}$ is similarity of node i_1 and i_2, $k^q_{c_1 c_2}$ is the similarity of edge c_1 and c_2, edge c_i connects node i_i and node $j_i (i = 1, 2)$. h and g are two matrix

defining node relations of direct graph. For more detailed information, please referred to (Zhou & De la Torre 2016).

It is more convenient to write $J(X)$ in a quadratic form, $x^T K x$, where $x = vec(X) \in \{0,1\}^{n_1 n_2}$ is an indicator vector and $K \in R^{n_1 n_2 \times n_1 n_2}$ is computed as follows:

$$k^p_{i_1 i_2 j_1 j_2} = \begin{cases} k^p_{i_1 i_2} & \text{if } i_1 = j_1 \text{ and } i_2 = j_2 \\ k^q_{c_1 c_2} & \text{if } i_1 \neq j_1 \text{ and } i_2 \neq j_2 \text{ and} \\ & h^1_{i_1 c_1} g^1_{j_1 c_1} h^2_{i_2 c_2} g^2_{j_2 c_2} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

With K defined, the graph matching problem 2.1 can be converted into the maximization of the quadratic assignment problem (QAP) (Loiola, de Abreu, Boaventura-Netto, Hahn & Querido 2007),(Zhou & De la Torre 2016):

$$arg \max_X J(X) = vex(X)^T K vec(X) \tag{2.3}$$

According to (Zhou & De la Torre 2016), the above graph matching problem is a NP-hard problem. Different relaxation solutions are needed to solve the problem. In the following, factorized graph matching is reviewed to solve the graph matching problem.

2.2.2 Factorized graph matching

In order to solve the efficiency issue and constrain the graphs to geometric transformation, (Zhou & De la Torre 2016) provides a method to factorize the affinity matrix of the graph into many components so there is no need to compute the expensive affinity matrix. Therefore, matrix K in equation 2.3 can be factorized as:

$$K = diag(vec(K_p)) + (G_2 \otimes G_1) diag(vec(K_q)) (H_2 \otimes H_1)^T \tag{2.4}$$

where G_1, H_1, G_2, H_2 are edges and nodes similarity matrices from the similarity of K_p and K_q, $diag(a)$ is to create a diagonal matrix by using the element of a, $vec(a)$ is to format a as a vector.

Based on the factorization of K, the graph matching problem in 2.3 can be reformulated as:

$$J_gm(X) = tr(K_p^T X) + tr(K_q^T(G_1^T X G_2 \circ H_1^T X H_2)) \tag{2.5}$$

In the above equation function, suppose

$$Y = G_1^T X G_2 \circ H_1^T X H_2 \tag{2.6}$$

According to (Zhou & De la Torre 2016), Y can be interpreted as a correspondence matrix for edges, because $y_{ij} = 1$ means that edge i is correspondent to edge j.

A factorized graph matching (FGM) method (Zhou & De la Torre 2016) is used to develop an initial-free optimization scheme with no accuracy loss to address the non-convex issue. This method divides matrix K into many smaller matrices. Using these smaller matrices, the graph matching optimization problem can be transformed to iteratively optimize the following non-linear problem:

$$\max_X J_\alpha(X) = (1-\alpha)J_{vex}(X) + \alpha J_{cav}(X) \tag{2.7}$$

where J_{vex} and J_{cav} are two relaxations in FGM (Zhou & De la Torre 2016), which are defined as following convex relaxation and concave relaxation respectively:

$$\begin{aligned} J_{vex}(X) &= J_{gm}(X) - \frac{1}{2}J_{con}(X) \\ &= tr(K_p^T X) - \frac{1}{2}\sum_{i=1}^{c} \|X^T A_i^1 - A_i^2 X^T\|_F^2 \end{aligned} \tag{2.8}$$

$$\begin{aligned} J_{cav}(X) &= J_{gm}(X) + \frac{1}{2}J_{con}(X) \\ &= tr(K_p^T X) + \frac{1}{2}\sum_{i=1}^{c} \|X^T A_i^1 + A_i^2 X^T\|_F^2 \end{aligned} \tag{2.9}$$

The above convex relation and concave relaxation functions are all a combination of J_gm and J_{con}. According to (Zhou & De la Torre 2016), they

are all doubly-stochastic matrices. Therefore, both matrices have double derivatives and have their Hessian matrices. J_{con} is defined as

$$J_{con}(X) = \sum_{i=1}^{c} \{tr((A_i^1)^T X X^T A_i^1) + tr((A_i^2)^T X^T X A_i^{2T})\} \tag{2.10}$$

Maximizing equation 2.7 is a typical nonlinear programming problem when α is specific. According to (Zhou & De la Torre 2016), Frank-Wolfe's algorithm (FW) (Fukushima 1984) is utilized to solve the maximization of graph matching at specific α. Then, different α is selected and optimized from $0.1 \sim 1$.

2.2.3 High-order graph matching

After structure extraction, we obtain N_1 points (the central points of segments) in point cloud C_1 and N_2 points (the central points of segments) in point cloud C_2. The goal of the cross-source point cloud registration algorithm is to estimate the transformation matrix between two point clouds C_1 and C_2. To estimate the transformation matrix, instead of working on the original point clouds, we compute a few pairs of accurate correspondences on the extracted structure points. In this thesis, we firstly find a robust correspondence in a tensor space. Then, we estimate the new transformation matrix based on the correspondence. After the new transformation matrix is computed, the tensor space is updated using the new transformation matrix. These two processes are conducted iteratively until convergence.

Firstly, the correspondence estimation is introduced. Suppose P_i^s is the i_{th} point of point cloud C_s. The problem of estimating the correspondence between point cloud C_1 and C_2 is equivalent to finding an $N_1 \times N_2$ *assignment matrix* X such that $X_{ij} = 1$ when P_i^1 is correspondent to P_j^2,$X_{ij} = 1$ otherwise. In this thesis, we assume a point can find exactly one correspondent point from any point cloud to another point cloud (one-to-one correspondence). Mathematically, we assume both the sum of each row and each column are equal to one (e.g. sum the row and column of X_{ij} when we estimate

whether two points P_i^1 (row) is correspondent to P_j^2 (column)). Thus, the correspondence estimation problem of ICP considering pairwise constraints (Leordeanu & Hebert 2005, Zhou & De la Torre 2013a) can be formulated as the maximization of the following score over X:

$$\begin{aligned} S(X) &= \sum_{i_1 i_2, j_1 j_2}^{N_1, N_2} H_{i_1 i_2 j_1 j_2} X_{i_1 i_2} X_{j_1 j_2}, \\ \forall ij, \sum_{i_1, i_2}^{N_1} X_{i_1 i_2} &= 1, \sum_{j_1, j_2}^{N_2} X_{j_1 j_2} = 1 \end{aligned} \tag{2.11}$$

$H_{i_1 i_2 j_1 j_2}$ is a positive confidence describing point pair ($P_{i_1}^1$,$P_{i_2}^1$) in point cloud C_1 is correspondent to point ($P_{j_1}^2$,$P_{j_2}^2$) in point cloud C_2. The problem in equation 2.11 is similar to the graph matching problem, which is actually an integer quadratic programming problem. According to (Leordeanu & Hebert 2005, Cour, Srinivasan & Shi 2007), we can reformulate this problem as an integer quadratic program, which is maximizing

$$S(X) = \sum_{i,j}^{N_1, N_2} \bar{X}^T \bar{H} \bar{X}, X \in \{0, 1\} \tag{2.12}$$

where $\bar{X} \in \{0, 1\}^{N_1 N_2}$ is a binary vector by concatenating the columns of X, and, likewise, $\bar{H}$ is the $N_1 N_2 \times N_1 N_2$ symmetric matrix obtained by unfolding tensor H. This is a classical Rayleigh quotient problem, whose solution $\bar{X}$ is equal to the eigenvector associated with the largest eigenvalue of $\bar{H}$. According to (Leordeanu & Hebert 2005, Duchenne, Bach, Kweon & Ponce 2011), it can be solved very efficiently by a greedy algorithm.

Tensor multiplication: Because we use tensor multiplication in the following sections, we introduce this firstly. A tensor and a vector can be multiplied in different ways. In this thesis, we use the following notation:

$$\begin{aligned} B &= A \otimes_k V \\ B_{i \ldots i_{k-1}, i_{k+1}, \ldots i_n} &= \sum_{i_k} A_{i_1 \ldots i_k \ldots i_n} V_k \end{aligned} \tag{2.13}$$

where V is an n-dimensional vector and A is an n-dimensional tensor. Similar to the way a matrix multiplied by a vector produces a vector, an n-dimensional tensor multiplied by a vector is $(n-1)$-dimensional. Also, similar to the matrix-vector multiplication that can be done in two ways (on the left or on the right), tensor-vector multiplication can be done in n different ways. The index k in the notation $\otimes_k$ indicates that we multiply on the k_{th} dimension.

In the following section, we use the following calculus:

$$\begin{aligned} S(\bar{X}) &= \bar{H} \otimes_2 \bar{X} \otimes_1 \bar{X} \\ &= (\bar{H} \otimes_2 \bar{X}) \otimes_1 \bar{X} \\ &= (\sum_i H_{ij} X_i)_j \otimes_1 \bar{X} \\ &= \sum_{ij} H_{ij} X_i X_j \end{aligned} \tag{2.14}$$

Higher order tensor multiplication follows the same procedure as Equation 2.14.

2.2.4 Connection between graph matching and cross-source point cloud registration

Definition 2.1 *Cross-source point clouds are heterogeneous point clouds from different types of sensors.*

Definition 2.2 *Cross-source point cloud registration is a problem to estimate the transformation matrix between two or more scans of cross-source point cloud.*

Based on the above definitions, we summarize the following two points:

(1). the current registration methods contain two technique keys namely, finding the corresponding points and undertaking transformation estimation based on the corresponding points. The final goal of the registration methods is to estimate the transformation matrix of two point

clouds. The necessary step of estimating the transformation matrix is to find the corresponding points from the pair of point clouds. The differences between these methods is that they consider different constraints of corresponding similarity [1]. For example, (1) some methods consider two points as correspondent points when their descriptors are matched; (2) apart from (1), some methods further consider a pair of points as correspondent to a pair of points, (3) some methods even consider high-order constraints.

(2). to achieve the goal of registration, optimization strategies are needed to estimate the transformation matrix from the correspondent points. Usually, outlier removal strategies are integrated into the transformation matrix estimation because the estimated correspondence in the previous step containsx some outliers.

Therefore, the key point of registration is correspondence estimation. Graph matching methods provide solutions for correspondence estimation. For example, 3D points are regarded as nodes of a graph and the connections of neighbors around the 3D point are regarded as edges of the graph. Then, the correspondence estimation problem is converted into the graph matching problem. However, a scan of a point cloud usually has thousands or millions of 3D points. The complexity of graph matching becomes extremely challenging if we directly use the 3D points. So, the key tasks for graph matching-based point cloud registration are that (1) how to build up graphs, (2) how to efficiently solve graph matching optimization.

Regarding the graph build-up, the structure information is the key information of the 3D point cloud which can be used to undertake point cloud processing. Thus, how to build up a reasonable graph to keep the structure information and also to maintain as much detail as possible is a trade-off problem. In the following chapter, a macro- and micro-structure method is

[1]Here, constraint is how many points are considered at each time when we justify whether two points are correspondent or not.

proposed to keep the overall global structure and also to maintain the local detail as much as possible.

Regarding graph matching optimization, previous methods mostly aim to obtain an optimal solution by finding a better relaxation strategy. They try to relax the quadratic assignment problem (QAP) problem into a low-complexity convex or concave problem which can be solved very quickly. Also, reasonable results can be achieved. In the following chapter, a graph matching method which considers a rigid geometrical transformation constraint is proposed. The experiment achieves good results in both cross-source and same-source point cloud registration.

In summary, a graph matching method provides a solution for correspondence estimation. With the correspondence found, optimization methods are applied to estimate the transformation matrix between point clouds based on their correspondences.

2.3 Gaussian mixture models

According to (Anzai 2012), a single Gaussian distribution suffers from significant limitation in modelling real data sets. To better model a real data set, (Anzai 2012) shows that a linear combination of Gaussian components can result in very complex densities. Extending this conclusion, we can use a sufficient number of Gaussian components to approximate continuous density. Density estimation accuracy can be guaranteed by adjusting their coefficients of the combination as well as their means and covariances. Therefore, the density of K Gaussian can be formulated as

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{2.15}$$

where $N(x|\mu_k, \Sigma_k)$ represents a Gaussian mixture model which is also called a component of the Gaussian mixture model. μ_k and Σ_k are the mean and covariance of the component of the Gaussian mixture model. π_k is the mixing coefficients and the sum of all the mixing coefficients is equal to one:

$$\sum_{k=1}^{K} \pi_k = 1 \tag{2.16}$$

Because $N(\mu_k, \Sigma_k) \geq 0$ and $p(x) \geq 0$, this implies $\pi_k \geq 0$. Also, because the sum of all the mixing coefficients is 1, we obtain the range of π_k as follows:

$$0 \leq \pi_k \leq 1 \tag{2.17}$$

In order to better understand the Gaussian mixture model, according to Bayes theorem, the above representation of Gaussian mixture models can also be reformulated as

$$p(x) = \sum_{k=1}^{K} p(k)p(x|k) \tag{2.18}$$

where $p(k) = \pi_k$ is similar to the prior probability of picking k^{the} component, and the density $N(x|\mu_k, \Sigma_k) = p(x|k)$ is similar to the probability of x conditioned on k.

Another quantity that is important for the analysis of Gaussian mixture models is the conditional probability of z given x. This can also be interpreted as the probability that a k component contributes to all the Gaussian mixture models with all the parameters already known. Conditional probability can be written as

$$\begin{aligned} p(z_k|x) &= \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x|z_j = 1)} \\ &= \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j Z)} \end{aligned} \tag{2.19}$$

Therefore, we have defined the Gaussian mixture model. From the above definition, we find that the Gaussian mixture models are dominated by the parameters $\{\pi = \pi_1, ..., \pi_K, \mu = \mu_1, ...\mu_K, \Sigma = \Sigma_1, ...\Sigma_K\}$. Given many set of data, Gaussian mixture models provide a method to robustly represent its density. The estimation of a Gaussian mixture model is to estimate the parameters. According to (Anzai 2012), the parameters can be solved by the

maximum likelihood of the following likelihood function:

$$\ln p(X|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)\} \tag{2.20}$$

An expectation-maximization (EM) can be utilized to solve maximum likelihood. Firstly, we set the derivatives of $\ln p(X|\pi,\mu,\Sigma)$ with respect to μ_k to zero:

$$-\sum_{n=1}^{N} \gamma(z_{nk})\Sigma_k(x_n - u_k)) = 0 \tag{2.21}$$

where $\gamma(z_{nk})$ is interpreted as the contribution of k components to the Gaussian mixture models. It is similar to Eq. 2.19 that is defined as:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k)}{\sum_j \pi_j N(x_n|u_j,\Sigma_j)} \tag{2.22}$$

From equation 2.21, we can solve the u_k:

$$u_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})x_n \tag{2.23}$$

where N_k is a constant value,

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{2.24}$$

In the same way, we set the derivative of $lnp(X|\pi,\mu,\Sigma)$ with respect to Σ to zero and we obtain the solution of Σ_k:

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(x_n - u_k)(x_n - u_k)^T \tag{2.25}$$

Finally, the mixing coefficients π_k are estimated by considering: (1) the derivatives of $lnp(X|\pi,\mu,\Sigma)$ with respect to π_k, (2) the constraint of 2.16, that is the sum of all mixing coefficients is equal to one. Then, the solution of π_k can be estimated using a Lagrange multiplier:

$$\max\{\ln p(X|\pi,\mu,\Sigma) + \lambda(\sum_{k=1}^{K} \pi_k - 1)\} \tag{2.26}$$

By setting the derivative of the above equation with respect to π_k to zero, and considering equation 2.20, we obtain:

$$\sum_{n=1}^{N} \frac{N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} + \lambda = 0 \tag{2.27}$$

We multiply both sides by π_k and sum over k by making use of the constraint of sum to one in equation 2.16, hence we obtain

$$\begin{aligned} 0 &= \sum_{n=1}^{N} \frac{N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} + \lambda \\ &= \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} + \pi_k \lambda \\ &= \sum_{n=1}^{N} \frac{\sum_k \pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} + \sum_{k=1}^{N} \pi_k \lambda \\ &= \sum_{n=1}^{N} 1 + \lambda \end{aligned} \tag{2.28}$$

So, we obtain $\lambda = -N$. According to equation 2.19, we find that each component of $z_k = 1$ contributes a component of $\frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j Z)}$. After summing all the components which belong to $z_k = 1$ and defining the number as N_k, we obtain:

$$N_k = \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{2.29}$$

By combining equations 2.29 and $\lambda = -N$, the solution of π_k is:

$$\pi_k = \frac{N_k}{N} \tag{2.30}$$

By combining equation 2.23, 2.25, 2.30 and 2.22, we estimate the current results of these parameters. However, because the current results of μ_k, Σ_k, π_k are all related to $\gamma(z_{nk})$ and $\gamma(z_{nk})$ relies on these parameters, this is a complex optimization way. We use a EM optimization process to optimize

the process. In the E step, we estimate $\gamma(z_{nk})$; in the M step, we estimate μ_k, Σ_k, π_k. These two steps are conducted iteratively until convergence.

In summary, EM procedures can be done iteratively by using the following equations:

1. E step. Estimate the probability of each component:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} \tag{2.31}$$

2. M step.

$$u_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{2.32}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - u_k)(x_n - u_k)^T \tag{2.33}$$

$$\pi_k = \frac{N_k}{N} \tag{2.34}$$

and

$$N_k = \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(x_n|u_j, \Sigma_j)} = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{2.35}$$

In matching or registration problem, suppose two scans of point clouds have an overlapping region which describe the same scene, the goal is to align the point clouds in one coordinate system by aligning the Gaussian mixture models. There are two types of GMMs which can deal with the matching and registration problem:

- **Similarity-based Gaussian Mixture Models.** The idea is that Gaussian mixture models is utilized as a natural and simple way to represent the given point sets. Then, the point cloud registration problem can be regarded as the alignment of two Gaussian mixtures by minimizing the discrepancy between two Gaussian mixture models. A typical example is (Jian & Vemuri 2011*b*).

- **Clustering-based Gaussian Mixture Models.** The idea is that suppose there is a Gaussian mixture model to describe the 3D scene, point clouds are samples generating from the Gaussian mixture models. To conduct the registration, it is similar to estimate the Gaussian mixture models by clustering the two samplses into one Gaussian mixture models. Typical example is (Evangelidis & Horaud 2018).

2.3.1 Similarity-based Gaussian Mixture Models

Suppose point cloud 1 can be represented as a Gaussian Mixture Model, we name it g; and point cloud 2 can be represented as a Gaussian mixture model, we name it f. So, the spatial transformation matrix T between two Gaussian mixture models g and f can be optimized by the following functions:

$$S = \int \{g - f(T)\}^2 dx = \int \{g^2 - 2gf(T) + f^2(T)\} dx \tag{2.36}$$

According to (Jian & Vemuri 2011*b*), the close-form solution of the above function can be easily derived by the following formula:

$$\int \phi(x|u_1, \Sigma_1)\phi(x|u_2, \Sigma_2)dx = \phi(0|u_1 - u_2, \Sigma_1 + \Sigma_2) \tag{2.37}$$

When we solve the problem, we obtain both the parameters of the Gaussian mixture models and spatial transformation matrix. For implementation details, please refer to (Jian & Vemuri 2011*b*).

2.3.2 Clustering-based Gaussian Mixture Models

According to (Evangelidis & Horaud 2018), point set registration can be done by assuming a virtual Gaussian mixture model and point clouds are samples of Gaussian mixture models. Figure 2.1 visually illustrates the theory of clustering-based Gaussian mixture models. Suppose V is many scans of observed point clouds, which are independent and identically distributed. We assume they can be represented by Gaussian mixture models. Therefore,

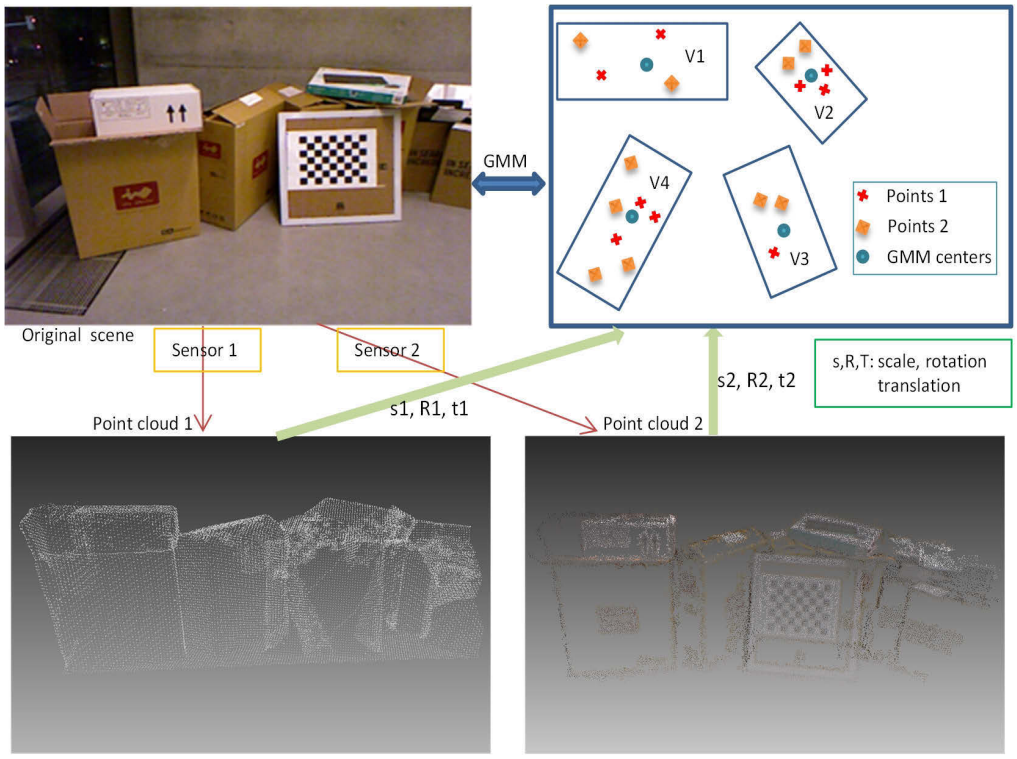

Figure 2.1: Theory for clustering-based Gaussian mixture models.

the registration problem is to project the observed data V into the Gaussian mixture models:

$$P(V_{ji}) = \sum_{k=1}^{K} p_k N(\phi(v_{ji})|x_k, \Sigma_k) + p_{K+1} U(a-b) \tag{2.38}$$

$\phi_j(v_{ji}) = R_j v_{ji} + t_j$. R and T are the 3×3 rotation matrix and 3×1 translation vector, respectively. This formulation is the key component of this category. We project many observed point clouds into a virtual space where many observed point clouds in this space can be represented as a Gaussian mixture model. p_k are the mixing coefficients and they have the constraint $\sum_{k=1}^{K+1} p_k = 1$. x_k and Σ_k are the means and covariance of each component of the Gaussian mixture models. U is the uniform distribution where the parameters belong to $a - b$. We use uniform distribution and p_{K+1} to deal with outliers. Here, $P_{K+1} = \gamma \sum_{k=1}^{K} p_k$. According to (Evangelidis & Horaud 2018), equation 2.38 can be solved by the expectation and maximization (EM) steps.

In summary, GMM-based methods utilize the statistic model to describe the point sets and integrate the spatial transformation matrix into the parameters estimation process. They usually use EM algorithms. Due to their use of statistical information, these kinds of methods have the following advantages:

- **Noise and outliers.** Because of the use of statistical information, noise and outliers can be largely overcome. Even though there is some extent of noise and outliers, the statistical information is similar. Therefore, noise and outliers can be largely tackled.

- **Density.** Because Gaussian mixture models use statistical information to describe a scene, the parameters of Gaussian mixture models are similar if the number of Gaussian mixture models is similar, although there is a different density on the data captured by different sensors.

- **Partial overlap.** For partial overlap, most components of the GMM

are similar if the point clouds have a large overlap. Therefore, the parameters of Gaussian mixture models in the overlapping regions should be similar because the statistical information is similar.

2.4 Deep Convolution Neural Network

Deep learning shows a strong ability in representation. Recently, deep convolution neural networks achieved high performance in 2D tasks, such as detection, recognition and semantic segmentation. A large number of neural networks has been proposed, such as AlextNet, FastRCNN, FCN, VGG, GoogleNet, MaskR-CNN and Pix2Pix. These algorithms focus on 2D images. A question is that how well does deep learning handle 3D data, such as 3D point clouds. There are many existing works which focus on this. Because point clouds have irregular data, how to integrate irregular data with a regular CNN is an open research problem. There are four categories of methods to use deep learning to handle 3D data: multi-view, mesh, 3D voxels and point clouds. There are also some algorithms which deal with 3D representation (e.g. PointNet, PointNet++ and 3DMatch), 3D detection (e.g. VOTE3D, VoxelNet, Vote3Deep, MV3D (Chen, Ma, Wan, Li & Xia 2017)), and 3D segmentation (ScanNet, foldingNet(Yang, Feng, Shen & Tian 2018), SEG-Cloud, Squeezeseg). Also, there is a method to use deep learning to deal with registration for localization by using auto-encoder (Elbaz, Avraham & Fischer 2017). This thesis introduces deep learning for 3D point cloud and the details of the convolution operation, the convolution network and deep learning for 3D matching.

2.4.1 Convolution Operation

A traditional convolution neural network is targeted in 2D images. We begin with 2D convolution as an example. The convolution operation is defined as follows:

$$y = w_{ij}x_{ij} + b_{ij} \tag{2.39}$$

where w_ij and b_{ij} are the parameters of the constitutional kernel operation. Take 3×3 as an example, $i, j \in \{0, 1, 2\}$. x_{ij} is the image pixel corresponding to the weight parameters. This is very easy to understand if we slide a square box into an image. We multiply the corresponding position between the square box and the image region, and add the values of the square boxes. The mathematic is shown in the above equation 2.39.

3DCNN based on 3DVoxel: For 3D convolution neural networks (3DCNNs), the convolution operation is similar. Firstly the 3D voxel for 3DCNN is introduced, then other solutions for using 3D data are discussed. Different to 2D images, the square box changes into a cube for 3D data. The operation is the same as that in equation 2.39 and the multiplication is conducted on the correspondent position and the addition is conducted on the cube region. For example, if the kernel of 3DCNN is $3 \times 3 \times 3$, the kernel slides among the 3D voxel. For each step, multiplication is conducted on the correspondent position between the kernel and the 3D voxel data, and then the addition is conducted on the whole kernel region. Therefore, the mathematics of the 3D convolution neural network is

$$y = w_{ijl}x_{ijl} + b_{ijl} \tag{2.40}$$

where $i, j, l \in \{0, 1, 2\}$ if the kernel is $3 \times 3 \times 3$. See Figure 2.2 as an example for the 3DCNN slide one step on the X axis.

3DCNN based on Multi-view images: Except for the aforementioned 3DCNN, another category of using deep learning in 3D data is to project the 3D into multi views in 2D. An example is (Su, Maji, Kalogerakis & Learned-Miller 2015), which projects 3D point clouds into 12 views around the point cloud. The performance is good however it faces many challenges. One of the key challenges is how to select the 12 projection views. Because the objects are rotated arbitrarily in 3D space, selecting 12 views is not easy.

3DCNN based on 3D points: Another category of using deep learning in 3D data is to directly utilize 3D point clouds as input, and perform the CNN operation on original 3D points. This category of using deep learning

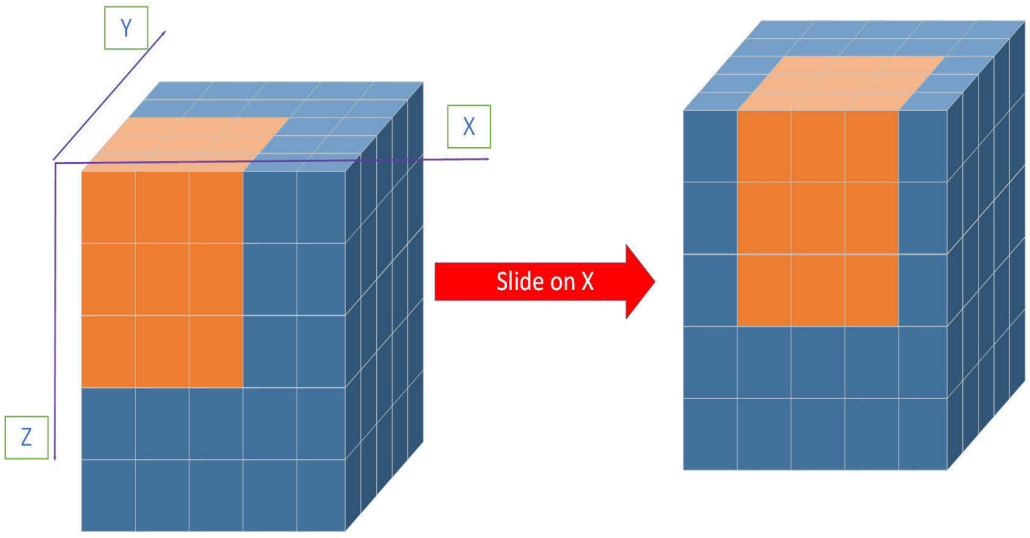

Figure 2.2: An example for 3DCNN. 3D kernel $3 \times 3 \times 3$ slide one step on X axis.

in 3D conducts 1×1 2D operation on 3D points, and the coordinates on the X, Y, Z axes are the three channels of 2D CNN. Many 1×1 2D convolution kernels are utilized to generate many features of the 3D points. Then, max-pooling is utilized to generate a global feature of the point cloud.

2.4.2 Convolution Networks

The convolution neural network utilizes many convolutions, pooling, and fully-connected layers to extract deep feature information to generate descriptors. In more detail, the descriptor of a 3D point cloud is a combination of 3DCNN, 3D pooling and 3D fully-connected layers. With different combinations, different kinds of information relating to point clouds are extracted. For example, 3DMatch (Zeng, Song, Nießner, Fisher, Xiao & Funkhouser 2017) designs a deep convolution network with 8 3D convolution networks, 1 fully-connected layer and 1 fully-connected layer to extract local descriptors for 3D point clouds. Because the network aims to solve matching problem, the network is a parallel structure. Two sub-networks extract deep features and connect together as a network to justify whether two patches

are matched or non-matched. Pointnet (Qi, Su, Mo & Guibas 2017) designs a deep network to extract deep 3D descriptors and utilizes the descriptors to solve 3D recognition and 3D semantic segmentation.

2.4.3 Deep convolution neural network for matching & registration

The deep convolution neural network for matching begins with 2D images in stereo matching. (Zbontar & LeCun 2016) designs a neural network to compare image patches of matching and non-matching pairs. The network uses parallel networks to extract two descriptors of the two patches and computes the similarity based on the generated descriptors. It is a parallel network which is different to a single network such as those used in solving the tasks of recognition and segmentation. (Zeng et al. 2017) proposes a parallel network for 3D point cloud matching. The network utilizes contrastive loss to train the network and to train on RGB-D reconstruction point clouds.

Therefore, a deep convolution neural network for matching or registration is usually a parallel network structure. Two descriptors are generated and their similarity is computed. In the training process, the same number of matching and non-matching pairs is input into the network. In the testing process, two patches are input into the network and the network computes the similarity. Based on the similarity, we can decide whether the patches are matching or non-matching.

2.5 3D sensors and their applications

With the development of software and hardware, many new 3D sensors and new applications have emerged. In this section, the 3D sensors and their related applications are reviewed.

2.5.1 Advances in 3D sensors

Today, there are a lot of 3D sensors which produce 3D point clouds or 3D data. Because of the development of stereo matching algorithms, depth sensors have undergone much development. There are many kinds of 3D sensors currently available. The main limitation is that depth sensors usually have a limited range, as the depth ranges from $0.5m - 6m$. Recently, some sensors have tried to deal with this limitation. For example, ZED stereo camera utilizes wide baseline stereo to increase the depth range.

In addition to passive depth sensors, there are a lot of active 3D sensors, such as different kinds of LiDAR. LiDAR utilizes an active laser to measure the distance between the LiDAR machine and the object surface. The 3D point coordinates can easily be computed by using the distance information. Previously, Velody used a mechanical machine to quickly rotate 360° and generate 3D point clouds around the LiDAR device. The advantage is that the accuracy of the 3D points is very high; the disadvantages are that (1) the device is heavy; and (2) the price is expensive $\$1000,000$ for a 64-line LiDAR. Recently, a solid-state LiDAR has been developed. The solid-state LiDAR solves the weight and price issue of the mechanical LiDAR.

With the development of many kinds of 3D sensors, because their sensing mechanisms are totally different, their acquisition of point clouds is a heterogeneous problem. So, cross-source point cloud registration is a research problem emerging based on this background. In this thesis, we introduce our research outcomes on cross-source point cloud registration.

2.5.2 Applications of cross-source point cloud registration

3D reconstruction: Registration is a key step in 3D reconstruction. Therefore, cross-source point cloud registration can be integrated into cross-source 3D reconstruction when there are different kinds of 3D sensors. For example, 3D points from stereo sensors usually capture good 3D points on a textured

region however some materials with texture may absorb the laser signal. Therefore, a combination of several 3D sensors usually obtains better 3D reconstruction results. Here, cross-source point cloud registration is a key component.

3D localization: Registration can be utilized to solve 3D localization. If two point clouds are correctly registered, the localization information is obtained. For example, given a small scene and a large scene of point clouds, firstly, several potential regions in the large scene are proposed; secondly, the registration algorithm is utilized to estimate which region is the correctly matched region. Therefore, the 3D localization problem is solved.

Chapter 3

Graph matching for cross-source point cloud registration

3.1 Introduction

Graph is a mathematical tool to describe the structure data. Graph matching is a method to align graphs. The behind theory of graph matching methods is that the graphs have similar structures. In the 3D point cloud field, the structure information is the most valuable. Especially for 3D cross-source point clouds, the structure of these point clouds are more important than the original points because the original points contain a lot of noise and outliers as discussed in the above Chapter 1. If we can correctly construct and describe the structure of the 3D cross-source point clouds, we may overcome some of the challenges in cross-source point cloud registration by utilizing the graph matching techniques. In this chapter, we will explore the value of structure information and propose a framework in solving the 3D cross-source point cloud registration problem.

The fact that point clouds come from different kinds of sensors (e.g. SFM with mobile phones, Kinect, range cameras and Lidar) present many chal-

lenges and the existing related methods have many limitations. There is a paucity of research in the literature on this issue. Based on our knowledge, there is a single paper about cross-source point cloud matching/registration (Peng, Wu, Fan, Zhang, You, Lu & Yang 2014*b*), but its registration is executed using conventional iterative closest point (ICP) (Best & McKay 1992) and many assumptions are made, including removing sparse outliers and manually selecting the dense point regions. A 4-Points Congruent Sets-based method (4PCS) shows elements of experiments that deal with cross-source problems (Mellado et al. 2014), although such 4PCS-based methods are sub-optimal in the direction of slippage as a result of operating at a point level (Gelfand & Guibas 2004, Aiger et al. 2008).

Despite the large variations in cross-source point clouds (discussed in Chapter 1), our human vision system seems able to align them effortlessly with high accuracy. This is probably due to the fact that humans exploit the similarities between the *structures of two cross-source point clouds* instead of the detailed points. Motivated by this insight, a method is proposed to extract and describe the macro structure (e.g. the global outline of objects) and the micro structure (e.g. voxels and segments) of point clouds. These macro and micro structures act like a net to robustly describe the invariant components of cross-source point clouds, and graph theory is a strong tool for preserving these structures from a mathematical viewpoint. A structure preserved representation method that ignores local point cloud details is proposed to deal with missing data and varying density. A scale normalization method is proposed to deal with the scale problem, and a systematic approach using these two methods is proposed to deal with all cross-source point cloud registration problems.

