

POINT SET REGISTRATION NETWORKS

DISSERTATION

Submitted in Partial Fulfillment

of the Requirements for the

Degree of

DOCTOR OF PHILOSOPHY (Mathematics)

at the

NEW YORK UNIVERSITY

TANDON SCHOOL OF ENGINEERING

by

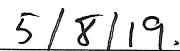
Lingjing Wang

May 2019

Approved:



Department Chair Signature



Date

ProQuest Number: 13886429

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 13886429

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

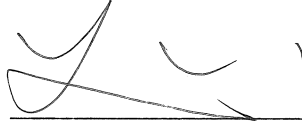
All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

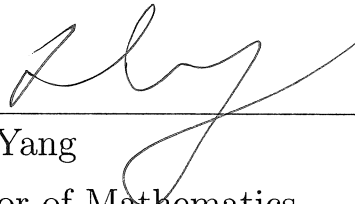
Approved by the Guidance Committee :

Major : Mathematics



Yi Fang

Assistant Professor of
Electrical And Computer Engineering




Deane Yang

Professor of Mathematics



Gaoyong Zhang

Professor of Mathematics



Edward Wong

Associate Professor of
Computer Science And Engineering

Microfilm or copies of this dissertation may be obtained from:

UMI Dissertation Publishing

ProQuest CSA

789 E. Eisenhower Parkway

P.O. Box 1346

Ann Arbor, MI 48106-1346

Vita

Lingjing Wang was born in Hunan, China.

He received his Bachelor degree from Moscow State University in 2011 in Russia and his Master degree from Georgetown University in 2012 in United States.

He has been studying for Ph.D in Mathematics at New York University from 2013 to present.

I dedicate this thesis to my wife and parents.

Acknowledgements

I would like to thank my adviser Prof. Yi Fang. I have joined MMVC lab in 2016 and closely worked with him since then. He taught me everything about becoming a creative researcher in the computer science field hand by hand. More importantly he taught me how to consistently hold an ambitious and active attitude towards any my academic works and non-academic life as he always dose. I am in great debt to him for all these lessons he taught me.

I would like to thank my committee members Prof. Gaoyong Zhang, Prof. Deane Yang and Prof. Edward Wong for your time and patience to read this work. I would like to thank the Tandon Math department (now Courant Institute) which gave me the chance to pursue this PhD degree, especially to Prof. Gaoyong Zhang, Prof. Erwin Lutwak who taught me fundamental mathematical tools to allow me leverage a mathematical thinking all the time, Prof. Deane Yang who always supported me whenever I needed his help. I would like to thank you again.

I would like to thank my fantastic teammates from MMVC, especially Xiang Li and Jianchun Chen. This dissertation involves all our team works and I really enjoyed the time we worked together. Moreover, I would like to thank all my friends, especially Jifei Wang, Yizhang Chen, Han Han, Manzhi Tan, my brothers Han Tu and Xiaoran Mo, for your consistent encouragement for all these years. In the end, I would like to thank my dear wife Sheng Long and my parents who support me all the time for pursuing what I am interested.

ABSTRACT

POINT SET REGISTRATION NETWORKS

by

Lingjing Wang

Advisors: Prof. Yi Fang, Ph.D.

Submitted in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy (Mathematics)

May 2019

Point set registration is defined as a process to determine the spatial transformation from the source point set to the target one. Existing methods often iteratively search for the optimal geometric transformation to register a given pair of point sets, driven by minimizing a predefined alignment loss function. In contrast, we firstly focus on learning neural network-based structure for point set registration, which allow us to actively learn the registration pattern from a training dataset, and then predict the desired geometric transformation to align a pair of point sets. Consequently, the learning-based structure tends to transfer the learned knowledge (i.e. registration pattern) from registering training pairs to testing ones without additional iterative optimization.

In this work, we provide model-based (PR-Net) and model-free (CPD-Net) solutions for modeling desired transformation in a learning framework. In comparison to the model-based modeling approach which targets to predict the pa-

rameters in a Thin Plate Spline model, the model-free structure can directly learn a continuous displacement field to align the source point set with the target one without further penalization on it. The model-free learning structure dramatically improves the registration performance from the model-based, especially for pairs with high deformation levels. Moreover, the model-free learning structure can be easily extended to 3D or nD cases. Furthermore, based on it, we propose the first model-free learning structure (MF-GeoNet) for image matching, which achieves the state-of-the-art 2D images registration performance.

Our approaches achieve robust and superior performance for registration of point sets, even in presence of noise, outliers, and missing points, but requires much less time for registering large number of pairs in comparison with classical iterative methods. More importantly, for a new unseen pair of point sets, we are able to directly predict the desired transformation using the learned model without repetitive iterative optimization routine. Our methods can be extended to register 2D images, 3D shapes and be applied in medical image domain as well.

Contents

Dedication	v
Acknowledgements	vi
Abstract	vii
List of Figures	xix
List of Tables	xx
1 Introduction	1
1.1 Background and Motivation	1
1.2 Our solutions and contributions	5
1.3 Related Works	9
1.3.1 Iterative registration methods.	9
1.3.2 Learning-based registration methods.	15
2 Learning-based Registration Networks	17
2.1 Problem statement	23
2.2 Methods	24
2.2.1 PR-Net	24
2.2.1.1 Learning shape descriptor tensor	24
2.2.1.2 Shape correlation tensor	26

2.2.1.3	Shape transformation prediction	27
2.2.1.4	From statistical alignment to loss functions	28
2.2.1.5	Model settings	30
2.2.2	CPD-Net	31
2.2.2.1	Learning shape descriptor	31
2.2.2.2	Coherent PointMorph architecture	32
2.2.2.3	Loss functions	35
2.2.2.4	Settings of CPD-Net	35
2.3	Experimental Results	36
2.3.1	PR-Net	36
2.3.1.1	Dataset preparation	36
2.3.1.2	Comparison to Non-learning based Approach	37
2.3.1.3	Robust to Geometric Deformation	39
2.3.1.4	Robust to Data Noise	42
2.3.1.5	Results on Data Variety	45
2.3.2	CPD-Net	49
2.3.2.1	Experimental Dataset	50
2.3.2.2	2D non-rigid point set registration	51
2.3.2.3	3D non-rigid point set registration	56
2.3.2.4	Resistance to Noise	59
2.3.2.5	Comparison to CPD	63
3	Learning-based Point Correspondence Networks	65
3.1	Methods	68
3.1.1	Problem statement	68
3.1.2	Reconstruction pipeline	70

3.1.3	From reconstructed results to unsupervised correspondences.	73
3.2	Experimental Results	74
3.2.1	Dataset and implementation Details	74
3.2.2	Illustration of Motion-Driven Process	75
3.2.3	Generalization ability on test dataset	77
3.2.4	Linear shape interpolation	78
3.2.5	Robust to deformation, noise, and missing points	78
3.2.6	Rotation discussion	82
3.2.7	Correspondence for 3D point sets	82
3.3	Discussion	84
4	Application In 2D Image Matching	85
4.1	Methods	91
4.1.1	Image-Matching Feature Learning	91
4.1.2	Geometric Transformation Network	94
4.1.3	Loss Function	95
4.1.4	Implementation Details	96
4.2	Experimental Results	97
4.2.1	Experiment Setup	97
4.2.2	Comparisons on synthetic image matching	99
4.2.3	Comparisons on real image point correspondence matching .	103
4.2.3.1	Experiment One.	104
4.2.3.2	Experiment Two.	105
4.2.4	Qualitative results on real image corresponding estimation .	105
4.3	Discussion	106

List of Figures

- 1.1 Point set registration task. The point set registration is mathematically defined as a process to determine the spatial geometric transformations (i.e. rigid and non-rigid transformation) that can optimally register the source point set to the target one. 2
- 2.1 PR-Net pipeline. The proposed PR-Net includes three parts: learning shape descriptor tensor (SCT), learning correlation tensor, and shape transformation prediction. For a pair of source point set \mathbf{S}_i and target point set \mathbf{G}_j , we first generate two reference grids and map points of source and target point sets on them as two shape descriptor tensor \mathbf{F}_s and \mathbf{F}_g . We define the shape correlation tensor \mathbf{C} between the source and target shape descriptor tensors. By leveraging 2D-CNN, we learn the desired parameters θ of transformation T_θ based on the shape correlation tensor. The learned optimal model transforms source point set to be statistically aligned with the target point set. 20

2.2	Our pipeline. The proposed structure includes three parts: learning shape descriptor, coherent PointMorph, and the alignment Loss. For a pair of source point set \mathbf{S}_i and target point set \mathbf{G}_j , we firstly leverage MLPs to learn two global descriptors $\mathbf{L}_{\mathbf{S}_i}$ and $\mathbf{L}_{\mathbf{G}_j}$. We then concatenate these two descriptors to each coordinate $\{x_k\}_{k=1,2,\dots,m}$ of source points as the input ($[\mathbf{x}_k, \mathbf{L}_{\mathbf{S}_i}, \mathbf{L}_{\mathbf{G}_j}]$) for PointMorph structure. We further use MLPs to learn the drifts for each source point. Finally we move the source point set by our predicted drifts and define the alignment loss function between target and transformed source point sets for back-propagation.	22
2.3	The schema of learning shape descriptor tensor process.	25
2.4	The schema of formulating correlation tensor process.	27
2.5	The schema of learning shape descriptor tensor process.	32
2.6	The schema of learning coherent PointMorph process.	34
2.7	Testing results for 2D fish shape point set registration at different deformation levels. The deformation level increases from 0.3 to 1.5 from left to right. The presented shapes are randomly selected from same testing batch. The blue shapes are source point sets and the red shapes are target point sets. Please zoom-in for better visualization.	41

2.8	Testing results for 2D fish shape point set registration at deformation level 0.5 in presence of various noise. (A) Performance in presence of Data Incompleteness (D.I.) noise. (B) Performance in presence of Point Drift (P.D.) noise. (C) Performance in presence of Data Outlier (D.O.) noise. Blue shapes are source point sets and red ones are target point sets. Please zoom-in for better visualization.	44
2.9	Testing performance for skull, hand and human skeleton shapes. Blue shapes are source point sets and red ones are target point sets. Please zoom-in for better visualization. The corresponding C.D. for each input and output pair is presented below it.	47
2.10	Testing registration performance for 3D face and cat point sets. The blue shapes are source shapes and the red shapes are target ones. We plot the mesh of shapes for better visualization.	48
2.11	The qualitative registration result for Fish shape at different deformation level. The blue shape is target point set. The red shape is source point set. The black lines are predicted coherent drifts for source point set. Please zoom-in for better visualization.	52
2.12	The testing qualitative registration results for Fish shape at different deformation level. The blue shape is target point set. The red shape is source point set. Please zoom-in for better visualization.	53
2.13	The C.D. between source and target point sets, pre (blue line) and post (red line) registration.	55

2.14	Registration examples for Mushroom, Fork and Face shapes. The blue shape represents target and the red shape represents source point set. The corresponding C.D. score is listed underneath the registered point sets.	56
2.15	The charts show C.D. between source and target point sets, pre (blue line) and post (red line) registration in left. Selected qualitative registration results are demonstrated in right. The red points represents the source points and the blue ones represent the target points. The black lines represent the predicted drifts for source point set. Please zoom-in for better visualization.	57
2.16	The testing qualitative registration results for 3D shapes at different deformation level. The red points represents the source points and the blue ones represent the target points. Please zoom-in for better visualization.	58
2.17	The charts of C.D. between transformed source point set and target one in presence of different level of G.D. noise, P.O. noise and D.I.noise are shown in left. The selected qualitative results are demonstrated in right. The red shape represents the source point set and the blue one represents the target point set. Please zoom-in for better visualization.	61
3.1	Illustration of our unsupervised point correspondence. Our model drifts all landmark points of a template circle to match the corresponding positions of target shapes.	66

3.2	The pipeline of proposed PC-Net model. The pipeline mainly includes four parts. The first part is an encoder to learn the “global shape descriptor” from a point set that captures essential global and deformation-insensitive geometric properties. The second part is forming “shape morphing initiator” and the third component is “Motion-driven Embedding” for reconstruction. The fourth component is “Point Correspondence Mapping” to map the correspondence of reconstructed landmarks back to the original point sets.	69
3.3	Point Correspondence procedure with the Hungarian matching algorithm. The Left part shows the input point sets, and the right part shows their reconstructed shapes from a template circle.	73
3.4	Illustration of our Motion-driven embedding process. Landmark points are numbered with color and the blue fishes in last column are input shapes.	76
3.5	Correspondence performance on test set.	78
3.6	Examples of linear interpolation. ‘GT1’ and ‘GT2’ show two input shapes, ‘Rec1’ and ‘Rec1’ show their reconstructed shapes, and the middle columns show the interpolated shapes. Our model got continuous shape reconstruction using interpolated feature vectors. Note that the landmark points on each shape are corresponded to the landmark points on the other shapes.	79
3.7	Robustness test. (a) Correspondence quality at different deformation levels. (b) Correspondences quality at different noise level. (c) Correspondence quality with different number of missing points.	79

3.8	Examples of point correspondence at (a) different deformation level, (b) different noise level, and (c) different number of missing points. The top rows in (a)-(c) show the reference shapes, the middle rows in (a) show the reconstructed shapes, and the bottom rows in (a)-(c) show the predicted shapes with correspondences. We annotate some corner points in the reference shapes (top row) and find out their matching points in the target shapes (bottom row); whereas numbers on the shapes of the middle rows in (a) indicate the indexes of landmark points. The ‘red triangle’ indicates ground truth point with a point label ‘18’, while the ‘green cross’ indicates its corresponding point.	81
3.9	Selected examples of arbitrary rotations. Numbers in the top indicate the rotation angles. The first row show the reconstructed shapes, the second row shows the predicted shape correspondence. Even though our model fails to match the correspondence points (see second rows for illustration), it maintains a good correspondence for each corner (see the first row for illustration). The color pattern in each reconstructed shape remains the same order as our input circle.	83
3.10	Selected examples of 3D shape correspondence on FAUST dataset. The first shape is used as a reference. First row shows successful correspondence examples, second row shows failure examples. Note that our results are generated in an unsupervised way.	83
4.1	Comparison between geometric transformation model (A) and our proposed model-free geometric transformation network (B).	87

4.2 Main pipeline: Our proposed architecture comprises of two parts - Image-Matching Feature Learning and Geometric Transformation Network. For a given pair of input images, we first extract the Image Description Tensor f_A and f_B through convolutional neural network. We next generate Image Correlation Tensor C_{AB} between the two Image Description Tensors. Furthermore, we embed this Image Correlation Tensor into a latent feature d_{AB} which represents Image-Matching Feature. Finally, we pass this feature through Geometric Transformation Network comprising of MLPs which predicts desired displacement field and consequently transform points in the source plane to target plane. We minimize the Mean Square Error Loss between the corresponding points between predicted point set and the ground truth point set. 89

4.3 Visualization of hidden features after each MLP layer. The feature of each point is reduced to 2 dimension by Principal Component Analysis (PCA). The last figure is the offset prediction of a set of grid point, which is also called deformation field. All MLPs and input feature are randomly initialized. 92

4.4 Quantitative comparison between our model and CNNGeo [1] in synthesis testing datasets with different number of controlling point (A) and visualization of images transformed by TPS with different controlling point. Subplot (B), (C), (D) and (E) are from TPS transformation with 3×3 , 4×4 , 5×5 , 6×6 controlling points respectively. 100

4.5	Qualitative comparison between our model and CNNGeo [1] in dataset synthesized by 6×6 controlling points TPS with deformation level equals to 0.2. Red mesh denotes the displacement field.	101
4.6	Qualitative results and comparisons on Proposal Flow dataset [2]. The selected pictures from 3 categories (motorbike, car and winebottle) are of complex transformation.	103

List of Tables

2.1	Performance comparison with CPD for registering $10k$ pairs of point sets at deformation level 0.5.	38
2.2	Quantitative testing performance for 2D fish shape point set registration at different deformation level (Deform. Level)	40
2.3	Quantitative testing performance for 2D fish shape point set registration at different deformation level 0.5 in presence of various noise such as Point Drift (P.D) noise, Data Outlier (D.O.) noise, and Data Incompleteness (D.I.) noise.	43
2.4	Quantitative testing performance for skull, hand, and skeleton 2D shapes at different deformation level from 0.3 to 1.0.	46
2.5	Quantitative testing performance for 2D point set registration.	55
2.6	Performance and Time comparison with CPD.	63
4.1	Comparison of PCK with our baseline methods in full Proposal Flow dataset [2]. Learning-based methods are trained on synthetic dataset. The settings of our four models are described in Section 4.3.1.	102
4.2	Comparison of PCK with our baseline methods in Proposal Flow dataset [2] under K-fold setting. Details are described in Section 4.3.2	102

Chapter 1

Introduction

In the introduction chapter, we firstly introduce the background and our motivation for this thesis in section 1.1. In section 1.2, we introduce several important historical researches which are mostly related to our work. Section 1.3 lists the contributions of this thesis to our research community.

1.1 Background and Motivation

Over past decades, point set matching and registration is one of the most important computer vision tasks [3, 4, 5, 6, 7, 8, 9, 10], serving a widespread applications such as stereo matching, medical image registration, large-scale 3D reconstruction, 3D point cloud matching, semantic segmentation and so on [11, 12, 13, 14, 15, 16, 17, 18]. The point set registration is mathematically defined as a process to determine the spatial geometric transformations (i.e. rigid and non-rigid transformation) that can optimally register the source point set to the target one as shown in Figure 1.1. The desired registration algorithm can find both rigid (i.e. rotation, reflection, and shifting) and non-rigid (i.e. dilation and

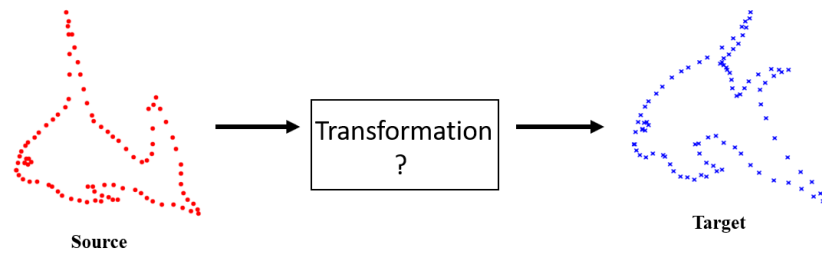


Figure 1.1: Point set registration task. The point set registration is mathematically defined as a process to determine the spatial geometric transformations (i.e. rigid and non-rigid transformation) that can optimally register the source point set to the target one.

stretching) transformations, as well as being robust to outliers, Gaussian point drift, data incompleteness and so on.

To formulate the problem of point set registration, existing methods [8] often iteratively search the optimal geometric transformation to register two sets of points, driven by minimizing a predefined alignment loss function. The alignment loss is usually pre-defined as a certain type of distance metric (e.g. Euclidean distance loss) between the transformed source point set and the target one. Previous efforts [8, 19] have achieved great success in point set registration through the development of a variety of optimization algorithms and distance metrics as summarized in [19]. However these methods are often not designed to handle the real-time point set registration or to deal with a large volume dataset. This limitation is mainly contributed by the fact that, for each given pair of point sets, the iterative method needs to start over a new iterative optimization process even for the trivial similar cases. This observation suggests that the existing efforts are mainly concentrated on the stand-alone development of the optimization strategies rather than the techniques to smartly transferring the registration pattern acquired from

aligning one pair to another. This triggers the motivation to develop our proposed learning-based registration networks with the hope to actively learn the registration pattern from a set of training data, consequently, to adaptively utilize that knowledge to directly predict the geometric transformation for a new pair of unseen point sets.

Different from image data with a regular grid, point cloud data is often recorded in an irregular and disordered non-grid format. Learning the point set registration requires the deep neural networks to be applicable to irregular and non-grid point cloud data. In addition, unlike the image containing rich texture and color information, the point cloud is solely represented with geometric information (i.e. coordinates, curvature, normal). This suggests that a learning-based solution for point set registration needs to address two main technical challenges: 1) robust learning of both local and global geometric feature from point cloud set and 2) robust learning of the transformation from well-defined correlation measure between pairwise geometric feature sets.

To clarify this problem, we firstly introduce some notations and we will keep use them for the rest of our paper. Assume that we have a pair of source $\mathbf{S}_i \subset \mathbb{R}^N$ and target $\mathbf{G}_j \subset \mathbb{R}^N$ point sets for registration. $N = 2$ or $N = 3$ here. In general, assuming that there exists a transformation $\phi : \mathbb{R}^N \rightarrow \mathbb{R}^N$. Iterative methods usually define the optimization problem as:

$$\phi^{\text{optimal}} = \underset{\phi}{\operatorname{argmin}} \mathcal{L}(\mathbf{S}_i, \mathbf{G}_j, \phi), \quad (1.1)$$

where

$$\mathcal{L}(\mathbf{S}_i, \mathbf{G}_j, \phi) = \mathcal{L}_{sim}(\mathbf{G}_i, \phi(\mathbf{S}_j)) + \lambda \mathcal{L}_{smooth}(\phi), \quad (1.2)$$

Here the function \mathcal{L}_{sim} represents a similarity metric between transformed source point set $\phi(\mathbf{S}_j)$ and target point set $(\mathbf{G}_i$. Typical similarity function can be L_2 norm or correlation-based metric. Moreover some previous models treat target and source point sets as two densities by GMM. Therefore a distance to measure the difference of two densities can be applied as well. Some works treat one point set as data to fit the other point set as a distribution. Therefore the similarity measure can be a likelihood function. the function \mathcal{L}_{smooth} represents a penalization term to enforce the smooth deformation. For registration field ϕ , we can penalize its spatial gradients. For a motion ϕ , we can penalize its velocity field. In comparison, learning-based models do not require iterative optimization process for each pair of source and target point sets independently but optimize parameters of our network using a dataset of pairs of point sets. After training process, our model is able to directly predict the motion of source point set so that it can be statistically aligned with the target point sets for a new pair from testing dataset.

1.2 Our solutions and contributions

The proposed PR-Net[20] investigates two major research problems: 1) the design of the techniques for point cloud learning by introducing a novel reference operator to enable formulating the correlation measure on arbitrary-structured data, and 2) the development of learning paradigm for the geometric transformation learning from pairwise feature sets. As a result, PR-Net is capable of handling the real-time point set registration or a large volume datasets with a similar pattern. To better understand the point set registrations, we briefly review related works as follows. However, PR-Net has its limitation with a model-based transformation as its target. It is not obvious to define an appropriate geometric transformation to transform source point set to the target one. The PR-Net provides the shape descriptor tensor and correlation tensor for the solution of feature learning, and uses the thin plate spline to model the geometric transformation. Though PR-Net is capable of learning the point registration, there are still some challenges that are left to be addressed. Firstly, our current PR-Net indirectly uses the regular grids to assist with the shape feature learning. A continuous operator, which can directly be applied on point for feature learning, would be more applicable for point registration. Secondly, PR-Net uses the TPS to model the geometric transformation. Though it predicts impressive registration performance for shapes with moderate deformation, the unsatisfactory performance for shapes with large deformation motivates us to study a model-free geometric transformation (e.g. the displacement field).

We introduce our next learning-based network PC-Net for unsupervised point correspondence. In contrast to the registration task, point sets correspondence concerns with the establishment of point-wise correspondence for a group of 2D or

3D point sets with similar shape description. Existing methods often iteratively search for the optimal point-wise correspondence assignment for two sets of points, driven by maximizing the similarity between two sets of explicitly designed point features or by determining the parametric transformation for the best alignment between two point sets. Without depending on the explicit definitions of point features or transformation, our paper introduces a novel point correspondence neural networks (PC-Net) that is able to learn and predict the point correspondence among the populations of a specific object (e.g. fish, human, chair, etc) in an unsupervised manner.

Considering the limitation of PR-Net, we further provide a model-free learning-based network CPD-Net for learning point set registration. This chapter presents a novel method, named coherent point drift networks (CPD-Net), for unsupervised learning of geometric transformation towards real-time non-rigid point set registration. In contrast to PR-Net, which learns the parameters of a specific parametric transformation, CPD-Net can learn a model-free displacement field function to estimate geometric transformation from a training dataset, consequently, to predict the desired geometric transformation for the alignment of previously unseen pairs. CPD-Net leverages the power of deep neural network to fit an arbitrary function, that adaptively accommodates different levels of complexity of the desired geometric transformation. Particularly, CPD-Net is proved with a theoretical guarantee to learn a continuous displacement vector function that could further avoid imposing additional parametric smoothness constraint as in previous works.

We demonstrate an application of continuous registration field-based learning network for geometric matching problem. Recent efforts introduce convolutional neural network to learn a geometric model (i.e. an affine or thin-plate spline

transformation) for image matching and determine correspondences between two images. The incapability of a geometric model in estimating a high complexity parametric transform limits their use in applications to coarse image alignment/matching. This paper presents a novel approach to learn a model-free geometric transformation to estimate a continuous smooth displacement field and identify two images with a significant geometric deformation. In contrast to model-based method, our proposed method, named Model-Free Geometric Transformation Networks (MF-GeoNet), can learn displacement vector function to estimate geometric transformation from a training dataset. MF-GeoNet is trained to have robust generalization ability to directly predict the desired geometric transformation to identify the correspondence between unseen new pair of images.

We summarize the main contributions of this thesis work as follows:

- In the chapter two, we propose a first learning-based point set registration paradigm PR-Net which learns registration patterns from training data, consequently, to adaptively utilize that knowledge to directly predict the geometric transformation for aligning a new pair of point sets, without the necessity to start over a new iterative search process. Futhurmore, the model-free network CPD-Net can be more flexible to accommodate different levels of complexity of the target geometric transformation for best aligning the pair of point sets and can be easily extend to 3D or nD dataset in comparison with PR-Net. It theoretically guarantees the continuity of predicted displacement field as geometric transformation, which naturally eliminate the necessity to impose a parametric hand-crafted smoothness constraint. The CPD-Net is free of specific geometric model selection for modeling the desired transformation, which avoids the potential mismatch between the transformation

described by specific adopted models and the actual transformation required for point set registration.

