Algorithms for 3D Isometric Shape Correspondence

by

Yusuf Sahillioğlu

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

Koç University

August, 2012

Koç University
Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Yusuf Sahillioğlu

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.

Chair of Supervisory Committee:

Reading Committee:

Assoc. Prof. Yücel Yemez

Prof. Bülent Sankur

Asst. Prof. T. Metin Sezgin

Assoc. Prof. Engin Erzin

Prof. Murat Tekalp

Date:

To those who should have seen this.

ABSTRACT

There are many pairs of objects in the digital world that need to be related before performing any comparison, transfer, or analysis in between. The shape correspondence algorithms essentially address this problem by taking two shapes as input with the aim of finding a mapping that couples similar or semantically equivalent surface points of the given shapes.

We focus on computing correspondences between some featured or all present points of two semantically similar 3D shapes whose surfaces overlap completely or partially up to isometric, i.e., distance-preserving, deformations and scaling. Differently put, our isometric shape correspondence algorithms handle several different cases for the shape correspondence problem that can be differentiated based on how similar the shape pairs are, whether they are partially overlapped, the resolution of the desired mapping, etc.

Although there exist methods that can, in most cases, satisfactorily establish 3D correspondences between two given shapes, these methods commonly suffer from certain drawbacks such as high computational load, incapability of establishing a correspondence which is partial and dense at the same time, approximation and embedding errors, and confusion of symmetrical parts of the shapes. While the existing methods constitute a solid foundation and a good starting point for the shape correspondence problem, our novel solutions designed for a given scenario achieve significant improvements as well as contributions.

We specifically explore the 3D shape correspondence problem under two categories as complete and partial correspondences where the former is categorized further according to the output resolution as coarse and dense correspondences. For complete correspondence at coarse resolution, after jointly sampling evenly-spaced feature vertices on shapes, we formulate the problem as combinatorial optimization over the domain of all possible mappings between source and target features, which then reduces within a probabilistic framework to a log-likelihood maximization problem that we solve via EM (Expectation Maximization) algorithm. Due to computational limitations of this approach, we design a fast coarse-

to-fine algorithm to achieve dense correspondence between all vertices of complete models with specific care on the symmetric flip issue. Our scale normalization method based on a novel scale-invariant isometric distortion measure, on the other hand, handles a particular and rather restricted setting of partial matching whereas our rank-and-vote-and-combine (RAVAC) algorithm deals with the most general matching setting, where both two solutions produce correspondences that are partial and dense at the same time.

In comparison with many state-of-the-art methods, our algorithms are tested by a variety of two-manifold meshes representing 3D shape models based on real and synthetic data.

ÖZETÇE

Sayısal dünyada, aralarında herhangi bir karşılaştırma, aktarım, veya analiz yapabilmek için ilişkilendirilmesi gereken çok sayıda nesne modeli vardır. Şekil eşleme algoritmaları, bu probleme çözüm olarak verili iki nesne modeli arasında benzer veya anlamsal olarak denk yüzey noktalarını eşleştirmeyi hedeflerler.

Bu çalışmada anlamsal olarak yakın ve yüzeyleri tamamen veya kısmi olarak örtüşen iki 3B şeklin öznitelik noktaları veya tüm noktaları arasında eşleştirme hesaplama problemine odaklanıyoruz. Bu örtüşmenin izometrik, yani uzaklık-koruyan, deformasyonlara ve ölçeklemeye karşı değişimsiz olduğunu varsayıyoruz. Bir başka deyişle, bizim geliştirdiğimiz izometrik şekil eşleme algoritmaları şekil eşleme probleminin, şekiller arasındaki benzerlik miktarı, örtüşmenin kısmi ya da tam olması, istenilen eşlemenin çözünürlüğü gibi çeşitli etkenlere bağlı olarak ayrışan birçok farklı durumun üstesinden gelirler.

Verilen iki şekil arasında çoğu zaman tatmin edici 3B eşlemeler bulabilen yöntemler olsa da bu yöntemlerin yüksek hesap yükü, hem kısmi hem yoğun eşleme yapamama, yaklaşıklık ve gömme hataları, simetrik parçaların karıştırılması gibi çeşitli sorunları vardır. Mevcut yöntemler şekil eşleme problemi için sağlam bir temel ve iyi bir başlangıç noktası oluştururken, bu çalışmada, verilen senaryoya göre tasarlanan yeni çözümler bu temel problemin ele alınmasında belirgin gelişmeler ve katkılar sağlamaktadır.

3B şekil eşleme problemini tam ve kısmi eşleme olarak iki ana grupta inceliyoruz, ve ilk grubu çıktı çözünürlüğüne göre kaba ve yoğun eşlemeler olarak kendi içinde ikiye ayırıyoruz. Kaba çözünürlükteki tam eşleme problemi için, eşit uzaklıklarla ayrılan öznitelik noktalarını iki şekil yüzeyi üzerinden ortaklaşa örnekledikten sonra, problemi kaynak ve hedef şekillerdeki örnekler arasında olası tüm gönderimler üzerinden tanımlanan bir kombinatoryal eniyileme olarak formule ediyoruz ve bunu, olasılıksal bir yaklaşımla EM algoritması kullanarak çözebileceğimiz bir olasılıksal çatı içindeki log-olabilirlik enbüyütme problemine dönüştürüyoruz. Bu yöntem yüksek hesap yükü nedeniyle ancak kaba çözünürlükte görece az sayıda nokta arasında eşleme yapabilir. Tez çalışmasının bir sonraki aşamasında, şekil modellerindeki bütün noktalar arasında yoğun eşleme yapabilen hızlı, kabadan-inceye (çoklu çözünürlüklü), ve simetrik flip problemini de dikkate alan yeni bir algoritma tasarlıyoruz. Ölçek-değişimsiz ölçütümüz üzerine dayalı şekil ölçek düzgeleme yöntemimiz, diğer yandan, kısmi eşleme probleminin özel ve kısıtlı bir halinin üstesinden gelirken, diz-oyla-ve-birleştir (RAVAC) algoritmamız en genel kısmi eşleme durumunu ele alır. Bu iki yöntem de hem kısmi hem yoğun eşlemeler üretir.

Bu çalışmada geliştirdiğimiz bütün yöntemleri, gerçek ve sentetik veriye dayalı çeşitli 3B şekil vertabanları üzerinde, literatürde mevcut diğer yöntemlerle karşılaştırmalı olarak sınıyoruz.

ACKNOWLEDGMENTS

No single paragraph can express my gratitude to my sagacious guide Yücel Yemez who has not only supervised this PhD research successfully but also tolerated my occasional relaxed behaviors and primitive paper drafts. Just like in MS years, he has eased the entrance into the research topic in the initial year and then the rest has followed, again with his full support whenever necessary.

I thank to my friends that have made these years exciting. With Mehmet Ali Yatbaz we hit the top in foosball. With Emre Ünal we defined what awesome is. With Emrah Çem we showed how to play soccer in style. With Arda Aytekin we started wrong ended well. With Eray Varlık we played chess. With Barış Çağlar, we won in a row. With Akif Yalçınkaya, Mustafa Haboğlu, Mehmet Ayyıldız, Talha Akyol, and Enis Sert we competed endlessly.

Last but not least, I thank a lot to my beloved mother Nevin Sahillioğlu for always being there for me. We were together in my conferences abroad with plenty of adventures and her watching me presenting. She helped me during all my relocating, nourished me with her delicious foods, respected and supported my academic moves, and made me a decent person. My brother Erdem deserves thanks for motivational talks and comments. Finally, I remember with love and respect my father Edip Sahillioğlu, the top in the list of people who should have seen this.

