Isometric Correspondence Estimation and Symmetry Detection on Point Clouds Acquired by Depth Cameras

by

Emel KÜPÇÜ

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

KOÇ
UNIVERSITY

September, 2018

Isometric Correspondence Estimation and Symmetry Detection on Point Clouds Acquired by Depth Cameras

Koç University
Graduate School of Sciences and Engineering
This is to certify that I have examined this copy of a doctoral dissertation by

Emel KÜPÇÜ

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Yücel YEMEZ (Advisor)

Prof. A. Murat TEKALP

Assoc. Prof. Engin ERZİN

Assoc. Prof. Yusuf SAHİLLİOĞLU

Asst. Prof. Helin DUTAĞACI

Date:

To surprises of life...

ABSTRACT

We propose iterative isometric point correspondence methods that rely on diffusion distance to handle challenges posed by commodity depth cameras, which usually provide incomplete and noisy surface data exhibiting holes and gaps. We formulate the correspondence problem as finding an optimal partial mapping between two given point sets, minimizing deviation from isometry. Our algorithm starts with an initial rough correspondence between keypoints, obtained via any point matching technique. This initial correspondence is then pruned and updated by iterating a perfect matching algorithm until convergence to find as many reliable correspondences as possible. The resulting set of sparse but reliable correspondences then serves as a base matching from which a dense correspondence set is estimated.

We additionally provide a global intrinsic symmetry detection technique which clusters a point cloud into symmetric sides. We incorporate this technique into our point-based correspondence methods, further improving the reliability of our results on shapes exhibiting intrinsic symmetries such as human models. The experiments show that our methods provide state of the art performance over depth frames exhibiting occlusions, large deformations, and topological noise.

ÖZETÇE

Kolaylıkla bulunabilen derinlik sensörlerinden elde edilen, genellikle delikler ve boşluklar içeren, kısmi ve gürültülü yüzey bilgisinin problemlerinin üstesinden gelen ve yayılım uzaklığına dayanan, yineleyen izometrik nokta eşlemesi yöntemleri önermekteyiz. İki nokta kümesi arasında eşleme problemini, izometriden sapmayı en aza indiren en uygun kısmi eşleme bulma olarak formüle etmekteyiz. Algoritmamız, özellikli noktalar arasında herhangi bir nokta eşleme yöntemi ile elde edilen kabaca bir ilk eşleme ile başlar. Bu ilk eşleme, olabildiğince güvenilir bir eşleme kümesi bulunana kadar tekrarlayan tam eşleme algoritması ile budanıp güncellenir. Elde edilen seyrek ama güvenilir eşlemeler, daha sonra yoğun bir eşleme kümesi elde etmede taban olarak kullanılır.

Ek olarak, bir nokta kümesini iki simetrik tarafa kümeleyen bir bütünsel içsel simetri algılama yöntemi sağlamaktayız. Bu tekniği, insan modelleri gibi içsel simetriler sergileyen şekiller üzerindeki sonuçlarımızın güvenilirliğini daha da arttıracak şekilde, nokta temelli eşleme yöntemlerimiz ile birleştirmekteyiz. Yapılan deneyler yöntemlerimizin tıkanma, büyük bozulumlar ve topolojik gürültü sergileyen derinlik verileri üzerinde karşılaştırmalı olarak en iyi performansı verdiğini göstermektedir.

ACKNOWLEDGMENTS

I thank my advisor Yücel Yemez for his support throughout my PhD. I thank all my jury members (A. Murat Tekalp, Engin Erzin, Yusuf Sahillioğlu, and Helin Dutağacı) for their constructive feedback.

I sincerely thank my friends Özlem Yalçın and Didem Unat for their help during tough personal conditions that I faced. I also want to thank my friends in the ENG Z20 office and other friends and faculty at Koç University, Marmara University, and elsewhere for their very enjoyable chats and times we spent together.

I especially thank my precious mother and father who taught me patience and perseverance, and my dear siblings Devrim, Özlem, and Deniz, who are always with me whenever I need. Finally, I thank my strong husband Alptekin Küpçü for making our life beautiful under the daily adventures that we experience together.

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grants 114E628 and 215E201. We thank the authors of [Guo et al., 2015] for sharing their code and dataset with us.

TABLE OF CONTENTS

List of Tables ix

List of Figures x

Chapter 1: Introduction 1

- 1.1 Contributions . . . 2
- 1.2 Organization . . . 5

Chapter 2: Related Work 6

- 2.1 Correspondence Estimation . . . 6
- 2.2 Intrinsic Symmetry detection . . . 9

Chapter 3: Correspondence Estimation 11

- 3.1 Overview . . . 11
- 3.2 Preprocessing . . . 13
  - 3.2.1 Point-based Laplacian . . . 13
  - 3.2.2 Keypoint Detection and Matching . . . 15
- 3.3 Base Correspondence Estimation . . . 15
  - 3.3.1 Diffusion-based Perfect Matching . . . 15
  - 3.3.2 Iterative Pruning and Update . . . 18
- 3.4 Dense Correspondence Estimation . . . 21
- 3.5 Complexity . . . 26

Chapter 4: Intrinsic Symmetry Detection 27

- 4.1 Symmetry with Plane Fitting . . . 27

4.2 Clustering-based Symmetry 29
    4.2.1 Symmetry Detection 29
    4.2.2 Clustering into Sides 32
    4.2.3 Symmetry-aware Matching 35
4.3 Complexity 36

Chapter 5: Experimental Preliminaries **39**
5.1 Datasets 39
    5.1.1 Existing Datasets Employed 40
    5.1.2 New Human Dataset 40
5.2 Baseline Methods 40
5.3 Evaluation Strategy 42

Chapter 6: Results **43**
6.1 Implementation Details 43
6.2 Intrinsic Symmetry Results 44
    6.2.1 Comparison to Planar Symmetry 54
6.3 Sparse Correspondence Results 55
    6.3.1 Examples 58
6.4 Dense Correspondence Results 64
    6.4.1 Examples 71
6.5 Results Regarding Applications 75

Chapter 7: Conclusion and Future Work **82**

Bibliography **83**

LIST OF TABLES

6.1 Quantitative evaluation of our **sparse** (base) correspondence estimation method using **plane fitting symmetry** extension in comparison to the baseline methods. (GT: Ground-truth, Errors $\times 10^{-5}$) 55

6.2 Quantitative evaluation of our **sparse** (base) correspondence estimation method in comparison to the baseline methods. (GT: Ground-truth, Errors $\times 10^{-5}$) 57

6.3 Quantitative evaluation of our **sparse** (base) correspondence estimation method **with and without symmetry detection**. (GT: Ground-truth, GT-SF: GT Error with symmetric flips ignored, C-SD: Clustering-based Symmetry Detection, P-SD: Planar Symmetry Detection, Errors $\times 10^{-5}$) 58

6.4 Quantitative evaluation of our **dense** correspondence method in comparison to baseline methods. (Human^d: Human dataset downsampled to half resolution; Errors $\times 10^{-5}$) 65

6.5 Quantitative evaluation of our **base** correspondences in comparison to the **fine-tuned base** correspondences. (Human^d: Human dataset downsampled to half resolution; Errors $\times 10^{-5}$) 69

6.6 Quantitative evaluation of **sparse** correspondence results between **complete and partial** models in comparison to the baseline methods. (GT: Ground Truth, Errors $\times 10^{-5}$) 76

6.7 Quantitative evaluation of **across dataset dense** correspondence estimation. (GT: Ground Truth, Errors $\times 10^{-5}$) 76

LIST OF FIGURES

1.1 Red line: incorrect, blue line: correct correspondences. (a) Initial random correspondences with **0%** precision. (b) Results of our method with **100%** precision. 3

3.1 Block diagram of our overall depth correspondence method. 12
3.2 Flow chart of Mesh-free Discrete Laplacian Matrix Calculation. . . . 14
3.3 Diffusion distance to a surface point (marked with red). 17
3.4 A cost entry calculation for a candidate matching. 17
3.5 a) Subdividing the point clouds into patches, b) Sampling of vertices in each patch, c) Matching sampled points in each patch, d) Pruned correspondences in each patch. 22
3.6 a) Regrouping the vertices into subpatches, b) Sampling of new points in each subpatch, c) Matching the sampled vertices and pruning, d) Regrouping the vertices into further subpatches. 23

4.1 a) Sampling of points from point cloud, b) Initial pairs, c) One-to-one pairs, d) Pruned symmetric pairs, e) Two symmetric segments of the point cloud, f) Representation of absolute votes. 31
4.2 Symmetry-aware base correspondence estimation: a) Finding symmetric sides via plane fitting and matching, b) Clustering into symmetric sides with our method and matching. 37

5.1 An example challenging depth frame from the MHAD dataset. 39
5.2 Depth datasets with ground-truth keypoints displayed in green. . . . 41

6.1 Two symmetric sides of the point clouds of the Human dataset. . . . 45

6.2 Two symmetric sides of the point clouds of the MHAD dataset. . . . 46

6.3 The absolute values of the symmetry weights of the example point clouds of the Human dataset. 47

6.4 The absolute values of the symmetry weights of the example point clouds of the MHAD dataset. 48

6.5 Penalized (yellow) and unpenalized (blue) area examples from the Human dataset. 49

6.6 Penalized (yellow) and unpenalized (blue) area examples from the MHAD dataset. 50

6.7 Further examples showing two symmetric sides and symmetry weights of the point clouds. 51

6.8 Further examples showing two symmetric sides and symmetry weights of the point clouds. 52

6.9 Further examples showing two symmetric sides and symmetry weights of the point clouds. 53

6.10 The worst matchings on the MHAD dataset for each method according to ground-truth (left column, black lines) and isometric errors (right column, green lines). 60

6.11 The worst matchings on the MHAD dataset for each method according to ground-truth (left column, black lines) and isometric errors (right column, green lines). 61

6.12 Example correspondence results with automatic keypoint detection. . 62

6.13 Example base correspondence results (a) **without** and (b) **with** our clustering-based symmetry solution. 63

6.14 Patchwise dense correspondences, (from left to right) at levels $k = 0$, $k = 1$ and $k^* = 9$ (last level). 64

6.15 Error fraction plots of dense correspondence results of the methods on different datasets. 66
6.16 Example dense correspondence results on the Human dataset. 67
6.17 Dense correspondence on a sample pair from the MHAD dataset. . . 68
6.18 Dense correspondence on a sample pair from the dataset of [Guo et al., 2015]. 70
6.19 Sparse and Dense correspondences on sample pairs from the Human dataset. 72
6.20 Sparse and Dense correspondences on sample pairs from the MHAD dataset. 73
6.21 Sparse and Dense correspondences on sample pairs from the dataset of [Guo et al., 2015]. 74
6.22 Example results for matching partial depth frames to a reference complete model. 77
6.23 Across dataset depth correspondence examples. 78
6.24 Across dataset depth correspondence examples. 79
6.25 Across dataset depth correspondence examples. 80
6.26 Across dataset depth correspondence examples. 81

Chapter 1

INTRODUCTION

Depth sensors have become popular in the last decade, and this has opened up new opportunities in the field of computer vision and graphics as well as brought new challenges. The data collection hardware set-ups to construct accurate results, such as moving camera, multiple cameras, laser scanner, and projector set-ups, are mostly expensive. With the technological improvements, we are able to use much cheaper hardware set-ups for data collection.

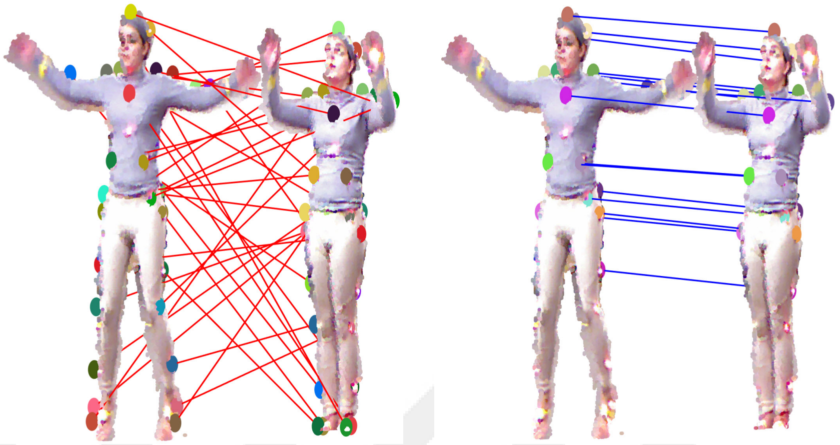
Finding correspondences from depth is a key step for the success of various tasks in 3D computer vision, such as registration [Chang and Zwicker, 2011] and reconstruction [Liao et al., 2009, Tong et al., 2012, Wang et al., 2012, Weiss et al., 2011]. We can categorize 3D correspondence problem into *rigid* and *non-rigid* cases. Finding matching points between different views of a rigid object is easier than that of a non-rigid object [Tam et al., 2013], since the camera sees the same stable structure of the rigid object from each different view. So, the information consistency of points on the object is higher in the rigid case. For the non-rigid case, it is more difficult to get the same consistency, especially for a complicated object such as a dynamic human body. The non-rigid and articulated structure, the continuous and uncontrolled muscle movements, and other biological activities of the human body make finding accurate correspondences much more difficult than that of rigid objects.

Although the field of 3D shape correspondence has become quite mature in the last decade, finding reliable correspondences from depth, especially for non-rigid objects, is still an open problem. The first challenge is due to noisy data provided by commodity depth sensors, exhibiting holes and large gaps. This makes estimation of geodesic distances on the surface geometry very difficultMoreover, when objects undergo non-rigid deformation, their topology can change drastically. Second, depth data is incomplete by acquisition since objects can be sensed only from one direction; hence correspondences exist only partially. Third, due to holes and gaps, surface information cannot reliably be triangulated; hence there is a need for point-based matching techniques that are currently less mature compared to existing mesh-based correspondence techniques.

The most common and generic approach for non-rigid point correspondence is to match individual surface points based on local shape descriptors [Boscaini et al., 2016, Carrière et al., 2015, Johnson, 1997, Rusu et al., 2009, Tombari et al., 2010]. However, since a local approach discards global shape cues such as isometry, it can easily yield incorrect correspondences especially when the shapes exhibit large variations in their local geometry, or when there are many points that are locally similar. A number of works in the literature address the problem of isometric point matching [Berger and Silva, 2012, Guo et al., 2015, Huang et al., 2008, Li et al., 2008, Ma et al., 2014, Mateus et al., 2008, Tevs et al., 2009]. Most of these works however face a decline in performance in the case of noisy and incomplete data since they rely on fitting intermediate mesh-based representations to point clouds. We show the advantages of our method against three point-based state of the art methods [Myronenko and Song, 2010, Ma et al., 2016, Guo et al., 2015] in Chapter 6.

1.1 Contributions

In this thesis, we have 3 major contributions. The first contribution is a mesh-free point-based method that can estimate reliable sparse (*base*) correspondences on non-rigid objects undergoing large isometric deformations from noisy and incomplete depth data. We formulate the correspondence problem as finding an optimal partial mapping between two given point sets, minimizing deviation from isometry. We measure deviation from isometry based on a diffusion-based distance metric that we compute in a robust manner over noisy point clouds using an approximation of graph



(a) (b)

Figure 1.1: Red line: incorrect, blue line: correct correspondences. (a) Initial random correspondences with **0%** precision. (b) Results of our method with **100%** precision.

Laplacian.

We show that starting from an initial correspondence obtained by any point matching technique, it is possible to iteratively prune and update the initial matching, and obtain a set of sparse but *reliable* correspondences even with challenging noisy and incomplete data under occlusion that can further be extended to a partial dense correspondence. Reliability is achieved thanks to our iterative pruning and update strategy that eliminates the unreliable matchings according to an isometric criterion. Our algorithm outputs one-to-one but not necessarily onto correspondences. In our experimental results, we have cases where the initial correspondences are completely incorrect, and we still obtain up to 100% precision (see Figure 1.1 for an example). The sparse correspondence estimation algorithm is presented in detail in Section 3.3 and was published as [Küpçü and Yemez, 2017].

Second, we develop a point-based partial dense correspondence estimation method. We use a coarse-to-fine matching strategy to achieve dense correspondence in a computationally efficient manner. We start with a sparse set of correspondences, and add more correspondences based on the initial sparse set with a recursive patch-based strategy. We sample points in each patch, match the points in between the corresponding patches, and prune the unreliable matchings at each recursive step. Our algorithm takes the reliable base correspondences resulting from our sparse matching technique mentioned above, and outputs a dense set of reliable correspondences. The details are given in Section 3.4.

Another major intricacy in correspondence estimation is the symmetrical flip problem, which is actually inherent to all isometric methods. Our third major contribution is a global intrinsic symmetry detection technique, which clusters a point cloud into symmetric sides. The input of our algorithm is a noisy, partial point cloud of a human model with nonrigid motion. We first sample points from the point cloud to match them using a distance-based descriptor to find initial pairs. Then we eliminate the non-symmetric pairs using a diffusion distance-based deviation from symmetry criterion. The resulting symmetric pairs vote for the side of each point such that we segment the point cloud into two symmetric sides. Finally, we incorporate our clustering-based symmetry technique into our point-based correspondence method that further improves the reliability of our results on shapes with intrinsic symmetries such as human models. The solution is presented in Chapter 4.

We have two additional contributions. We adapted the planar symmetry methodology of Lipman et al.. [Lipman et al., 2010] to our point-based correspondence estimation method penalizing the points on the symmetrically flipped sides of the two models during our iterative sparse matching process. This technique was also published in [Küpçü and Yemez, 2017] and is explained in Section 4.1. Moreover, we collected depth frames of a freely moving human subject using a commodity depth sensor. The collected dataset exhibits noisy and partial structure with large deformation which also causes severe topological changes and occlusions. We employed this challenging 3D point clouds of human model dataset to ensure our experiments are realistic. The dataset is discussed and displayed in Section 5.1.

1.2 Organization

In the next chapter, we discuss the related work in the literature about correspondence estimation and symmetry detection. In Chapter 3, we explain the preprocessing of the point cloud dataset (Section 3.2), the detailed methodology of our sparse (Section 3.3) and dense (Section 3.4) correspondence estimation techniques. Afterward, we explaind the works on symmetry detection methods and provide the details in Chapter 4. In Section 4.1, we explain the plane fitting symmetry detection strategy we used in our conference paper [Küpçü and Yemez, 2017]. We give also the details of our new clustering-based symmetry method in Section 4.2. We provide the integration details of our correspondence estimation and symmetry detection methods in Section 4.2.3. In Chapter 5, we explain the datasets we employed, the baseline methods we compare against, and our evaluation strategy. Finally, in Chapter 6, we provide the results of our work based on symmetry detection (Section 6.2), sparse (Section 6.3) and dense (Section 6.4) correspondence estimation, and then conclude.

Chapter 2

RELATED WORK

In this chapter, we discuss the most related works to our correspondence estimation and symmetry detection algorithms.

2.1 *Correspondence Estimation*

The correspondence estimation problem is still popular and attracts attention to improve the results for different environmental and technological cases. Especially, finding correspondences between non-rigid shapes, such as a human body, need to be researched more deeply. The non-rigid correspondence estimation methods make use of *isometry* for obtaining accurate results. An isometry is a transformation that preserves the distance between surface points.

Isometric deformations are the most common forms of non-rigidity. There exists a vast literature on isometric shape matching [Biasotti et al., 2016, Tam et al., 2013], including various methods that can find sparse [Bronstein et al., 2006, Küpçü and Yemez, 2017, Rodola et al., 2012, Rodola et al., 2013, Sahillioğlu and Yemez, 2014, Zhang et al., 2013] and/or dense [Bronstein et al., 2010, Chen and Koltun, 2015, Van Kaick et al., 2011, Kim et al., 2011, Ovsjanikov et al., 2012, Raviv et al., 2013, Sahillioğlu and Yemez, 2011, Sharma et al., 2011] correspondences between complete mesh representations of 3D shapes. Yet, finding correspondences between representations with partial overlap is still an active research topic with solutions mostly devised to handle partially correspondent mesh models [Bronstein et al., 2006, Bronstein and Bronstein, 2008, Cosmo et al., 2016, Lipman and Funkhouser, 2009, Litany et al., 2016, Litany et al., 2017, Ovsjanikov et al., 2010, Sahillioğlu and Yemez, 2014, Wang et al., 2011, Yoshiyasu et al., 2016, Yoshiyasu et al., 2014].

Most of the existing mesh-based isometric correspondence methods rely on geodesic distance information [Bronstein et al., 2006, Chen and Koltun, 2015, Sahillioğlu and Yemez, 2011, Sahillioğlu and Yemez, 2012, Sahillioğlu and Yemez, 2014]. However, conventional ways of computing geodesic distance, which provide the shortest path on a surface using the connections between neighboring points, become problematic on noisy surfaces with holes and gaps. A better alternative for noisy mesh data with large gaps is employing diffusion-based distance [Bronstein et al., 2010, Coifman and Lafon, 2006, Rodola et al., 2012, Wang et al., 2011, Yoshiyasu et al., 2014]. Diffusion distance takes into account all the paths existing between two surface points based on eigen decomposition of the Laplacian matrix (see Section 3.2.1), thereby reducing the negative impact of topological noise and incompleteness on estimation of distances.

There exist a good number of point correspondence estimation techniques that can handle unorganized point cloud data [Berger and Silva, 2012, Brunton et al., 2014, Guo et al., 2015, Huang et al., 2008, Kovnatsky et al., 2015, Li et al., 2008, Rodolà et al., 2017, Tevs et al., 2009, Wei et al., 2016]. Most of these methods, except [Berger and Silva, 2012, Guo et al., 2015, Kovnatsky et al., 2015, Rodolà et al., 2017, Tevs et al., 2009], are not actually mesh-free techniques, relying on intermediate mesh representations fit to input point clouds, so that geodesic distances can be computed.

The mesh-free method presented by Guo et al. [Guo et al., 2015] addresses the correspondence problem through piecewise rigid point registration by discovering parts in an iterative process. Their method hence relies heavily on approximation of geodesic distances by k-nearest neighbor graph distances and correct estimation of rigid parts, which can be problematic especially in the case of occlusions as with depth data provided by commodity sensors. The updating and pruning core of our correspondence algorithm resembles in spirit to the RANSAC-like method proposed in [Tevs et al., 2009] which relies again on geodesic distances through a probabilistic approach, while our formulation is completely different than that of [Tevs et al., 2009]. Another related method is the point-based dense correspondence technique of [Berger and Silva, 2012], which uses medial diffusion to deal with incomplete data. However, their medial axis prior is heavily dependent on shape topology and intolerant to large missing data, making it inapplicable to partial depth data with severe occlusions.

Another line of correspondence methods is based on the functional maps framework which was first introduced in [Ovsjanikov et al., 2012]. In theory, given the graph Laplacian of two shapes represented as point clouds or meshes, these methods can match complete models with resiliency against missing parts and noise [Kovnatsky et al., 2015], and can even handle the parts matching problem via matrix completion [Rodolà et al., 2017]. These methods however cannot yet explicitly cope with partial and occluded depth data as provided by commodity sensors.

Probabilistic non-rigid registration techniques also provide us, as a byproduct, with point correspondences between point clouds [Billings et al., 2015, Chui and Rangarajan, 2003, Horaud et al., 2011, Jian and Vemuri, 2011, Ma et al., 2016, Mateus et al., 2008, Myronenko and Song, 2010]. They find correspondences by optimizing a global objective to align point sets. Myronenko and Song [Myronenko and Song, 2010] and Ma et al. [Ma et al., 2016] introduce non-rigid point registration methods that estimate parameters of transformations using Gaussian mixture models (GMMs). In addition to global cues, Ma et al. [Ma et al., 2016] incorporate local features to take into account similarity of neighboring structure of points. While point registration methods generate correspondences by matching all the points available, our focus is on finding partial mappings between point clouds based on isometric cues, with as many reliable correspondences as possible.

2.2 Intrinsic Symmetry detection

Consider a symmetric shape where a point has intrinsic distances to the other points similar to a point symmetric to it with respect to a plane, line, or a point. For example, on a human body, a point on the left hand is potentially symmetric to a point on the right hand. A symmetric flip occurs when a point is matched with its symmetric point instead of the true correspondence due to their intrinsic metric similarity. The symmetric flip problem is inherent to all isometric matching methods. There exist several shape correspondence methods in the literature, addressing explicitly this problem such as in [Dubrovina and Kimmel, 2011, Ovsjanikov et al., 2012, Ovsjanikov et al., 2010, Sahillioğlu and Yemez, 2013, Yoshiyasu et al., 2016, Yoshiyasu et al., 2014, Zhang et al., 2013], which are all mesh-based. The symmetric flip problem is also closely related to the problem of symmetry detection. We focus on intrinsic symmetry, which is addressed in various works with different applications [Jiang et al., 2013, Lipman et al., 2010, Mitra et al., 2006, Mitra et al., 2013, Ovsjanikov et al., 2008, Raviv et al., 2007, Raviv et al., 2010a, Raviv et al., 2010b, Sipiran et al., 2014, Wang et al., 2014, Xu et al., 2009, Xu et al., 2012].