To the best of our knowledge, this is the first time a method has been proposed that successfully registers two cross-source point clouds in adverse scenarios. The proposed approach preserves the structure properties well by firstly, extracting reliable macro and micro structures to be robust to large noise, outliers and some of the missing data; secondly, integrating the

point cloud structures as graphs and describing them; thirdly, finding the optimal graph-matching solution, and lastly, refining the solution with 3D RANSAC (Random Sample Consensus) to remove outliers and ICP to finalize the outlier-free registration.

The contributions of this work are (1) a feasible structure-based framework to deal with the cross-source point cloud registration problem; (2) a new graph construction method to practically integrate macro and micro structures as a graph and robustly describe these structures; and (3) a new iteration method to solve the graph matching problem taking the global geometrical constraint into consideration.

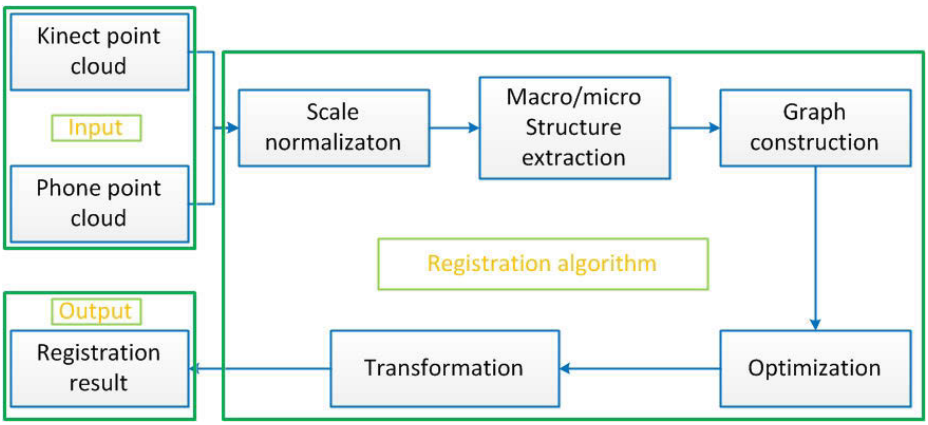

Figure 3.1: Overall system workflow.

The remaining sections of this Chapter are organized as follows: Section 3.2 describes the proposed macro and micro structure representation; Section 3.3 describes the proposed registration method based on our novel concept; and Section 3.4 describes the experiments, and Section 3.5 concludes the Chapter.

3.2 Macro and micro structure representation

As mentioned in Section 3.1, the significant challenges for 3D cross-source point cloud registration are the large variations in density, missing data, scale and angle between two point clouds. To address these variations, we define two structures, known as macro and micro structures, to describe the point clouds based on our observations. In our work, we extract structures from the cross-source point clouds and use these structures to indirectly register cross-source point clouds, instead of try to deal with these difficult changing points directly. Similar to our human ability, these structures robustly describe the global and local invariance of the cross-source point clouds, even though there are many variations in relation to these point clouds.

The macro structure is the overall outline or large-scale structure of an object or scene. It is important to note that it represents the global properties of the structure, such as the boundary outline, the contour and the shape, but not the global light, global color or global material. Figure 3.2(a) illustrates that the rectangle above the square (the blue outline) is the macro structure. When humans judge whether two objects are similar, they usually first consider the macro structure, and an overall alignment is obtained on this basis. We define a micro structure to work alongside the macro structure. The micro structure is defined as a small scale structure, such as a stable cell or part of the object or scene. It is a local property that describes the internal details of the object or scene. In our work, the micro structure consists of a 3D region, such as a super voxel in 3D point clouds. Figure 3.2(b) illustrates that, super voxels contain points with the same properties of 3D spatial geometry. We use these micro and macro structures to iteratively obtain the corresponding relations between two point clouds.

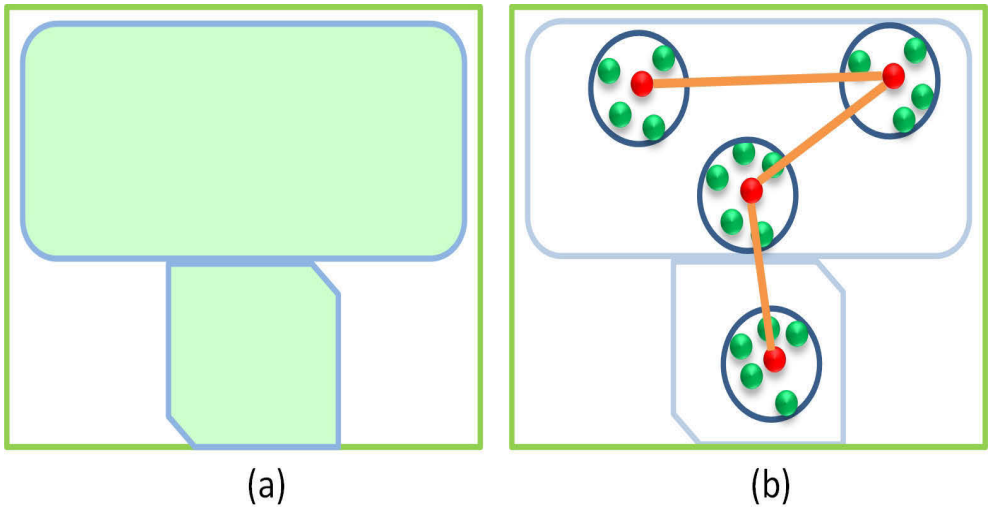

Figure 3.2: Schematic diagram of macro and micro structures.

3.3 Registration algorithm

In this section, we describe the registration method based on the proposed macro and micro structure theory and describe the components that make up our system. Figure 3.1 provides an overview of our method in block form. It comprises the following five components:

Step 1, Scale normalization: The pre-processing stage. Two cross-source point clouds, which come from different sensors, are normalized to the same scale. The details are given in Section 3.3.1.

Step 2, Macro/micro Structure Extraction: The main novelty of this stage is the reliable extraction of the structure from large variable cross-source point clouds, which is robust to most cross-source problems. These point clouds are segmented into many super voxels, using their 3D geometric properties, and the statistical property of each super voxel is used for its robust to local variations. These super voxels are integrated as the macro structure and the statistical property of each super voxel becomes the micro structure, as detailed in Section 3.3.2. These structures are integrated in the next step.

Step 3, Graph construction: The main novelty of this stage is the combination of micro and macro structures using graphs. Although there are many variations in two cross-source point clouds, the invariant structure properties are preserved in this method. The nodes are the extracted voxels and the edges are the adjacent relations. In addition, a new similarity measure method is proposed which robustly describes these two graphs, as detailed in Section 3.3.3. After the graph has been constructed, the registration problem is converted to a graph matching problem. An optimization method is thus needed to optimize the graph matching problem.

Step 4, Optimization: The novelty of this stage is the proposal of an enhanced optimization method. Factorized graph matching (Zhou & De la Torre 2016) is an optimization algorithm that optimizes graph matching at a constant time and is less prone to local optimization. To better suit to our problem and to pursue global optimal, we consider the geometry constraints in our optimization as detailed in Section 3.3.4. With this matching result, further refinement is needed.

Step 5, Transformation estimation: Transformation matrix computation stage. RANSAC is performed to first remove outliers, following which ICP refines the initial matching from the graph matching, as detailed in Section 3.3.5.

3.3.1 Scale normalization

The two point clouds come from different sensors and therefore have different scales. To remove scale variation, we conduct scale normalization before the super voxel extraction step. Previously, the scale was normalized by manual measurements in the real world and these two point clouds were calibrated, but although manual measurement is accurate, it is sometimes difficult. We propose an automatic method to estimate the scale without the need for manual work. To achieve this goal, we first compute the mean distance of two 3D points and then compute the scale by comparing these two means as follows:

$$scale = \frac{max||P_i - \overline{P}||_2}{max||Q_i - \overline{Q}||_2} \tag{3.1}$$

where $\overline{P} = (\sum_{i=1}^{N} P_i)/N$ and $\overline{Q} = (\sum_{j=1}^{M} Q_j)/M$

We use this scale to transform other point clouds and remove the scale difference in cross-source point clouds as far as possible. Although we cannot deal with the scale problem completely, the results of this stage are sufficient for the graph matching stage since most of the scale difference is eliminated. After the scale difference has been removed, the voxels can be extracted.

3.3.2 Macro/micro Structure extraction

Due to the large variations in cross-source point clouds, a method is needed to extract the invariable components. Figure **??** shows that even though the two cross-source point clouds have many variations, the structure can still be recognized. For these cross-source point clouds, therefore, the focus is on the structure information rather than the detailed information, since the latter is full of noise, outliers and different densities.

We are motivated by the idea of cluster, where points with the same property are clustered together. As shown in Figure 1, humans have the ability to register these monitors at first glance. This is because the macro structure information remains in the cross-source data and when humans conduct the registration work, they are not concerned with information detail (e.g. the location of a point). However, if we want to accurately register these two point clouds, macro structure information alone is insufficient, and micro structure information is also needed. Hence, to develop an intelligent registration algorithm, we need a method that will retain the common macro and micro structure information and ensure it is robust to varying densities and missing data.

To fulfill this goal, we improve the recently developed segmentation method (Papon, Abramov, Schoeler & Wörgötter 2013) to segment the two point

clouds into many super voxels and extract the direct adjacency graph of these voxels. As the segmentation method adheres to object boundaries while remaining efficient by only using the 3D geometric property, it obtains robust results for two point clouds, regardless of different density, angle, noise and missing data (see the third column of Figure 3.3). Figure 3.3 shows that the center of the segmented super voxels deals with much of the noise, density and missing data problem. Unlike (Papon et al. 2013), we do not flow back at the extraction of each edge in the adjacency graph extraction step, which means that the direction information is considered in our new adjacency graph. This is because in the following optimization step (Section 3.3.4), direct graph matching achieves more robust results than indirect graph matching (Zhou & De la Torre 2013*b*). This revision is a key element to ensure that these extracted voxels are correctly and robustly registered. At the same time, the ESF descriptor (Wohlkinger & Vincze 2011) for each voxel is extracted to describe the statistical property as a local structure. Based on the definition of macro and micro structures, therefore, each segmented super voxel is a micro structure and the whole of the adjacency graph and voxel centers are macro structures. After these structures have been extracted, they are integrated as a graph in the graph construction stage.

3.3.3 Graph construction

A new graph construction method is proposed to utilize macro and micro structures to deal with the cross-source point cloud registration problem. The new graph construction method integrates these structures and forms the registration problem into a graph matching problem. We select graph because it is a strong tool for maintaining the properties(e.g. topology) of macro structures. At the same time, the nodes and the edges of the graph are able to maintain properties of the micro structure.

Before introducing the new method, the graph matching notations are introduced. A graph with n nodes and m directed edges is defined as $\breve{\mathrm{C}}= \{P, Q, G, H\}$. P and Q are the features for the nodes and edges of the graph,

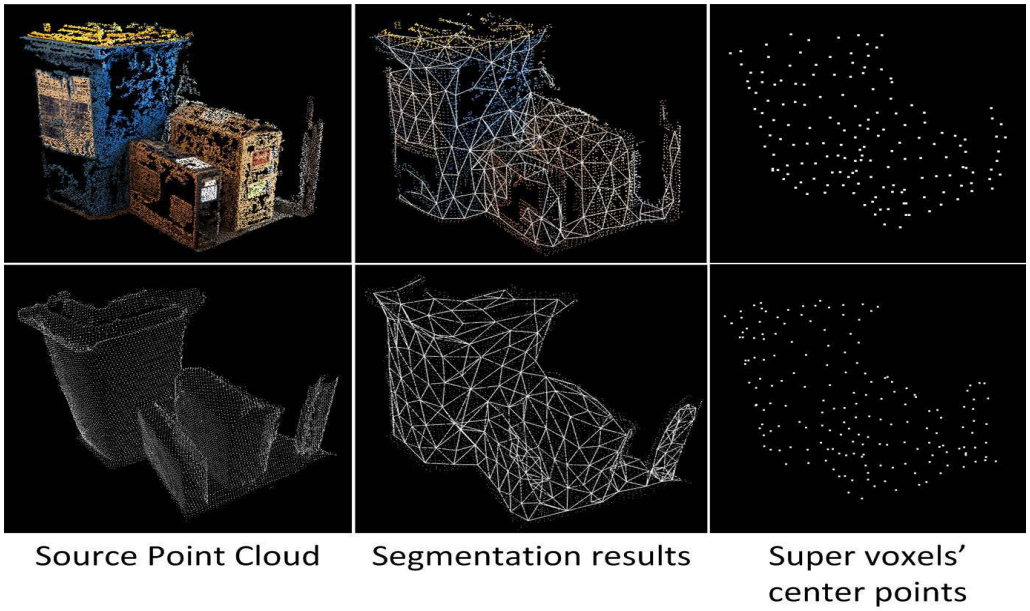

Figure 3.3: Results of macro/micro structure extraction. The first column is the source point clouds of SFM (above) and KinectFusion (bottom); the second column is the segmentation results and the connection relationship; the third column is the segmented super voxels' central points.

which are defined as $P = [p_1, \cdots, p_n] \in R^{d_p \times n}$and $Q = [q_1, \cdots, q_m] \in R^{d_q \times m}$ respectively. For example, p_i could be a SIFT descriptor or ESF descriptor extracted from the original data around the $i_t h$ node and q_i could be the length of the i^{th} edge. $G, H \in \{0,1\}^{n \times m}$ is a node-edge incidence matrix which describes the topology of the graph. We define $g_{ic} = h_{jc} = 1$ if the c^{th} edge connects the i^{th} node and the j^{th} node, and zero otherwise. To perform graph matching, given a pair of graphs, we first need to define P and Q. Next, we compute two affinity matrices, $K_p \in R^{n_1 \times n_2}$ and $K_q \in R^{m_1 \times m_2}$ to measure the similarity of each node and edge pair, then $k^p_{i_1 i_2} = \phi_p(p^1_{i_1}, p^2_{i_2})$ measures the similarity between the i^{th}_1 node of $\breve{\mathrm{C}}_1$ and the i^{th}_2 node of $\breve{\mathrm{C}}_2$, and $k^q_{c_1 c_2} = \phi_q(q^1_{c_1}, q^2_{c_2})$ measures the similarity between the c^{th}_1 edge of $\breve{\mathrm{C}}_1$ and the c^{th}_2 edge of $\breve{\mathrm{C}}_2$. Only when we define these matrices correctly, can we use graph matching method.

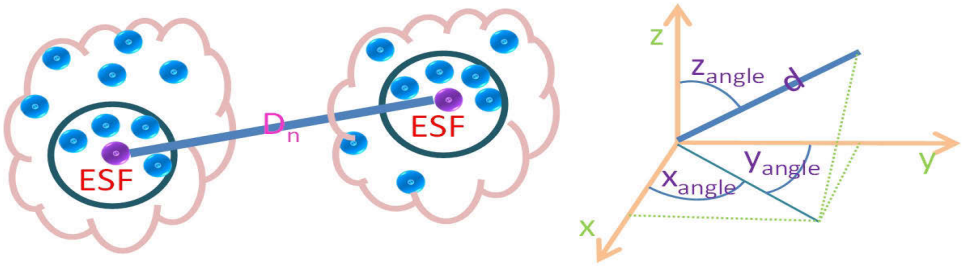

Figure 3.4: Schematic diagram of graph nodes and edges.

A robust structure-based graph construction method is proposed in this chapter. To robustly deal with the many variations in cross-source problem, with exception of structure extraction, a structure-retaining similarity measurement method is needed. In other words, the graph should be robustly described despite the cross-source problem. As previously discussed, humans can still register cross-source point clouds correctly by their structure. Similar to the human register's process, the graph is constructed as a expression of the relations between structures. This is another key element obtaining robust registration results. The micro structures are utilized as the node descriptors and the spatial relations of the centers of micro structures are utilized as the edge descriptors. The graph has the ability of being robust to large variations in density, angle and missing data of cross-source point clouds. Here, we describe how to design the nodes and edges of these graphs, and their similarity measurement.

Graph node and similarity measurement

To robustly represent the micro structures of point clouds, the method should be resilient to the large variations in density and missing data. We segment the super voxels of two point clouds and extract the centroid point of each super voxel. The graph node E is constituted by these centroid points. To correctly match these nodes, they need to be described discriminately. Due to the cross-source problems discussed above (i.e. varying density, missing

data and large variations in scale and rotation angles), using only the coordinates of these centroid points cannot describe discriminately for node description and the original matched nodes pairs are very rare. To robust deal with the cross-source problem, we select the ESF descriptor (Wohlkinger & Vincze 2011) instead of using conventional nodes' coordinate because the ESF descriptor is a global descriptor that adds up the properties of the distance, angles and area of the point clouds. Using the ESF descriptor, we transform the variable Euclidean space into feature space (ESF 640). If two points come from the corresponding segments, the ESF descriptors will mostly be the same and should be matched, even though the centroid point may not perfectly match in the Euclidean space.

The node similarity matrix K_p is computed by comparing the distance between the nodes' ESF descriptors(see left hand of Figure 3.4). Here, the node similarity is not computed in Euclidean space but in feature space. Because ESF is a statistic and global descriptor, it has the ability to avoid the large local variations in Euclidean space and hence is more robust to the cross-source problem. The node similarity is

$$K_p = \overline{D}_{esf} \tag{3.2}$$

where $\overline{D}_{esf}$ is the normalized distance of two 3D points' ESF descriptors, $\overline{D}_{esf} = D_{esf}/max(max(D_{esf}))$. D_{esf} is the distance of two 3D points' ESF descriptors and $D_{esf} = \|P^1_{esf}(i) - P^2_{esf}(j)\|_2$.

Graph edge and similarity measurement

To robustly and discriminately describe the point cloud, it is necessary to build the edges accurately to reflect its macro structure. We record the adjacent relations (extracted in Section 3.3.2) between super voxels and use these adjacent relations as edges Q. The adjacent relations correctly reflect the relations of the super voxels through the boundary property. The edges need to be described discriminatingly and meaningfully to ensure they are

correctly matched. We need to reiterate that humans can register these two cross-source point clouds because their structures are almost the same. We therefore need to retain the structure property of these two graphs in describing edges. Edge direction is also an important factor for the structure of the graph, in spite of the edge distance.

In this chapter, we use the spatial distance and geometric properties of these edges (see right hand of Figure 3.4). The Euclidean distance and Eular angles of two connected nodes are combined to construct a descriptor vector for describing the edges Q: $(x_{angle}, y_{angle}, z_{angle}, d)$, where $d = \|P_i - P_j\|_2, z_{angle} = acos(z/d), x_{angle} = acos(x/(d * sin(z_{angle})), y_{angle} = acos(y/(d * sin(z_{angle}))$. We compare the similarity by comparing the similarity of these descriptors and obtain D_e, where $D_e = \|Q_i^1 - Q_j^2\|_2$. To make a more robust comparison, we normalize the descriptor $\overline{D_e} = D_e/max(D_e)$, and the edge similarity matrix K_q is computed by

$$K_q = \overline{D}_e \tag{3.3}$$

This is a simple means of obtaining features in 3D point clouds (Euclidean distance and Eular angles of two points). At the same, it describes the edges, taking the spatial relations and structures into consideration. Tts ability to register the cross-source point clouds will be demonstrated in the experiment section.

3.3.4 Optimization

We propose an enhanced factorized graph matching method which considers global geometry constraint to deal with the local minima problem in graph matching. Before introducing our method, we briefly review graph matching and FGM (Zhou & De la Torre 2016). Suppose there is a pair of graphs, $\breve{\mathrm{C}}_1 = \{P_1, Q_1, G_1\}$ and $\breve{\mathrm{C}}_2 = \{P_2, Q_2, G_2\}$. The problem of graph matching consists of finding a correspondence between the nodes of $\breve{\mathrm{C}}_1$ and $\breve{\mathrm{C}}_2$ that maximizes the following score of global consistency:

$$J(X) = \Sigma_{i_1 i_2} x_{i_1 i_2} k^p_{i_1 i_2} + \Sigma_{\substack{i_1 \neq i_2, j_1 \neq j_2 \\ h^1_{i_1 c_1} g^1_{j_1 c_1} = 1 \\ h^2_{i_2 c_2} g^2_{j_2 c_2} = 1}} x_{i_1 i_2} x_{j_1 j_2} k^q_{c_1 c_2} \tag{3.4}$$

where $X \in \{0,1\}^{n_1 \times n_2}$ denotes the node correspondence, for example, if i_1^{th} node of $\breve{\mathrm{C}}_1$ and the i_2^{th} node of $\breve{\mathrm{C}}_2$ correspond, $x_{i_1 i_2} = 1$. $k^p_{i_1 i_2}$ is an element of K_p in i_1^{th} row and i_2^{th} col, $k^q_{c_1 c_2}$ is an element of K_q in c_1^{th} row and c_2^{th} col.

It is more convenient to write $J(X)$ in a quadratic form, $x^T K x$, where $x = vec(X) \in \{0,1\}^{n_1 n_2}$ is an indicator vector and $K \in R^{n_1 n_2 \times n_1 n_2}$ is computed as follows:

$$k^p_{i_1 i_2 j_1 j_2} = \begin{cases} k^p_{i_1 i_2} & \text{if } i_1 = j_1 \text{ and } i_2 = j_2 \\ k^q_{c_1 c_2} & \text{if } i_1 \neq j_1 \text{ and } i_2 \neq j_2 \text{ and} \\ & h^1_{i_1 c_1} g^1_{j_1 c_1} h^2_{i_2 c_2} g^2_{j_2 c_2} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

A factorized graph matching (FGM) method (Zhou & De la Torre 2016) is used to develop an initial-free optimization scheme with no accuracy loss to address the non-convex issue. This method divides matrix K into many smaller matrices. Using these smaller matrices, the graph matching optimization problem can be transformed to iteratively optimize the following non-linear problem:

$$\max_X J_\alpha(X) = (1-\alpha) J_{vex}(X) + \alpha J_{cav}(X) \tag{3.6}$$

where J_{vex} and J_{cav} are two relaxations in FGM (Zhou & De la Torre 2016).

Enhanced factorized graph matching. Although FGM iteratively uses a different α to apply the Frank-Wolfe (FW) algorithm to avoid local optimal, it still exists to some extent. To effectively deal with the local optima in FGM, we improve the algorithm by considering global geometry

constraint and introduce a new iteration method to solve the new algorithm. The improved energy function is :

$$\begin{aligned} \max_X J_\alpha(X) = (1-\alpha)J_{vex}(X) + \alpha J_{cav}(X) \\ + J_{smooth}(X) \end{aligned} \tag{3.7}$$

As our registration problem only has rigid rotation and translation, these rigid transformation relations always have neighbor projection errors nearby. We use this property to avoid the local minima and obtain more accurate transformation relations. We design this regulation term by considering the projection difference of neighboring correspondence points. $J_{smooth}(X)$ is defined as

$$J_{smooth}(X) = -\sum_{i \in X} \sum_{j \in D} \frac{|\|p_i - p_j\| - \|p_{im} - p_{jm}\||}{(n_1 * n_2)} \tag{3.8}$$

where D represents connection points with point i, p_{im} is the matched point of p_i and p_{jm} is the matched point of p_j. We can easily obtain these points in D by searching matrix G in the graph.

To optimize this nonlinear problem, we use FW (Zaslavskiy, Bach & Vert 2009), which iteratively updates the solution of $X^* = X + \lambda Y$. Given an initial X_0, we update X through optimal direction Y and step size λ. As a smooth term needs a correspondence relation, we divide the computation of optimal direction Y into two steps: (1) compute initial Y_0 using J_{vex} and J_{cav}. We compute an initial Y_0 by solving the Hungarian algorithm which is linear programming similar to FGM (Zhou & De la Torre 2016). (2) computes the final Y by using J_{vex}, J_{cav} and J_{smooth}. We compute the energy of the smooth terms using Y_0 and obtain the final Y using the new energy. As the computation of Y involves linear programming, adding one more computation step of Y is not computationally costly. Similar to the FGM strategy, we also use 100 times iteration to discard the inferior temporary solution and compute an alternative solution using another FW step to optimize J(X). The final transformation matrix is computed in the next stage, following optimization.

3.3.5 Transformation estimation

Our goal is the registration of two cross-source point clouds. As the results of the graph matching contain a small number of outliers, we cannot use these results directly to compute the transformation matrix (used to combine two point clouds into a coordinate system). We need to remove the outliers to obtain the final transformation matrix. We use 3D RANSAC (Papazov & Burschka 2010) to remove the outliers, after which we use these inners to compute the transformation matrix and perform the transformation for the point clouds. The transformation matrix may sometimes still contain small errors, so to deal with this situation, we add an ICP step to locally refine the registration after the outlier removal process. After completing these steps, we register the two cross-source point clouds together. The pseudo code of the complete registration algorithm is shown in Algorithm 3.1.

3.4 Experiments

The proposed method provides a solution to the cross-source point cloud registration problem. In this section, we conduct comparative experiments with many state-of-the-art registration methods: first, we compare the performance of the method on same-source datasets, and then conduct thoroughful experiments on challenging cross-source datasets.

3.4.1 Experimental setup

For comparison purposes, we select the representative 3D registration algorithms ICP (Best & McKay 1992), Go-ICP (Yang et al. 2013), 4PCS (Aiger et al. 2008), super-4PCS (Mellado et al. 2014), TPS-RPM (Chui & Rangarajan 2000), GMMReg (Jian & Vemuri 2011*a*), CPD (Myronenko & Song 2010) and JP-MPC (Evangelidis & Horaud 2018) as our compared methods. Experiments cannot be conducted on a large number of point cloud registrations using TPS-RPM and JR-MPC due to the memory cost,

Algorithm 3.1 Pseudo-code of the registration algorithm.

Require: Cross-source point clouds.

Ensure: Registration result and Transformation matrix.

Scale normalization by Eq. 3.1.
Macro/micro Structure extraction.
Graph construction using Eq. 3.3 and Eq. 3.2.
Initialize X to be a doubly stochastic matrix;
for $\alpha = 0 : 0.01 : 1$ **do**
  for $nIt = 1 : 100$ **do**
    Compute J_{vex} and J_{cav} from X_0
    Compute Y_0 using J_{vex} and J_{cav}
    Compute J_{smooth} using Y_0 as Eq. 3.8
    Compute Y using J_{vex}, J_{cav} and J_{smooth}
    Compute the update direction $Y = Y - X_0$
    Compute update step λ
    Compute the updated X and set $X_0 = X$
  end for
end for
Transformation estimation

so to make a fair and reasonable comparison, we downsample the original point cloud and let the number of points be approximately 2000.

For the same-source database, we conduct a quantitative evaluation experiment with the 3D models "Bunny", "Lucy" and "Armadillo" from the Stanford 3D scanning repository[1]. We only consider points with positive z coordinates. For each view, following (Evangelidis & Horaud 2018), the original models are rotated in the xz-plane and the points with negative z coordinates are rejected. In this way, only a part of the object is viewed in each set; the point sets do not fully overlap, and the extent of the overlap depends on the rotation angle, as in real scenarios.

There are three types of cross-source database:

Database A: KinectFusion and Phones' RGB camera. We build a database with four sets of cross-source objects, which are typical examples of the different properties of cross-source point clouds. We use KinectFusion to build one source, and use VSFM to build another source for images which are captured by IPhone 6S Plus. As KinectFusion uses a physical device to capture 3D points, it can usually obtain dense and uniform point clouds on an object's surface. However, VSFM is a method by which 3D point clouds are built from 2D images. It uses keypoints to initially build highly accurate 3D points and uses CMVC (Furukawa, Curless, Seitz & Szeliski 2010) to build more dense 3D points. These two sources are typical examples of cross-source problems, as previously discussed.

Database B: KinectFusion and KinectFusion's RGB camera. We build the database in the following steps: Step 1, the original KinectFusion SDK [2] is revised to output the image sequence and camera pose of each image when capturing KinectFusion point clouds. Step 2, another point cloud is computed using these images and VSFM. A set of camera poses is computed using VSFM. As these two cross-source point clouds come from the same set of image sequences, the camera poses of KinectFusion and VSFM should be

[1] https://graphics.stanford.edu/data/3Dscanrep/3Dscanrep.html

[2] https://www.microsoft.com/en-au/download/details.aspx?id=40276

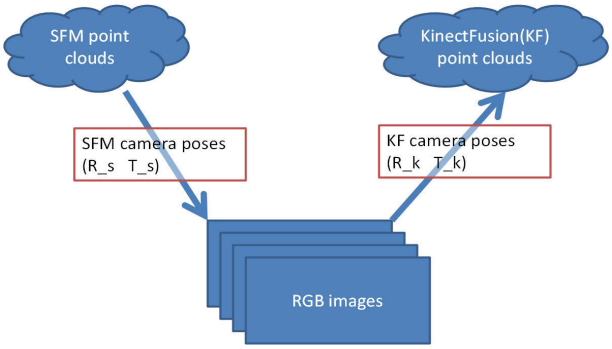

Figure 3.5: Theory of database B build-up. VSFM point cloud is back-projected into image coordinate system and is re-projected into KinectFusion coordinate system

the same. Using this theory, a cross-source point cloud database is produced. The theory is illustrated shown in Figure 3.5. The VSFM point cloud is back-projected into the image coordinate system and then re-projected into the KinectFusion coordinate system. To avoid the inaccuracy of camera pose computation in VSFM and KinectFusion, we consider many poses whose reprojection error is less than σ (σ=0.5), and use these camera pose center points and the least-squares method to compute the final rigid transformation between these two camera center points. The rigid transformation matrix is built on critical prior information and can therefore be used as ground-truth. These benchmark data contain 13 datasets and can be used to perform quantitative evaluation for cross-source point cloud registration.

Database C: Synthetic cross-source point clouds. We build the synthetic datasets according to the cross-source properties. Simulating the cross-source problems discussed in Section 3.1, we build the synthetic datasets in three steps. Step 1: Different density and different viewpoints. We up-sample the original point cloud by adding one point to the gravity center of each triangle of the original surface. We then remove all points whose z coordinate is less than 0 in the upsampling point cloud, and obtain view 1 as S1. The coordinate system is rotated 60° relative to the y axis and down-samples every 3 points. We obtain view 2 by removing all the $z \leq 0$ points. Step

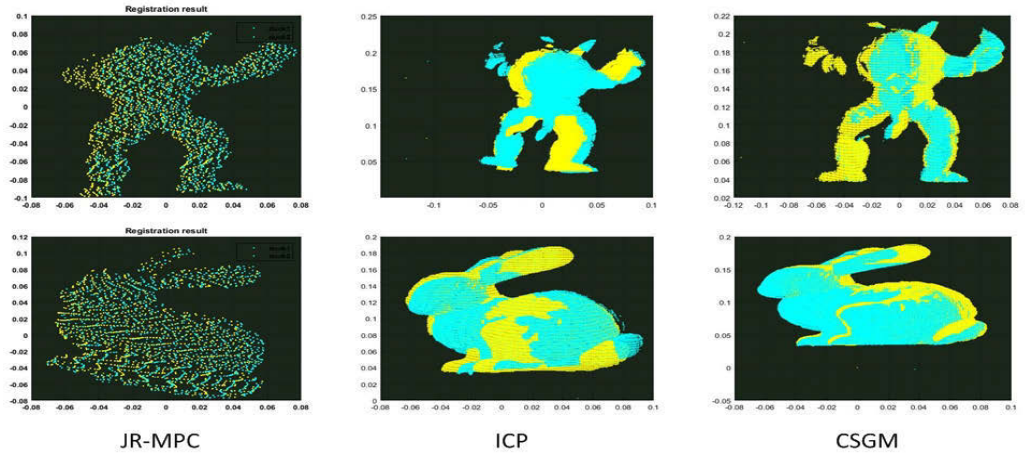

Figure 3.6: Two point clouds registration results on same-source datasets.

2: Missed point cloud construction. Starting from view 2, we randomly delete ten parts in the plane to simulate a VSFM point cloud. Step 3: Rigid transformation. A random scale of 3 to 5, a random rotation matrix in the x, y, z axis of 30° to 60°, and a random translation in the z axis of 0 to 50% of the largest point-point distance are added to view 2. Step 4: Construction of noise and outliers. 40DB of noise is added to the original view 2 point cloud. The outliers are constructed by down-sampling the original view 2 to 30% and adding random offset[3] to the coordinate of the down-sampled point cloud. The noise and outliers are combined to form the final point cloud S2. The S1 and S2 point clouds are simulating cross-source point clouds which perceive the cross-source problems. Ten cross-source datasets are synthesized using Stanford 3D objects[4]. Figure 3.7 shows one sample of the synthetic datasets.

We first compute the radius of the point clouds for parameter setting by $radius = max||P_i - \overline{P}||_2$, where $\overline{P} = (\sum_{i=1}^{N} P_i)/N$ is the centroid point of the point cloud. To retain the same density and the same cross-source point cloud structure, we set the radius of the super voxels as 1% of the point cloud radius

[3]offtset ranges from 0 to 1% of the largest point-point distance
[4]http://graphics.stanford.edu/data/3Dscanrep/

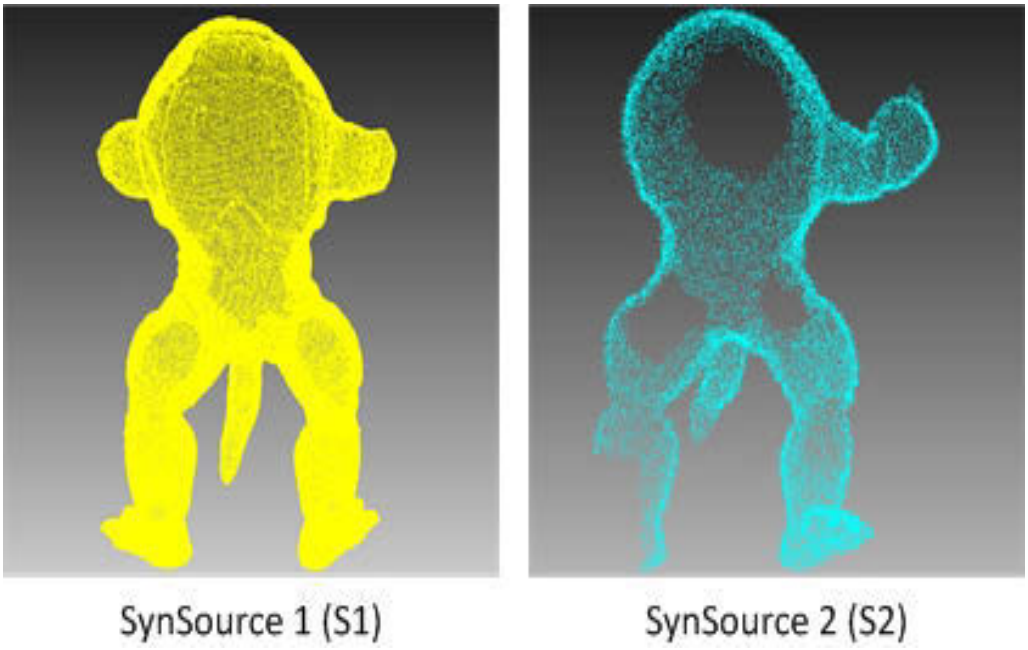

Figure 3.7: Samples of synthetic cross-source datasets.

for both the KinectFusion and SFM point clouds. For the proposed method, we first compute the transformation matrix on macro and micro structures and then use the transformation matrix to perform transformation on the original cross-source point cloud.