TABLE OF CONTENTS

List of Tables **xiv**

List of Figures **xv**

Chapter 1: Introduction **1**

- 1.1 Scope . . . 1
- 1.2 Contributions . . . 2
- 1.3 Overviews . . . 3
  - 1.3.1 Complete correspondence . . . 4
    - 1.3.1.1 Coarse correspondence . . . 4
    - 1.3.1.2 Dense correspondence . . . 4
  - 1.3.2 Partial correspondence . . . 4
- 1.4 Putting It All Together . . . 5
- 1.5 Organization . . . 5

Chapter 2: Related Work and Preliminaries **7**

- 2.1 Classification of Correspondence Methods . . . 7
  - 2.1.1 Non-isometric methods . . . 7
  - 2.1.2 Isometric methods . . . 8
- 2.2 Shape Classes . . . 8
  - 2.2.1 Rigid shapes . . . 8
  - 2.2.2 Non-rigid shapes . . . 10
- 2.3 Shape Similarity . . . 11
  - 2.3.1 Global similarity: distance metrics . . . 11
    - 2.3.1.1 Isometry . . . 11
    - 2.3.1.2 Geodesic distance . . . 12

2.3.1.3 Diffusion distance 14
2.3.1.4 Commute-time distance 15
2.3.1.5 Biharmonic distance 15
2.3.2 Local similarity: descriptors 15
2.3.2.1 Curvature 16
2.3.2.2 Average geodesic distance function 16
2.3.2.3 Spectral signatures 16
2.3.2.4 Signatures by geometric primitives 17
2.4 Shape Sampling 18
2.4.1 Uniform sampling 18
2.4.2 Evenly-spaced sampling 18
2.4.2.1 COES sampling 19
2.4.2.2 FPS sampling 19
2.4.2.3 Centroidal Voronoi sampling 19
2.4.2.4 Stratified sampling 20
2.4.3 Extremity sampling 21
2.5 Shape Embedding 22
2.5.1 Euclidean embedding 22
2.5.1.1 MDS 22
2.5.1.2 Laplacian embedding 24
2.5.2 Non-Euclidean embedding 25
2.5.2.1 Generalized MDS 26
2.5.2.2 Spherical embedding 26
2.5.2.3 Möbius embedding 27

Chapter 3: Complete Shape Correspondence 28
3.1 Coarse Correspondence of Complete Shapes 28
3.1.1 Literature review 28
3.1.2 Problem formulation and overview 32
3.1.3 Sampling 34

3.1.4 EM framework 35
3.1.5 Initialization 37
    3.1.5.1 Spectral embedding 37
    3.1.5.2 Alignment 38
    3.1.5.3 Isometric distortion 38
3.1.6 Optimization 38
    3.1.6.1 Bipartite graph matching 39
    3.1.6.2 Greedy optimization 40
    3.1.6.3 EM iterations 42
3.1.7 Computational Complexity 42
3.1.8 Experiments 43
    3.1.8.1 Datasets 43
    3.1.8.2 Evaluation metrics 43
    3.1.8.3 Parameter setting 44
    3.1.8.4 Results 44
3.1.9 Conclusion 57
3.2 Dense Correspondence of Complete Shapes 59
3.2.1 Literature review 59
3.2.2 Problem statement and overview 60
3.2.3 Coarse-to-fine sampling 61
3.2.4 Correspondence algorithm 62
    3.2.4.1 Patch-based combinatorial matching 63
    3.2.4.2 Correspondence merging 65
    3.2.4.3 Overall algorithm 66
3.2.5 An insight to why the algorithm works 67
3.2.6 Computational complexity 69
3.2.7 Experimental results 71
3.2.8 Conclusion 78
3.3 Symmetric Flip Problem With A Solution 80
3.3.1 Introduction 80

3.3.2 Coarse-to-fine combinatorial matching 82
3.3.3 Symmetric flip problem 83
3.3.4 Tracking symmetric flips 84
3.3.5 Computational complexity 87
3.3.6 Experimental results 87
3.3.6.1 Dataset: TOSCA 88
3.3.6.2 Dataset: Watertight 91
3.3.6.3 Dataset: SHREC'11 93
3.3.6.4 Dataset: SCAPE 94
3.3.6.5 Application to GMDS 95
3.3.6.6 Timing 97
3.3.6.7 Discussion 97
3.3.7 Conclusion 99

Chapter 4: Partial Shape Correspondence 103

4.1 Scale Normalization for Isometric Shape Matching 103
4.1.1 Literature review 103
4.1.2 Isometric distortion 106
4.1.2.1 Scale-invariant isometric distortion 106
4.1.2.2 Isometric distortion with normalized geodesics 108
4.1.3 Feature point selection 109
4.1.4 Combinatorial matching 110
4.1.5 Extension to dense correspondence 110
4.1.6 Computational complexity 112
4.1.7 Experimental results 113
4.1.8 Conclusion 119
4.2 Isometric Partial Matching Algorithm 121
4.2.1 Literature review 121
4.2.2 Sampling 122
4.2.3 Ranking 123

4.2.3.1 Safe map generators 124

4.2.4 Voting 126

4.2.4.1 Finding regions of interest 127

4.2.4.2 Dense region sampling 128

4.2.4.3 Dense region matching 128

4.2.4.4 Vote matrix 128

4.2.5 Combining 129

4.2.6 Extension to dense map 130

4.2.7 Computational complexity 132

4.2.8 Experimental results 133

4.2.9 Conclusion 140

Chapter 5: Conclusions and Future Work 143

Bibliography 145

LIST OF TABLES

1.1 Our shape correspondence algorithms 6

3.1 Quantitative comparisons with varying number of samples 44
3.2 Quantitative comparisons with the spectral method 52
3.3 Quantitative comparisons with GMDS (coarse correspondence) 55
3.4 Quantitative comparisons with the spectral method 77
3.5 Quantitative comparisons with GMDS (dense correspondence) 77
3.6 Quantitative comparisons with BIM and GMDS # 1 89
3.7 Quantitative comparisons with BIM # 1 89
3.8 Quantitative comparisons with GMDS # 1 91
3.9 Quantitative comparisons with BIM and GMDS # 2 92
3.10 Quantitative comparisons with BIM and GMDS # 3 93
3.11 Quantitative comparisons with BIM and GMDS # 4 94
3.12 Quantitative comparisons with BIM and GMDS # 5 95
3.13 Quantitative comparisons with BIM and GMDS # 6 95
3.14 Quantitative comparisons with BIM # 2 95
3.15 Quantitative comparisons with GMDS # 2 97

4.1 Quantitative comparisons with MV (part matching) 119
4.2 Quantitative comparisons with MV (generic partial matching) 141
4.3 Quantitative comparisons with BIM 141