Xu et al. [Xu et al., 2012] find and group symmetrical point pairs with respect to an isometric symmetry criterion between sampled points. They then generate symmetry scales based on intrinsic distance profiles which are histograms constructed using geodesic distances. For each scale, they perform spectral clustering on symmetric point pairs such that partially symmetric objects or parts existing in the scene are grouped together. Jiang et al. [Jiang et al., 2013] detect symmetric regions on complete or partial point clouds, where a given point cloud represents a single object. Their method requires skeletal information to first find symmetric point pairs on the skeleton, and then based on these pairs, it outputs a set of non-overlapping clusters each representing an object part which is intrinsically symmetric in itself. Likewise, Raviv et al. [Raviv et al., 2010b] find symmetric point pairs and detect symmetric parts on non-rigid smooth mesh models using the diffusion distance metric, whereas the works in [Raviv et al., 2007, Raviv et al., 2010a] address intrinsic symmetry of non-rigid models based on local and global shape asymmetry measures calculated using geodesic distances.

Rather than detecting symmetric parts in a scene or an object as in the works cited above, in this work we tackle the problem of clustering a given point cloud into "sides" which are globally and intrinsically symmetric such as in human shapes. We then incorporate this side information into our correspondence estimation method in order to alleviate the symmetric flip problem.

Inspired by the work of Podolak et al. [Podolak et al., 2006], which finds partial reflectional symmetries on 3D mesh models in a scene, Xu et al. [Xu et al., 2009] propose a voting based method to segment a model into two symmetric sides. They find point pair sets that vote for Voronoi boundaries between pairs to obtain the best intrinsic reflectional curve for each object in the scene.Similarly, Lipman et al. [Lipman et al., 2010] apply their intrinsic symmetry detection methodology to represent global reflectional symmetry with respect to a plane rather than a curve as in [Xu et al., 2009]. To this end, they fit a plane to the centroids of the symmetry orbits, which are calculated by measuring the global intrinsic symmetry dissimilarity for each point using the eigenvectors of the Laplacian matrix. For a given point cloud, a symmetry plane –even if approximate– can always be found using their method. Küpçü and Yemez [Küpçü and Yemez, 2017] use the plane fitting methodology of Lipman et al. to reduce the symmetrically flipped results in their correspondence estimation scheme. In this current work, we develop a clustering-based approach to determine the symmetric sides of a point cloud, which does not require any mesh or skeleton information and yet works reliably on incomplete noisy depth data. We will describe our clustering-based method in Chapter 4 in detail and discuss the advantages of this method over the plane fitting strategy of [Lipman et al., 2010]. In brief, we first find a sparse set of intrinsically symmetric point pairs on the point cloud based on isometric cues. Then, each pair votes for the side of every point on the point cloud based on a proximity measure to cluster the points into two symmetric sets, e.g., right and left sides of a human shape.

Chapter 3

CORRESPONDENCE ESTIMATION

In this chapter, we provide the details of our mesh-free sparse (*base*) and dense correspondence estimation approaches for non-rigid objects undergoing large isometric deformations. We start with a brief overview, and finish with the complexity analysis of our methods.

3.1 Overview

Our overall isometric depth correspondence algorithm consists of three main parts as visualized in Figure 3.1: preprocessing, base correspondence estimation and dense correspondence estimation. The input is a pair of point clouds obtained from depth frames of the object of interest. In the preprocessing step (Section 3.2), we first apply a standard keypoint detection technique, and then match the detected points using any point-based matching algorithm. The resulting matching serves as the initial correspondence to be further improved. In this step, we also compute the graph Laplacian matrices of both point clouds, which we will then be used to calculate diffusion based distances between keypoints.

During base correspondence estimation (Section 3.3), the initial correspondence set computed in the preprocessing step is iteratively pruned and updated until convergence, employing a diffusion-based perfect matching technique. This step contains two nested loops, the outer one for update and the inner one for pruning. For the diffusion-based matching task of the inner loop, we construct an isometric cost matrix and apply perfect matching. Each entry of the cost matrix represents the deviation of a given correspondence pair from isometry, which can be computed using diffusion distances only if a set of (base) correspondences is known a priori; hence there is a need

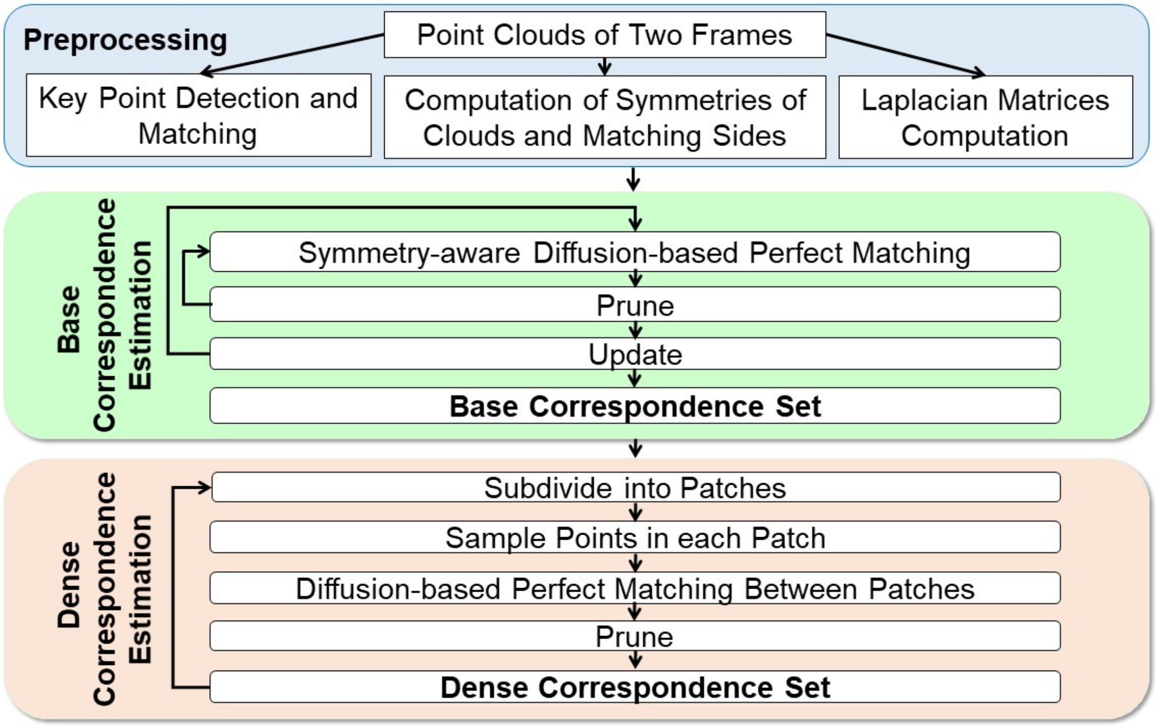


Figure 3.1: Block diagram of our overall depth correspondence method.

for the initial correspondence obtained via descriptor matching. The diffusion-based matching yields an updated correspondence set that usually includes incorrect matchings due to incomplete and noisy nature of the data, and thus needs to be pruned. The pruning iterations eliminate these outliers one by one based on an isometric error criterion, each time repeating the diffusion-based matching process, hence providing a smaller but more reliable set of correspondences. This relatively more reliable set replaces the existing correspondences at the beginning of each outer loop iteration, and is gradually improved until convergence.

We then use the resulting base correspondence set for estimating a dense set of correspondences (Section 3.4). To this effect, we recursively subsample the point clouds into sub-patches relying on the base correspondences, and perform matching in a patch by patch basis, each time followed by elimination of outliers. The recursion continues until no point is left to match. The initial base correspondences are also gradually fine-tuned throughout the levels of this recursive process.

An optional task during preprocessing is to cluster both point clouds (partially if shape data is not complete) into intrinsically symmetric sides and then to match the corresponding sides (Chapter 4). The goal is to address the symmetrical flip problem that inherently occurs especially while matching symmetric human shapes. We introduce a symmetry-aware version of the cost matrix that we use for perfect matching, along with a novel voting-based algorithm to find corresponding symmetric sides on point clouds. We penalize matching of the points located on the non-corresponding sides of the given shapes (hence those which are potentially symmetrically flipped).

3.2 Preprocessing

3.2.1 Point-based Laplacian

To calculate the diffusion distance between two given points on a point cloud, we compute the graph Laplacian directly on the point representation in a similar way as described in Belkin and Niyogi [Belkin and Niyogi, 2008]. To do this, we first find, for each point x_n in the point set $X = \{x_1, x_2, ..., x_N\}$, the K nearest neighboring points $\{y_1, y_2, ..., y_K\}$ within a predesignated fixed distance threshold and generate a graph structure. We then calculate a weight $w(i,k)$ between x_i and each y_k in the neighborhood (all initially set to 0) by $w(i,k) = e^{-\frac{||x_n - y_k||^2}{\epsilon}}$, where the parameter ϵ is related to the average extent of the neighborhood [Petronetto et al., 2013] and given by $\epsilon = \frac{1}{N}\sum_{n=1}^{N} \max_k ||x_n - y_k||$. The weights form a symmetric weight matrix W. We then construct a diagonal matrix D, where each diagonal entry (i,i) is the sum of the weights $w(i,k)$ over k for a given point x_i. Finally, the Laplacian matrix is given by $L = D - W$. The flow chart of Laplacian matrix calculation is given in Figure 3.2. We denote the source and target point clouds extracted from depth frames by P^S and P^T, and their corresponding Laplacian matrices by L_S and L_T, respectively. The smallest M non-zero eigenvalues $\Lambda_S = \{\lambda_{S,1}, \ldots, \lambda_{S,M}\}$, $\Lambda_T = \{\lambda_{T,1}, \ldots, \lambda_{T,M}\}$ and the corresponding eigenvectors $\Phi_S = \{\phi_{S,1}, \ldots, \phi_{S,M}\}$, $\phi_T = \{\phi_{T,1}, \ldots, \Phi_{T,M}\}$ are used for computing diffusion distances as explained in Section 3.3.1, where M is an experimentally chosen parameter.

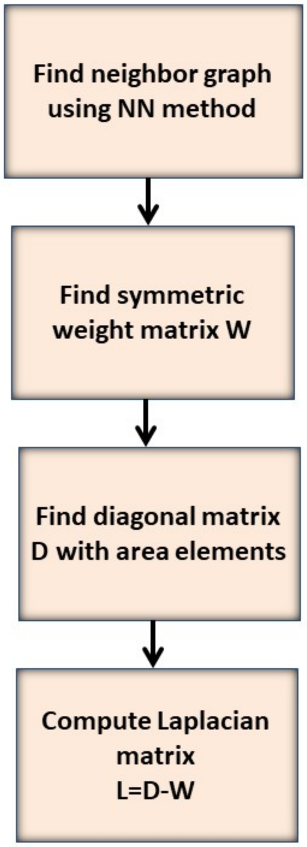


Figure 3.2: Flow chart of Mesh-free Discrete Laplacian Matrix Calculation.

3.2.2 Keypoint Detection and Matching

For initialization, we detect keypoints on the given pair of point clouds and match them using a point matching algorithm. Any point-based 3D keypoint detection algorithm, such as SIFT [Lowe, 2004] or ISS [Zhong, 2009], can be used for this purpose. We represent the keypoint sets for the source and target with $S = \{s_1, s_2, ..., s_{|S|}\}$ and $T = \{t_1, t_2, ..., t_{|T|}\}$, respectively, where the number of detected keypoints usually varies with shape complexity. We match the detected keypoints with a standard descriptor matching algorithm [Aldoma et al., 2012] by using, for example, SHOT [Tombari et al., 2010] or FPFH [Rusu et al., 2009] descriptors, or with a point registration method such as CPD [Myronenko and Song, 2010] and PR-GLS [Ma et al., 2016] (see Section 6.3). We use the resulting correspondence set as the initial base correspondence set B_0 in the next step.

3.3 Base Correspondence Estimation

3.3.1 Diffusion-based Perfect Matching

The detected keypoints can be reconsidered for a better matching based on global isometric clues. For this purpose, we construct an isometric cost matrix C, where each entry c_{ij} represents the deviation from isometry of a candidate correspondence pair. To compute deviations from isometry, we need to rely on a set of known correspondences. We refer to this set as base correspondence, denoted by $B = \{(b_1^S, b_1^T), \ldots, (b_{|B|}^S, b_{|B|}^T)\}$. The set B is initially set to B_0, that is, the correspondence obtained in the preprocessing step via a point-based matching method. We employ the diffusion distance for calculation of the isometric cost [Coifman and Lafon, 2006, Lafon, 2004]:

$$d_{X,t}(x, y) = \sum_{m=1}^{M} e^{-2\lambda_m t}(\phi_m(x) - \phi_m(y))^2, \tag{3.1}$$

where the distance $d_{X,t}(x, y)$ is calculated between two points x and y at time t, and $\{\lambda_m\}$ and $\{\phi_m\}$ are the smallest M eigenvalues and the corresponding eigenvectors of the Laplacian matrix. An important issue here is how to set time parameter t, where a small t emphasizes the local features and large t takes into account the global features of the shape. We define our diffusion distance metric as the average of diffusion distances over a set of time steps $\mathcal{T}$:

$$d_X(x, y) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} d_{X,t}(x, y) \tag{3.2}$$

Hence the diffusion distance $d_X(x, y)$, which is independent of t, can be interpreted as the average of the lengths of all the paths existing between two surface points. Note that $d_X(x, y)$ is an approximation of the commute-time distance metric [Bronstein et al., 2010], and has the advantage of bounding the time scale parameter from above (or below). Figure 3.3 demonstrates the distribution of diffusion distance over the surface.

The isometric cost of matching a keypoint s_i on the source keypoint set S with t_j on the target T is then calculated as follows:

$$c_{ij} = \frac{1}{|B|} \sum_{(b_l^S, b_l^T) \in B} |d_S(s_i, b_l^S) - d_T(t_j, b_l^T)|. \tag{3.3}$$

Figure 3.4 visually represents the calculation of a cost entry for a candidate matching pair.

The resulting cost matrix C is bipartite, so we can apply the Blossom V algorithm, minimum-weight perfect matching algorithm of Kolmogorov [Kolmogorov, 2009], to match all the keypoints from scratch minimizing the sum of cost of one-to-one correspondences, similar to [Sahillioğlu and Yemez, 2014]. Each entry $c_{ij} \in C$ is normalized to be in the range $[0, 1)$ by $c_{ij} \leftarrow (1 - e^{-c_{ij}})$.

Note that the number of keypoints may vary between point sets, whereas the perfect matching algorithm that we employ requires a square cost matrix. Thus, we add virtual nodes to the smaller keypoint set and set the corresponding costs for non-existent pairs to infinity, making C a matrix of size $(max(|S|, |T|))^2$. The perfect matching algorithm results in a one-to-one mapping, from which we then remove the pairs including virtual nodes.

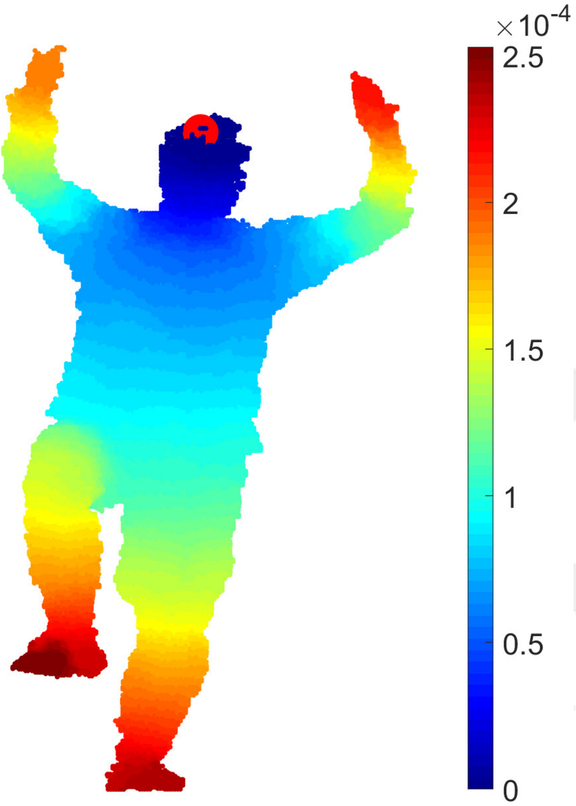


Figure 3.3: Diffusion distance to a surface point (marked with red).

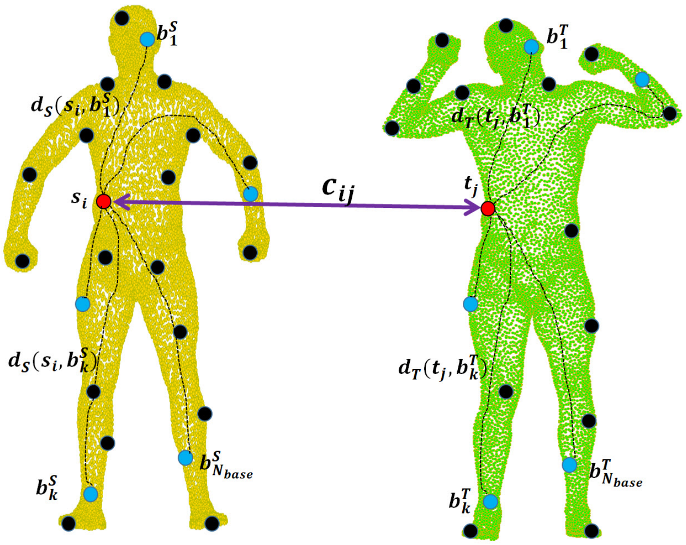


Figure 3.4: A cost entry calculation for a candidate matching.

3.3.2 Iterative Pruning and Update

The perfect matching process described in Section 3.3.1 results in a one-to-one mapping B. We iteratively modify this matching B to make it as reliable as possible. We achieve this goal with a nested loop of *pruning* and *update* iterations. While the inner loop prunes unreliable correspondences based on an isometric error criterion, the outer iterations gradually update the initial correspondence B_0 with which we initialize the base correspondence set in the first place. At the end of these iterations, we expect to end up with a partial one-to-one mapping that establishes a reliable but possibly sparse correspondence between keypoints.

We compute the isometric error, $E_{\text{iso}}(b_i^S, b_i^T)$, of a given correspondence pair $(b_i^S, b_i^T) \in B$, in terms of its deviation from isometry with respect to other available pairs in B:

$$E_{\text{iso}}(b_i^S, b_i^T) = \frac{1}{|B|-1} \sum_{(b_l^S, b_l^T) \in B, i \neq l} |d_S(b_i^S, b_l^S) - d_T(b_i^T, b_l^T)| \tag{3.4}$$

If $(b_i^S, b_i^T) \in B$ is a correct matching pair, its isometric error is expected to be close to zero. Hence the correspondence set B can be pruned by eliminating the pairs having relatively larger errors compared to the others. The reliability of the isometric error defined in Eq. 3.4 depends on the correctness of B itself. Thus we perform pruning in an iterative scheme, one pair (the worst one) at a time, and each time we reinvoke the diffusion-based perfect matching algorithm with the pruned base correspondence set. At each iteration, we also remove the keypoints of the eliminated pair from the keypoint sets S and T. The pruning and perfect matching tasks are iterated until the gap between maximum and minimum isometric errors over the pairs becomes small enough according to a predesignated threshold value τ_1.

The modified correspondence set B resulting from the iterative pruning algorithm is smaller but usually much more reliable than the input correspondence. Hence it can be used to update the initial base correspondence for the next run of the iterative pruning algorithm in the outer loop. At the beginning of each outer iteration, the initial base correspondence is renewed with the current B, whereas the keypoint sets, S and T, are set back to their original content. Hence, the keypoints that are discarded during iterative pruning due to mismatches are reconsidered for other possible matches based on a more reliable estimation of isometric errors. The outer iterations terminate when the mean isometric error converges, i.e., when there is no further improvement on the base correspondence set B. The pseudocode of the base correspondence estimation part is given in Algorithm 1.

Algorithm 1 Base Correspondence Estimation

```
Input: Source and target point clouds $P^S$ and $P^T$
  Error threshold coefficient $\tau_1$
Output: Base correspondence set between keypoints
  $B = \{(b_1^S, b_1^T), \dots, (b_{|B|}^S, b_{|B|}^T)\}$
  PreProcessing
  Detect keypoint sets: $S$ for $P^S$ and $T$ for $P^T$;
  Calculate initial correspondence $B_0$ between keypoints;
  Compute eigenvalues and eigenvectors of Laplacian matrices: $(\lambda_S, \phi_S)$ from $L_S$, and $(\lambda_T, \phi_T)$ from $L_T$;
  Calculate symmetry weights and match the sides of the point clouds (Sections 4.2 and 4.2.3);
  Base correspondence estimation
  $S_{\text{orig}} \leftarrow S, \quad T_{\text{orig}} \leftarrow T, \quad B \leftarrow B_0$;
  repeat
    $S \leftarrow S_{\text{orig}}, \quad T \leftarrow T_{\text{orig}}$:
    repeat
      Update $B$ via diffusion-based perfect matching between $S$ and $T$ (Sections 3.3.1 and 4.2.3);
      Calculate isometric error $E_{\text{iso}}(b_i^S, b_i^T)$ for each pair by Eq. 3.4;
      $\{b_{\max}^S, b_{\max}^T\} \leftarrow \text{argmax } E_{\text{iso}}(b_i^S, b_i^T)$;
      $S \leftarrow S - \{b_{\max}^S\}, \quad T \leftarrow T - \{b_{\max}^T\}$;
      $B \leftarrow B - \{(b_{\max}^S, b_{\max}^T)\}$;
      $maxcost \leftarrow E_{\text{iso}}(b_{\max}^S, b_{\max}^T)$;
      $mincost \leftarrow \min E_{\text{iso}}(b_i^S, b_i^T)$;
    until $maxcost < \tau_1 * mincost$
    $totalcost \leftarrow \sum_{i=1}^{|B|} E_{\text{iso}}(b_i^S, b_i^T)$;
  until $totalcost$ does not improve
```

Sahillioglu and Yemez [Sahillioğlu and Yemez, 2014] perform pruning iteratively using minimum-weight perfect matching algorithm [Kolmogorov, 2009] to match the extremity points of the shapes, similar to our inner loop given in Algorithm 1. In their case, there is no outer loop that updates the base correspondences, which increases reliability in our solution. On the other hand, [Sahillioğlu and Yemez, 2012] also use geodesic-distance based perfect matching iteratively, but instead of pruning, they perform neighbor search to improve the matching quality. Moreover, while their method works on meshes, we work on point clouds.