- In the chapter three, we introduce a novel learning-based point correspondence paradigm PC-Net which can establish the point correspondence among two or more groups of point sets. With generalization ability, PC-Net is able to directly predict the point correspondence on the testing dataset without a new iterative searching process.
- In the chapter four, we propose MF-GeoNet as a model-free geometric transformation method which does not require model selection procedure. Consequently, we avoid the critical mismatching problem of selected transformation model and actual desired geometric transformation between image pair. Our proposed Geometric Transformation Network is theoretically guaranteed to produce a spatially continuous displacement field. With this property, we avoid imposing additional penalization term on displacement field as smoothness constraint.

1.3 Related Works

1.3.1 Iterative registration methods.

Current mainstream point set registration methods focus on the development of optimization algorithms to estimate the rigid or non-rigid geometric transformations in an iterative routine. With the assumption that a pair of point sets are related by a rigid transformation, a registration approach is to estimate the best translation and rotation parameters in the iterative search routine aiming to minimize a distance metric between two sets of points. One of the most popular methods for rigid registration, the Iterative Closest Point (ICP) algorithm [13], was proposed to handle point set registration with least-squares estimation of transformation parameters. ICP starts with an initial estimation of rigid transformation, followed by iteratively refining the transformation by alternately choosing corresponding points from the point sets as estimate transformation parameters. The ICP algorithm is reported to be vulnerable to the selection of corresponding points for initial transformation estimation, and also incapable of dealing with non-rigid transformation. To accommodate the deformation (e.g. morphing, articulation) between a pair of point sets, many efforts were spent in the development of algorithms to address the challenges of a non-rigid transformation.

Thin plate splines (TPS)[21] is a widely used spline-based technique for modeling non-rigid transformation, which is introduced to geometric design by Duchon. TPS can be regarded as an important special case of a polyharmonic spline. Assume that there is a mapping function $f(x)$, which transform the source point sets to the target one. $f(x)$ can be fitted using a set of corresponding points collected from source and target domain. Let the collected corresponding points $\{y_i\}$

collected from target point sets domain and $\{x_i\}$ collected from source point sets domain, $x_i, y_i \in \mathbb{R}^2$. Our task is to minimize the following energy function:

$$E_{\text{tps}}(f) = \sum_{i=1}^K \|y_i - f(x_i)\|^2 \quad (1.3)$$

By introducing a penalization term with parameter λ , we balance the variant smoothness with the goodness of fit. Therefore, we rewrite our target energy function as follows:

$$E_{\text{tps,smooth}}(f) = \sum_{i=1}^K \|y_i - f(x_i)\|^2 + \lambda \iint \left[\left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 \right] dx_1 dx_2 \quad (1.4)$$

For a set of controlling points $\{c_i\}$, we define the parametric mapping function as follows:

$$f(x) = \sum_{i=1}^K w_i \varphi(\|x - c_i\|) \quad (1.5)$$

where $\|*\|$ represents common Euclidean distance and $\{w_i\}$ are coefficients to optimize. $\varphi(r) = r^2 \log r$ is used here as the radial basis kernel.

TPS provides an useful and applicable parametric non-rigid transformation modeling tool. In practice, once we have the two sets of corresponding points from source and target domain, we can fit the non-rigid mapping function. In

other words, the corresponding points in target domain can be regarded as a set of parameters to define the desired function and in PR-Net, we focus on learning this parameters set. Based on TPS, Chui and Rangarajan [22] further proposed a robust method to model non-rigid transformation. Following RPM, the results of TPS parametrization of the target transformation can be extended as the TPS-RPM method. The authors showed that TPS-RPM can be equivalent to EM for GMM. They proposed TPS-RSM algorithm with penalization on second order derivatives to optimize the parameters of the desired transformation.

In addition, Myronenko et al. [3] proposed non-parametric coherence point drift (CPD) algorithm which leverages Gaussian mixture likelihood and penalizes derivatives of all orders of the velocity field to enforce velocity coherence so that centroids of source point set move coherently to target point set. We summarize CPD here as our baseline model for comparison. Let source point set \mathcal{M} be the centroids of a Gaussian mixture model (GMM) and the target point set \mathcal{N} be a data. We assume the optimal statistical alignment when we can maximize the probability of GMM by fitting all the data points from set \mathcal{N} . We force the GMM centroids move as a group while it can preserve topological structure. We use expectation maximization algorithm for the optimization task.

We assume that there be M points in \mathcal{M} and M points in \mathcal{N} . The probability density function for a point s is:

$$p(s) = \sum_{i=1}^{M+1} P(i)p(s|i) \quad (1.6)$$

where $p(s|i)$ is the Gaussian distribution with center $m_i \in \mathcal{M}$. Therefore we have:

$$p(s|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{\|s - m_i\|^2}{2\sigma^2}\right) \quad (1.7)$$

where $P(i) = \frac{1}{M}$ and it is equal for all components of GMM. A parameter $w \in [0, 1]$ is further introduced to balance the uniform distribution with $p(s|i)$ for cases in presence of noise. And the mixture model can be rewritten as:

$$p(s) = w\frac{1}{N} + (1 - w) \sum_{i=1}^M \frac{1}{M} p(s|i) \quad (1.8)$$

From here, the optimization target is formed as the negative log-likelihood function of the GMM which are re-parametrized by a set of parameters θ .

$$E(\theta, \sigma^2) = - \sum_{j=1}^N \log \sum_{i=1}^{M+1} P(i)p(s_j|i) \quad (1.9)$$

The correspondence probability can be defined as the posterior probability of the GMM centroid (m_i from source point set) given the data point (s_j from target point set).

$$P(i|s_j) = \frac{P(i)p(s_j|i)}{p(s_j)} \quad (1.10)$$

According to the Bayes' theorem, the E-step is to compute the posterior "old" probability distribution $P^{\text{old}}(i, s_j)$ and the M step is the optimize the "new" dis-

tribution by minimizing the following cost function:

$$\text{cost} = - \sum_{j=1}^N \sum_{i=1}^{M+1} P^{\text{old}}(i|s_j) \log(P^{\text{new}}(i)p^{\text{new}}(s_j|i)) \quad (1.11)$$

Ignoring all the constants which don't affect the optimization task, the cost function can be rewritten as:

$$\text{cost}(\theta, \sigma^2) = \frac{1}{2\sigma^2} \sum_{j=1}^N \sum_{i=1}^{M+1} P^{\text{old}}(i|s_j) \|s_j - T(m_i, \theta)\|^2 + \frac{N_{\mathbf{P}}D}{2} \log \sigma^2 \quad (1.12)$$

where

$$N_{\mathbf{P}} = \sum_{j=0}^N \sum_{i=0}^M P^{\text{old}}(i|s_j) \leq N \quad (1.13)$$

with $N = N_{\mathbf{P}}$ if $w = 0$. D is the dimension of points. And we have the P^{old} from previous parameters is:

$$P^{\text{old}}(i|s_j) = \frac{\exp\left(-\frac{1}{2\sigma^{\text{old}2}} \|s_j - T(m_i, \theta^{\text{old}})\|^2\right)}{\sum_{k=1}^M \exp\left(-\frac{1}{2\sigma^{\text{old}2}} \|s_j - T(m_k, \theta^{\text{old}})\|^2\right) + (2\pi\sigma^2)^{\frac{D}{2}} \frac{w}{1-w} \frac{M}{N}} \quad (1.14)$$

For the furthure step, the displacement function v and the transformation T is defined as:

$$T(Y, v) = Y + v(Y) \quad (1.15)$$

CPD algorithm leverages the Motion Coherence Theory (MCT) [23], which states that points close to on another tend to move coherently, and thus, the displacement function between the point sets should be smooth. A norm of v in the Hilbert space \mathbb{H}^m is defined as:

$$\|v\|_{\mathbb{H}^m}^2 = \int_{\mathbb{R}} \sum_{k=0}^m \left\| \frac{\partial^k v}{\partial x^k} \right\|^2 dx \quad (1.16)$$

And this norm can be alternatively defined in the Reproducing Kernel Hilbert Space [23] as:

$$\|v\|_{\mathbb{H}^m}^2 = \int_{\mathbb{R}^D} \frac{|\tilde{v}(s)|^2}{\tilde{G}(s)} ds \quad (1.17)$$

where G is a unique kernel function associated with RKHS with \tilde{G} is its Fourier transform. \tilde{v} is the Fourier transform of the function v and s is a frequency domain variable. In CPD, the regularization term is chosen according to 1.17:

$$\phi(v) = \int_{\mathbb{R}^D} \frac{|\tilde{v}(s)|^2}{\tilde{G}(s)} ds \quad (1.18)$$

where G is a Gaussian.

They reported that their algorithm can be easily extended to N-dimensional space compared to TPS-RSM algorithm and more over, CPD can control the locality of spatial smoothness by changing the Gaussian filter width, whereas TPS dose not have such flexibility. In comparison, we introduce our learning-based model-free network CPD-Net, which can guarantee learning a smooth registration displacement field without any additional regularization term such as in CPD and TPS.

Ma et al. [24] introduced a L_2E estimator for non-rigid registration for handling significant scale changes and rotations. Ma et al. [8] proposed a non-parametric vector field consensus algorithm to establish the robust correspondence between two sets of points. Their experimental result demonstrated that the proposed method is quite robust to outliers. In [7], the authors emphasized the importance to preserve local and global structures for non-rigid point set registration. Wang et al. [25] proposed path following strategy for graph matching in order to improve the computation efficiency. Zhou et al. [26] proposed a fast alternating minimization algorithm for multi-image matching.

Existing methods have achieved great success for both rigid and non-rigid point set registration over past decades. However, they are mainly concentrated on the stand-alone development of the optimization strategies for point set registration rather than the techniques to learn the registration process as a pattern. In this paper, the deficiency of these current algorithms drives us to develop a learning-based registration paradigm that is able to actively learn the knowledge about how to register two point sets, consequently, to adaptively utilize those knowledge to directly predict the geometric transformation without the necessary to start over a new iterative search process for each similar case.

1.3.2 Learning-based registration methods.

Recent great success of deep learning in various computer vision fields [27, 28, 29, 30, 31, 32, 33] motivates researchers to start modeling the registration problem using deep neural networks [1, 34, 33, 30, 31, 32]. Earlier attempt in this direction is mainly concentrated on the development of learning-based registration methods for pairwise image registration. For example, Rocco et al. [1] developed a

CNN architecture to predict both rigid and non-rigid transformation for 2D image matching. Balakrishnan et al. [34] proposed a deep learning method to predict the non-rigid deformation field with application in deformable medical image registration. Both works share the common use of deep learning for visual feature learning from image to formulate the pairwise image correlations. The method presented in [1] tends to predict the parameters of TPS-based transformation function for pairwise image registration, while the authors in [34] aim to predict a smooth registration field to approximate non-rigid transformation. Though it is not a direct registration model, Zeng et al. [33] proposed a volumetric 3D-CNN to learn local shape descriptor geometric patch matching. The aforementioned learning-based registration methods, despite not working on point set registration, are encouraging for us to take a further step in this paper to investigate the possibility of learning point set registration using deep neural networks.

Chapter 2

Learning-based Registration Networks

(This chapter is submitted as paper “Non-Rigid Point Set Registration Networks” under review and paper “Coherent Point Drift Networks: Unsupervised Learning of Non-Rigid Point Set Registration” under review. The code for PR-Net is available at: <https://github.com/Lingjing324/PR-Net>.)

In this chapter, we introduce two learning-based registration networks PR-Net and CPD-Net, which can directly predict the desired transformation to align the source and target point sets. In contrast to previous methods, our network jointly learn a registration pattern from a training dataset and is able to instantly predict the desired transformation for an unseen pair from testing dataset without additional iterative optimization process.

We firstly introduce a model-based learning networks for point set registration network PR-Net. Different from image data with a regular grid, point cloud data is often recorded in an irregular and disordered format. Learning the point set registration requires the deep neural networks to be applicable to irregular point cloud data. In addition, unlike the image containing rich texture and color information, the point cloud is solely represented with geometric information (i.e. coordinates, curvature, normal). This suggests that a learning-based solution for point set registration needs to address two main technical challenges: 1) robust learning of both local and global geometric feature from point clouds and 2) robust learning of the transformation from well-defined correlation measure between pairwise geometric feature sets. Therefore, the proposed PR-Net investigates two major research problems: 1) the design of the techniques for point cloud learning by introducing a novel reference operator to enable formulating the correlation measure on arbitrary-structured data, and 2) the development of learning paradigm for the geometric transformation learning from pairwise feature sets.

Figure 2.1 illustrates the pipeline of the proposed PR-Net which is composed of three main components. The first component is “learning shape descriptor tensor”. In this component, the proposed grid-reference structure is developed to enable feature learning and formulate the correlation relationship on arbitrary-

structured data. The second component is “learning shape correlation tensor”. In this component, the shape correlation tensor is developed as a metric to further evaluate the correlation between two shape descriptor tensors of point sets to be registered. The shape correlation tensor is formulated as “all-to-all” point-wise computation from the pair of shape descriptor tensors evaluated in the first component. The third component is “learning of the parameters of transformation”. In this component, we exploit the function mapping between space of the “shape correlation tensor” and “the parameters of transformation” to determine the best geometric transformation that statistically aligns the source point cloud set and the target one. In this chapter, PR-Net utilizes the CNN as functional regression model to approximate the aforementioned mapping function for the parameters learning of the desired transformation.

In PR-Net, we propose a novel technique to learn the global and local shape aware “shape descriptor tensor” directly from the point cloud with irregular and disordered format. The shape descriptor tensor is proved to be effective and efficient in extracting the geometric shape features, even for point cloud in presence of missing points, noise, and outliers. A novel shape correlation tensor is proposed to comprehensively evaluate the correlation between two point sets to be registered. We propose a novel statistical alignment loss function that drives our structure to determine the optimal geometric transformation that statistically aligns the source point cloud set and the target one. In all, we propose a novel learning-based point set registration paradigm which learns registration patterns from training data, consequently, to adaptively utilize that knowledge to directly predict the geometric transformation for aligning a new pair of point sets, without the necessity to start over a new iterative search process. In conclusion, given a large number

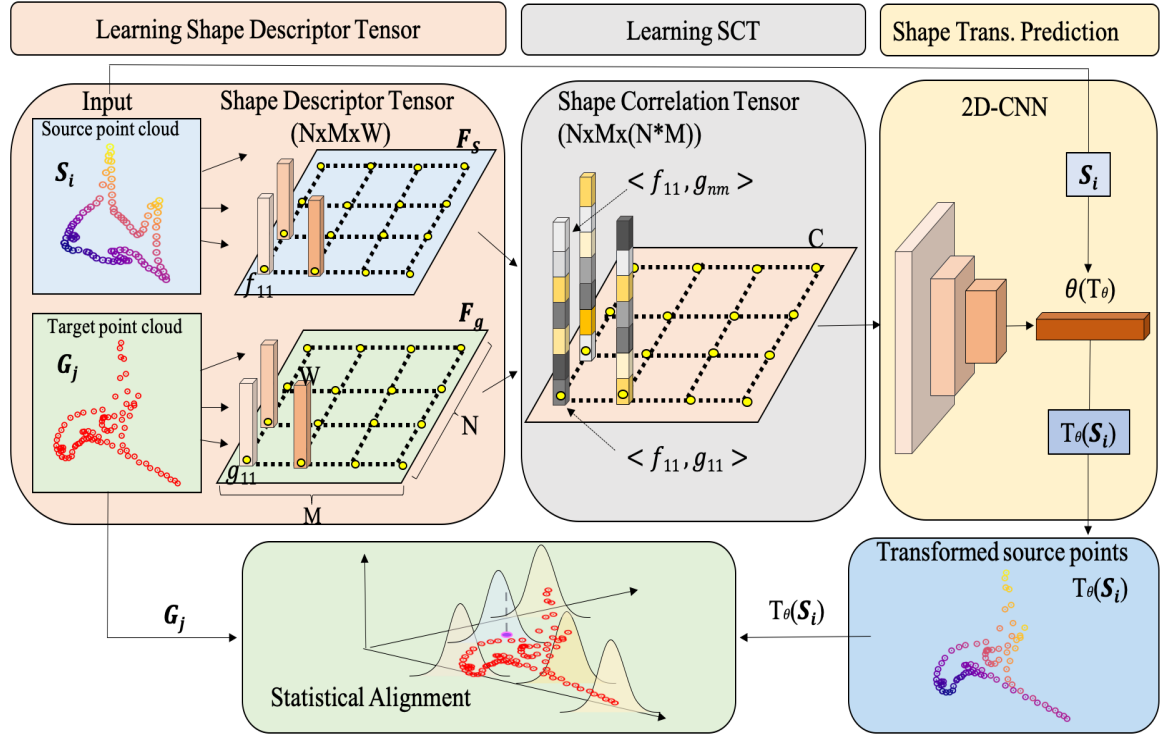


Figure 2.1: PR-Net pipeline. The proposed PR-Net includes three parts: learning shape descriptor tensor (SCT), learning correlation tensor, and shape transformation prediction. For a pair of source point set S_i and target point set G_j , we first generate two reference grids and map points of source and target point sets on them as two shape descriptor tensor F_s and F_g . We define the shape correlation tensor C between the source and target shape descriptor tensors. By leveraging 2D-CNN, we learn the desired parameters θ of transformation T_θ based on the shape correlation tensor. The learned optimal model transforms source point set to be statistically aligned with the target point set.

of data set for training, PR-Net demonstrates a stable generalization ability to directly predict the desired non-rigid transformation for the unseen point clouds data even in presence of a great level of noise, missing points, and outliers.

Though PR-Net is capable of learning the point registration, there are still some challenges that are left to be addressed. Firstly, our current PR-Net indirectly uses the regular grids to assist with the shape feature learning. A continuous operator, which can directly be applied on point for feature learning, would be more applicable for point registration. Secondly, PR-Net uses the TPS to model the geometric transformation. Though it predicts impressive registration performance for shapes with moderate deformation, the unsatisfactory performance for shapes with large deformation motivates us to study a model-free geometric transformation (e.g. the displacement field). Thirdly, PR-Net is difficult to be extended to 3d or nd dimensional domain. Therefore, considering these limitations, we further introduce a model-free learning-based network for deformable point set registration CPD-Net, named coherent point drift networks (CPD-Net). CPD-Net leverages the power of deep neural network to fit an arbitrary function, that adaptively accommodates different levels of complexity of the desired geometric transformation. Particularly, CPD-Net is proved with a theoretical guarantee to learn a continuous displacement vector function that could further avoid imposing additional parametric smoothness constraint as in previous works. Our experiments verify CPD-Net’s impressive performance for non-rigid point set registration on various 2D/3D datasets, even in presence of significant displacement noise, outliers, and missing points.

Figure 2.2 illustrates the pipeline of the proposed CPD-Net which consists of three major components. The first component is “Learning Shape Descriptor”. In this component, the global shape descriptor is learned with a multilayer perceptron

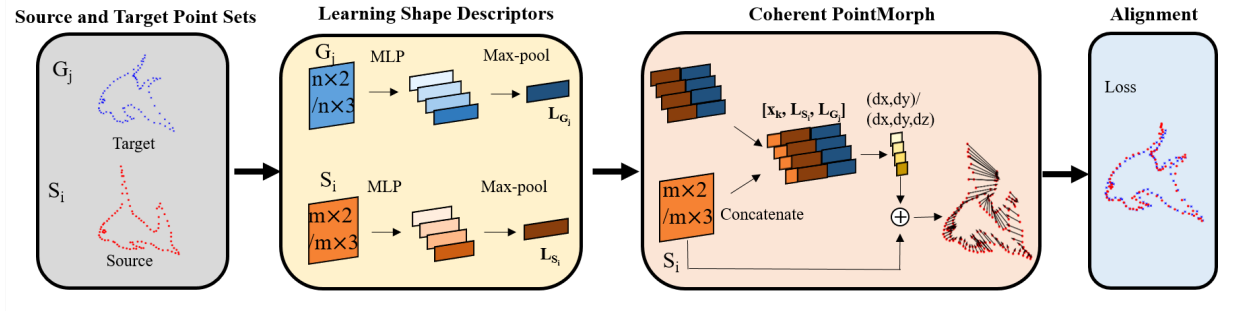


Figure 2.2: Our pipeline. The proposed structure includes three parts: learning shape descriptor, coherent PointMorph, and the alignment Loss. For a pair of source point set \mathbf{S}_i and target point set \mathbf{G}_j , we firstly leverage MLPs to learn two global descriptors $\mathbf{L}_{\mathbf{S}_i}$ and $\mathbf{L}_{\mathbf{G}_j}$. We then concatenate these two descriptors to each coordinate $\{x_k\}_{k=1,2,\dots,m}$ of source points as the input ($[\mathbf{x}_k, \mathbf{L}_{\mathbf{S}_i}, \mathbf{L}_{\mathbf{G}_j}]$) for PointMorph structure. We further use MLPs to learn the drifts for each source point. Finally we move the source point set by our predicted drifts and define the alignment loss function between target and transformed source point sets for back-propagation.

(MLP). The second component is “Coherent PointMorph”. In this component, we firstly concatenate the point coordinate of source point, the global shape descriptor of source point set, and global shape descriptor of target point set to form a new descriptor for each source point. Three successive MLP takes the new descriptor to learn the continuous displacement vector field. The third component is “Point Set Alignment”. In this component, a loss function is defined to assess the quality of alignment.

Overall, we introduces a novel Coherent Point Drift Networks (CPD-Net) that can be trained in unsupervised manner, consequently it can be generalized to predict geometric transformation for non-rigid point set registration. CPD-Net leverages the power of deep neural network to fit an arbitrary function, that is able to accommodate different levels of complexity of the target geometric transformation for best aligning the pair of point sets. The CPD-Net theoretically guarantees

the continuity of predicted displacement field as geometric transformation, which naturally eliminate the necessity to impose a parametric hand-crafted smoothness constraint. The CPD-Net is free of specific geometric model selection for modeling the desired transformation, which avoids the potential mismatch between the transformation described by specific adopted models and the actual transformation required for point set registration.

2.1 Problem statement

Prior to discussion of our approach, we first define the point set registration task. Let the training data set $\mathbf{D} = \{(\mathbf{S}_i, \mathbf{G}_j)\}$, where $\mathbf{S}_i, \mathbf{G}_j \subset \mathbb{R}^N$. We denote \mathbf{S}_i source point set and \mathbf{G}_j target point set. In this chapter, we mainly discuss the situation when $N = 2$ and $N = 3$. We assume $\forall (\mathbf{S}_i, \mathbf{G}_j) \in \mathbf{D}, \exists \theta_i, T_{\theta_i} : \mathbb{R}^N \rightarrow \mathbb{R}^N$, such that $T_{\theta_i} : \mathbf{x}_i \rightarrow \mathbf{x}'_i$ where $\mathbf{x}_i \in \mathbf{S}_i$ and $\mathbf{x}'_i \in \mathbf{G}_j$. T_{θ_i} can be rigid or non-rigid transformation with parameters θ_i . For previous methods, θ_i is optimized in a iterative searching process to optimally align a given target and source point sets. For our method, we assume the existence of a neural network structure g with a set of all its weights γ , such that $g_{\gamma}(\mathbf{S}_i, \mathbf{G}_j) = \theta_i$. Our optimization task becomes:

$$\gamma^{\text{optimal}} = \underset{\gamma}{\operatorname{argmin}} [\mathbb{E}_{(\mathbf{S}_i, \mathbf{G}_j) \sim \mathbf{D}} [\mathcal{L}(T_{g_{\gamma}(\mathbf{S}_i, \mathbf{G}_j)}(\mathbf{S}_i), \mathbf{G}_j)]], \quad (2.1)$$

Therefore, for a given training set \mathbf{D} , our task is to optimize parameters γ instead

of θ/T_θ . The desired θ/T_θ is our model’s output. $\mathcal{L}(\cdot)$ represents a similarity measure.

2.2 Methods

2.2.1 PR-Net

We introduce our approach in the following sections. From section 2.2.1.1 to 2.2.1.4, four successive parts are illustrated to explain each module of our method in details. Section 2.2.1.1 illustrates our structure for learning shape descriptor tensor for point sets. In section 2.2.1.2, we introduce shape correlation tensor based on the learned shape descriptors. The non-rigid shape transformation prediction is introduced in section 2.2.1.3. The definition of the loss function is discussed in section 2.2.1.4 and the settings of the training and model configuration are explained in section 2.2.1.5.

2.2.1.1 Learning shape descriptor tensor

The first part of our structure is learning the shape descriptor for point sets. To address the problem of irregular format of point set, we introduce two point grids $\mathbf{M}_{\mathbf{S}_i}$ and $\mathbf{M}_{\mathbf{G}_j}$ as reference point sets, which are overlaid on the source point set \mathbf{S}_i and the target point set \mathbf{G}_j respectively.