LIST OF FIGURES

1.1 Input scenarios for our correspondence algorithms 2

2.1 Rigid transformations 9

2.2 ICP demonstration 10

2.3 Non-rigid transformations 11

2.4 Geodesic distances with shortcut edges 13

2.5 Diffusion distances 14

2.6 Local shape descriptors 17

2.7 General point sampling 20

2.8 Extreme point sampling 21

2.9 Euclidean embedding 23

3.1 Overview of our coarse correspondence scheme 34

3.2 EM algorithm 37

3.3 Euclidean embedding and alignment 39

3.4 Greedy algorithm 41

3.5 Greedy optimization algorithm pseudocode 41

3.6 Limitations of our coarse correspondence method 45

3.7 Coarse correspondences between a horse pair at different resolutions 46

3.8 Coarse correspondences between dancing man pairs 47

3.9 Coarse correspondences between ballerina pairs 48

3.10 Coarse correspondence between a hybrid human pair 49

3.11 Improvements to initial spectral matching 50

3.12 One-to-one and many-to-one coarse maps between a hybrid human pair . . . 50

3.13 Visual comparisons with the spectral method 53

3.14 Visual comparisons with GMDS 56

3.15 Coarse-to-fine sampling algorithm pseudocode 63
3.16 Coarse-to-fine sampling 64
3.17 Coarse-to-fine patch-based combinatorial matching 65
3.18 Merging of patch-based correspondences 67
3.19 Dense correspondence algorithm pseudocode 68
3.20 Inclusion assertion showing why the algorithm works 69
3.21 Inclusion assertion in action 70
3.22 Dense correspondence between a ballerina pair 72
3.23 Dense correspondences between horse pairs and hybrid animal pairs 73
3.24 Correspondences at increasing levels of detail 74
3.25 Graph of the surface coverage and isometric distortion as levels proceed . . . 75
3.26 Dense correspondences between a hybrid human pair at different resolutions . 76
3.27 Visual comparisons with the spectral method (dense correspondence) 78
3.28 Visual comparisons with GMDS (dense correspondence) 79
3.29 Visual comparison with BIM 81
3.30 Block diagram of our multiresolution isometric dense shape matcher 83
3.31 Symmetric flip problem due to coarse sampling 84
3.32 Symmetric flip handling of our original dense matcher 85
3.33 Tracking of local minima as our symmetric flip solution 86
3.34 Comparative dense correspondences between TOSCA models 90
3.35 Comparative dense correspondences between collapsed TOSCA models 92
3.36 Comparative dense correspondences between Watertight models 94
3.37 Comparative dense correspondences between SHREC'11 models 96
3.38 Comparative dense correspondences between SCAPE models 101
3.39 Tracking of local minima for centaur models 102

4.1 Scale normalization overview 106
4.2 Our scale-invariant isometric distortion measure 107
4.3 Our distortion measure based on normalized geodesics 108
4.4 Feature sampling 109

4.5 Our combinatorial matching framework 111
4.6 Partial correspondences between man pairs 115
4.7 Partial correspondences between horse pairs 116
4.8 Partial correspondences between SHREC'11 models 116
4.9 Complete correspondences between animal and man and gorilla-man pairs . . 117
4.10 Comparative partial correspondences between human and animal pairs 118
4.11 Different choices of initial map size 119
4.12 Point sampling 123
4.13 Common shape parts 125
4.14 Safe map generators 126
4.15 Voting process 127
4.16 Confidence computation 129
4.17 RAVAC algorithm pseudocode 131
4.18 Partial and complete correspondences between horse pairs 135
4.19 Partial correspondences between shapes both having uncommon parts 136
4.20 Partial and complete correspondences between SHREC'11 models 137
4.21 Complete correspondences between gorilla-man pairs 137
4.22 Visual comparisons with MV 138
4.23 Complete dense correspondences between SCAPE models 139
4.24 Visual comparisons with BIM (generic partial matching) 140

Chapter 1

INTRODUCTION

3D shape correspondence is a fundamental problem in both computer vision and computer graphics; it aims to find a mapping between some or all of the surface points of two given shapes, in other words, it seeks on two given shapes for pairs of surface points that are similar or semantically equivalent. Dealing with the 3D shape correspondence problem is important because it lays the foundations of numerous practical applications such as shape morphing and interpolation [1][2], mesh parameterization [3][4], rigid or non-rigid shape registration [5][6], time-varying reconstruction [7][8], shape recognition and retrieval [9][10], shape segmentation [11][12], texture mapping [13], deformation transfer [14], mesh watermarking [15], and statistical shape analysis [16], all of which and many others can be reviewed in [17], [18].

Establishing consistent maps or improving the existing ones between more than two shapes is also of interest as there has recently been a growing trend for 3D reconstruction of time-varying real scenes, and there already exist several methods that can generate mesh sequences representing the geometry and the motion of the dynamic objects. However, there is yet relatively little work in the literature on the analysis of such shape collections [19][20].

1.1 Scope

The goal of this research is to design algorithms that can efficiently establish robust coarse and/or dense correspondences between

(a) two perfectly isometric shapes, e.g., two different poses of the same object, (Fig. 1.1-a),

(b) two nearly isometric shapes, e.g., two different people, (Fig. 1.1-b),

(c) two partially isometric shapes where

(1) one shape is an isometric part of the other, e.g., upper body vs. man (Fig. 1.1-c_1),

(2) shapes have parts which are not in common, centaur vs. man (Fig. 1.1-c_2),

where the shapes to be matched are always allowed to come in arbitrary scales.

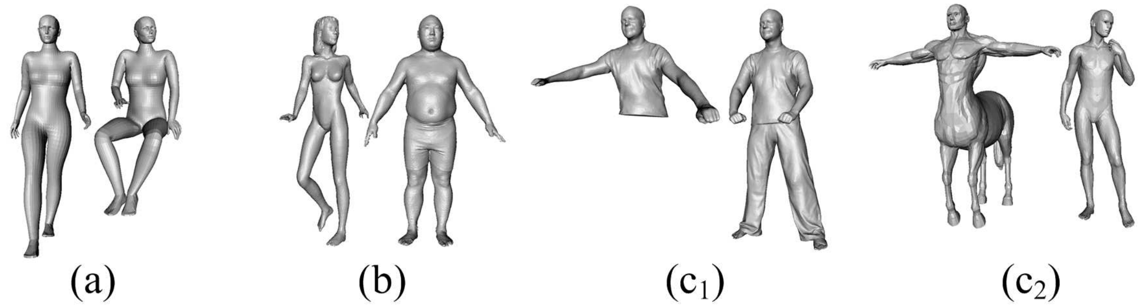

Figure 1.1: Correspondences to be sought between two perfectly isometric (a), nearly isometric (b), and partially isometric (c_1 and c_2) shapes.

We use geodesic distance metric to capture the global intrinsic structure of the shapes which is invariant to isometric deformations such as rotation, translation, and bending. If two shapes are perfectly isometric, then there exists an isometry, i.e., a distance-preserving mapping, between these shapes such that the geodesic distance between any two points on one shape is exactly the same with the geodesic distance between their correspondences on the other. However, since two digital shapes are hardly ever perfectly isometric, even for different poses of a rigid object, due to imperfections of the modeling process and/or geometry discretization errors, it is not usually possible to find a zero distortion mapping, hence the goal rather becomes minimization of an conveniently defined isometric distortion function.

1.2 Contributions

We present robust algorithms for various aspects of 3D isometric shape correspondence that are not only efficient but also fully automatic. Contributions towards this goal are as follows:

- We introduce four new sampling algorithms, namely curvature-oriented evenly-spaced (COES), coarse-to-fine, and two different extremity sampling, that can both be embedded into any geometry processing algorithm working on manifold meshes on which geodesic distances can be computed.

- For each input scenario, we define isometric distortion functions that measure, for a given map, deviation from isometry in the original 3D Euclidean space wherein isometry itself is defined, hence free of any embedding and approximation errors.

- We optimize these functions again in the 3D Euclidean space by employing well-established paradigms such as bipartite graph matching, greedy optimization, EM algorithm, combinatorial optimization, and voting.

- We share the fastest computational complexity on dense correspondence with [21] which, however, comes with sphere topology restriction and triangulation sensitivities.

- We present a map tracking mechanism with which the symmetric flip problem that is inherent to multiresolution isometric shape matching algorithms is substantially handled.

- We address partial shape correspondence in the most general setting where our methodology admits shapes with quite small similarity overlap. Besides, we are capable of establishing a correspondence which is partial and dense at the same time.

- Our algorithms have no restriction on shape topology and are all insensitive to the peculiarities of the particular triangulation.

1.3 Overviews

In the sequel, we overview two different and important aspects of 3D isometric shape correspondence problem for both of which we propose published or to be published solutions at coarse and/or dense resolutions.

1.3.1 Complete correspondence

Complete correspondence solutions seek for a plausible mapping between two completely common input shapes at arbitrary scales. Our output correspondence in this setting can be coarse as well as dense.

1.3.1.1 Coarse correspondence

For coarse correspondence between jointly sampled feature vertices, our preliminary contribution, that is accepted by IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010, is based on greedy optimization of our isometric distortion function [22]. This optimization mechanism improved within EM (Expectation-Maximization) framework and coupled with a more sophisticated sampling scheme leads to an extension work accepted by IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 2012 [23].