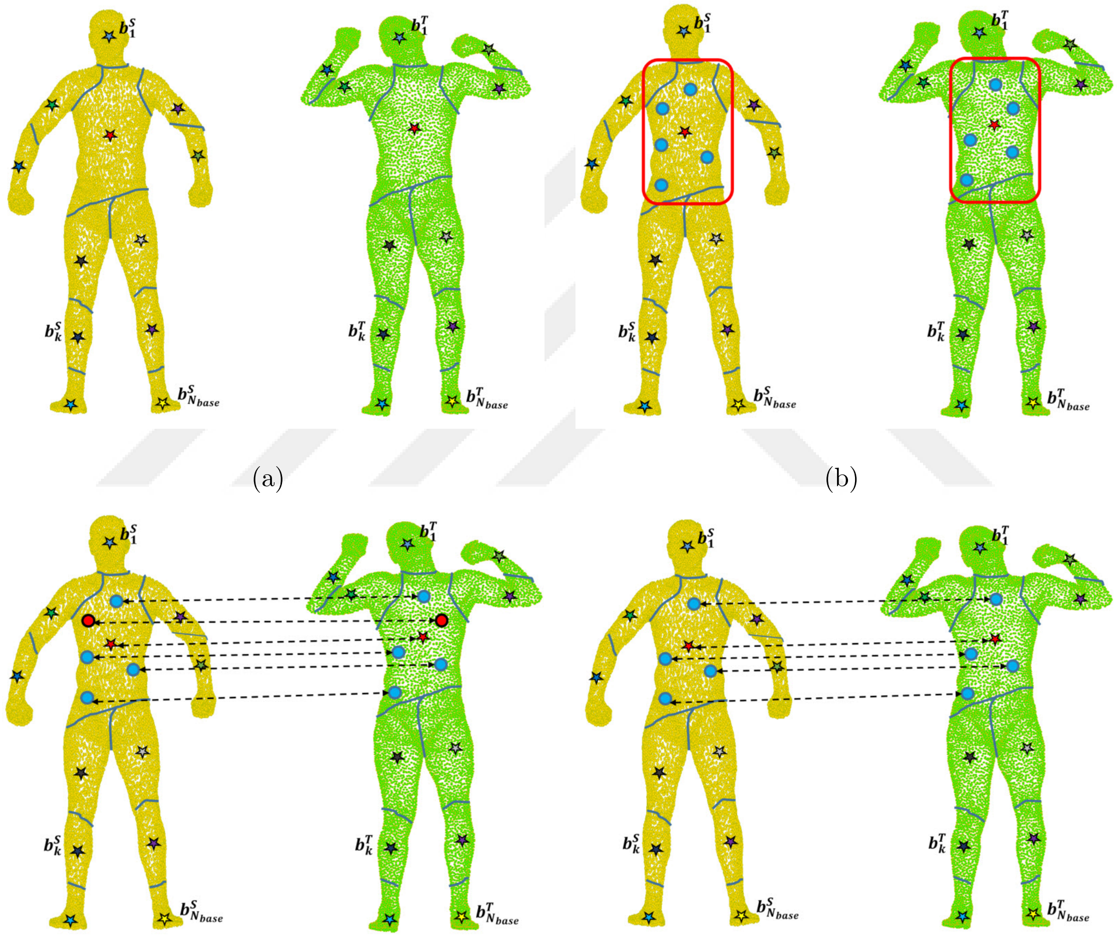
3.4 Dense Correspondence Estimation

We extend the base correspondence set B to a sequence of denser base correspondences $\{B^k\}$ with increasing levels of detail using a coarse-to-fine strategy whose steps are represented in Figure 3.5 and Figure 3.6.

We initialize the base correspondence set at the coarsest level $k = 0$ with $B^0 = B$, and then recursively subdivide the point clouds P^S and P^T into patches via clustering followed by subsampling. Initially, $P^{S,0} = P^S$ and $P^{T,0} = P^T$. We cluster the points within each patch P_l^{k-1} at level $k-1$ starting from $k = 1$ (dropping the superscripts S and T for notational simplicity) into subpatches P_i^k. This is done by grouping points $p_j \in P_l^{k-1}$ according to their diffusion distance to the base points b_i^{k-1} of the previous level patch:

$$P_i^k = \{p_j \in P_l^{k-1} | i = \arg\min_m d(p_j, b_m^{k-1})\}. \tag{3.5}$$

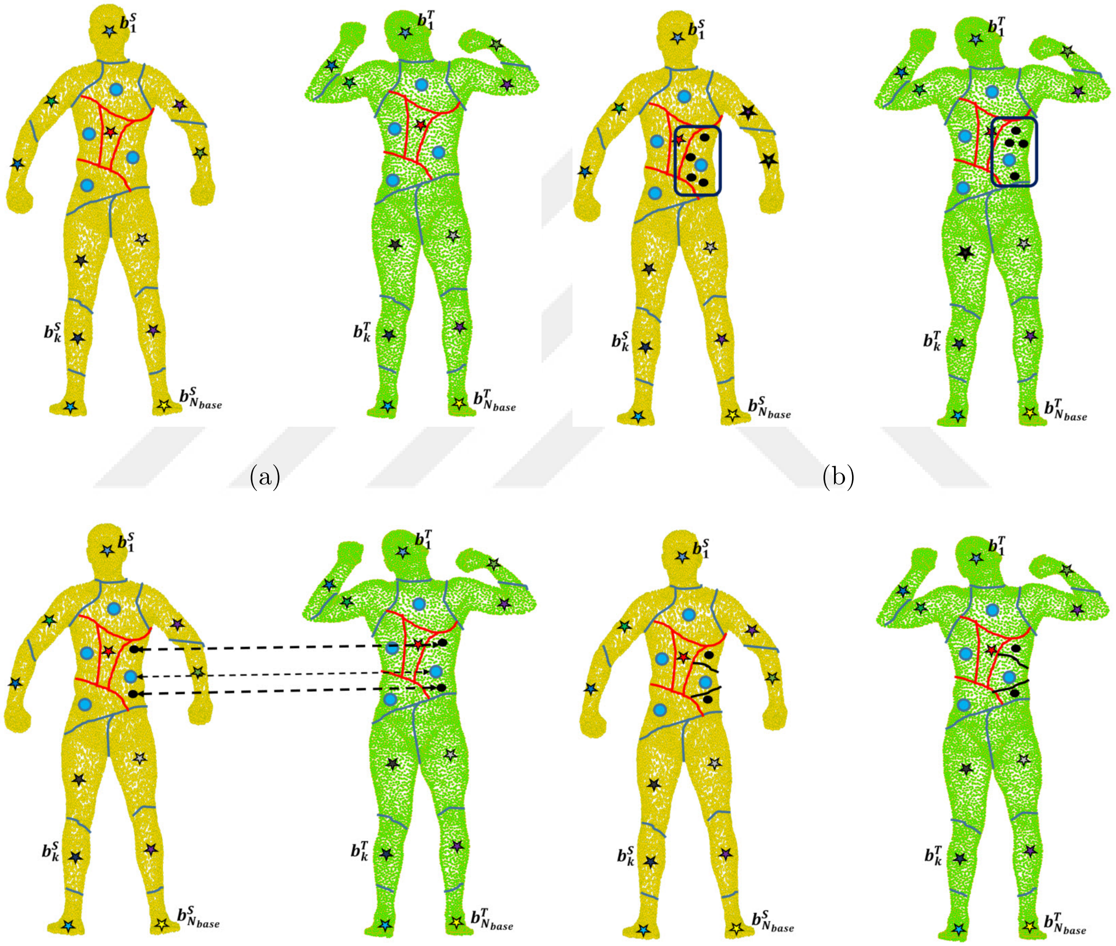
Hence each base point b_i^{k-1} represents a subpatch P_i^k at the next level as displayed in Figure 3.5(a). We then sample at most N_S points as uniformly as possible from each patch as in Figure 3.5(b). We denote the subsampled patch by $\bar{P}_i^k$. For subsampling, we sort all the points within a patch P_i^k in ascending order with respect to their diffusion distances to the representing base point b_i^{k-1}, and uniformly subsample the sorted list. Hence the sampled points tend to be equally distributed within each patch. If the number of points in P_i^k is already less than or equal to N_S, we set



(a) (b)

(c) (d)

Figure 3.5: a) Subdividing the point clouds into patches, b) Sampling of vertices in each patch, c) Matching sampled points in each patch, d) Pruned correspondences in each patch.



(a) (b)

(c) (d)

Figure 3.6: a) Regrouping the vertices into subpatches, b) Sampling of new points in each subpatch, c) Matching the sampled vertices and pruning, d) Regrouping the vertices into further subpatches.

$\bar{P}_i^k = P_i^k$ and stop further recursion for that patch.

The points within each subsampled patch $\bar{P}_i^k$ are then matched with the points in the corresponding subsampled patch of the other point cloud (Figure 3.5(c)). We use perfect matching (as described in Section 3.3.1) to match the sampled points in the corresponding patches, adding virtual vertices whenever there is a difference in the number of sampled points. The isometric costs for perfect matching are computed using Eq. 3.3 based on the correspondence set B^0 regardless of the current level. We denote the resulting patch correspondence by B_i^k. We calculate the isometric error of each correspondence pair in B_i^k based on B^0 by:

$$E_{\text{iso}}(b_i^{S,k}, b_i^{T,k}) = \frac{1}{|B^0|} \sum_j |d_S(b_i^{S,k}, b_j^{S,0}) - d_T(b_i^{T,k}, b_j^{T,0})| \tag{3.6}$$

where $(b_j^{S,0}, b_j^{T,0}) \in B^0$.

The correspondences with isometric errors larger than a threshold are eliminated from the patch correspondence B_i^k (Figure 3.5(d)). To define the threshold, we calculate the maximum error between base correspondences B^0 according to Eq. 3.4. We select the threshold as a factor ($\tau_2 > 1$) of the maximum error. The union of all patch correspondences B_i^k over i generates the base correspondence B^k to be used at the next level. The recursive sampling and matching process displayed in Figure 3.6 continues until no pair of points is left to match between two corresponding patches. We set the resulting correspondences at the highest level as the final dense correspondences B^{k^*}.

We note that the base correspondences in B^0 are fine-tuned gradually throughout the patchwise recursive matching process. At the end of each level k, we consider every base correspondence pair $(b_i^{S,0}, b_i^{T,0})$ for a possible update. If the base point $b_i^{S,0}$ is already matched with $b_i^{T,0}$ at that level (or vice versa), then we continue without updating the base pair. If $b_i^{S,0}$ is matched with a different point x^T and likewise $b_i^{T,0}$ with x^S, then we pick the pair $(b_i^{S,0}, b_i^{T,0})$ or $(b_i^{S,0}, x^T)$ or $(x^S, b_i^{T,0})$, whichever yields the minimum isometric error computed via Eq. 3.6 over the dense correspondence B^k of level k (hence B^0 is simply replaced by B^k in Eq. 3.6 for computation of isometric error in this case).

The above coarse-to-fine matching strategy has two advantages regarding computational cost and robustness. First, patchwise matching reduces the search space considerably providing efficiency as well as robustness, since a point in a patch is very likely to match a point in the corresponding patch on the other point cloud, assuming that initial base correspondences are mostly reliable. Second, it is indeed computationally very demanding to match all the points at once due to large complexity of the perfect matching algorithm, and constraining the number of points sampled within each patch to a reasonably small number (such as $N_S = 50$) makes perfect matching affordable.

Algorithm 2 Dense Correspondence Estimation

```
Input: Base correspondence $B$ and point clouds $P^S, P^T$
Output: Sequence of denser correspondences $\{B^k\}_{k=0}^{k^*}$

  Initialize with $B^0 = B$, $k = 1$, $P^{S,0} = P^S$, $P^{T,0} = P^T$;
  repeat
    For each $(b_i^S, b_i^T) \in B^{k-1}$
      Create patches $P_i^{S,k}$ and $P_i^{T,k}$ using Eq. 3.5;
      Find subsampled $\bar{P}_i^{S,k} \subseteq P_i^{S,k}$ if $|P_i^{S,k}| > N_S$;
      Find subsampled $\bar{P}_i^{T,k} \subseteq P_i^{T,k}$ if $|P_i^{T,k}| > N_S$;
      Find $B_i^k$: $\bar{P}_i^{S,k} \rightarrow \bar{P}_i^{T,k}$ relying on $B^0$ via
        diffusion-based perfect matching (Section 3.3.1);
      Eliminate outliers from $B_i^k$;
    Set $B^k = \bigcup_i B_i^k$;
    Fine-tune base correspondence set $B^0$ using $B^k$;
    Increment level $k$;
  until no unvisited point left to match
```

We finally note that the idea of patchwise recursive matching was originally proposed in [Sahillioğlu and Yemez, 2011] in the context of matching complete mesh representations, whereas our algorithm is designed to robustly work on noisy and partial point cloud representations. We provide the pseudocode of our dense correspondence estimation method in Algorithm 2.

3.5 Complexity

The complexity of our base correspondence estimation algorithm is dominated by the min-weight perfect matching algorithm with $O(N_0^2 \log N_0)$ cost [Kolmogorov, 2009], where $N_0 = \max(|S|, |T|)$, S and T are source and target keypoint sets. In the dense correspondence estimation part, we match $O(\frac{N_P}{N_S})$ patches with $O(N_S)$ points in each, where N_S is the number of points sampled from each patch, $N_P = \max(|P^S|, |P^T|)$, P^S and P^T are source and target point cloud sets. Again, the cost is dominated by the min-weight perfect matching algorithm, which is repeated for each patch and incurs a cost of $O(N_S^2 \log N_S)$ per patch. Therefore, the total complexity of the dense correspondence estimation part is $O(N_S^2 \log N_S) O(\frac{N_P}{N_S}) = O(N_P N_S \log N_S)$. In practice, we have $O(N_0) \sim O(N_S) \sim O(\sqrt{N_P})$. Thus, the overall complexity of our solution can be written as $O(N_P^{\frac{3}{2}} \log N_P)$, which is better than $O(Q^2 N_P)$ complexity of the PR-GLS algorithm [Ma et al., 2016] (one of our baselines), where $Q \sim N_S$ in practice. We note that the CPD algorithm [Myronenko and Song, 2010], which can be seen as precedent to the PR-GLS method, has $O(N_P)$ time complexity but less accuracy as we next demonstrate with experiments in Chapter 6.

Chapter 4

INTRINSIC SYMMETRY DETECTION

Since the shapes to be matched may in general exhibit global intrinsic symmetries such as those in human objects, as an isometric technique, the output of the correspondence method that we have previously described may contain symmetrically flipped correspondences, where a point is matched with its symmetric point instead of the true correspondence due to their intrinsic metric similarity. Indeed, the symmetric flip problem is inherent to all isometric matching methods. In this chapter, we first describe the methods to cluster a given point cloud into its symmetric sides and then explain how to incorporate this side information into our correspondence estimation method in order to alleviate the symmetric flip problem.

4.1 Symmetry with Plane Fitting

The isometric cost calculated using diffusion distance as given in Eq. 3.3 fails to handle the symmetric flip problem, which is inherent to all isometric correspondence methods. Hence, while dealing with objects with intrinsic symmetries, we propose to use a modified version of this cost function so as to penalize the symmetrically flipped correspondences.

To this effect, we first find the symmetry plane on each point cloud by implementing the method of Lipman et al. [Lipman et al., 2010]. They measure the global intrinsic symmetry dissimilarity for each point using the eigenvectors (Φ_k) of the Laplacian matrix of the shape as the symmetry-invariants by:

$$Sym_{ij}^2 = \sum_{k=1}^{K}(|\phi_k(x_i)| - |\phi_k(x_j)|)^2, \tag{4.1}$$

where $x_i \in X$ and $x_j \in X$.

Then, they calculate the symmetry correspondence matrix by $C_S = e^{(-\frac{Sym_{ij}}{\sigma * diam})^2}$, where $diam = max_{ij}||x_i - x_j||$ and σ is a localization parameter experimentally set to 0.0001. Multiplying the symmetry correspondence matrix C_S with the points of the shape X provides us with orbit centroids X_S for the points. In other words, the entry $C_{S_{ij}}$ represents how much the points x_i and x_j belong to the same orbit. The symmetric points to x_i with high values in C_S play a more important role to find the centroid of the corresponding orbit. To find reflectional symmetry information for the point cloud, a plane can be fit to the orbit centroids X_S for symmetric shapes with reflectional intrinsic symmetry, such as human shapes.

We fit planes to our dataset with the partial and noisy point clouds. Next, we use the resulting two planes of two point clouds to identify the symmetric sides of the given shapes. We decide on the correct orientation of the symmetry planes of the models to differentiate the left and right hand sides of the bodies, assuming that they are both in frontal pose (or both facing backward). We incorporate this symmetry information into our cost calculation so as to penalize matching of the points that are located on non-corresponding sides of the two shapes, in proportion to their distances from the symmetry planes:

$$c'_{ij} = c_{ij} * (1 + \alpha * \frac{d_{sym}(s_i, t_j) - \min\limits_{k,l} d_{sym}(s_k, t_l)}{\max\limits_{k,l} d_{sym}(s_k, t_l)}) \tag{4.2}$$

where $d_{sym}(s_i, t_j)$ denotes the joint distance from points s_i and t_j to the corresponding symmetry planes, calculated by $d_{sym}(s_i, t_j) = \max(d^S_{sym}(s_i), d^T_{sym}(t_j))$, where $d^S_{sym}(s_i)$ and $d^T_{sym}(t_j)$ are the Euclidean distances to the corresponding symmetry planes. The penalization factor is $\alpha > 0$. Note that for $\alpha = 0$, the symmetry-aware cost c'_{ij} becomes identical to c_{ij} in Eq. 3.3.

Due to the partial and noisy nature of depth data, the estimated symmetry plane for a given shape may deviate from the true symmetry axis, which might in turn lead to wrong left/right hand side assignments, especially on the parts that are too close to the plane. Therefore, we penalize matching of points only if they are both far from the symmetry plane. A pair of points (s_i, t_j) is considered far enough if $d_{sym}(s_i, t_j) > \beta_{plane}$.

Figure 4.2 represents the symmetrically flipped correspondence on the right arm using the plane fitting strategy. To address such problematic cases, we may incorporate a fallback mechanism into our algorithm, that can automatically decide to continue with or without employing the symmetry extension.

The fallback procedure checks whether the fitted plane on a point cloud satisfactorily represents a reflectional symmetry. For this, we may follow [Lipman et al., 2010] and check the two largest eigenvalues of the average point set X_S used in the principal component analysis (associated with the two eigenvectors specifying the fitting plane). Only if the two largest eigenvalues are above a predefined threshold, the given shape is found to be sufficiently symmetric around a plane, otherwise we resort back to the original cost function given in Eq. 3.3.

While we first worked with an existing solution regarding symmetry, realizing its drawbacks for some problematic shapes, we generated a novel global intrinsic symmetry solution that works on noisy partial point clouds of intrinsic human shapes. We discuss the details of our new symmetry method in the next section.

4.2 Clustering-based Symmetry

4.2.1 Symmetry Detection

A shape P is said to be *intrinsically symmetric* with respect to a transformation $\mathcal{F} : P \rightarrow P$, if an intrinsic metric d is preserved by the transformation $\mathcal{F}$, i.e., $d_P(p, q) = d_P(\mathcal{F}(p), \mathcal{F}(q))$ for all $p, q \in P$ [Mitra et al., 2013]. Since we deal with noisy and incomplete point clouds, we use the diffusion distance already defined in Eq. 3.2 as our intrinsic metric for symmetry computations.

The first step in symmetry detection is to find this transformation $\mathcal{F}$, which provides us with the set of all symmetric point pairs in the point cloud. However since symmetries are never perfect on real shapes, we rather seek for an approximation of this transformation. Hence one can define it as the transformation that maximally preserves the diffusion distances between points, so as to minimize a global symmetry distortion function as given below:

$$\begin{aligned}\mathcal{F} = \arg\min_{\mathcal{G}} \sum_{p,q\in P} \max\{&|d_P(p,q) - d_P(\mathcal{G}(p),\mathcal{G}(q))|,\\ &|d_P(p,\mathcal{G}(q)) - d_P(\mathcal{G}(p),q)|\}.\end{aligned} \tag{4.3}$$

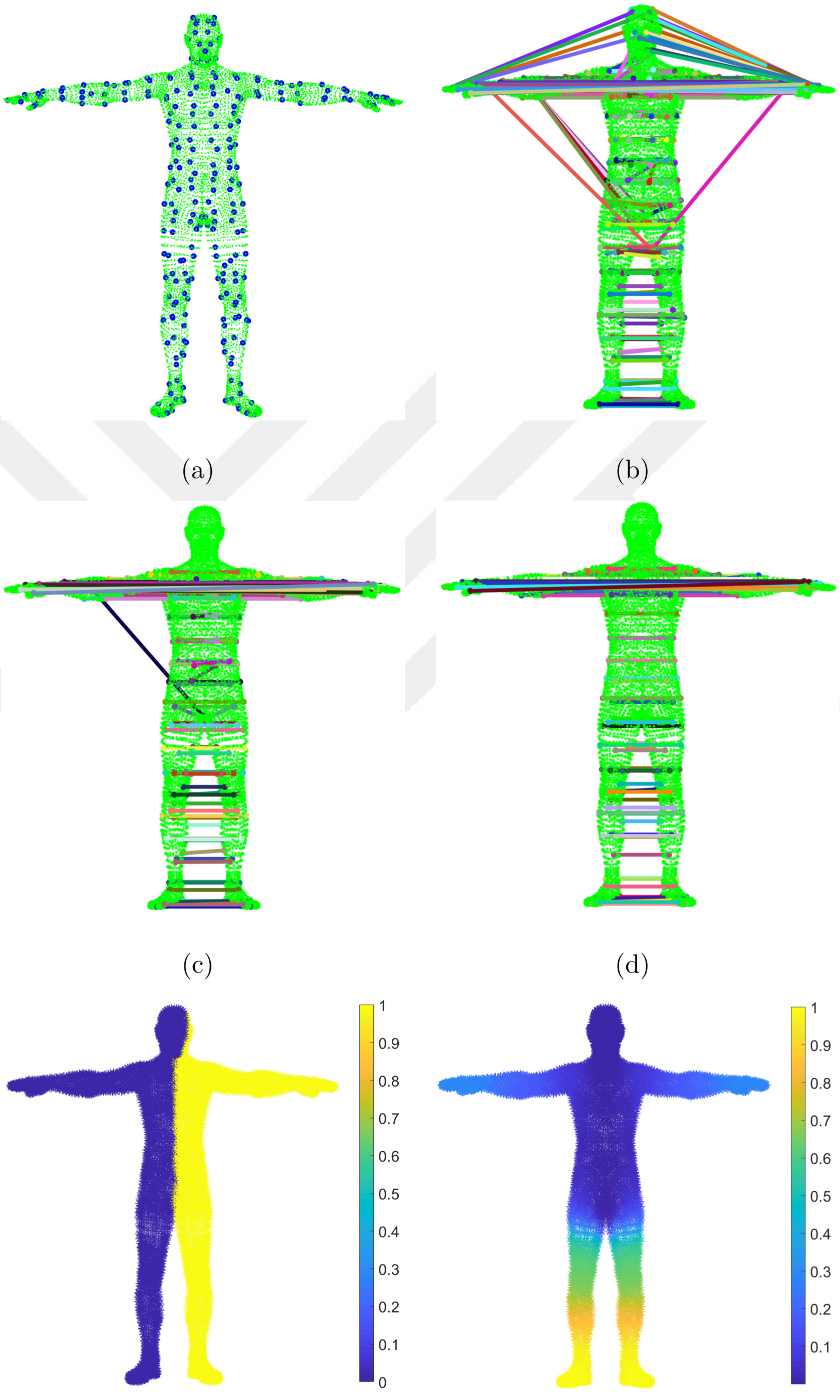
Note that the transformation $\mathcal{F}$ is partial in most cases, even if the object is globally symmetric as a whole, due to incompleteness of real acquisition data, such as in the case of depth sensors. Thus, to address this optimization problem, we resort to an iterative algorithm with a sub-optimal solution that aims to maintain the global symmetry distortion as low as possible while trying to find as many symmetric point pairs as possible. We explain this algorithm in the sequel.

In practice, for efficiency, we sample points $\bar{P} = \{\bar{p}_1, \ldots, \bar{p}_{|\bar{P}|}\}$ from the point cloud P and use this set to find the mapping $\mathcal{F} : \bar{P} \rightarrow \bar{P}$. For sampling, we use a voxel grid based downsampling and filtering method, as implemented in the Point Cloud Library (PCL)^1. We represent the results of each step in Figure 4.1, and Figure 4.1(a) shows the sampled points (blue ones) on the point cloud.