For each point in the reference point sets, we learn a shape descriptor tensor \mathbf{F}_s^i or \mathbf{F}_g^j by mapping the local and global information of non-regular source or target point set on it. Specifically, as shown in Figure 2.3, taking \mathbf{S}_i for example, $\forall \mathbf{x}_i \in \mathbf{M}_{\mathbf{S}_i}$, \mathbf{x}_i is 2D/3D geometric coordinates and we define the single layer mapping $U : (\mathbf{x}_i, \mathbf{S}_i) \rightarrow \mathbb{R}^d$ as following:

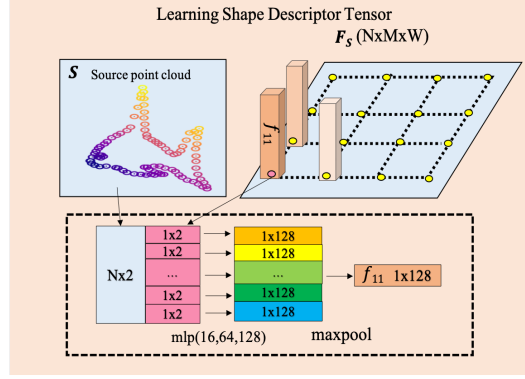


Figure 2.3: The schema of learning shape descriptor tensor process.

$$\mathcal{U}(\mathbf{x}_i, \mathbf{S}_i) = \text{Maxpool}\{\text{ReLU}(\mathbf{u}_m[\mathbf{x}_i, \mathbf{y}_i] + \mathbf{c}_m)\}_{\mathbf{y}_i \in \mathbf{S}_i} \quad (2.2)$$

,where parameters $\mathbf{u}_m \in \mathbb{R}^{m \times 4/6}$, $c_m \in \mathbb{R}^{m \times 1}$ and $[\ast, \ast]$ means concatenation. For multi-layers' structure, we repeat the linear combination and Leaky-ReLU [35] activation parts before applying the Max-pool layer. The MLP-based structure was firstly introduced in PointNet [30] for directly learning geometric features from point cloud. Please refer to PointNet [30] for more details. The single layer MLP-based function $\mathcal{U}(\ast)$ can be regarded as a mapping to exact features from non-regular point set, which is driven by the loss function. In our case, we have three layers MLP. In this way, we transfer information of source and target point sets to two shape descriptor tensors on reference grids. We define the shape descriptor tensor \mathbf{F}_S^i and \mathbf{F}_G^j . $\mathbf{F}_S^i, \mathbf{F}_G^j \in \mathbb{R}^{n \times m \times d}$ where

$$\mathbf{F}_{\mathbf{S}}^i = \begin{bmatrix} \mathcal{U}(\mathbf{x}_{11}, \mathbf{S}_i), & \mathcal{U}(\mathbf{x}_{12}, \mathbf{S}_i) & \dots & \mathcal{U}(\mathbf{x}_{1n}, \mathbf{S}_i) \\ \mathcal{U}(\mathbf{x}_{21}, \mathbf{S}_i) & \mathcal{U}(\mathbf{x}_{22}, \mathbf{S}_i) & \dots & \mathcal{U}(\mathbf{x}_{2n}, \mathbf{S}_i) \\ & & \dots & \\ \mathcal{U}(\mathbf{x}_{n1}, \mathbf{S}_i) & \mathcal{U}(\mathbf{x}_{n2}, \mathbf{S}_i) & \dots & \mathcal{U}(\mathbf{x}_{nm}, \mathbf{S}_i) \end{bmatrix}$$

, where $\mathbf{x}_{nm} \in \mathbf{M}_{\mathbf{S}_i}$. Similarly, we have the shape descriptor tensor $\mathbf{F}_{\mathbf{G}}^j$ for $\mathbf{M}_{\mathbf{G}_j}$.

2.2.1.2 Shape correlation tensor

As shown in Figure 2.4, for the two source and target grid points $\mathbf{M}_{\mathbf{S}_i}$ and $\mathbf{M}_{\mathbf{G}_j}$ with shape descriptor tensors $\mathbf{F}_{\mathbf{S}}^i = [\mathbf{f}_{ij} = \mathcal{U}(\mathbf{x}_{ij}, \mathbf{S}_i)]$ and $\mathbf{F}_{\mathbf{G}}^j = [\mathbf{g}_{ij} = \mathcal{U}(\mathbf{x}_{ij}, \mathbf{G}_i)]$, our next step is to define the shape correlation tensor between the input and target shape descriptor tensors. We define the shape correlation tensor in the following step. Let \mathcal{M} be a similarity metric, such that $\mathcal{M} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. In this chapter, we simply let \mathcal{M} as inner product. $\forall \mathbf{f}_{ij} \in \mathbf{F}_{\mathbf{S}}^i$, we sort the its point-wise correlation with elements in $\mathbf{F}_{\mathbf{G}}^j$ as $\mathbf{C}_{ij} \in \mathbb{R}^t$ and $t = nm$, where

$$\mathbf{C}_{ij} = [\mathcal{M}(\mathbf{f}_{ij}, \mathbf{g}_{11}), \mathcal{M}(\mathbf{f}_{ij}, \mathbf{g}_{12}), \dots, \mathcal{M}(\mathbf{f}_{ij}, \mathbf{g}_{md})] \quad (2.3)$$

We define $\mathbf{C} = [\mathbf{C}_{ij}] \in \mathbb{R}^{n \times m \times t}$ as the shape correlation tensor. It has t -dimensional channel to save the correlation information between each the point in $\mathbf{M}_{\mathbf{S}_i}$ with all the points in $\mathbf{M}_{\mathbf{G}_j}$. We normalize each channel of element \mathbf{C}_{ij} in the

shape correlation tensor.

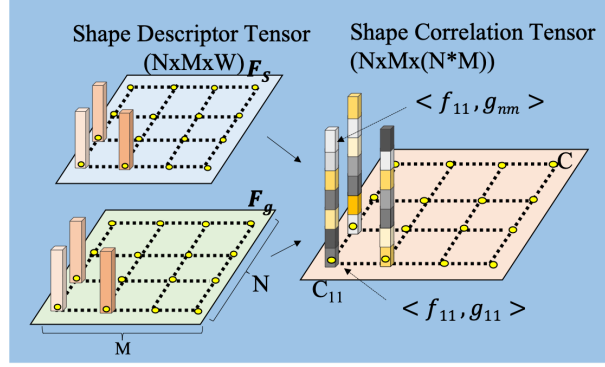


Figure 2.4: The schema of formulating correlation tensor process.

2.2.1.3 Shape transformation prediction

Before we discuss shape transformation prediction, we firstly review two classical parametric functions for rigid and non-rigid transformations. For affine transformation including translation, scaling, rotation and shear. Let $\theta_{rigid} = \{\alpha, r_1, r_2, r_3, r_4, s_1, s_2\}$ and we have

$$T_{\theta} = \begin{bmatrix} r_1 \cos \alpha & r_2 \sin \alpha & s_1 \\ r_4 \sin \alpha & r_5 \cos \alpha & s_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Even though we do not discuss the rigid case in this chapter, our model can be easily adjusted for rigid registration.

For non-rigid transformation, let $\theta_{nonrigid}$ be the controlling points in Thin Plate Spine. In this chapter, we choose 9/27 controlling points distributed as a $3 \times 3 / 3 \times 3 \times 3$ grid for 2D/3D data. For a pair of 2D source and target point sets, our target $\theta_{nonrigid} = \{(\theta_1, \theta_2), \dots, (\theta_{17}, \theta_{18})\}$, are a set of coordinates of nine controlling

points in TPS [21]. Let the original controlling points in TPS be θ_0 and $\theta_0 = [(0, 0), (-1, 0), \dots, (1, -1)]$. For a pair of 3D source and target point sets, our target $\theta_{\text{nonrigid}} = \{(\theta_1, \theta_2, \theta_3), \dots, (\theta_{79}, \theta_{80}, \theta_{81})\}$, are a set of coordinates of nine controlling points in TPS [21]. Let the original controlling points in TPS be θ_0 and $\theta_0^{2D} = [(0, 0), (-1, 0), \dots, (1, -1)]$ and $\theta_0^{3D} = [(0, 0, 0), (-1, 0, 0), (1, 0, 0), \dots, (1, 1, 1)]$. After achieving new positions of controlling points θ_{nonrigid} , together with θ_0 , we can solve the non-rigid transformation T_θ according to TPS. In this case, we have 18/81 parameters to be optimized for defining the non-rigid transformation to align the 2D/3D source and target point sets. For a given pair of source point set \mathbf{S}_i and target point set \mathbf{G}_j as inputs, based on their shape correlation tensor \mathbf{C} from the previous step, we further use 2D-CNN/3D-CNN with a successive of fully connected layers to predict the desired parameters θ in transformation T_θ .

2.2.1.4 From statistical alignment to loss functions

The last step is to define the loss function between the transformed source point set $T_\theta(\mathbf{S}_i)$ and the target point set \mathbf{G}_j . Due to the disorderliness of point cloud, there is no direct corresponding relationship between these two point sets. Therefore, a distance metric between two point sets instead of point/pixel-wise loss used in image registration should be desired. Besides, the suitable metric should be differentiable and efficient to compute. For 3D point set generation, Fan et al. [36] first proposed Chamfer Distance loss, which is widely used in practice. Registration problem can be treated as statistical alignment between two distributions of source and target point sets. We treat target point set as centroids of a Gaussian Mixture Model and we fit the transformed source point set as data into this GMM model so that we can maximize the likelihood of the GMM.

Chamfer Distance (C.D.). Chamfer loss is a simple and effective metric to be defined on two non-corresponding point sets. It does not require the same number of points and in many tasks and it provides high quality results in practice. We define the Chamfer loss on our transformed source point set $T_\theta(\mathbf{S})$ and target points set \mathbf{G} as:

$$L_{\text{Chamfer}}(T_\theta(\mathbf{S}), \mathbf{G}|\gamma) = \sum_{x \in T_\theta(\mathbf{S})} \min_{y \in \mathbf{G}} \|x - y\|_2^2 + \sum_{y \in \mathbf{G}} \min_{x \in T_\theta(\mathbf{S})} \|x - y\|_2^2 \quad (2.4)$$

where γ represents all the parameters in MLP layers and 2D-CNN layers from section 2.2.1.1, 2.2.1.2 and 2.2.1.3. In experiments for PR-Net, we use Chamfer Distance (C.D.) as evaluation metric.

Gaussian Mixture Model (GMM) loss. Let our source point set $\mathbf{S} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ and transformed target point set $\mathbf{T}_\theta(\mathbf{S}) = (T_\theta(\mathbf{x}_1), T_\theta(\mathbf{x}_2), \dots, T_\theta(\mathbf{x}_N))$. The target point set is $\mathbf{G} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M)$ where \mathbf{x}_i and $\mathbf{y}_i \in \mathbb{R}^2/\mathbb{R}^3$ in this chapter. We consider Gaussian-mixture model $p(T_\theta(\mathbf{x}_i)) = \sum_{m=1}^M \frac{1}{M} p(T_\theta(\mathbf{x}_i)|m)$ with $\mathbf{x}|m \sim N(\mathbf{y}_m, \sigma^2 \mathbf{I}_2)$, where our target point set acts as the 2/3-dimensional centroids of equally-weighted Gaussian mixture model. In general we want our predicted point set to maximally satisfy the Gaussian Mixture model. Therefore, we

define the loss function (GMM loss) as :

$$L_{\text{GMM}}(T_{\theta}(\mathbf{S}), \mathbf{G}|\gamma) = - \sum_{\mathbf{x} \in \mathbf{S}} \log \sum_{\mathbf{y} \in \mathbf{G}} e^{-\frac{1}{2} \left\| \frac{T_{\theta}(\mathbf{x}) - \mathbf{y}}{\sigma} \right\|^2} \quad (2.5)$$

,where γ represents all the parameters in MLP layers and 2D-CNN layers from section 2.2.1.1, 2.2.1.2, and 2.2.1.3. σ is the standard deviation in GMM. We set σ to be identical for each Gaussian distribution in GMM. σ is a hyper-parameter to choose in practice. Even though it is a constant for each input, we have more sophisticated strategy for choosing it in practice as discussed in section 2.2.1.4. We use GMM loss as our loss function for PR-Net.

2.2.1.5 Model settings

We train our network using batch data form training data set $\{(\mathbf{S}^i, \mathbf{G}^i) | (\mathbf{S}^i, \mathbf{G}^i) \in \mathbf{D}\}_{i=1,2,\dots,b}$. b is the batch size and is set to 16. For learning the shape descriptor tensor in 2.2.1.1, the input is $N \times 4/N \times 6$ matrix and we use 4 MLP layers with dimensions (16,32,64,128) and a Maxpool layer to convert it to a 128-dimensional descriptor for each point in 11×11 reference grid. For the shape correlation tensor \mathbf{C} discussed in 2.2.1.2 and 2.2.1.3, we use three 2D-CNN/3D-CNN layers with kernel size (3,3),(4,4),(5,5) and dimension (128,256,512) with two successive fully connected layers with dimensions (64, 18)/(512,81). Learning rate is set as 0.0001 with 0.995 exponential decay with Adam optimizer. We use leaky-ReLU [35] activation function and implement batch normalization [37] for every layer except the output layer. We use deterministic annealing for the standard deviation σ which

is initially set to 1, and for each step n we reduce it to $\sqrt{1/n}$ until a margin value of 0.1. Gradual reducing σ leads to a coarse-to-fine match. For outlier and missing points case, we slightly increase the margin value to 0.12.

2.2.2 CPD-Net

We introduce our approach in the following sections. In section 2.2.2.1, our first module is introduced for learning shape descriptor from a 2D/3D source/target point sets. Section 2.2.2.2 illustrates coherent PointMorph module for learning the smooth drifts to align the source point set with the target one. In section 2.2.2.3, The definition of the loss function is provided. The model configurations and the settings for training are described in section 2.2.2.4

2.2.2.1 Learning shape descriptor

For a given input point set, we firstly learn a shape descriptor that captures representative and deformation-insensitive geometric features. Let $(\mathbf{S}_i, \mathbf{G}_j)$ denotes the input source and target point sets and $(\mathbf{L}_{\mathbf{S}_i}, \mathbf{L}_{\mathbf{G}_j})$ denotes their shape descriptor, where $\mathbf{L}_{\mathbf{S}_i}, \mathbf{L}_{\mathbf{G}_j} \subset \mathbb{R}^m$ as shown in Figure 2.5. To address the problem of irregular format of point set, we introduce the following encoding network, which includes t successive multi-layer perceptrons (MLP) with ReLu activation function $\{f_i\}_{i=1,2,\dots,t}$, such that: $f_i : \mathbb{R}^{w_i} \rightarrow \mathbb{R}^{w_{i+1}}$, where w_i is the input layer's dimension and w_{i+1} is the output layer's dimension. The encoder network is defined as: $\forall(\mathbf{S}_i, \mathbf{G}_j)$,

$$\mathbf{L}_{\mathbf{S}_i} = \text{Maxpool}\{f_t f_{t-1} \dots f_1(\mathbf{x}_i)\}_{\mathbf{x}_i \in \mathbf{S}_i} \quad (2.6)$$

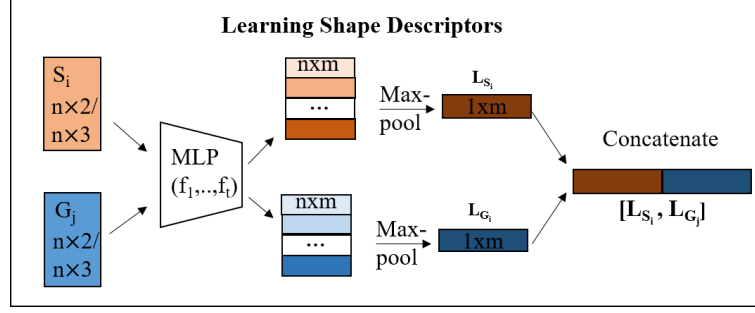


Figure 2.5: The schema of learning shape descriptor tensor process.

$$\mathbf{L}_{\mathbf{G}_j} = \text{Maxpool}\{f_t f_{t-1} \dots f_1(\mathbf{x}_i)\}_{\mathbf{x}_i \in \mathbf{G}_j} \quad (2.7)$$

We use the Maxpool function to extract the order-invariant descriptors from the input point sets. The readers can refer to PointNet [30] for detailed discussion. One can also use other symmetric operators such as summation, average pooling function and so on. This structure can be easily adapted for 3D point set inputs. Other point signature learning architecture such as PointNet++ [38] can be easily implemented here as well.

2.2.2.2 Coherent PointMorph architecture

For the next step, we define a PointMorph MLP (multi-layer perceptrons) architecture for learning the coherent point drifts to move the source point set towards alignment with the target one as shown in Figure 2.6. This architecture includes successive multi-layer perceptrons (MLP) with ReLu activation function: $\{g_i\}_{i=1,2,\dots,s}$, such that: $g_i : \mathbb{R}^{v_i} \rightarrow \mathbb{R}^{v_{i+1}}$, where v_i is the input layer's dimension and v_{i+1} is the output layer's dimension. Therefore, $\forall(\mathbf{S}_i, \mathbf{G}_j)$,

$$\mathbf{d}\mathbf{x}_i = g_s g_{s-1} \dots g_1([\mathbf{x}_i, \mathbf{L}_{\mathbf{S}_i}, \mathbf{L}_{\mathbf{G}_j}]) \quad (2.8)$$

$$\mathbf{S}'_i = \phi(\mathbf{S}_i) = \{\mathbf{x}_i + \mathbf{d}\mathbf{x}_i\}_{\mathbf{x}_i \in \mathbf{S}_i} \quad (2.9)$$

, where \mathbf{S}'_i is the transformed source point set and $\mathbf{d}\mathbf{x}_i$ represents the predicted drift for each point $\mathbf{x}_i \in \mathbf{S}_i$. The notation $[\ast, \ast]$ represents concatenation of vectors in same domain. We should notice that due to the high non-linearity of neural networks, there is no difficulty to minimize the similarity loss function between \mathbf{S}' and \mathbf{G} . However, the main challenge is to make sure the predicted drifts are coherent [3]. The coherency of drifts is quite important for holding reasonable correspondence and points interpolation for registering large scale point sets as well. Most previous methods deal with this problem by adding a penalization term on smoothness to trade-off target alignment loss for smoothness [34]. Our network naturally predicts smooth drifts because (1) by formula (6), for any neighbor points $\mathbf{x}_k \in \mathbf{S}$, with concatenating two identical global descriptors, the inputs $[\mathbf{x}_k, \mathbf{L}_S, \mathbf{L}_G]$ should be very close to each other as input for function g . (2) Function g is a simple linear combination with ReLU activation. This function is continuous and its first derivative should be a constant almost everywhere. Based on the special function g and our assumption, we have the following theorem.

Claim: For a single layer perceptron with ReLU activation function g , $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{S}, \forall \varepsilon > 0, \exists \delta > 0$, such that $\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon \implies \|\mathbf{d}\mathbf{x}_i - \mathbf{d}\mathbf{x}_j\| < \delta$, where $\mathbf{d}\mathbf{x}_i$ is defined as $\mathbf{d}\mathbf{x}_i = g([\mathbf{x}_i, \mathbf{L}_S, \mathbf{L}_G])$ similarly in equation (6).

Proof in sketch. Since g is a linear function with ReLU activation and we assume its weights \mathbf{w} converges to \mathbf{w}' after training. $\|\mathbf{x}_i - \mathbf{x}_j\| = \|[\mathbf{x}_i, \mathbf{L}_S, \mathbf{L}_G] - [\mathbf{x}_j, \mathbf{L}_S, \mathbf{L}_G]\|$ since $\mathbf{L}_S, \mathbf{L}_G$ are identical for each $\mathbf{x}_i \in \mathbf{S}$. Since \mathbf{w}' is constant, $\exists C > 0$, s.t. $\|\mathbf{x}_i - \mathbf{x}_j\| > C\|\mathbf{w}'[\mathbf{x}_i, \mathbf{L}_S, \mathbf{L}_G] - \mathbf{w}'[\mathbf{x}_j, \mathbf{L}_S, \mathbf{L}_G]\|$. Since function g is continuous, if $\mathbf{w}'[\mathbf{x}_i, \mathbf{L}_S, \mathbf{L}_G] \geq 0$, $\exists \delta_1 > 0$ such that $\forall \mathbf{x}_i$, if $\|\mathbf{x}_i - \mathbf{x}_j\| <$

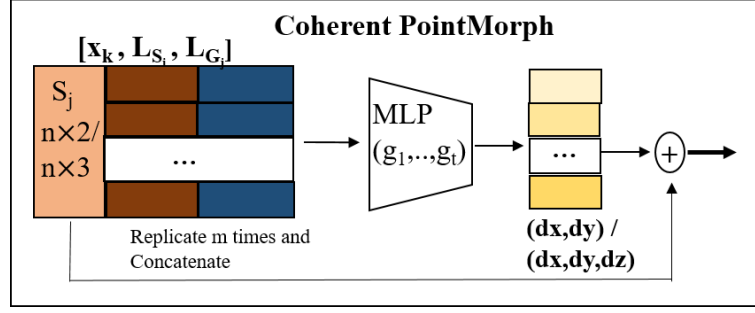


Figure 2.6: The schema of learning coherent PointMorph process.

$\delta_1, \mathbf{w}'[x_j, \mathbf{L}_S, \mathbf{L}_G] \geq 0$. Similarly if $\mathbf{w}'[x_i, \mathbf{L}_S, \mathbf{L}_G] \leq 0$, $\exists \delta_2 > 0$ such that $\forall x_j$, if $\|x_i - x_j\| < \delta_2, \mathbf{w}'[x_j, \mathbf{L}_S, \mathbf{L}_G] \leq 0$. Therefore, we pick $\delta = \min(\delta_1, \delta_2, \varepsilon/C)$,

$$\begin{aligned}
\|dx_i - dx_j\| &= \|g([x_i, \mathbf{L}_S, \mathbf{L}_G]) - g([x_j, \mathbf{L}_S, \mathbf{L}_G])\| \\
&= \|\max(\mathbf{w}'[x_i, \mathbf{L}_S, \mathbf{L}_G], 0) - \max(\mathbf{w}'[x_j, \mathbf{L}_S, \mathbf{L}_G], 0)\| \\
&= \max(\|\mathbf{w}'[x_i, \mathbf{L}_S, \mathbf{L}_G] - \mathbf{w}'[x_j, \mathbf{L}_S, \mathbf{L}_G]\|, 0) \\
&< \max\left(\frac{1}{C}\|x_i - x_j\|, 0\right) \\
&< \max(0, \varepsilon/C) \\
&= \varepsilon/C \\
&< \delta
\end{aligned} \tag{2.10}$$

For each point $x_i \in \mathbf{S}$ in the source point set, the weights in the PointMorph MLP are shared cross all points. Moreover, the two shape descriptors are duplicated for each point. Since our MLP layers are all continuous function, even though the drifts depend on the two global descriptors, but this motion of deformation is strictly limited by source points' original geometric locations, which guarantees the continuity of the drifts on each source point.

2.2.2.3 Loss functions

In this part, we define the similarity measure between the transformed source point set $\phi(\mathbf{S}_i)$ and the target point set \mathbf{G}_j as both our loss function and evaluation metric. For two point sets, due to the absence of the corresponding relationship for each point, we cannot adopt the pixel-wise loss in image registration. Fan et al. [36] first proposed Chamfer Distance (C.D.), which is widely used in practice. We define the Chamfer loss on our transformed source point set \mathbf{S}' and target points set \mathbf{G} as:

$$L(\mathbf{S}', \mathbf{G}|\theta) = \sum_{x \in \mathbf{S}'} \min_{y \in \mathbf{G}} \|x - y\|_2^2 + \sum_{y \in \mathbf{G}} \min_{x \in \mathbf{S}'} \|x - y\|_2^2 \quad (2.11)$$

where θ represents all the weights in the our network structure. Chamfer loss (C.D.) is our main choice in this pater. For dataset in presence of outliers and missing points noise, we use the following clipped Chamfer loss:

$$L(\mathbf{S}', \mathbf{G}|\theta) = \sum_{x \in \mathbf{S}'} \max(\min_{y \in \mathbf{G}} \|x - y\|_2^2, c) + \sum_{y \in \mathbf{G}} \max(\min_{x \in \mathbf{S}'} \|x - y\|_2^2, c) \quad (2.12)$$

where c is a hyper-parameter to choose. In our experiment 3 of CPD-Net, we choose c equal to 0.1.

2.2.2.4 Settings of CPD-Net

We train our network using batch data form training data set $\{(\mathbf{S}^i, \mathbf{G}^i) | (\mathbf{S}^i, \mathbf{G}^i) \in \mathbf{D}\}_{i=1,2,\dots,b}$. b is the batch size and is set to 16. As we explain in section 2.2.2.1,

for learning the shape descriptor tensor, the input is $N \times 4/6$ matrix and we use 5 MLP layers with dimensions (16, 64, 128, 256, 512) and a Maxpool layer to convert it to a 512-dimensional descriptor. For learning the coherent PointMorph discussed in 2.2.2.2, we use three layers MLP with dimension (256, 128, 2/3). We use ReLU activation function and implement batch normalization [37] for every layer except the output layer. Learning rate is set as 0.0001 with 0.995 exponential decay with Adam optimizer. The model is trained on single Tesla K80 GPU.

2.3 Experimental Results

2.3.1 PR-Net

In this section, we implement a set of experiments to validate the performance of our proposed PR-Net for non-rigid point set registration from different aspects (i.e. accuracy and time). In section 1.3.1.1, we discuss how we prepare the experimental dataset. In section 1.3.1.2, we compare PR-Net with non-learning based non-rigid point set registration method. In section 1.3.1.3, we validate the robustness of PR-Net against the different level of geometric deformation. In section 1.3.1.4, we validate the robustness of PR-Net against the different types of noise. In section 1.3.1.5, we further verify that PR-Net can handle registration tasks for various types of dataset.

2.3.1.1 Dataset preparation

The point cloud data is often featured with geometric structural variations with presence of a variety of noise (e.g. outliers, missing points), which poses challenges for point set registration. An effective registration solution should be robust to the

presence of those noise to provide the desired geometric transformation. Therefore, in order to assess PR-Net's performance, we simulate the commonly recognized noise to the raw point sets to prepare the experimental data. To prepare the geometric structural variation, we randomly choose a certain number of samples from the point set and use them as the controlling points of a thin plate spline (TPS) transformation. A zero-mean Gaussian is superposed to each controlling point to simulate a random drift from their original positions. The TPS is then applied to synthesize the deformed point set with different level of structural variation. The $1/2$ of standard deviation of the above mentioned Gaussian is used to measure the deformation level. To prepare the position drift (P.D.) noise, we applied a zero-mean Gaussian to each sample from the point set. The level of P.D. noise is defined as the standard deviation of Gaussian. To prepare the data incompleteness (D.I.) noise, we randomly remove a certain amount of points from the entire point set. The level of D.I. noise is defined as ratio of the eliminated points and the entire set. To prepare the data outlier (D.O.) noise, we randomly add a certain amount of points generated by a zero-mean Gaussian to the point set. The level of D.O. noise is defined as the ratio of the added points to the entire point set. For all tests, we use the Chamfer Distance (C.D.) between a pair of point sets to provide a quantitative score to evaluate the registration performance.