3.4.2 Experiments on same-source point cloud datasets

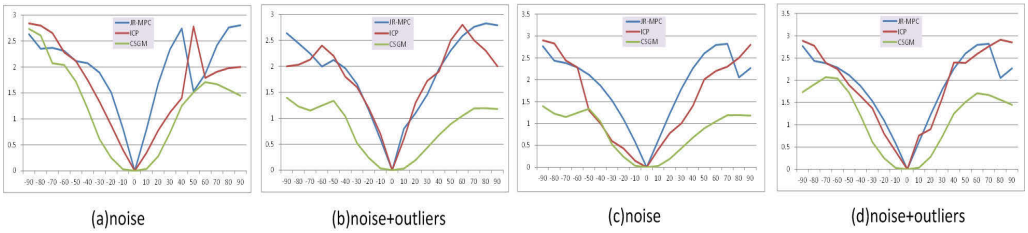

Figure 3.8: RMSE as a function of the overlap (rotation angle) when two point sets are registered (SNR=20dB, 30% outliers) (a),(b) "Armadillo" (c), (d) "Lucy".

We use the root-mean-square error (RMSE) of the rotation parameters for the registration error since translation estimation is not challenging. We select "Armadillo" and "Bunny" with 30° and 45° respectively(SNR = 10db and 20% outliers).

Table 3.1: RSME results of the JR-MPC, ICP and CSGM.

RSME-D	JR-MPC	ICP	CSGM
Armadillo	1.456	1.725	0.508
Bunny	1.789	2.022	1.792

Extensive evaluation and comparison of registration methods has been conducted by JR-MPC on same-source databases. We only run JR-MPC, ICP and the proposed method(CSGM) on the same-source database. Table 3.1 shows the quantitative comparison results. Note that ICP is more affected by the presence of outliers as a result of the one-to-one correspondence and incurs a higher rate of error. JR-MPC demonstrates similar performance to the proposed method, because GMM models perform well when the overlapping areas do not have a significant amount of missing data or the scale problem. We can see from this experiment that the proposed method is robust to outliers, noise and angle variations on same-source point clouds. The visual results are shown in Figure 3.6.

In addition, we test the robustness of the algorithms in terms of the rotation angle between two point clouds to capture the difference degree of the angles. We register the points under different angles from $-90°$ to $90°$ and use RMSE to test the performance. The results are shown in Figure 3.8 and it can be seen that the angles have a different effect on the final error. As the proposed method uses a macro and micro structure to describe the point clouds, it shows robustness in dealing with outliers, noise and missing data on same-source database. However, the error increases when the rotation angle increases, similar to other methods. With the increase in the rotation angle, the outliers and the mismatched parts become a larger proportion of

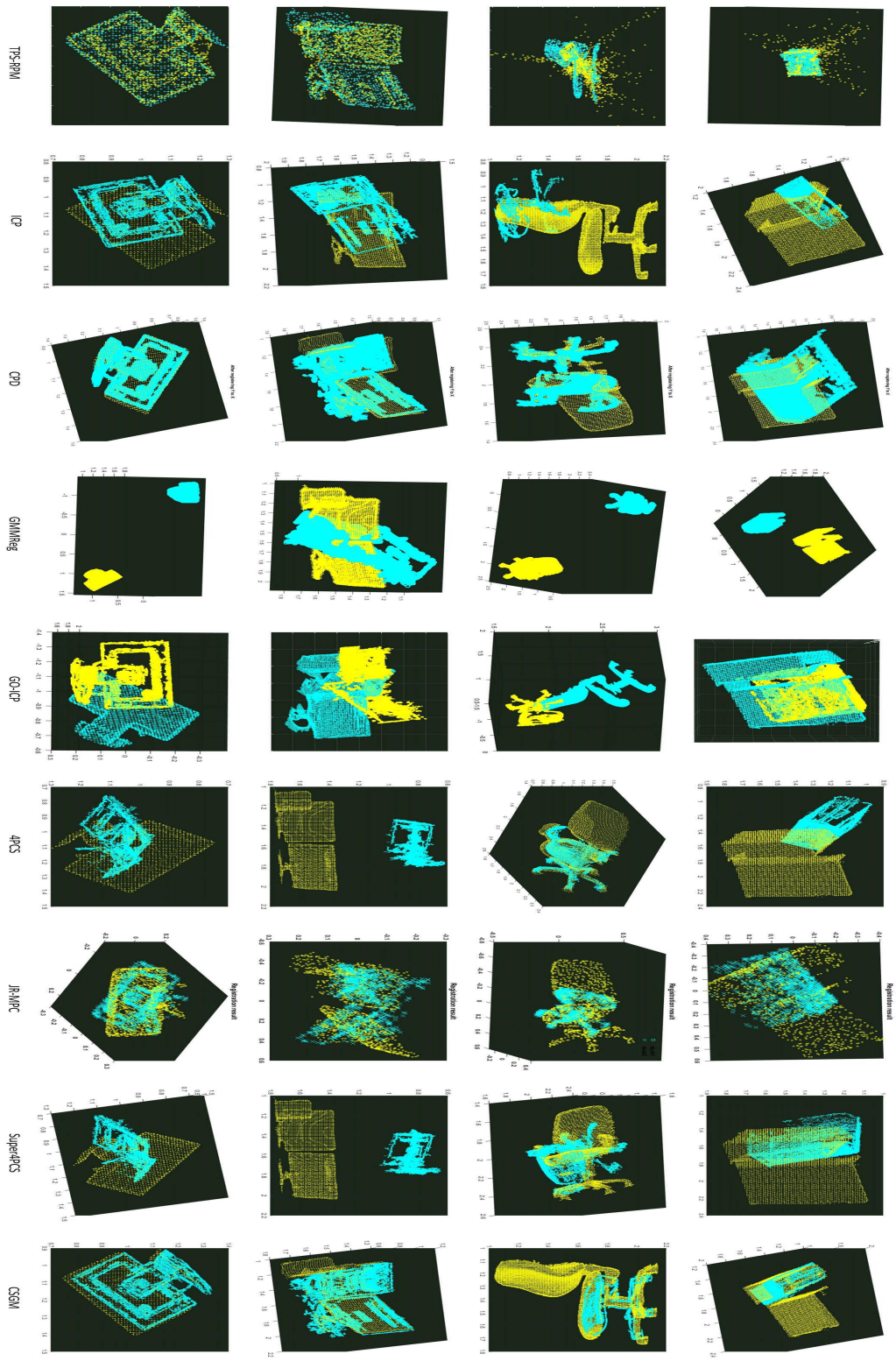

Figure 3.9: Cross-source point cloud registration results on Database A.

each point cloud.

3.4.3 Qualitative evaluation on real cross-source point clouds

As discussed previously, cross-source point clouds have a large variations in density, scale, angle and missing data which makes the already difficult point cloud registration problem even more challenging. To test the ability of our method to register cross-source point clouds and compare with other related methods, we conduct qualitative analysis experiments on four real cross-source datasets: *Twobox, Chair, Threemonitor and Monitor*. To make a thorough comparison, TPS-RPM (Chui & Rangarajan 2000), ICP (Best & McKay 1992), CPD (Myronenko & Song 2010) , GMMReg (Jian & Vemuri 2011*a*), Go-ICP (Yang et al. 2013), 4PCS (Aiger et al. 2008), JP-MPC (Evangelidis & Horaud 2018) and super-4PCS (Mellado et al. 2014) are selected as our comparison methods. Since many of the selected methods are unable to handle the scale problem, we first normalize the scale difference for ICP, Go-ICP, 4PCS, super-4PCS, TPS-RPM, GMMReg and JP-MPC using our scale normalization method. In our proposed method, scale normalization is an integrated step.

Figure 3.9 shows the final registration results which indicate that the proposed method gives successful registration results, whereas the other methods fail in almost all cases. Note that TPS-RPM obtains good result in *Threemonitor* and *Monitor*, but fails in *Twobox* and *Chair*. Also, TPS-RPM is a non-rigid registration method. The proposed method obtains good results in cross-source datasets because it describes the micro and macro structure of point clouds, and uses the new optimization method to obtain correspondence relations.

Note that we do not iteratively conduct enhanced graph matching and outlier detection (RANSAC). We find that when we use the outlier detection method to remove graph nodes, the graph structure in some cases is totally different. As a alternative solution, we use ICP to smoothly refine the graph

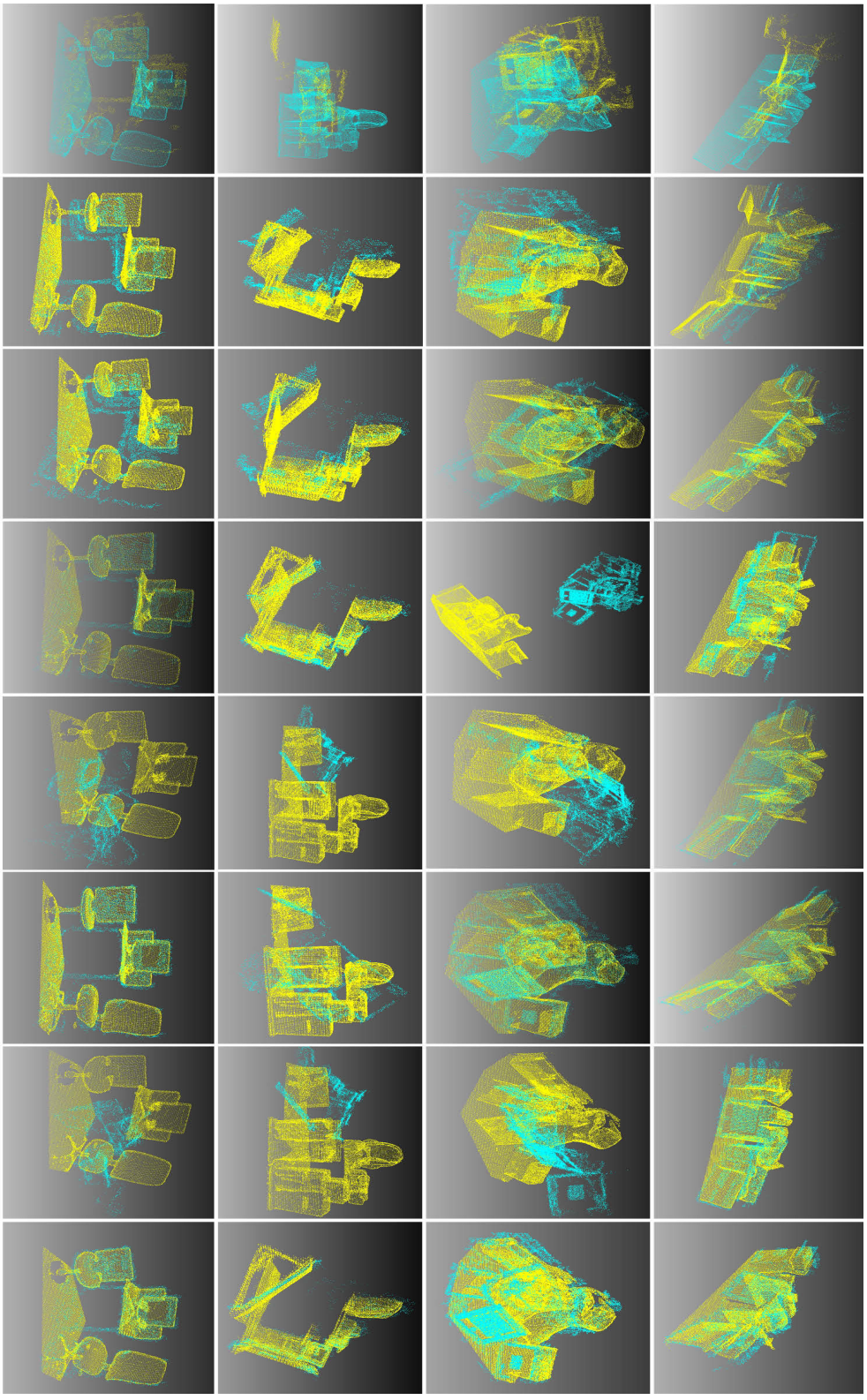

Figure 3.10: Selected visual effect of cross source point clouds registration results on the Database B. Rows are datasets and columns are methods.

matching result to obtain a final registration result.

3.4.4 Quantitative evaluation on real and synthetic cross-source point clouds

To test the ability of the proposed method, we conduct quantitative evaluation on real and synthetic cross-source databases.

We first conduct quantitative evaluation on Databases B which contains real cross-source point clouds. We compare it in the quantitative evaluation experiments with methods that deal with rigid registration. Based on our knowledge, we compare our proposed method with ICP (Best & McKay 1992), GO-ICP (Yang et al. 2013), GMMReg (Jian & Vemuri 2011*a*), JP-MPC (Evangelidis & Horaud 2018), CPD (Myronenko & Song 2010) and 4PCS (Aiger et al. 2008) and super-4PCS (Mellado et al. 2014) on a cross-source database.

Many rigid methods are unable to handle the scale problem. To make a fair comparison, scale normalization is performed before running these methods except for CPD which estimates scale internally. The transformation matrix for each comparison method is then computed and these matrices are used for quantitative evaluation. In this experiment, the matrices are all transformed from VSFM point clouds to KinectFusion point clouds. The VSFM point clouds are initially performed by using new computed and ground truth transformation matrices. These transformed VSFM point clouds are then compared with the ground truth transformed point clouds. As in (Evangelidis & Horaud 2018), we compare the Frobenius Norm (F-norm) between the newly computed matrices and the ground truth transformation matrices. To obtain a better visual representation of comparison results, we use $log(RSME)$ as the final performance value. The smaller the value, the better performance of the algorithm. We also compute the mean of the F-norm of all 13 datasets for each method and the results are shown in Figure 3.11.

Figure 3.11 shows the quantitative evaluation results. It illustrates the

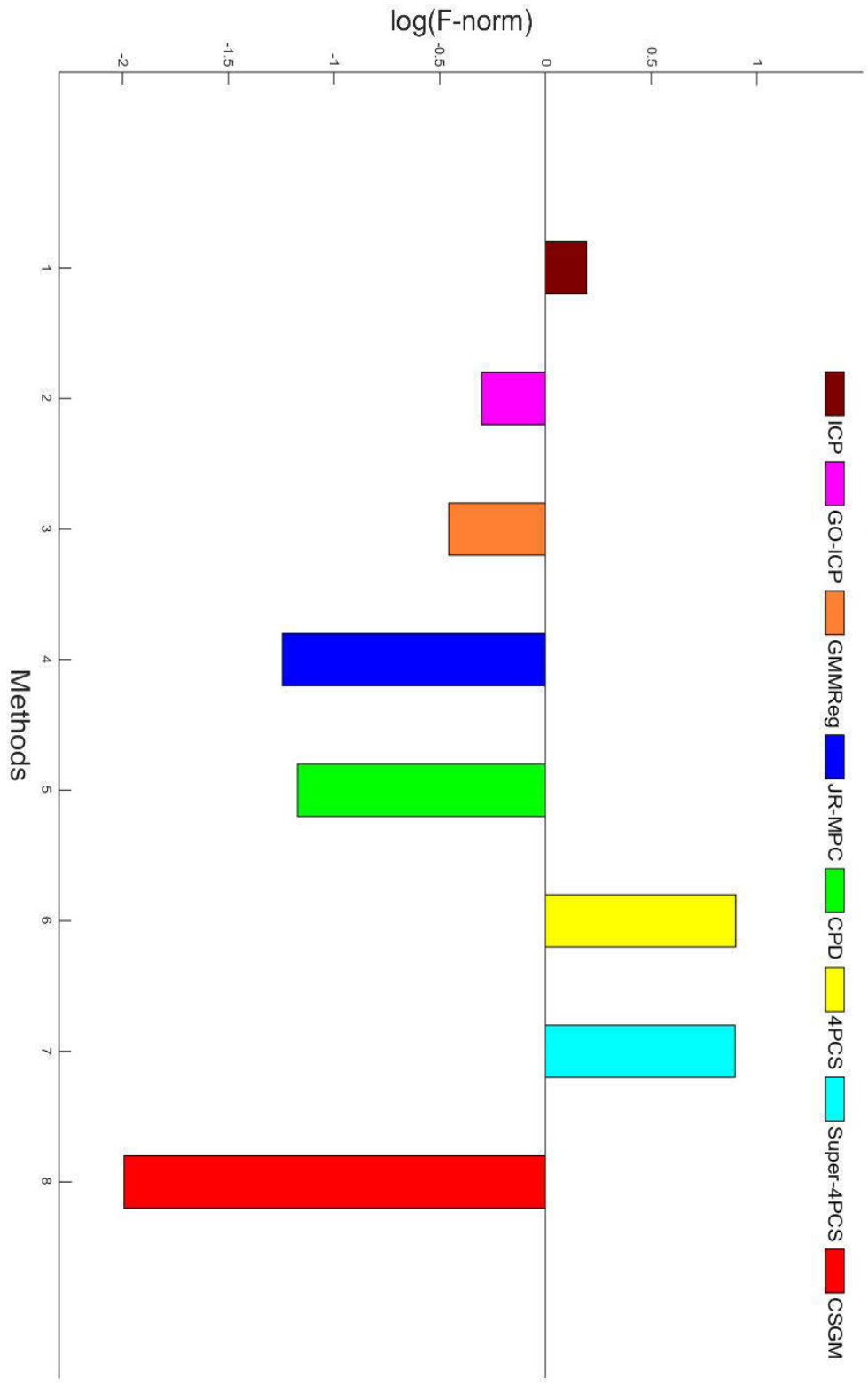

Figure 3.11: Quantitative evaluation results of mean F-norm between transformation matrices on Database B.

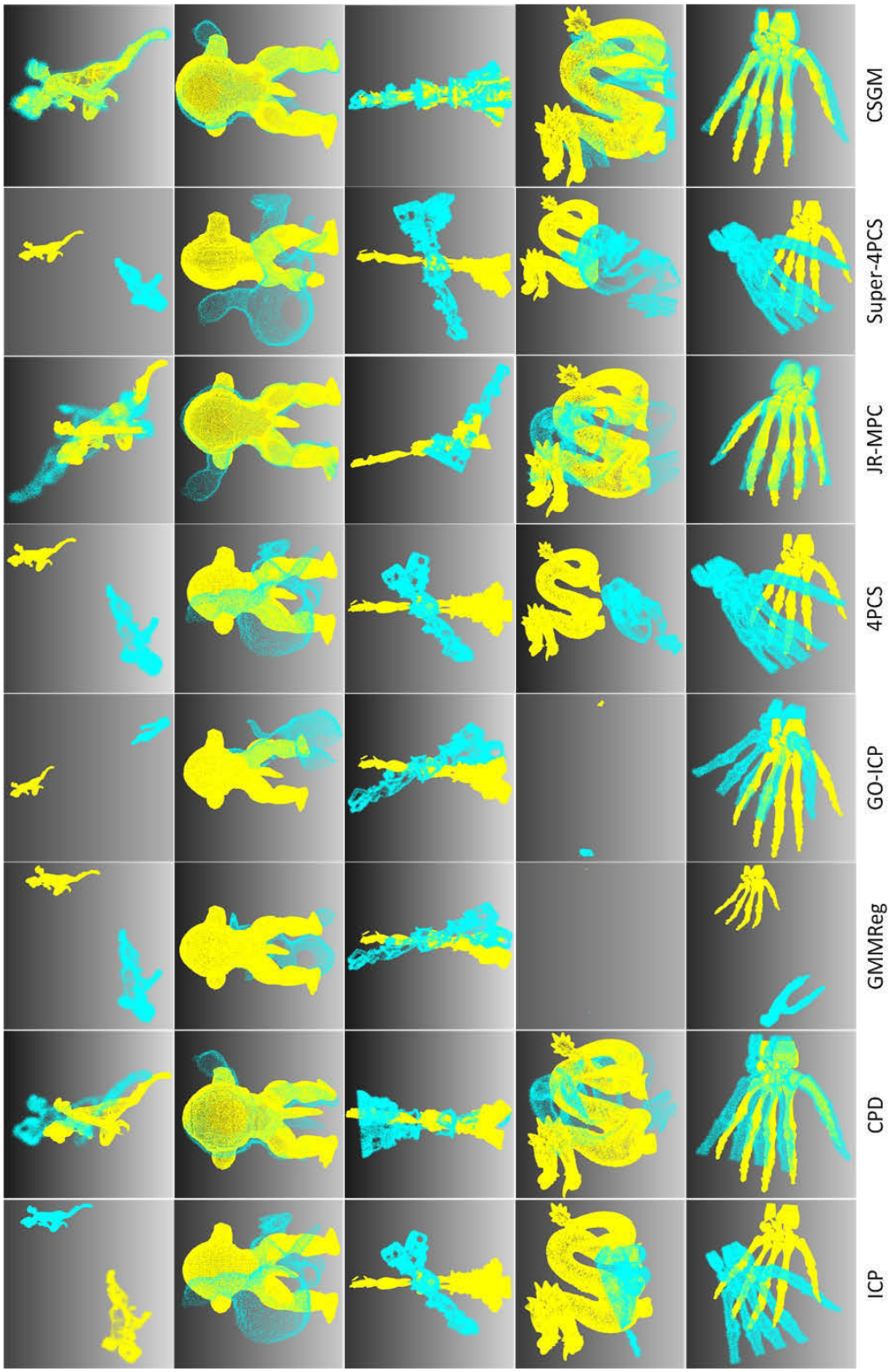

Figure 3.12: Visual effect of registration results on Database C. Rows are different datasets and columns are methods.

4PCS and Super-4PCS obtain worst results, and ICP follows. It is because the point-point level strategy shows poor ability in cross-source problems. The GMMReg, JR-MPC and CPD show more robust and higher accuracy than other comparison methods; to some extent, they demonstrate the advantage of using the statistical property. The proposed CSGM method obtains the highest accuracy on all dataset. This is because we use the macro structure to globally register two point clouds with little attention to the detail, and use the micro structure to accurately register the two point clouds. We also use RANSAC and ICP to further improve the accuracy and robustness.

Figure 3.10 shows several sample visual results of these methods. The results show that the proposed CSGM clearly achieves better results than the other methods. Go-ICP and JR-MPC obtain similar results to the proposed CSGM in the fourth row dataset. Because of the BnB strategy in Go-ICP and the generative strategy in JR-MPC, good results are obtained if the scale normalizes very well and no large data are missing. If these conditions do not exist, these methods will completely fail. In the first two rows of Figure 3.10, for example, these methods show the results of that failure. However, the proposed CSGM achieves robust and accurate registration results in all cross-source datasets.

The proposed method is also compared on Database C which consists of synthetic cross-source point clouds. Transformation relation is estimated by the compared methods and the proposed method from view 2 to view 1 point cloud. The computed and ground truth transformation matrix are then utilized to transform the synthetic point cloud. The RSME error is computed according to the statistical distance of these two transformed point clouds. Also, we compare the F-norm of the error of difference between transformation matrices.

Figure 3.13 shows the evaluation results of mean RMSE and Figure 3.14 shows the evaluation results of mean F-norm of the computed transformation matrix and the ground-truth transformation matrix on whole ten sets of Database C. The results show that our method achieves robust registration

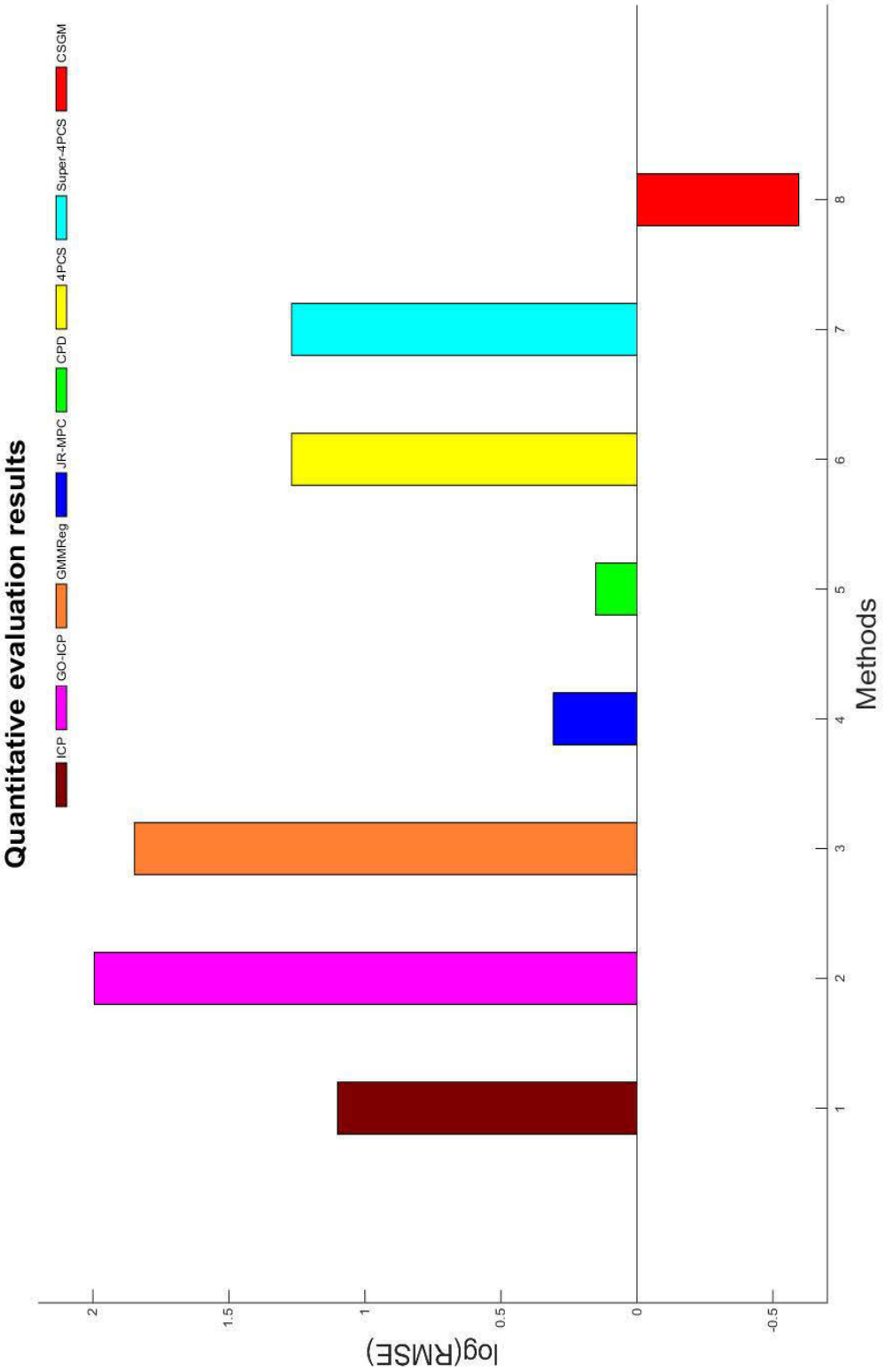

Figure 3.13: Quantitative evaluation results of RMSE on Database C.

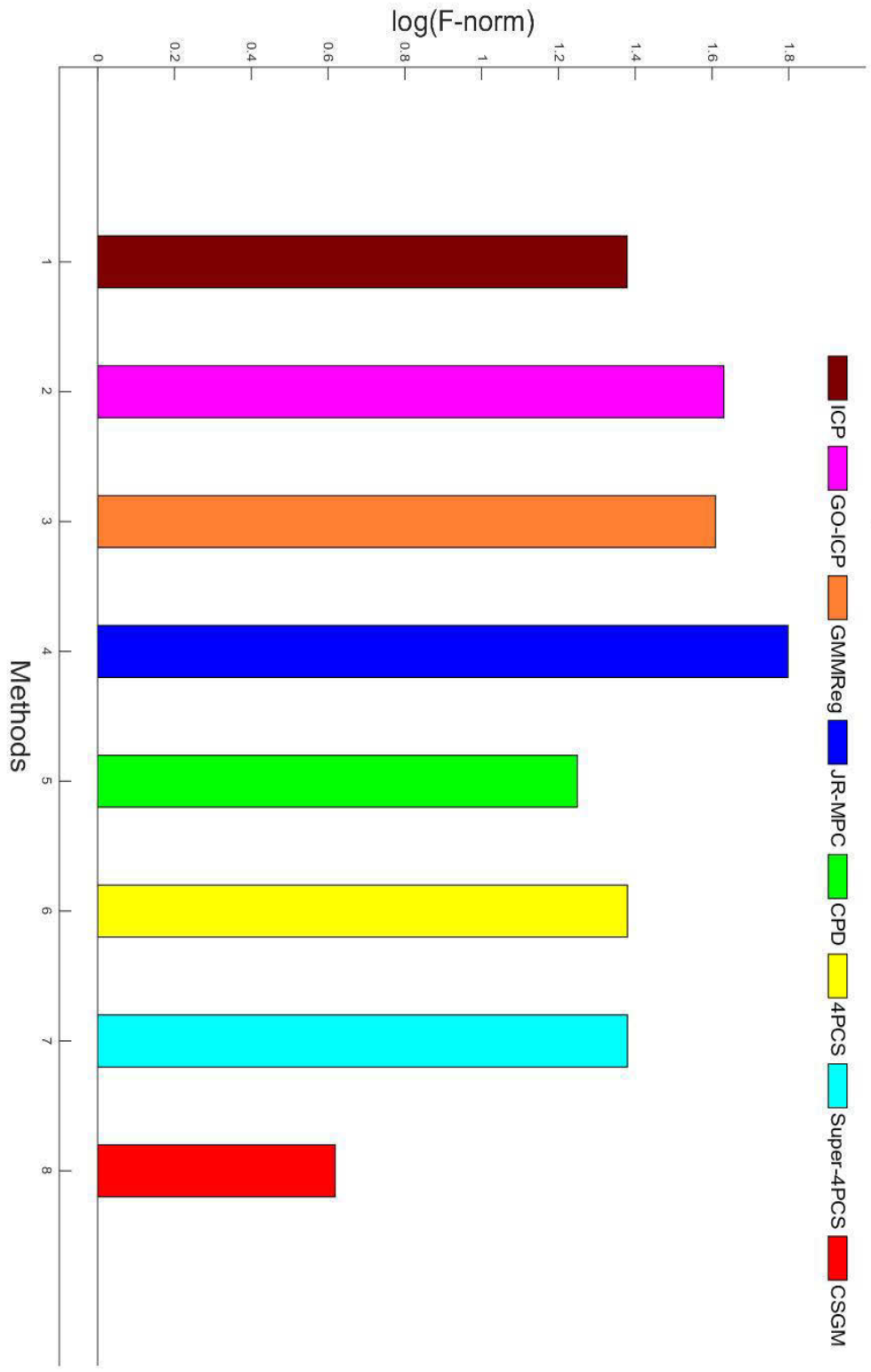

Figure 3.14: Quantitative evaluation results of F-norm on Database C.

results which are better than the other methods. Figure 3.12 illustrates the visual effects of the Synthetic evaluation. The results show that the proposed CSGM obtains visually correct registration results which are clearly better than those of the compared methods.

3.5 Conclusion

In the above section, we proposed a new registration pipeline to deal with the cross-source point cloud registration problem using four novelty components. A scale normalization method was first proposed to eliminate the scale problem. Secondly, a micro and macro structure concept was proposed to describe the point clouds, and a new graph construction method was used to combine these structures. Thirdly, an optimization method was proposed to solve the problem. Lastly, a registration pipeline was proposed which combines the initial correspondence from graph matching and refinement using RANSAC and ICP.

The above segment is the introduction of how to extract structures and utilize the structures of point cloud to solve the registration problem. However, the efficiency is the remaining problem. For each pair of point cloud registration, it costs about 20 minutes. The efficiency is highly impact its application. In the following segments, the efficiency tensor optimization is introduced into registration problem. With two proposed structure components, the registration can be solved with both high efficiency and accuracy.

We do not need the two graphs to be similar number of supervoxel. Our method has the similar generalization performance as graph matching.

However, the method in the above section faces high computation complexity. In the following section, weak regional affinity (global) and pixel-wise refinement (local) components are introduced and an algorithm is proposed to align cross-source point clouds efficiently and accurately.

Chapter 4

Tensor-based matching for cross-source point cloud registration

4.1 Introduction

Structure has already shown great value in solving cross-source point cloud registration problem which is discussed in the above Chapter 3. However, the method of Chapter 3 is using graph to describe the structure information and converts the registration problem into the graph matching problem. We found its computation complexity is extremely high because of the sophisticated optimization strategy. Also, graph construction has the assumption of strong structure. We found that the strong structure assumption is not always the case in cross-source point clouds. In this chapter, we try to explore an efficient and accurate solution to utilize the weak structure information to solve the cross-source point cloud registration.

Definition 4.1 *Same-source point clouds are homogeneous point clouds from same types of sensors. Cross-source point clouds are heterogeneous point clouds from different types of sensors.*

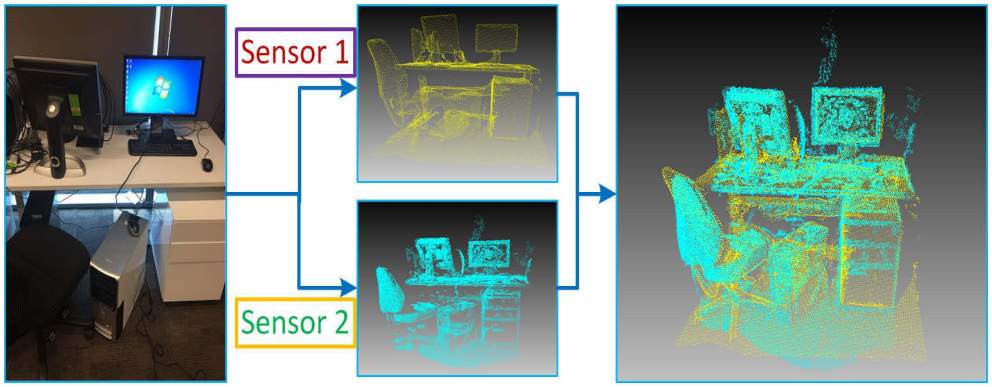

Figure 4.1: Two cross-source point clouds are captured about a same indoor scene. They contain mixture of variants of density, outliers, noise and missing data. The proposed algorithm can successfully register them.

There are several attempts to solve the cross-source point cloud registration. CSGM (Huang, Zhang, Fan, Wu & Yuan 2016) converts registration problem into graph matching problem. Then, ICP refinement (local) is done based on the graph matching results (global). (Peng et al. 2014*b*) proposes a coarse-to-fine (global-to-local) strategy to solve the cross-source point cloud. (Mellado, Dellepiane & Scopigno 2016) utilizes RANSAC and ICP to global align and local refine the cross-source point cloud registration. However, all the existing methods either cannot solve the cross-source point cloud registration accurately or cannot solve the problem efficiently. All the existing registration methods assume the strong structure consistency between two point-cloud sets being registered. However, it is not the case of cross source point cloud registration in which structure consistence is weak. The reliable correspondences which can be located between two sets are sparse. Larger inconsistency caused by different point cloud density, noise model and various outliers significantly degrade the estimation on rigid transformation.

According to the observation (One example is Figure 4.1), any two given cross-source point clouds still hold the intrinsic structure similarity although it is weak and degraded by the various noise, inconsistent point cloud density and outliers. The proposed method is able to discover the salient but weak

geometric structure affinity (not statistic feature affinity) by joining sufficient number of local matching i.e. co-affinity. In order to adjust the mismatching in co-affinity search process, a pixel-wise refinement process is proposed. It is similar to global and local strategy in previous existing methods. Different to previous separated streams, in the proposed method, these two processes i.e. co-affinity search and pixel-wise refinement is unified together. Thus, it can be sorted out in an uniform optimization process. Compared to separated streams, the advantages of uniform optimization process is that feedback between local and global components will be given to the optimization process by jointly considering both local and global information in one unified process.

To mathematically formalize the above motivation, we assemble the weak regional affinity and pixel-wise refinement into unified tensors. The weak regionals are triplet constraints that are stored in third-order tensors. The pixel-wise refinement is point-point or point-plane residual error that is stored in first-order tensor. We select tensor as the mathematics tool to assemble this idea, because tensor provides an elegant and unified mathematics format to assemble the global weak region affinity and local pixel-wise refinement. Then, the correspondent finding problem can be optimized in a unified whole potential tensor space. Compared to previous separated global and local strategy, feedback between these components is given to the whole optimization process thanks to the tensor optimization. To solve the final registration problem, instead of doing tensor optimization once, we propose an iterative tensor optimization solution and a new energy function is formulated to obtain optimal geometric transformation. During the iterative optimization solution, in order to get feedback between transformation matrix estimation and correspondent optimization, the geometric transformation T is integrated into tensor optimization and the tensor space will be updated when the T is updated. When the energy function obtains convergence, both the correspondence and the geometric transformation are optimized.