We first estimate an initial $\mathcal{F}$ based on the distribution of diffusion distances from each sampled point $\bar{p}_i \in \bar{P}$ to all other points $p \in P$. To this end, we compute the normalized diffusion distance histogram, $\boldsymbol{h}_i$, as descriptor for each sampled point $\bar{p}_i$ as in [Raviv et al., 2007, Raviv et al., 2010a, Raviv et al., 2010b]. We calculate the diffusion distances using the eigenvalues and eigenvectors of the Laplacian matrix of P, as discussed before. We then match each sampled point $\bar{p}_i$ with the point $\bar{p}_j$ that provides the minimum distance between histograms:

$$j \leftarrow \text{argmin}_{k\neq i}||\boldsymbol{h}_i - \boldsymbol{h}_k||_2. \tag{4.4}$$

The matching process explained above provides us with an initial *many-to-one* relation, i.e., a set of point pairs as presented in Figure 4.1(b). We first make this relation *one-to-one*. For this purpose, we calculate the symmetric distortion $E_{sym}(\bar{p}_i, \bar{q}_i)$ of



(a) (b) (c) (d) (e) (f)

Figure 4.1: a) Sampling of points from point cloud, b) Initial pairs, c) One-to-one pairs, d) Pruned symmetric pairs, e) Two symmetric segments of the point cloud, f) Representation of absolute votes.

each point pair $(\bar{p}_i, \bar{q}_i)$ as in [Jiang et al., 2013] (see also Eq. 4.3):

$$\begin{aligned} E_{sym}(\bar{p}_i, \bar{q}_i) =& \frac{1}{|\mathcal{F}| - 1} \sum_{(\bar{p}_j, \bar{q}_j) \in \mathcal{F} - (\bar{p}_i, \bar{q}_i)} \max\{|d_P(\bar{p}_i, \bar{p}_j) \\ & -d_P(\bar{q}_i, \bar{q}_j)|, |d_P(\bar{p}_i, \bar{q}_j) - d_P(\bar{p}_j, \bar{q}_i)|\}. \end{aligned} \tag{4.5}$$

Note that we use $\mathcal{F}$ to denote the symmetry transformation as well as the set of symmetric point pairs, hence the relation, induced by this transformation. In order to make the relation $\mathcal{F}$ one-to-one, we first pick the pair $(\bar{p}_*, \bar{q}_*)$ with minimum symmetric distortion, and eliminate each $(\bar{p}_*, \cdot)$, $(\cdot, \bar{p}_*)$, $(\bar{q}_*, \cdot)$ and $(\cdot, \bar{q}_*)$ except $(\bar{p}_*, \bar{q}_*)$ from $\mathcal{F}$. Then, we pick from the remaining the pair with the next minimum distortion, and likewise eliminate all other pairs including any of those points. We repeatedly perform this process until $\mathcal{F}$ becomes one-to-one.

The resulting one-to-one mapping may still include pairs with large distortion as in Figure 4.1(c). Therefore, we prune the set using the same iterative procedure that we employed for base correspondence estimation (see Section 3.3.2). At each iteration, we first re-calculate E_{sym} values of the pairs in the set (since it depends on $\mathcal{F}$ that is modified through iterations). Then, we eliminate the worst point pair from the set according to symmetric distortion. We continue re-calculating symmetric distortion and eliminating the worst pair until the minimum and the maximum distortions are close to each other within a given ratio τ_{sym}. The whole process of minimization of symmetry distortion outputs a mapping $\mathcal{F}$ that is one-to-one, but mostly partial, due to pruning (Figure 4.1(d)).

4.2.2 Clustering into Sides

We use the obtained symmetry set $\mathcal{F}$ to cluster the point cloud P into symmetric sides, P_- and P_+. Each symmetric point pair $(\bar{p}_i, \bar{q}_i)$ votes for every point $p_j \in P$, in order to determine on which side of the shape p_j is. After the voting process, the points with negative-valued accumulation of votes are assigned to side P_- whereas the others to side P_+. The votes of the pairs take value depending on what we call a distance weight vector $\boldsymbol{w}(\bar{p}_i) = (w_1(\bar{p}_i), w_2(\bar{p}_i), ..., w_{|P|}(\bar{p}_i))$, which is calculated for each sampled point $\bar{p}_i$ by $w_j(\bar{p}_i) = e^{-d_P(\bar{p}_i, p_j)}$. Hence a distance weight vector $\boldsymbol{w}(\bar{p}_i)$ assigns higher weights to the points closer to $\bar{p}_i$ and lower weights to the points further from $\bar{p}_i$ in the point cloud P.

We start the voting process with the best pair $(\bar{p}_*, \bar{q}_*)$ having the least symmetric distortion E_{sym}. We define a vote vector $\boldsymbol{v} = (v_1, v_2, ..., v_{|P|})$ and initialize it by $\boldsymbol{v} = \boldsymbol{w}(\bar{p}_*) - \boldsymbol{w}(\bar{q}_*)$, where each v_k represents the amount of votes for p_k. Thus initially, the points close to $\bar{p}_*$ take higher absolute votes with positive sign, whereas the points close to $\bar{q}_*$ take higher absolute votes with negative sign.

We then proceed with the next pair which is the closest point pair to $(\bar{p}_*, \bar{q}_*)$, say $(\bar{p}_i, \bar{q}_i)$. We note that proceeding with the closest pair improves the stability of the voting process, and the distances between point pairs are calculated as in [Xu et al., 2012] by:

$$d_P(\{\bar{p}_k, \bar{q}_k\}, \{\bar{p}_l, \bar{q}_l\}) = min\{\frac{d_P(\bar{p}_k, \bar{p}_l) + d_P(\bar{q}_k, \bar{q}_l)}{2}, \frac{d_P(\bar{p}_k, \bar{q}_l) + d_P(\bar{q}_k, \bar{p}_l)}{2}\} \quad (4.6)$$

For $(\bar{p}_i, \bar{q}_i)$, we first need to decide the sides of the points of the pair, $\bar{p}_i$ and $\bar{q}_i$, based on the current votes accumulated in $\boldsymbol{v}$. For this, we compare the two values calculated by $\sum_{p_k \in P} w_k(\bar{p}_i)v_k$ and $\sum_{p_k \in P} w_k(\bar{q}_i)v_k$, respectively for $\bar{p}_i$ and $\bar{q}_i$. The point yielding higher value, say $\bar{p}_i$, is assigned to side P_+ with $s(\bar{p}_i) = 1$, and the other to side P_- with $s(\bar{q}_i) = -1$, where s is an indicator function taking value either 1 or -1. Then, the pair $(\bar{p}_i, \bar{q}_i)$ votes for all the points in P by incrementing the vote vector by $\boldsymbol{v} = \boldsymbol{v} + s(\bar{p}_i)(\boldsymbol{w}(\bar{p}_i) - \boldsymbol{w}(\bar{q}_i))$. We repeat this process until no more point pair is left for voting. Finally we normalize the values in vote vector $\boldsymbol{v}$ by the maximum absolute vote in the vector. We refer to the resulting vote values as *symmetry weights*, the signs of which provide us with two symmetric sides as displayed in Figure 4.1(e). We will denote the symmetry weight vector by $\tilde{\boldsymbol{v}}$ and the symmetry weight for an individual point p_i by $\tilde{v}(p_i)$. Moreover, the absolute values of these weights indicate confidence values for the side assignments of the points in P (Figure 4.1(f)). The pseudocode of our method for clustering into symmetric sides is given in Algorithm 3.

Algorithm 3 Clustering into Symmetric Sides

```
Input: Point cloud $P$
  Error threshold coefficient $\tau_{sym}$
  Eigenvalues $\lambda$ and eigenvectors $\phi$ of Laplacian matrix $L$ of $P$
Output: Symmetry weight vector $\tilde{v}$ of $P$
  Sample points from $P$ s.t. $\bar{P} = \{\bar{p}_1, \ldots, \bar{p}_{|\bar{P}|}\} \subset P$
  Initial matching of symmetric points
  Calculate the descriptor $h_i \in H$ for each sampled point $\bar{p}_i$ using the distribution of normalized diffusion distances;
  initialize the set of symmetric pairs $\mathcal{F} \leftarrow \emptyset$
  For each $\bar{p}_i \in \bar{P}$,
    $j \leftarrow \text{argmin}_{k \neq i} ||h_i - h_k||_2$;
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{\bar{p}_i, \bar{p}_j\}$
  Eliminate unreliable pairs
  Make many-to-one set $\mathcal{F}$ one-to-one;
  repeat
    Calculate the symmetry distortion $E_{sym}(\bar{p}_i, \bar{q}_i)$ for each symmetric pair $(\bar{p}_i, \bar{q}_i) \in \mathcal{F}$ via Eq. 4.5;
    $\{\bar{p}_{\max}, \bar{q}_{\max}\} \leftarrow \text{argmax}\ E_{\text{sym}}(\bar{p}_k, \bar{q}_k)$;
    $\mathcal{F} \leftarrow \mathcal{F} - \{\bar{p}_{\max}, \bar{q}_{\max}\}$;
  until $\max\limits_{(\bar{p}_k, \bar{q}_k)} E_{sym}(\bar{p}_k, \bar{q}_k) < \tau_{sym} * \min\limits_{(\bar{p}_k, \bar{q}_k)} E_{sym}(\bar{p}_k, \bar{q}_k)$;
  Voting
  Sort the pairs based on the distance to the pair with the lowest symmetry distortion;
  Find the distance weights for each $\bar{p}_i$ and $p_j$ by $w_j(\bar{p}_i) \leftarrow e^{-d_P(\bar{p}_i, p_j)}$;
  Initialize the voting vector by $\boldsymbol{v} \leftarrow \boldsymbol{w}(\bar{p}_*) - \boldsymbol{w}(\bar{q}_*))$;
  For each $(\bar{p}_i, \bar{q}_i) \in \mathcal{F} - (\bar{p}_*, \bar{q}_*)$,
    $s(\bar{p}_i) \leftarrow sign(\sum_{p_k \in P} w_k(\bar{p}_i) v_k - \sum_{p_k \in P} w_k(\bar{q}_i) v_k)$ ;
    $\boldsymbol{v} \leftarrow \boldsymbol{v} + s(\bar{p}_i)(\boldsymbol{w}(\bar{p}_i) - \boldsymbol{w}(\bar{q}_i))$;
  $\tilde{\boldsymbol{v}} \leftarrow \boldsymbol{v} / \max(|\boldsymbol{v}|)$;
```

4.2.3 Symmetry-aware Matching

We now describe how we incorporate the symmetry weights computed previously into our base correspondence estimation algorithm in order to alleviate the symmetric flip problem. Recall that the symmetry weight calculated for a given point is used to assign it to a side (P_+ or P_-) on the point cloud based on sign information, and also specifies a confidence for that assignment.

Given two shapes (source and target) to be matched, the first step is then to match the sides of the corresponding point clouds P^S and P^T. For example, if the shapes are frontal human models, then the left-hand side of the source (P^S_+ or P^S_-) is to be matched with the left-hand side of the target. For this purpose, we first register the two point clouds using the well-known rigid ICP method [Rusinkiewicz and Levoy, 2001] so as to make the corresponding sides as close to each other as possible. Then, we check whether the signs of the sides of the registered point clouds are matching. Let the symmetry weights of source P^S and target P^T be $\tilde{\boldsymbol{v}}^T$ and $\tilde{\boldsymbol{v}}^S$, respectively. We define a side matching score $\kappa_{T\rightarrow S}$ that decides on the corresponding sides of the point clouds by:

$$\begin{aligned}\kappa_{T\rightarrow S} = &\sum_{p_i \in P^S} |\tilde{v}^S(p_i) + \tilde{v}^T(q_i)| \\ &- \sum_{p_i \in P^S} |-\tilde{v}^S(p_i) + \tilde{v}^T(q_i)|.\end{aligned} \tag{4.7}$$

where $q_i = \text{argmin}_{q_j \in P^T} ||p_i - q_j||$ is obtained via registration. Then we update the sign of the weights of P^S by $\tilde{\boldsymbol{v}}^S = \text{sign}(\kappa_{S\rightarrow T})\tilde{\boldsymbol{v}}^S$. In practice, we register the point clouds both ways. We also calculate $\kappa_{T\rightarrow S}$, and update the signs of the weights based on the larger of the side matching scores $\kappa_{S\rightarrow T}$ and $\kappa_{T\rightarrow S}$.

Once the sides are matched as explained above, we next modify the cost function in Eq. 3.3 based on the symmetry weights of the point clouds so as to penalize matching of the points that are located on non-corresponding sides of the shapes. So whenever a source point $s_i \in P^S$ and a target point $t_j \in P^T$ are found to be located on the non-corresponding sides, the following modified cost function c'_{ij} becomes in use:

$$c'_{ij} = c_{ij}(1 + \frac{\alpha}{2}(|\tilde{v}^S(s_i)| + |\tilde{v}^T(t_j)|)) \tag{4.8}$$

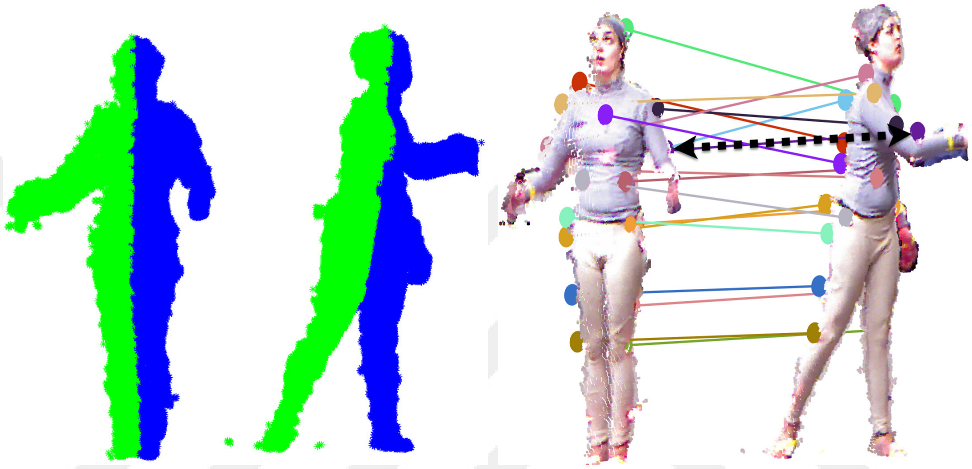
The penalization factor $\alpha > 0$ is set as $\alpha = 1.5$ in all our experiments. Note that for $\alpha = 0$, the symmetry-aware cost c'_{ij} becomes identical to c_{ij} in Eq. 3.3.

Due to partial and noisy nature of depth data, the estimated symmetry information of a given shape may deviate from the true symmetry on some body parts, especially on the parts that have topological misconnections or missing parts. We observe that the points at those parts with ambiguities usually have low weights. Therefore, we penalize matching of points only if they both have high symmetry weights, i.e., the parameter α in Eq. 3.3 is reset to zero whenever $\min(|\tilde{v}^S(s_i)|, |\tilde{v}^T(t_j))| \leq \beta_{sym}$, where β_{sym} is set experimentally to 0.01.

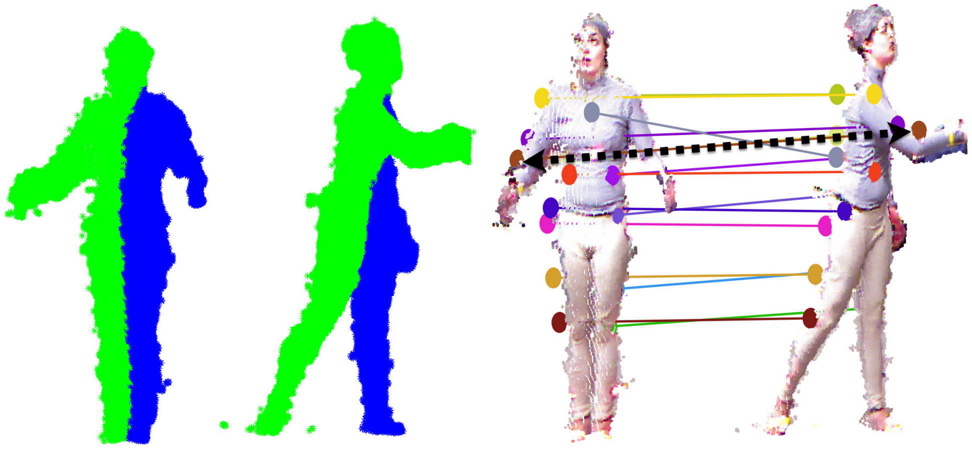
We note that, as an alternative, the method of Lipman et al. [Lipman et al., 2010] can also be used to find symmetric sides by fitting a symmetry plane onto the point cloud. However, isometric deformations, such as those in human shapes, cannot always be represented by planar symmetries as illustrated in Figure 4.2(a) with a failure case. In this example, the fitted symmetry plane fails to differentiate the right arm of the target model from the left arm (mainly due to the twisted movement of the right arm). Hence, when incorporated into correspondence estimation as in [Küpçü and Yemez, 2017], it yields incorrect correspondences as also shown in Figure 4.2(a), e.g., the symmetrically flipped correspondence on the right arm. In this respect, our clustering-based method is more general, and can better address such problematic cases, as observed in Figure 4.2(b).

4.3 Complexity

For efficiency, we sample point set $\bar{P} \subset P$ where points spread at equal distances. The dominant complexity of matching and pruning process of our clustering-based symmetry technique is $O(|\bar{P}|^2)$. The complexity of the voting part is $O(|\bar{P}||P|)$. Thus, overall, our clustering-based symmetry calculation costs $O(|\bar{P}||P| + |\bar{P}|^2)$. The size of



(a) Using plane-based symmetry



(b) Using our clustering-based symmetry

Figure 4.2: Symmetry-aware base correspondence estimation: a) Finding symmetric sides via plane fitting and matching, b) Clustering into symmetric sides with our method and matching.

the sampled set is $|\bar{P}| \sim O(\sqrt{|P|})$. Therefore, the overall complexity of our technique is $O(|P|^{\frac{3}{2}})$.

Chapter 5

EXPERIMENTAL PRELIMINARIES

5.1 Datasets

Microsoft Kinect is one of the recent depth camera technologies, which provides us with both depth and color information of each pixel in a scene. But the accuracy in the frames taken from a depth camera is worse when it is compared to the expensive technologies such as multiple camera setups and laser scanners. Figure 5.1 shows an example noisy partial depth frame with topological connections seen from different angles.

Figure 5.1: An example challenging depth frame from the MHAD dataset.

We evaluate the performance of our algorithm on three depth datasets: The Berkeley motion human action dataset (MHAD) [Ofli et al., 2013, Teleimmersion Lab,], the dataset that we collected (Human) and the dataset of Guo et al. [Guo et al., 2015], all containing noisy and incomplete depth data of freely moving subjects, captured using Kinect v1. We convert all depth frames to 3D point cloud representations and discard color.

5.1.1 Existing Datasets Employed

We pick 12 depth frames from the MHAD dataset and generate their 66 pair combinations. We select frames which exhibit large non-rigid deformations with respect to each other, and each from a different action set, such as jumping and throwing (see Figure 5.2(a)). Each frame in this dataset has approximately 24K points and 43 ground-truth marker positions.

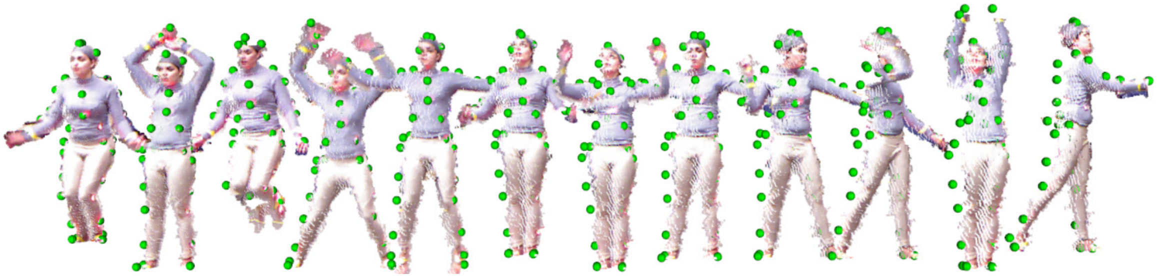
We also use the Guo et al. [Guo et al., 2015] dataset (see Figure 5.2(c)) to compare our dense correspondence results with the results of [Guo et al., 2015]. This dataset includes 9 frames with approximately 29K points on each. We manually selected 14 ground-truth keypoints on each frame.

5.1.2 New Human Dataset

The other dataset (Human) that we have collected contains RGB-D frames of a human subject exhibiting larger non-rigid motion compared to MHAD. The Human dataset, displayed in Figure 5.2(b), includes 6 frames with 33 manually selected ground-truth keypoints out of approximately 24K points on each, and 15 model pairs.

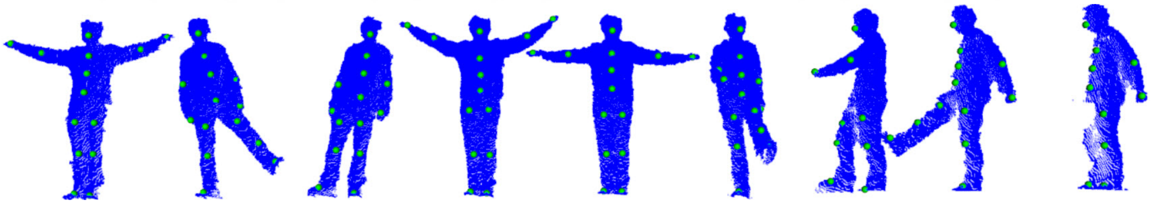
5.2 Baseline Methods

The baseline methods to which we compare our base correspondence estimation algorithm are the SHOT descriptor matching (using a publicly available implementation[1]), and two state of the art non-rigid point registration methods with publicly available codes: the CPD method of Myronenko and Song [Myronenko and Song, 2010], which is based on global registration of point clouds using Gaussian mixture models, and the PR-GLS method of Ma et al. [Ma et al., 2016], which incorporate local



(a) MHAD

(b) Human



(c) Guo et al. [Guo et al., 2015]

Figure 5.2: Depth datasets with ground-truth keypoints displayed in green.

features to take into account similarity of neighboring structure of points in addition to CPD. Both methods also provide point correspondences. The descriptor matching algorithm is based on the Euclidean distance between the SHOT descriptors [Tombari et al., 2010] in two directions: from target to source and from source to target. The intersection of the resulting correspondence sets is the final output and is referred to as *reciprocal correspondences* [Pajdla and Van Gool, 1995].

We compare our dense correspondence results with the results of three state of the art methods. Besides CPD and PR-GLS, we additionally compare our method with the method of Guo et al. [Guo et al., 2015], which is a recent point-based correspondence technique relying on geodesic distances. They perform piecewise rigid point registration by discovering new parts iteratively. Therefore, they rely on the existence of the rigid parts. To experiment with this method, we use the code provided by the authors.

5.3 Evaluation Strategy

We employ two evaluation metrics: 1) deviation from isometry, and 2) deviation from ground-truth. The deviation from isometry, i.e., the isometric error, for a given correspondence pair is computed via E_{iso} measure given in Eq. 3.4. The deviation from ground-truth, i.e., the ground-truth error, is calculated based on each pair's closest ground-truth correspondence. The ground-truth error, denoted by E_{grd}, is computed for each correspondence pair (b_i^S, b_i^T) by

$$E_{\text{grd}}(b_i^S, b_i^T) = \min(|d_S(b_i^S, g_l^S) - d_T(b_i^T, g_l^T)|, |d_S(b_i^S, g_k^S) - d_T(b_i^T, g_k^T)|) \tag{5.1}$$

where g denotes a ground-truth keypoint on the source or target, $l = \text{argmin}_m(d_S(b_i^S, g_m^S))$ and $k = \text{argmin}_m(d_T(b_i^T, g_m^T))$. Each of these measures is averaged over all the pairs of a given correspondence set and then eventually over the whole dataset. We note that Eq. 5.1 is used for ground-truth error calculation of both sparse and dense correspondence results.

To make a fair comparison between our method and the baseline methods, we equalize the number of matchings for each model pair, by selecting the best R correspondences resulting from each method, where R is the size of the correspondence set with the least number of pairs.

Chapter 6

RESULTS

6.1 Implementation Details

We normalize the coordinates of each point cloud in a given dataset such that all the points lie within the unit sphere centered at the origin. For computation of the Laplacian matrices, we select up to $K = 400$ neighboring points.For diffusion distance, we use the smallest $M = 60$ eigenvalues and the corresponding eigenvectors of the Laplacian matrix. The time step parameter t is incremented from 1 to 600 to compute the average diffusion distance. The error threshold coefficients for base and dense correspondence algorithms are set experimentally as $\tau_1 = 2.1$ (see Algorithm 1) and $\tau_2 = 1.2$ (see Algorithm 2). The number of samples generated in each patch for dense correspondence estimation is fixed to $N_S = 50$. While finding the symmetric point correspondence of a point p via histogram matching in our global intrinsic symmetry method, we ignore the closest 10% of the sampled points in the candidate set. We experimentally set the symmetry error threshold coefficient as $\tau_{sym} = 2.4$. When using our symmetry detection method, for cost calculation, we set the penalization factor as $\alpha = 1.5$, and to decide on which points to penalize, we set the minimum symmetry weight threshold as $\beta_{sym} = 0.01$ in all our experiments. while matching with the plane fitting strategy, we also set the minimum distance threshold as $\beta_{plane} = 0.01$.

6.2 Intrinsic Symmetry Results

Our clustering-based symmetry method outputs a symmetry weight with a positive or negative sign for each point of a given point cloud. We provide three different representations of the resulting symmetry weights for the analyses. The first representation is the two symmetric sides of a point cloud (left and right hand side of a human model), which is segmented using the signs of the weights. Figure 6.1 and Figure 6.2 show some examples of the segmented symmetric sides for the Human and MHAD datasets, respectively. Our method successfully provides two approximate segments of the noisy partial point clouds of moving human models under occlusions and large topological changes caused by large deformations.