2.3.1.2 Comparison to Non-learning based Approach

Different from previous efforts, the proposed PR-Net is a learning-based non-rigid point set registration method, which can learn the registration pattern to directly predict the non-parametric geometric transformation for the point sets alignment. As a learning-based approach to predict the non-rigid registration, it

Methods	CD	Time
CPD (Train) [3]	0.0038 ± 0.0031	~ 12 hours
PR-Net (Train)	0.0037 ± 0.0014	~ 13 minutes
CPD (test) [3]	0.0038 ± 0.0032	~ 12 hours
PR-Net (Test)	0.0044 ± 0.0016	~ 8 seconds

Table 2.1: Performance comparison with CPD for registering $10k$ pairs of point sets at deformation level 0.5.

is not applicable to have a direct comparison between PR-Net and other existing non-rigid iterative registration methods. To compare our method to non-learning based iterative method (i.e. Coherent Point Drift (CPD) [3]), we design the experiment as follows to assess both time and accuracy performance.

Experimental Setup: We conduct tests to compare PR-Net with the non-learning based approach. Coherent Point Drifts (CPD)[3] is a highly recognized non-rigid point set registration method. In this test, we synthesize 2D deformed fish data with deformation level of 0.5 to prepare $10k$ training dataset and $10k$ testing dataset. Our PR-Net is firstly trained before applied to the $10k$ testing dataset. The CPD is directly applied to the $10k$ testing dataset.

Result: We list the experimental result in the table 2.1. The second column shows the mean and standard deviation of all $10k$ C.D. after registration. The third column shows the time used for registering the $10k$ pairs of point sets. As we expect, after training PR-Net can perform the real-time non-rigid point set registration. The time used to register $10k$ pairs of point sets is around 8 seconds, which is order of magnitude less than the time (12 hours) consumed by CPD for point set registration of the entire $10k$ dataset. This is because of the fact that CPD

needs to repeatedly start over a new iterative process for a new pair of point sets. PR-Net clearly gains advantage over the non-learning based method by providing a faster solution to non-rigid point registration. We also want to note that it takes around 13 minutes to train our PR-Net on the $10k$ dataset with a comparative performance, which is also significantly less than 22 hours used by CPD.

In addition to the efficiency (registration speed), we are also interested in the effectiveness that indicates how well PR-Net can generalize from training data to directly predict the desired geometric transformation for non-rigid point set registration. The comparative training and testing C.D. results are listed on the second column. The small difference between training and testing C.D. indicates a comparative small performance degradation from training to testing. Furthermore, we notice that C.D. of PR-Net has a smaller standard deviation than that of CPD, which suggests that PR-Net can provide a more stable registration as it obtains generalization ability to adapt properly to previously unseen data. In contrast, the CPD treats every new pair of point sets independently and has to repeatedly register them from the start.

2.3.1.3 Robust to Geometric Deformation

In this experiment, we take a detailed investigation on how well the PR-Net performs point set registration for 2D shapes at different deformation levels. This experiment shows a basic assessment of our model's performance and capacity for registering unseen highly deformed testing shapes.

Experimental Setup: We conduct tests to verify how well PR-Net performs on the data with different levels of geometric deformation. In this test, we synthesize

Deform. Level	Chamfer Distance
0.2	0.0013±0.0005
0.3	0.0019±0.0008
0.8	0.0161±0.0057
1.0	0.0153±0.0052
1.5	0.1267±0.0872

Table 2.2: Quantitative testing performance for 2D fish shape point set registration at different deformation level (Deform. Level)

2D deformed fish data with deformation levels from 0.3 to 1.5 to cover a good range of shape structural variation. The deformed 2D fish shapes are shown in Figure 2.7. For each level of deformation, we simulate $20k$ point sets as target point sets for training and simulate additional $10k$ point sets for testing. The quantitative result is shown in Table 2.2.

Result: After training, the PR-Net is applied to register testing datasets with different deformation levels. The quantitative experimental results are listed in Table 2.2. The second column lists the C.D. scores for a registered pair of source and target point sets with different deformation levels. As we can see from the evaluation, PR-Net can achieve impressive performance on non-rigid point set registration when the deformation level is less than 1.0 and the Chamfer Distance remains as low as 0.0153 as shown in Table 2.2. However when the deformation level reaches 1.5, there is a huge jump of C.D. from 0.0153 to 0.1267. This indicates that our model’s registration capacity dose have a clear upper bond. Once the deformation level reaches or higher than this upper bond, the performance of PR-Net can be dramatically reduced. We further check the qualitative results for better understanding PR-Net’s performance.

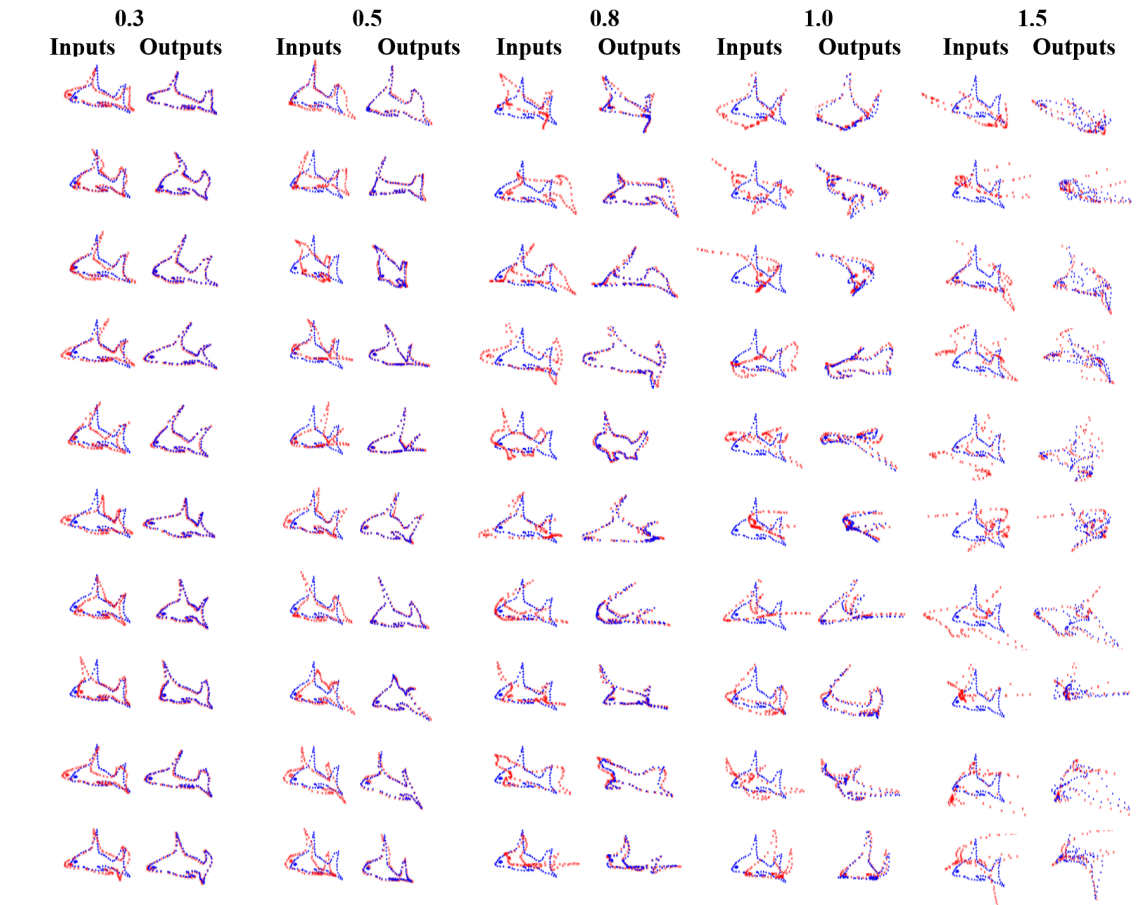


Figure 2.7: Testing results for 2D fish shape point set registration at different deformation levels. The deformation level increases from 0.3 to 1.5 from left to right. The presented shapes are randomly selected from same testing batch. The blue shapes are source point sets and the red shapes are target point sets. Please zoom-in for better visualization.

The corresponding qualitative results are demonstrated in Figure 2.7, which illustrates the pairs of point sets before and after registration. From the Figure 2.7, we can clearly see that the transformed source point set (in blue color) structurally aligns well with the target point set (in red color), which verifies PR-Net's registration capacity. Especially when deformation level is equal or less than 1.0, as shown in 2.7, PR-Net almost perfectly aligns the source and target point sets. As we mentioned before, when the deformation level reaches 1.5, the quantitative result experiences a dramatic drop. As displayed in Figure 2.7, for this deformation level 1.5, the geometric structure of 2D fish is significantly deteriorated, which poses much more challenges in determining the desired geometric transformation. Even for human beings, it is hard to tell the geometric meaning of the target point sets (Red shapes in Figure 2.7). But this also indicates that TPS, as a parametric geometric transformation model, might be limited in modeling the large structural variation in our test. We further investigate more complex geometric transformation model or model-free geometric transformation in our separate research reports.

2.3.1.4 Robust to Data Noise

While using the sensors such as LIDAR sensor and laser scanner, it is unavoidable that the data might be acquired with a variety types of noises. An effective non-rigid registration method should be robust to those noise in addition to the structural variations as discussed in previous section. Therefore, in this section, we focus on testing how well PR-Net can predict the non-rigid registration from the noisy dataset.

P.D. Level	C.D.	D.O. Level	C.D.	D.I. Level	C.D.
0.05	0.0052±0.0009	0.05	0.003±0.001	0.05	0.0134±0.0038
0.08	0.0074±0.001	0.15	0.0033±0.001	0.2	0.0147±0.0053
0.1	0.0093±0.0012	0.25	0.0088±0.0029	0.3	0.0154±0.0053
0.15	0.0145±0.002	0.3	0.0103±0.003	0.45	0.0178±0.0053
0.2	0.0204±0.0029	0.5	0.0195±0.0061	0.6	0.021±0.0067

Table 2.3: Quantitative testing performance for 2D fish shape point set registration at different deformation level 0.5 in presence of various noise such as Point Drift (P.D) noise, Data Outlier (D.O.) noise, and Data Incompleteness (D.I.) noise.

Experimental Setup: In this experiment, we carry out a set of tests to validate PR-Net’s performance against different types of data noise including P.D. noise, D.I. noise, and D.O. noise. We simulate the noisy data through introducing three types of noise with five different levels to the target point set at deformation level of 0.5. The level of noise is defined in the section of data preparation. The Figure 2.8 illustrates the noisy target point set (in red color) in contrast to the source point set (in blue color). The quantitative result is demonstrated in Table 2.3.

Result: Figure 2.7 demonstrates the PR-Net’s performance with clean data for comparison. Given the source point set (in red color) and target point set (in blue color), PR-Net succeeds in transforming source point set to align with the target one for the clean data.

For investigating PR-Net’s performance on noise data, in Figure 2.8 (A), we apply D.I. noise to target point set by increasingly removing point samples as shown from left to right in a row. The registration results show that our PR-Net is capable of robustly aligning the source point set (red) with target (blue) in this condition. Even for the situation when D.I. noise level is 0.6 and the majority of target shape is missing, PR-Net can still align the remaining parts such as the top

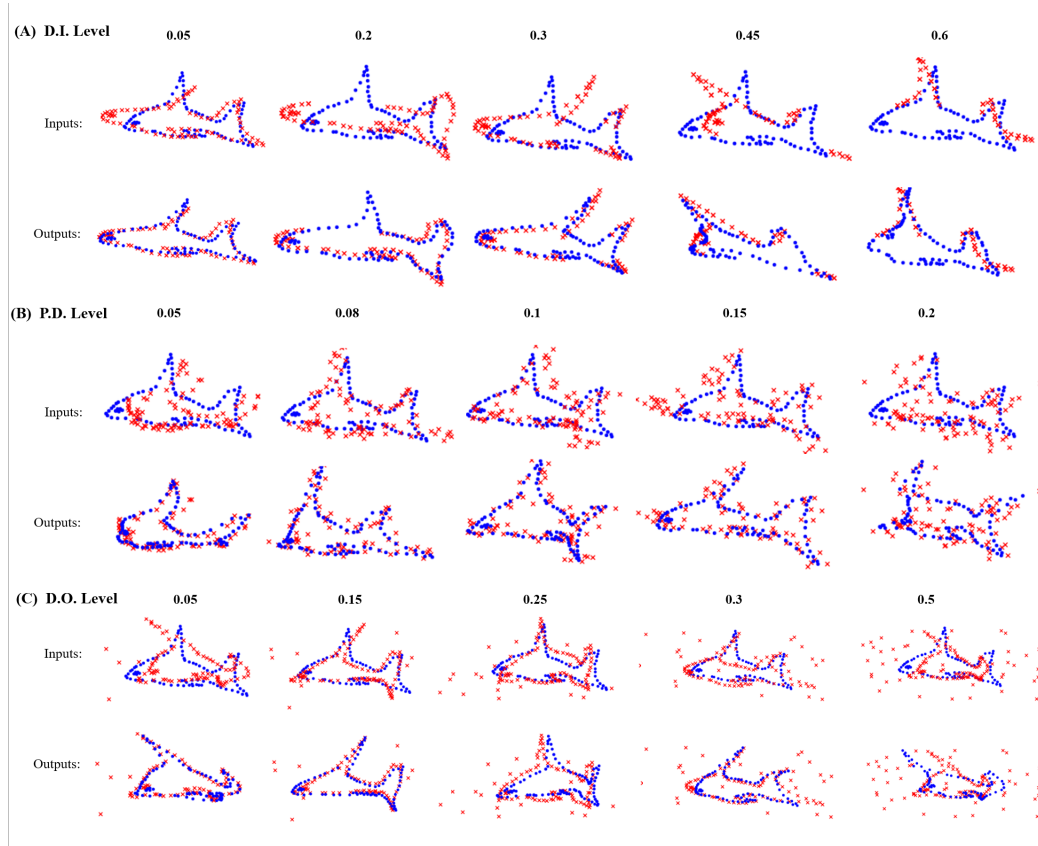


Figure 2.8: Testing results for 2D fish shape point set registration at deformation level 0.5 in presence of various noise. (A) Performance in presence of Data Incompleteness (D.I.) noise. (B) Performance in presence of Point Drift (P.D.) noise. (C) Performance in presence of Data Outlier (D.O.) noise. Blue shapes are source point sets and red ones are target point sets. Please zoom-in for better visualization.

and tail of the target fish. An interesting observation is that for the missing parts, even without any target information, the transformed source point sets seem to be natural and preserve the original geometric meaning. For example when the D.I. level reaches 0.6, the transformed source point sets not only match the targets, but the shape in general still has the geometric meaning for the missing parts and it can be easily recognized as a “fish” shape. As shown in Table 2.3, the quantitative result shows that C.D. linearly increases when D.I. Level increases from 0.05 to 0.6, which indicates PR-Net’s high resistance to D.I. Noise.

The D.O. noises is added to target point set in Figure 2.8 (C) as shown from left to right in a row. The registration result demonstrates that the alignment between the source and target shapes is not significantly affected by outlier points in target set when the D.O. noise level is less than 0.3. The quantitative results show that the C.D. of registered pairs remain as low as 0.0103 when the D.O noise level is 0.3. However, when the D.O. noise level reaches as high as 0.5 the C.D. of registered pairs jumps from 0.0103 to 0.0195, which indicates that PR-Net starts suffering dramatic performance degradation affected by the large amount of added outliers.

2.3.1.5 Results on Data Variety

In this experiment, we take a further step to investigate how well the PR-Net performs point set registration for other 2D/3D shapes at different deformation levels. We are especially interested in point set registration of non-contour based 2D shapes, as well as 3D shapes since the 3D data have been gaining great attention in community with recent advancements in 3D acquisition and computation resources.

Deform. Level	0.3	0.5	0.8	1.0
Hand	0.0013±0.0006	0.0025±0.0013	0.0056±0.0025	0.0105±0.0047
Skeleton	0.0012±0.0005	0.0022±0.0010	0.0081±0.0049	0.0087±0.0047
Skull	0.0017±0.0008	0.0029±0.0011	0.0052±0.0022	0.01±0.0036

Table 2.4: Quantitative testing performance for skull, hand, and skeleton 2D shapes at different deformation level from 0.3 to 1.0.

Experimental Setup: We further conduct tests to verify how well PR-Net performs on the dataset of various shapes and patterns, such as skeleton, skull, hand, face (3D shape) and cat (3D shape). For each type of dataset, with different levels of geometric deformation, we simulate $20k$ point sets as target point sets for training and simulate additional $10k$ point sets for testing. For 3D shapes, we randomly sample points from the mesh data set. While we training PR-Net on 3D shapes, we only sample 512 out of $20K$ points from an input 3D model and $125(5 \times 5 \times 5)$ controlling points for learning descriptor tensor and correlation tensor, which already provided reasonable registration. Due to the computation complexity, there is a clear trade-off between performance with computation efficiency. We randomly select a few samples at deformation level of 0.5 to visualize the alignment result in Figure 2.9 and Figure 2.10. We present the quantitative evaluation of registration in Table 2.4, which presents the mean and standard deviation of C.D. between registered pairs.

In Figure 2.8 (B), we apply P.D. noise to target point set by increasingly adding Gaussian noise as shown from left to right in a row. As shown in Figure 2.8, though the positions of target point sets are dramatically drifted by Gaussian noise, our PR-Net still effectively predicts the desired geometric transformation. Especially when the P.D. noise level is higher than 0.15, even though the boundary of the fish

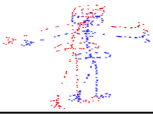
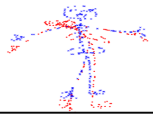
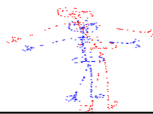
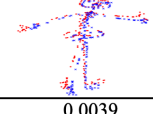
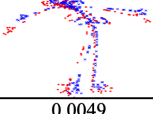
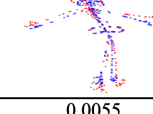

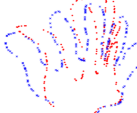



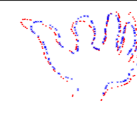


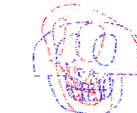
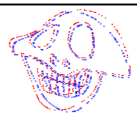
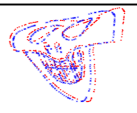
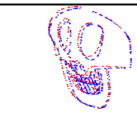
Inputs:			
CD:	0.2189	0.1577	0.0357
Outputs			
CD:	0.0039	0.0049	0.0055
Inputs			
CD:	0.0131	0.0179	0.0195
Outputs			
CD:	0.0037	0.0051	0.0050
Inputs			
CD:	0.0151	0.0145	0.0131
Outputs			
CD:	0.0051	0.0048	0.0019

Figure 2.9: Testing performance for skull, hand and human skeleton shapes. Blue shapes are source point sets and red ones are target point sets. Please zoom-in for better visualization. The corresponding C.D. for each input and output pair is presented below it.

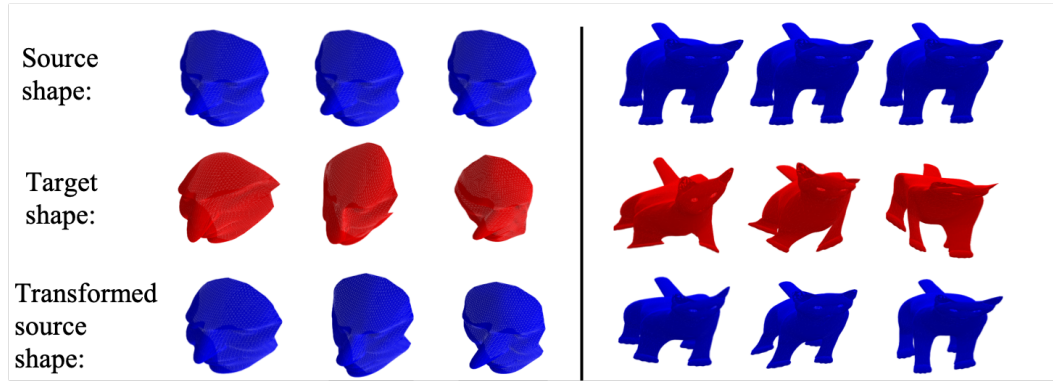


Figure 2.10: Testing registration performance for 3D face and cat point sets. The blue shapes are source shapes and the red shapes are target ones. We plot the mesh of shapes for better visualization.

shape is dramatically drifted, the transformed source shapes have smooth boundary and acceptable alignment with the target ones. From the quantitative results, as shown in Table 2.3, the C.D. of registered pairs is less than 0.01 when the P.D. noise level is under 0.15, which indicates almost perfect alignment. **Result:** PR-Net demonstrates robust registration performance for various categories of 2D shapes (e.g. skull, skeleton and hand), based on the selected examples from the testing dataset are demonstrated in Figure 2.9 and the corresponding quantitative testing results for comparison of these three different shapes are shown in Table 2.4. The decrease in C.D. values from pre to post registration suggests that PR-Net can successfully align deformable pairs of various shapes. As shown in Figure 2.9, for the current deformation level 0.5, PR-Net shows robust performance regarding to different shapes. There is no obvious difference among the registration results of them. When zooming in for a more detailed observation, the missing registration part can still be noticed such as the upper line of the skull in row 2. As shown in Table 2.4, for comparing the quantitative results on these three shapes, the result on Skull Shape is slightly worse than other two shapes when

deformation level is low. But for higher deformation level, the performance on Skull shape becomes comparative to other two shapes. This validates the robust performance of PR-Net towards non-rigid point set registration over a variety of shapes in presence of different geometric deformation level. In Figure 2.10, we demonstrate that PR-Net is applicable for 3D point set registration. As shown in Figure 2.10, for the general part of the target shape, our model can correctly predict the registration transformation to align them. As to aligning the more subtle part of source and target point sets, there is still space to improve PR-Net's performance. The straightforward method to improve the performance is to increase the number of sampling points from surface and as well as the controlling points for learning the shape descriptor tensor with acceptable computation cost. The comparison result across different categories of shapes indicates the consistent performance of PR-Net.

2.3.2 CPD-Net

In this section, we carry out a set of experiments for non-rigid point set registration and assess the performance of our proposed CPD-Net. In section 1.3.2.1, we describe the details of datasets that are used for training and testing of CPD-Net. We report the experimental results to evaluate the performance of our trained CPD-Net on 2D and 3D datasets in section 1.3.2.2 and 1.3.2.3. Section 1.3.2.4 discusses the resistance of CPD-Net to various types of noise. In section 1.3.2.5, we compare CPD-Net with non-learning based method.

2.3.2.1 Experimental Dataset

A variety of different 2D and 3D shapes (i.e examples shown in Figure 2.11, Figure 2.14, and Figure 2.15) are used in the experiments to train and test the CPD-Net. In experiments, we prepare the dataset as follows:

- To prepare the deformable shape dataset (as shown in the first column row of Figure 2.11), we simulate non-rigid geometric transformation on the normalized raw point sets by thin plate spline (TPS) [21] transformation with different deformation levels. The deformation level is defined as the perturbing degree of controlling points in TPS. Specifically, given the deformation level set at l (e.g. 0.5), a Gaussian random shift with zero-mean and $2l$ standard deviation is generated to perturb the controlling points.
- To prepare the Gaussian Displacement (G.D.) noise dataset (as shown in the first row of Figure 2.17), we simulate the random displacement superimposed on a deformed point set (deformation level at 0.5) by applying an increasing intensity of zero-mean Gaussian noise. The G.D. noise level is defined using the standard deviation of Gaussian.
- To prepare the Point Outlier (P.O.) noise for the shape (as shown in the second row of Figure 2.17), we simulate the outliers on the deformed point set (deformation level at 0.5) by adding an increasing number of Gaussian outliers. The P.O. noise level is defined as a ratio of Gaussian outliers and the entire target point set.
- To prepare the Data Incompleteness (D.I.) noise (as shown in the second row of Figure 2.17), we remove an increasing number of neighbouring points from

target point set (deformation level at 0.5). The D.I. noise level is defined as the percentage of the points removed from the entire target point set.

2.3.2.2 2D non-rigid point set registration

CPD-Net can generalize from training towards the prediction of geometric transformation for aligning unseen testing point sets in an unsupervised fashion. In this experimental section, we demonstrate the point set registration performance of the CPD-Net on various categories of 2D shapes at different deformation levels.

Experiment Setting: In this experiment, we use four different type of 2D shapes (i.e. fork, face.) to prepare the dataset. For each shape, we first synthesize a set of $20k$ deformed shapes at each deformation level. The deformation level ranges from 0.3 to 2.0. To prepare the training dataset, for each type of shape at each deformation level, we split synthesized dataset into two groups. We randomly choose a pair of shapes from group one to form $20k$ pairs of training. Similarly, we randomly choose two shapes from the other synthesized dataset to form $10k$ testing pairs. Note that there is no intersection between training and testing datasets. To evaluate the registration performance, we use the Chamfer Distance (C.D.) between the transformed source point set and target one as quantitative assessment, and we visualize the pairwise point sets before and after registration for qualitative assessment. We conduct two tests based on the shapes we used in the test. For test one, we use the commonly adopted fish shape, and in test two, we use other 2D shape categories, such mushroom, face, and fork, to assess CPD-Net’s performance, particularly on the non contour-based shapes.

Deform. Level	Inputs	Predicted Drifts	Zoomed-in Pred. Drifts	Outputs
0.1				
0.3				
0.5				
0.9				
1.5				

Figure 2.11: The qualitative registration result for Fish shape at different deformation level. The blue shape is target point set. The red shape is source point set. The black lines are predicted coherent drifts for source point set. Please zoom-in for better visualization.