1.3.1.2 Dense correspondence

For dense correspondence between all vertices, our EM-based coarse correspondence algorithm does not scale well due to computational burden that becomes apparent as the cardinality of the mapping set approaches thousands. An alternative coarse-to-fine (C2F) strategy that replaces the user-defined sampling distance parameter of the former with patch-based subdivisions captures level of details not only efficiently but also fully automatically. With these motivations, we present our novel C2F dense correspondence algorithm in Eurographics Symposium on Geometry Processing (SGP) 2011 [24].

Since the goal of [24] is to achieve a dense correspondence, the focus is rather on computational efficiency, and hence the method is less accurate than [23] in achieving sparse correspondence. Moreover, the dense correspondence method in [24] severely suffers from symmetric flips due to initial coarse sampling problem which has then been addressed by the tracking mechanism introduced in our extension work to be published in Computer Graphics Forum (CGF) 2012 [25].

1.3.2 Partial correspondence

Partial correspondence solutions, in the most general setting, seek for a plausible mapping between two shapes with multiple common parts as well as uncommon parts. In a restricted setting, partial correspondence reduces to part matching where one of the two shapes to be matched is an isometrically deformed part of the other. For both settings, shapes come at arbitrary scales and the resolution of the output correspondence is generally coarse although we manage to establish correspondences that are partial and dense at the same time. Note also that, in general, if a method can find solutions for the partial case, it will also be able to handle the complete correspondence case naturally, which applies to our partial matching algorithms as well.

Bringing two shapes to the same scale is the essential problem to be handled before performing the partial matching. To this end, we first explore scale normalization issue and propose an algorithm based on our novel scale-invariant isometric distortion measure as described in the paper to appear in the Pacific Conference on Computer Graphics and Applications (Pacific Graphics) 2012 [26].

With shapes that have multiple common parts at arbitrary scales as well as uncommon parts, we propose a more involved algorithm that we call rank-and-vote-and-combine (RAVAC) with the idea of collecting partial isometry cues from the given shapes by considering all possible partial mappings (relations) between shape extremities and accumulating the collected information into a vote matrix to be analyzed for the overall optimal partial correspondence. A journal paper describing our RAVAC algorithm is currently under preparation to be submitted to a journal [27].

1.4 Putting It All Together

We wrap up all of our shape correspondence algorithms reviewed above in Table 1.1 and release their source codes and executables to public through author's current web site.

1.5 Organization

The rest of this thesis expands our works referenced above with a corresponding literature review, solution, and discussion of comparative results, as well as future research directions and conclusions, hence touching various aspects of 3D isometric shape correspondence problem, which are complete and partial correspondences at coarse and dense resolutions. Preceding that, related concepts and numerical tools some of which form the foundations of our (and many other) correspondence algorithms are investigated.

Input scenario	Output resolution	Solution paradigm	Computational complexity	Publication
Isometric or nearly isometric	Coarse	Greedy optimization	$O(NV \log V)$	CVPR [22]
Isometric or nearly isometric	Coarse	Greedy optimization and EM algorithm	$O(NV \log V)$	PAMI [23]
Isometric or nearly isometric	Coarse or dense	Combinatorial	$O(V \log V)$	SGP/CGF [24]
Isometric or nearly isometric	Coarse or dense	Combinatorial (with symmetric flip care)	$O(V \log V)$	CGF [25]
Isometric or nearly isometric or partially isometric	Coarse or dense	Combinatorial (part matching)	$O(\binom{N}{M} M! M^3)$	PG/CGF [26]
Isometric or nearly isometric or partially isometric	Coarse or dense	Combinatorial (most general setting)	$O(N^3 V \log V)$	to be submitted [27]

Table 1.1: Summary of our algorithms that handle various aspects of 3D isometric shape correspondence. V is the number of vertices in the original mesh, $N \ll V$ is the number of samples at coarse resolution, and M is the cardinality of the sample subset which is 5 in all of the corresponding experiments. [25] is in revision cycle, [27] is to be submitted.

Chapter 2

RELATED WORK AND PRELIMINARIES

In this chapter, we first categorize the 3D shape correspondence algorithms along with the basic references to the related work (Section 2.1) which will be populated more in the following chapters. We then explore the concepts and computational tools that are closely related to the 3D shape correspondence problem. Our correspondence algorithms, as well as most of the others, benefit from some subset of the material discussed in this chapter (Sections 2.2–2.5.2.3).

2.1 Classification of Correspondence Methods

3D shape correspondence methods can be grouped as non-isometric and isometric according to the type of deformations the input shapes are exposed to. In case of isometric deformations, the distance between any two points on one shape is expected to be preserved such that the distance between their images on the other shape is, at least roughly, the same. For non-isometric inputs, on the other hand, lack such distance preservation property. Whether isometric or not, a correspondence method is expected to produce output at coarse and dense resolutions where a subset or the full set of the vertices are matched for the former and the latter, respectively.

2.1.1 Non-isometric methods

The methods that address non-isometric shape correspondence rely on local shape similarity information using shape descriptors [28, 19, 29, 30]. Although local shape similarity is fast to measure and an important clue for non-isometric shape correspondence, it is considered as less reliable than global shape information such as isometry. The methods which rely only on local geometric information may not perform well when the shapes to be matched exhibit large variations in their local geometry, or may easily confuse surface parts when there are many points that are locally similar.

2.1.2 Isometric methods

Isometric shape correspondence methods benefit from isometry information as a global similarity measure in addition to the local similarity information that still applies in this setting. For isometric methods, local shape descriptors may still come handy in the preprocessing step where the samples to be matched need to be selected consistently [23, 24, 25, 27, 31, 21, 32] and/or in the optimization process where the main pairwise geodesic or diffusion-based distance consistency condition is backed up with pointwise descriptor terms [33, 34, 35, 36, 37]. There also exist methods that completely discard local shape descriptors to proceed according to isometric clues only [22, 26, 38, 39]. All of our algorithms fall into this category, hence respecting geodesic consistency in the maps to be produced.

2.2 Shape Classes

In computer graphics and vision applications, the surface of a shape is represented either explicitly as a polygon mesh or disconnected set of points, or implicitly as an isosurface to be extracted from a signed distance function stored in a grid of specified resolution. While we work on explicit triangulations of two-manifolds embedded in 3D Euclidean space, for a 3D shape correspondence application, more important than the representation is the range of the transformations a shape can admit, which can be rigid or non-rigid.

2.2.1 Rigid shapes

Rigid shapes are exposed to distance-preserving rigid transformations which are translation, rotation, and reflection. Due to low degree of freedom, which is 2 translation plus 1 rotation plus 2 reflection axes in 2D plane and similarly 6 in total in 3D space, it is generally easier to register/align two rigid shapes compared to the non-rigid case; one needs to resolve only the translation ambiguity and rotation ambiguity between shapes (Figure 2.1).