The second representation we provide shows the absolute values of symmetry weights of a point cloud. The symmetry weights are in the range [0,1]. High symmetry weight means that the symmetry side decision is strong for that part. From the examples of absolute weight distribution in Figure 6.3 and Figure 6.4, we observe that the symmetry side decisions on the arms and the legs are stronger than the decision on the torso, in general. We also see that the parts presenting occlusion with missing parts and topological changes have unstable symmetry information, such as in Figure 6.4. One of the reasons of this problem is finding insufficient symmetry pairs on those areas. Another reason is that when the points of a symmetry pair are too close to each other, the votes of this pair to other points calculated in Algorithm 3 is small (they cancel out each other), which can easily occur on the topologically connected legs in the bottom two example results in Figure 6.4.

The third representation presents the penalized and unpenalized areas based on symmetry weights compared to the minimum weight threshold β_{sym}. Figure 6.5 and Figure 6.6 show examples from the Human and MHAD datasets, respectively, where our algorithm only penalizes the keypoints on the yellow areas, since the blue area denotes uncertain symmetry weight assignment.

Further visual examples of segmented sides and associated symmetry weights from both datasets can be found in Figures 6.7 to 6.9.

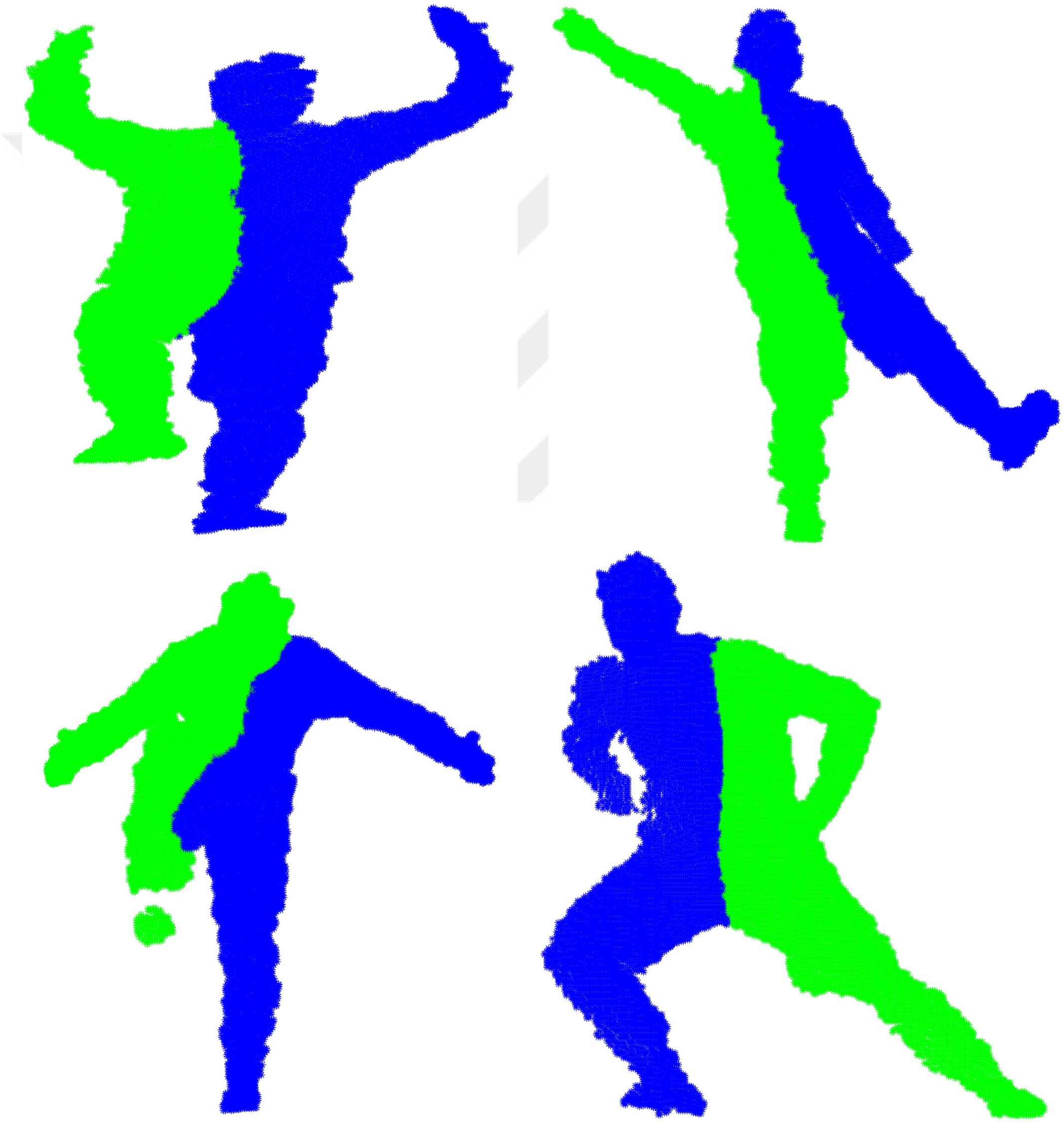


Figure 6.1: Two symmetric sides of the point clouds of the Human dataset.

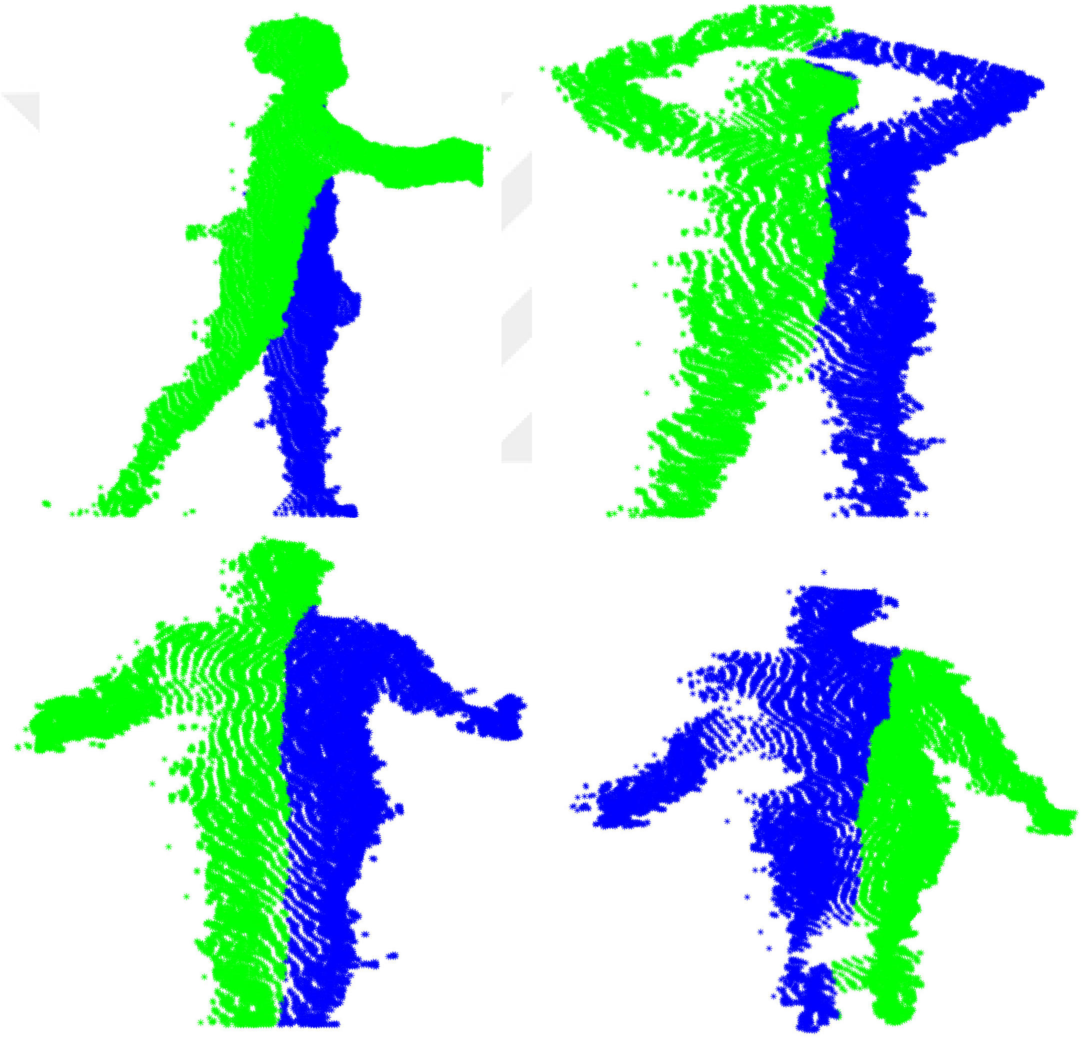


Figure 6.2: Two symmetric sides of the point clouds of the MHAD dataset.

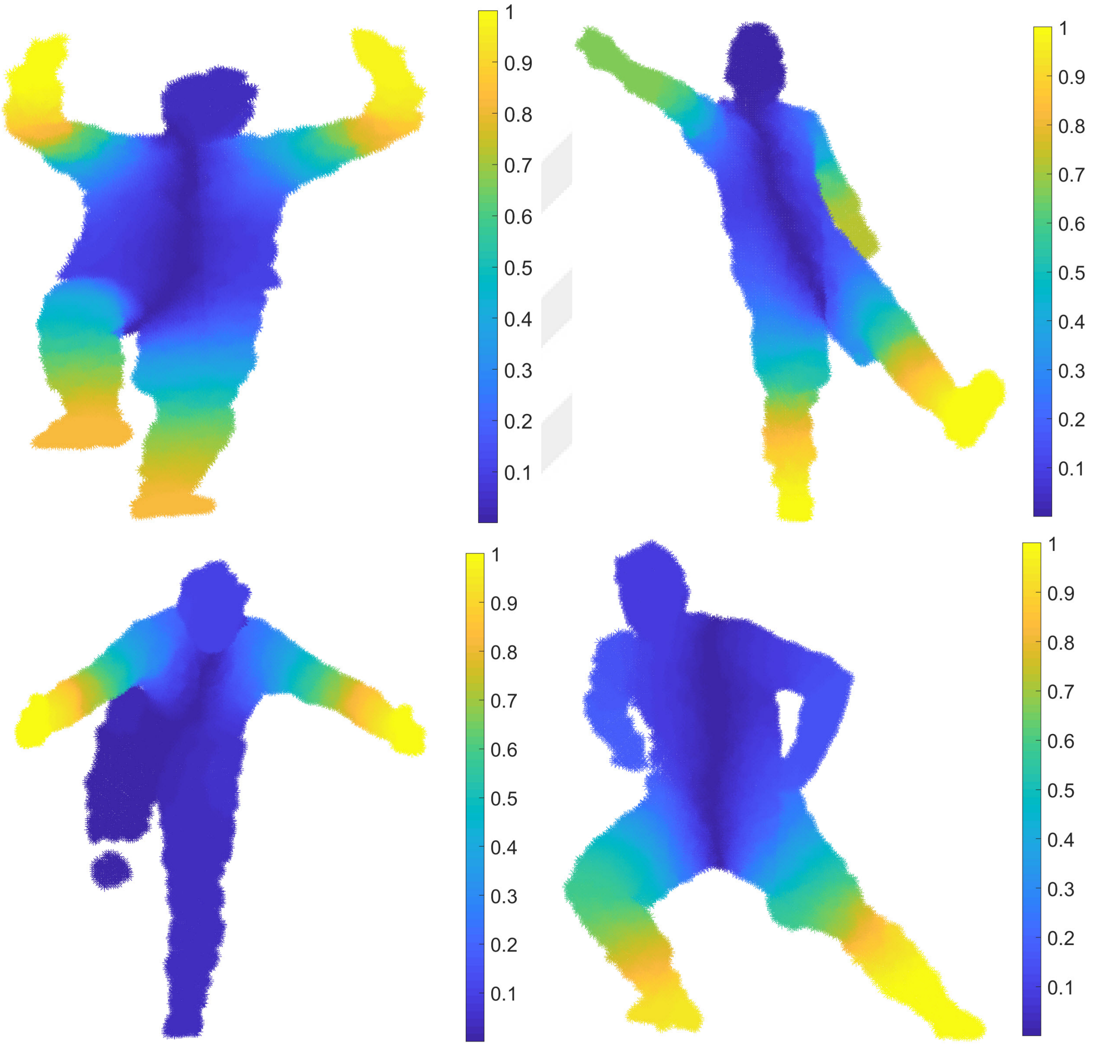


Figure 6.3: The absolute values of the symmetry weights of the example point clouds of the Human dataset.

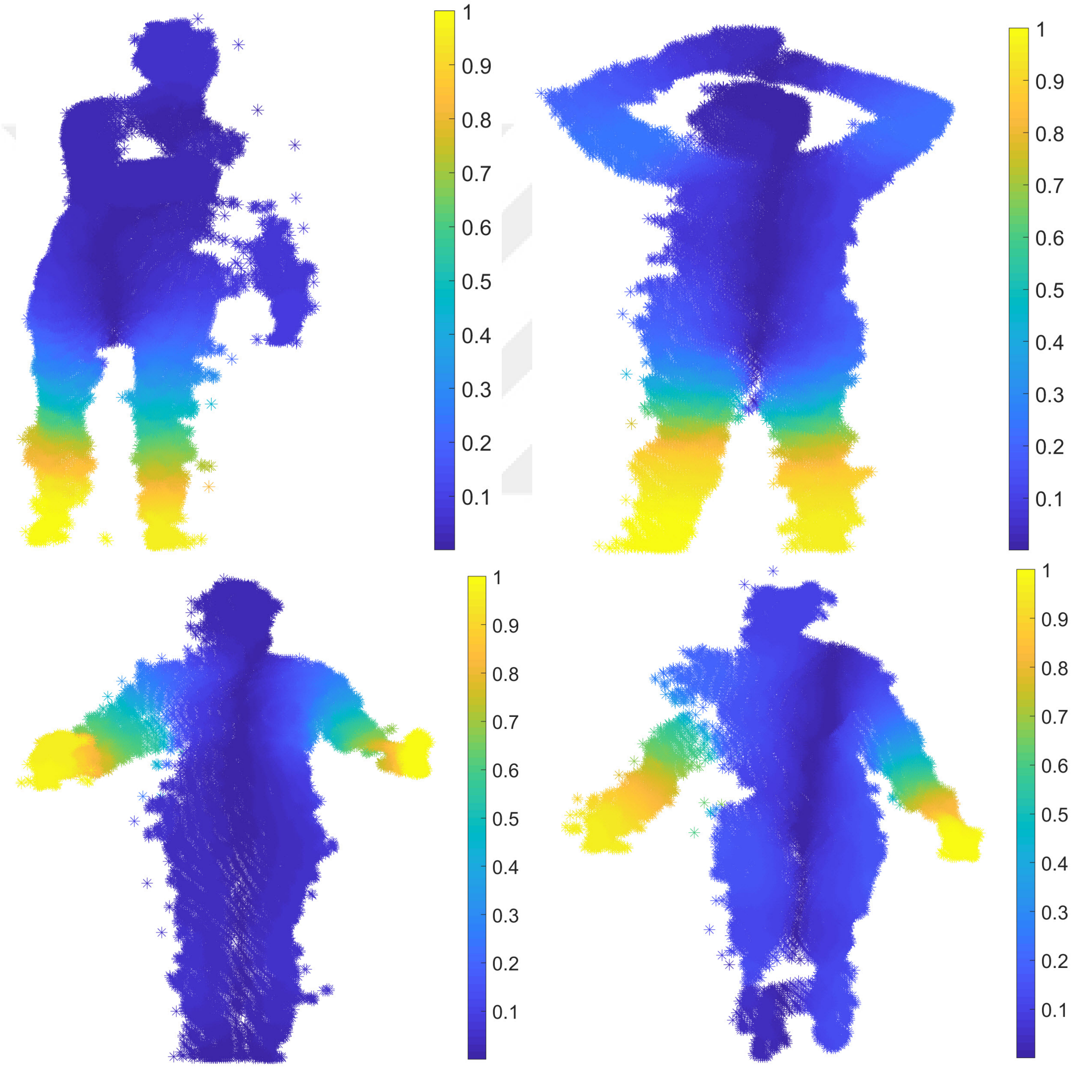


Figure 6.4: The absolute values of the symmetry weights of the example point clouds of the MHAD dataset.

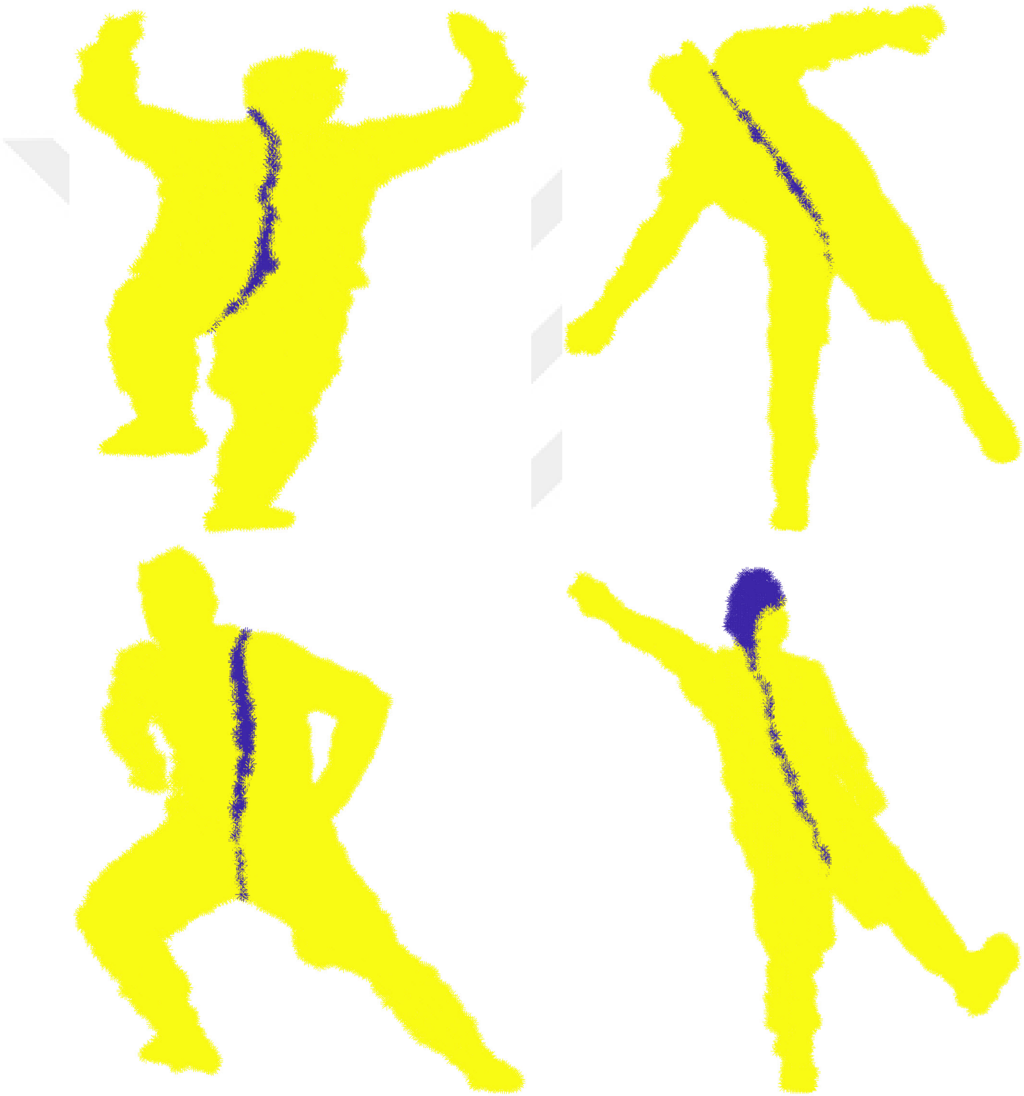


Figure 6.5: Penalized (yellow) and unpenalized (blue) area examples from the Human dataset.

Figure 6.6: Penalized (yellow) and unpenalized (blue) area examples from the MHAD dataset.

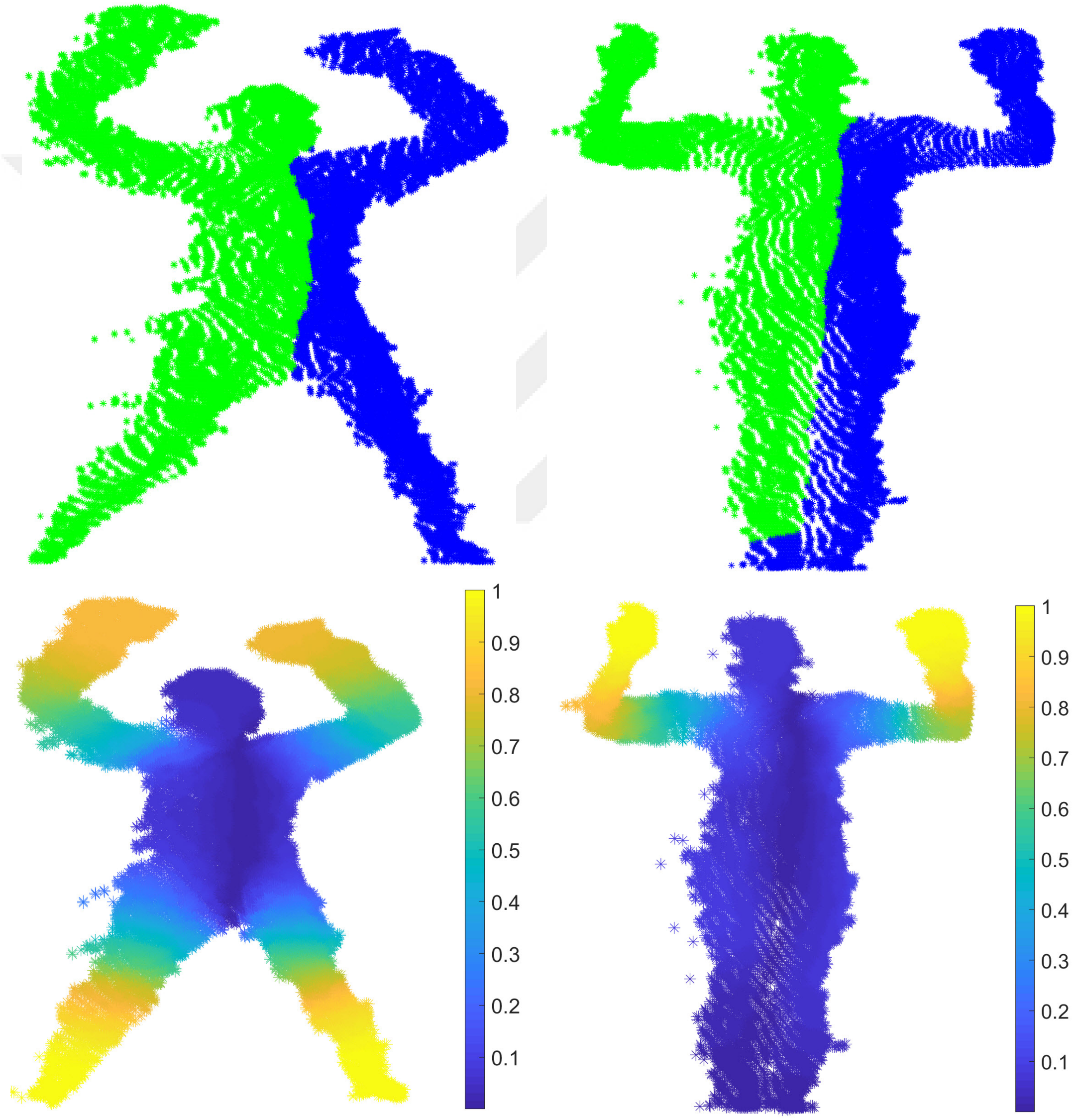


Figure 6.7: Further examples showing two symmetric sides and symmetry weights of the point clouds.