The contributions of this Chapter are three aspects: firstly, weak regional

affinity and pixel-wise refinement are proposed to keep global and local information in cross-source point clouds where structures are usually weak; secondly, an unified algorithm is proposed to integrate these two components to solve the cross-source point cloud registration.

4.2 Tensor mathematical preparation

In this thesis, a unified algorithm is proposed to by extending the tensor optimization method to solve the cross-source point cloud registration. To better understand the algorithm, in this section, we firstly review tensor mathematical basis, then, the existing cross-source point cloud registration methods using local and global information are reviewed.

4.2.1 Tensor Basis

Tensor is a technique that can be regarded as a follow-up on linear algebra. In classical linear algebra one deals with vectors and matrices. Tensor is generalization of scalars, vectors, and matrices to multidimensional array. Previously, tensors are widely used in physics because they provide a concise mathematical framework for formulating and solving physics problems in areas such as stress, elasticity, fluid mechanics, and general relativity. In this thesis, we introduce tensors into registration problem and use it to concisely represent weak region affinity and pixel-wise refinement. In this section, basic knowledge about tensor is introduced, which is helpful for understanding of the proposed algorithm.

Tensor orders: The order (also degree or rank) of a tensor is the dimensionality of the array needed to represent it, or equivalently, the number of indices needed to label a component of that array. For example, a matrix is represented by a 2-dimensional array in a basis, and therefore is a 2nd-order tensor. A vector is represented as a 1-dimensional array in a basis, and is a 1st-order tensor. Scalars are single numbers and are thus 0th-order tensors.

First-order tensor: First-order tensor H_α is a 1-dimensional array. it is a vector. Each element of the first-order tensor is a scalar.

Third-order tensor: Third-order tensor is a 3-dimensional array. Just as a n-dimensional vector is a combination of n scalar and a $m \times n$ matrix is a combination of m vector with n dimension, a tensor $T_{\alpha\beta\gamma}$ is a combination of α matrix with dimension $\beta \times \gamma$. Figure 4.2 shows a visual example of a third-order tensor. To index the value of a third-order tensor, one index in each dimension is needed. For example, H_{321} and H_{211} in Figure 4.2 can index the value of its correspondent position value.

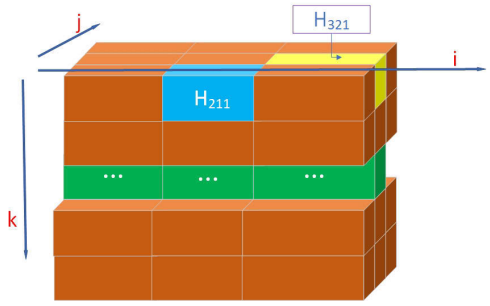

Figure 4.2: Third-order tensor. The value of tensor can be indexed as combination of three indexes in three dimensions.

Conversion between tensor Kronecker product and multiplication: In the following section, we also use the following calculus to calculate the relationship between Kronecker product of tensor and multiplication. According to (Duchenne et al. 2011), (Regalia & Kofidis 2000),

$$\begin{aligned} S(\bar{X}) &= \sum_{ijk} H_{ii'jj'kk'} X_{ii'} X_{jj'} X_{kk'} \\ &= \bar{H} \otimes_3 \bar{X} \otimes_2 \bar{X} \otimes_1 \bar{X} \end{aligned} \tag{4.1}$$

where H is a 6D supersymmetric tensor and $\bar{H} = vec(H)$ is a third-order tensor. X is an assignment matrix and $\bar{X} = vec(X)$ is a first-order tensor. The elements of H describe the similarity of point correspondence (detailed in Section 4.3). X is an assignment matrix that only contains 0 or 1, where $X_{ij} = 1$ means point i matches to point j and 0 otherwise. The product

$X_{ii'}X_{jj'}X_{kk'} = 1$ means that the triplet points $\{i, j, k\}$ are all matched with points $\{i', j', k'\}$ and 0 otherwise. In the optimization process, $H_{ii'jj'kk'}$ will be added to the total score function if triplet points are matched to triplet points. Here, $H_{ii'jj'kk'}$ is a similarity measure, where the value will be high if the feature of triplet points $\{i, j, k\}$ is similar to points $\{i', j', k'\}$. The larger the value is, the more probability triplet point pairs are correspondent points.

Power iteration: The power iteration method is a very simple algorithm for computing the main eigenvector of a matrix. The algorithm is guaranteed to converge geometrically to the main eigenvector of the input matrix (Golub & Van Loan 2012). To compute the main eigenvector b of a diagonalizable matrix A, the power iteration algorithm starts from a vector b_0, then do the following equation recurrently:

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|} \tag{4.2}$$

The above equation shows the matrix A is left multiplied by the vector b_k and normalized.

4.2.2 local and global mixture methods for cross-source point cloud registration

They are several attempts to solve the cross-source point cloud registration problem. (Peng et al. 2014*b*) proposes a coarse-to-fine method. It uses Ensemble of Shape Functions (ESF) to globally select the potential matching regions and uses iterative closest point (ICP) to refine the local registration. This is the first method to deal with cross-source point cloud and it faces the same limitation of ICP, such as sensitive to noise, outliers, initialization and missing data. (Peng et al. 2014*b*) uses several manually handling to deal with the limitation of ICP. However, it can be less precise for large disturbances of the initial alignments and loses its advantages in unstructured environments.

Based on (Peng et al. 2014*b*), (Huang, Zhang, Wu, Fan & Yuan 2016) proposes a method to use Gaussian mixture models (GMM) to replace the

original ICP. Because GMM utilizes the statistic information, it shows more robust results than ICP. It follows the same coarse-to-fine framework and utilizes global and local separately. (Huang, Zhang, Wu, Fan & Yuan 2017*a*) improves (Huang, Zhang, Wu, Fan & Yuan 2016) by solving the scale problem. However, there are an increasing number of GMM models to robustly represent point clouds. When the point number increases to tens of thousands or millions, these methods are impractical in terms of both computational and memory cost. On the other hand, the GMMs depicting two point clouds are shown a lot of difference when there is missing data and large noise and outliers variations in cross-source point clouds, which makes the registration inaccurate or even fail. Global methods.

CSGM (Huang, Zhang, Fan, Wu & Yuan 2016) extracts macro and micro structures of point cloud and integrates them into graph. Then, the registration problem is converted into graph matching problem. This method uses ICP to refine the registration results. This method also use graph matching to obtain the initial global alignment and use ICP to conduct the local refinement. Also, CSGM faces high computation complexity and graph construction relies on strong structure maintained.

(Mellado et al. 2016) solves the registration by using RANSAC and a novel scale estimation method. To deal with density, they use downsample strategy. However, this method faces the same problems of RANSAC, such as outliers and different noise model.

Tensor is a powerful tool in storing data and there is a lot of efficient optimization algorithms. Recently, (Zeng, Wang, Gu, Samaras & Paragios 2016) convert surface registration into tensor optimization which is composed of the elements of geometric and appearance matching costs, as well as higher-order deformation priors. (Duchenne et al. 2011, Shi, Ling, Hu, Xing & Zhang 2016) use tensor optimization to solve graph matching problem, which casts the corresponding points finding problem as tensor optimization problem. However, there are no methods tailored in the challenging cross-source point cloud registration problem. Compared to the existing tensor-based methods

(Zeng et al. 2016, Duchenne et al. 2011, Shi et al. 2016), we integrate pixel-wise refinement and weak region affinity to solve the cross-source point cloud registration problem. Mathematically, (Duchenne et al. 2011) is the most similar to the proposed algorithm, the algorithm is

$$\begin{aligned} \arg\max_X S &= \sum_{ijk} H_{ii'jj'kk'} X_{ii'} X_{jj'} X_{kk'} \\ &\quad + H_{ii'jj'} X_{ii'} X_{jj'} + H_{ii'} X_{ii'} \\ &= \bar{H}_{ijk} \otimes_3 \bar{X} \otimes_2 \bar{X} \otimes_1 \bar{X} \\ &\quad + \bar{H}_{ij} \otimes_2 \bar{X} \otimes_1 \bar{X} + \bar{H}_i \otimes_1 \bar{X} \end{aligned} \tag{4.3}$$

where $H_{ii'jj'kk'}$, $H_{ii'jj'}$, $H_{ii'}$ are 6D, 4D and 2D tensor. The elements of H describe the similarity of point correspondence (detailed in Section 4.3). X is an assignment matrix that only contains 0 or 1, where $X_{ij} = 1$ means point i matches to point j and 0 otherwise. The product $X_{ii'} X_{jj'} X_{kk'} = 1$ means that the triplet points $\{i, j, k\}$ are all matched with points $\{i', j', k'\}$ and 0 otherwise. In the optimization process, $H_{ii'jj'kk'}$ will be added to the total score function if triplet points are matched to triplet points. Here, $H_{ii'jj'kk'}$ is a similarity measure, where the value will be high if the feature of triplet points $\{i, j, k\}$ is similar to points $\{i', j', k'\}$. The larger the value is, the more probability triplet point pairs are correspondent points. The score contributions of 4D and 2D tensor components are similar, which describe the edge-edge and point-point similarity. Based on our experiments, edges similarity are not robust to utilize in 3D point cloud registration. According to eq 4.1, the sum of similarity can be converted into tensor multiplication. The solution of equation 4.3 can be estimated by power iteration algorithm. Therefore, the correspondence map estimation is converted into tensor optimization.

The above algorithm only optimizes once in a stable tensor space. The correspondence solution can only estimated in the current tensor space and the solution is local optimal (Duchenne et al. 2011). In the registration problem, the goal is to estimate transformation matrix by iteratively conducting the correspondence and transformation estimation. In this iterative proce-

dure, the similarity of point pair changes when one point cloud is transformed by an estimated transformation matrix. In other words, the value of the current tensor is changed after transformation. Therefore, the tensor space is also needed to be optimized in order to obtain better correspondence for registration problem. Especially in cross-source point cloud, the current tensor space is needed to be optimized due to the large variation of cross-source problem. In this thesis, the proposed method updates the tensor space based on the feedback from transformation estimation, so that we can obtain better correspondence points at the optimized tensor space. Mathematically, we integrate spatial transformation T into tensor optimization process and get feedback between correspondence estimation and transformation estimation.

4.2.3 Summary

The above existing methods separately use local and global information and they all suppose strong structures. In this chapter, weak region affinity and pixel-wise refinement are proposed, which are more suitable to describe the weak structures in cross-source point clouds. Then, an algorithm is proposed to uniformly integrate global weak region affinity and local pixel-wise refinement into an tensor optimization process. We select tensor mathematic tool because tensor optimization is optimized at a whole potential tensor space so that feedback is considered between global and local components by the iterative optimization process. To effectively solve the registration problem, in order to obtain feedback between transformation matrix estimation and correspondence estimation, the geometric transformation T is integrated into tensor optimization and the tensor space will be updated when the T is updated. We proposed an Expectation-maximization (EM) solution to solve the problem. In the following section, we will describe problem formulation and the optimization solution.

4.3 Proposed algorithm

To integrate the feedbacks, an iterative tensor optimization algorithm is proposed to integrated transformation matrix T into tensor optimization. In each tensor space, the correspondence estimation is solved by tensor optimization. Based on the updated correspondence, the transformation matrix is updated. Then, the tensor space is adjusted based on the updated transformation. These processes do iteratively until convergence. The advantages are that two kinds of feedback are integrated into the proposed method: feedback between global and local components and feedback between transformation estimation and correspondence estimation. To achieve this goal, we integrate spatial transformation T into equation 4.3 and only consider global region affinity and pixel-wise refinement. Therefore, the objective function is maximized:

$$\begin{aligned} S &= \sum_{i,j,k} H_{ii'jj'kk'}(T) X_{ii'} X_{jj'} X_{kk'} + H_{ii'}(T) X_{ii'} \\ &= \bar{H}_{\alpha\beta\gamma}(T) \otimes_3 \bar{X} \otimes_2 \bar{X} \otimes_1 \bar{X} + \bar{H}_l(T) \otimes_1 \bar{X} \end{aligned} \tag{4.4}$$

where X (correspondence matrix) and T (transformation matrix) are two parameters need to estimate. $H_{ii'jj'kk'}$ is a 6D supersymmetric tensor. Each node pair's (i.e. P_i^1 and $P_{i'}^2$) similarity contributes a 2D dimension matrix to $H_{ii'jj'kk'}$, i.e. P_i and P_i' can form a 2D matrix, and P_j and P_j' can form another 2D matrix. $H_{ll'}$ is a 2D matrix to describe the pixel-wise similarity. X is the $N_1 \times N_2$ assignment matrix where 1 means two points are matched and 0 otherwise. $\bar{X} = vec(X)$ obtains $N_1 N_2$ vector form of X by concatenating the columns of X. $\bar{H_{\alpha\beta\gamma}} = vec(H_{ii'jj'kk'})$ is a three-order tensor of size $(N_1 N_2)^3$ where each element represents the similarity of two triplets. It is a rewritten of tensor $H_{ii'jj'kk'}$. $\bar{H}_l = vec(H_{ii'})$ is vector form of $H_{ii'}$ by concatenating the columns of $H_{ii'}$, where each element represents the point-point similarity. With a geometric transformation $T(.)$ given, $\bar{H}_{ii'jj'kk'}(T)$ and $\bar{H}_{ii'}(T)$ are the two specific tensors and correspondence matrix X can be estimated by

tensor optimization. With an optimized X, transformation matrix T can be estimated by singular-value decomposition (SVD).

The proposed method has two components: $\bar{H}_{\alpha\beta\gamma}$ represents global weak region affinity and $\bar{H}_l$ represents local pixel-wise refinement. Tensor optimization considers the feedback between global and local components. Updated tensor space algorithm considers the feedback between transformation estimation and correspondence estimation. By considering the above two feedbacks, the registration problem is formulated as equation 4.4. Now, we will give details about how to define two components in cross-source point clouds, and how to optimize the uniform optimization by using Expectation-maximization (EM) process.

4.3.1 Definition of pixel-wise refinement and weak region affinity

In the following, we will introduce how to formulate the above pixel-wise refinement and weak region affinity into tensors. First-order tensor and third-order tensor are utilized to store pixel-wise refinement and weak region affinity separately. In the following, we suppose Point cloud C_1 has N_1 points and point cloud C_2 has N_2 points.

Figure 4.3: First-order tensor $\bar{H}_\alpha$ to represent point-to-point similarity in two point clouds with N_1 and N_2 points. The first-order tensor is constructed by concatenating the columns of H. To index a value of the first-order tensor $\bar{H}_\alpha$, for example the red box $H_{ii'}$, it is $\bar{H}_{(i+i'\times N_1)}$.

- **Pixel-wise Refinement:** Pixel-wise refinement is the potential point-to-point correspondence. Based on the tensor mathematical background in section 4.2.1, in our algorithm, first-order tensor is used

to store pixel-wise refinement represents correspondent similarity. For correspondent similarity, it is computed as the Euclidean distance between pixel-wise point pair:

$$H_{ii'} = exp(-\|f_i - f_{i'}\|^2) \tag{4.5}$$

where H is a $N_1 \times N_2$ correspondent point similarity matrix, $i \in [0, N_1)$ and $i' \in [0, N_2)$. Each element of H stores the similarity of point i of point cloud C_1 and point i' of point cloud C_2.

For the first-order tensor $\bar{H}_\alpha$, it is a $N_1 N_2$ vector by concatenating the rows of a similarity matrix H. The index conversion between $\bar{H}_\alpha$ to $H_{ii'}$ is $i = i + i' \times N_1$ (See Figure 4.3). f_i is the feature vector of point P_i in point cloud C_1 and $f_{i'}$ is the feature vector of point $P_{i'}$ in point cloud C_2. For feature vector, we use 3D point coordinate. The first-order tensor stores local information.

- **Weak Region Affinity:** Weak region affinity is the potential triplet-to-triplet correspondence(triplet points are a simple region). According to tensor mathematical basis in section 4.2.1, in our algorithm, third-order tensor is used to store weak region affinity. In particular, triplet points are selected and are used to represent weak salient structure of cross-source point cloud (triplet points selection is detailed in Section 4.4). To estimation the correspondence between triplet points, we need to compute the similarity of these triplets. The similarity is computed by

$$H_{ii'jj'kk'} = exp(-\|f_{ijk} - f_{i'j'k'}\|^2) \tag{4.6}$$

where $H_{ii'jj'kk'}$ is a 6D supersymmetric tensor such as invariant under permutations of indices in (i, j, k) or i', j', k'. Each element of the 6D tensor stores the similarity of the two triplets. Points (P_i, P_j, P_k) and $(P_{i'}, P_{j'}, P_{k'})$ are two triplet points with the correspondent relations based on their orders. In particular, the above order represent Point P_i of triplet 1 is correspondent to point $P_{i'}$ of triplet 2, and the same

to point correspondence of $(P_j, P_{j'})$ and $(P_k, P_{k'})$. $\bar{H}_{\alpha\beta\gamma}$ is a three-order tensor of size $(N_1 N_2)^3$ which is rewritten from tensor $H_{ii'jj'kk'}$. According to 4.4, the third-order tensor stores global information.

- **Triplet similarity:** For each triplet, we compute cosine value of three inner angles of the correspondent triangle combined by the triplet. Then, a descriptor ($f_{i_s j_s k_s}$, $s \in \{1, 2\}$) is formed to describe the triplet. The similarity between triplets are computed by using their descriptors $f_{i_s j_s k_s}$. Using this similarity computation strategy, all the elements of the above three-order tensor can be computed. In the three-order tensor, each dimension reflects the potential of point-point correspondence. Therefore, three dimensions are same and the node correspondent will be permutation of all points between two point clouds. Therefore, the tensor is a symmetric tensor.

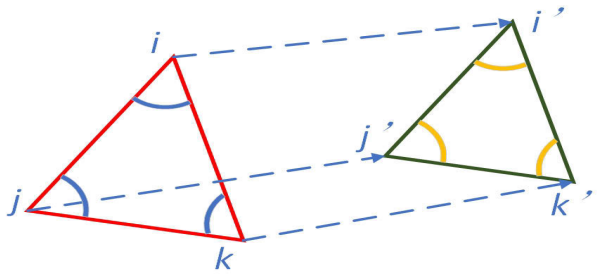

Figure 4.4: Similarity of each triplet point , which is computed by two descriptor of two triplets. For the descriptor of each triplet, it is three cosine value of three inner angles of the triangle.

4.3.2 Uniform algorithm considering global and local components

With a transformation matrix given, the correspondence X estimation is a tensor optimization problem. For correspondence estimation, different from previous methods in separately using global or local information of point

clouds, we uniformly utilize global and local information. In particular, we formulate weak region affinity and pixel-wise refinement into an unified tensor optimization process. Therefore, the feedback between global and local components is utilized in the tensor optimization process. The reason is that the optimization process is optimized in a tensor space with both local and global information. Finding optimal solution in this mixed tensor space will consider both global and local information.

With transformation matrix given, for each tensor optimization, following (Leordeanu & Hebert 2005, Duchenne et al. 2011), we convert the optimization problem as an integer quadratic program. According to section 4.2.1 and equation 4.4, the objective function with known T only can be formulated to estimate correspondence X as following:

$$\begin{aligned} \max_{X}(S(\bar{X})) &= \sum_{i,j,k} H_{ii'jj'kk'} X_{ii'} X_{jj'} X_{kk'} + H_{ii'} X_{ii'} \\ &= \bar{H}_{\alpha\beta\gamma} \otimes_3 \bar{X} \otimes_2 \bar{X} \otimes_1 \bar{X} + \bar{H}_l \otimes_1 \bar{X} \end{aligned} \tag{4.7}$$

The above energy function can be solved by power iteration solution (see section 4.2.1) and the optimal correspondence is obtained based on current transformation matrix T. Then, the transformation matrix T is updated by using the current optimal correspondence and SVD factorization (Best & McKay 1992), and we update the tensor space by transforming the point cloud. In the proposed method, we iteratively and simultaneously solve the optimal correspondence X as well as the optimal transformation T. In the following section, we will introduce how to formulate two processes into one energy function.

For the scale computation, we compare triplet correspondent edges and compute the mean ratio as the final scale:

$$s = \frac{\sum_{i=1}^{n-1} (r_{ai}/r_{bi})}{n-1} \tag{4.8}$$

where r_{ai} is the length of point A_i and point A_{i+1} in point set A, B_i is the correspondence of A_i and B_{i+1} is the correspondence of A_{i+1}, r_{bi} is the length of B_i and point B_{i+1}. n is the number of correspondent pairs.

With the problem definition, the cross-source registration problem is to find the solution that maximizes objective function (4.4).

4.3.3 Iterative tensor optimization

The maximum of the objective function (4.4) is achieved by expectationmaximization (EM) algorithm, which is two optimizations conduct iteratively. They are the correspondence X optimization with the geometric transformation T firmed and the geometric transformation T optimization with the new correspondence X firmed. Algorithm 4.1 shows the whole process.

E-step: Optimization for the correspondence

With specific geometric transformation T given, the optimization of formulation 4.4 is a tensor optimization problem. According to (Shi, Ling, Xing & Hu 2013), two terms in objective function 4.4 are rank-1 tensors, so that the optimization can be formulated as $Rank-1$ *tensor approximation* (R1TA) problem. Inspired by (Shi et al. 2013, Duchenne et al. 2011), we use tensor power iteration to solve the above R1TA problems. This is efficient because this algorithm solve the problem by estimating the main eigenvector of the input matrix. Line 6-10 in Algorithm 4.1 shows the correspondence optimization procedures.

M-step: Optimization for the geometric transformation

With the correspondences X are given, the estimation of the geometric transformation T is similar to ICP which can be solved in close-form solution. Suppose A and B is n matched pairs from correspondence matrix X, A is the points from point set C_1 and B is the points from point set C_2. If U_A is the mean point of A and U_B is the mean point of B, $UDV = svd(AB^T)$, the geo-

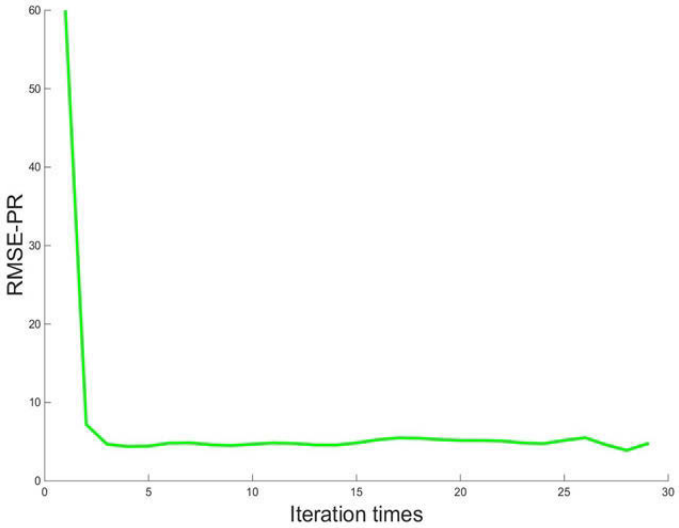

Figure 4.5: Iteration results.The RMSE during iteration.

metrical transformation T is computed by $R = UDV^T, t = U_A - s*R*U_B, s = (\sum_{i=1}^{n-1}(r_{ai}/r_{bi}))/n - 1$. In this step, the key elements are scale estimation and tensor update. For scale estimation computation, we use formulation 4.8 defined before. For tensor update, we use formulation 4.6, 4.5.

Based on the salient structures, the proposed method convergences quickly. Figure 4.6 shows the RMSE of transformation during the iteration, and the first iteration and final registration result. The efficient optimization in Algorithm 4.1 contains several key components: (1) salient structure extraction and tensors build-up (line 1,3,4); (2) update the salient structure similarity score (β_m) and pixel-wise refinement score (γ_m),listed in line 6-7; (3) a row and column $L_1 - norm$ normalization in line 10 makes the assignment matrix double-stochastic; (4) line 13-15 is the geometric transformation optimization. With R, t, s given, we update the point set C_2 using the current geometric transformation (line 16). Then, we updated the new point sets to construct new tensors and conduct iteratively.

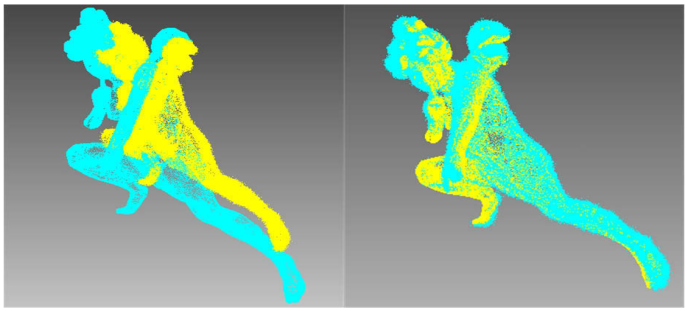

Figure 4.6: Iteration results. The first column is the registration result using transformation matrix at the first iteration, the second column is the final registration result.

4.4 Implementation details

Inspired by ICP variants (Bouaziz, Tagliasacchi & Pauly 2013, Chui & Rangarajan 2003) that considering some constraints, we propose a registration method to integrate salient structure and pixel-wise refinement. The proposed method likes ICP considering triplet constraint and we call it as geometric constraint tensor-based registration (GCTR).

Salient structure Extraction: According to our observation, the overall structure remains similar despite of local high variations in density, noise and outliers. Hence, we firstly segment the point clouds into many segments based on their geometrical topology. Following (Huang, Zhang, Fan, Wu & Yuan 2016), we firstly compute the descriptor of each 3D point and uses local clustering method to group points with similar descriptor around some seeds. The point cloud is then clustering into many segments and the central points of these segments are used as representation of salient structures. The above salient structure extraction handles the density variation and some extent of noise. Figure 4.7 shows the results of salient structure extraction, where the overall structure extraction results remain similar without density variance.

Triplet points selection: For cross-source point cloud registration, we care more about the global structure alignment accuracy instead of the details. Inspired by ICP with some constraints (Chui & Rangarajan 2003, Zhou

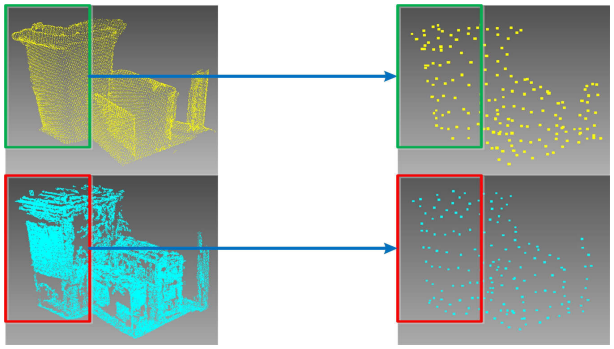

Figure 4.7: Results of salient structure extraction. The first column is the source point clouds of two different sensors where the above is mobile phone+VSFM and the bottom is KinectFusion; the second column is the extraction results. The red and green boxes show two examples of similar structures.

& De la Torre 2013*a*), we recognize a point-point match as a correct matcher only if the selected triplet points are matched. It likes ICP with triplet constraint. Therefore, the first step of the proposed algorithm is to select many triplets (detailed in the following subsection). We select triplets because the triplets are an elegant way to remain the salient structure that is invariant to rigid+scale transformation problem (Duchenne et al. 2011). Although the weak region affinity consider global information, the local information is still valuable for registration, we use pixel-wise refinement to keep the detail local information and refine the correspondence. Both elements are integrated into an unified tensor optimization process.

Algorithm details: Initially, we introduce line 1 of Algorithm 4.1 in detail. To robustly extract the structures, we follow the recently developed segmentation method (Papon et al. 2013) to segment the two point clouds into many super voxels and use the centers of super voxels to represent the structures (See Figure 4.7). (Papon et al. 2013) computes descriptor of each point in point cloud, then gives many seeds in the point cloud and clusters around these seeds by comparing their descriptors' similarity between points to seed. If their similarity below a threshold, they belong to a same segment.

Algorithm 4.1 Iterative tensor optimization

Require: Cross-source point clouds C_1, C_2

Ensure: Transformation matrices R,t,s Salient structures extraction. (in Sec. 4.4)

while condition1 **do**
  Random triplets selection in C_1, C_2. (Sec 4.4)
  Compute triplet point and pixel-wise descriptor f_{ijk}, f_i in one point cloud. (Sec. 4.3.1)
  Initialize or update transformation matrix T
  Compute triplet point and pixel descriptor f_{ijk}^T, f_i^T in another point cloud after transformation. (Section 4.3)
  $H_{ii'jj'kk'}(T) = exp(-\|f_{ijk} - f_{i'j'k'}^T\|^2)$
  $\bar{H}_{ii'}(T) = exp(-\|f_i - f_{i'}^T)\|^2)$
  Rewritten similarity tensor $H_{ii'jj'kk'}$ and $\bar{H}_{ii'}$ as third-order tensor $H_{\alpha\beta\gamma}$ and first-order tensor H_l.
  E-step: correspondence matrix X estimation
  while condition2 **do**
    $X^{(m+1)} = H_{\alpha\beta\gamma} \otimes X^{(m)} \otimes X^{(m)} + H_l$
    $X^{(m+1)} \leftarrow \frac{1}{\|X^{(m+1)}\|_2} X^{(m+1)}$
  end while
  M-step: transformation matrix T estimations
  while condition3 **do**
    get the correspondent pair: $A_1, B_1 \leftarrow \|X^{(m+1)}\|_1^r$
    remove the outliers $A, B \leftarrow RANSAC(A_1, B_1)$
    calculate the rotation: $R = UDV^T$
    calculate the translation $t = U_A - s * R * U_B$
    calculate the scale $s = \frac{\sum_{i=1}^{n-1}(r_{ai}/r_{bi})}{n-1}$
    Update point cloud $C_2 = (sRC_2 + t)$
  end while
end while

To be efficient, the method define a segment searching radius to constraint the segment belongs to a local continuous region. We define the segmented radius and seed interval as the 1% of the diameter of the point cloud.

Then, we introduce line 3-4 of Algorithm 4.1 in detail. Inspired by (Mellado et al. 2014), we select triplets satisfying wide baseline strategy, so that they are more likely to be global aligned. In this algorithm, we use wide baseline strategy to randomly select N_1N_2 large triangles in point cloud 1. We define the large triangles as three edges of the triangle are large than 50% of the overlapping 3D containing voxel's the diameter. That guarantees the selected triangles are large triangles and make the final registration more prone to globally registered. For the overlapping ratio, if there is unknown, we automatically search as ratio is 0.25, 0.5, 0.75, 1.0.

4.5 Experimental results

In this section, we conducted extensive experiments on synthetic and real datasets.

- **Synthetic cross-source point cloud datasets:** we provide a synthetic cross-source benchmark datasets. Several state-of-the-art registration methods have been run on it and compared with the proposed method.

- **Real cross-source point cloud datasets:** to demonstrate the proposed method in deal with real problem, we compare on real cross-source dataset where one sensor is KinectFusion and the other is mobile RGB camera.

Based on the above datasets, we conduct comprehensive evaluation for the proposed algorithm. Two kinds of evaluations are conducted: contribution evaluation and pipeline evaluation.

- **Contribution evaluation** is to compare the proposed GCTR with the existing related registration methods. Some other tensor combination

methods are also evaluated. It aims to demonstrate the accuracy and efficiency in dealing with cross-source point cloud registration. All the compared methods are used the same input poses.

- **Pipeline evaluation** is to compare the proposed GCTR with Super4PCS linking ICP refinement. We conduct this evaluation because our iterative tensor optimization of GCTR can obtain accurate local solution based on the interaction between internal global triplet alignment and local pixel-wise refinement. It is similar to firstly obtain global registration and then obtain local registration based on the global registration result. But we integrate these two processes into a unique optimization procedure and obtain interaction between them.

4.5.1 Experiments setup

The proposed algorithm is implemented by using standard C and Matlab. All the comparison experiments are executed on an I5 CPU, 8GB memory computer. Because the proposed method (GCTR) likes ICP WITH triplet constraint, we select ICP (Best & McKay 1992) and recent GO-ICP (Yang et al. 2013), Super-4PCS(Mellado et al. 2014), CPD(Myronenko & Song 2010), JR-MPC(Evangelidis & Horaud 2018) and CSGM(Huang, Zhang, Fan, Wu & Yuan 2016) as the comparison methods. To thoughtfully evaluate the framework in dealing with 3D cross-source point cloud registration problem, we evaluate several variations from the proposed method, (1) U_{23} represents the combination of 2rd and 3rd order tensor, (2) U_{123} represents the combination of 1st, 2rd and 3rd order tensor, (3) U_{13} represents the combination of 1st and 3rd order tensor of the framework. To overcome the density variation, we follow CSGM to extract the structure points. To solve the missing data, we select large triangles (each edge is large than the radius of the point cloud) for our third-order constraints. Because existing methods have not designed for scale variation, to compare fairly, we conduct a scale normalization by following CSGM. Also, because JR-MPC becomes

not practical when the point number increases significantly, we segment the point cloud to approximately 2000 points for JR-MPC.

For the evaluation metric, we compute the Frobenius Norm (F-norm) of the transformation matrix (TM) error between ground-truth transformation matrix and estimated transformation matrix. In all the point cloud experiments, we compute the log of transformation matrix error $log(TM)$. The lower the value is, the higher accuracy the method achieves.

4.5.2 Synthetic cross-source benchmark dataset

For the construction of benchmark dataset, we start from Stanford 3D Scanning Models [1] and simulate ten sets of cross-source benchmark dataset. Each set of cross-source dataset contains source A and source B which simulate cross-source problems (discussed in Section 4.1). The cross-source simulation parameters are listed in Table 4.1. The simulated V1 and V2 are pairs of cross-source point clouds.