Translation disambiguation is handled by simply translating the objects such that their center of masses coincide at the origin, whereas rotation ambiguity can be resolved by Principal Component Analysis (PCA) on covariance matrix $\mathbf{C}$ that encodes variances between

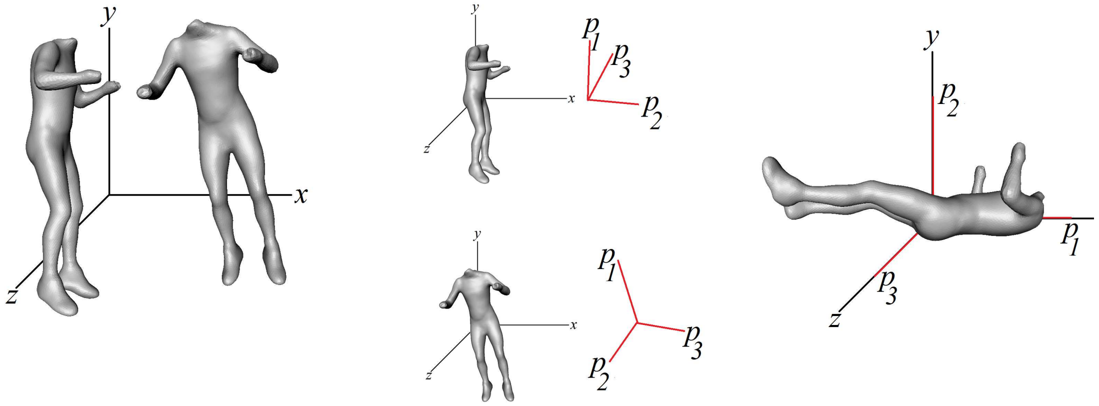

Figure 2.1: Rigid shapes that differ by translation and rotation (left). Translation disambiguation (middle) and rotation disambiguation (right) to perfectly align the two.

x, y, and z coordinate pairs of all n shape points.

$$C = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i z_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i z_i \\ \sum_{i=1}^{n} x_i z_i & \sum_{i=1}^{n} y_i z_i & \sum_{i=1}^{n} z_i^2 \end{bmatrix}$$

The eigenvectors of $\mathbf{C}$ provide principal axes/directions of the shape, which are then aligned with the standard Euclidean axes. Same alignment applied to the other shape disambiguates rotation between two rigid inputs. Our works in [22][23] perform this rigid alignment on isometry-invariant spectral embeddings of the input shapes. The scale invariance, if necessary, is then achieved by setting the average Euclidean distance from the set of vertices to coordinate origin to the same value for both shapes.

PCA-based solution to rigid registration problem does not work when the surfaces are given only partially, a more common scenario that, for example, arises in a range scan of a 3D object as each of multiple scans has missing parts due to occlusion (Figure 2.2). Iterative closest points (ICP) algorithms comes handy at this time since they cast the problem as surface-to-surface distance difference minimization task. There are variants distinguished by the way surface-to-surface distance measured and/or the numerical minimization method preferred, both of which eventually produces desired rotation and translation to be applied to one shape for perfect alignment with the other [5][40][41][42][43]. Correspondence between shape points are extracted trivially once the shapes are rigidly aligned.

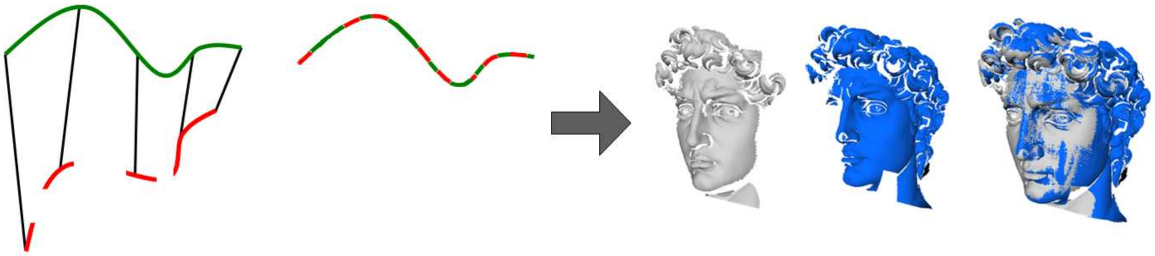

Figure 2.2: Generic ICP algorithm aligns two rigid shapes given partially. Surface-to-surface distance minimization guided by black correspondences (left) and an example of a final registration (right, taken from [42]).

2.2.2 Non-rigid shapes

A more involved yet realistic approach to represent real world objects in digital world brings non-rigid objects which may admit bending, folding, stretching, and scaling in addition to the rigid transformations (Figure 2.3). Other than stretching, to which we tolerate to some short extent, all of our isometric correspondence algorithms are invariant to these non-rigid transformations that preserve pairwise geodesic distances on shapes. Note that, although scaling does not preserve distances in its original form, our appropriate scale normalization methods produce normalized distances with which scale-invariance is achieved [22][24][23][25]. For partial matching where normalization is not trivial, we still enforce isometry by proper choice of scale-invariant isometric distortion functions [26][27].

While non-rigid correspondence can easily be inferred from non-rigid alignment that brings input shapes to the final common pose [44][32], there exist many other ways to compute correspondences which is crucial for various follow-up applications such as mesh morphing, keyframe animation and attribute transfer fed by objects most likely in non-rigid poses. We discuss various non-rigid correspondence methods in the following chapters.

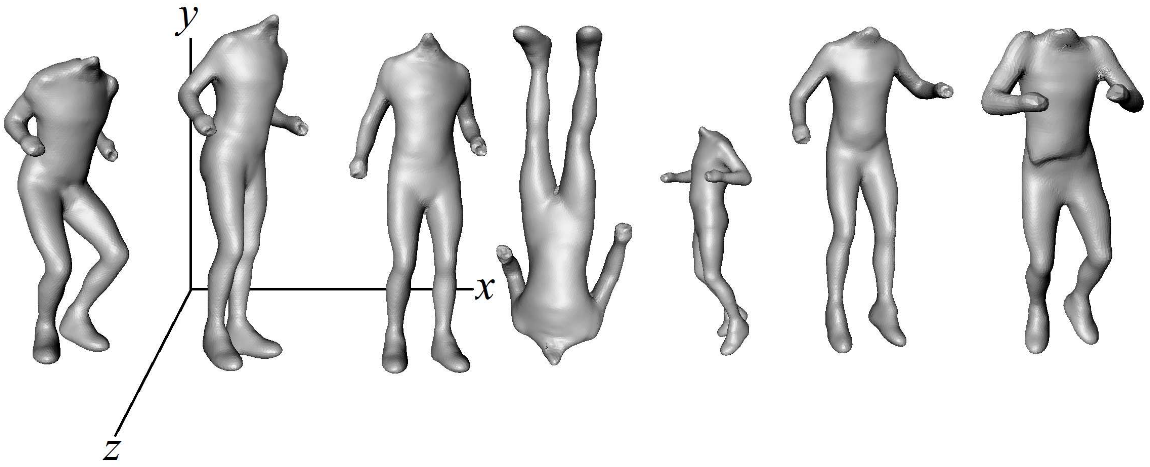

Figure 2.3: Non-rigid shapes from the same mesh sequence exhibiting translation, bending, rotation, reflection, scaling, and stretching (artifacts around belly at the rightmost pose).

2.3 Shape Similarity

In this section, we review important computational tools that can be used to perform pairwise similarity comparisons between isometrically deformed non-rigid shapes which in turn enables correspondence computation in between. The main theme is to develop isometry-invariant local shape descriptors and global metrics upon the key observation that similar shapes possess similar first-order descriptors between the corresponding points and similar second- or third-order metrics between the corresponding pair or triplet of points. The former, in other words, deals with local similarity whereas the latter addresses the global similarity, and their combination is analyzed to decide the similarity measure based on which one can compute a plausible correspondence that couples similar or semantically equivalent surface points of the given shapes.

2.3.1 Global similarity: distance metrics

2.3.1.1 Isometry

Isometry is an important global shape information that is defined as distance-preserving mapping between two metric spaces each equipped with a pair of a point set and an appropriate metric between points that are invariant under the transformations the spaces can be exposed to. If two metric spaces $(V, d_V(,))$ and $(U, d_U(,))$ are perfectly isometric, then a correspondence algorithm exploits that fact by favoring a mapping $f : V \rightarrow U$ such that $d_V(v_i, v_j) = d_U(f(v_i), f(v_j))$ for all vertex pairs. However, two digital shapes are hardly ever perfectly isometric, even for different poses of a rigid object, due to imperfections of the modeling process and/or geometry discretization errors. Hence the goal of isometric correspondence methods existing in the literature and proposed in this thesis is rather to find a mapping that minimizes the amount of deviation from isometry.