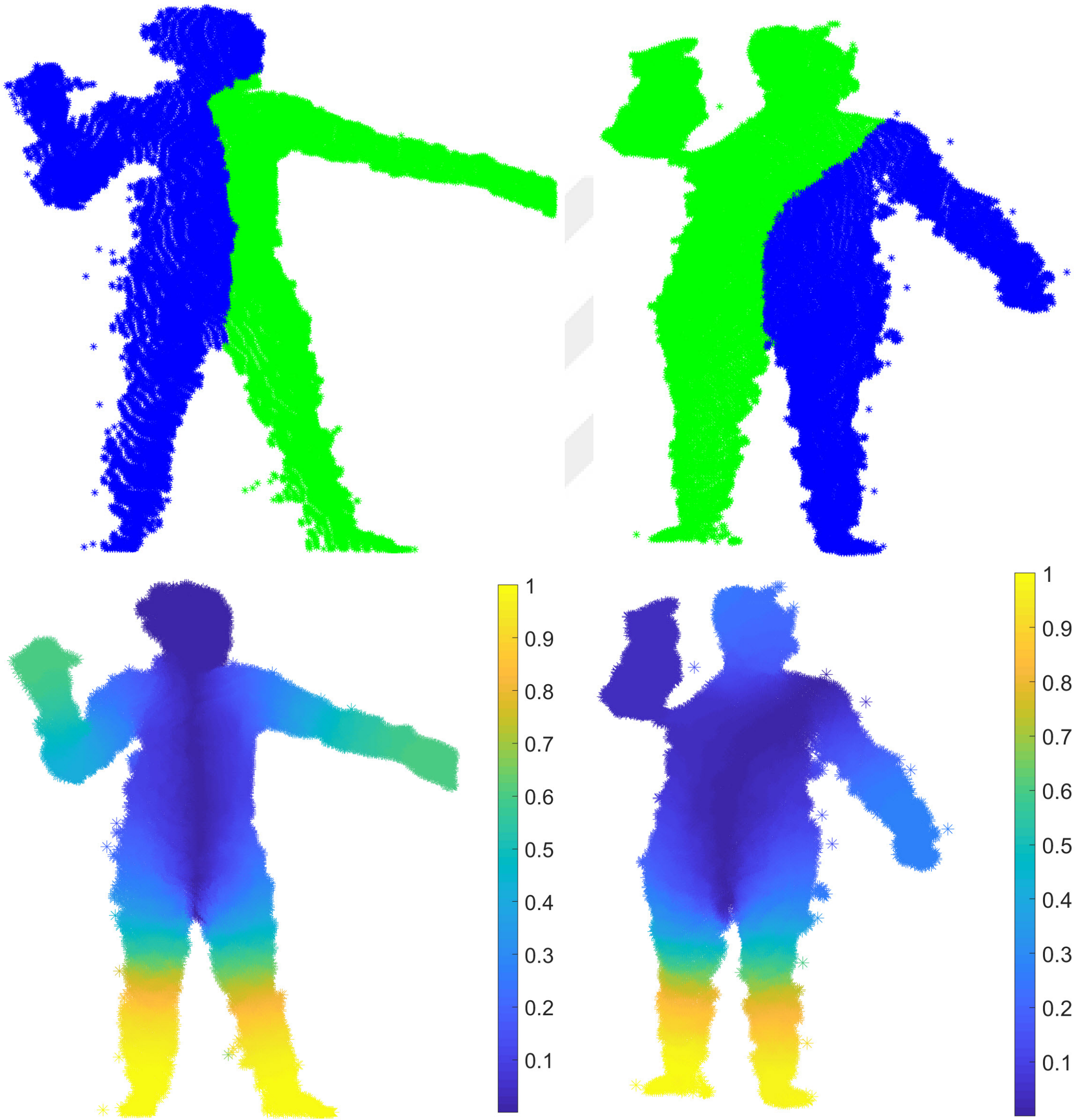


Figure 6.8: Further examples showing two symmetric sides and symmetry weights of the point clouds.

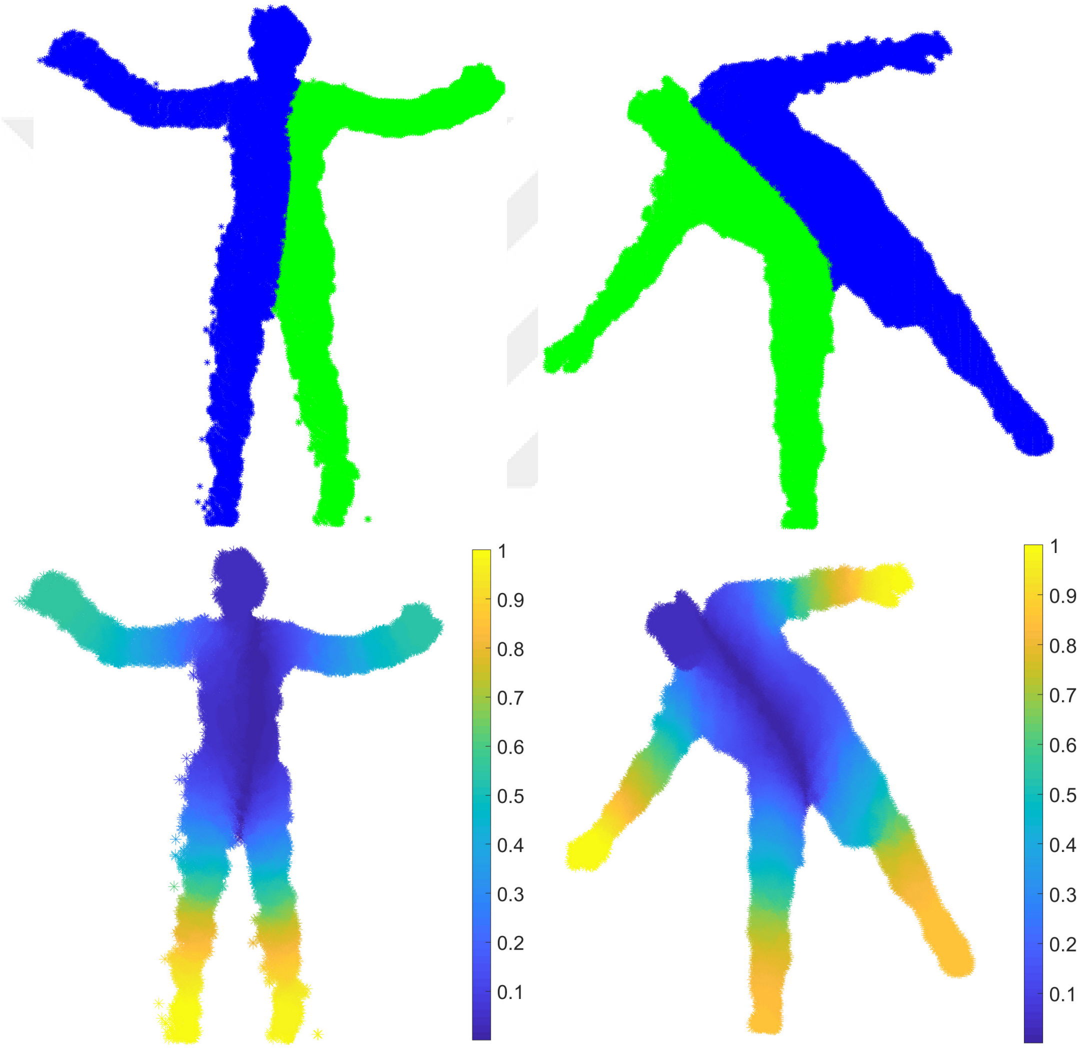


Figure 6.9: Further examples showing two symmetric sides and symmetry weights of the point clouds.

6.2.1 Comparison to Planar Symmetry

We first compared our sparse correspondence estimation technique employing the plane fitting symmetry extension proposed in [Küpçü and Yemez, 2017] against the baseline correspondence estimation methods. We use CPD algorithm [Myronenko and Song, 2010] to find the initial matchings in the experiments. In Table 6.1, we observe that our solution together with the plane fitting symmetry extension significantly outperforms the baseline algorithms in terms of isometric and ground truth errors. This is especially pronounced in the Human dataset, which contains very large deformations, mainly because local similarities in this case quickly drop due to occlusions resulting from large motion, whereas the MHAD dataset exhibits more rigidity between model pairs. Therefore, the baseline methods have higher success on MHAD compared to the Human dataset.

We test the performance of our base correspondence algorithm also with automatically detected keypoints. For keypoint detection, we use Intrinsic Shape Signatures (ISS) [Zhong, 2009], though any keypoint detection algorithm could be employed for this purpose. The average number of base correspondences found using this automatic keypoint detection algorithm is 17 for the Human and 22 for MHAD dataset. Our algorithm still provides more reliable matching results compared to baseline methods in terms of both error types.

Table 6.1: Quantitative evaluation of our **sparse** (base) correspondence estimation method using **plane fitting symmetry** extension in comparison to the baseline methods. (GT: Ground-truth, Errors $\times 10^{-5}$)

	With Ground Truth Keypoints			
Method	**GT Error**		**Isometric Error**	
	MHAD	Human	MHAD	Human
Descriptor Matching (DM)	3.85	7.09	7.73	6.33
CPD	0.49	4.79	5.15	6.54
PR-GLS	0.97	3.46	5.47	3.63
Our method (with CPD and Plane Fitting)	**0.31**	**0.08**	**0.60**	**0.62**
	With Detected Keypoints			
	GT Error		**Isometric Error**	
	MHAD	Human	MHAD	Human
Descriptor Matching (DM)	5.53	6.74	5.24	6.52
CPD	2.52	3.39	4.06	6.01
PR-GLS	3.01	4.27	3.52	5.60
Our method (with CPD and Plane Fitting)	**1.21**	**1.17**	**0.76**	**0.91**

6.3 Sparse Correspondence Results

In Table 6.2, we provide the results of the three baseline methods compared to our method, which employs our clustering-based symmetry method. We initialize our algorithm alternatively with descriptor matching, CPD, PR-GLS and with random correspondences. We observe that our algorithm considerably improves the ground-truth and isometric error results of the baseline algorithms (using their output as our initial base correspondences) on Human and MHAD datasets. The advantage of our method is pronounced especially when working with the challenging Human dataset that contains larger deformations, hence less local similarity and smaller surface overlap due to severe occlusions. We refer to Figure 3.3 for visual interpretation of the error values presented in the table. We also recall that, while evaluating the correspondence obtained on a shape pair, we take into account only the best R matchings resulting from each method, and discard the rest. This R value comes out to be on average 8 for the Human dataset and 13 for MHAD in our experiments.

To test our method under more realistic situations, we also experimented with automatically detected keypoints, employing the ISS (Intrinsic Shape Signature) method [Zhong, 2009], instead of directly using the available ground-truth keypoints (see Table 6.2). The performances of all methods deteriorate in this case as expected, mainly due to possible inconsistencies of the detected keypoints on source and target models. Yet our algorithm still significantly boosts the performances of all baseline methods in terms of all error types and datasets. More interestingly, even when randomly initialized, our algorithm provides better results than the baseline methods in almost all cases, and its performance is only slightly effected by the choice of the initialization technique. For the rest of our experiments, to compute the initial base correspondence input of our algorithm, we use the CPD method which is both performant and computationally efficient.

In Table 6.3, we evaluate the performance of our method, using our intrinsic symmetry methods described in Section 4 versus the case where it is not used. To analyze how much of the error is caused by symmetric flip issues, we define another ground-truth error measure $\tilde{E}_{\text{grd}}$ for each correspondence pair (b_i^S, b_i^T), which discards symmetric flip errors:

$$\begin{aligned}\tilde{E}_{\text{grd}}(b_i^S, b_i^T) = \min(&|d_S(b_i^S, g_l^S) - d_T(b_i^T, g_l^T)|,\\ &|d_S(b_i^S, g_k^S) - d_T(b_i^T, g_k^T)|,\\ &|d_S(b_i^S, g_l^S) - d_T(b_i^T, \tilde{g}_l^T)|,\\ &|d_S(b_i^S, \tilde{g}_k^S) - d_T(b_i^T, g_k^T)|)\end{aligned} \tag{6.1}$$

where $l = \text{argmin}_m(d_S(b_i^S, g_m^S))$, $k = \text{argmin}_m(d_T(b_i^T, g_m^T))$, and $\tilde{g}_l^T$ and $\tilde{g}_k^S$ are the symmetrically flipped versions of g_l^S and g_k^T, respectively. To compute this measure,

Table 6.2: Quantitative evaluation of our **sparse** (base) correspondence estimation method in comparison to the baseline methods. (GT: Ground-truth, Errors $\times 10^{-5}$)

Method	**With Ground Truth Keypoints**			
	GT Error		**Isometric Error**	
	MHAD	Human	MHAD	Human
Descriptor Matching (DM)	3.79	7.09	7.48	6.33
CPD	0.48	4.79	4.89	6.54
PR-GLS	0.97	3.46	5.29	3.63
Our method (with DM)	0.49	0.21	0.57	0.61
Our method (with CPD)	**0.21**	**0.16**	**0.55**	**0.60**
Our method (with PR-GLS)	0.28	0.37	0.57	0.62
Our method (with Random)	1.99	0.39	0.71	0.65
	With Detected Keypoints			
	GT Error		**Isometric Error**	
	MHAD	Human	MHAD	Human
Descriptor Matching (DM)	5.55	6.74	5.22	6.52
CPD	2.53	3.39	4.03	6.01
PR-GLS	3.02	4.27	3.50	5.60
Our method (with DM)	1.76	1.76	0.72	0.88
Our method (with CPD)	**1.14**	**1.06**	**0.72**	0.88
Our method (with PR-GLS)	1.23	1.22	0.73	**0.88**
Our method (with Random)	2.61	1.86	0.74	0.96

we manually marked the symmetrically flipped version of each ground-truth keypoint over all dataset models.

Table 6.3 shows that with our clustering-based symmetry detection method, we manage to reduce the symmetric flips on both datasets compared to the results of our algorithm without any symmetry detection. The table also includes a comparison of our clustering-based strategy with the plane fitting-based strategy used in [Küpçü and Yemez, 2017]. We see that the clustering-based symmetry handling solution provides better ground-truth error improvement than the plane symmetry solution provides.

Table 6.3: Quantitative evaluation of our **sparse** (base) correspondence estimation method **with and without symmetry detection**. (GT: Ground-truth, GT-SF: GT Error with symmetric flips ignored, C-SD: Clustering-based Symmetry Detection, P-SD: Planar Symmetry Detection, Errors $\times 10^{-5}$)

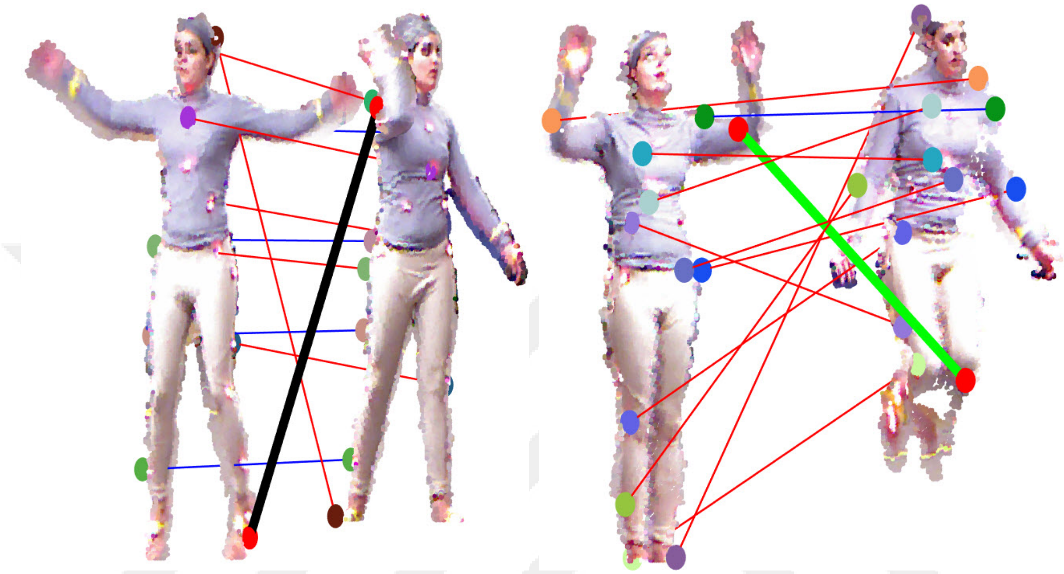
Method	**GT Error**		**GT-SF Error**	
	MHAD	Human	MHAD	Human
Without SD	0.43	0.32	0.09	**0.01**
With P-SD	0.32	0.18	0.09	0.09
With C-SD	**0.27**	**0.15**	**0.03**	0.02

6.3.1 Examples

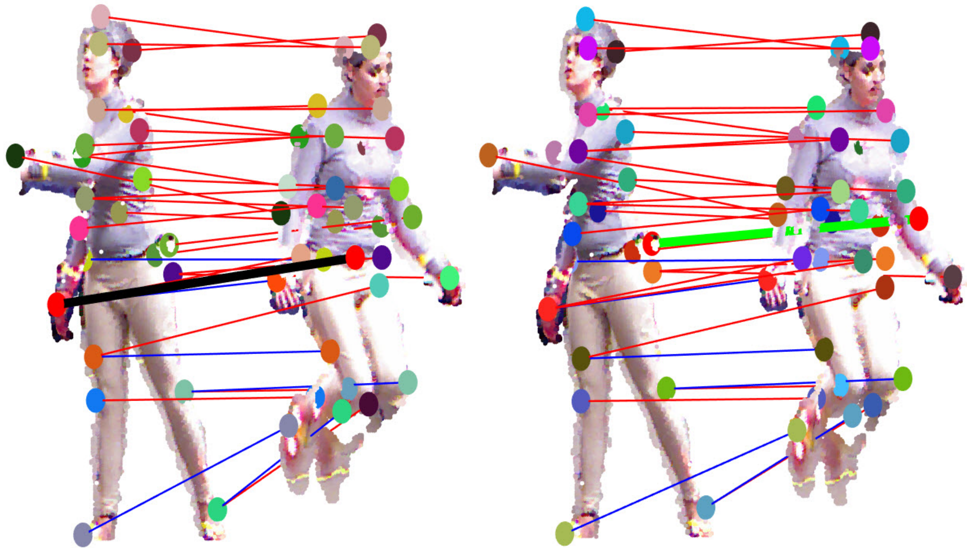
In Figure 6.10 and Figure 6.11, we display the matchings resulting from each method, which are the worst over the whole MHAD dataset according to ground-truth error (left column, black lines) and isometric error (right column, green lines), using the ground-truth points as keypoints. Regarding the isometric error, we observe that the worst matching on the MHAD dataset in the case of our method in Figure 6.11(b) (elbow to head) has smaller isometric error compared to the other methods' worst matchings. With respect to the ground-truth error, our worst matching is a symmetric flip, whereas other methods' worst matches are between different body parts.

In Figure 6.12, we visualize an example matching result for each method, using automatically detected keypoints, in order to further support our method's improvement over the other methods. While the results of the three baseline methods include matchings that are between different parts of the body, such as arm to leg, head to hand, and left leg to right leg, our method finds correspondences that are reasonably close to the true matchings, even on the right arm and left leg, where nonrigid motion is considerably large.

Figure 6.13 shows an example comparison with and without our clustering-based symmetry detection method, where we observe that our symmetry handling strategy corrects most of the symmetric flips, especially those occurring on the arms.

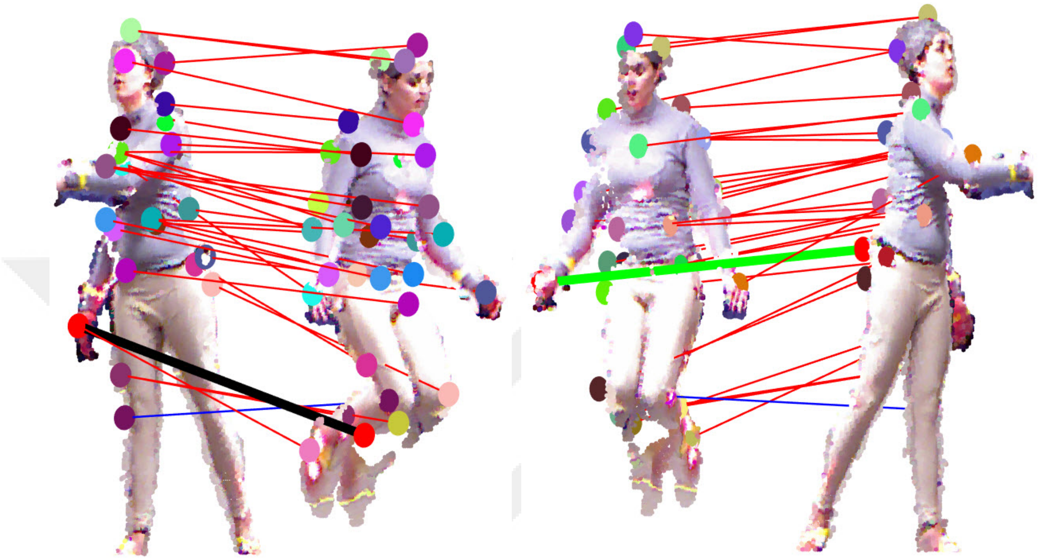


(a) Descriptor matching

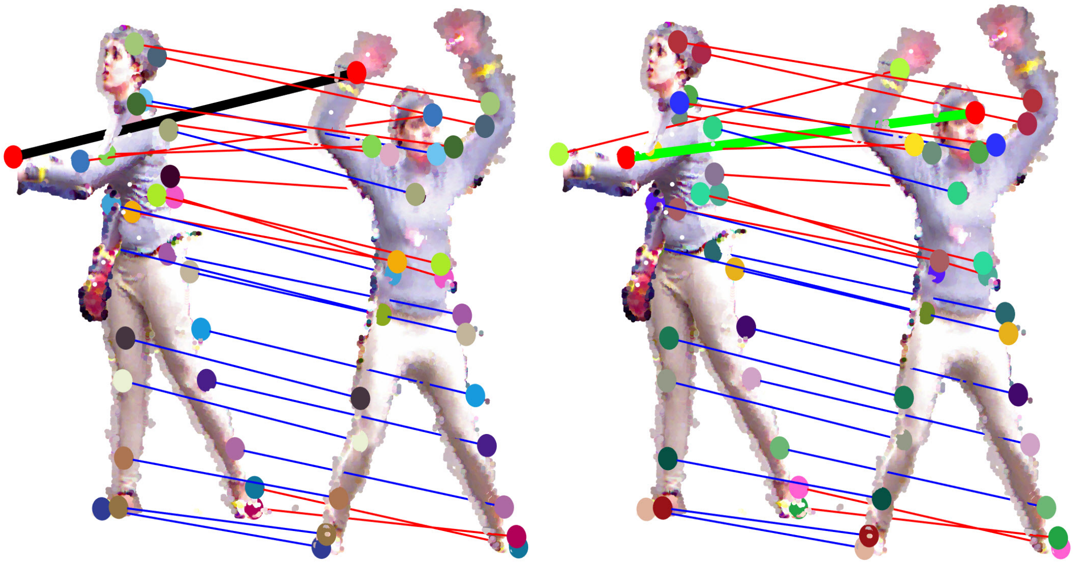


(b) CPD

Figure 6.10: The worst matchings on the MHAD dataset for each method according to ground-truth (left column, black lines) and isometric errors (right column, green lines).



(a) PR-GLS



(b) Our method

Figure 6.11: The worst matchings on the MHAD dataset for each method according to ground-truth (left column, black lines) and isometric errors (right column, green lines).

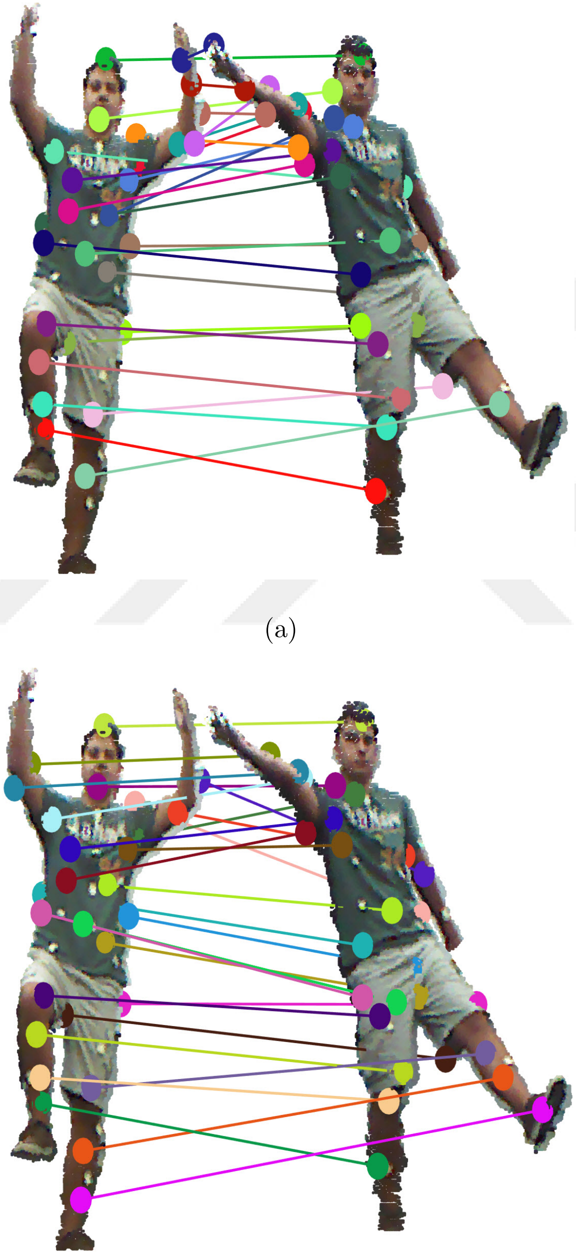
(a) Descriptor matching

(b) CPD

(c) PR-GLS

(d) Our method

Figure 6.12: Example correspondence results with automatic keypoint detection.