There are four steps to simulate the hetegeneous point cloud: Step 1,a modified KinectFusion method is used to output the image sequence and camera pose of each image when capturing KinectFusion point clouds. Step 2, another point cloud is computed using these images and VSFM. A set of camera poses is computed using VSFM. As these two cross-source point clouds come from the same set of image sequences, the camera poses of KinectFusion and VSFM should be the same. Figure 6 illustrates a general schematic diagram in which cross-source point clouds are reconstructed using VSFM and KinectFusion respectively. Ground truth registration between these point clouds are established via following procedures. The VSFM point cloud is back-projected into the image coordinate system and then reprojected into the KinectFusion coordinate system. To avoid the inaccuracy of camera pose computation in VSFM and KinectFusion, we consider many poses whose reprojection error is less than θ (θ=0.5), and use these camera pose center points and the least-squares method to compute the final rigid

[1] http://graphics.stanford.edu/data/3Dscanrep/

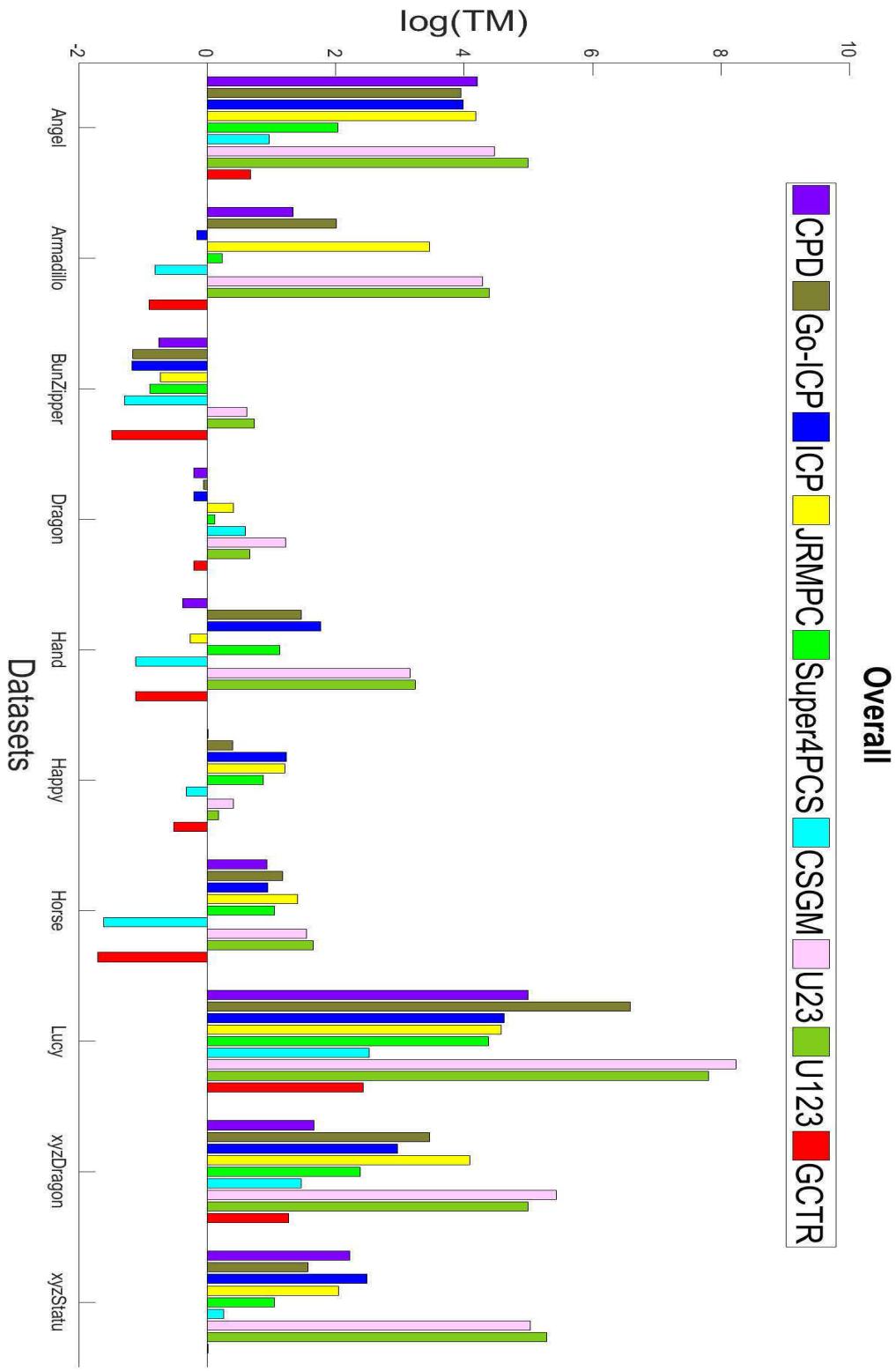

Figure 4.8: Overall performance (rotation,transformation and scale) on synthetic cross-source benchmark dataset.

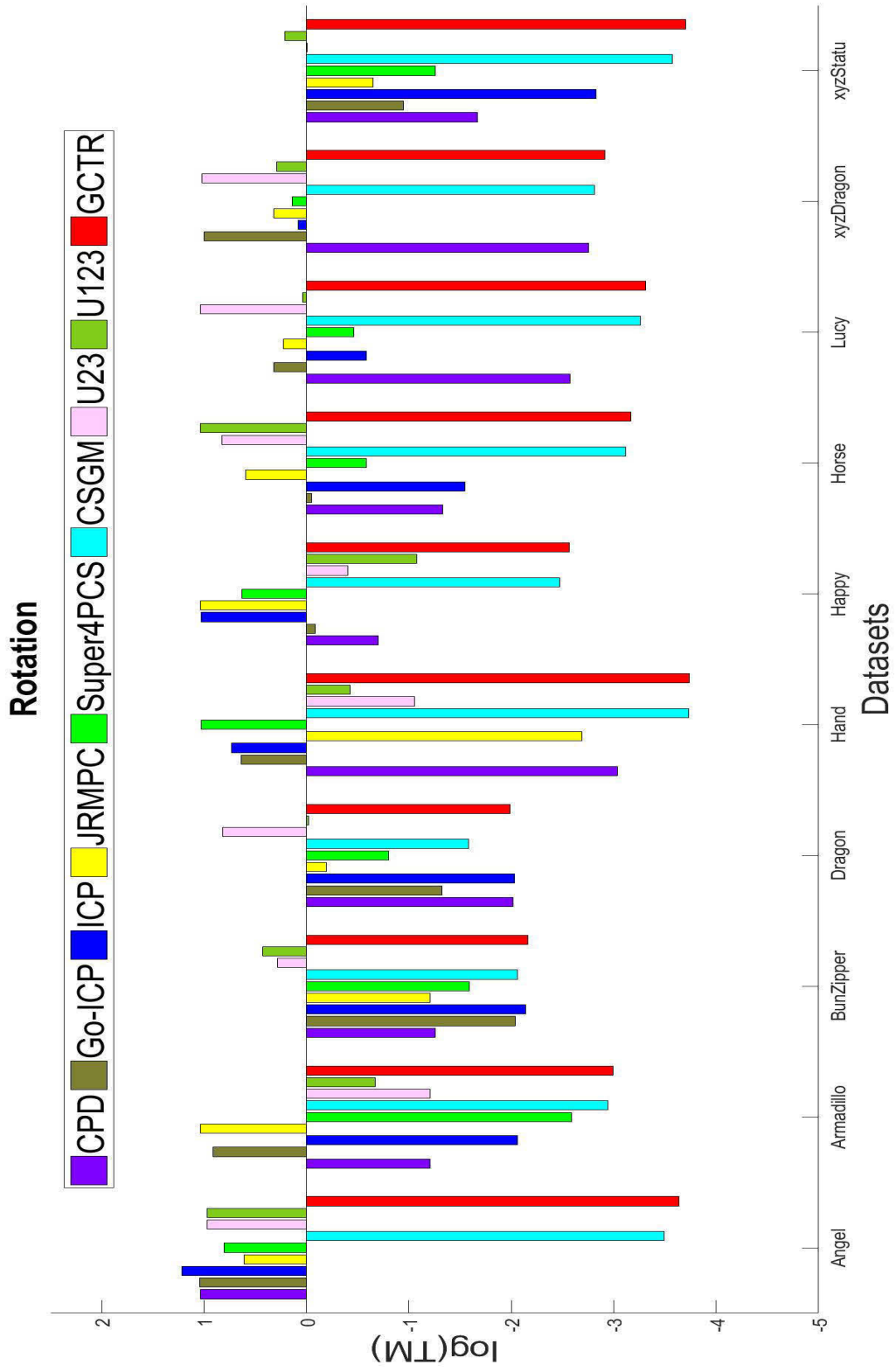

Figure 4.9: Rotation performance on synthetic cross-source benchmark dataset.

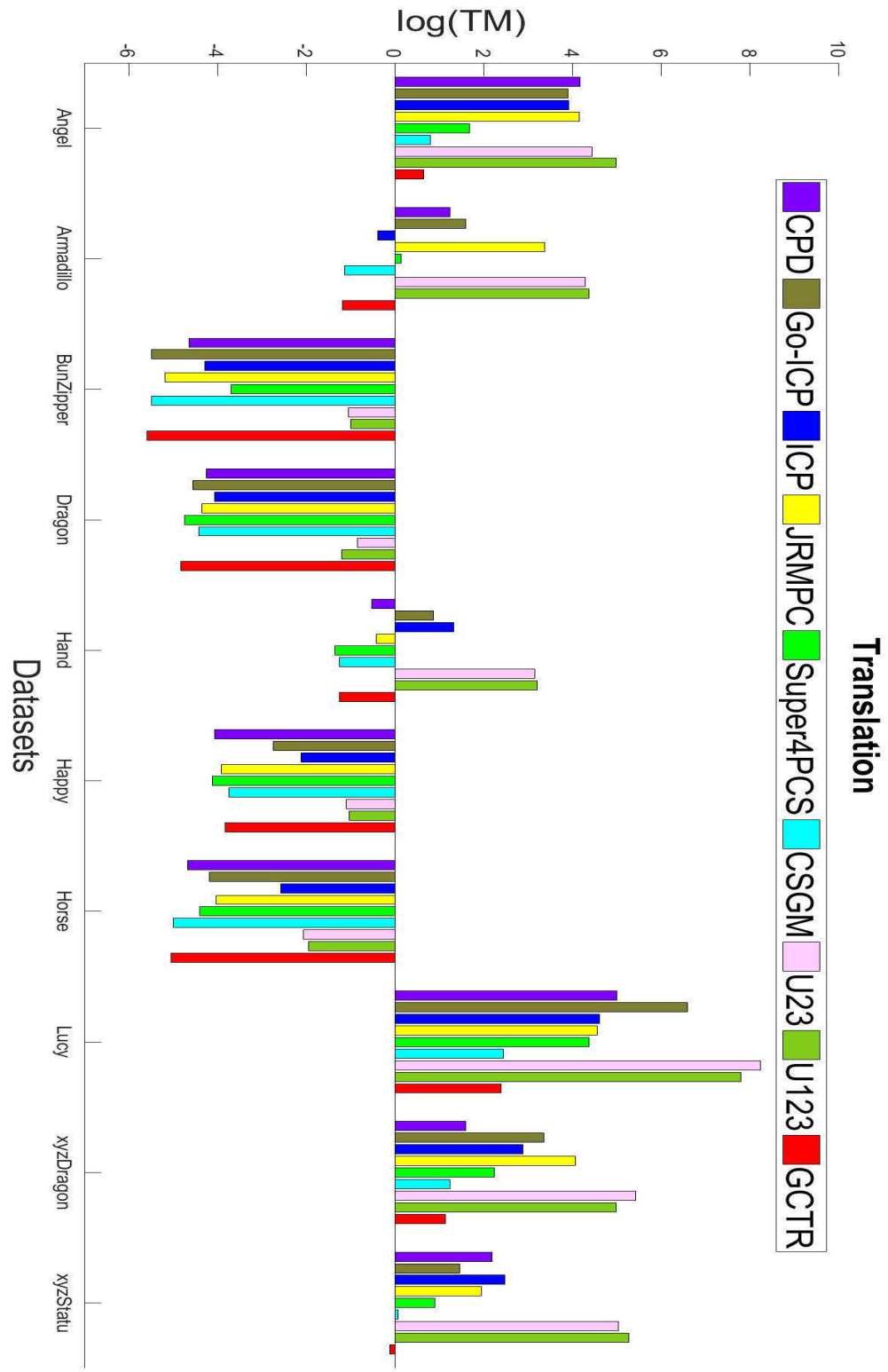

Figure 4.10: Translation performance on synthetic cross-source benchmark dataset.

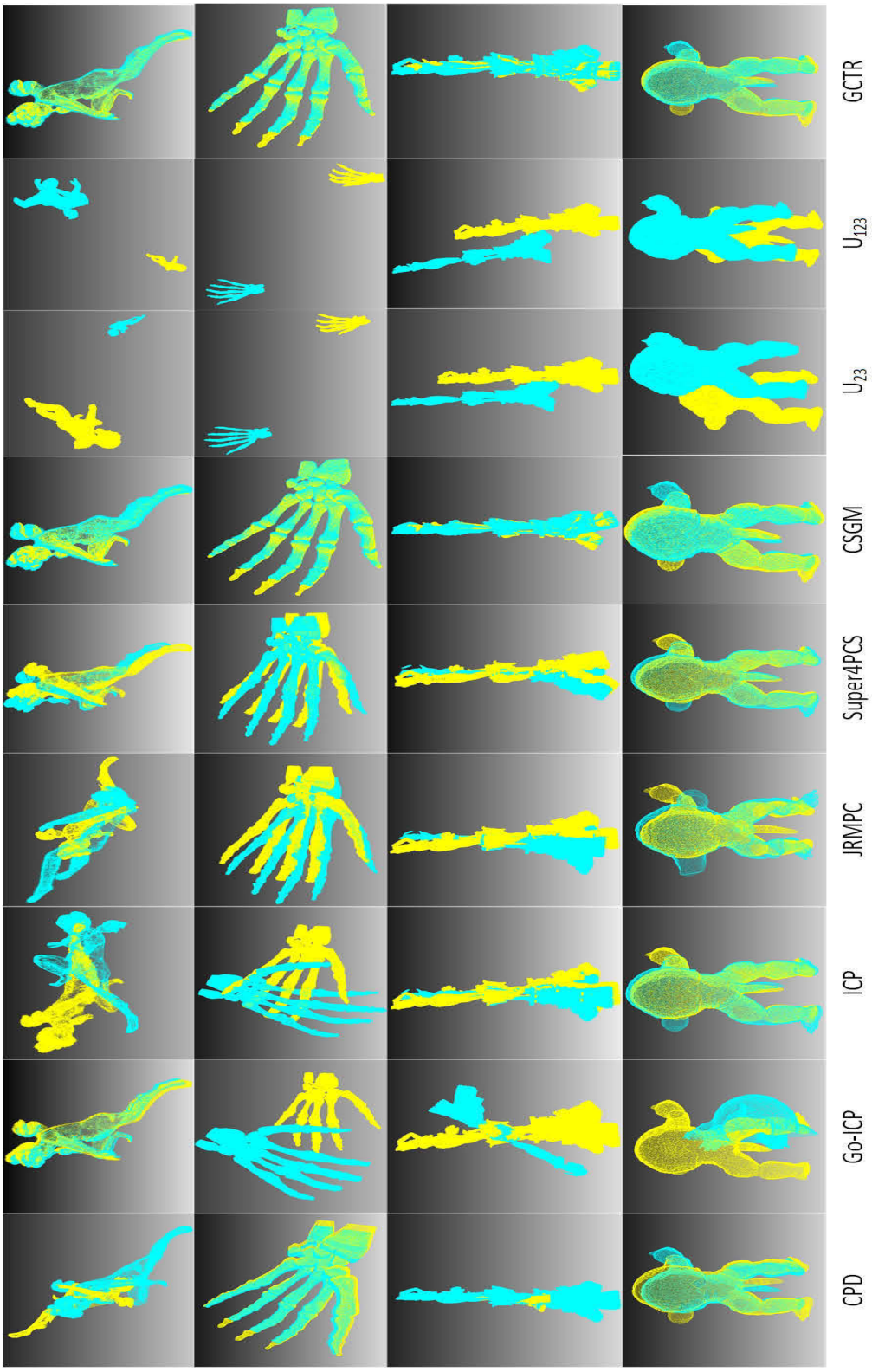

Figure 4.11: Visual registration results of benchmark synthetic cross-source point clouds.

Steps	Descriptions
1	Rotate on y axis and keep $z \leq 0$ points to obtain V1,V2 (different coverage)
2	Random delete some parts on V2
3	Add scale:2-4,rotation:30°-210°,translation:0-10% of D_{max} on V2
4	Add 40DB noise and 50% outliers on V2

Table 4.1: The benchmark dataset construction.

Method	R	T	S	log(TM)	time(s)
CPD	2.82	64.4	0.24	4.21	435
Go-ICP	2.7	48.8	0.06	3.94	73.8
ICP	3.38	50.1	0.06	3.98	103
JR-MPC	1.84	63.3	0.06	4.18	255
Super4PCS	3.38	50.0	0.06	2.03	333
CSGM	*0.104*	*2.1*	0.06	*0.97*	*3024*
U_{23}	2.63	84.5	0.06	4.48	63.2
U_{123}	2.64	114	0.06	4.99	71.8
GCTR	*0.026*	*1.89*	**0.03**	*0.67*	**61.6**

Table 4.2: Comparison on the Angle benchmark dataset.

transformation between these two camera center points. The rigid transformation matrix is built on critical prior information and can therefore be used as ground-truth. These benchmark data contain 13 datasets and can be used to perform quantitative evaluation for cross-source point cloud registration.

We select *Angle* as example presentation for detailed quantitative evaluation on the cross-source benchmark datasets. We evaluate the *Translation(T), Rotation(R), Scale (S)* error separately, and compare the whole error *log(TM)*. Also, we compare the runtime on these datasets. Table 4.2 shows the evaluation results. Compared to other methods, the proposed methods obtain higher accuracy and efficiency. CSGM shows comparable accuracy to the proposed method while the efficiency is lower. Go-ICP ob-

Method	Runtime(s)
CPD	650
Go-ICP	363
ICP	1491
JR-MPC	324
Super4PCS	1941
CSGM	4648
U_{23}	294
U_{123}	307
GCTR	**287**

Table 4.3: Average running time on the 10 pairs of cross-source benchmark datasets.

tains comparable efficiency to our methods while the accuracy is much lower than our method U_{13}. The reason of variations U_{123} and U_{12} obtain low accuracy is that the large noise and scale variation make the second order constraint (edges) not robust. In our experiments, it plays negative impact to final experiments.

Figure 4.8 shows the comparison results on whole datasets. We can see that our method obtains comparable accuracy to CSGM which outperforming all the other methods and obtains very robust results in all datasets. In the other comparison methods, Super4PCS achieves the second performance for most datasets. To detailed compare the performance, we also show the performance of rotation and translation on synthetic benchmark dataset, which is very important to evaluate the performance of registration algorithms. Figure 4.9 shows the proposed method obtains higher accuracy on rotation performance. The accuracy of CSGM follows the proposed method and the performance of others differs at different datasets. Figure 4.10 also shows the proposed method obtains higher accuracy on translation performance.

Figure 4.11 visually shows the registration results. From the visual view-

point, the proposed methods achieve better results than other methods. From visual viewpoint, we obtain better results than CSGM in the fist row and fourth row datasets.

To compare the efficiency, we compute the average runtime on the 10 cross-source datasets. Table 4.3 show the proposed method is much faster than other compared methods. Although CSGM obtains similar accuracy to the proposed method, the proposed method achieves much higher efficiency than CSGM (more than 16 times faster).

Method	R	T	S	log(TM)
Super4PCS	3.38	50.0	0.06	2.03
Super4PCS+ICP	2.20	6.06	0.06	0.92
GCTR	**0.026**	**1.89**	**0.03**	**0.67**

Table 4.4: The pipeline evaluation on Angle benchmark dataset.

Pipeline evaluation: We compare with Super4PCS + ICP with the proposed method on Angle benchmark dataset. Table 4.4 shows ICP can improve the accuracy of Super4PCS. However, the accuracy is lower than that of the proposed method. The reason is that the proposed method utilizes interaction between global and local, between transformation matrix estimation and correspondence estimation. These interactions are integrated into tensor optimization. This experiment also shows that interactions between local and global components, between transformation matrix and correspondence estimation steps are helpful for solving registration problem.

4.5.3 Real cross-source point clouds

We capture the cross-source point cloud dataset by using KinectFusion (Newcombe et al. 2011) and VSFM (Furukawa et al. 2010). KinectFusion outputs point cloud directly while VSFM reconstructs 3D point cloud from the images taking by mobile phone camera. We captured more than 30 datasets for different

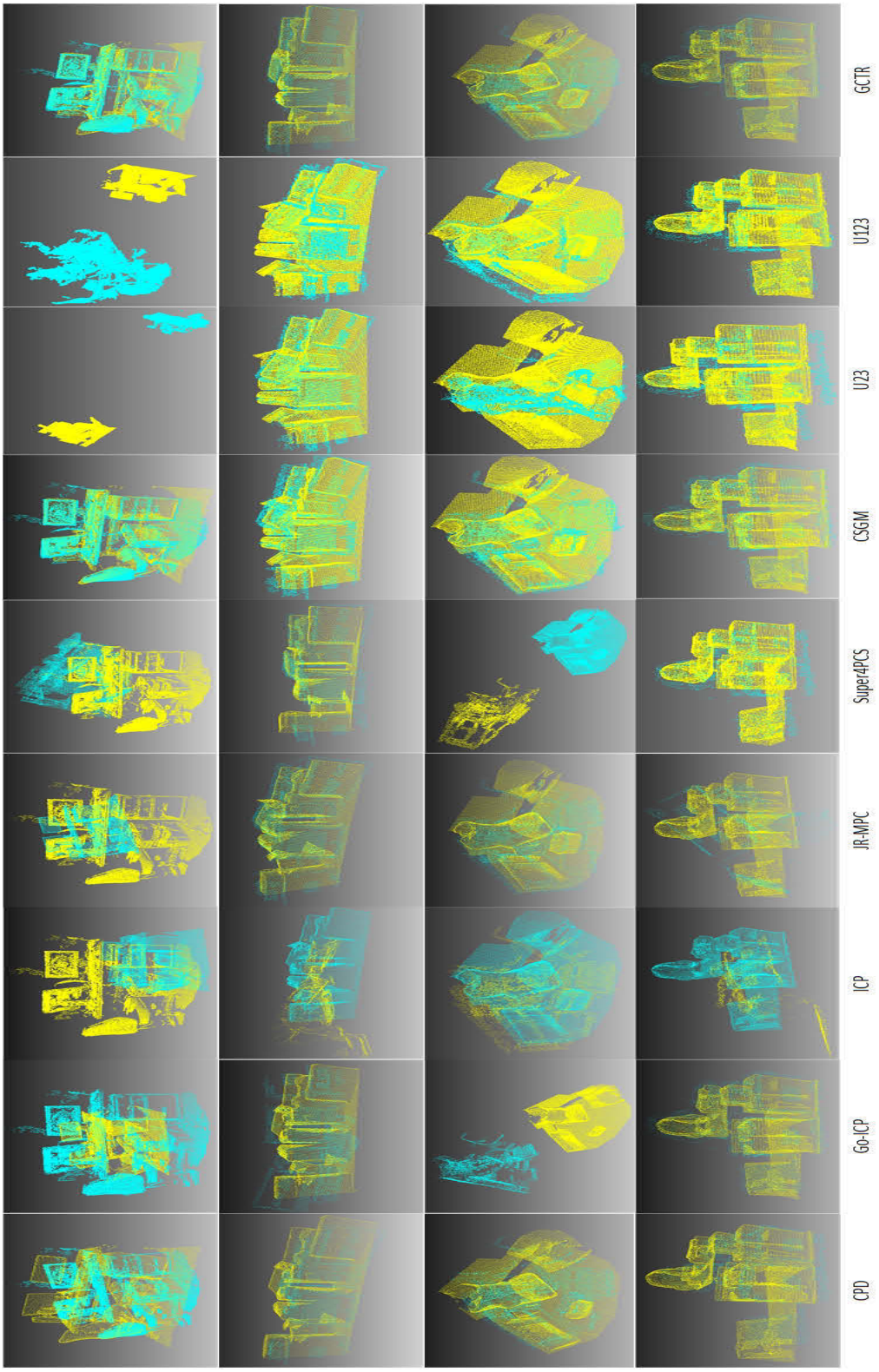

Figure 4.12: Visual registration results of real cross-source point clouds.

indoor scenes and the proposed methods obtain promising results in all the datasets.

Figure 4.12 shows the selected datasets and compares with other methods. The results illustrate our method obtain robust and visually correct registration results in real application. For all these datasets, the proposed methods align at a fast speed in approximately 120 seconds which is much faster than other methods. Although sometimes JR-MPC and Go-ICP obtains registration results similar to our methods, their computation and memory complexity are very high. Our method can align the real cross-source point cloud accurately at a fast speed.

4.6 Conclusion

In this chapter, we propose an fast tensor-based framework for cross-source point cloud registration. We have done two main works: firstly, weak regional affinity and pixel-wise refinement are proposed to keep global and local information in cross-source point clouds where structures are usually weak; secondly, an unified algorithm is proposed to integrate these two components to solve the cross-source point cloud registration. The experimental results show that the proposed method aligns the challenging cross-source point cloud fast and accurately.

Chapter 5

Cross-source point cloud registration by using Gaussian mixture models

5.1 Introduction

Gaussian mixture models (GMM) are widely utilized models in solving registration problem. It shows great ability of handling noise, outlier and density variants in registration. We discourage that the learned GMM can also be described the global structure information. Although structure-based methods in the Chapter 3 and Chapter 4 show great ability in solving registration problem, they still need to do pre-processing steps (e.g. segmentation) to solve the density and some of the noisy problem. To solve the limitation, we explore how to use statistic model to directly describe the structure information without segmentation steps. More specifically, a Gaussian mixture model has been used to describe the 3D scene. Two cross-source point clouds of the 3D scenes are recognized as two scaled samples from the Gaussian mixture models. The two cross-source point clouds from different sensors are similar to using different sampling strategies to obtain two samples from a same distribution. In this chapter, an algorithm will be introduced to ex-

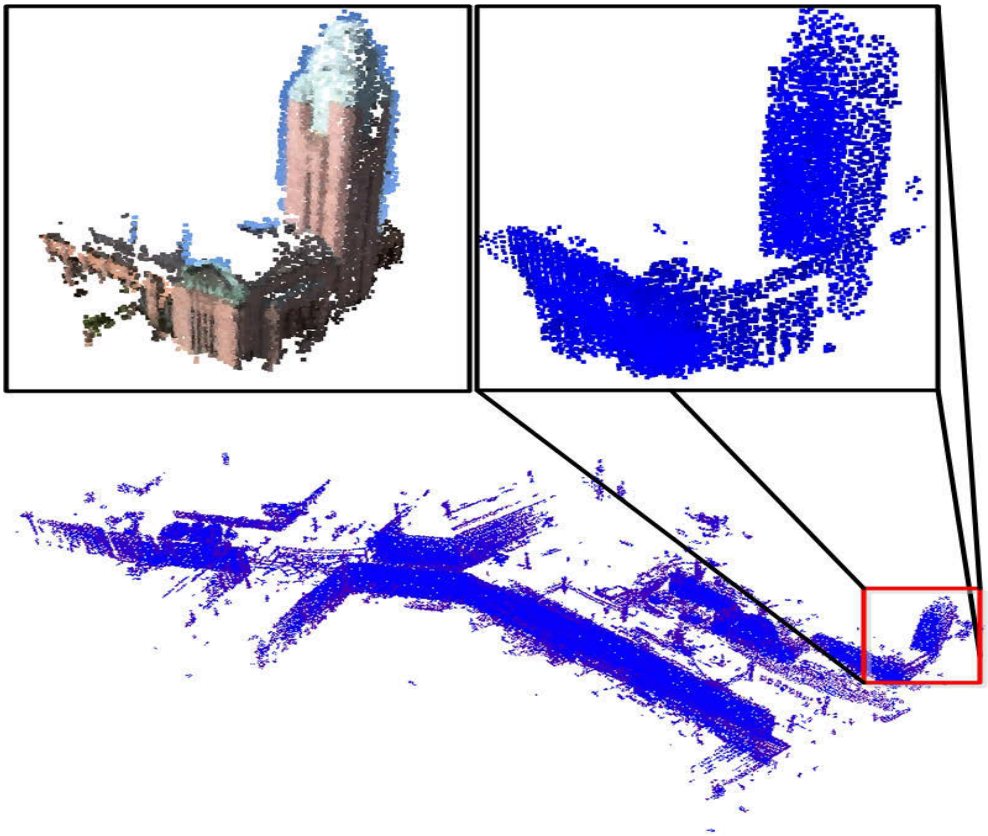

Figure 5.1: An example of cross-source point clouds of SFM and LiDAR highlighted from the street view scene. The top left is the SFM point cloud and the top right is the detected registration result on LiDAR point cloud.

plore structure information by using statistic model. To demonstrate our algorithm on solving cross-source point cloud registration problem, we focus on a general case that aligning a small point cloud from one source to another large point cloud from another source. Figure 5.1 shows an example. It is a match and registration problem.

In this chapter, a novel coarse-to-fine algorithm is proposed to match and register two cross-source point clouds (one is whole candidates, the other is target which is a small part). There are mainly two stages: 1) at the coarse stage, top K potential regions are detected by a coarse matching in the large-scale candidate point clouds or database for a small-scale point

cloud. 2) at the fine stage, two cross-source point clouds are assumed as scaled samples from a same virtual GMM model [1]. We register them based on this assumption and use the registration error to refine the ranking in the first stage.

The main contributions can be summarized as follows: (1) an effective coarse-to-fine pipeline is proposed with embedded solution for scale variation in different sources. The scale variation is a common problem in cross-source point cloud registration. The key aspects of our pipeline include: on one hand, Top K potential regions are coarsely selected by us ing efficient ESF descriptor, and on the other hand, a generative GMM-based method is proposed to refine the coarse selected regions, which takes into account the impact of scale variation in two cross-source point clouds. (2) To deal with scale problem, two scaled samples of point clouds from different sources are used for generating the GMM. To smoothly drift two point clouds into registration, the rigid transformation and generated GMM are estimated along its convergence. Different from [9] and other previous generative GMM-based method, we reformulate the registration with the cross-source scales into a generative GMM cost function.

5.2 Coarse-to-fine Algorithm

The algorithm is illustrated in Figure 5.2. It contains coarse matching and fine registration, among which coarse matching aims at finding the top K potential regions in candidate point cloud that potentially match with target point cloud. It substantially reduces the number of candidate regions and hence saves computation cost of the next stage. We compute ESF descriptors of these potential regions and use them to conduct the first coarse matching. Then, a generative GMM-based registration is performed to obtain the transformation of two cross-source point clouds and use the transformation

[1]We assume a complexity scene can be described by a GMM model, and there is a virtual GMM model to describe the complexity scene. We will estimate it later.

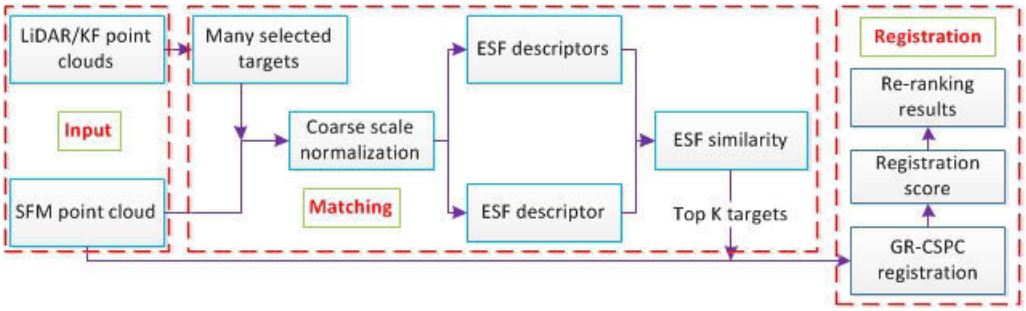

Figure 5.2: Overview of the proposed coarse-to-fine algorithm. With Lidar/KinectFusion(KF) and SFM cross-source point clouds input, matching stage aims at detecting most potential registration targets; registration stage aims at finding optimal registration relation and refining the previous matching results.

error to refine the matching results. Furthermore, main steps of the second stage are (1) obtaining transformation matrix of each registration; (2) acquiring residual error of each registration by applying transformation matrix to the original two cross-source point clouds (e.g. selected LiDAR region and SFM); (3) using residual error to re-rank the matching results and output the ranked registration results. After registration, accurate transformation matrix is obtained, and they can be used for applications such as location based service.

5.2.1 Coarse Matching

In the coarse matching stage, Top K potential targets are obtained for the SFM point cloud from LiDAR/KinectFusion(KF) point clouds. In this stage, ESF descriptors are computed for two selected point clouds (SFM point cloud and LiDAR/KF targets) and used for coarse matching. Due to ESF alone is very hard to find the correct result in the cross-source problems, to detect the most reliable result, the fine registration step is indispensable.

5.2.2 Fine Registration

In this chapter, we propose a generative Gaussian mixture model (GMM) for the cross-source point cloud registration (GR-CSPC). We consider two cross-source point clouds are scaled samples from a virtual GMM and the GMM is generated from these two point clouds. If they are coming from same GMM, the registration error will be very small. We select GMM because it is a robust model to describe the complexity scene(Bishop 2006). The GMM is robust to density, missing data, noise and outliers. The model is simple while very robust to describe a complexity scene in real world.

In the previous GMM-based registration methods (Jian & Vemuri 2011*a*, Myronenko & Song 2010, Ma, Zhao & Yuille 2016), they estimate one GMM using one point clouds or two GMMs using two point clouds. This makes the reasonable assumption that points from one set are normally distributed around points belonging to the other set. Hence, the point-to-point assignment problem can be recast into that of estimating the parameters of a mixture distribution or minimizing the GMMs distance. In general, these probabilistic methods are able to exploit global relationships in the point sets, since the rough structure of a point set is typically preserved; otherwise, even people cannot find correspondences reliably under arbitrarily large deformations. Except the overall pipeline, as an another contribution of this Chapter, we assume there is one virtual GMM to describe the scene or object, and the two cross-source point clouds are different samples with scale variant from different sensors (Figure 5.3). We need to address the how to estimate the only one GMM and scale, rotation and translation between two cross-source point clouds.

More precisely, we formulate the registration of cross-source point clouds into a model generative problem, where two point clouds are different samples of a GMM and they have scale, rotation and translation transformation. We propose a generative Gaussian mixture model method to solve it. At the optimum, two cross-source point clouds become registered and the rigid transformation is obtained using the maximum of the GMM posterior proba-

bility for the two cross-source point clouds. Core to our registration method is the generative concept considering two point clouds as different samples for a virtual object/scene (describing as GMM), and optimize the GMM parameters and transformation matrix simultaneously.

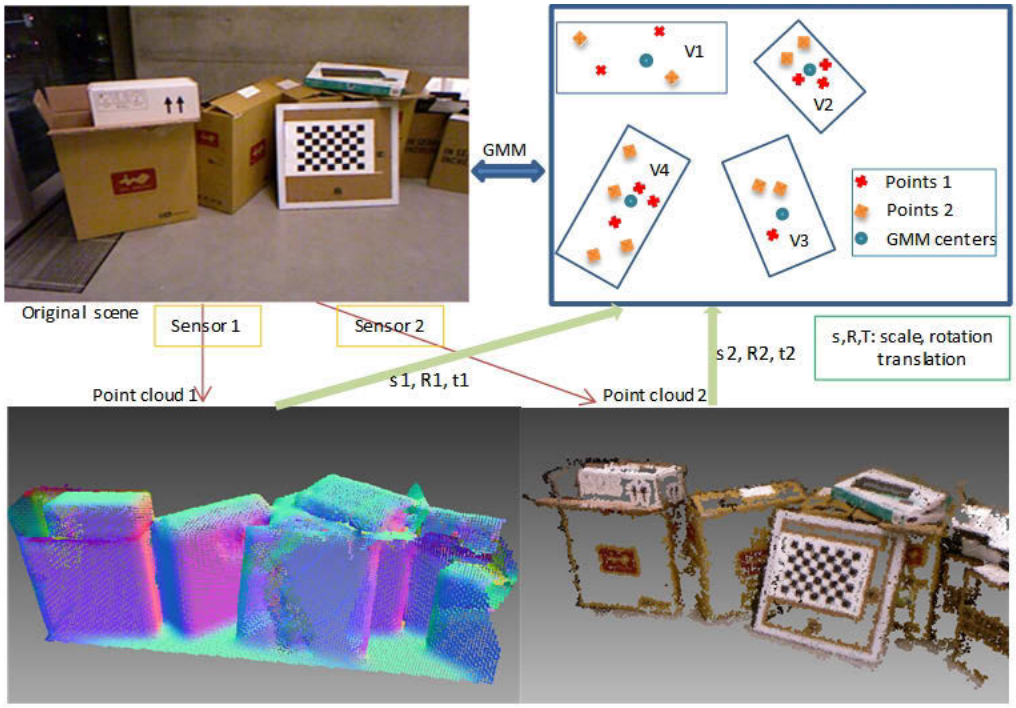

Figure 5.3: The proposed generative model for cross-source point cloud registration (GM-CSPC). The proposed algorithm simultaneously estimates both the GMM and transformation parameters (scale, rotation and translation).

For cross-source point cloud registration, the conventional point-to-point level methods face much difficulty in registering these large variable cross-source point clouds. This is because of their simple least square mean error of point-level correspondence can be easily lead to sub-optimal in the large variant cross-source point clouds. Also, these methods are susceptible to occlusion and partial overlap. In order to address the limitation issues of previous direct methods in terms of cross-source problems, we estimate the successive model of two cross-source point clouds described for. Considering

the missing data variations, noise and outliers, and different density of cross-source problem, GMM is selected to consider global statistical properties (e.g. global shape or distribution). GMM-based method focuses on whether the two cross-source point clouds are globally registered and ignores the large variation in local structure.