Isometry type of a given mapping can be labeled by means of Lipschitz continuity constant $C = \frac{d_U(f(v_i), f(v_j))}{d_V(v_i, v_j)}$ which returns $C \leq 1$, $C < 1$, and $C = 1$ for nonexpanding map, contraction map, and perfectly isometric map, respectively.

We describe in the sequel several well-known distances that are used to define isometry between deformable shapes. The desired properties of such a distance are that it is a metric (non-negative, symmetric, satisfies triangle inequality and indistinguishability), invariant to isometric deformations, computationally inexpensive, insensitive to noise and small topology changes, and free of any parameter that must be set differently for specific meshes or applications.

2.3.1.2 Geodesic distance

Geodesic distance is a metric defined between a point pair as the length of the shortest path along the surface, hence not as relaxed as the Euclidean distance that is allowed travel through anywhere on $\mathbb{R}^m$ (Figure 2.4–left). Such a restriction comes handy for exploring the global intrinsic structure of the non-rigid shapes as geodesic distance is invariant to non-rigid transformations such as bending. The sensitivity to topology changes is the main drawback of this metric that arises due to the consideration of only one path which may severely change by even the slightest cutting or gluing operation. We use geodesic distance g for all of the isometric correspondence algorithms in this thesis.

Shortest paths through faces. When shape is represented explicitly with a polygon mesh, Dijkstra's algorithm [45][46] can be employed to decide the shortest paths which, however, typically pass through faces in the mesh and are therefore not found by the traditional method [45]. We, therefore, use Dijkstra's algorithm, accelerated by Fibonacci heap [47] for

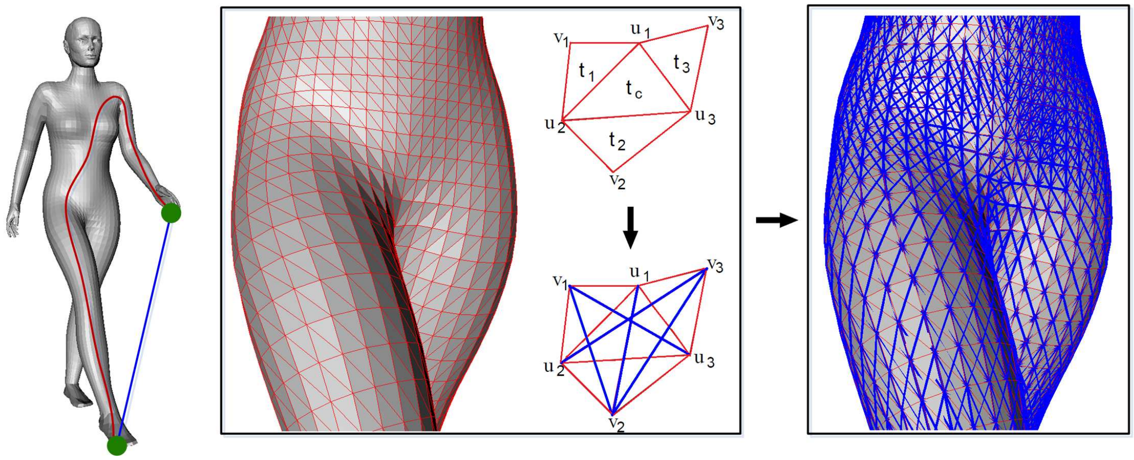

Figure 2.4: Left: Euclidean (blue) vs. geodesic (red) distance between a pair of points on the surface (green). Two boxes at right: Dijktra's shortest paths solution to geodesic distance computation is refined by the introduction of shortcut edges (blue) that permit traveling over mesh faces.

edge selection, with a slight shortcut edge modification for smoothing effect [48]. Movement on shortcut edges in addition to the original mesh edges improves the measurements as they allow traveling on the mesh faces as well (Figure 2.4–right). Triangles adjacent to a centering triangle are unfolded to the plane of the centering triangle, and a shortcut edge is generated if it falls inside the unfolded polygon.

Although Dijkstra's shortest paths algorithm boosted by shortcut edges provides sufficiently accurate geodesic distances efficiently, one can improve results even further with no additional computational cost. Fast marching (FM) method, for this purpose, exploits the roads on mesh faces to the fullest. To this effect, when shortest path is at a particular vertex, marcher checks not only adjacent edges (traditional Dijkstra) but also adjacent faces [49]. Note that, shortcut edges provide a limited version of this effect by allowing face travel starting from a triangle vertex, whereas FM is capable of entering the face from an arbitrary point on the triangle edge, hence increasing accuracy furthermore. Being restricted to triangular faces and requirement of special processing for triangles with obtuse angles are drawbacks of FM over Dijkstra's algorithm which works on any polygonal face. A raster scan algorithm permits parallel process of FM method yielding the fastest geodesic computation time to this date [50]. More accurate solution than that of FM at almost the same speed for triangle meshes is achieved by edge partitioning in [51] whose bottleneck is the intense memory requirement due to large number of windows that represent the partitions.

2.3.1.3 Diffusion distance

Diffusion distance is a metric defined between a point pair by averaging over all paths of length t connecting the two points [52] which in turn renders it more robust to topological noise than the geodesic distance yet less intuitive and accurate due to the spectral embedding involved and the choice of t. To realize this metric, heat diffusion properties on manifolds are exploited in multiscale fashion by defining the heat kernel function $k_t(p, v)$ at different time scales to represent the heat transferred from source p to v in time t, or as an equivalent interpretation, the probability of Brownian motion of heat starting at p to reach v in time t (Figure 2.5):

$$k_t(p, v) = \sum_{i=0}^{\infty} e^{-t\lambda_i} \phi_i(p)\phi_i(v) \tag{2.1}$$

where λ_i and ϕ_i are the eigenvalues and eigenvectors of the Laplace-Beltrami operator (Section 2.5.1.2) on the input shape.

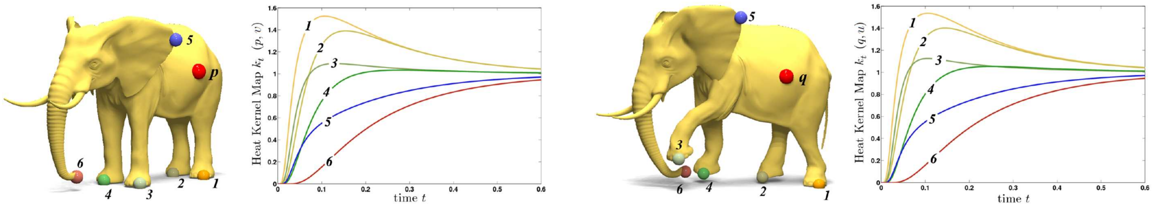

Figure 2.5: Diffusion distances started from point p towards 6 distinct numbered points at different time scales for two shapes. Notice the similar distances between corresponding points under isometric deformations, i.e., isometry invariance. Image taken from [33].

The diffusion distance $d_t(,)$ is then defined using two heat kernels on the input shape S:

$$d_t(p, v) = ||k_t(p, \cdot) - k_t(v, \cdot)||_{L_2} = \int_{s \in S} |k_t(p, s) - k_t(v, s)|^2 ds \tag{2.2}$$

which expands to

$$d_t^2(p, v) = \sum_{i=1}^{\infty} e^{-2t\lambda_i} (\phi_i(p) - \phi_i(v))^2 \tag{2.3}$$

Multiscale property captures local geometry for small t, and the global connectivity/topology for large t, yet the time scale depends on the shape diameter which makes the choice of t problematic.

2.3.1.4 Commute-time distance

Commute-time distance is a metric that is quite similar to diffusion distance except it measures the connectivity of two points by paths of any length, hence eliminating the scale parameter [53][54]. To this end, commute-time distance $d_{\text{com}}(,)$ is computed by

$$d^2_{\text{com}}(p,v) = 2\int_0^\infty d^2_t(p,v)dt = \sum_{i=1}^{\infty} \frac{1}{\lambda_i}(\phi_i(p) - \phi_i(v))^2 \tag{2.4}$$

which essentially sums diffusion lengths of all possible paths between points p and v. Despite being scale-invariant, multiscale property of diffusion distance that describes different properties of the shape at different scales is lost in this distance due to integration over time scale which in turn may fail to realize the fact that two shapes can be similar at small scales and dissimilar at large scales or vice versa.