(a)

(b)

Figure 6.13: Example base correspondence results (a) **without** and (b) **with** our clustering-based symmetry solution.

Figure 6.14: Patchwise dense correspondences, (from left to right) at levels $k = 0$, $k = 1$ and $k^* = 9$ (last level).

6.4 Dense Correspondence Results

We visually demonstrate our coarse-to-fine dense correspondence algorithm in Figure 6.14 on a sample pair from the MHAD dataset with large deformation that causes topological changes at the legs and arms. We display the correspondence results at increasing levels of detail, and represent the matched patches with the same color. The leftmost models show our reliable base correspondences, and the rightmost models are our final dense correspondence outputs. Note that the keypoints matched are detected automatically as in all our dense correspondence experiments.

In Table 6.4, we provide quantitative evaluation of our dense correspondence algorithm in comparison to the baseline methods. Our algorithm is consistently better than the other methods in all cases. We note that the public code for PR-GLS fails to run on the MHAD and Human datasets due to their high resolution. Therefore, we downsampled the Human dataset (to generate Human^d) to be able to experiment with the PR-GLS method. Also, for a fairer comparison with the method of Guo et al. [Guo et al., 2015], we used their dataset.

Figure 6.15 plots the fraction of the dense correspondences within the error ranges of the evaluated methods for each dataset. The x axis represents the ground truth error, and the y axis represents the percentage of the correspondences. Thus, a point

Table 6.4: Quantitative evaluation of our **dense** correspondence method in comparison to baseline methods. (Human^d: Human dataset downsampled to half resolution; Errors $\times 10^{-5}$)

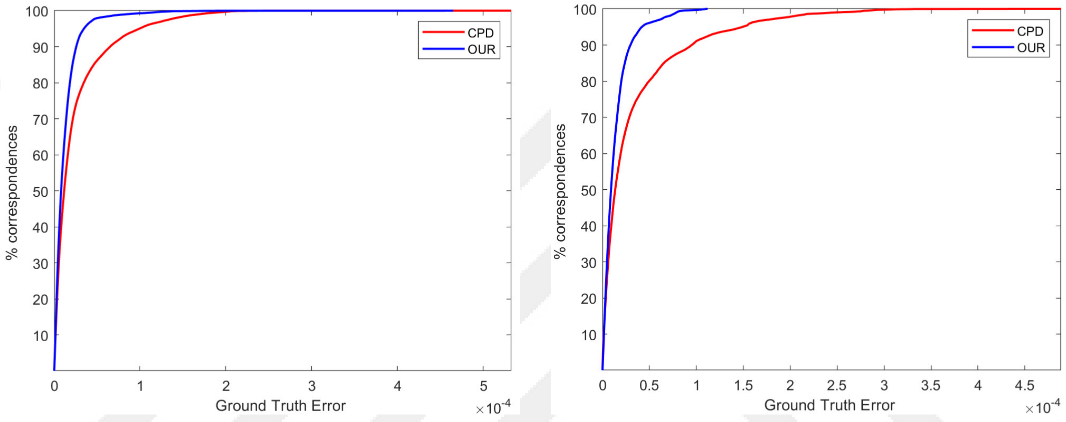
Method	**Ground Truth Error**			
	MHAD	Human	Human^d	[Guo et al., 2015]
Our method	**1.25**	**1.32**	**1.18**	**0.91**
CPD	2.41	3.28	2.80	-
PR-GLS	-	-	3.79	-
Guo et al. [Guo et al., 2015]	-	-	-	2.50

(x,y) on the plots means that y percentage of correspondences have less than or equal to x ground truth error. The blue lines, which are the best for all cases, represent our technique. We additionally observe that our maximum error result is less than the baseline methods' results for each dataset.

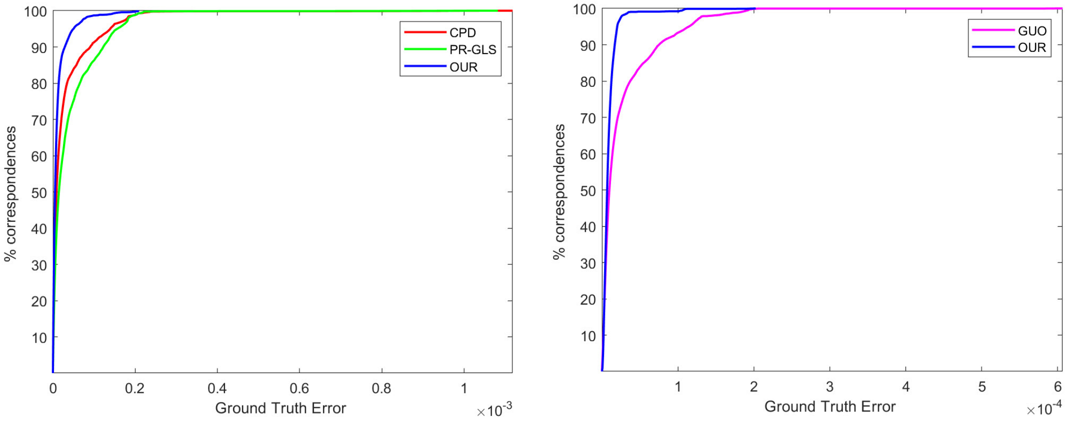
In Figures 6.16 and 6.17, we visualize our dense correspondence results on sample pairs from the Human and MHAD datasets, respectively, in comparison to CPD and PR-GLS. We observe in Figure 6.16 that our algorithm is capable of finding visually correct correspondences even on the parts exhibiting large deformation, such as arms, and parts with large gaps, such as legs. While the other methods suffer from symmetric flips on the legs and can even match the right leg to the right arm, our method successfully matches the left leg despite the unconnected part, and leaves the right leg unmatched since a reliable matching cannot be established.

Figure 6.17 displays a challenging sample pair from the MHAD dataset. On this pair, our algorithm provides visually correct correspondences on the arms exhibiting large deformation as well as on the legs undergoing a significant topological change.

In Figure 6.18, we display our dense correspondence results on a sample pair from the Guo et al. [Guo et al., 2015] dataset in comparison to the method used in [Guo et al., 2015]. This sample pair is of particular interest since the point clouds



(a) MHAD (b) Human



(c) Human Downsampled (d) Guo et al. [Guo et al., 2015]

Figure 6.15: Error fraction plots of dense correspondence results of the methods on different datasets.

(a) CPD (b) PR-GLS

(c) Our method

Figure 6.16: Example dense correspondence results on the Human dataset.

(a) CPD

(b) Our method

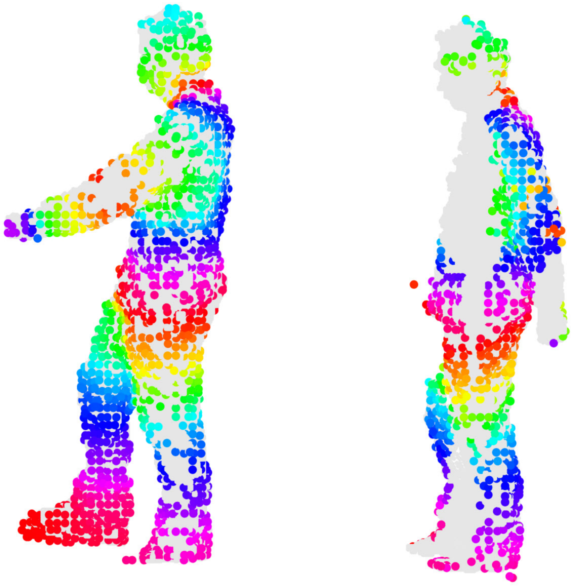
Figure 6.17: Dense correspondence on a sample pair from the MHAD dataset.

of this pair are side-views of the body, where the right arm and most of the right leg are severely occluded. Hence, this pair demonstrates the benefits of our method well. First, the method of [Guo et al., 2015] provides much sparser matching results compared to ours. Second, it can successfully match the left arm in spite of large deformation.

During the dense correspondence estimation, we further fine-tune the base correspondences at each level using all dense correspondences at that level, since we may find better corresponding points for each base points in their neighborhood. As we see in Table 6.5, both ground truth and isometric errors for each dataset are improved with the fine-tunning process.

Table 6.5: Quantitative evaluation of our **base** correspondences in comparison to the **fine-tuned base** correspondences. (Human^d: Human dataset downsampled to half resolution; Errors $\times 10^{-5}$)

	Ground Truth Error			
	MHAD	Human	Human^d	[Guo et al., 2015]
Bases	1.30	1.37	1.19	1.33
Fine-tuned bases	**1.24**	**1.30**	**1.12**	**1.31**
	Isometric Error			
	MHAD	Human	Human^d	[Guo et al., 2015]
Bases	1.09	1.47	1.45	1.07
Fine-tuned bases	**0.92**	**1.29**	**1.23**	**0.93**



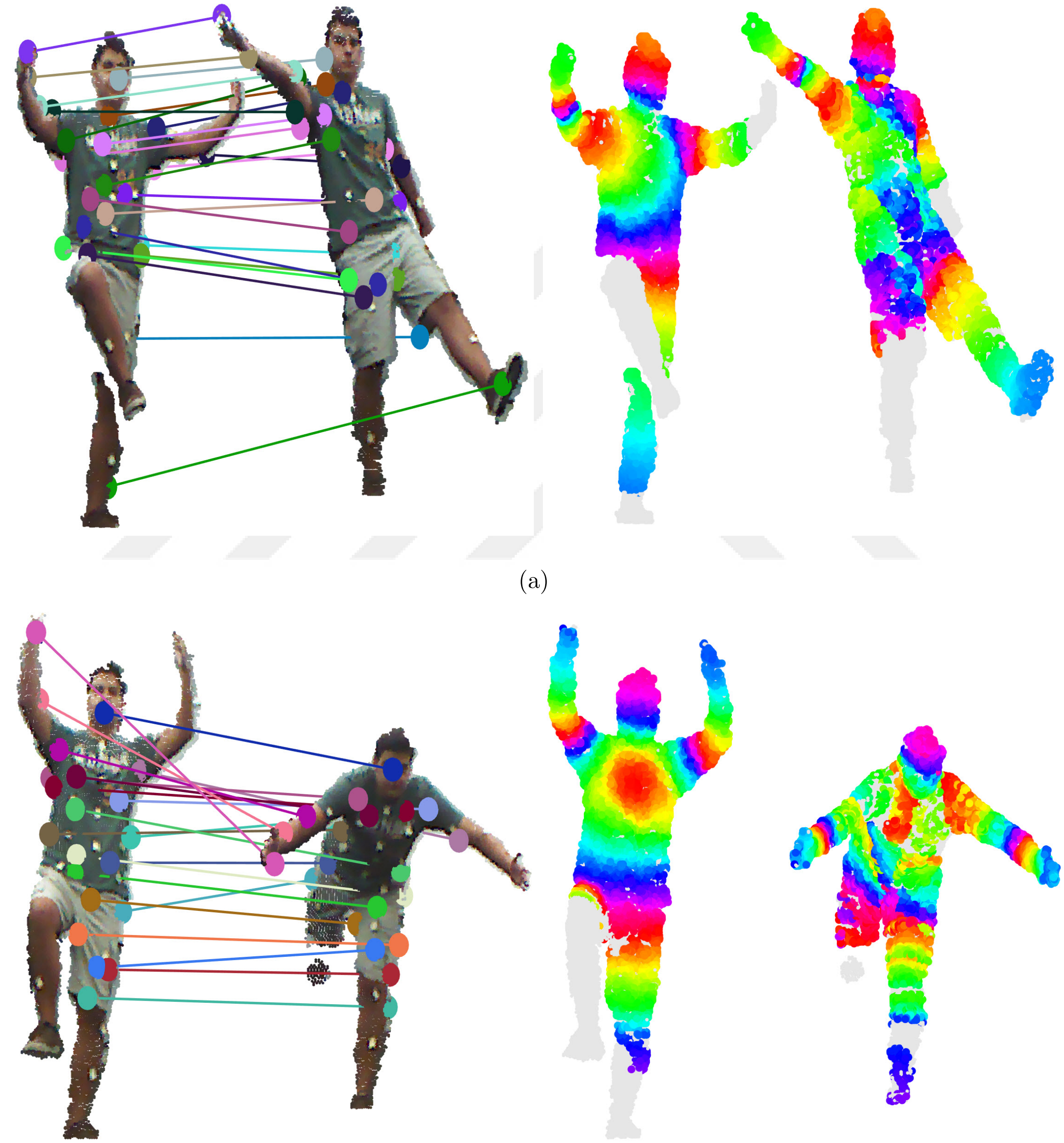
(a) Method in [Guo et al., 2015]

(b) Our method

Figure 6.18: Dense correspondence on a sample pair from the dataset of [Guo et al., 2015].

6.4.1 Examples

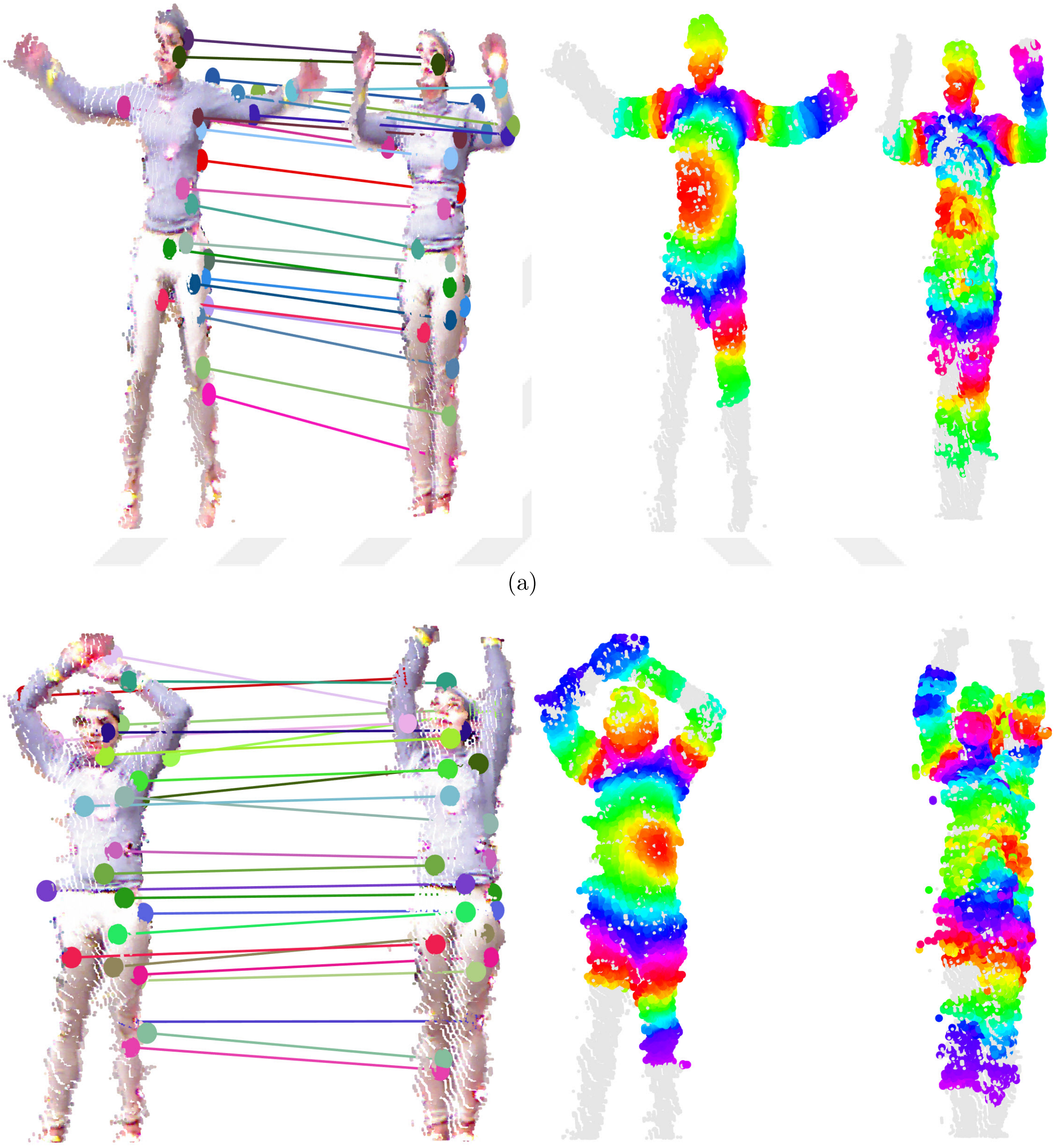
We provide more results for our sparse and dense correspondence estimation for each dataset. Figure 6.19 provides examples from the Human dataset, where there exists large deformation between the pairs, and the models include large gaps on the legs caused by occlusion. The example pairs from the MHAD and GUO datasets in Figure 6.20 and Figure 6.21, respectively, present results which include matchings on the pairs with topological changes.



(a)

(b)

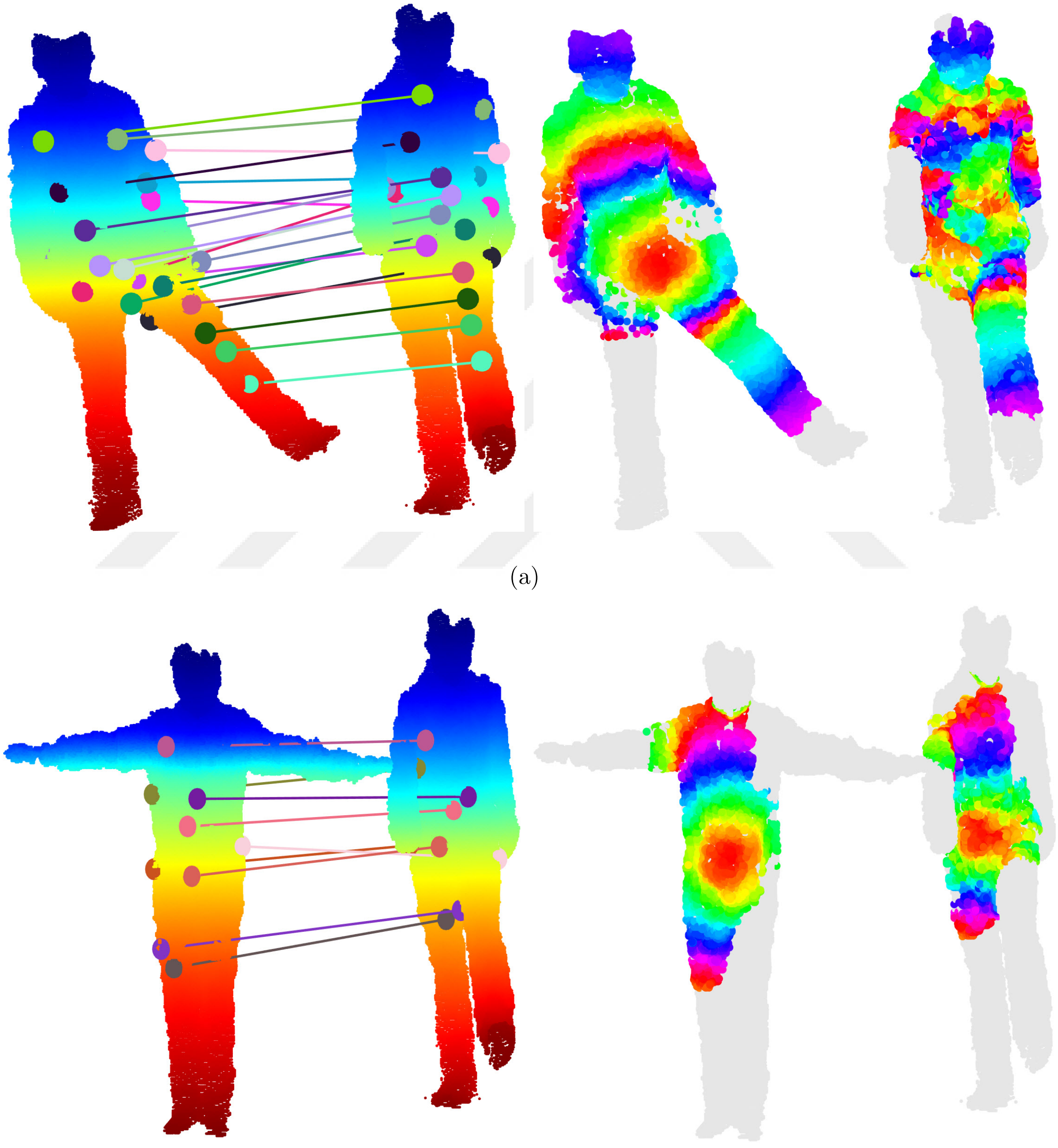
Figure 6.19: Sparse and Dense correspondences on sample pairs from the Human dataset.



(a)

(b)

Figure 6.20: Sparse and Dense correspondences on sample pairs from the MHAD dataset.



(a)

(b)

Figure 6.21: Sparse and Dense correspondences on sample pairs from the dataset of [Guo et al., 2015].

6.5 Results Regarding Applications

We also run our base correspondence estimation algorithm to find sparse correspondences between a partial depth frame and a reference complete model that is known in advance. This particular matching setup can especially be useful for semantic and pose analysis of objects and its parts from partial depth frames.

In Figure 6.22, we visually demonstrate this matching setup on some sample pairs. We first manually select 13 salient keypoints on the complete human model taken from [Bronstein et al., 2008, Rodolà et al., 2017]. We match these keypoints with automatically detected keypoints (displayed in green) from our depth frames. We visually observe that our algorithm can match most of the keypoints correctly, discarding the invisible keypoints and unreliable matchings. In Table 6.6, we provide the average ground-truth and isometric errors computed over the Human dataset. We observe that our method outperforms the baseline methods in terms of both error measures. For the ground-truth error calculation, we manually labeled the corresponding 13 ground-truth points on each frame of the Human dataset.

We further experiment with our dense correspondence algorithm to match models across datasets, i.e., between Human and MHAD datasets. We select 20 ground-truth correspondences for each dataset to be used in error calculation. Table 6.7 shows that our algorithm provides better accuracy than the CPD method (PR-GLS is excluded due to its problems with high resolution data as mentioned previously). Figure 6.23 presents visual results for sparse and dense matching across datasets, using models that suffer from occlusion. Figure 6.24 also represents visually correct matchings in spite of large missing part on the right leg. Figure 6.25 and Figure 6.26 show across dataset matching on pairs with topological changes on hands and legs.