The two cross-source point clouds are represented as Y_1 and Y_2, which is $N_1 \times 3$ and $N_2 \times 3$ matrix respectively, N_1 and N_2 are 3D point number of Y_1 and Y_2. According to (Myronenko & Song 2010, Bishop 2006), considering all GMM components as equal membership, the mixture model considering the noise and outliers can be written as:

$$p(X_{ji}) = (1-w)\sum_{k=1}^{K}\frac{1}{K}p(T_j(X_{ji})|u_k,\sigma_k) + w\frac{1}{h} \tag{5.1}$$

where T_j is rigid transformation model, $T_j(X_{ji}) = s_j R_j X_{ji} + t_j$, s_j is a scale factor, R_j is a 3×3 rotation matrix, t_j is 3×1 translation matrix; u_k and σ_k are mean and variance parameters of k^{th} Gaussian model; w is the weight of noise and outliers, $0 \leq w \leq 1$; h is the volume of the 3D convex hull encompassing the data (Horaud, Forbes, Yguel, Dewaele & Zhang 2011); K is the number of Gaussian models.

The parameters can be estimated by using the framework of expectation-maximization. We define a corresponding latent variable $Z = z_{ji}$, where $z_{ji} = k$ means $T_j(X_{ji})$ is assigned to the k-th component of the GMM. The complete data set is $\{X, Z\}$. In order to compute the parameters of GMM, we need to maximize the complete-data log likelihood. However, the complete data is usually not given and only incomplete data X can be utilized. According to (Bishop 2006), the complete-data log likelihood can be computed by E and M steps. In the E step, we estimate the posterior probability of the latent variables given by $P(Z|X, u_k, \sigma_k, T_j)$; in the M step, we use this posterior probability to find the maximization of the expectation

of the complete-data likelihood function, which is

$$Q(\theta) = \sum_{Z} p(Z|X,\theta) log(p(X,Z|\theta)) \tag{5.2}$$

where θ represents the parameters containing u_k, σ_k, $k = 1...K$ and T_j, $j = 1, 2$.

According to (Bishop 2006), ignoring the constants independent of θ, (5.2) can be rewritten as

$$\begin{aligned} Q(\theta) = &- \frac{1}{2} \sum_{j=1}^{2} \sum_{i=1}^{N_j} \sum_{k=1}^{K} \alpha_{jik} (\|T_j(X_{ji}) - u_k\|^2 \\ &+ log|\sigma_k|), \ s.t. \ R^T R = I, det(R) = 1. \end{aligned} \tag{5.3}$$

where α_{ijk} is the posterior probability which can be computed by the previous parameter values:

$$\alpha_{jik} = \frac{p_k \sigma_k^{-3} exp(-\frac{1}{2\sigma_k^2} \parallel T_j(X_{ji}) - u_k \parallel^2)}{\sum_{s=1}^{K} [p_s \sigma_s^{-3} exp(-\frac{1}{2\sigma_s^2} \parallel T_j(X_{ji}) - u_s \parallel^2)] + \beta} \tag{5.4}$$

where $w = w/h(w+1)$ accounts for the outlier term and $\alpha_{ji(K+1)} = 1 - \sum_{k=1}^{K} \alpha_{jik}$ accounts for the posterior could be an outlier.

E-step. Computing posterior probability is the E-step. In this step, previous θ value and equation (5.4) are used to compute the posterior probability α_{jik}. Note that, the computation of posterior probability in i^{th} step need the parameters of $(i-1)^{th}$ step.

M-step. With the posterior probability known, maximization(M) step aims at estimating the parameter of θ by maximizing the objective function $Q(\theta)$. Due to T_j associate with each point cloud are shared with a same GMM parameters, they can be estimated independently. By setting current GMM parameters, the estimation of T_j can by reformulated as the following constraint problem (Myronenko & Song 2010)

$$\begin{cases} \min\limits_{s_j, R_j, t_j} \|(s_j R_j V_j + t_j - X)\Lambda_j\|_F^2 \\ s.t. \qquad R_j^T R_j = I, \ |R_j| = 1 \end{cases} \tag{5.5}$$

where X is the weighted value of whole means of GMM components. $\|\cdot\|_F$ is the Frobenois norm; V_j is the virtual 3D points related to given points which is given by

$$V_{jk} = \frac{\sum_{i=1}^{N_j} \alpha_{jik} X_{ji}}{\sum_{i=1}^{N_j} \alpha_{jik}} \tag{5.6}$$

In order to solve formulation 5.5, we introduce the following Theorem 1. It has a close-form solution.

Theorem 1. *Let A and B be two $m \times n$ point clouds, m is the points' dimension, and UDV is the singular value decomposition of $\bar{A}\Lambda\Lambda^T\bar{B}^T$ ($\bar{A} = A - \frac{A\Lambda^2}{tr(\Lambda^2)}, \bar{B} = A - \frac{B\Lambda^2}{tr(\Lambda^2)}$),$\Lambda$ is a weight matrix. The minimum value of ξ of the* ***weighted mean squared error***

$$\xi(s, R, t) = \|(sRA + t - B)\Lambda\|_F^2 \tag{5.7}$$

of two point clouds with respect to their transformation matrices (s: scale factor, R: rotation and t:translation matrix) are given as

$$R = USV^T \tag{5.8}$$

$$t = -\frac{1}{tr(\Lambda)^2}(sRA - B)\Lambda^2 \tag{5.9}$$

$$s = \frac{tr((\bar{A}\Lambda\Lambda^T\bar{B}^T)^T R)}{tr\{(R\bar{A})^T(R\bar{A})\}} \tag{5.10}$$

where $S = diag(1, 1, det(UV^T))$.

Proof: Equation (5.7) can be rewritten as

$$\begin{aligned} \xi(s, R, t) &= ((sRA + t - B)\Lambda)^T((sRA + t - B)\Lambda) \\ &= \{((sRA - B)\Lambda)^T((sRA - B)\Lambda)\} \\ &\quad + 2\{((sRA - B)\Lambda)^T t\Lambda\} + \{(t\Lambda)^T(t\Lambda)\} \end{aligned} \tag{5.11}$$

Taking the partial derivative of $\xi(s, R, t)$ with respect to t, we obtain:

$$\frac{\partial\xi(s, R, t)}{\partial t} = ((sRA - B)\Lambda)^T\Lambda + \Lambda^T\Lambda t$$

Setting $\frac{\partial \xi(s,R,t)}{\partial t} = 0$, we obtain:

$$t = -\frac{1}{tr(\Lambda^2)}(sRA - B)\Lambda^2) \tag{5.12}$$

Substituting (5.12) back into (5.7) and represent $\bar{A} = A - \frac{A\Lambda^2}{tr(\Lambda^2)}$, $\bar{B} = B - \frac{B\Lambda^2}{tr(\Lambda^2)}$, $R^T R = I$, we obtain:

$$\xi(s, R, t) = -2s * tr((\bar{A}\Lambda\Lambda^T\bar{B}^T)^T R) \tag{5.13}$$

In order to proof rotation matrix R in equation (5.8), we need to introduce a Lemma (Myronenko & Song 2009).

Lemma 1. *Let $R_{D\times D}$ be an unknown rotation matrix and $A_{D\times D}$ be a known real square matrix. Let USV^T be a Singular Value Decomposition of A, where $UU^T = VV^T = I$ and $S = d(S_i)$, with $s_1 \geq s_2 \geq, ..., \geq s_D \geq 0$. Then, the optimal rotation matrix R that maximizes $tr(A^T R)$ is $R = UCV^T$, where $C = d(1, 1, ..., det(UV^T))$.*

Using this Lemma and equation (5.13), we can conclude that

$$R = USV^T \tag{5.14}$$

where, U and V are matrices from the singular value decomposition of matrix $\bar{A}\Lambda\Lambda^T\bar{B}^T$, and $S = diag(1, 1, det(UV^T))$.

To proof scale equation in (5.10), we need to rewrite (10) and take partial derivative of $\xi(s, R, t)$ with respect to s,

$$\begin{aligned}\frac{\partial \xi(s, R, t)}{\partial s} =& 2s * tr(R\bar{A}^T(R\bar{A})^T) \\ & - 2tr((\bar{A}\Lambda\Lambda^T\bar{B}^T)^T R)\end{aligned}$$

setting $\frac{\partial \xi(s,R,t)}{\partial s} = 0$, we obtain

$$s = \frac{tr((\bar{A}\Lambda\Lambda^T\bar{B}^T)^T R)}{tr\{(R\bar{A})^T(R\bar{A})\}} \tag{5.15}$$

We have proofed the theorem.

Using $Theorem1$, the optimal parameters in equation (5.5) are obtained as a close-form solution given as

$$R_j^{new} = USV^T \tag{5.16}$$

$$t_j^{new} = -\frac{1}{tr(\Lambda)^2}(sRV_j - X)\Lambda_j^2 \tag{5.17}$$

$$s_j^{new} = \frac{tr((\bar{V}_j\Lambda\Lambda^T\bar{X}^T)^T R)}{tr\{(R\bar{V}_j)^T(R\bar{V}_j)\}} \tag{5.18}$$

where $UDV^T = svd(\bar{V}_j\Lambda_j\Lambda_j^T\bar{X}^T)$, $S = diag(1, 1..., det(UV^T))$.

After the transformation parameters θ are obtained, we use the new θ and the posterior probability to compute the GMM parameters. For the means x_k of GMM, it can be easily obtained by taking the partial derivative of (5.3) with respect to x_k and setting to 0. $\partial Q(\theta)/\partial x_k = 0$. Then, we substitute the new x_k to equation (5.3) and set $\partial Q(\theta)/\partial\theta_k = 0$ to obtain optimal variances. These formulas of these parameters are given as

$$u_k^{new} = \frac{\sum_{j=1}^{2}\sum_{i=1}^{N_j}\alpha_{jik}(s_j^{new}R_j^{new}X_{ji} + t_j^{new})}{\sum_{j=1}^{2}\sum_{i=1}^{N_j}\alpha_{jik}} \tag{5.19}$$

$$(\sigma_k^{new})^2 = \frac{\sum_{j=1}^{2}\sum_{i=1}^{N_j}\|\alpha_{jik}(s_j^{new}R_j^{new}X_{ji} + t_j^{new} - u_k^{new}\|}{3\sum_{j=1}^{2}\sum_{i=1}^{N_j}\alpha_{jik}} + \varepsilon^2 \tag{5.20}$$

where ε^2 is a very small positive value to avoid singularities (Horaud et al. 2011).

5.3 Implementation details and discussion

In all GMM-based methods, they need to estimate posterior probability of each point belonging to every Gaussian model in the expectation(E) step. The computation and memory complexity are very large which is $O(M * K + N * K)$, where M and N are the number of two point clouds and K

Algorithm 5.1 Generative registration algorithm

Require: Two cross-source point clouds P_1, P_2

Ensure: θ. $(u_k, \sigma_k, s_j, R_j, t_j)$

Initialization : $\theta \leftarrow 1$

EM optimization,repeat until convergence :

- E-step: compute α_{jik} by using Eq.(5.4)
- M-step: compute optimal $s_j, R_j, t_j, u_k, \sigma_k$.
  - Solve R_j, t_j, s_j by using Eq.(5.16), (5.17), (5.18).
  - Solve u_k, σ_k by using Eq.(5.19), (5.20).

Return T: $s = s_1/s_2, R = R_1/R_2, t = R_1(t_2 - t_1)$.

Return aligned points: $P_2' = sRP_1 + t$.

is the Gaussian model. In the model of generative GMM like the proposed GM-CSPC, even worse, the complexity is $O(M * N * K)$. It is prohibitive for large scale cross-source point cloud. We will describe how to effectively deal with these problems.

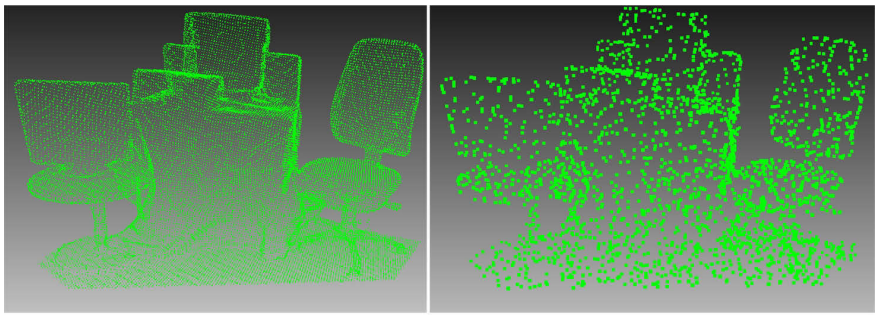

Figure 5.4: Visual results of original and down-sample point clouds.

In this chapter, the cross-source point cloud registration problem only contains rigid transformation. Hence, if we uniformly down-sample the point cloud, the global shape or structure and rigid transformation still keep the same as the original point cloud (see Figure 5.4). Due to the shape of two point clouds are all kept, the region of the GM-CSPC depicted are the same. So, the transformation matrix computed by the down-sample point cloud

are the same to the original point cloud. In this way, these two point clouds registration can be successfully converted from a large complexity problem to a feasible problem. If a rigid transformation is computed by using down-sampled point cloud, we can directly apply the rigid transformation to the original point clouds and obtain the final registration results.

In coarse matching stage, we do a roughly scale normalization by assuming the 3D containing box of two cross-source point clouds are same. The ratio between two 3D containing box are used to conduct scale normalization as a pre-processing step before ESF selection. We do this roughly scale normalization because ESF is not scale invariant. In the fine registration stage, when the GM-CSPC is completed, a residual error is computed to re-rank the matching results in the previous stage. To compute the residual error, the computed transformation matrix from revised JR-MPC is applied to perform transformation to the original point clouds. Next, the nearest neighbour of each point in one point cloud (e.g. transformed VSFM point cloud) is computed in another point cloud (e.g. transformed Lidar point cloud) and the mean of residual error between points and their neighbors is computed following. The residual error is computed by

$$E(T) = \frac{1}{N}\sum_{i}^{N} \|m_i - T(d_i)\|_2 \tag{5.21}$$

where m_i is the i^{th} point in point cloud A; d_i is the nearest neighbor of m_i in the matched point cloud B. A lower $E(T)$ means the two point clouds are more similar. However, based on our observation, the $E(T)$ always shows lower value in small-scale point clouds. To eliminate this scale bias, a penalty is defined related to the scale value:

$$E'(T) = exp(-\frac{s^2}{\alpha}) * E(T) \tag{5.22}$$

where, $exp(-\frac{s^2}{a})$ is the penalty for scale variation. α is the parameter to control the penalty, scale is estimated by the proposed generative GMM

registration method. The final ranking regions are sorted by the $E'(T)$ value and the top ranked one represents the best matching to the SFM point cloud. The whole coarse-to-fine algorithm is shown in Algorithm 2.

Algorithm 5.2 Pseudocode of coarse-to-fine algorithm

Require: cross-source point clouds

Ensure: Top 5 Registered regions

Matching :

1. Select multi-scale regions from LiDAR
2. Scale normalization
3. Compute ESF for these regions
4. Select Top K regions

Registration :

5. Down - sample point cloud
6. Compute Transformation T by **Algorithm 1**
7. Compute $E'(T)$ by Eq. (5.22)
8. Re-ranking using $E'(T)$
9. Cut off at Top 5

5.4 Experimental results

5.4.1 3D detection and localization in street-view cross-source point clouds

The experiments are conducted on real cross-source point clouds that are combined by LiDAR and SFM point cloud. LiDAR point clouds are captured from three different scenes in Helsinki (Helsinki Cathedral, Helsinki station and Library of University of Helsinki), with hundreds of millions of points on each original LiDAR point cloud. To efficiently match and register on the large volume data, the LiDAR point clouds are down-sampled into 10% of the original points. For SFM point clouds, three typical buildings are selected

and 2D images are captured by digit camera. Helsinki Station is divided into two objects: station south and station east.

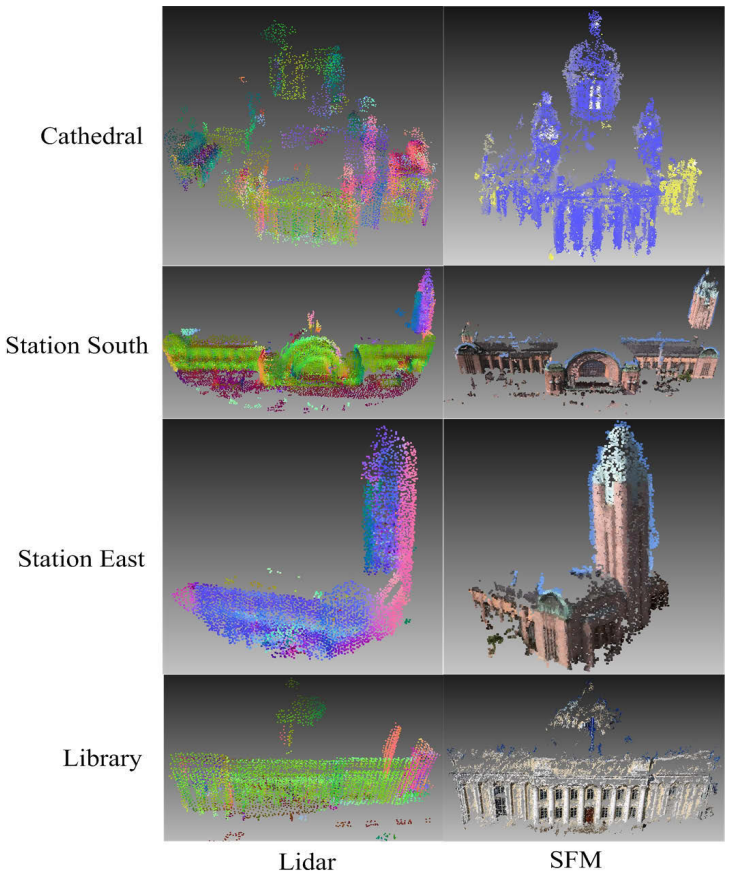

Figure 5.5: Eight point clouds of four objects named Cathedral, Station south, Station east and Library. Each row represents one object with two cross-source point clouds. The left column is LiDAR and the right is SFM.

We use 2D images and VSFM (Wu 2011) to build a software-reconstructed point clouds. The four objects of LiDAR and SFM point clouds are illustrated in Figure 5.5. Before applying the proposed algorithm, standard preprocessing, such as removal of sparse outliers, is conducted for both point clouds. Considering computation complexity reduction, the performance of the proposed algorithm is evaluated on a subset data. The subset data is generated by 7 different scale spheres scanning all the LiDAR point clouds.

The radius of the spheres ranges from 30 to 60 with an interval of 5. A hundred regions are selected under each scale. The subset data are regarded as candidate regions for matching and registration. The candidate regions will cover more than 50% areas of LiDAR point clouds. The matching and registration is then regarded as a retrieval problem. The target point cloud(SFM) is retrieved from the 700 candidate LiDAR regions (100 candidates for each one of the 7 scales).

Based on our study, the first matching stage can achieve the best performance when the number of ESF sampling level is 64. The number of potential regions kept for the second fine registration stage K is selected as 20 in all the experiments.

We define two single stage baseline systems and select ESF+ICP, ESF+GO-ICP as our compared methods. For baseline systems, like (Peng et al. 2014*c*), one is retrieved by ESF only to measure the ESF similarity of point clouds. The other is applying ICP to compute the residual error on each region in every point clouds. Since scale variation is a common problem existing in cross-source point cloud, we normalize the scale by the scale estimation method in (Peng et al. 2014*c*) before applying ICP. In the baseline system, one difference is that the scales do not adjust the residual produced by ICP, as that in the proposed method. To compare the performance of proposed method, we regard the candidate regions which cover $> 90\%$ area of the target object and $< 10\%$ points associated with the background are regarded as ground-truth data. In this chapter, rank-5 measurement is proposed and the ground-truth number is more than 5. According to their ESF similarity or final residual error, candidate regions selected from LiDAR point clouds are sorted and the rank is cut off at top 5. The algorithm shows better performance when there are more retrieved ground-truth regions. All experiments are conducted in a computer with 4-core 3.2GHz CPU and 8GB memory. The results are illustrated in Table 5.1.

As shown in Table 5.1, the single stage ESF performs faster but suffers from low accuracy. The baseline of single stage ICP, however, possesses

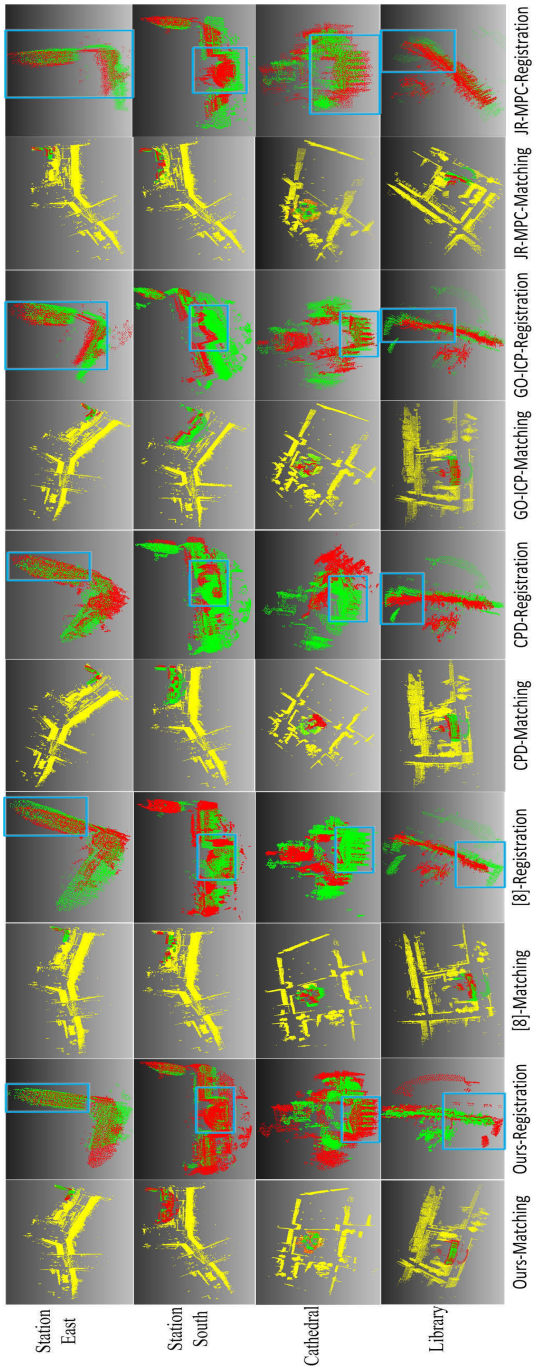

Figure 5.6: The top 1 matching and registration results of the 4 objects with the proposed method and CoarseToFine, CPD, GO-ICP. Each row represents the results for one object. The odd columns show matching results and the even columns show the detailed registration results with LiDAR in green and SFM in red. For each row, the blue regions are selected regions showing better registration accuracy of the proposed method.

Table 5.1: The performance of the proposed method and the compared methods

	cathedral		library		station south		station east	
	accuracy	time(s)	accuracy	time(s)	accuracy	time(s)	accuracy	time(s)
Baseline: single stage ESF	4	24	0	25	2	23	0	25
Baseline: single stage ICP	5	305	5	241	0	167	5	522
ESF-64 + ICP without adjusting final residual (Peng, Wu, Fan, Zhang, You, Lu & Yang 2014*c*)	5	85	3	73	0	56	4	139
ESF-64 + ICP with adjusting final residual (Peng et al. 2014*c*)	5	85	3	73	4	56	5	129
ESF-64 + CPD(Myronenko & Song 2010)	2	623	4	656	4	443	5	115
ESF-64 + Go-ICP(Yang et al. 2013)	5	4345	3	586	4	506	5	503
The proposed method (2000)	5	300	**5**	223	**5**	320	5	256
The proposed method (150)	5	54	**5**	65	**5**	57	5	67

higher accuracy, but it is the most time consuming method. The compared method CoarseToFine (Peng et al. 2014*c*) runs much faster than the baseline of single stage of ICP. It uses ESF to quickly remove many incorrect candidates, and then ICP is applied to refine the result, saving a large amount of time. However, it shows lower accuracy and efficiency to the proposed method, which can be visually seen from Station South and Cathedral in Figure 5.6. Using ESF as well, the generative GMM in the proposed method is based on the assumption that two point clouds come from the same object. If the two point clouds have plenty of differences, they are original not registered and it shows high residual error. For the proposed method, we assume two cases in our experiments: the object describe by 2000 Gaussian and 150 Gaussian. Table 5.1 shows both of two cases obtain similar high accuracy while 150 Gaussian shows obviously high efficiency than other methods. The details of accuracy and time on different Gaussian model are shown in Figure 5.7. It shows the accuracy turning point is 150 Gaussian models. Compared with other registration methods, the results of library and station south results in Table 5.1 show that the proposed algorithm is more robust in registration. The proposed method can be robust in detecting the top 5 ground-truth regions from cross-source point clouds (as described before, the ground-truth regions are more than 5). Therefore, the proposed method not only retrieves the correct regions but also registers them more accurately and efficiently than the compared methods.

In addition, the proposed method is conducted on the whole data sets, where multi-scale regions over the whole scene are tested, which means that much more negative regions are included. The proposed method shows robustness in these large challenging situation. Figure 5.6 shows the visual comparison of Top 1 matching and registration results. The results indicate that our algorithm has achieved much better performance than any other methods in those blue regions, especially for the Station South and Cathedral datasets. In terms of the efficiency, our algorithm has achieved the fast performance in Table 5.1.

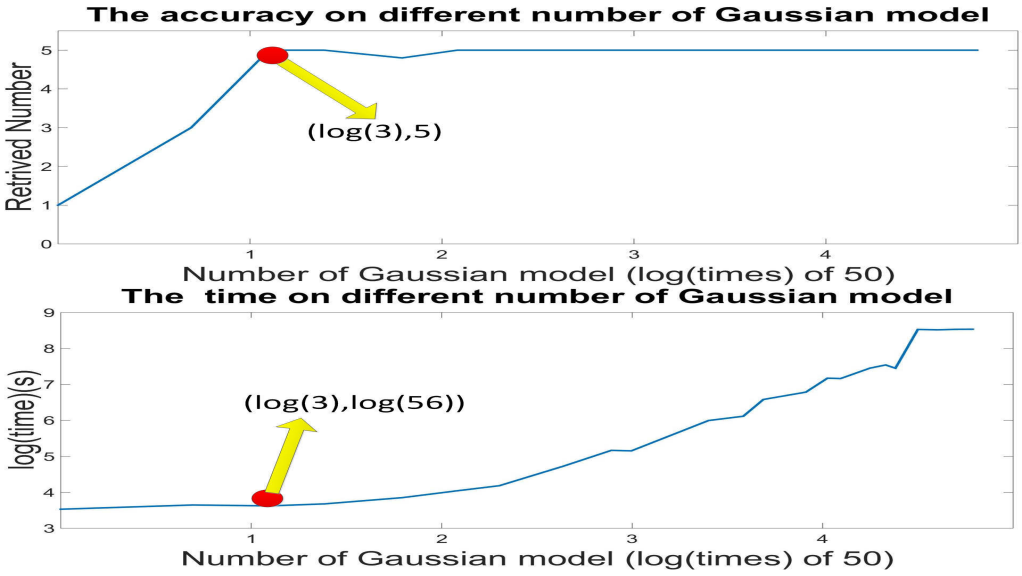

Figure 5.7: The accuracy and time performance on different Gaussian models.

5.4.2 3D scene matching and registration

In this section, we test the proposed method on 3D scene matching and registration. We use KinectFusion (KF) and IPhone 6S plus to capture 17 indoor scenes. One data is 17 sets of KinectFusion point cloud. The other is nearly 200×17 images where 200 images for one scene. The point clouds are built from these images and 17 sets of VSFM point clouds are constructed. The groundtruth is correspondent relations of set number (e.g. the first set of KF point cloud are describing the same scene with the third set of VSFM point cloud).

In this experiment, we test the accuracy of the proposed method running for 3D scene matching and registration on KF and VSFM datasets. Starting from VSFM point cloud, firstly, we detect 10 potential objects; then, we use (Corsini, Cignoni & Scopigno 2012) to downsample the point cloud to 10% of the original points and run our GM-CSPC method. The results show we successfully find the correctly matched scene at all 17 cases.

Figure 5.8 shows the final selected registration results of (Peng et al. 2014*c*) and the proposed method. In these selected datasets, both of them

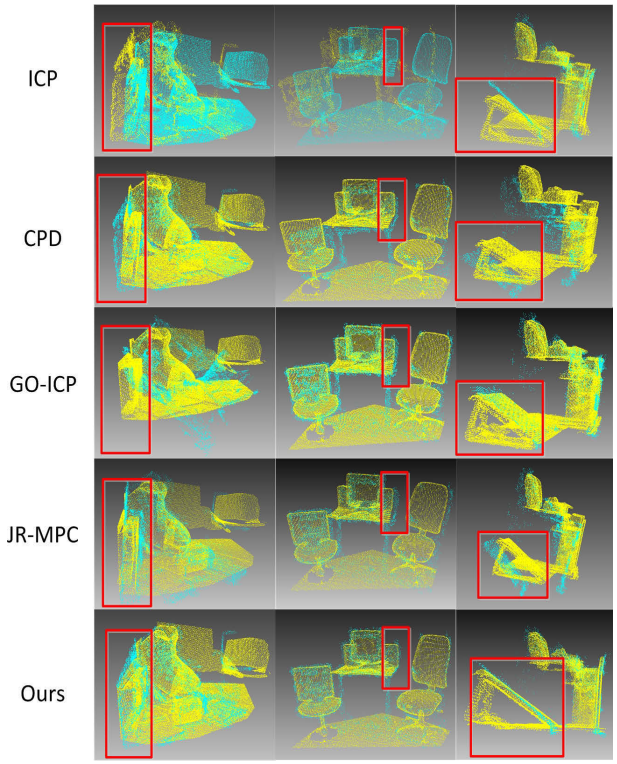

Figure 5.8: The registration results of CoarseToFine and the proposed method match and register successfully. (a) shows the results of CoarseToFine and (b) shows the results of the proposed method.

are matched and registered successfully. However, the final results show our result is much better than ICP from the visual aspect. Especially in the red box region, our results are visually better than ICP. The proposed method obtains better results because three reasons: (1) ESF is very effective in selecting potential candidates; (2) GMM is a robust model to describe a complexity scene; (3) we consider two cross-source point clouds as two samples from this virtual GMM and estimates the parameters and rigid transformation (scale, rotation and translation). If the cross-source point clouds are describing different scenes, which show different virtual GMMs, the registration error will be very large.

5.4.3 Evaluation of the proposed registration algorithm

In order to test the our proposed GMM-based registration algorithm, we use synthetic and real cross-source data to test the performance. We also compare with CPD (Myronenko & Song 2010), JR-MPC (Evangelidis & Horaud 2018) and GO-ICP (Yang et al. 2013).

Firstly, to build synthetic datasets. The datasets are simulated by three steps according to the cross-source properties discussed in Section 5.1. Step 1: Simulation on different densities and different viewpoints. For the different densities, the original point cloud is up-sampled by adding one new point to the gravity center of each small triangle on the original surface. Around 300% points will be added. For the different viewpoints, those 3D points are removed if their z coordinates are less than 0. The current view and its point cloud are known as view-1 and S1 respectively. To generate another view as view-2 and its point cloud S2, the coordinate system is rotated 60° relative to the y axis and down-samples to 30% of original point cloud. Step 2: Construction of missing point. Starting from view 2, we target to remove 10% of whole point cloud. By defining a plane region with its radius of 5% of the diameter of a ball that contains the whole point cloud, we then randomly delete several plane regions to simulate a VSFM point cloud. Step 3: Rigid transformation. A random scaling up of 3 to 5 times of the point cloud, a random rotation matrix in the x, y, z axis between 30° and 60°, and a random translation in the z axis between 0 and 50% of the largest point-point distance are added to the view-2. Step 4: Construction of noise and outliers. A white Gaussian noise with predefined signal-to-noise ratio (SNR)[2] SNR = 40dB is added to the point cloud of view-2. The outliers are constructed by down-sampling 30% of point clouds in the view-2 and adding random offsets to the coordinates for these down-sampled points. The simulated noises and outliers are combined to form a final point cloud S2. The simulated S1 and

[2]Note that SNR is inversely proportional to the variance of added Gaussian noise. In this work we set SNR at a fixed value, thus, the variance of added Gaussian noise is actually adaptive to the variance of input point clouds.

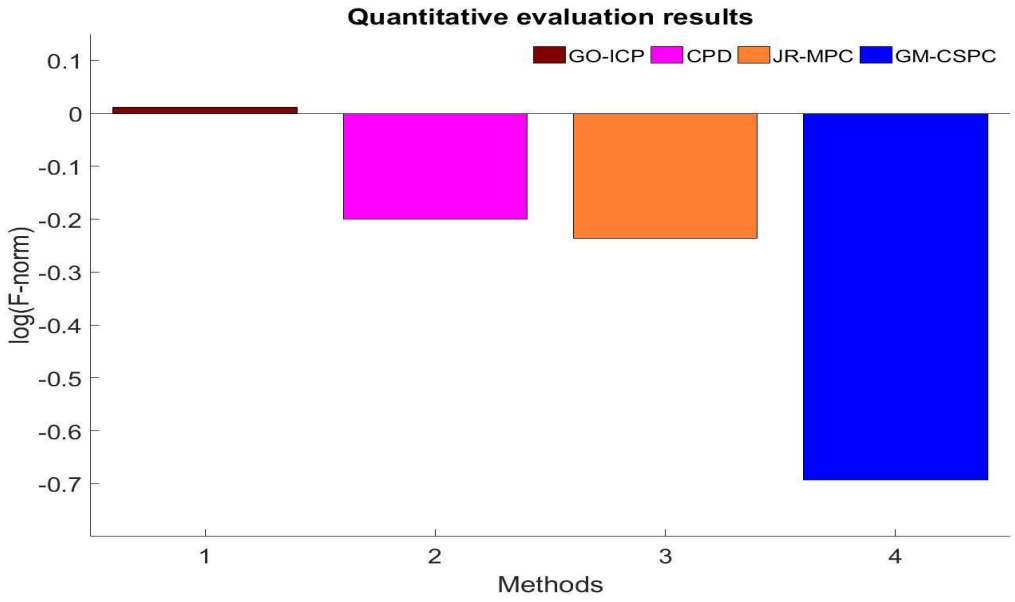

Figure 5.9: Quantitative evaluation results of F-norm metric. Our method achieves highest accuracy among these comparison methods.

S2 point clouds perceive the cross-source problems. Ten cross-source datasets are synthesized using Stanford 3D objects [3].