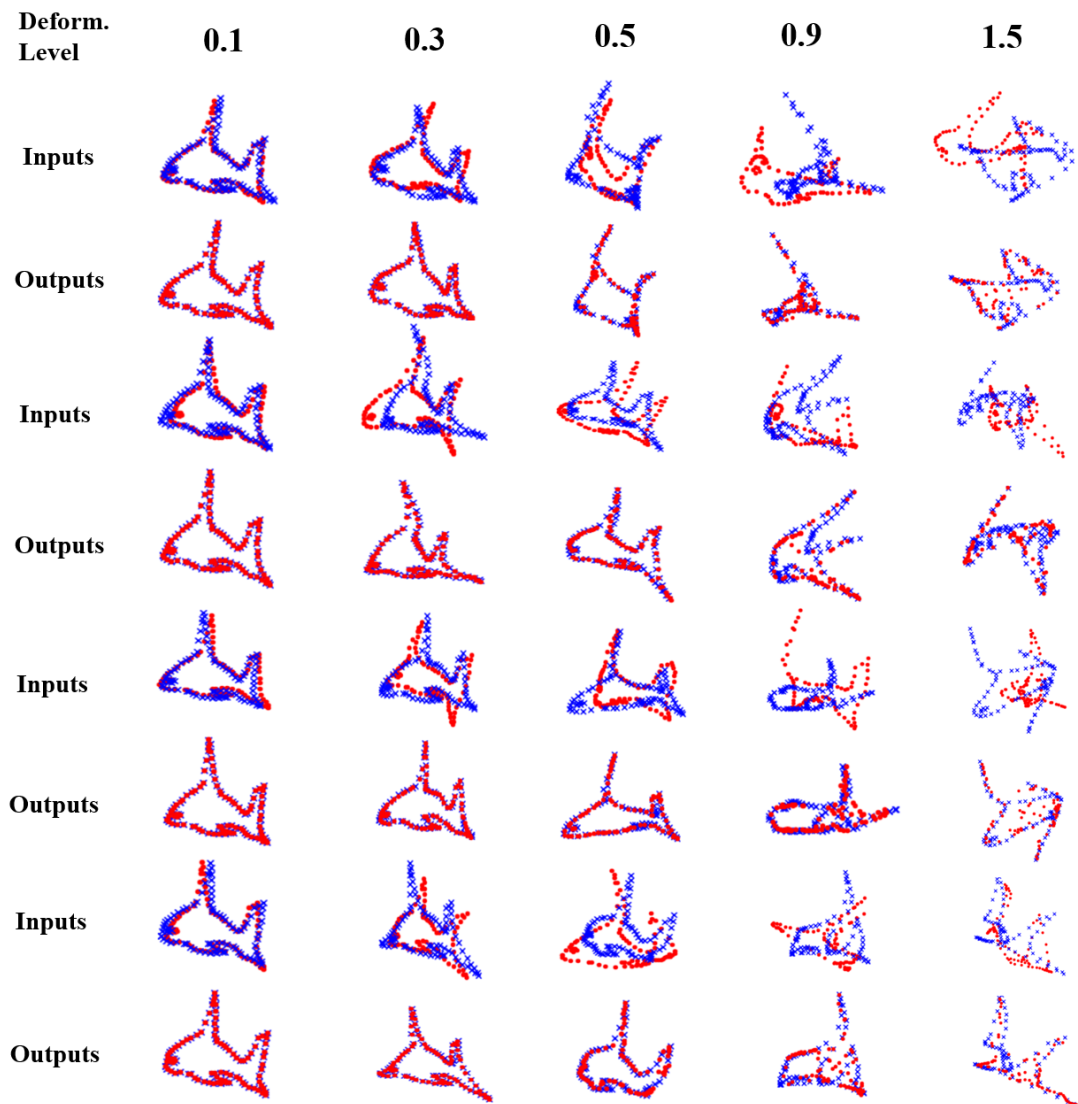


Figure 2.12: The testing qualitative registration results for Fish shape at different deformation level. The blue shape is target point set. The red shape is source point set. Please zoom-in for better visualization.

Results of Test 1: After training CPD-Net, we applied the trained model to testing dataset prepared as described above. In Figure 2.11, the first column illustrates the pair of source point set and target one before registration, where red shape denotes source point set and blue one denotes the target set. The second column illustrates the predicted drift vectors (depicted by the black arrow) by our trained CPD-Net for each point in source point set. The third column is zoom-in view of predicted drift vectors in a focused region. The fourth column shows the registered pairs of transformed source set and target one after registration. Shapes from different rows have different deformation levels from 0.1 to 1.5. From the fourth column, we can observe that CPD-Net can predict nearly perfect registration when the deformation level is smaller than 0.9. While we increase the deformation to the level greater than 1.5, the source and target point sets have significant shape structural variation, which dramatically increase the difficulty of point set registration. However, CPD-Net can still reliably transform the source point set to align the main portion of the shape of the target point set. In addition, it is interesting to observe from the second and third columns that source point set moves coherently as a whole towards the target one. This observation verifies that CPD-Net is able to predict a continuous smooth displacement field without necessity to impose additional coherence constraint term. We further provide the mean and standard deviation of C.D. calculated from the 10K testing pairs at each deformation level for quantitative evaluation. In the Figure 2.13, we plot the mean and standard deviation for the set of C.D. between the source point set and the target one at all deformation levels. We can see from the comparison that the red curve (post registration) is consistently below the blue one (pre registration), which indicates CPD-Net is able to robustly register the source and target point

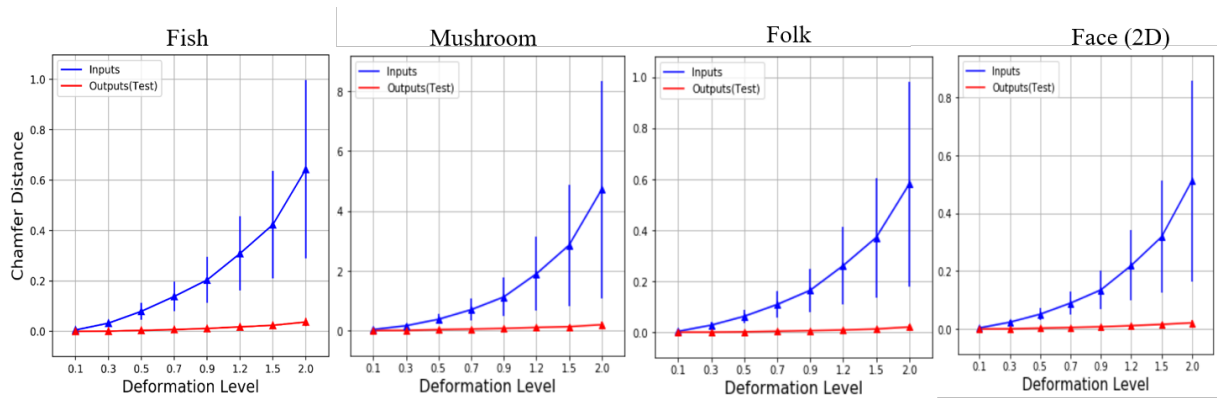


Figure 2.13: The C.D. between source and target point sets, pre (blue line) and post (red line) registration.

Def. level	0.3	0.5	0.7
Fish	0.0008±0.0004	0.0037±0.0009	0.0072±0.0022
Mushroom	0.0006±0.0003	0.0031±0.0009	0.0051±0.00184
Fork	0.0002±0.0001	0.0014±0.0011	0.0038±0.0019
Face (2D)	0.0005±0.0003	0.0028±0.0011	0.0053±0.0017
Def. level	1.2	1.5	2.0
Fish	0.0178±0.0069	0.0239±0.0096	0.037±0.016
Mushroom	0.0103±0.0045	0.0129±0.0058	0.0196±0.0094
Fork	0.0089±0.0048	0.0126±0.0074	0.0203±0.0127
Face (2D)	0.0114±0.0049	0.0158±0.0074	0.0213±0.01

Table 2.5: Quantitative testing performance for 2D point set registration.

set with a small C.D. Moreover, the red curve stays nearly flat as the deformation increases from 0.1 to 2.0, which indicates that CPD-Net’s robust performance at high deformation level. The detailed qualitative result is presented in Table 2.5.

Results of Test 2: In this test, we further use CPD-Net to perform the non-rigid registration on other three types of 2D shapes including Mushroom, Fork, and Face shapes as shown in Figure 2.14. To visualize the registration result, we compare the pair of testing shapes before and after registration at deformation level 0.5 as shown in Figure 2.14. All randomly selected samples show nearly perfect


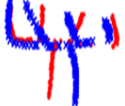


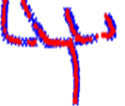

	Mushroom	Fork	Face (2D)
Inputs:			
C.D.	0.0018	0.0051	0.0102
Outputs:			
C.D.	0.0002	0.0001	0.0002

Figure 2.14: Registration examples for Mushroom, Fork and Face shapes. The blue shape represents target and the red shape represents source point set. The corresponding C.D. score is listed underneath the registered point sets.

registration. Similar to test 1, we present quantitative evaluation using C.D. metric for the non-rigid registration of those three types of shapes in the Table 2.5. Each row contains the mean and standard deviation of C.D. measurement for all testing pairs of shapes at deformation level from 0.3 to 2.0. Based on the quantitative results shown in the Figure 2.13, for all the four shapes, CPD-Net demonstrates the remarkable performance of non-rigid registration as evidenced by the fact that the C.D. is dramatically reduced and consistently stays low after alignment at all deformation levels. Especially after the deformation level increases to 1.5 when the shape structure has been dramatically deteriorated (as shown on the last row of Figure 2.13).

2.3.2.3 3D non-rigid point set registration

In this experiment, we take a further step to investigate how well the CPD-Net performs 3D point set registration at different deformation levels since the 3D data have been gaining great attention in community with recent advancements in 3D

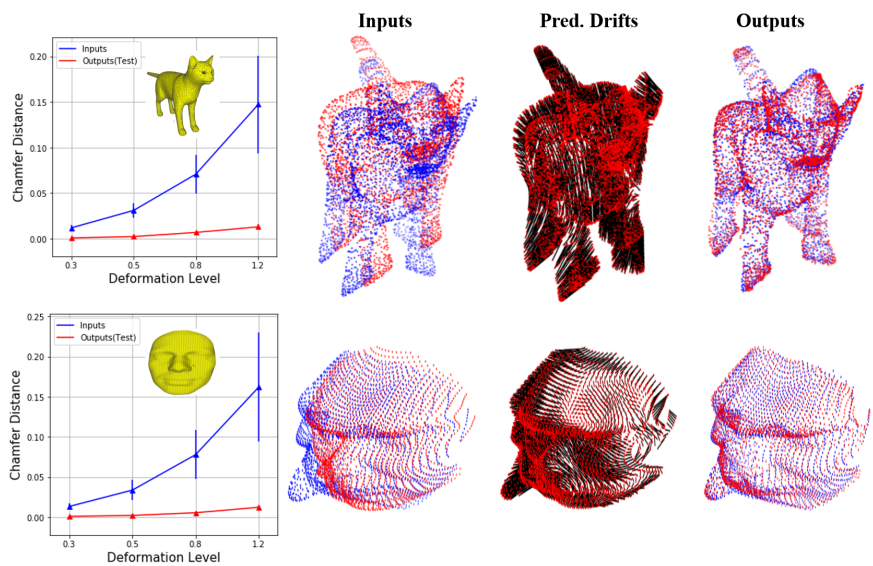


Figure 2.15: The charts show C.D. between source and target point sets, pre (blue line) and post (red line) registration in left. Selected qualitative registration results are demonstrated in right. The red points represents the source points and the blue ones represent the target points. The black lines represent the predicted drifts for source point set. Please zoom-in for better visualization.

acquisition and computation resources.

Experimental Setting: In this experiment, we use two categories of 3D shapes (i.e. 3D Face and 3D Cat) to prepare the dataset. Similar to 2D data preparation, we synthesize $10k$ training pairs of 3D shapes and $2k$ testing pairs for both 3D Face and Cat shapes at various deformation levels (0.3, 0.5, 0.8, and 1.2). We use the same measurement methods for both qualitative and quantitative results (as shown in Figure 2.15).

Result: In Figure 2.15, we illustrate the quantitative evaluation curves on the left and visualize one registration cases at deformation level 0.3 on the right. For both 3D cat and face shapes, CPD-Net demonstrates impressive performance with

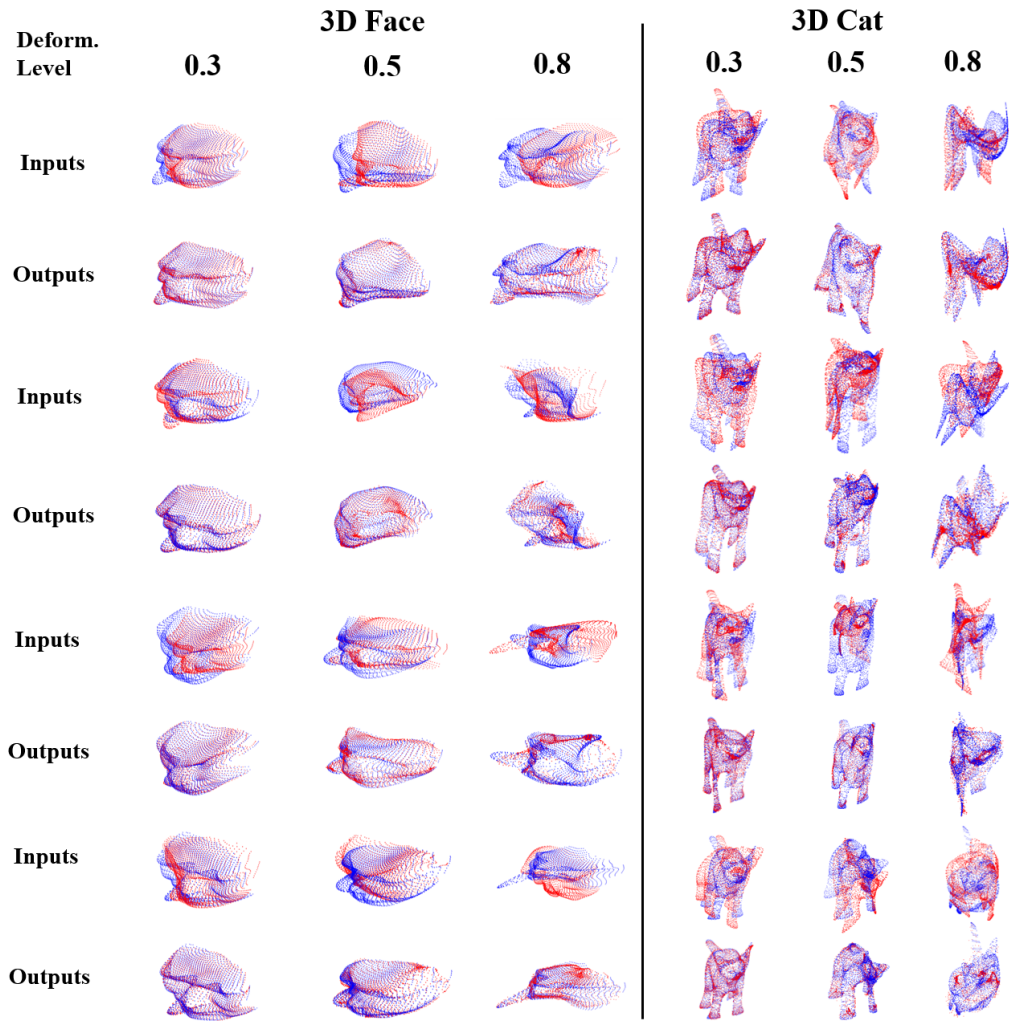


Figure 2.16: The testing qualitative registration results for 3D shapes at different deformation level. The red points represents the source points and the blue ones represent the target points. Please zoom-in for better visualization.

a quite small pairwise C.D. after registration, and the C.D. measurement remains consistently low while we increase the deformation level. At the level of 1.2, the mean of C.D. between source and target point sets is nearly 10 times less after alignment, which indicates that the trained CPD-Net is able to align most portion of the shapes between a source and target point sets. To further visualize the registration performance, on the right side of Figure 2.15, we show the registration results by plotting the figures for 3D source and target point sets, pre and post registration in first and third columns and the second column depicts the predicted coherent point drift vector. In the Figure, the first column depicts the shape pair prior to registration, the second column depicts the coherent point drift vector for each point on the source point set, the third column depicts the shape pair after registration, and the remain column show the mesh surface for the 3D shapes for better visualization effects. Those plots clearly prove that CPD-Net is able to predict an accurate smooth non-rigid transformation. The registration performance is particularly impressive to find the accurate non-rigid registration after the deformation level is greater than $l = 0.8$, when the 3D structure of the shape objects are significantly deteriorated to a degree that human has hard time to register the pair of 3D objects.

2.3.2.4 Resistance to Noise

While using the sensors such as LIDAR sensor and laser scanner, it is unavoidable that the data might be acquired with a variety types of noises. An effective non-rigid registration method should be robust to those noise in addition to the structural variations as discussed in previous section. Therefore, in this section, we focus on testing how well CPD-Net can predict the non-rigid registration from

the noisy dataset.

Experimental Setting: In this section, we use fish shape data at deformation level of 0.5 to prepare the experimental noisy dataset. We simulate three types of noise (i.e. Gaussian Displacement (G.D.) noise, Point Outlier (P.O.) noise and Data Incompleteness (D.I.) noise). For each type, we gradually increase the level of noise added to the deformed target point set of fish dataset as shown in Figure 2.17). We prepare $10k$ pairs of source and target point sets for each type of noise for testing. As in previous section, the same quantitative and qualitative performance measurement is used in this experiment.

Result: In this section, we illustrate the experimental result using the C.D. as the quantitative metric, and plot one pair of source and target point sets at different noise level, pre and post registration. All experimental results are listed in Figure 2.17. As shown in the Figure 2.17, we plot the quantitative evaluation curves on the left and visualize five registration cases at different noise level on the right. For the clean data, the mean of C.D. for the fish shape at deformation level of 0.5 is around 0.08. We need to validate if CPD-Net can significantly reduce the C.D. for pairs of source and target point sets of noisy dataset after registration, and if CPD-Net consistently keep the C.D. comparatively low when the noise level increases.

For the G.D. noise, in the Figure 2.17 the first row depicts the registration by our CPD-Net for the G.D. noise corrupted data. As we notice the plot, the C.D. after registration remains constantly lower than 0.08, even when the G.D. noise increases to the level of 0.2. CPD-Net can still predict the non-rigid transforma-

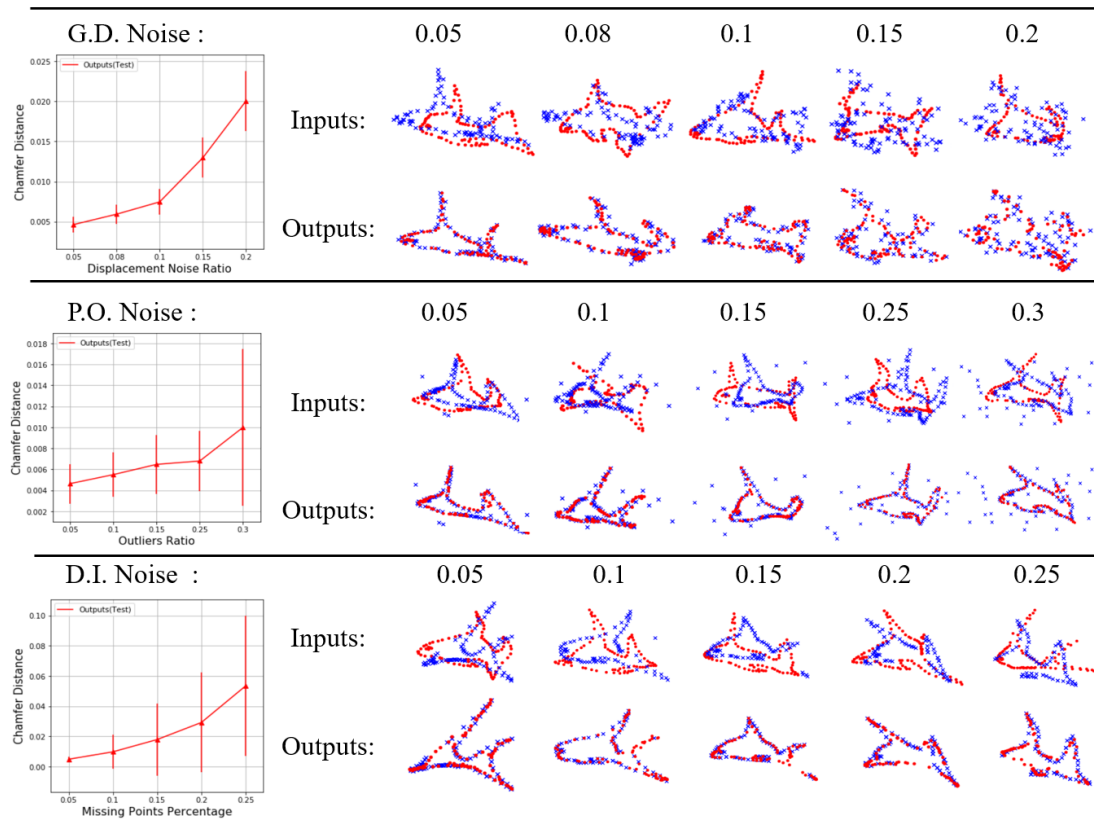


Figure 2.17: The charts of C.D. between transformed source point set and target one in presence of different level of G.D. noise, P.O. noise and D.I. noise are shown in left. The selected qualitative results are demonstrated in right. The red shape represents the source point set and the blue one represents the target point set. Please zoom-in for better visualization.

tion to align the source set (the red points) to the target one (the blue points) with a relatively small C.D. between two point sets, even though the shape was dramatically altered by the Gaussian Displacement noise. The shape was dramatically altered by the Gaussian noise which makes it difficult to recognize overall shape as a fish. However, our CPD-Net can still predict the non-rigid registration to align the source set (the red points) to the target one (the blue points) with a relatively small C.D. between two point sets. For the P.O. noise, as shown in the second row, outlier points is increasingly added to the target point set (blue ones) from left to right in a row. Different from the P.O. noise, we would like to check if CPD-Net is able to successfully ignore those outlier points to contribute the registration process. The registration result is impressive that the main body of the source and target shapes can robustly aligned to each other with a small C.D. between them after registration, especially when the P.O. noise level reaches as high as 0.3. For D.I. noise, as shown in the third row, an increasing number of points is removed from the target point set (blue ones) to check if CPD-Net is able to successfully align the source point set to the incomplete portion of the target one. The visualization of pairwise registration result in the third row clearly demonstrates that CPD-Net is able to robustly align the source point set (red) to the incomplete target point set (blue). When D.I. noise level reaches 0.25, the missing part is aligned with a straight line, which is less desired. But the aligned portions from transformed source point set and target one show consistent common geometric structure, which is not affected by the missing portion of the target point set.

Methods	CD	Time
CPD [3] (Train)	0.0039 ± 0.0032	~ 22 hours
Ours (Train)	0.0035 ± 0.0008	~ 25 minutes
CPD [3] (Test)	0.0039 ± 0.0033	~ 22 hours
Ours (Test)	0.0037 ± 0.0009	~ 15 seconds

Table 2.6: Performance and Time comparison with CPD.

2.3.2.5 Comparison to CPD

Different from previous efforts, the proposed CPD-Net is a learning-based non-rigid point set registration method, which can learn the registration pattern to directly predict the non-parametric geometric transformation for the point sets alignment. As a learning-based approach to predict the non-rigid registration, it is not applicable to have a direct comparison between CPD-Net and other existing non-rigid iterative registration methods. To compare our method to non-learning based iterative method (i.e. Coherent Point Drift (CPD) [3]), we design the experiment as follows to assess both time and accuracy performance.

Experimental Setting: In this experiment, we use the fish shape at deformation level of 0.5 to prepare the dataset. We synthesize $20k$ pairwise source and target point sets to form the training set, and another $20k$ pairs to form the testing set. The CPD-Net is firstly trained with the $20k$ training dataset. The trained CPD-Net is then applied to directly predict registration for the $20k$ testing dataset. In contrast, CPD is directly applied to both $20k$ training and testing dataset.

Result: The C.D. based quantitative comparison is presented in the Table 2.6. The first and third row list the experimental results for CPD on training and

testing dataset respectively. The second row lists the results for the CPD-Net on training dataset, and the fourth row lists the results for the trained CPD-Net on the testing dataset. Based on the comparison between first and third rows, we can see that our model can achieve better performance (i.e. smaller C.D.) than that of CPD within a shorter time (25 minutes versus 22 hours) to align $20k$ pairs. Unlike the CPD needs to start over a new iterative optimization process to register a new pair of shapes independently, CPD-Net actively learns the registration pattern from training and consequently become capable of handling real-time point set registration or a large volume dataset by direct predicting the geometric transformation. As shown on second and fourth row of the Table 2.5, we notice that the trained CPD-Net is able to achieve nearly the same training performance, which indicates that CPD-Net has great generalization capability. The trained CPD-Net is able to achieve better performance than CPD on the same dataset with orders of magnitude less time (15 seconds versus 22 hours).

Chapter 3

Learning-based Point

Correspondence Networks

(This chapter is submitted as paper “PC-Net: Unsupervised Point Correspondence Learning with Neural Networks” with Dr.Xiang Li et al. as co-authors under review.)

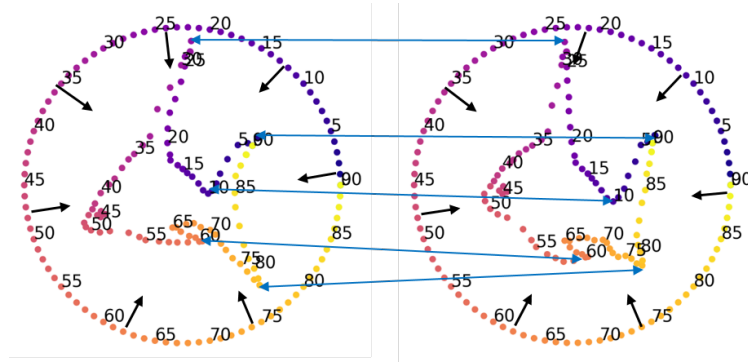


Figure 3.1: Illustration of our unsupervised point correspondence. Our model drifts all landmark points of a template circle to match the corresponding positions of target shapes.