Table 6.6: Quantitative evaluation of **sparse** correspondence results between **complete and partial** models in comparison to the baseline methods. (GT: Ground Truth, Errors $\times 10^{-5}$)

Method	**GT Error**	**Isometric Error**
CPD	1.77	6.66
PR-GLS	3.01	5.63
Our method	**0.96**	**1.32**

Table 6.7: Quantitative evaluation of **across dataset dense** correspondence estimation. (GT: Ground Truth, Errors $\times 10^{-5}$)

Method	**GT Error**
CPD	3.27
Our method	**1.18**

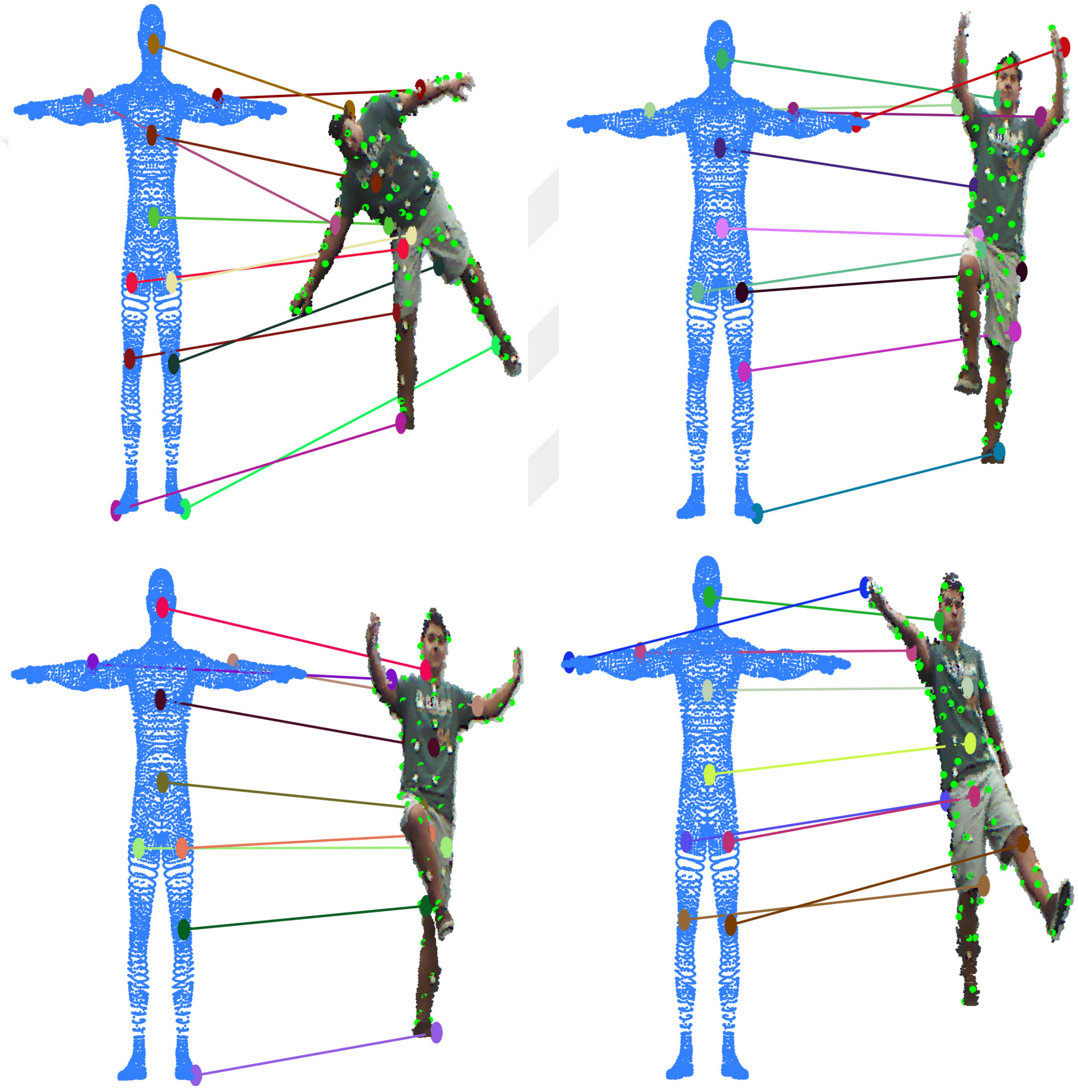
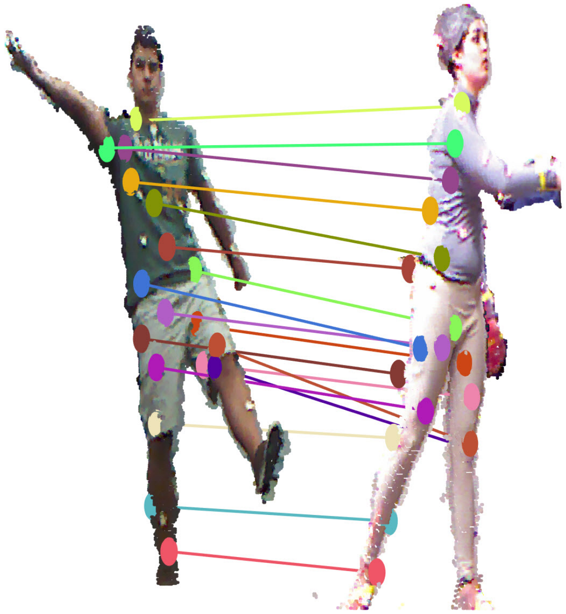
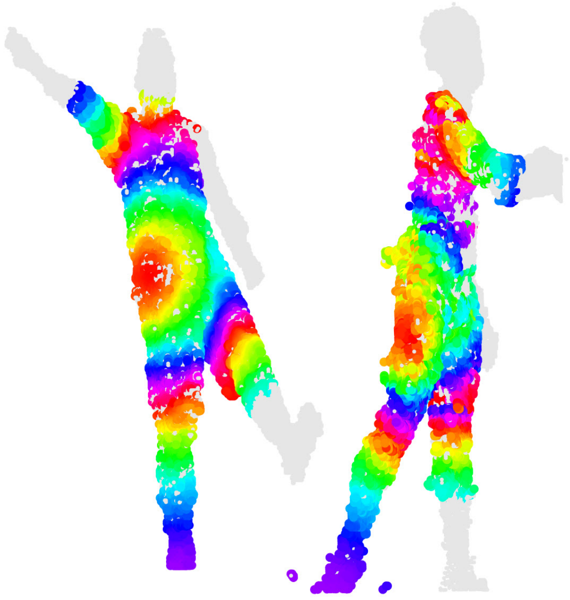


Figure 6.22: Example results for matching partial depth frames to a reference complete model.

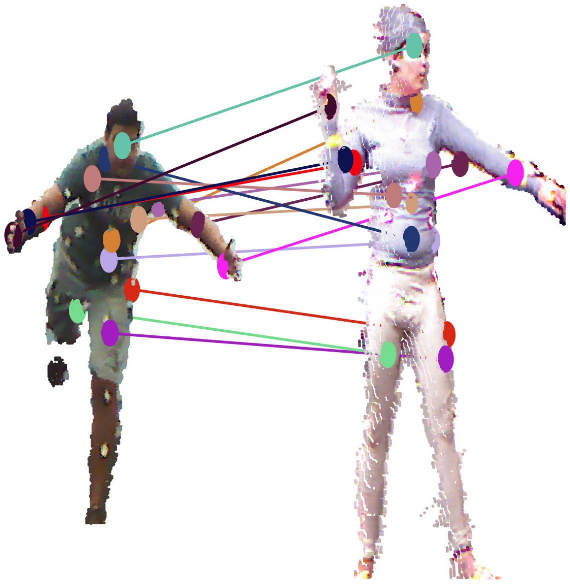


(a) Sparse

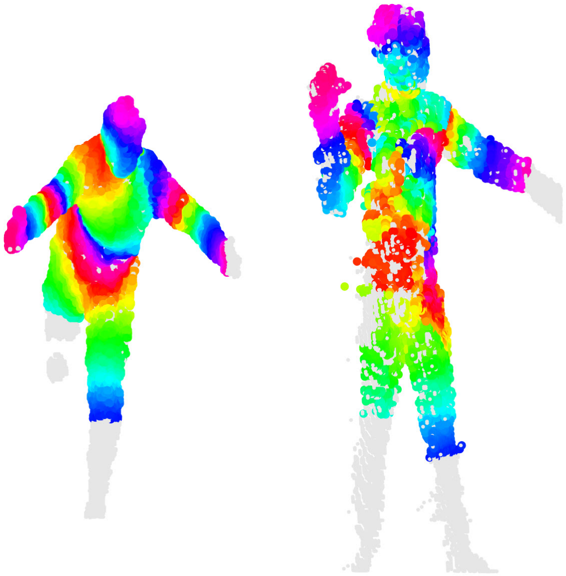


(b) Dense

Figure 6.23: Across dataset depth correspondence examples.

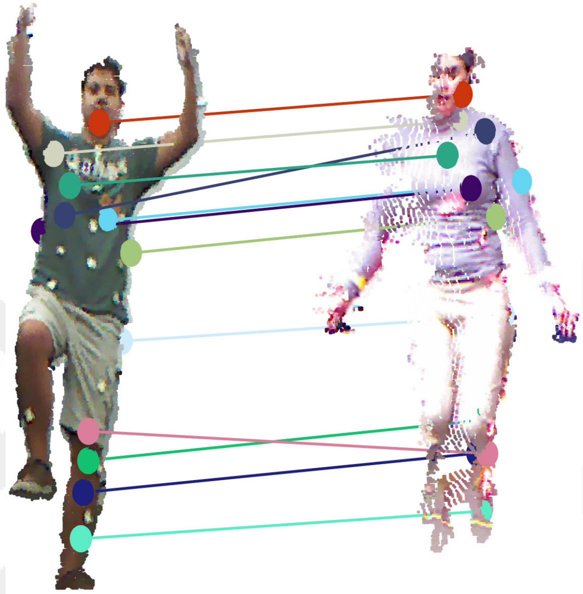


(a) Sparse

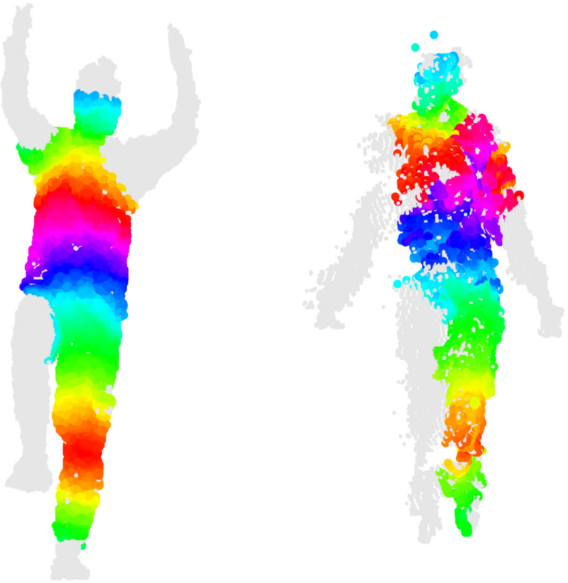


(b) Dense

Figure 6.24: Across dataset depth correspondence examples.



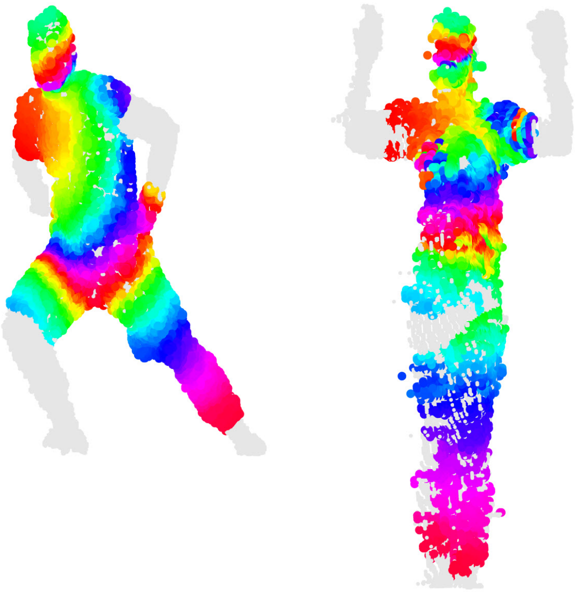
(a) Sparse



(b) Dense

Figure 6.25: Across dataset depth correspondence examples.

(a) Sparse



(b) Dense

Figure 6.26: Across dataset depth correspondence examples.

Chapter 7

CONCLUSION AND FUTURE WORK

We have proposed isometric mesh-free diffusion-based point correspondence methods to find reliable sparse/dense correspondences between point clouds generated from partial depth data exhibiting noise, large deformation, and occlusion. Our experiments show that our method provides state of the art performance on such challenging datasets, particularly on those exhibiting large deformations. We stress that our correspondence estimation methods focus on finding as many reliable correspondences as possible, pruning whenever matching is not reliable. Our results show that our algorithm is applicable in case of across dataset and complete to partial settings, which may prove useful in various applications. We have also designed a new global intrinsic symmetry detection method that addresses the symmetric flip problem when matching human shapes exhibiting reflectional symmetry, and employed it within our correspondence estimation technique.

As future work, we plan to generalize our methods to partially correspondent and non-human shapes with rigid and/or isometric deformations, that would certainly be useful for applications involving object detection and retrieval from depth images. Another possible research direction is to extend our clustering-based global symmetry detection method to handle also local intrinsic symmetries.

BIBLIOGRAPHY

[Aldoma et al., 2012] Aldoma, A., Marton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S., and Vincze, M. (2012). Point cloud library. *IEEE Robotics and Automation Magazine*, 1070(9932/12).

[Belkin and Niyogi, 2008] Belkin, M. and Niyogi, P. (2008). Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308.

[Berger and Silva, 2012] Berger, M. and Silva, C. T. (2012). Nonrigid matching of undersampled shapes via medial diffusion. *Computer Graphics Forum*, 31(5):1587–1596.

[Biasotti et al., 2016] Biasotti, S., Cerri, A., Bronstein, A., and Bronstein, M. (2016). Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum*, 35(6):87–119.

[Billings et al., 2015] Billings, S. D., Boctor, E. M., and Taylor, R. H. (2015). Iterative most-likely point registration (imlp): a robust algorithm for computing optimal shape alignment. *PloS one*, 10(3):e0117688.

[Boscaini et al., 2016] Boscaini, D., Masci, J., Rodolà, E., Bronstein, M. M., and Cremers, D. (2016). Anisotropic diffusion descriptors. *Computer Graphics Forum*, 35(2):431–441.

[Bronstein and Bronstein, 2008] Bronstein, A. M. and Bronstein, M. M. (2008). Not only size matters: regularized partial matching of nonrigid shapes. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE.

[Bronstein et al., 2006] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2006). Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172.

[Bronstein et al., 2008] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). *Numerical geometry of non-rigid shapes*. Springer Science & Business Media.

[Bronstein et al., 2010] Bronstein, A. M., Bronstein, M. M., Kimmel, R., Mahmoudi, M., and Sapiro, G. (2010). A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286.

[Brunton et al., 2014] Brunton, A., Wand, M., Wuhrer, S., Seidel, H.-P., and Weinkauf, T. (2014). A low-dimensional representation for robust partial isometric correspondences computation. *Graphical Models*, 76(2):70–85.

[Carrière et al., 2015] Carrière, M., Oudot, S. Y., and Ovsjanikov, M. (2015). Stable topological signatures for points on 3d shapes. *Computer Graphics Forum*, 34(5):1–12.

[Chang and Zwicker, 2011] Chang, W. and Zwicker, M. (2011). Global registration of dynamic range scans for articulated model reconstruction. *ACM Transactions on Graphics (TOG)*, 30(3):26.

[Chen and Koltun, 2015] Chen, Q. and Koltun, V. (2015). Robust nonrigid registration by convex optimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2039–2047.

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

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.

[Cosmo et al., 2016] Cosmo, L., Rodola, E., Masci, J., Torsello, A., and Bronstein, M. M. (2016). Matching deformable objects in clutter. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 1–10. IEEE.

[Dubrovina and Kimmel, 2011] Dubrovina, A. and Kimmel, R. (2011). Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. *Advances in Adaptive Data Analysis*, 3(01n02):203–228.

[Guo et al., 2015] Guo, H., Zhu, D., and Mordohai, P. (2015). Correspondence estimation for non-rigid point clouds with automatic part discovery. *The Visual Computer*, pages 1–14.

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

[Huang et al., 2008] Huang, Q.-X., Adams, B., Wicke, M., and Guibas, L. J. (2008). Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457.

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

[Jiang et al., 2013] Jiang, W., Xu, K., Cheng, Z.-Q., and Zhang, H. (2013). Skeleton-based intrinsic symmetry detection on point clouds. *Graphical Models*, 75(4):177–188.

[Johnson, 1997] Johnson, A. E. (1997). *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, CMU-RI-TR-97-47.

[Kim et al., 2011] Kim, V. G., Lipman, Y., and Funkhouser, T. (2011). Blended intrinsic maps. *ACM Transactions on Graphics*, 30(4):79.

[Kolmogorov, 2009] Kolmogorov, V. (2009). Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1(1):43–67.

[Kovnatsky et al., 2015] Kovnatsky, A., Bronstein, M. M., Bresson, X., and Vandergheynst, P. (2015). Functional correspondence by matrix completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 905–914.

[Küpçü and Yemez, 2017] Küpçü, E. and Yemez, Y. (2017). Reliable isometric point correspondence from depth. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 1266–1273. IEEE.

[Lafon, 2004] Lafon, S. S. (2004). *Diffusion maps and geometric harmonics*. PhD thesis, Yale University.

[Li et al., 2008] Li, H., Sumner, R. W., and Pauly, M. (2008). Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5):1421–1430.

[Liao et al., 2009] Liao, M., Zhang, Q., Wang, H., Yang, R., and Gong, M. (2009). Modeling deformable objects from a single depth camera. In *ICCV*.

[Lipman et al., 2010] Lipman, Y., Chen, X., Daubechies, I., and Funkhouser, T. (2010). Symmetry factored embedding and distance. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 103:1–103:12, New York, NY, USA. ACM.

[Lipman and Funkhouser, 2009] Lipman, Y. and Funkhouser, T. (2009). Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):72.

[Litany et al., 2017] Litany, O., Remez, T., Rodolà, E., Bronstein, A. M., and Bronstein, M. M. (2017). Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*.

[Litany et al., 2016] Litany, O., Rodolà, E., Bronstein, A. M., Bronstein, M. M., and Cremers, D. (2016). Non-rigid puzzles. In *Proceedings of the Symposium on Geometry Processing*, SGP '16, pages 135–143, Goslar Germany, Germany. Eurographics Association.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[Ma et al., 2014] Ma, J., Zhao, J., Tian, J., Yuille, A. L., and Tu, Z. (2014). Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721.

[Ma et al., 2016] Ma, J., Zhao, J., and Yuille, A. L. (2016). Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on image Processing*, 25(1):53–64.

[Mateus et al., 2008] Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., and Boyer, E. (2008). Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *CVPR*.

[Mitra et al., 2006] Mitra, N. J., Guibas, L. J., and Pauly, M. (2006). Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568.

[Mitra et al., 2013] Mitra, N. J., Pauly, M., Wand, M., and Ceylan, D. (2013). Symmetry in 3d geometry: Extraction and applications. In *Computer Graphics Forum*, volume 32, pages 1–23. Wiley Online Library.

[Myronenko and Song, 2010] Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275.

[Ofli et al., 2013] Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley MHAD: A comprehensive multimodal human action database. In *WACV*.

[Ovsjanikov et al., 2012] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. (2012). Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30.

[Ovsjanikov et al., 2010] Ovsjanikov, M., Mérigot, Q., Mémoli, F., and Guibas, L. (2010). One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564.

[Ovsjanikov et al., 2008] Ovsjanikov, M., Sun, J., and Guibas, L. (2008). Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library.

[Pajdla and Van Gool, 1995] Pajdla, T. and Van Gool, L. (1995). Matching of 3-d curves using semi-differential invariants. In *ICCV*.

[Petronetto et al., 2013] Petronetto, F., Paiva, A., Helou, E. S., Stewart, D., and Nonato, L. G. (2013). Mesh-free discrete Laplace-Beltrami operator. *Computer Graphics Forum*, 32(6):214–226.

[Podolak et al., 2006] Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., and Funkhouser, T. (2006). A planar-reflective symmetry transform for 3d shapes. *ACM Transactions on Graphics (TOG)*, 25(3):549–559.

[Raviv et al., 2007] Raviv, D., Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2007). Symmetries of non-rigid shapes. In *Workshop on Nonrigid Registration and Tracking (NRTL)*.

[Raviv et al., 2010a] Raviv, D., Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2010a). Full and partial symmetries of non-rigid shapes. *International journal of computer vision*, 89(1):18–39.

[Raviv et al., 2010b] Raviv, D., Bronstein, A. M., Bronstein, M. M., Kimmel, R., and Sapiro, G. (2010b). Diffusion symmetries of non-rigid shapes. In *Proc. 3DPVT*, volume 2.

[Raviv et al., 2013] Raviv, D., Dubrovina, A., and Kimmel, R. (2013). Hierarchical framework for shape correspondence. *Numerical Mathematics: Theory, Methods and Applications*, 6(01):245–261.

[Rodola et al., 2012] Rodola, E., Bronstein, A. M., Albarelli, A., Bergamasco, F., and Torsello, A. (2012). A game-theoretic approach to deformable shape matching. In *CVPR*.

[Rodolà et al., 2017] Rodolà, E., Cosmo, L., Bronstein, M. M., Torsello, A., and Cremers, D. (2017). Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library.

[Rodola et al., 2013] Rodola, E., Torsello, A., Harada, T., Kuniyoshi, Y., and Cremers, D. (2013). Elastic net constraints for shape matching. In *ICCV*.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *3DIM*.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *ICRA*.

[Sahillioğlu and Yemez, 2011] Sahillioğlu, Y. and Yemez, Y. (2011). Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum*, 30(5):1461–1470.

[Sahillioğlu and Yemez, 2012] Sahillioğlu, Y. and Yemez, Y. (2012). Minimum-distortion isometric shape correspondence using em algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2203–2215.

[Sahillioğlu and Yemez, 2013] Sahillioğlu, Y. and Yemez, Y. (2013). Coarse-to-fine isometric shape correspondence by tracking symmetric flips. *Computer Graphics Forum*, 32(1):177–189.

[Sahillioğlu and Yemez, 2014] Sahillioğlu, Y. and Yemez, Y. (2014). Partial 3-d correspondence from shape extremities. *Computer Graphics Forum*, 33(6):63–76.

[Sharma et al., 2011] Sharma, A., Horaud, R., Cech, J., and Boyer, E. (2011). Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In *CVPR*.

[Sipiran et al., 2014] Sipiran, I., Gregor, R., and Schreck, T. (2014). Approximate symmetry detection in partial 3d meshes. *Computer Graphics Forum*, 33(7):131–140.

[Tam et al., 2013] Tam, G. K., Cheng, Z.-Q., Lai, Y.-K., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X.-F., and Rosin, P. L. (2013). Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217.

[Teleimmersion Lab,] Teleimmersion Lab, University of California, B. Berkeley multimodal human action database (MHAD). http://tele-immersion.citris-uc.org.

[Tevs et al., 2009] Tevs, A., Bokeloh, M., Wand, M., Schilling, A., and Seidel, H.-P. (2009). Isometric registration of ambiguous and partial data. In *CVPR*.

[Tombari et al., 2010] Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *ECCV*.

[Tong et al., 2012] Tong, J., Zhou, J., Liu, L., Pan, Z., and Yan, H. (2012). Scanning 3d full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):643–650.

[Van Kaick et al., 2011] Van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. (2011). A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707.

[Wang et al., 2011] Wang, C., Bronstein, M. M., Bronstein, A. M., and Paragios, N. (2011). Discrete minimum distortion correspondence problems for non-rigid shape matching. In *SSVM*.

[Wang et al., 2014] Wang, H., Simari, P., Su, Z., and Zhang, H. (2014). Spectral global intrinsic symmetry invariant functions. In *Proceedings of Graphics Interface 2014*, pages 209–215. Canadian Information Processing Society.

[Wang et al., 2012] Wang, R., Choi, J., and Medioni, G. (2012). Accurate full body scanning from a single fixed 3d camera. In *3DIMPVT*.

[Wei et al., 2016] Wei, L., Huang, Q., Ceylan, D., Vouga, E., and Li, H. (2016). Dense human body correspondences using convolutional networks. In *CVPR*.

[Weiss et al., 2011] Weiss, A., Hirshberg, D., and Black, M. J. (2011). Home 3d body scans from noisy image and range data. In *ICCV*.

[Xu et al., 2012] Xu, K., Zhang, H., Jiang, W., Dyer, R., Cheng, Z., Liu, L., and Chen, B. (2012). Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (TOG)*, 31(6):181.

[Xu et al., 2009] Xu, K., Zhang, H., Tagliasacchi, A., Liu, L., Li, G., Meng, M., and Xiong, Y. (2009). Partial intrinsic reflectional symmetry of 3d shapes. In *ACM Transactions on Graphics (TOG)*, volume 28, page 138. ACM.

[Yoshiyasu et al., 2016] Yoshiyasu, Y., Yoshida, E., and Guibas, L. (2016). Symmetry aware embedding for shape correspondence. *Computers & Graphics*, 60:9–22.

[Yoshiyasu et al., 2014] Yoshiyasu, Y., Yoshida, E., Yokoi, K., and Sagawa, R. (2014). Symmetry-aware nonrigid matching of incomplete 3d surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4193–4200.

[Zhang et al., 2013] Zhang, Z., Yin, K., and Foong, K. W. (2013). Symmetry robust descriptor for non-rigid surface matching.

[Zhong, 2009] Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *ICCV Workshops*.