2.3.1.5 Biharmonic distance

Biharmonic distance is a metric that is related to diffusion and commute-time distances with a slight modification on the eigenvalue normalization [55] as given by

$$d^2_{\text{bi}}(p,v) = \sum_{i=1}^{\infty} \frac{1}{\lambda_i^2}(\phi_i(p) - \phi_i(v))^2 \tag{2.5}$$

which provides a good balance between local and global properties as $1/\lambda_i^2$ decays slow enough to get good local properties around the source point and fast enough to be globally shape-aware in far areas. The usability of this theoretically sound distance, however, has not yet been verified in shape analysis applications domain.

2.3.2 Local similarity: descriptors

Local shape descriptors capture important local shape information by characterizing the surface within a fixed distance about the feature point of interest. By not incorporating any information from far areas, local shape descriptors perform well for matching under non-isometric deformations thanks to the omitted global isometry information, but otherwise it is considered as less reliable especially when the shapes to be matched exhibit large variations or similarities in their local geometry.

We describe in the sequel several well-known local shape descriptors that are used to define isometry between deformable shapes. The desired properties of such a descriptor are that it is discriminative, isometry-invariant, quick to compute, insensitive to noise and small topology changes, and parameter-free.

2.3.2.1 Curvature

Gaussian curvature $\kappa(v)$ measures the deviation of the neighborhood of vertex v from being a flat plane (Figure 2.6–left) and plays an important role for our curvature-oriented evenly-spaced (COES) joint point sampling [23][24][25]. A basic approach for curvature computation is to subtract sum of angles incident to query vertex v from 2π as it approximates the flatness amount, 0 being plain flat. We, on the other hand, use [56] which normalizes the basic result by the adjusted area information from the surrounding obtuse and non-obtuse triangles to increase accuracy. Curvatures satisfy all of the desired properties but stumble on discriminativeness due to small neighborhoods in consideration.

2.3.2.2 Average geodesic distance function

The second and last shape descriptor we employ [24][25] assigns average geodesic distance $\mu(v)$ from v to all other mesh vertices by launching Dijkstra's shortest paths algorithm from uniformly distributed few number of samples $\{s_i\}$, yielding $\mu(v) = \sum_i g(v, s_i) \cdot a(s_i)$ where a is the area of the patch centered by s_i [48]. A good coarse sampling can benefit from μ or its local maxima to detect extreme and/or center vertices because the higher the value of $\mu(v)$ the more extreme the vertex v is, as demonstrated in Figure 2.6–rightmost. Besides, μ satisfies all of the desired properties but topology insensitivity.

2.3.2.3 Spectral signatures

Heat kernel signature [57] is a multiscale descriptor based on eigenfunctions of the Laplace-Beltrami operator (Section 2.5.1.2) and can be interpreted as the probability of random walk of heat to return to the starting point p in different times, that is $K(p) = (k_{t_1}(p,p), k_{t_2}(p,p), .., k_{t_n}(p,p))$. One can relate the heat kernel signature $K(p)$ to curvature as heat is hard to escape from

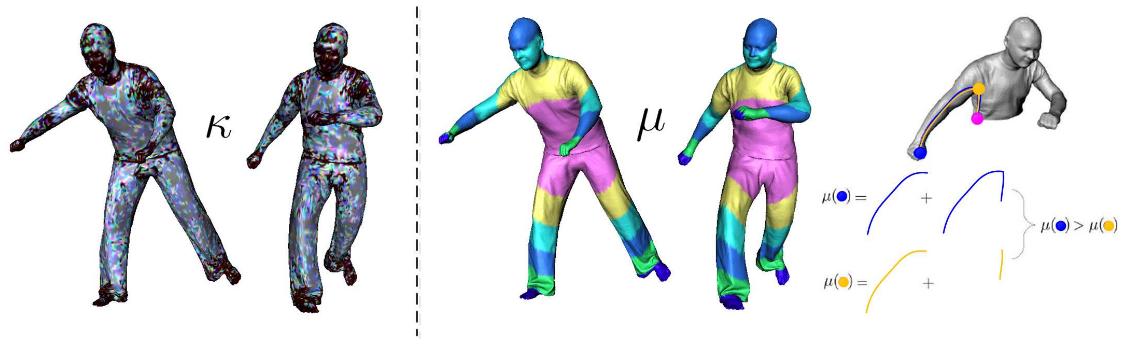

Figure 2.6: Two local shape descriptors employed by our algorithms. Vertices with similar Gaussian curvature (left) and average geodesic distance (right) values are colored the same. Blue vertex at the tip of the shape has a higher average geodesic distance than the yellow one at armpit (rightmost).

high-curvature areas which yields, for example at hands, high values even at large scales. The drawback of the heat kernel signature is its dependence on the shape scale which is addressed in [58] with a scale-invariant version of the signature which however requires setting a time scale parameter that itself depends on the shape scale. The Global Point Signature [59] is based on the same spectral invariant, i.e., $\text{GPS}(x) = (\frac{\phi_1(p)}{\sqrt{\lambda_1}}, \frac{\phi_2(p)}{\sqrt{\lambda_2}}, .., \frac{\phi_i(p)}{\sqrt{\lambda_i}}, ..)$, but severely suffers from sign and order switch of eigenfunctions by treating them individually as vector components unlike $K(p)$ which takes their weighted average via the heat kernel function k_t. All these signatures are invariant under isometric deformations of the shape, yet do not use geodesic distances explicitly which in turn renders them robust against topological noise.

2.3.2.4 Signatures by geometric primitives

There are many descriptors under this category which essentially decompose the spherical space around a feature point into a collection of shells and then the area or volume of the shape intersected by each shell is stored in a histogram bin indexed by radius. The area-based bin values are obtained by adding up the polygon areas within a corresponding shell [60][28] whereas the integral volume descriptor [42] approximates the intersected volume of the shape for vertex v that centers the undecomposed sphere of radius r via

$$V_r(v) = \frac{2\pi}{3}r^3 - \frac{\pi\kappa(v)}{4}r^4 + O(r^5) \tag{2.6}$$

Shape diameter function, as a similar volume-based descriptor, measures the diameter of the shape volume in the neighborhood of the query point by taking weighted average of all conical ray lengths which fall within one standard deviation from the median length [61]. Shape contexts [30], on the other hand, use histograms with log-polar bins to capture the relative distribution of all other points in the plane relative to each feature point on the shape. Since of all these descriptors are invariant only to rigid transformations, they do not match feature points under isometric deformations.

2.4 Shape Sampling

The representative points sampled on the surface of a shape should be dense enough for sufficient coverage, yet sparse enough for computational efficiency. For shape correspondence, one should also perform joint sampling as consistent as possible for computational accuracy.

2.4.1 Uniform sampling

In uniform sampling, triangles are first picked randomly with probabilities proportional to their areas. Random sample points are then generated inside the selected triangles with equal probability per unit area. While this scheme is easy to implement and fast to execute, generated samples may be too close to each other or fail to cover the informative parts of the shape. Another drawback is joint sampling as the potential inconsistency between samples may not permit a plausible mapping even with a robust correspondence algorithm. Comparison of uniform sampling with evenly-spaced sampling (Figure 2.7–left) for surface matching algorithms favors the latter [62].

2.4.2 Evenly-spaced sampling

When performed at a sufficiently high resolution, evenly-spaced sampling produces a consistent joint sampling for correspondence algorithms so that two point sets to be matched involve plausible correspondence candidates.