In this chapter, we introduce our next network PC-Net for unsupervised point correspondence. Point sets correspondence concerns with the establishment of point-wise correspondence for a group of 2D or 3D point sets with similar shape description as shown in Fig 3.1. Existing methods often iteratively search for the optimal point-wise correspondence assignment for two sets of points, driven by maximizing the similarity between two sets of explicitly designed point features or by determining the parametric transformation for the best alignment between two point sets. In contrast, without depending on the explicit definitions of point features or transformation, this chapter introduces a novel point correspondence neural networks (PC-Net) that is able to learn and predict the point correspondence among the populations of a specific object (e.g. fish, human, chair, etc) in an unsupervised manner. Specifically, in this chapter, we first develop an encoder to learn the shape descriptor from a point set that captures essential global and deformation-insensitive geometric properties. Then followed with a novel motion-driven process, our PC-Net drives a template shape, that consists of a set of landmark points, morph and conform around a target shape object which is recon-

structured through decoding the previously characterized shape descriptor. As a result, the motion-driven process progressively and coherently drifts all landmark points from the template shape to corresponding positions on the target object shape. The experimental results demonstrate that PC-Net can establish robust unsupervised point correspondence over a group of deformable object shapes in the presence of geometric noise and missing points. More importantly, with great generalization capability, PC-Net is capable of instantly predicting group point corresponding for unseen point sets.

Figure 3.2 illustrates pipeline of the proposed PC-Net which is composed of four main components. The first component is “learning global shape descriptor“. In this component, the global shape descriptor is learned with a deep neural network to capture global geometric properties. The second component is “forming shape morphing initiator“. In this component, a circle or sphere is selected as a template shape that consists of a set of landmark points (i.e. points preserving correspondence between the object and its population). The shape morphing initiator is vector array with each element represented as a vector concatenation of the coordinate of each landmark point with the global shape descriptor. The third component is “Motion-driven Embedding“. In this component, landmarks of a template shape morph and conform towards the target shape, guided by the previously characterized shape descriptor. As a result, all landmarks are progressively and coherently drifted from the template shape to corresponding positions on the target shape. In the last component, we map the correspondence of reconstructed landmarks back to the original point sets. Accordingly, the main components of the pipeline indicate the main contributions of our proposed PC-Net. We propose a novel concept of “shape morphing initiator” that is a representation of

a combination of landmark coordinates and learned shape descriptors, which introduces a topology constraint of landmarks on learned shape descriptors. We propose a “motion-driven embedding” module. This unsupervised self-morphing process progressively and coherently drifts all landmark points from the template shape to corresponding positions on the target object shape without using ground truth labels. We introduce a novel learning-based point correspondence paradigm which can establish the point correspondence among two or more groups of point sets. With generalization ability, PC-Net is able to directly predict the point correspondence on the testing dataset without a new iterative searching process.

3.1 Methods

In this section, we introduce our method for unsupervised point correspondence. The overall framework is illustrated in Figure 3.2. In Section 3.1.1, we state the unsupervised learning-based correspondence problem. Section 3.1.2 introduces our reconstruction structure of four successive parts: learning shape descriptor, formulating shape morphing initiator, motion-driven embedding process and the loss function. In section 3.1.3, we illustrate the correspondence mapping process from reconstructed shapes to input shapes.

3.1.1 Problem statement

Prior to the detail discussion of our approach, we first formally define the point correspondence task as follow. Given a set of point clouds $\mathbf{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_n\}_{n \geq 2}$. $\mathbf{P}_i = \{\mathbf{x}_1^i, \dots, \mathbf{x}_M^i, \mathbf{y}_1^i, \dots, \mathbf{y}_{N_i}^i\}$, is a point set in \mathbf{P} , where $\mathbf{x}_j^i, \mathbf{y}_t^i \in \mathbb{R}^n$ ($n = 2$ or $n = 3$). $\forall \mathbf{P}_i \in \mathbf{P}$, we assume that there exists a subset $\mathbf{C}_i = \{\mathbf{x}_1^i, \dots, \mathbf{x}_M^i\} \subset \mathbf{P}_i$ which

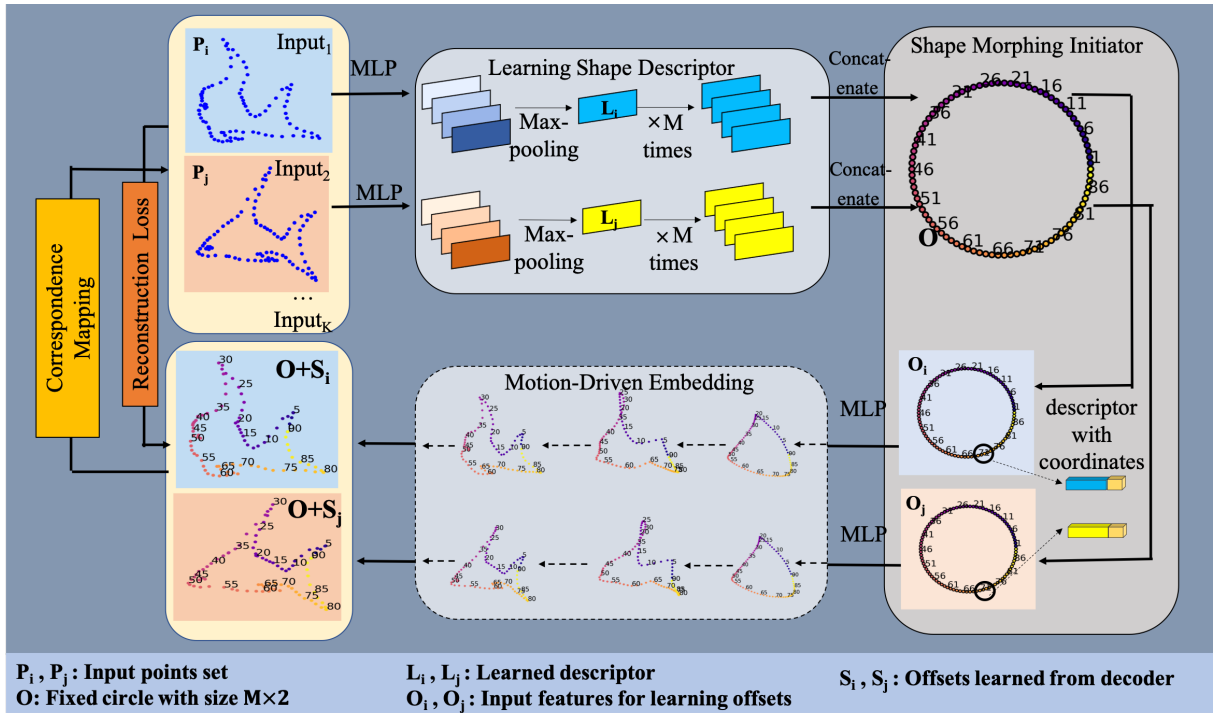


Figure 3.2: The pipeline of proposed PC-Net model. The pipeline mainly includes four parts. The first part is an encoder to learn the “global shape descriptor” from a point set that captures essential global and deformation-insensitive geometric properties. The second part is forming “shape morphing initiator” and the third component is “Motion-driven Embedding” for reconstruction. The fourth component is “Point Correspondence Mapping” to map the correspondence of reconstructed landmarks back to the original point sets.

includes all corresponding points in \mathbf{P} . Our task is to match all the corresponding point sets $\{\mathbf{C}_k | \mathbf{C}_k \subset \mathbf{P}_k\}$. In other words, $\forall \mathbf{x}_j^k \in \mathbf{C}_k$, we assume that there exists a learning structure \mathcal{M} such that, $\forall \mathbf{C}_i (i \neq k)$,

$$\mathbf{x}_j^i = \mathcal{M}_\gamma(\mathbf{x}_j^k, \mathbf{C}_i | \gamma, \mathbf{D}) \quad (3.1)$$

, where $\mathbf{D} \subset \mathbf{P}$ is a given training dataset; γ represents all the parameters; $\mathbf{x}_j^i \in \mathbf{C}_i$ is a corresponding point to \mathbf{x}_j^k . Therefore, for the point \mathbf{x}_j^k , we have all its predicted corresponding points in the set $\{\mathbf{x}_j^1, \dots, \mathbf{x}_j^m | \mathbf{x}_j^i \in \mathbf{P}_i\}$. For unsupervised learning model, we do not use the ground truth corresponding information during the training process.

3.1.2 Reconstruction pipeline

In this section, we introduce the reconstruction pipeline of our model to embed a template of landmarks to reconstruct the input shape. This process is the key part for correspondence prediction, which includes four modules: learning shape descriptor, formulating shape morphing initiator, motion-driven embedding process and Chamfer loss.

Learning shape descriptor. For a given input point set, we first learn a shape descriptor that captures representative and deformation-insensitive geometric features. Let $\mathbf{P}_i \in \mathbf{P}$ denotes the input point sets and $\mathbf{L}_i \in \mathbb{R}^m$ denotes the shape descriptor learned from the input \mathbf{P}_i . To address the problem of irregular format of point set, we introduce the following encoding network F_1 , which includes three successive multi-layer perceptrons (MLP) f_1 , f_2 and f_3 , such that: $f_1 : \mathbb{R}^2 \rightarrow \mathbb{R}^{64}$,

$f_2 : \mathbb{R}^{64} \rightarrow \mathbb{R}^{128}$, $f_3 : \mathbb{R}^{128} \rightarrow \mathbb{R}^{512}$. The encoder network F_1 is defined as: $\forall \mathbf{P}_i \in \mathbf{P}$,

$$F_1(\mathbf{P}_i) = \text{Maxpool}\{f_3 f_2 f_1(\mathbf{x}_i)\}_{\mathbf{x}_i \in \mathbf{P}_i} \quad (3.2)$$

We use the Maxpool function to extract the order-invariant descriptors from the input point set. The readers can refer to PointNet [30] for detailed discussion. One can also use other symmetric operators such as summation, average pooling function and so on. This structure can be easily adapted for 3D point set inputs. Other point signature learning architecture such as PointNet++ [38] can be easily implemented in our model as well.

Shape morphing initiator. In this part, our shape morphing initiator is formulated as a combination of landmark coordinates and learned shape descriptors. We use a circle/spherical surface as a template of landmarks for 2D/3D shapes. Taking the 2D point sets for example, assuming that we have k landmarks uniformly distributed on the template circle, $\mathbf{O} = \{(r \cos(\frac{2(i-1)\pi}{k}), r \sin(\frac{2(i-1)\pi}{k}))\}_{i=1,2,\dots,k}$.

We further concatenate the learned descriptor with each landmark point of the template. $\forall \mathbf{P}_i \in \mathbf{P}$, we have its shape descriptor $\mathbf{L}_i = F_1(\mathbf{P}_i) = (l_1^i, \dots, l_{512}^i) \in \mathbb{R}^{512}$. We denote the shape morphing initiator $\mathbf{O}_i = \{(r \cos(\frac{2(p-1)\pi}{k}), r \sin(\frac{2(p-1)\pi}{k}), l_1^i, \dots, l_{512}^i)\}_{p=1,2,\dots,k}$ as our input to the motion-driven embedding network. On one hand, the shape descriptor decides the movement of landmarks during the embedding process. On the other hand, this template circle provides a type of topology constrain on the learned descriptor.

Motion-driven embedding process. We introduce motion-driven embedding

process to learn the offsets (we denote \mathbf{S}_i) of landmark points based on its shape morphing initiator \mathbf{O}_i to further reconstruct the target shape. We denote the motion-driven embedding network $F_2 : \mathbf{O}_i \rightarrow \mathbf{S}_i$. F_2 includes three successive multi-layer perceptrons g_1 , g_2 and g_3 , such that $g_1 : \mathbb{R}^{514} \rightarrow \mathbb{R}^{256}$, $g_2 : \mathbb{R}^{256} \rightarrow \mathbb{R}^{64}$, $g_3 : \mathbb{R}^{64} \rightarrow \mathbb{R}^2$. We define F_2 as: $\forall \mathbf{P}_i \in \mathbf{P}$, we have \mathbf{O}_i such that,

$$F_2(\mathbf{O}_i) = g_3(g_2(g_1(\mathbf{O}_i))) \quad (3.3)$$

, where $\mathbf{S}_i = F_2(\mathbf{O}_i)$ is the learned offsets for landmark points. The reconstructed shape $\mathbf{P}'_i = \mathbf{O} + \mathbf{S}_i$. Thanks to the continuity of function F_2 , the local topology of landmark points remains unchanged during the embedding process. In this way, our motion-driven process coherently drift all landmark points to the target position without changing their local topology.

Chamfer Loss. In this chapter, we adopt the Chamfer Loss to compare our reconstructed point clouds $\mathbf{P}' \subset \mathbb{R}^2$ and input point clouds $\mathbf{P} \subset \mathbb{R}^2$ as:

$$\begin{aligned} L_{\text{Chamfer}}(\mathbf{P}, \mathbf{P}' | \gamma) &= \sum_{x \in \mathbf{P}'} \min_{y \in \mathbf{P}} \|x - y\|_2^2 \\ &+ \sum_{y \in \mathbf{P}} \min_{x \in \mathbf{P}'} \|x - y\|_2^2 \end{aligned} \quad (3.4)$$

, where γ represents all network parameters to be optimized.

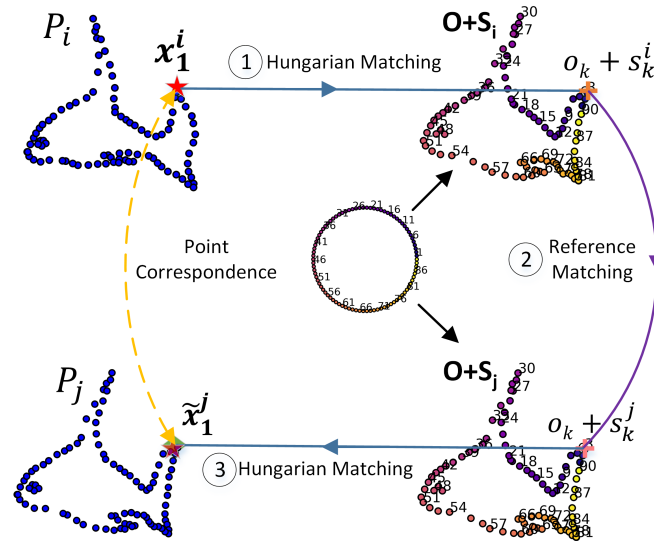


Figure 3.3: Point Correspondence procedure with the Hungarian matching algorithm. The Left part shows the input point sets, and the right part shows their reconstructed shapes from a template circle.

3.1.3 From reconstructed results to unsupervised correspondences.

After network training, our proposed model produces high-quality shape reconstructions. Although our training process does not receive any explicit supervision for shape correspondence, the self-morphing process from a template circle inherently leads to the natural shape correspondence on reconstructed shape since all the reconstructed shapes are embedded from the same circle of landmarks. The inherent point correspondences provide a reference for the desired correspondence generation. We can turn this correspondence relationship back to the original shape pairs by filling the gap between each shape and its reconstruction. In this chapter, we use the Hungarian matching algorithm to build the link between each point set with its reconstructed shape.

The overall matching procedure is shown in Figure 3.3. For better understanding, we illustrate the process of finding the correspondence point \tilde{x}_1^j in shape \mathbf{P}_j for the referring point x_1^i in shape \mathbf{P}_i ($(j \neq i)$). Firstly, through the Hungarian matching algorithm, we find the matching point $o_k + s_k^i$ in its reconstruction $\mathbf{O} + \mathbf{S}_i$. Secondly, we use the point index k to get its corresponded point $o_k + s_k^j$ in $\mathbf{O} + \mathbf{S}_j$. Since all the reconstructed shapes are embedded from the same template circle, these two corresponded points have the same number in the circle \mathbf{O} . Finally, we use the Hungarian matching algorithm again to find the point back in \mathbf{P}_j from $o_k + s_k^j$ in its reconstructed shape, and generate the final correspondence point \tilde{x}_1^j in shape P_j .

3.2 Experimental Results

Section 3.2.1 describes our dataset and the implementation details. In section 3.2.2, we analyze the embedding and unsupervised correspondence mechanism by examples. We test the robustness of model under deformation, noise and missing points in section 3.2.3. We show the performance of our model on 3D point sets in section 3.2.4.

3.2.1 Dataset and implementation Details

In the experiments, we use a 2D point set of a fish shape which contains 91 points for demonstrating the effectiveness of our proposed method. Following [1], the thin-plate spline (TPS) [39] transformation with a uniform 3x3 grid of control points is adopted to generate our synthetic target point sets. The target point sets at different deformation levels are simulated by perturbing the controlling points

using various levels of numerical shifting, followed by a smooth 2D interpolation. Specifically, given the deformation level k , a Gaussian random shifting vector with zero-mean and $2k$ variance is generated to perturb the controlling points. We follow the same setting to synthesize the testing dataset which is not used for training.

To evaluate the correspondence performance, we randomly pick out 100 shapes from the training/test dataset and evaluate the pair-wise correspondence among all pairs. For each pair, one is used as a reference shape, and the other one is regarded as the target shape for evaluation. The final correspondence accuracy is calculated as an average over all shape pairs.

Our network is optimized using Adam optimizer with an initial learning rate of 0.001. We decay the learning rate by 0.995 every 100 steps, with a minimum value of $1e-6$. We set batch size to 32, momentum to 0.9, and weight decay to $1e-5$. We used leaky-ReLU [35] activation function and applied batch normalization [37] to each convolution layers except the output layer. We implemented our method with Tensorflow Library on a Tesla K80 GPU.

3.2.2 Illustration of Motion-Driven Process

In this section, we look into detailed steps of the motion-driven embedding process to better understand the mechanism of the coherent drift of the landmark points from a template shape (i.e. circle) to the corresponding positions of the target shape during the training process. As shown in Figure 3.4, three fish shapes are randomly selected from the synthesized training dataset with a deformation level of 0.3 for the demonstration of the motion-driven process. Each fish shape is wrapped around a template circle consisting of 91 landmark points with unique indexes assigned. The landmark points preserve the correspondence among three

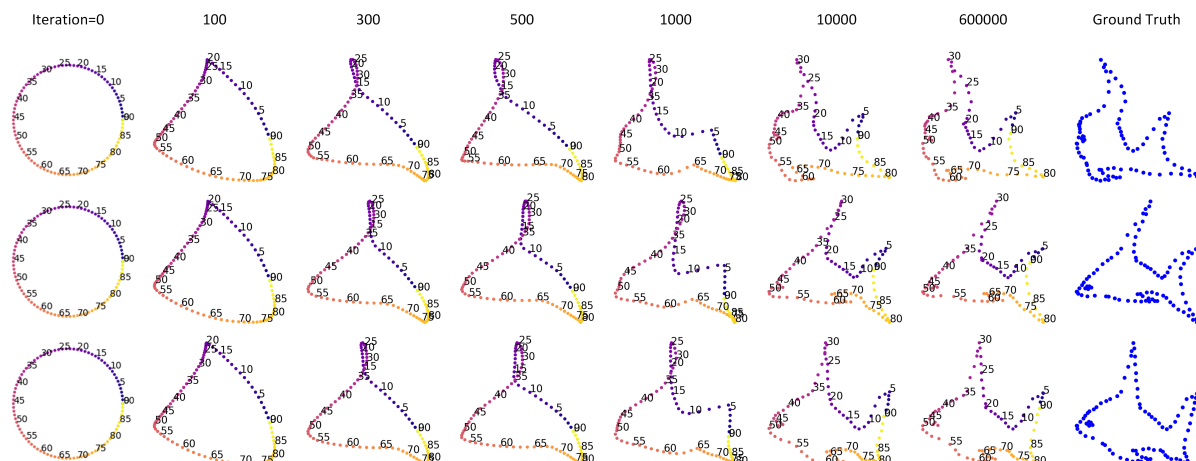


Figure 3.4: Illustration of our Motion-driven embedding process. Landmark points are numbered with color and the blue fishes in last column are input shapes.

template circles.

In the experiment, we illustrate how the template circle morph and conform around the target shape (e.g. fish) step by step at iteration 100, 300, 500, 1000, 10000 and 600000 as shown in Figure 3.4, which consequently lead to the gradual drift of the landmark points from the template circle to corresponding points on the target shape in order to further determine the point set correspondence. Under the guidance of decoding a global shape descriptor to the original input fish shape, the motion-driven embedding process starts with globally morphing the template circle. The second column of Figure 3.4 illustrate the globally morphed template shape after 100 iterations of shape deformation. We can observe that the globally morphed shapes share similar geometric configuration with all represented in a similar triangle shape. It is also interesting to notice that the landmark points from all three template circle coherently move towards the corresponding regions of these three triangles. As indicated in the figure, the vertices from three triangles uniquely corresponding to the landmark points with landmark indexes: [20, 50,

80].

The fifth row of Figure 3.4 illustrates the morphed template shape after 1000 iterations of deformation. As shown in the figure, the template shape starts to capture the local geometric structure context information and morph toward the associated fish shape. It comes to our attention that all landmark points progressively drift to the corresponding parts of further morphed shape. The motion-driven embedding process converges at 600000 iterations. The template circle shape morph and conform around the desired the fish shape as shown in Figure 3.4. All of the landmark points coherently drift the corresponding points of the morphed shape as indicated in the figure, which leads to a natural correspondence among the populations of fish shapes.

3.2.3 Generalization ability on test dataset

To further prove the generalization ability of our proposed model, we randomly generate another 100 shapes under a deformation level of 0.3 and use our PC-Net model to predict the point correspondence among all shapes. Following the above pair-wise evaluation strategy, we plot the training and test correspondence metric for various Euclidean distance thresholds. As shown in Figure 3.5, our model achieves nearly the same performance for unseen shapes on the test set as on the training set. This further demonstrates the powerful generalization ability of our proposed model. Note that our model is non-parametric and can predict the point correspondences among a group of point sets instantly without any iterative searching process.

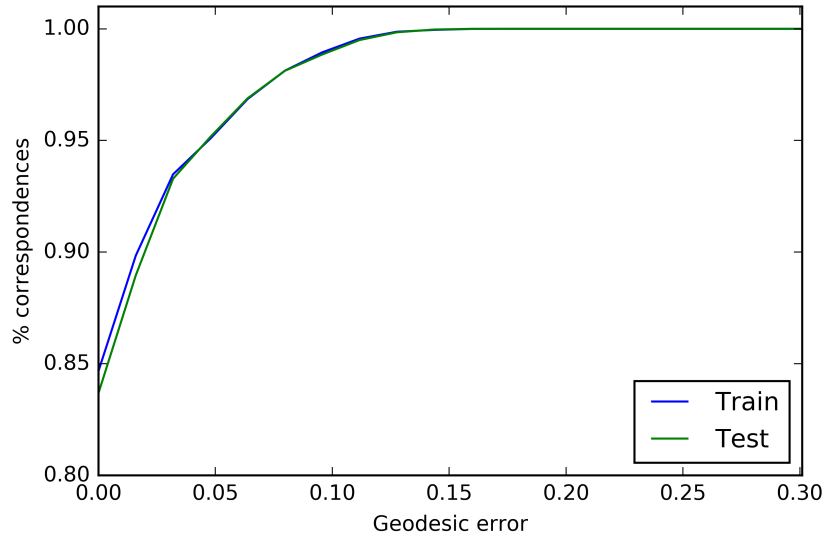


Figure 3.5: Correspondence performance on test set.

3.2.4 Linear shape interpolation

To demonstrate the effectiveness of our point-based feature encoder network, we checked the feature interpolation ability between two inputs. Figure 3.6 visualizes some examples of intra-class interpolations. One can see that our proposed model can get continuous shape reconstruction using interpolated feature vectors.

3.2.5 Robust to deformation, noise, and missing points

Firstly, we evaluate the robustness of our model under various deformation levels with (0.3, 0.4, 0.5, 0.6, 0.7, 0.8). Figure 3.7(a) shows the correspondence accuracy within a given Euclidean distance from the matching point to the reference point on the target shape. As shown in Figure 3.7(a), our model can achieve nearly all correct point correspondence with a Euclidean distance threshold of 0.3. Moreover, with the increase of deformation level, our model gets a reasonable per-

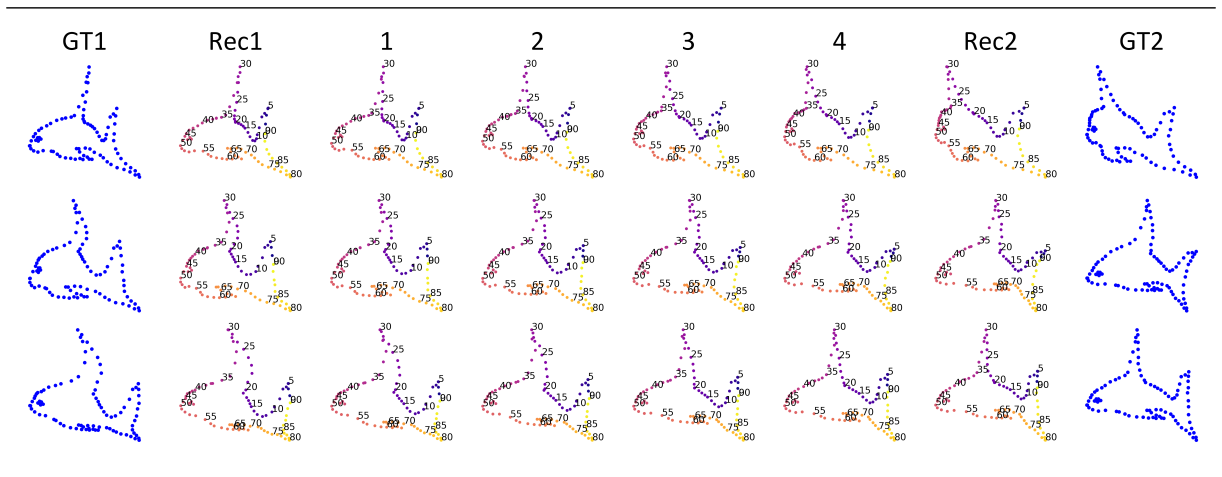


Figure 3.6: Examples of linear interpolation. ‘GT1’ and ‘GT2’ show two input shapes, ‘Rec1’ and ‘Rec1’ show their reconstructed shapes, and the middle columns show the interpolated shapes. Our model got continuous shape reconstruction using interpolated feature vectors. Note that the landmark points on each shape are corresponded to the landmark points on the other shapes.