Then, we conduct the comparison experiments. The scale variation is normalized for JR-MPC and GO-ICP by using method in (Peng et al. 2014*c*). In the proposed method, scale is automatically estimated by the reformulated generative GMM model. To conduct a quantitative evaluation, follows JR-MPC (Evangelidis, Kounades-Bastian, Horaud & Psarakis 2014), F-norm of transformation matrix are used to evaluation the performance of algorithms. The lower the F-norm value, the high accuracy the algorithm is. Figure 5.9 shows the comparison result of log(F-norm) of transformation matrix. The proposed method obtains obviously higher accuracy than other methods. Figure 5.10 shows the visual registration results of our method and these comparison methods on synthetic datasets. The results show that our method obtains the highest accuracy on cross-source point clouds, such as

[3]http://graphics.stanford.edu/data/3Dscanrep/

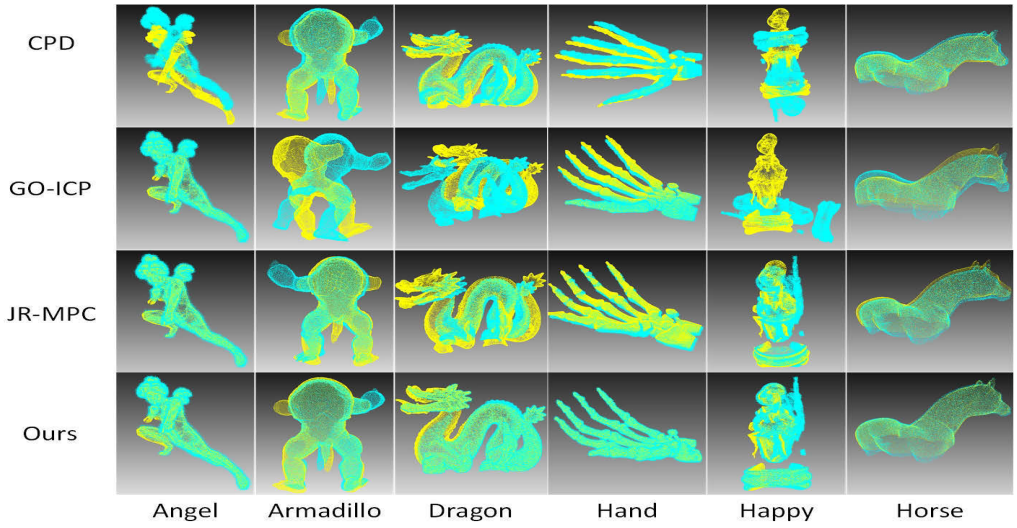

Figure 5.10: The visual registration results of our method and comparison methods on Synthetic datasets.

Armadillo, Dragon, and horse. Also, Figure 9 shows many existing methods have difficulties to handle the cross-source registrations (e.g. Happy), while our generative GMM-based algorithm still obtains the accurate results with the robustness.

We also test our algorithm on PISA[4] dataset, which is a real cross-source large building. Figure 5.11 shows the visual registration results of our method which can successfully handles the large building case (e.g. PISA). The detailed regions in the bottom row show that our method can accurately align the cross-source point clouds.

5.4.4 Scale estimation comparison

The scale estimation is a challenging problem in cross-source point clouds. There are several methods in estimating the scale variance (Lin et al. 2014)(Mellado et al. 2015). In order to test the ability of scale estimation, we compare our method with other methods on the dataset of (Lin et al. 2014). Table 5.2

[4]https://www.irit.fr/recherches/VORTEX/MelladoNicolas/category/datasets/

Table 5.2: Comparison of Relative Scale Estimation for Several Methods, with Estimated Scale and Percentage Error.

Dataset	Ground Truth	Standard Deviation	Mesh Resolution (Johnson & Hebert 1999*a*)	Keyscale (Tamaki, Tanigawa, Ueno, Raytchev & Kaneda 2010)	Standard ICP (Best & McKay 1992)	RatioICP (Lin et al. 2014)	GLS (Mellado et al. 2015)	GLS+ICP (Mellado et al. 2015)	Ours
Bunny	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000	5.000
Small Blocks (no change)	2.364	4.855 (105.37%)	1.162 (50.85%)	1.400 (40.78%)	3.029 (28.13%)	2.502 (5.84%)	2.430 (2.81%)	2.382 (1.01%)	**2.351 (1.21%)**
Small Blocks (with change)	2.424	3.833 (58.13%)	1.684 (30.53%)	2.250 (7.18%)	2.561 (5.65%)	2.543 (4.91%)	2.525 (4.16%)	2.505 (3.34%)	**2.400 (1.81%)**
Real Blocks	2.364	4.855 (105.37%)	1.162 (50.85%)	1.400 (40.78%)	3.029 (28.13%)	2.502 (5.84%)	2.430 (2.81%)	**2.382 (1.01%)**	2.462 (3.20%)

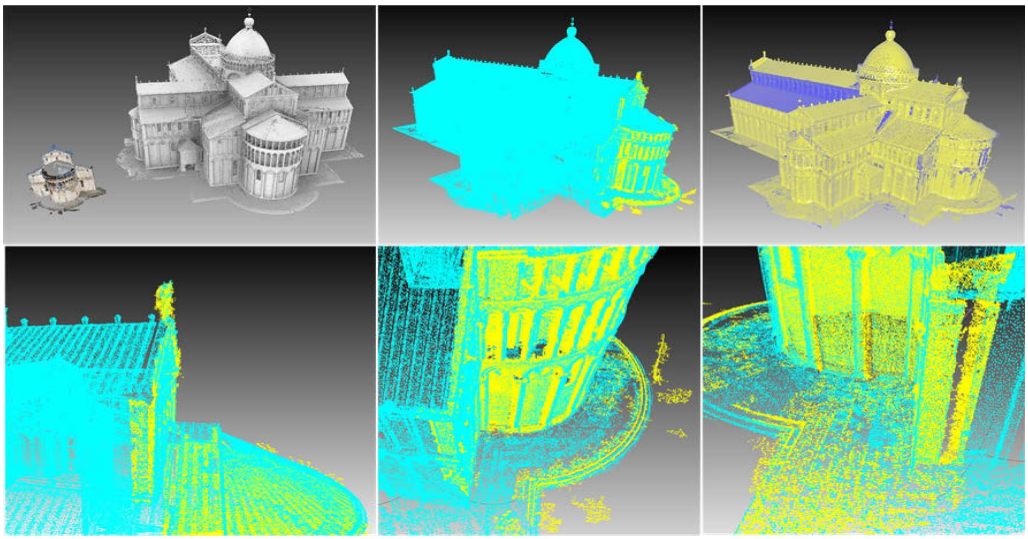

Figure 5.11: The two rows show visual results of original point clouds, the registration result in different colors, the registration in different shading techniques. The bottom row shows three sample regions of our registration result in top middle picture.

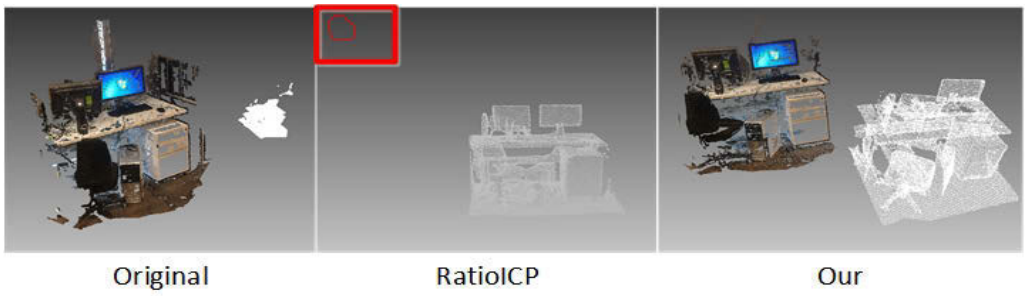

Figure 5.12: Cropped point clouds from two sensing techniques. ratioICP estimates the scale is 0.0032 and the proposed method estimates the scale is 0.62. The right two images are results of two scaled point clouds displaying in the same coordinate system.

shows the results, which show that the proposed method is able to estimate the scale of cross-source point cloud. Due to the dataset in (Lin et al. 2014) is not fully reflecting cross-source problem, in order to test the ability in cross-source cases, we also manually build a dataset using KF and VSFM point cloud. The comparison experiments are also run on our cross-source dataset. Due to the code of (Mellado et al. 2015) is not available to our community, we can only compare our method with (Lin et al. 2014). Figure 5.12 shows our method can estimate a reasonable scale for cross-source point clouds which can transform the cross-source into a unified coordinate system. However, the ratioICP has difficulty in achieving the same performance.

s

5.5 Conclusion

In this chapter, a novel coarse-to-fine algorithm is proposed to address the problem of cross-source point cloud matching and registration. In the first stage, coarse matching is performed to quickly detect a few potential matched regions. In the second stage, a generative GMM-based method is proposed to robustly deal with cross-source point clouds registration problem. We consider two cross-source point clouds as scaled samples from a virtual GMM. The registration error will be very small if they describe a same scene. It refines the matching results and accurately finds out registration regions. The proposed method does not rely on least square mean error, but rather utilizes the statistical property that is robust to the cross-source problems. It can efficiently detect the potential regions from a large scene and register them accurately. The proposed method can not only detect the regions where the reference point cloud is located in the big scene but also obtain the accurate pose related to the big scene. The future work is to develop many applications with this method in areas such as location-based service in smart city and robotics.

Compared to previous CSGM and GCTR, the method of this chapter

(GM-CSPC) is more suitable to the cases with large overlap regions. GM-CSPC and GCTR will cost a lot of memory when the GMM components and tensor dimension increase. CSGM is the most accurate algorithm and has no memory issue, but the efficiency is high.

Chapter 6

Cross-source point cloud registration by using deep neural network

6.1 Introduction

As detailed explained in the above chapter 1, point clouds captured by different kinds of sensors are cross-source point clouds, which face the mixture variations of missing data, noise and outliers, different viewpoint, density and spatial transformation. A robust and discriminative descriptor between these cross-source point clouds will highly contribute the accuracy and efficiency to the related applications such as matching/registration. To solve matching/registration problem, most of the existing methods obtain gained performance when using descriptors. However, there is no descriptor available for 3D cross-source point clouds. Recent sophisticated descriptor method (Zeng et al. 2017), which is generated by neural network, designs for same-source point clouds and assumes keypoints can be found, then selects a regular stable size of local regions around the keypoints. However, the keypoints are very hard to be robustly extracted in cross-source point clouds. Without robust keypoints, how to train the network is a research problem. Also, in 3D

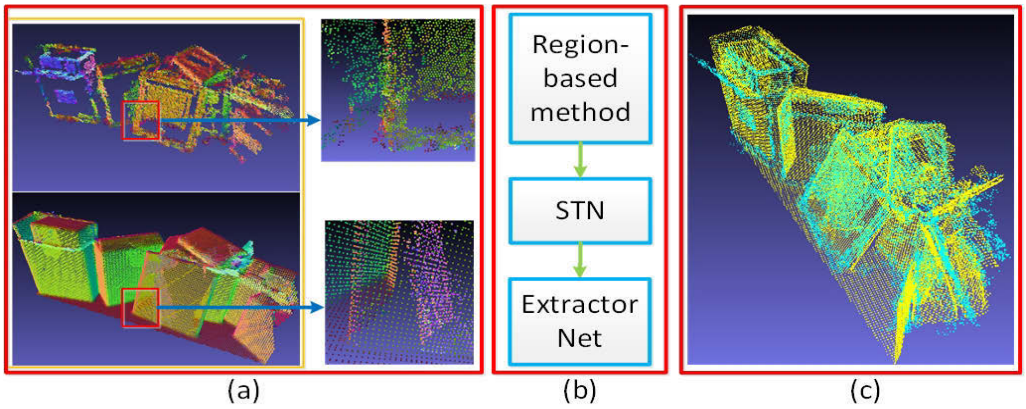

Figure 6.1: The outline of the proposed method. (a) two cross-source point clouds input into our network and details of two regions of the cross-source point clouds to show the cross-source problem, (b) shows the outline of the proposed method, (c) shows the registration result.

cross-source registration and matching, large rotation is a common problem. To solve these remaining problems, in this Chapter, we propose a learning method to generate a rotation-invariant descriptor for 3D cross-source point clouds where keypoints are difficult to be found. We are trying to use deep neural network to automatically generate a feature to describe the structure information of the point clouds.

No robust keypoints in cross-source point clouds. Cross-source point clouds are 3D data from different kinds of 3D sensors. For example, one point cloud comes from LiDAR and the other comes from reconstructing point clouds by stereo camera or structure-from-motion (SFM). Because of different sensing mechanisms, no matter whether point clouds come from active sensing (LiDAR) or passive sensing (stereo camera), the point density and where we can capture a point vary due to the change of the devices. Since it is very hard to generate a 3D point in the same place, it is very difficult to robustly extract keypoints between cross-source point clouds (e.g. Figure 6.1 (a)). Recent works on cross-source point clouds (Huang, Zhang, Wu, Fan & Yuan 2016, Mellado et al. 2016, Huang, Zhang, Fan, Wu & Yuan 2016, Huang,

Zhang, Wu, Fan & Yuan 2017*b*, Peng et al. 2014*a*) show that cross-source problems could be summarized as differences on density, missing data, noise, outliers, different viewpoint and spatial transformation. The mixture of these differences makes cross-source point clouds much more challenging. Because robust keypoints are difficult to be extracted, what is to be described by the descriptors in cross-source point clouds is a research problem. **Why do we need to care about rotation ability in cross-source point clouds.** In same-source point clouds, data acquisition usually conducts sequentially and the consecutive scans usually have small rotation. However, in cross-source point clouds, because they are from different sensors, their world coordinate systems are totally different between each other with a relationship of spatial transformation (rotation and translation). Because the rotation in cross-source point clouds is arbitrary and maybe large, even for a same 3D point cloud region, the descriptor generated by a regular CNN is totally different. Therefore, descriptor on cross-source point clouds should cares more about rotation ability than same-source. To generate a discriminative descriptor in cross-source point clouds, we argue that firstly, the point clouds are projected to a rotation-invariant space and then a discriminative descriptor is generated from this space.

The existing 3D descriptors can be summarized as two categories: local descriptor and global descriptor. The typical examples of local descriptor are Spin image, FPFH (Fast Point Feature Histogram)(Rusu, Blodow & Beetz 2009*b*), SHOT (Signatures of Histograms of Orientations) (Tombari, Salti & Di Stefano 2010*b*) and 3DMatch (Zeng et al. 2017). Local descriptors are computed for individual points and depending on local regions around the points. The typical examples of global descriptor are CVFH (Clustered Viewpoint Feature Histogram)(Aldoma, Tombari, Rusu & Vincze 2012) and ESF (Ensemble of Shape Functions)(Wohlkinger & Vincze 2011). Global descriptors are computed for the whole regions of the point clouds and not computed for individual points. Although 3DMatch shows similarity to our work, all the existing descriptors including 3DMatch face challenges in han-

dling cross-source problems (demonstrated in experimental part). Designing a descriptor on cross-source point clouds, which are from different kinds of sensors, is a highly urgent work in the environment where sensor development is fast and sensor fusion devices are widely used (e.g. LiDAR and RGB cameras in autonomous driving).

The **behind principle** of the proposed method is that, the structures of two aligned cross-source point clouds remain similar, even though they have variation on rotation and cross-source problems. This is the reason why human can align them. To use the structure information in our method, we need a descriptor to robustly describe it. However, recognizing the structure information is easy to human but very challenging to a handcraft descriptor. In this chapter, we use a deep neural network to learn a general function, which allows us to generate rotation-invariant descriptor for 3D cross-source point clouds. The key features of the descriptor are that 1) it is invariant to rotation transformation, 2) it works significantly well in 3D cross-source point clouds, because region-based method not only overcomes situations where keypoints are difficult to be robustly found but also uses the structure information; 3) it is suitable to different irregular voxel size (irregular regions).

The contribution of this Chapter are four points: (1)) a region-based method is proposed to utilize the structure information on cross-source point clouds where keypoints are difficult to be extracted; (2) a rotation-invariant descriptor is generated for cross-source point cloud by a novel parallel spatial transformation network; (3) a structure-based registration framework is proposed by using the proposed region-based descriptor; (4) we design a synthetic 3D cross-source point cloud training database and use it to train the network, which can be generalized to real cross-source point clouds.

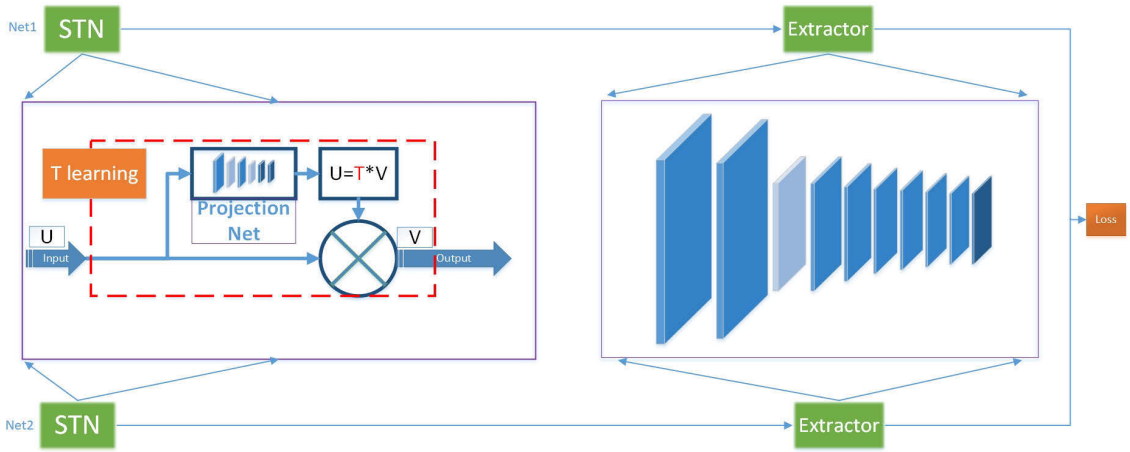

Figure 6.2: 3D parallel spatial transformation network (*PSTN*) contains 3D spatial transformation network (STN) and feature extraction network (Extractor). The goal of STN is transformation T learning. The *Projection Net* in the figure aims to learn the parameters of transformation matrix T. *Extractor* is a deep convolution network which contains eight 3D convolution layers, one 3D pooling layer and one fully convolution layer

6.2 Region-based descriptor learning

In this section, the parallel transformation network is proposed to generate rotation-invariant descriptors for matching problem. Firstly, the region-based method is proposed to feed 3d cross-source point clouds into neural network where keypoints are difficult to extract. Secondly, the parallel spatial transformation network is proposed which incorporates 3D spatial transformation network and feature extractor network. Thirdly, 3D spatial transformation network is introduced in details. Fourthly, the loss function and the details of gradient forward are introduced.

The advantages of the proposed method are three aspects: first, the region-based method proposes a feasible way to use the structure information of cross-source point clouds where robust keypoints are difficult to extracted. Second, we deal with rotation transformation in our descriptor by designing a new parallel transformation network containing 3D spatial transformation

network and feature extraction network. It can perceive the ability to solve the rotation in consistent rotation space. Third, matching and registration with the proposed descriptor is both more accurate and efficient than previous registration methods which use sophisticated optimization strategies.

6.2.1 Region-based volumetric method

Previous learning methods usually extract 3D keypoints and using a local region around the keypoints to describe the keypoints. However, due to the mixture problem of large noise, outliers, density, spatial transformation and missing data in cross-source point clouds, it is very hard to extract robust keypoints. Therefore, it is very hard to generate positive samples in cross-source point clouds by using keypoint-based methods. In this chapter, we use a region-based method to extract the meaningful region to feed the network. We do not need to extract keypoints.

Supposing two cross-source point clouds PC1 and PC2, our goal is to construct positive and negative samples from them. Firstly, features (e.g. curvature, FPFH, normal or 3D coordinate) are computed for every point in PC1 and PC2. Secondly, starting from PC1, we randomly select a point and use the features of the point to cluster similar neighbor points. If the number of clustering points is larger than a threshold (e.g. 50), the 3D box containing these points are recorded. During the process, the aim is to keep those regions containing enough points for network to learn the similarity. Thirdly, using this 3D box, we crop a region from PC2. If the number of points is larger than a threshold, we consider these two regions from PC1 and PC2 as positive samples (matched regions). A negative sample (non-matched regions) can also be captured. The center of its containing box is far away from this positive sample. The distance ranges from $3 * r$ to max radius of point clouds. The r is the radius of the 3D containing box. For each scene, we sample the same number of positive and negative samples to construct a training dataset. An interesting thing is that, our method can delete the samples whose most parts are located in the missing data areas.

This is reasonable because the regions whose most points are missing have no stable structure maintained.

Because 3D point clouds are irregular data, to apply 3D CNN to 3D cross-source point clouds, the irregular data needs convert to regular data. Truncated signed distance function (TSDF) is used to convert the 3D point cloud to volumetric data. In this chapter, we find the nearest neighbor for each voxel if there is a 3D point located in the search radius. The distance of each voxel is truncated to $[-1, 1]$. If the signed distance beyond this searching range, we set it to 0.

Then, we introduce the parallel spatial transformation network to generate rotation-variant descriptors for 3D cross-source point clouds registration.

6.2.2 Parallel spatial transformation network

Rotation transformation is a common problem in cross-source point cloud registration application. For example, users may take a picture vertically while LiDAR database may take horizontally, it is rotation variation which contains 90° rotation and a scale changing. Reconstructing point clouds from these images faces large rotation variation to LiDAR. Previous learning methods (e.g. 3DMatch (Zeng et al. 2017)) have not touched this problem and show challenging in dealing with this problem. Also, data augmentation needs large amount of samples and is difficult to train a rotation-invariant ability in consistent space. To deal with the rotation variation in registration problem, we propose a parallel spatial transformation network (PSTN) to generate rotation-invariant descriptor in rotation-invariant space.

Parallel spatial transformation network incorporates 3D spatial transformation network and feature extraction network. It is a symmetric network that two networks share a set of parameters. Input two TSDFs, which describe two regions of cross-source point clouds, they are both transformed by their 3D spatial transformation network and further input into feature extraction network. Therefore, many rotation versions of the same TSDF go in then the same resampled TSDF should come out, such that the subsequent

descriptor network can produce a rotation-invariant descriptor.

This network structure is shown in Figure 6.2. The projection net is $C \rightarrow A \rightarrow P \rightarrow C \rightarrow A \rightarrow P \rightarrow F1 \rightarrow A \rightarrow F2$, where C represents standard 3D CNN with 3*3*3 kernel size, A represents rectified linear activation function, P represents 3D pooling with window size of 2*2*2 and zero padding, and $F1$ and $F2$ are 3D full convolution layers separately having 70 kernels and 7 kernels. This is the key revision for spatial transformation network. The learned 7 parameters are used to generate 3×4 transformation matrix. The input grid will be transformed by using the transformation matrix. Then both transformed samples are used as inputs for the further deep CNN to generate the descriptors. Deep CNN is $C \rightarrow C \rightarrow P \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow C \rightarrow F3$, where C represents standard 3D CNN with 3*3*3 kernel size, P represents 3D pooling with window size of 2*2*2 and zero padding, and $F3$ is 3D full convolution layers with 512 kernels.

6.2.3 3D spatial transformation network

The details of 3D spatial transformation network (STN) are shown in Figure 6.2. In this part, the goal is to learn the transformation matrix T. We consider the 3D spatial transformations contain rotation and translation, so that the transformation matrix T is 3×4 that can be easily built from these seven spatial transformation parameters including four quaternion and three translation parameters.

$$T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_{14} \\ r_{21} & r_{22} & r_{23} & t_{24} \\ r_{31} & r_{32} & r_{33} & t_{34} \end{pmatrix} \tag{6.1}$$

Given an input data, we use a projection net to learn the seven spatial transformation parameters. The projection net is standard convolution neural network and ends with a fully convolution layer of 7 kernels. Using the seven parameters, the transformation matrix T can be built. Defining the input feature map as U and the output feature map as V, the transformation

between feature maps is $V = T^{-1}U$. After the transformation, we use the bilinear sampling method to extract the data of V from the input feature map U. Then, the resampled output V is used as the input for the following layers.

The proposed parallel spatial transformation network integrates 3D STN in a parallel stream. Both the network branches transform their input grid into canonical space consistently. Then, the subsequent descriptor network can produce a rotationally-invariant descriptor. The method in (Jaderberg, Simonyan, Zisserman et al. 2015) is close to our 3D spatial transformation network. However, the differences between them are: (1) The original method (Jaderberg et al. 2015) is designed for recognition, but we extend the it to parallel network structure, which is elaborated to deal with matching problem. (2) We only learn seven independent parameters which is perfectly suitable to 3D point cloud rigid transformation, while the original algorithm tries to learn the whole transformation matrix.

6.2.4 Loss function

To learn the network, the contrastive loss is minimized,

$$loss = \frac{1}{2}(Yd^2 + (1-Y)max(0, m-d)^2) \tag{6.2}$$

where Y is the label (match labels 1, non-match labels 0), d is the distance of two network stream outputs, m is a margin to control when the negative sample begins to affect the network training.

In the forward and backward process of the proposed network, because the convolution layer and max pooling layer are standard neural network layers, their forward and backward are standard way. The 3D spatial transformation network is the only segment we need to care about.

In the forward process, the relationship between input feature map U and output feature map V has been built after the transformation $V = T^{-1}U$. To obtain the value of each voxel in V, we use the 3D bilinear sampling method

to sample values in the input feature map U.

$$V_i^c = \sum_n^X \sum_m^Y \sum_l^Z (U_{uml}^c max(0, 1 - |x_i^s - m|) \\ max(0, 1 - |y_i^s - n|) max(0, 1 - |z_i^s - l|)) \tag{6.3}$$

where X, Y and Z are the dimensions of 3D input feature map U, x_i^s, y_i^s and z_i^s are the point positions in the output V, m, n and l are 8 neighbors of the point and the output feature map V has the dimension of X', Y' and Z'.

In the backward process, the gradient flows back from V to projection network output θ. Based on eq. (2), the gradient has two components: firstly, V is partial derivative to x, y and z; secondly V is partial derivative to θ. For partial derivative of x

$$\frac{\partial V_i^c}{\partial x_i^s} = \sum_m^Y \sum_l^Z (U_{uml}^c max(0, 1 - |y_i^s - n|) \\ max(0, 1 - |z_i^s - l|)) C(m, x_i^s) \tag{6.4}$$

where $C(m, x_i^s)$ is a constant value to limit the scope of the partial region. Here, we consider the 8 near regions. Therefore,

$$C(m, x_i^s) = \begin{cases} 0 & if\ |m - x_i^s| > 1 \\ 1 & if\ m - x_i \geq 0 \\ -1 & if\ m - x_i^s < 0 \end{cases} \tag{6.5}$$

There are similar equations for $\frac{\partial V_i^c}{\partial y_i^s}$ and $\frac{\partial V_i^c}{\partial z_i^s}$. Then, the partial derivative of θ is very easy to be derived from transformation equation $U = TV$. Take x as a example, $x_s = x_t * r_{11} + y_t * r_{12} + z_t * r_{13} + r_{14}$, combined with eq. (4), we get

$$\frac{\partial V_i^c}{\partial r_{11}} = \frac{\partial V_i^c}{\partial x_s} * x_t \tag{6.6}$$

The other elements of derivative of θ can be computed in a similar way.

6.3 Network learning

In this section, we will describe the our contribution about how to learn our network in 3D cross-source point clouds. They include the preparation of the network input and the network training strategy. The advantage of this contribution is that a fast way is proposed to use the synthetic method to train the network and experiments show that it can be generalized to real cross-source point clouds.

6.3.1 Cross-source point cloud dataset construction

To train the network to perceive the ability in dealing with cross-source point cloud registration problem, we need a cross-source point cloud database with groundtruth. The following section will introduce how to build the database.

Synthetic dataset construction

Database construction with groundtruth is highly time-consuming and labour-consuming. There is currently no available cross-source point cloud database. To train the network, instead of spending many years to label a cross-source point cloud database, we use synthetic method to build a synthetic cross-source database with groundtruth. The key element is that the cross-source problems are delicately summarized and simulated to synthesize a database. Also, training on synthetic datasets would be generalized to real cross-source point clouds, which is also our goal.

As summarized in (Huang, Zhang, Fan, Wu & Yuan 2016, Peng et al. 2014*a*), cross-source point clouds contain four main variations: missing data, noise and outliers, density, rotation transformation. To simulate cross-source point clouds problem and construct a synthetic dataset, four steps are necessary , including removing plane regions, simulating noise and outliers, simulating density and simulating spatial transformation. *Removing plane regions:* the goal is to simulate missing data variation built by SFM point cloud which usually cannot produce points on texture-less plane. We use

3D RANSAC to estimate the plane region in the point clouds and remove some of the plane regions. *Noise and outliers:* The goal is to simulate noise and outliers variation. We add Gaussian noise and outliers [1]. *Density*: the goal is to simulate density variation. We downsample to 50% of the original point clouds. *rotation transformation:* the goal is to simulate spatial transformation variation. we simulate spatial transformation by adding rotation, translation to the point clouds. In order to make the training samples more diverse, we do not add spatial transformation in each point cloud, but in each sample. Therefore, there will be much more cases of spatial transformation, by learning which our method can have a strong ability of transformation invariance. We use this method to build a synthetic database from same-source registration point cloud databases (Pomerleau, Liu, Colas & Siegwart 2012). We only consider the registration pairs whose overlap is large than 50%, because lower overlap ratio means two scans are far away and it is also very hard for human to align them.

Real dataset construction

Kinect and RGB camera are utilized to construct cross-source point clouds. We use KinectFusion and VSFM to reconstruct 50 indoor scenes to be real cross-source point cloud datasets.

6.3.2 Network Training

The network was trained using stochastic gradient descent with a base learning rate of 10^{-3}, momentum is 0.99, and weight decay is 5^4. Random sampling matching and non-matching 3D training patches from reconstructions is performed on-the-go during network training. We used a batch size of 128. We assume that if two 3D patches correspond to the same region, then they are similar and label is 1, dissimilar and label is 0 otherwise. Therefore, we

[1] Gaussian noise: SNR is 40B. Outliers are points disturbed with 1% $\sim$ 5% of the radius of their belonging point clouds

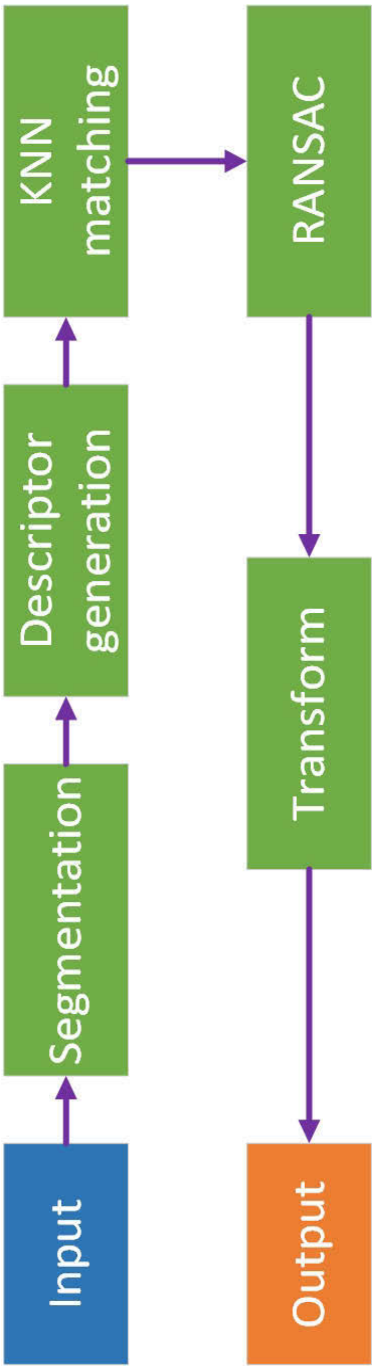

Figure 6.3: Structure-based registration framework.

train our network with two streams in a Siamese fashion where each stream independently computes a descriptor for a different local 3D region. The first stream takes the local 3D region from point cloud P1, while the second stream takes the local 3D region from point cloud P2. Both streams share the same architecture and underlying weights. In training process, the network has an input of a same number of positive and negative samples. With the same positive and negative samples fed for the network, it has shown high efficiency in learning discriminative descriptors (Simo-Serra, Trulls, Ferraz, Kokkinos, Fua & Moreno-Noguer 2015, Yi, Trulls, Lepetit & Fua 2016, Zeng et al. 2017).

6.4 Structure-based registration framework

The registration framework contains four steps which is shown in Figure 6.3. Firstly, the cross-source point clouds are segmented into many segments by using their geometrical topology. Following (Huang, Zhang, Fan, Wu & Yuan 2016), we firstly compute the descriptor of each 3D point and uses local clustering method to group points with similar descriptor around some seeds. The point cloud is then clustering into many segments and the central points of these segments are used as representation of structure of the point cloud. Secondly, the descriptor of these central points are generated by using the proposed region-based learning descriptor. For the region of each central point, we only consider the area enclosed by the bounding box of its correspondent segment. Thirdly, the nearest neighbor is computed for each point. In order to keep the structure of 3D point clouds, we build the octree of the point clouds and compute the nearest neighbor for each 3D points. We do the left and right check and the stable match are the points that the match is the same when both in left searching in the right tree and right searching in the left tree. This octree searching is high efficient. To be robust to the outliers, we use RANSAC to remove the outliers and compute the final transformation matrix. Then, the point clouds are registered by

using the transformation matrix above.

6.5 Experiments

In this section, we conduct experiments about the proposed method. Firstly, we analyze the robustness of the learned descriptor in handling different variants. Secondly, keypoint matching is evaluated to test the robustness of discriminative performance. Thridly, same-source and cross-source point clouds datasets are utilized to demonstrate the ability in solving registration problem.

Datasets: We use Princeton geometric registration benchmark (PGRB) dataset (Zeng et al. 2017), ASL Dataset (Pomerleau et al. 2012), Synthetic cross-source point cloud dataset as discussed above and real cross-source point clouds to test the performance descriptors. RGRB and ASL datasets are aimed to evaluate the ability in solving same-source point cloud registration and compare fairly with same-source registration methods and descriptors. Synthetic and real cross-source datasets are aimed to evaluate the performance in solving cross-source point cloud registration and descriptors.

- **Princeton registration benchmark dataset (Zeng et al. 2017)**: The dataset is captured in indoor scene by using RGBD sensor. The point cloud is reconstructed from the depth by using voxel fusion (Curless & Levoy 1996). It contains 54 scenes for training and 8 scenes for testing. Each scene contains more than 30 scans, usually 50 scans.

- **ASL dataset (Pomerleau et al. 2012):** The dataset is captured by Laser and record the position in CSV format. We have converted all the datasets into ply format. The dataset contains scenes including 2 indoor, 5 outdoor and 1 mixed scenes. Each scene contains about 30 scans of point cloud. We take 1 indoor and 1 outdoor scenes for testing. The other 6 scenes for training.

- **Synthetic cross-source point cloud datasets:** As discussed in

6.3.1, we construct synthetic datasets to simulate cross-source problem. In order to test the ability of the proposed method in both indoor and outdoor scenes, the cross-source point clouds are simulated from ASL dataset. We take 1 indoor and 1 outdoor scenes for testing. The other 6 scenes for training.

- **Real cross-source point clouds:** We capture real cross-source point cloud by using Kinect and iPhone 7 RGB camera. To capture large scene in indoor scene, we use KinectFusion to merge many scans. Visual structure from motion (VSFM) is utilized to reconstruct point cloud from RGB images.

Compared methods: Many recent state-of-the-art methods are selected as our comparison methods. The selection criterion is based on their reported ability in dealing with components of the cross-source problems.

- **GO-ICP (Yang et al. 2013)** utilizes bound-and-branch algorithm to estimate the global solution of registration problem.

- **Fast (Zhou, Park & Koltun 2016)** provides a fast registration method when the initial correspondence is known.

- **Super4PCS (Mellado et al. 2014)** utilizes coplanar 4-points sets and smart indexes to align the point cloud.

- **FPFH (Rusu et al. 2009*b*)+RANSAC** utilize FPFH descriptor and RANSAC, to demonstrate the performance of our descriptor in handling registration problem.

- **3DMatch (Zeng et al. 2017)+RANSAC** utilize 3DMatch descriptor and RANSAC.

6.5.1 Evaluation of the descriptor

6.5.2 Descriptor matching

In this section, we evaluate the quality of a learned descriptor by testing its ability to distinguish between matching and non-matching local 3D patches of keypoint pairs and compare with other state-of-the-art methods. Follows 3DMatch (Zeng et al. 2017), we build a collection of 30000 3D patch pairs, with 1:1 ratio between matches and non-matches. Then compute the descriptor of them by using descriptor methods. Follows the evaluation metric in (Zeng et al. 2017), our evaluation metric is the false-positve rate (error) at 95% recall, the lower the better. we compare the proposed algorithm with several state-of-the-art geometric descriptors in both same-source and cross-source point cloud datasets. The FPFH (Rusu et al. 2009*b*) and spin image (Johnson & Hebert 1999*b*) descriptors are selected as comparison descriptors. For both methods, we implemented by using point cloud library (PCL). Because we also record the center location of each 3D patches, their descriptors are computed on the depth-fused point cloud.