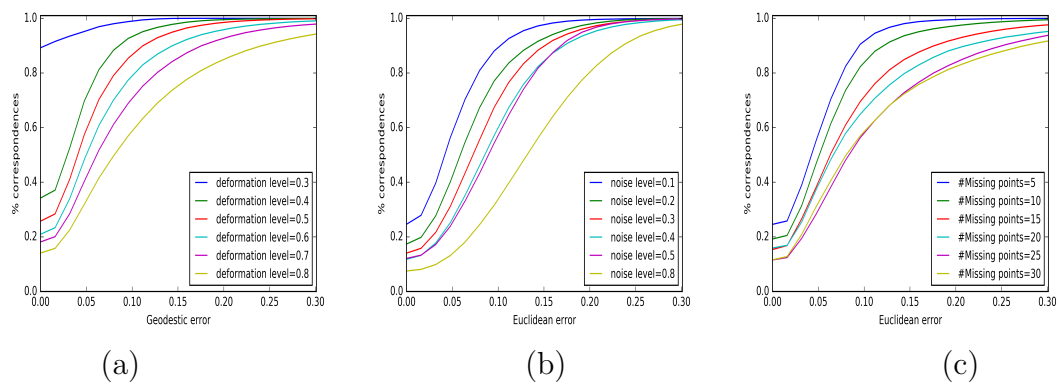


Figure 3.7: Robustness test. (a) Correspondence quality at different deformation levels. (b) Correspondences quality at different noise level. (c) Correspondence quality with different number of missing points.

formance degradation for shape correspondence. Same findings can be seen from Figure 3.7. In Figure 3.7, we show selected examples at different deformation levels. Even without explicit supervision, our PC-Net model can effectively produce accurate point correspondences at different deformation levels.

To further test the robustness of our proposed PC-Net model for point correspondence under the situation of noise, we add Gaussian noise to each point of the shapes, with different standard deviation values of [0.1, 0.2, 0.3, 0.4, 0.5, 0.8]. Quantitative correspondence results are shown in Figure 3.7(b). As was expected, our correspondence performance deteriorates with larger noise levels. When the noise level is smaller than 0.5, our PC-Net model can still maintain a satisfactory performance for shape correspondence. This is mainly due to the fact that the max-pooling operation in our point-based encoder can remain unchanged under small distortions. Selected visualization results are shown in Figure 3.8.

Furthermore, we test the performance of PC-Net with randomly missing points. Specifically, we first randomly select a point from a shape and then drop out its nearest K points. In our experiments, K varies from 5 to 30 with a step of 5. We train six separate models for each K . To evaluate the correspondence performance, we only take into account the points existing in paired shapes (we recorded the original point index during our data generation process). As shown in Figure 3.7(c), the correspondence performance deteriorates as K increases. This is because the corrupted input point sets have a larger difference in encoded feature which can disturb our motion-driven embedding process, and thus affect the point correspondences. Although our PC-Net model fails to achieve the accurate point correspondences under this situation, it can still match the overall shape given a Euclidean distance threshold of 0.3. Selected visualization results are shown in

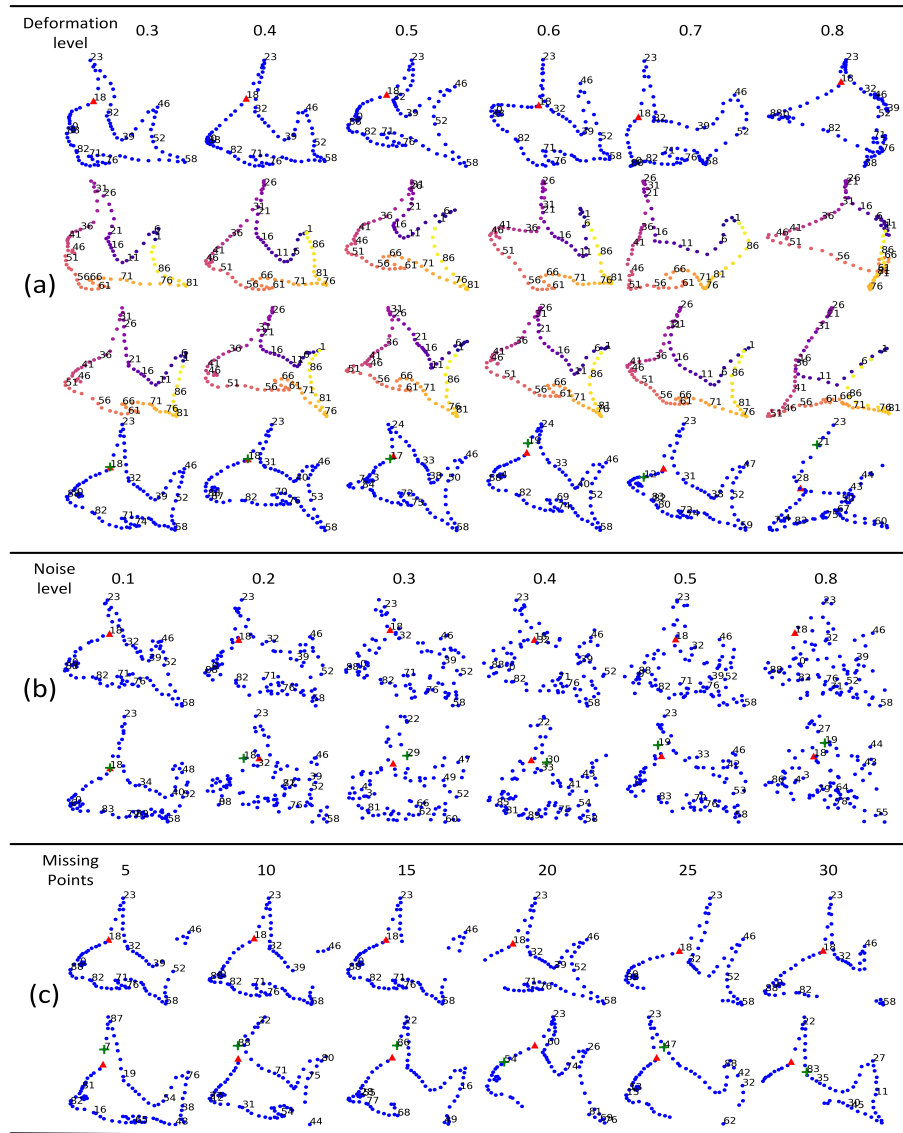


Figure 3.8: Examples of point correspondence at (a) different deformation level, (b) different noise level, and (c) different number of missing points. The top rows in (a)-(c) show the reference shapes, the middle rows in (a) show the reconstructed shapes, and the bottom rows in (a)-(c) show the predicted shapes with correspondences. We annotate some corner points in the reference shapes (top row) and find out their matching points in the target shapes (bottom row); whereas numbers on the shapes of the middle rows in (a) indicate the indexes of landmark points. The ‘red triangle’ indicates ground truth point with a point label ‘18’, while the ‘green cross’ indicates its corresponding point.

Figure 3.8. As shown in Figure 3.8, each pair of shape fits well overall, with a small error of Euclidean distance.

3.2.6 Rotation discussion

To evaluate the performance of our proposed model under arbitrary shape rotations, we trained a separate model using 5000 shapes with randomly rotation angles. Figure 3.9 gives some visualization examples. As shown in Figure 3.9, our correspondence quality is quite sensitive to rotation. This is because our model takes a fixed circle as input for point embedding. On one hand, this guarantees the shape correspondence during our motion-based embedding process; on the other hand, it limits the power of our model for rotated shapes correspondence. Therefore, to use our model for real-world point cloud data, it's necessary to align each shape to the same direction as a pre-processing. It is worth mention that despite our model fails to match the correspondence points under rotations (see second row in Figure 3.9), it still maintains a good correspondence for each corner (see the first row in Figure 10 for illustration). The color pattern in each reconstructed shape remains the same order as our input circle.

3.2.7 Correspondence for 3D point sets

For evaluate our approach on 3D point clouds dataset, we conduct experiments on the FAUST benchmark dataset [40]. FAUST dataset includes 100 human shapes in 10 different poses. For each shape, we sample 6890 points from the original point sets following [32]. For simplicity, we do not use mesh information in our experiments for these two data set.

In the above 2D shape correspondence task, a fixed circle is used to match the

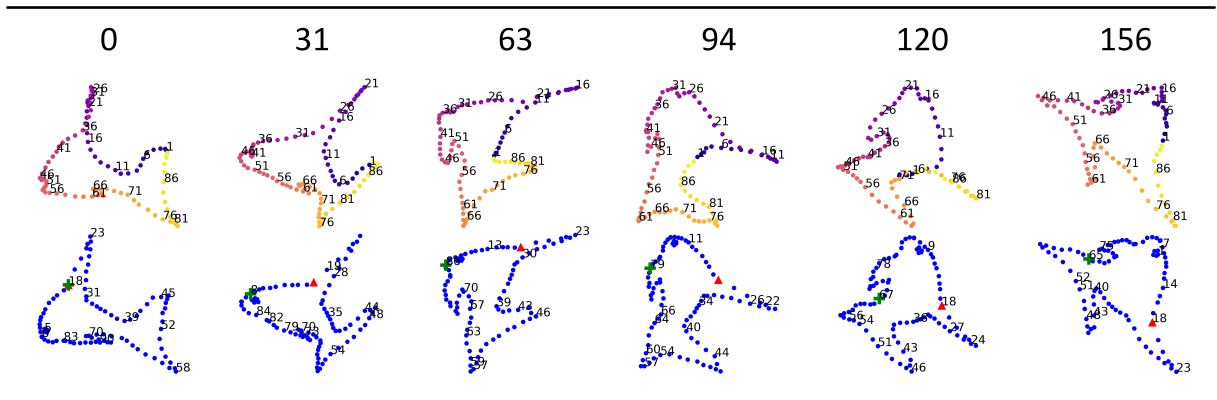


Figure 3.9: Selected examples of arbitrary rotations. Numbers in the top indicate the rotation angles. The first row show the reconstructed shapes, the second row shows the predicted shape correspondence. Even though our model fails to match the correspondence points (see second rows for illustration), it maintains a good correspondence for each corner (see the first row for illustration). The color pattern in each reconstructed shape remains the same order as our input circle.

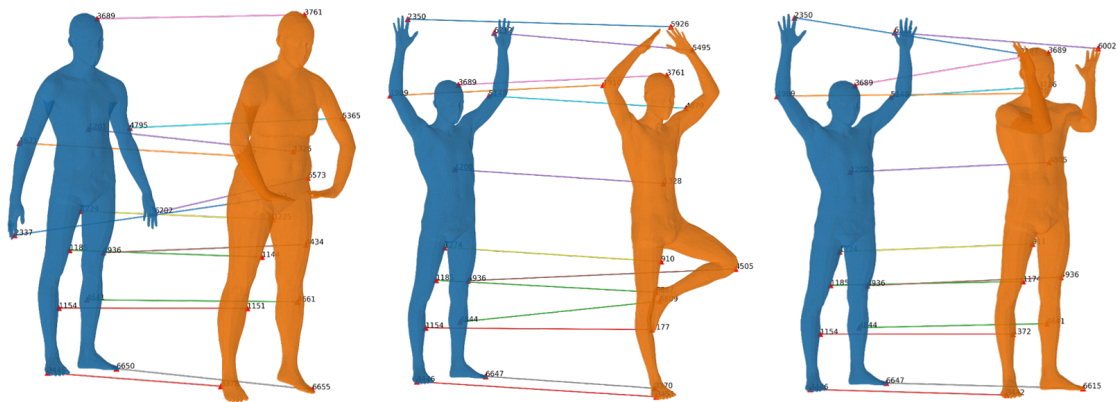


Figure 3.10: Selected examples of 3D shape correspondence on FAUST dataset. The first shape is used as a reference. First row shows successful correspondence examples, second row shows failure examples. Note that our results are generated in an unsupervised way.

input 2D points. In this experiment, instead, a spherical surface template is used to match the 3D point cloud for both FAUST dataset. The diameter is chosen based on the average shape size. Selected examples of 3D shape correspondences are illustrated in Figure 3.10. As shown in Figure 3.10, our model can achieve satisfactory correspondences for 3D point sets as well even without explicit supervision.

3.3 Discussion

In this chapter, we propose a novel non-parametric learning-based point correspondence framework in an unsupervised manner. We propose a “shape morphing initiator” with a “motion-driven embedding” network which progressively and coherently drifts all landmark points from a template shape to corresponding positions on the target shape without using ground truth labels. Experimental results demonstrate the robustness of our model under various levels of deformation, missing points, and noise. More importantly, with demonstrated generalization ability, our proposed PC-Net can directly predict the correspondences of two or more unseen point sets.

Chapter 4

Application In 2D Image

Matching

(This chapter is submitted as a paper “MF-GeoNet: Model-Free Geometric Transformation Network for Image Matching” with Jianchun Chen et al. as co-authors under review.)

In addition to the CPD-Net in chapter 2, we further introduce an application of learning based model-free network for image matching. Recent efforts introduce convolutional neural network to learn a geometric model (i.e. an affine or thin-plate spline transformation) for image matching and determine correspondences between two images. The incapability of a geometric model in estimating a high complexity parametric transform limits their use in applications to coarse image alignment/matching. This chapter presents a novel approach to learn a model-free geometric transformation to estimate a continuous smooth displacement field and identify two images with a significant geometric deformation. In contrast to model-based method, our proposed method, named Model-Free Geometric Transformation Networks (MF-GeoNet), can learn displacement vector function to estimate geometric transformation from a training dataset. MF-GeoNet is trained to have robust generalization ability to directly predict the desired geometric transformation to identify the correspondence between unseen new pair of images. Furthermore, MF-GeoNet is theoretically proved to learn a continuous displacement vector function to avoid imposing a parametric smoothness constraint by regularizing the displacement field. The experiments demonstrate that MF-GeoNet can generalize from training data to predict continuous smooth displacement field to reliably identify the correspondence between two images. We conducted experiments over both synthetic and real image dataset. The results demonstrate the superior performance of MF-GeoNet over other state-of-the-art techniques in identifying the correspondences, even when images are in the presence of significant geometric deformation.

Image level point correspondence matching has been a key challenge for computer vision society, since this technology is widely applied in various tasks such

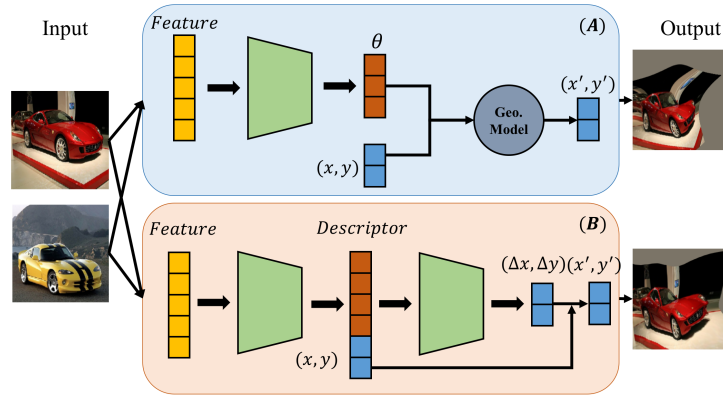


Figure 4.1: Comparison between geometric transformation model (A) and our proposed model-free geometric transformation network (B).

as tracking [41], medical image registration [34] and camera pose estimation [42]. Main stream methods tackle this problem through fitting a geometric transformation function between corresponding point set from image pairs.

To this end, traditional image correspondence estimation approaches normally carry out two-step implementations as follows. It starts with the computation of a pre-defined hand-crafted image feature descriptor such as SIFT and HOG [43, 44] to obtain a pixel-level description and followed by a process to iteratively optimize a specific geometric transformation model through the feature matching algorithm such as RANSAC [45] and Hough transform [46], aiming to minimize a matching loss function. In general, these methods perform well in identifying the reliable image correspondence but face challenges posed by 1) the dramatic image appearance variation (i.e. texture, color, lighting changes and so on) between two images, 2) the significant geometric structural variation between two images as those image conditions greatly impact the quality of the hand-crafted image descriptor.

With the success of deep neural network, researchers started to investigate the

learning-based approaches [1, 47, 48, 49] to address the above mentioned challenges with the hope to generalize from training to predict real-time image matching that is robust to various deteriorated image conditions. As shown in Figure 4.1 (A), those efforts are concentrated on the development of a paradigm to learn a geometric model (i.e. affine or thin-plate) for image matching, which converts the problem into the regression of parameters of the geometric model supervised by minimizing the image matching loss. The learning based approaches greatly improved the image matching performance, however it is suggested in [50] that their methods lack capability in estimation of high complexity geometry due to the incapability of the widely used geometric models (i.e. affine and TPS). In addition, the mismatch between the transformation described by geometric models adopted and the actual transformation required for image matching might potentially lead to inappropriate estimation of desired geometric transformation.

This triggers our motivation to develop a model-free geometric transformation networks (MF-GeoNet) with the hope to address the above mentioned issues faced by learning a model-based geometric transformation for image matching. As shown in Fig. 4.1 (B), this chapter presents a novel approach to learn a model-free geometric transformation which estimates a continuous smooth displacement field for image geometric matching. In contrast to model-based method, MF-GeoNet fully leverages the deep neural network for fitting arbitrary displacement vector function. Compared with previous efforts, our proposed structure has 1) large degree of freedom for complex transformation modeling; 2) dense displacement field prediction without inaccurate interpolation operation; 3) the spatial continuity of displacement field preserved by network structure without using any penalization term.

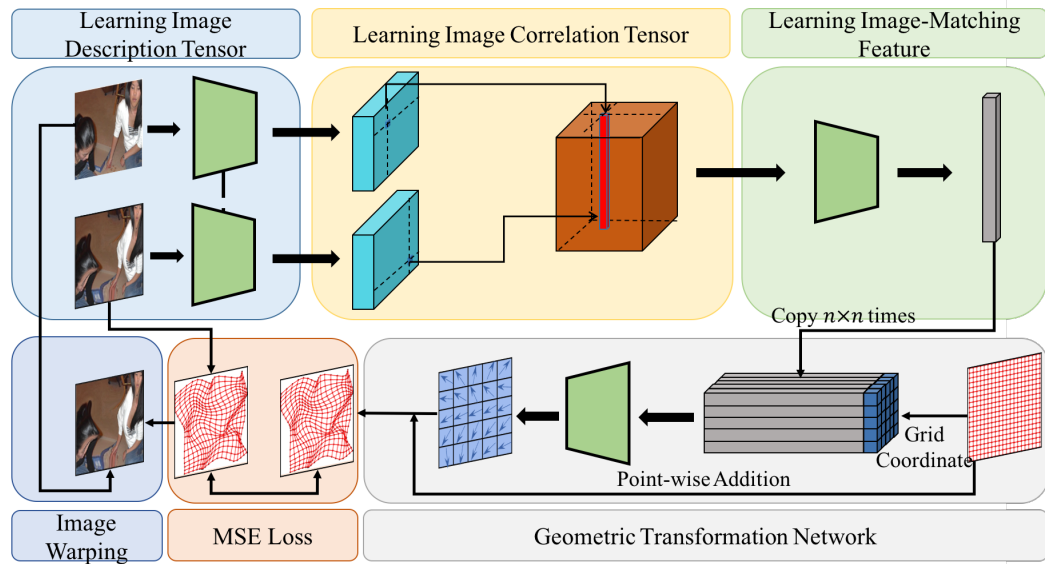


Figure 4.2: Main pipeline: Our proposed architecture comprises of two parts - Image-Matching Feature Learning and Geometric Transformation Network. For a given pair of input images, we first extract the Image Description Tensor f_A and f_B through convolutional neural network. We next generate Image Correlation Tensor C_{AB} between the two Image Description Tensors. Furthermore, we embed this Image Correlation Tensor into a latent feature d_{AB} which represents Image-Matching Feature. Finally, we pass this feature through Geometric Transformation Network comprising of MLPs which predicts desired displacement field and consequently transform points in the source plane to target plane. We minimize the Mean Square Error Loss between the corresponding points between predicted point set and the ground truth point set.

In this chapter, we present a novel model-free geometric transformation network for image correspondence matching. As illustrated in Fig. 4.2, the first component of our network is named “Learning Image Description Tensor”, where a fixed-weight convolution neural network is employed to extract two feature maps out of the input image pair. The following “Learning Image Correlation Tensor” component learns the dense correlation between two feature maps through *correlation layer* and *normalization layer*. The third component is “Learning Image-Matching Feature”. In this component we embed the Image Correlation Tensor into a latent vector, which describes the geometric transformation between the image pair. “Geometric Transformation Network” is the last component, which is visualized in Fig. 4.1. We exploit a stack of deep neural network which decodes the Image-Matching Feature to a desired geometric transformation function. With this Geometric Transformation Network, we estimate image correspondence by transferring key point from source image to the target image. We use a Mean Square Error as loss function to supervise the Geometric Transformation Network learning, as well as the Image-Matching Feature learning. The main contributions of method are briefly summarized. We proposed a MF-GeoNet which solves the incapability of geometric models in estimating high complexity parametric transformation. We leverage the power of neural network in fitting arbitrary transformation function to accommodate any different complexity level of geometric transformation according to actual needs. Our proposed Geometric Transformation Network is theoretically guaranteed to produce a spatially continuous displacement field. With this property, we avoid imposing additional penalization term on displacement field as smoothness constraint. Our MF-GeoNet is a model-free geometric transformation method which does not require model selection procedure.

Consequently, we avoid the critical mismatching problem of selected transformation model and actual desired geometric transformation between image pair. Our experiment demonstrates the effectiveness of our proposed MF-GeoNet in image correspondence matching. Our model achieved superior performance over other state-of-the-art approach for image matching, especially in high complexity transformation scenario.

4.1 Methods

4.1.1 Image-Matching Feature Learning

In this section, we present a deep neural network architecture that takes two images as inputs and outputs an Image-Matching descriptor that determines desired geometric transformation to align the source point set of one image with the target point set of the other. As illustrated in Fig. 4.2, the pipeline of this section consists of Image Description Tensor Learning (blue block), Image Correlation Tensor Learning (yellow block) and Image-Matching Feature Learning (green block). In following paragraphs, we describe each of the above stages in detail.

Learning Image Description Tensor: MF-GeoNet starts with extracting image features by convolutional neural network. The network is based on ResNet-101 [51] architecture with layers active till ‘global average pooling’ layer followed by L2-normalization. The neural network produces a feature map of the input image with dimensions $f \in R^{h \times w \times c}$ (which is defined as *Image Description Tensor* in this work), where $h \times w$ denotes the size of the feature map and c denotes the feature dimension. We visualize the Image Description Tensor using cyan blocks in Fig. 4.2. Two weight-shared CNN are used to generate two Image Description

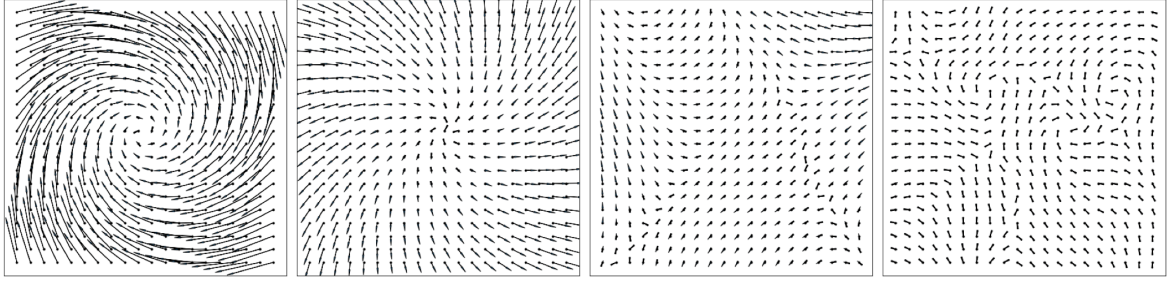


Figure 4.3: Visualization of hidden features after each MLP layer. The feature of each point is reduced to 2 dimension by Principal Component Analysis (PCA). The last figure is the offset prediction of a set of grid point, which is also called deformation field. All MLPs and input feature are randomly initialized.

Tensors from input image pair.

Learning Image Correlation Tensor: The second stage of the network deals in learning dense correlation between the two Image Description Tensor, where the result is defined as Image Correlation Tensor. We visualize the Image Correlation Tensor using an orange block in Fig. 4.2. The alignment layer generates similarity scores by mapping their spatial location while discarding the original Image Description Tensor. Our alignment layer comprises of two components. First is the *correlation layer* which is responsible for computing all pairs of similarities between the two Image Description Tensors f_A and f_B . Second, these similarity scores are then processed through *normalization layer* to eliminate unwanted ambiguous matches from the network. For a given image pair, the Image Description Tensor generated $f_A, f_B \in R^{h \times w \times d}$, the corresponding Image Correlation Tensor $C_{AB} \in R^{h \times w \times (h \times w)}$ then comes as the result of the correlation layer. Detail description of the function is shown in the Eq. 4.1.

$$C_{AB}(m, n, k) = f_B(m, n)^T f_A(m_k, n_k) \quad (4.1)$$

where (m, n) and (m_k, n_k) represent each individual position in the feature map of the two images. The variable k is calculated with the help of the Eq. 4.2.

$$k = h(n_k - 1) + m_k \quad (4.2)$$

Once we obtain the Image Correlation Tensor, it becomes equally important to post-process those similarity scores to remove the redundant values. The normalization operation is applied to each spatial location of the Image Correlation Tensor. This process refines the Correlation Tensor by reducing the ‘outliers’ matching value. Specifically, we first pass the raw similarity scores through ReLU function which cancels out all the negative correlation score. Following this, the intermediate results are processed with L2-normalization. The L2-normalization successfully amplifies scores of confident feature matching while reduces scores of ambiguity feature matching via a quadratic function. These two layers take local feature correlation into a part of end-to-end network, which further enhances the robustness by capturing patterns of ‘inliers’ and ‘outliers’ feature in network training.

Learning Image-Matching Feature: In this stage, we designed an architecture by sequentially stacking two modules that consists of convolutional neural network, batch normalization, ReLU and a final fully connected (FC) layers. The motivation here is to fully leverage the correlation score of neighbor points for a ‘patch’ to ‘patch’ alignment by CNN layer. Following FC layer embeds the features of correlation tensor into a latent vector called Image-Matching Feature, which encodes intrinsic geometric transformation information. The Image-Matching Feature is of dimension m .

4.1.2 Geometric Transformation Network

In this section, we introduce our Geometric Transformation Network, as illustrated in Fig. 4.2 (grey block). The Geometric Transformation Network learns a continuous transformation function $\mathbf{P} \rightarrow \mathbf{Q}$ from source plane \mathbf{P} to target plane \mathbf{Q} . We first repeat Image-Matching Feature for n times, where n is the number of point in source plane \mathbf{P} . As shown in the figure, for each point $p \in \mathbf{P}$, we concatenate its coordinate to a Image-Matching Feature d_{AB} to form a new point feature with dimension $m + 2$. A series of Multi-Layer Perceptrons (MLPs) with Batch Normalization (BN) layers and Rectified Linear Unit (ReLU) layers are carried out to learn the point-wise displacement. The overall transformation is defined as:

$$\mathcal{T}(p) = \mathcal{F}_{\theta_1, \dots, \theta_n}([d_{AB}, p]) + p \quad (4.3)$$

where $[.]$ means the concatenation operation and θ_i denotes the network parameters of layer i .

The spatial continuity of transformation function \mathcal{T} is necessary for a smooth image warping. Since our learned function \mathcal{T} is intrinsically continuous, we avoid imposing a parametric smoothness constraint by regularizing the displacement field. To prove this property, we first expand the first MLP layer as follows:

$$\mathcal{F}_{\theta_1, \dots, \theta_n}([d_{AB}, p]) = \mathcal{F}_{\theta_2, \dots, \theta_n}(W_1 d_{AB} + W_2 p + b) \quad (4.4)$$

Since all MLP, BN and ReLU layers are continuous function and d_{AB} is invariant for points in same plane space, $\mathcal{F}_\theta(\cdot)$ is continuous for $\forall p \in \mathbf{P}$. The concatenated

correlation descriptor d_{AB} only controls the transformation function by adding a bias term in the first layer of MLP, which does not break the continuity of the function. According to the definition of continuous function, we have our transformation function \mathcal{T} to be continuous, as shown in Eq. 4.5.

$$\lim_{p_i - p_j \rightarrow 0} \mathcal{T}(p_i) - \mathcal{T}(p_j) = \mathcal{F}(p_i) - \mathcal{F}(p_j) + (p_i - p_j) = 0 \quad (4.5)$$

In Fig. 4.3, we further demonstrate that the hidden features after any MLP layer preserves spatial continuity. The hidden features associated with the lower layer represent a linear transformation, which is similar to affine transformation. With the increase of MLP and ReLU layers, the network has a capacity in fitting complex non-linear transformations.

In general, our method leverages the power of deep neural network to fit an arbitrary function. Compared with lower complexity parametric models, our method is sufficient to accommodate an arbitrary high complexity geometric transformation. Moreover, our Geometric Transformation Network is end-to-end trainable and can be jointly trained with other modules of our MF-GeoNet such as Image-Matching Feature Learning network.

4.1.3 Loss Function

It is critical to train a deep neural network with proper loss function. For image matching, due to the presence of highly complex arbitrary geometric deformation in the image, the unsupervised distance measurement (i.e. Chamfer Distance) might not capture the essential geometric difference between two images. Therefore, in our current implementation, we train our MF-GeoNet with a supervised loss

function defined as below. Given two set of image keypoints \mathbf{P} and \mathbf{Q} , the loss function is defined based on the average of all pairwise distances between two corresponding keypoint (refer to Eq. 4.6)

$$L(\mathcal{T}(P), Q) = \frac{1}{N} \sum_{i=1}^N \|\mathcal{T}(p_i) - q_i\|_2^2 \quad (4.6)$$

The loss function is back-propagated for optimizing parameters in both Feature Extraction Network and Geometric Transformation Network.

4.1.4 Implementation Details

The Image-Matching Feature Learning procedure follows the similar setting in [1]. Taking two image I_A, I_B as input, a ResNet-101 [51] based architecture was carried out to learn Image Description Tensor f_A, f_B . We used the pretrained network weights from ImageNet dataset [52] and freeze these parameters from back-propagation. After we got Image Correlation Tensor C_{AB} , we conducted 2 CNN layers with channels (128,64) and kernel size (7,5) respectively. The last fully-connected layer regresses Image-Matching Feature $d_{AB} \in \mathbb{R}^{1024}$. For each single point in 2D image, we normalized it into $[-1, 1]$ and concatenate a repeated Image-Matching Feature d_{AB} on its coordinates. In synthetic dataset, we select a 20×20 grid points as keypoints. In Geometric Transformation Network, the hidden features after 4 MLPs are of size (1024,256,64,2), where the 2 dimension output is regarded as the displacement vector of each input point. We used Adam optimizer for training with learning rate 3×10^{-4} . Our model was implemented using PyTorch framework and ran on a single Nvidia GTX 1080Ti GPU with 11

GB on- board memory.

4.2 Experimental Results

We carried out two sets of experiments to assess the performance of our method on image correspondence matching problem. We conducted both qualitative and quantitative evaluation to better assess MF-GeoNet’s performance for image matching. In section 4.2.1, we describe the general experimental setting. In section 4.2.2, we verify the effectiveness of our proposed MF-GeoNet in estimating image geometric transformation with different complexity levels. In section 4.2.3 we produced quantitative comparisons on real image correspondence matching with baseline methods. Section 4.2.4 depicts the visualization result of real image semantic alignment.

4.2.1 Experiment Setup

Datasets. Pascal VOC 2011 dataset contains 28,952 images in total and each image has its object level annotation. In the experiments, we use Pascal VOC 2011 [53] dataset to prepare our training and testing dataset. For each image in Pascal VOC, we apply geometric transformation [1] (both affine and TPS) to obtain a synthesized transformed image. The paired raw and transformed images are used for training and testing during the experiments. For real image correspondence matching, we evaluate our model based on Proposal Flow dataset by Ham et. al [2]. This dataset comprises of 900 image pairs, with each image portraying multiple instances of same class or different class objects in them. Moreover, the images contain significant amount of background clutter to estimate model’s generalization

ability.

Baseline. We have compared MF-GeoNet’s performance with the popular methods such as SIFT Flow[43], Graph Matching Kernels [54], Deformable Spatial pyramid matching [55], DeepFlow [56] and different versions of proposal Flow[2]. These models serve as the benchmark to assess our model performance. We also compare our model with CNNGeo [1], which achieved best performance in supervised learning. CNNGeo has a similar backbone structure as our proposed method, containing feature extraction network and feature alignment network. However, CNNGeo regresses the parameter of specific geometric transformation model. They use two geometric models in their pipeline: affine transformation and TPS transformation. The CNNGeo use affine or TPS to model the geometric transformation. In addition, to enhance their performance, they implement a two-stage estimation of the geometric transformation. In the first stage, they estimate an affine transformation between source image and target image while using same parameter to warp the source image. In the second stage, they estimate the TPS transformation between the warped source image and target image to identify the image correspondence.

Evaluation Metric. For experiment in Proposal Flow dataset [2], we follow a standard evaluation metric used for the model benchmark, the average probability of correct keypoint (PCK)[57]. A keypoint is counted as a correct match if the predicted location is within an allowed distance of $\alpha \times \max(h, w)$ to target keypoint position, where we consider $\alpha = 0.1$. h and w are height and width of object respectively in our experiment. For experiment in synthesis dataset, we directly use Mean Square Error between corresponding points as evaluation metric, as illustrated in Eq. 4.6.

4.2.2 Comparisons on synthetic image matching

In this section, we demonstrate the effectiveness of our MF-GeoNet in modeling arbitrary and complex image geometric transformation. We compared our MF-GeoNet with CNNGeo[1] for image matching. In this experiment, the CNNGeo used TPS as its geometric model with 3×3 controlling point. The experiment is conducted on TPS synthesis train/test set from Pascal VOC 2011 dataset [53]. To simulate geometric transformation with different complexity, we adjust the number of controlling point for data generation. We design two tests under this section.

- In the first test, we are interested in understanding how well MF-GeoNet perform against different levels of transformation complexity. Therefore, we increase the complexity of the geometric transformation by gradually adding more TPS controlling points into our test settings. We compare the performance of MF-GeoNet with CNNGeo model against the same environmental settings.
- In the second test, we wanted to test our model for the real-world image matching scenarios. As we do not have a prior knowledge on the complexity and the type of transformation required for image matching, thus the mismatch between the transformation described by geometric models and the actual transformation required for image matching might potentially lead to inappropriate estimation of desired geometric transformation. Therefore, to test this, we simulate a mismatching scenario and compare the performance between our MF-GeoNet and CNNGeo models. We prepare the synthesized data by applying at TPS with 6×6 controlling points on the Pascal VOC dataset.

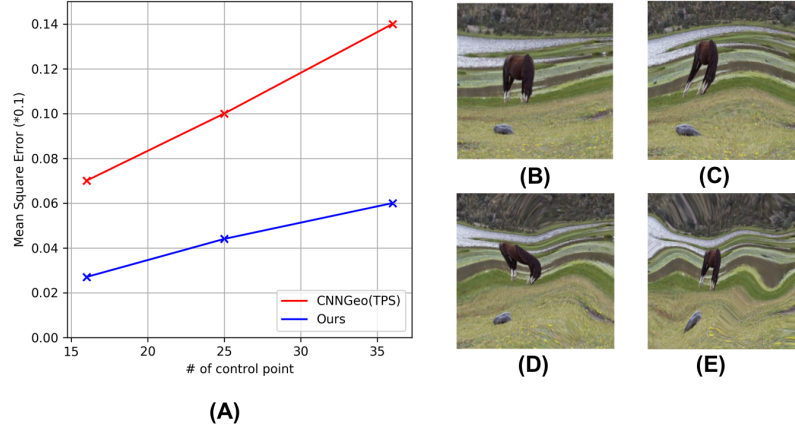


Figure 4.4: Quantitative comparison between our model and CNNGeo [1] in synthesis testing datasets with different number of controlling point (A) and visualization of images transformed by TPS with different controlling point. Subplot (B), (C), (D) and (E) are from TPS transformation with 3×3 , 4×4 , 5×5 , 6×6 controlling points respectively.

Result of Test 1: During this test, we increase the controlling points from 4×4 to 6×6 . The Fig. 4.4 demonstrates the comparative result for both methods. As we can see from the figure, MF-GeoNet (shown in blue) has a consistent lower MSE score than that of CNNGeo (shown in red) while we constantly increase the complexity of the transformation. In addition, from the illustration we can infer that as we gradually increase the controlling points, the corresponding MSE loss increases steeply for CNNGeo model as compared to our model. For instance, at a given controlling point (6×6), CNNGeo model has a MSE value of 0.014 whereas for the same controlling point, our model recorded a MSE value of 0.006. Moreover, for a given pair of controlling points, the slope of CNNGeo curve (red) is 200% higher than the slope of MF-GeoNet curve (blue).

Result of Test 2: We visualize the comparative results for MF-GeoNet and CNNGeo as shown in Fig. 4.5. In the Fig. 4.5, the first row shows the original

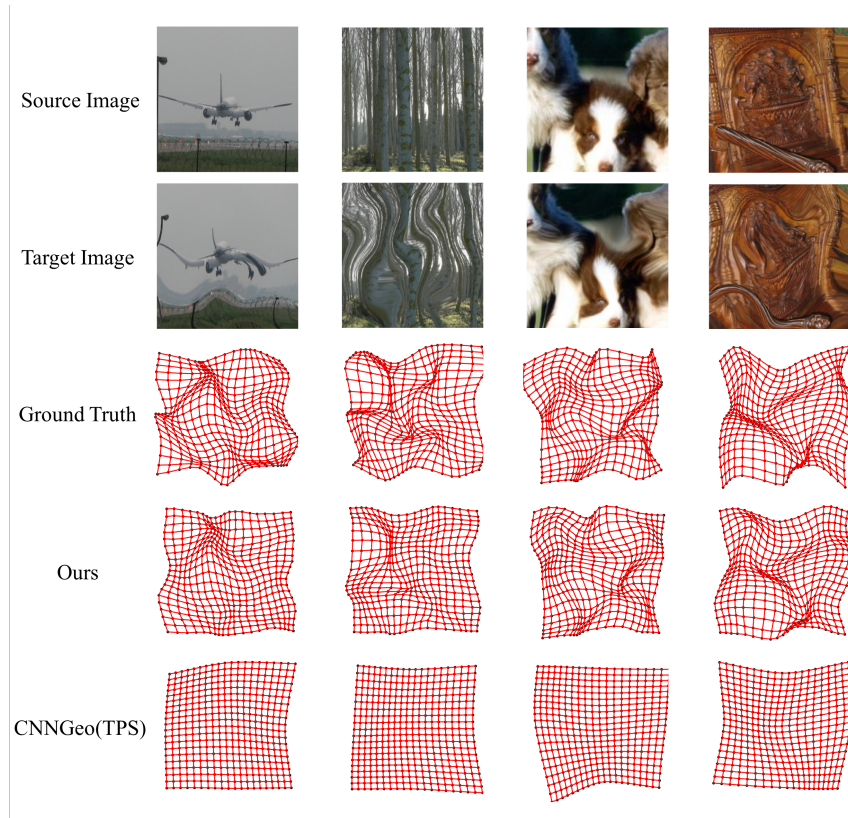


Figure 4.5: Qualitative comparison between our model and CNNGeo [1] in dataset synthesized by 6×6 controlling points TPS with deformation level equals to 0.2. Red mesh denotes the displacement field.

source image and the second row shows the transformed synthesized images. The third row demonstrates the ground truth displacement field, and the fourth and fifth row show the estimated displacement fields by MF-GeoNet and CNNGeo respectively. From the comparison, we can see that the predefined 3×3 control point TPS model significantly underfits the 6×6 control point TPS transformation, while MF-GeoNet is able to estimate complex local transformation.

Methods	PCK(%)
DeepFlow [56]	20
GMK [54]	27
SIFT Flow [43]	38
DSP [55]	29
Proposal Flow [2]	56
CNN feature + RANSAC	47
CNNGeo (affine) [1]	55.9
CNNGeo (TPS) [1]	58.2
CNNGeo (affine+TPS) [1]	67.6
Ours-A	59.5
Ours-B	60.0
Ours-C	66.5
Ours-(A+B)	67.8

Table 4.1: Comparison of PCK with our baseline methods in full Proposal Flow dataset [2]. Learning-based methods are trained on synthetic dataset. The settings of our four models are described in Section 4.3.1.

Methods	PCK(%)
Affine[1]	73.04±0.33
TPS [1]	78.07±1.43
Affine+TPS [1]	73.20±0.27
Ours	85.17±0.62

Table 4.2: Comparison of PCK with our baseline methods in Proposal Flow dataset [2] under K-fold setting. Details are described in Section 4.3.2

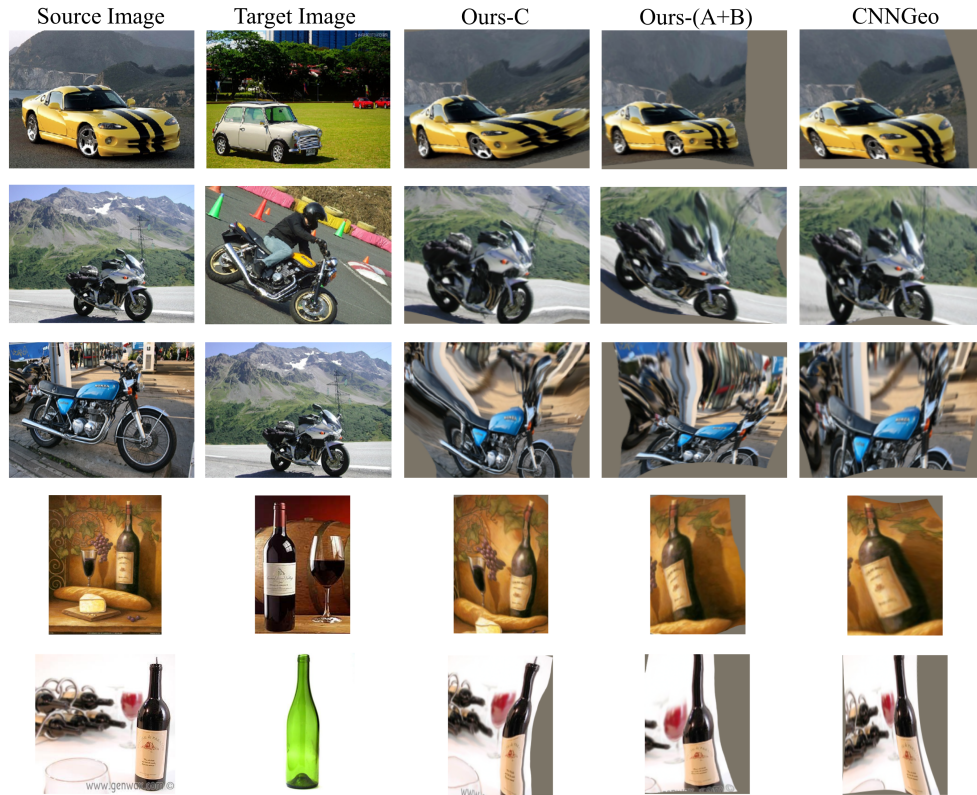


Figure 4.6: Qualitative results and comparisons on Proposal Flow dataset [2]. The selected pictures from 3 categories (motorbike, car and winebottle) are of complex transformation.

4.2.3 Comparisons on real image point correspondence matching

In this section, we conduct real image point correspondence matching experiment on Proposal Flow dataset [2] under two different experimental settings. In Section 4.2.3.1, we use complete Proposal Flow dataset to evaluate our MF-GeoNet trained on synthetic dataset. In Section 4.2.3.2 we split Proposal Flow dataset into training and testing set and perform K-fold evaluation of our MF-GeoNet.

4.2.3.1 Experiment One.

For this task, we prepared 3 MF-GeoNet models named Ours-A, Ours-B and Ours-C. These three models share same architecture as we described in Section 3. Each model is fed with different type of synthetic data. **Ours-A** is trained on image synthesized by affine transformation. **Ours-B** is trained on image synthesized by TPS transformation. Data used for **Ours-C** is synthesized by a combination of affine and TPS transformation. In addition, similar to CNNGeo [1], we combines Ours-A and Ours-B to construct a 2-stage model **Ours-(A+B)** in a coarse-to-fine paradigm.

From Table 4.1 we can see that our proposed method outperform all previous methods as listed in the table. We mainly compared our MF-GeoNet with CNNGeo in this test as follows. For one stage training, we compared CNNGeo (Affine) with Ours-A and CNNGeo (TPS) with Ours-B. Our models have about 4% and 2% PCK improvement respectively. For two-stage training, we compare CNNGeo (affine+TPS) with Ours-(A+B). The Ours-(A+B) gains 0.2% improvements over CNNGeo (affine+TPS). In addition, it's interesting to see that, in one stage, Ours-C can estimate the geometric transformation which is generated by a combination of affine and TPS transformation. It can even achieve superior result by 8% PCK improvement against CNNGeo (TPS). In contrast, CNNGeo needs to estimate transformation in two stages.

The possible reason for the marginal improvement of Ours-(A+B) over CNNGeo (affine+TPS) in the testing case is that the synthesized training data is simulated by affine and TPS transformation. Therefore, our MF-GeoNet might not generalize from those training datasets to predict geometric transformation other than affine and TPS. Consequently, it is reasonable that Ours-(A+B) and

CNNGeo (affine+TPS) achieved comparative result on real dataset. For this reason, we conducted additional experiment in Section 4.3.3.2 to prepare new training dataset from the real image.

4.2.3.2 Experiment Two.

In this section, we randomly split Proposal Flow Dataset into 3 folds. For each instance we select the first two folds as the training set and the other one as the test set. For CNNGeo, we prepared 3 models named CNNGeo (affine), CNNGeo (TPS), CNNGeo (affine+TPS). All network parameters of MF-GeoNet and CNNGeo in this section are trained from scratch except for the layers from ResNet-101.

As shown in Table 4.2, our MF-GeoNet network outperforms CNNGeo by a great margin. Compared with CNNGeo (affine), CNNGeo (TPS), CNNGeo (affine+TPS), our model achieves 7%, 12% and 12% improvement respectively. This experiment demonstrates that our proposed method is more desirable for real image transformation.

4.2.4 Qualitative results on real image corresponding estimation

Fig. 4.6 shows vivid illustrations of our model performances (under Section 4.3.3.1 setting) through real image based correspondence estimation. The images contained large intra-class variations with a lot of background clutter, however, our model prominently estimated large transformation as well as non-rigid transformations. Each row here represents a new test result with columns containing Source Image, Target Image, improved performance of our single **Model C** along

with the performance of our **Model (A+B)**. The last column is appended to this which represents the performance of CNNGeo on the test image. It prominently expresses the improved performance of our MF-GeoNet over CNNGeo. Unlike model-based parametric transformation, our model-free geometric transformation confirms the generalization ability towards a huge set of variations and takes into account the capacity to incorporate large transformation estimations. Therefore, our proposed MF-GeoNet proves to preserve spatial continuity during image transformation while achieving superior results against parametric methods.

4.3 Discussion

We have presented a novel model-free geometric transformation network for image correspondence matching, which has the capacity to fit complex geometric transformation. Moreover, our model has proved to be continuous and to accommodate arbitrary transformation, while avoiding mismatching problem earlier caused by model-base methods. The experiments demonstrate the effectiveness of our approach in image correspondence matching, especially for robustness in complex image transformation.

Bibliography

- [1] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017.
- [2] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3475–3484, 2016.
- [3] Andriy Myronenko, Xubo Song, and Miguel A Carreira-Perpinán. Non-rigid point set registration: Coherent point drift. In *Advances in Neural Information Processing Systems*, pages 1009–1016, 2007.
- [4] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2011.
- [5] Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE transactions on pattern analysis and machine intelligence*, 29(3), 2007.
- [6] Xiang Bai and Longin Jan Latecki. Path similarity skeleton graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1282–1292, 2008.

- [7] Jiayi Ma, Ji Zhao, and Alan L Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on image Processing*, 25(1):53–64, 2016.
- [8] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Trans. image processing*, 23(4):1706–1721, 2014.
- [9] Yi Wu, Bin Shen, and Haibin Ling. Online robust image alignment via iterative convex optimization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1814. IEEE, 2012.
- [10] Haibin Ling and David W Jacobs. Deformation invariant image matching. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1466–1473. IEEE, 2005.
- [11] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 15–18. IEEE, 2006.
- [12] JB Antoine Maintz and Max A Viergever. A survey of medical image registration. *Medical image analysis*, 2(1):1–36, 1998.
- [13] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
- [14] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consen-

- sus. In *European Conference on Computer Vision*, pages 500–513. Springer, 2008.
- [15] Alan L Yuille and Norberto M Grzywacz. A computational theory for the perception of coherent visual motion. *Nature*, 333(6168):71, 1988.
- [16] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [17] Lingjing Wang, Cheng Qian, Jifei Wang, and Yi Fang. Unsupervised learning of 3d model reconstruction from hand-drawn sketches. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1820–1828. ACM, 2018.
- [18] Meng Wang, Lingjing Wang, and Yi Fang. 3densinet: A robust neural network architecture towards 3d volumetric object prediction from 2d image. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 961–969. ACM, 2017.
- [19] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2013.
- [20] Lingjing Wang, Jianchun Chen, Xiang Li, and Yi Fang. Non-rigid point set registration networks. *arXiv preprint arXiv:1904.01428*, 2019.
- [21] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

- [22] Haili Chui and Anand Rangarajan. A new algorithm for non-rigid point matching. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 44–51. IEEE, 2000.
- [23] Alan L Yuille and Norberto M Grzywacz. A mathematical analysis of the motion coherence theory. *International Journal of Computer Vision*, 3(2):155–175, 1989.
- [24] Jiayi Ma, Weichao Qiu, Ji Zhao, Yong Ma, Alan L Yuille, and Zhuowen Tu. Robust l2e estimation of transformation for non-rigid registration. *IEEE Trans. Signal Processing*, 63(5):1115–1129, 2015.
- [25] Tao Wang and Haibin Ling. Path following with adaptive path estimation for graph matching. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [26] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4032–4040, 2015.
- [27] Hang Su, Subhansu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.
- [28] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *Computer Vision–ECCV 2016 Workshops*, pages 236–250. Springer, 2016.

- [29] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.
- [30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.
- [31] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR 2018-IEEE Conference on Computer Vision & Pattern Recognition*, 2018.
- [32] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.
- [33] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 199–208. IEEE, 2017.
- [34] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Gutttag, and Adrian V Dalca. An unsupervised learning model for deformable medical image registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9252–9260, 2018.

- [35] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [36] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. *arXiv preprint arXiv:1612.00603*, 2016.
- [37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
- [38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.
- [39] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.
- [40] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.
- [41] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

- [42] Enliang Zheng and Changchang Wu. Structure from motion using structureless resection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2075–2083, 2015.
- [43] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.
- [44] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005.
- [45] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [46] Yehezkel Lamdan, Jacob T Schwartz, and Haim J Wolfson. Object recognition by affine invariant matching. In *Proceedings CVPR'88: The Computer Society Conference on Computer Vision and Pattern Recognition*, pages 335–344. IEEE, 1988.
- [47] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016.
- [48] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6917–6925, 2018.

- [49] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Snet: Learning semantic correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1831–1840, 2017.
- [50] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems*, pages 1658–1669, 2018.
- [51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [52] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [53] M Everingham. The pascal visual object classes challenge 2011 (voc2011) results. In <http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html>, 2011.
- [54] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *2011 International Conference on Computer Vision*, pages 1792–1799. IEEE, 2011.
- [55] Jaechul Kim, Ce Liu, Fei Sha, and Kristen Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2314, 2013.

- [56] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323, 2016.
- [57] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2013.