Same-source descriptor matching

We sample 30000 3D patches on Princeton test datasets. The comparison results are reported in Table 6.1.

Methods	Error
Fast Point Feature Histograms (FPFH) (Rusu et al. 2009*b*)	82.6
SpinImage Descriptor (Johnson & Hebert 1999*b*)	61.3
3DMatch (Zeng et al. 2017)	35.3
PSTN (Ours)	**28.2**

Table 6.1: Keypoint matching error (95%) on ASL same-source point cloud datasets.

Cross-source descriptor matching

We sample 30000 3D patches on synthetic cross-source point cloud datasets. The comparison results are reported in Table 6.2.

Methods	Error
Fast Point Feature Histograms (FPFH) (Rusu et al. 2009*b*)	85.6
SpinImage Descriptor (Johnson & Hebert 1999*b*)	68.5
3DMatch (Zeng et al. 2017)	40.8
PSTN (Ours)	**30.2**

Table 6.2: Keypoint matching error (95%) on synthetic cross-source point cloud datasets.

6.5.3 Comparison and analysis

To evaluate the practical usage of the proposed descriptor, we combine the PSTN with RANSAC search algorithm for registration and compare with many state-of-the-art registration methods. The comparison experiments are conducted on standard same-source registration benchmark (Zeng et al. 2017) and synthetic cross-source benchmark datasets based on ASL dataset (Pomerleau et al. 2012). To test the generalization performance on real cross-source point cloud by training on synthetic training datasets, we manually conduct a real cross-source benchmark dataset and use it to do the comparison experiments.

More specifically, given two 3D scanned point clouds, we firstly randomly sample n keypoints for each point cloud. Then, the point clouds are divided into many voxels by the dimension size or physical size(e.g. divide into 0.01m at each dimension). We compute the descriptors of PSTN and 3DMatch by using $30 \times 30 \times 30$ patches around the keypoint. The descriptors are computed for all the $2n$ keypoints in the given two point clouds. Then, using the descriptors, we find the keypoints whose descriptors are mutually

closest to each other in Euclidean space. Finally, RANSAC is used on these keypoints over their 3D positions and the rigid transformation between the point clouds is estimated.

For the performance evaluation of registration comparison, following (Choi, Zhou & Koltun 2015, Zeng et al. 2017), we compute the recall and precision to measure that (1) how well it estimates rigid transformation matrices, (2) how well it finds loop closures. Given two frames (P_i, P_j) from a scene, we recognize the transformation estimation T_{ij} as true positive if the overlap ratio is larger than 30% between $T_{ij}P_i$ and P_j and T_{ij} is sufficiently close to the ground-truth transformation $T_{ij}^{\star}$. We recognize the transformation estimation T_{ij} as correct if the projection RMSE of the ground truth correspondences $K_{ij}^{\star}$ between P_i and P_j is less than a threshold $\gamma = 0.4$.

Comparison on same-source point cloud registration

In this section, we compare the rotation performance in dealing with real 3D same-source point cloud registration. We use the apartment set of ASL dataset (Pomerleau et al. 2012), which has 44 scans which are captured by laser sensor. It contains groundtruth of pair registration. This dataset is not included in the training dataset. The recall and precision are evaluated [2]. The comparison results are shown in table 6.3. It shows that our method obtains both higher recall and precision than other methods. This experiments show the high performance in handling challenging laser point cloud registration problem.

We also evaluate and compare on the Princeton registration dataset (Zeng et al. 2017), which is captured by using RGB-D datasets. Because the RGBD sensor is totally different from laser sensor, we retrain our PSTN by using the training dataset of Princeton registration dataset. For 3DMatch, we also

[2]The recall is $\frac{A}{B}$ and the precision is $\frac{A}{C}$, A is the pair number of accurate registration with registration error less than a threshold, B is the total pair number and C is the pair number that overlap ratio is larger than 30%. Registration error is $\|Tx_i - y_i\|^2/N$, T_i is the transformation matrix, x_i, y_i are two correspondent points, N is the pair number.

Methods	Precision (%)	Recall (%)
Go-ICP (Yang et al. 2013)	12.7	25.0
Super4PCS (Mellado et al. 2014)	10.3	23.6
Zhou et al. (Zhou et al. 2016)	26.4	54.5
FPFH (Rusu et al. 2009*b*) + RANSAC	18.2	45.5
3DMatch (Zeng et al. 2017) + RANSAC	29.3	56.1
PSTN + RANSAC	**31.0**	**68.5**

Table 6.3: Comparable registration performance on same-source ASL datasets. The proposed method with RANSAC highly outperforms other registration methods.

Methods	Precision (%)	Recall (%)
Go-ICP (Yang et al. 2013)	14.8	18.5
Super4PCS (Mellado et al. 2014)	10.4	17.8
Zhou et al. (Zhou et al. 2016)	23.2	51.1
FPFH (Rusu et al. 2009*b*) + RANSAC	19.1	46.1
3DMatch (Zeng et al. 2017) + RANSAC	40.1	66.8
PSTN + RANSAC	**42.3**	**78.4**

Table 6.4: Comparable performance on same-source Princeton datasets.

use the pre-trained model to conduct this experiments. The results are shows in Table 6.4. It shows that the proposed method is much better than the state-of-the-art method in aligning 3D point cloud by integrating RANSAC. The higher accuracy is contributed by the discriminative and robustness of the learned descriptor.

3D rotation variation is commonly existed in 3D point cloud. The proposed method is able to robustly transform the input point cloud into a rotation-invariant space. In this space, we can generate highly discriminative descriptor. Discrimnative descriptor is very important for registration and matching. In our experiments, we only use KNN to select highly related

points and use RANSAC to estimate the transformation matrix. All these process are all very efficient. The experiments demonstrate high accuracy performance.

Comparison on cross-source synthetic dataset

In this section, we evaluate the performance on cross-source synthetic dataset. The tested dataset uses the construction method in section 6.3.1. The groundtruth is the spatial transformation. The recall and precision are computed. The results in Table 6.5 show that the proposed method achieves both higher precision and recall than other methods. This is because our method is able to solve the rotation variant and cross-source problems.

Methods	Precision (%)	Recall (%)
Go-ICP (Yang et al. 2013)	10.8	18.5
Super4PCS (Mellado et al. 2014)	10.3	19.1
Zhou et al. (Zhou et al. 2016)	27.7	54.5
FPFH (Rusu et al. 2009*b*) + RANSAC	8.2	42.5
3DMatch (Zeng et al. 2017) + RANSAC	19.4	53.7
PSTN + RANSAC	**37.0**	**69.4**

Table 6.5: Comparable performance on cross-source synthetic benchmark datasets. The proposed method has a better ability to solve cross-source problems and the problems of rotation variant.

Comparison on real cross-source point cloud registration

In order to test the generalization performance of the proposed method, we also test on real cross-source point clouds. We use KinectFusion to capture one source of point cloud and VSFM to reconstruct point cloud from RGB cameras. These two point clouds from KinectFusion and RGB cameras are widely used in indoor scenes. Ten sets of cross-source point clouds are captured and their registration performance is evaluated. For learned descriptor

3DMatch and our PSTN, we use the model trained by using the cross-source benchmark dataset.

We compare GO-ICP, Super4PCS, RANSAC and 3DMatch + RANSAC with the proposed PSTN + RANSAC. To get the registration result, RANSAC is used to compute the transformation based on the descriptor similarity. The registration results are shown in Figure 6.4. These results show that the proposed method can successfully deal with the real cross-source point cloud registration. However, other methods show much difficulty in dealing with the real cross-source point cloud registration. These results also demonstrate that training on synthetic datasets can be generalized to real cross-source point clouds.

Methods	CSGM	GCTR	GM-CSPC	PSTN
error	-2.796	-1.7447	0.5155	-2.769

Table 6.6: Comparable performance of our proposed registration methods in solving challenging indoor registration problems. The results show that CSGM is the highest accurate algorithm while PSTN achieves comparative accuracy.

Comparison with CSGM, GCTR and GM-CSPC

In this section, we compare the performance on one example (frame1 and frame 3) in Princeton registration benchmark dataset mentioned above. We compare the log(F-norm) of the estimated and ground-truth transform matrix. The results of Table 6.6 show the comparison results. The CSGM obtains best result and PSTN follows with comparative accuracy. GM-CSPC fails to align these challenge dataset. The reason is that GMM-based algorithm rely on the point cloud distribution similarity. However, the distribution in these datasets shows a lot of difference when there is missing data and different viewpoint in different frames.

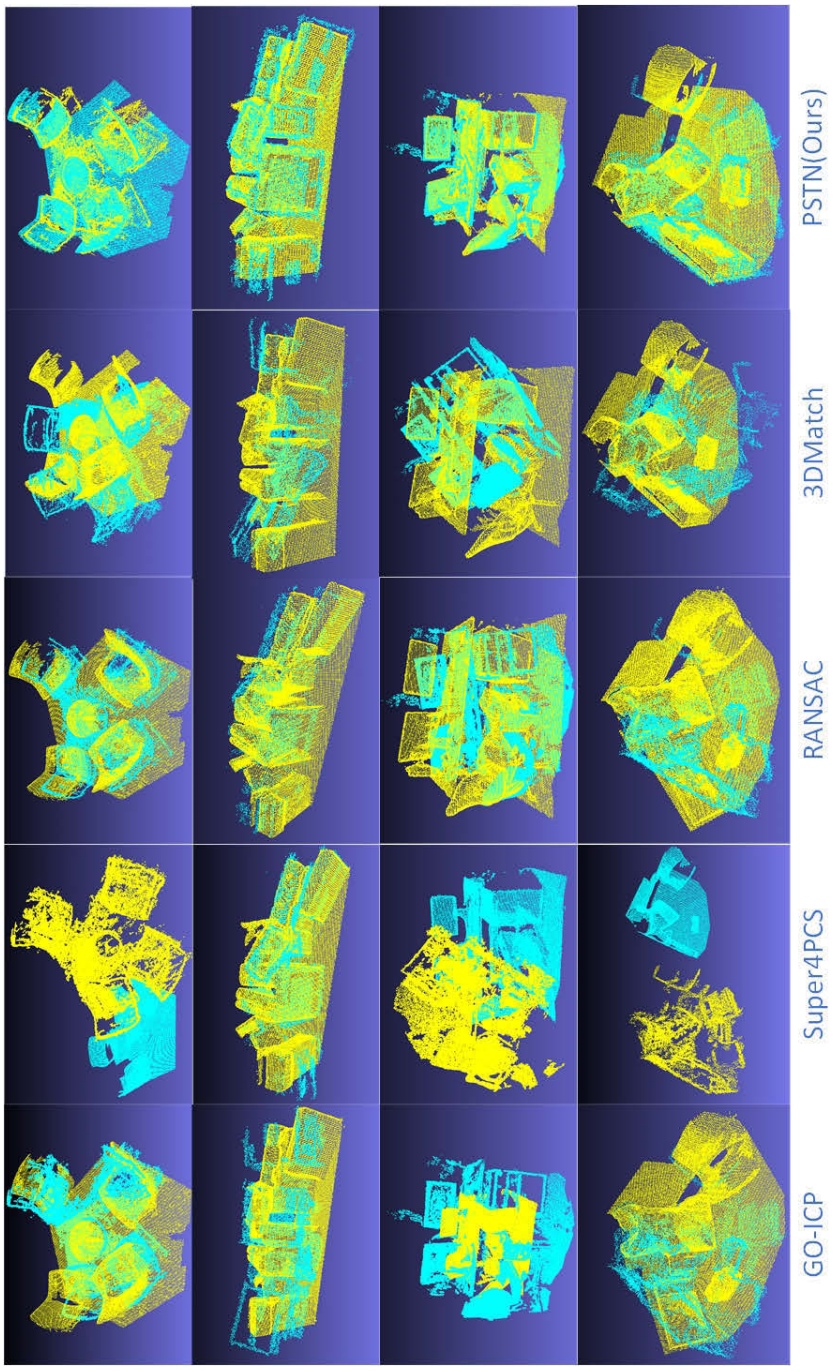

Figure 6.4: Visual registration results of real cross-source point clouds. The results show that our method can be generalized to real cross-source point clouds and can align them correctly, while other methods fail at most cases.

6.5.4 Runtime

The runtime of our experiments is mainly divided into three parts: (1) 3D point clouds are converted to STDF volumetric. The mean runtime is 9 ms for each STDF conversion. (2) Descriptors generated by PSTN. The mean runtime is 5.2 ms for each STDF forward pass. (3) Transformation matrix estimated by RANSAC. The mean runtime is 4 seconds for 5000 points registration. Actually, the registration time varies among different thresholds of RANSAC. We set the threshold as 0.04 for same-source point cloud registration and 0.4 for cross-source point cloud registration.

6.6 Conclusion

In this chapter, the focus is to generate a rotation-invariant descriptor in 3D cross-source point clouds. This is a challenging topic. A novel 3D parallel spatial transformation network is designed to generate a rotation-invariant descriptor for cross-source point clouds. Because of the cross-source problems, the keypoints are difficult to be found. To robustly use the structure information, a region-based method is proposed to prepare input data for our network. Then, an efficient and feasible way is proposed to train the network by using the synthetic datasets. The trained network shows the generalization ability to be applied to real cross-source point clouds. The experiments show that the proposed descriptor can robustly generate discriminative descriptor and show high performance in challenging cross-source point cloud registration.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In conclusion, this thesis presents several techniques to address the cross-source point cloud matching problem.

In chapter 3, we present a macro and micro structures concept of cross-source point clouds. The structures are extracted by super voxel based 3D segmentation method. After the structures are extracted, two algorithms are designed to to solve the registration problem of cross-source point clouds. Firstly, the central points of segmentations are assembled into graph and the registration problem is converted into graph matching problem. Node and edge representations are designed to solve the graph matching problem. After the optimal graph matching solution is obtained, RANSAC and ICP is utilized to obtain the refine the matching results and obtain optimal registration results. Compared with previous algorithms, the algorithm provides the first solution to cross-source point cloud registration problem with high accuracy.

In Chapter 4, weak regional affinity and pixel-wise refinement components are proposed based on the macro and micro structures. Then, these two components are integrated into unified tensors and the optimal solution is achieved by applying tensor optimization. Different to previous tensor op-

timization, the transformation matrix is integrated into tensor optimization. Therefore, both the transformation matrix and correspondence are estimated when the solution is obtained. Compared with the previous algorithm, this algorithm both obtains high accuracy and high efficiency.

In chapter 5, Gaussian mixture models are utilized to describe the 3D scene. Then, two cross-source point clouds are recognized as two scaled samples from the Gaussian mixture models. The objective function is to back-project two point clouds into an uniformed GMM. Expectation-Maximization algorithm is utilized to solve the parameters of transformation and GMM. When the objective function reaches convergence, all the parameters are obtained. This algorithm provide a fast and accurate solution for localization of cross-source point clouds. Compared with previous localization algorithms, this algorithm provide accurate registration results. The users can obtain both the position and pose information.

In chapter 6, we present a deep learning method to learn a rotation-invariant descriptor for cross-source point clouds. Firstly, a region-based method is proposed to prepare training dataset for cross-source point clouds. Then, a parallel spatial transformation network is proposed to learn a rotation-invariant descriptor. The neural network firstly transforms the 3D input into a rotation-invariant space, then a deep feature is generated followed by the transformation. A comprehensive experiments have conducted on both same-source and cross-source point clouds. The experimental results show both higher accuracy and higher efficiency than previous methods.

Each chapter (i.e. from Chapter 3 to Chapter 6) of this thesis is supported by at least one published conference papers[1] listed in **List of Publications**. Therefore, what we have done and propose in this thesis is of great significance to cross-source point cloud matching/registration area.

[1]The papers of chapter 3 and 5 are published, one paper of chapter 4 and the paper of chapter 6 is still under review

7.2 Future Work

With the development of 3D sensors, many applications occurs two or more 3D sensors. The alignment of different source of point clouds are necessity. Cross-source point cloud registration/matching can be extended to many tasks. These tasks includes:

(i). **3D map reconstruction(mapping).**

Lidar is known as high accuracy in acquisition of 3D position points. Using these points and registration methods, large-scale 3D map can be reconstructed with high accuracy. It can used in outdoor robotics and drone. The key challenges are 3D registration, 2D-3D matching and large-scale loop closure.

(ii). **High accurate localization (matching).**

GPS is a common device in outdoor localization. However, the accuracy drops largely when there is large building or plants. Using the fusion of Lidar and vision-based system to develop a high accurate system for localization is very exciting topic. It will be used autonomous driving and Augment reality. Based on my research background, we can solve the localization based on matching and registration

(iii). **3D measurement.**

When we capture 2D images of an object, human can estimate the size in real time. How to estimate the size by machine automatically from 2D images is a challenging problem. For indoor 3D measurement, there are many applications in Intelligent Manufacturing. For outdoor 3D measurement, there are many applications in robotics, Photogrammetry and autonomous driving.

Bibliography

Aiger, D., Mitra, N. J. & Cohen-Or, D. (2008), '4-points congruent sets for robust pairwise surface registration', *ACM Transactions on Graphics (TOG)* **27**(3), 85.

Albarelli, A., Rodola, E. & Torsello, A. (2010), Loosely distinctive features for robust surface alignment, *in* 'European Conference on Computer Vision', Springer, pp. 519–532.

Aldoma, A., Tombari, F., Rusu, R. B. & Vincze, M. (2012), Our-cvfh–oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation, *in* 'Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium', Springer, pp. 113–122.

Anzai, Y. (2012), *Pattern recognition and machine learning*, Elsevier.

Belongie, S., Malik, J. & Puzicha, J. (2002), 'Shape matching and object recognition using shape contexts', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(4), 509–522.

Best, P. J. & McKay, N. D. (1992), 'A method for registration of 3-d shapes', *IEEE Transactions on pattern analysis and machine intelligence* **14**(2), 239–256.

Bishop, C. M. (2006), 'Pattern recognition', *Machine Learning* .

Bouaziz, S., Tagliasacchi, A. & Pauly, M. (2013), Sparse iterative closest point, *in* 'Computer graphics forum', Vol. 32, Wiley Online Library, pp. 113–123.

Campbell, D. & Petersson, L. (2016), 'Gogma: Globally-optimal gaussian mixture alignment', *arXiv preprint arXiv:1603.00150* .

Chen, C.-S., Hung, Y.-P. & Cheng, J.-B. (1999), 'Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(11), 1229–1234.

Chen, X., Kundu, K., Zhu, Y., Berneshawi, A. G., Ma, H., Fidler, S. & Urtasun, R. (2015), 3d object proposals for accurate object class detection, *in* 'Advances in Neural Information Processing Systems', pp. 424–432.

Chen, X., Ma, H., Wan, J., Li, B. & Xia, T. (2017), Multi-view 3d object detection network for autonomous driving, *in* 'IEEE CVPR', Vol. 1, p. 3.

Cheng, Z.-Q., Chen, Y., Martin, R. R., Lai, Y.-K. & Wang, A. (2013), 'Supermatching: Feature matching using supersymmetric geometric constraints', *IEEE Transactions on Visualization and Computer graphics* **19**(11), 1885–1894.

Choi, S., Zhou, Q.-Y. & Koltun, V. (2015), Robust reconstruction of indoor scenes, *in* '2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', IEEE, pp. 5556–5565.

Chui, H. & Rangarajan, A. (2000), A new algorithm for non-rigid point matching, *in* 'Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on', Vol. 2, IEEE, pp. 44–51.

Chui, H. & Rangarajan, A. (2003), 'A new point matching algorithm for non-rigid registration', *Computer Vision and Image Understanding* **89**(2), 114–141.

Cleju, I. & Saupe, D. (2007), Stochastic optimization of multiple texture registration using mutual information, *in* 'Joint Pattern Recognition Symposium', Springer, pp. 517–526.

Corsini, M., Cignoni, P. & Scopigno, R. (2012), 'Efficient and flexible sampling with blue noise properties of triangular meshes', *IEEE Transactions on Visualization and Computer Graphics* **18**(6), 914–924.

Corsini, M., Dellepiane, M., Ganovelli, F., Gherardi, R., Fusiello, A. & Scopigno, R. (2013), 'Fully automatic registration of image sets on approximate geometry', *International journal of computer vision* **102**(1-3), 91–111.

Cour, T., Srinivasan, P. & Shi, J. (2007), Balanced graph matching, *in* 'Advances in Neural Information Processing Systems', pp. 313–320.

Curless, B. & Levoy, M. (1996), A volumetric method for building complex models from range images, *in* 'Proceedings of the 23rd annual conference on Computer graphics and interactive techniques', ACM, pp. 303–312.

Deng, Y., Rangarajan, A., Eisenschenk, S. & Vemuri, B. C. (2014), A riemannian framework for matching point clouds represented by the schrodinger distance transform, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 3756–3761.

Diez, Y., Martí, J. & Salvi, J. (2012), 'Hierarchical normal space sampling to speed up point cloud coarse matching', *Pattern Recognition Letters* **33**(16), 2127–2133.

Duchenne, O., Bach, F., Kweon, I.-S. & Ponce, J. (2011), 'A tensor-based algorithm for high-order graph matching', *IEEE transactions on pattern analysis and machine intelligence* **33**(12), 2383–2395.

Elbaz, G., Avraham, T. & Fischer, A. (2017), 3d point cloud registration for localization using a deep neural network auto-encoder, *in* '2017 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR)', IEEE, pp. 2472–2481.

Evangelidis, G. D. & Horaud, R. (2018), 'Joint alignment of multiple point sets with batch and incremental expectation-maximization', *IEEE transactions on pattern analysis and machine intelligence* **40**(6), 1397–1410.

Evangelidis, G. D., Kounades-Bastian, D., Horaud, R. & Psarakis, E. Z. (2014), A generative model for the joint registration of multiple point sets, *in* 'Computer Vision–ECCV 2014', Springer, pp. 109–122.

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

Fukushima, M. (1984), 'A modified frank-wolfe algorithm for solving the traffic assignment problem', *Transportation Research Part B: Methodological* **18**(2), 169–177.

Furukawa, Y., Curless, B., Seitz, S. M. & Szeliski, R. (2010), Towards internet-scale multi-view stereo, *in* 'Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on', IEEE, pp. 1434–1441.

Gelfand, N. & Guibas, L. J. (2004), Shape segmentation using local slippage analysis, *in* 'Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing', ACM, pp. 214–223.

Gelfand, N., Mitra, N. J., Guibas, L. J. & Pottmann, H. (2005), Robust global registration, *in* 'Symposium on geometry processing', Vol. 2, p. 5.

Golub, G. H. & Van Loan, C. F. (2012), *Matrix computations*, Vol. 3, JHU Press.

Guo, K., Zou, D. & Chen, X. (2015), '3d mesh labeling via deep convolutional neural networks', *ACM Transactions on Graphics (TOG)* **35**(1), 3.

Ho, J., Peter, A., Rangarajan, A. & Yang, M.-H. (2009), An algebraic approach to affine registration of point sets, *in* '2009 IEEE 12th International Conference on Computer Vision', IEEE, pp. 1335–1340.

Hontani, H. & Watanabe, W. (2010), Point-based non-rigid surface registration with accuracy estimation, *in* 'Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on', IEEE, pp. 446–452.

Horaud, R., Forbes, F., Yguel, M., Dewaele, G. & Zhang, J. (2011), 'Rigid and articulated point registration with expectation conditional maximization', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(3), 587–602.

Huang, X., Zhang, J., Fan, L., Wu, Q. & Yuan, C. (2016), 'A systematic approach for cross-source point cloud registration by preserving macro and micro structures'. arXiv:1608.05143v1.

Huang, X., Zhang, J., Wu, Q., Fan, L. & Yuan, C. (2016), A coarse-to-fine algorithm for registration in 3d street-view cross-source point clouds, *in* '2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)', pp. 1–6.

Huang, X., Zhang, J., Wu, Q., Fan, L. & Yuan, C. (2017*a*), A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds, Vol. PP, pp. 1–1.

Huang, X., Zhang, J., Wu, Q., Fan, L. & Yuan, C. (2017*b*), 'A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds', *IEEE Transactions on Circuits and Systems for Video Technology* **PP**(99), 1–1.

Huang, X., Zhang, J., Wu, Q., Yuan, C. & Fan, L. (2015), Dense correspondence using non-local daisy forest, *in* 'Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on', IEEE, pp. 1–8.

Huber, D. F. & Hebert, M. (2003), 'Fully automatic registration of multiple 3d data sets', *Image and Vision Computing* **21**(7), 637–650.

Jaderberg, M., Simonyan, K., Zisserman, A. et al. (2015), Spatial transformer networks, *in* 'Advances in Neural Information Processing Systems', pp. 2017–2025.

Ji, M., Gall, J., Zheng, H., Liu, Y. & Fang, L. (2017), Surfacenet: An end-to-end 3d neural network for multiview stereopsis, *in* 'The IEEE International Conference on Computer Vision (ICCV)'.

Jian, B. & Vemuri, B. C. (2011*a*), 'Robust point set registration using gaussian mixture models', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**(8), 1633–1645.

Jian, B. & Vemuri, B. C. (2011*b*), 'Robust point set registration using gaussian mixture models', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**(8), 1633–1645.

Johnson, A. E. (1997), Spin-images: a representation for 3-D surface matching, PhD thesis, Citeseer.

Johnson, A. E. & Hebert, M. (1999*a*), 'Using spin images for efficient object recognition in cluttered 3d scenes', *IEEE Transactions on pattern analysis and machine intelligence* **21**(5), 433–449.

Johnson, A. E. & Hebert, M. (1999*b*), 'Using spin images for efficient object recognition in cluttered 3d scenes', *IEEE Transactions on pattern analysis and machine intelligence* **21**(5), 433–449.

Leordeanu, M. & Hebert, M. (2005), A spectral technique for correspondence problems using pairwise constraints, *in* 'Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1', Vol. 2, IEEE, pp. 1482–1489.

Lin, B., Tamaki, T., Zhao, F., Raytchev, B., Kaneda, K. & Ichii, K. (2014), 'Scale alignment of 3d point clouds with different scales', *Machine Vision and Applications* **25**(8), 1989–2002.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P. & Querido, T. (2007), 'A survey for the quadratic assignment problem', *European journal of operational research* **176**(2), 657–690.

Ma, J., Zhao, J., Tian, J., Tu, Z. & Yuille, A. L. (2013), Robust estimation of nonrigid transformation for point set registration, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 2147–2154.

Ma, J., Zhao, J. & Yuille, A. L. (2016), 'Non-rigid point set registration by preserving global and local structures', *IEEE Transactions on Image Processing* **25**(1), 53–64.

Manferdini, A. M. (2012), A methodology for the promotion of cultural heritage sites through the use of low-cost technologies and procedures, *in* 'Proceedings of the 17th International Conference on 3D Web Technology', ACM, pp. 180–180.

Mellado, N., Aiger, D. & Mitra, N. J. (2014), Super 4pcs fast global pointcloud registration via smart indexing, *in* 'Computer Graphics Forum', Vol. 33, Wiley Online Library, pp. 205–215.

Mellado, N., Dellepiane, M. & Scopigno, R. (2015), 'Relative scale estimation and 3d registration of multi-modal geometry using growing least squares'.

Mellado, N., Dellepiane, M. & Scopigno, R. (2016), 'Relative scale estimation and 3d registration of multi-modal geometry using growing least squares', *IEEE Transactions on Visualization and Computer Graphics* **22**(9), 2160–2173.

Moussa, A. & Elsheimy, N. (2015), 'Automatic registration of approximately leveled point clouds of urban scenes.', *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* pp. 145–150.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., Gool, L. & Purgathofer, W. (2013), A survey of urban reconstruction, *in* 'Computer graphics forum', Vol. 32, Wiley Online Library, pp. 146–177.

Myronenko, A. & Song, X. (2009), 'On the closed-form solution of the rotation matrix arising in computer vision problems', *arXiv preprint arXiv:0904.1613* .

Myronenko, A. & Song, X. (2010), 'Point set registration: Coherent point drift', *IEEE transactions on pattern analysis and machine intelligence* **32**(12), 2262–2275.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S. & Fitzgibbon, A. (2011), Kinectfusion: Real-time dense surface mapping and tracking, *in* 'Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on', IEEE, pp. 127–136.

Nüchter, A., Lingemann, K., Hertzberg, J. & Surmann, H. (2007), '6d slam3d mapping outdoor environments', *Journal of Field Robotics* **24**(8-9), 699–722.

Pandey, G., McBride, J. R., Savarese, S. & Eustice, R. M. (2012), Toward mutual information based automatic registration of 3d point clouds, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems', IEEE, pp. 2698–2704.

Papazov, C. & Burschka, D. (2010), An efficient ransac for 3d object recognition in noisy and occluded scenes, *in* 'Computer Vision–ACCV 2010', Springer, pp. 135–148.

Papazov, C. & Burschka, D. (2011), 'Stochastic global optimization for robust point set registration', *Computer Vision and Image Understanding* **115**(12), 1598–1609.

Papon, J., Abramov, A., Schoeler, M. & Wörgötter, F. (2013), Voxel cloud connectivity segmentation - supervoxels for point clouds, *in* 'Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on', Portland, Oregon.

Peng, F., Wu, Q., Fan, L., Zhang, J., You, Y., Lu, J. & Yang, J.-Y. (2014*a*), Street view cross-sourced point cloud matching and registration, *in* 'Image Processing (ICIP), 2014 IEEE International Conference on', IEEE, pp. 2026–2030.

Peng, F., Wu, Q., Fan, L., Zhang, J., You, Y., Lu, J. & Yang, J.-Y. (2014*b*), Street view cross-sourced point cloud matching and registration, *in* 'Image Processing (ICIP), 2014 IEEE International Conference on', IEEE, pp. 2026–2030.

Peng, F., Wu, Q., Fan, L., Zhang, J., You, Y., Lu, J. & Yang, J. Y. (2014*c*), Street view cross-sourced point cloud matching and registration, *in* '2014 IEEE International Conference on Image Processing (ICIP)', pp. 2026–2030.

Pomerleau, F., Liu, M., Colas, F. & Siegwart, R. (2012), 'Challenging data sets for point cloud registration algorithms', *The International Journal of Robotics Research* **31**(14), 1705–1711.

Qi, C. R., Su, H., Mo, K. & Guibas, L. J. (2017), 'Pointnet: Deep learning on point sets for 3d classification and segmentation', *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* **1**(2), 4.

Regalia, P. A. & Kofidis, E. (2000), The higher-order power method revisited: convergence proofs and effective initialization, *in* 'Acoustics, Speech,

and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on', Vol. 5, IEEE, pp. 2709–2712.

Rodolà, E., Albarelli, A., Bergamasco, F. & Torsello, A. (2013), 'A scale independent selection process for 3d object recognition in cluttered scenes', *International journal of computer vision* **102**(1-3), 129–145.

Rusu, R. B., Blodow, N. & Beetz, M. (2009*a*), Fast point feature histograms (fpfh) for 3d registration, *in* 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', IEEE, pp. 3212–3217.

Rusu, R. B., Blodow, N. & Beetz, M. (2009*b*), Fast point feature histograms (fpfh) for 3d registration, *in* 'Robotics and Automation, 2009. ICRA'09. IEEE International Conference on', IEEE, pp. 3212–3217.

Rusu, R. B. & Cousins, S. (2011), 3d is here: Point cloud library (pcl), *in* 'Robotics and Automation (ICRA), 2011 IEEE International Conference on', IEEE, pp. 1–4.

Sharp, G. C., Lee, S. W. & Wehe, D. K. (2002), Icp registration using invariant features, Vol. 24, IEEE, pp. 90–102.

Shi, X., Ling, H., Hu, W., Xing, J. & Zhang, Y. (2016), Tensor power iteration for multi-graph matching, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5062–5070.

Shi, X., Ling, H., Xing, J. & Hu, W. (2013), Multi-target tracking by rank-1 tensor approximation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 2387–2394.

Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P. & Moreno-Noguer, F. (2015), Discriminative learning of deep convolutional feature point descriptors, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 118–126.

Sinha, T. K., Cash, D. M., Weil, R. J., Galloway, R. L. & Miga, M. I. (2002), Cortical surface registration using texture mapped point clouds and mutual information, *in* 'International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer, pp. 533–540.

Song, S. & Xiao, J. (2016), Deep sliding shapes for amodal 3d object detection in rgb-d images, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 808–816.

Su, H., Maji, S., Kalogerakis, E. & Learned-Miller, E. (2015), Multi-view convolutional neural networks for 3d shape recognition, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 945–953.

Tamaki, T., Tanigawa, S., Ueno, Y., Raytchev, B. & Kaneda, K. (2010), Scale matching of 3d point clouds by finding keyscales with spin images, *in* 'Pattern Recognition (ICPR), 2010 20th International Conference on', IEEE, pp. 3480–3483.

Tombari, F., Salti, S. & Di Stefano, L. (2010*a*), Unique signatures of histograms for local surface description, *in* 'European conference on computer vision', Springer, pp. 356–369.

Tombari, F., Salti, S. & Di Stefano, L. (2010*b*), Unique signatures of histograms for local surface description, *in* 'European Conference on Computer Vision', Springer, pp. 356–369.

Torki, M. & Elgammal, A. M. (2010), Putting local features on a manifold., *in* 'CVPR', Vol. 2, p. 4.

Torsello, A., Rodola, E. & Albarelli, A. (2011), Multiview registration via graph diffusion of dual quaternions, *in* 'Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on', IEEE, pp. 2441–2448.

Wang, D. Z. & Posner, I. (2015), 'Voting for voting in online point cloud object detection', *Proceedings of the Robotics: Science and Systems, Rome, Italy* **1317**.

Wang, F., Vemuri, B. C., Rangarajan, A. & Eisenschenk, S. J. (2008), 'Simultaneous nonrigid registration of multiple point sets and atlas construction', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **30**(11), 2011–2022.

Wohlkinger, W. & Vincze, M. (2011), Ensemble of shape functions for 3d object classification, *in* 'Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on', IEEE, pp. 2987–2992.

Wu, C. (2011), 'Visualsfm: A visual structure from motion system'.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. & Xiao, J. (2015), 3d shapenets: A deep representation for volumetric shapes, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1912–1920.

Yang, J., Li, H. & Jia, Y. (2013), Go-icp: Solving 3d registration efficiently and globally optimally, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 1457–1464.

Yang, Y., Feng, C., Shen, Y. & Tian, D. (2018), Foldingnet: Point cloud auto-encoder via deep grid deformation, *in* 'Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)', Vol. 3.

Yi, K. M., Trulls, E., Lepetit, V. & Fua, P. (2016), Lift: Learned invariant feature transform, *in* 'European Conference on Computer Vision', Springer, pp. 467–483.

Zaslavskiy, M., Bach, F. & Vert, J.-P. (2009), 'A path following algorithm for the graph matching problem', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **31**(12), 2227–2242.

Zbontar, J. & LeCun, Y. (2016), 'Stereo matching by training a convolutional neural network to compare image patches', *Journal of Machine Learning Research* **17**(1-32), 2.

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J. & Funkhouser, T. (2017), 3dmatch: Learning local geometric descriptors from rgb-d reconstructions, *in* 'Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on', IEEE, pp. 199–208.

Zeng, Y., Wang, C., Gu, X., Samaras, D. & Paragios, N. (2016), 'Higher-order graph principles towards non-rigid surface registration', *IEEE transactions on pattern analysis and machine intelligence* **38**(12), 2416–2429.

Zhou, F. & De la Torre, F. (2013*a*), Deformable graph matching, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 2922–2929.

Zhou, F. & De la Torre, F. (2013*b*), Deformable graph matching, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 2922–2929.

Zhou, F. & De la Torre, F. (2016), 'Factorized graph matching', *IEEE transactions on pattern analysis and machine intelligence* **38**(9), 1774–1789.

Zhou, Q.-Y., Park, J. & Koltun, V. (2016), Fast global registration, *in* 'European Conference on Computer Vision', Springer, pp. 766–782.