2.4.2.1 COES sampling

The resolution requirement above can be significantly relaxed by landing samples on salient surface parts with high information content related to the surface geometry and topology, such as curvature, extremities, boundaries, etc. Motivated by this, our curvature-oriented evenly-spaced (COES) sampling mechanism makes a good joint sampler by evenly sampling high-curvature vertices from both shapes, as demonstrated in Figure 2.7–right in comparison with another evenly-spaced sampling method in [48]. In COES sampling, samples are computed by launching the Dijkstras shortest paths algorithm from an arbitrary source vertex. When a sample is selected, all the vertices that are at most r distant from it are marked not to be a future sample. The next highest-curvature sample is then selected from the unmarked vertices. When this is repeated until no unmarked vertex is left, we eventually obtain all samples that are at least r apart from each other [23]. We also use shape extremities as salient vertices in the COES framework to meet our initial matching requirements in [24][25].

2.4.2.2 FPS sampling

Farthest point sampling (FPS) provides almost evenly-spaced sampling the next sample is placed in the center of the largest empty disk on the surface, or circle on the plane for 2D case [63]. The next sample, in other words, is placed at a point that is farthest from the previous samples. To this effect, each candidate vertex first decides the closest existing sample along with the minimum distance. Amongst all these decided distances, the candidate providing the maximum one is then set to be the next sample.

2.4.2.3 Centroidal Voronoi sampling

Unlike the greedy FPS that is unable to undo what was done in previous iterations, centroidal Voronoi sampling reiterates with a new sample set extracted from the intrinsic centroids of the current Voronoi tessellation of the previous sample set [64]. This alternating minimization guarantees more uniformity and less sample resolution compared to FPS in the expense of increased implementation complexity and computational load. The extension of 2D centroidal Voronoi sampling to surfaces in 3D is based on geodesic centroidal tessellation [65].

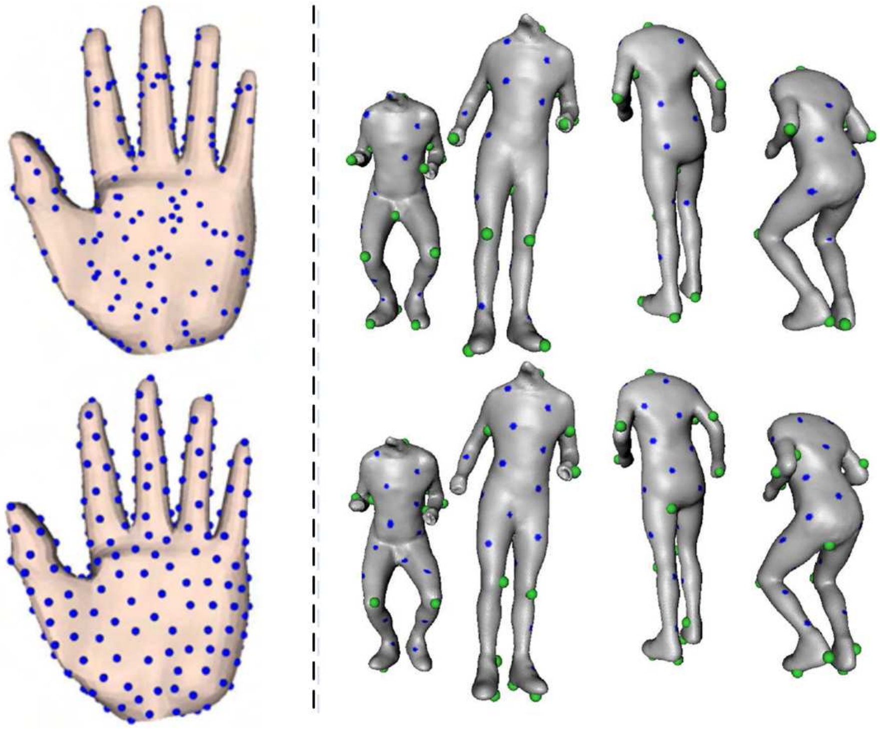

Figure 2.7: Left: Uniform sampling (top) vs. evenly-spaced sampling (bottom), both taken from [62]. Right: Two views from joint sampling of the source and target meshes by our COES sampling (top) vs. by [48] (bottom). Green spheres by COES highlight the first 12 highest-curvature points whose counterparts due to [48] are not that consistent.

2.4.2.4 Stratified sampling

We also mention a voxel-based evenly-spaced sampling method called the stratified sampling [66] which proceeds by first voxelizing the model and then selecting one sample per voxel, restricted to the original model's surface. Close samples are clustered to one in post-processing to achieve evenly spacing. Not only sampling distance, but also the size of the grid for voxelization is a parameter to be set manually. Besides, cubic grid cells may simply not be good enough to generate samples with fine radial isotropy.

2.4.3 Extremity sampling

Since vertices on the extremities or tips of the prominent components provide a good overview of the shape structure, we also discuss sampling methods targeted at these saliencies. To this end, one of our methods applies FPS (Section 2.4.2.2) starting from the most extreme point, i.e., the one with the maximum μ (Section 2.3.2.2), until N well-spread samples are placed where N is kept small not to produce spurious extremes since the purpose is to represent the shape, not all the extremities [26] (Figure 2.8–left).

To hit all, or at least most, of the shape extremities, we slightly modify our other sampler in [27] which also samples central regions that we do not need here. We initialize the sample sets with local maxima of μ which are expected to be on the tips of a given shape [37]. The initial sample sets are then exposed to two steps of pruning, first of which clusters geodesically close samples into the most extreme ones where the closeness threshold is determined based on the maximum geodesic distance $g_{\max}$ on the surface. The second step of pruning removes a local maximum v from the sample set if $\mu(v)$ is less than the average μ to cancel out redundant extremities that are not on tips (Figure 2.8–right).

Sampling in [67] relies similarly on these local maxima vertices but does the pruning based on the convex hull of the MDS embedding of the shape. Finally, we mention [68] that chooses the intersection of two local maxima sets based on the geodesic distance from the candidate vertex to two farthest points where closeness threshold is again required during intersection operation.

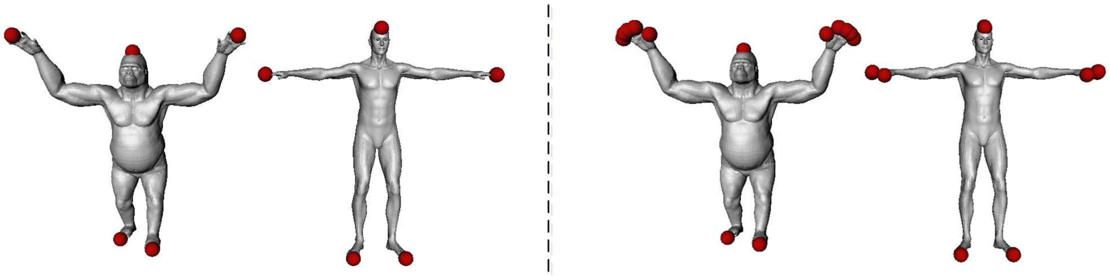

Figure 2.8: Our point samplers that capture representative (left) or all (right) shape extremities on the tips.

2.5 Shape Embedding

In this section, we discuss how the shapes in the original 3D Euclidean space are embedded into a different domain where they gain invariance against isometric deformations. An embedding is Euclidean or non-Euclidean if the distance in the definition of isometry is replaced by the Euclidean distance or non-Euclidean distance, respectively. Isometry is commonly defined with the geodesic distance which is more intuitive and accurate than the diffusion-based counterpart whose main advantage is robustness to topological noise.

2.5.1 Euclidean embedding

When the geodesic distance g on the surface is replaced by the Euclidean distance L_2 in the embedding space, this isometric embedding is impossible to be distortionless, as exemplified in Figure 2.9–left from [18]. Four points on a sphere cannot be isometrically embedded into an Euclidean space of any dimension, hence approximation is a must. After a distortionless, i.e., perfect, isometrical embedding, respective embedded vertices $\hat{v}_{\{1,2,3,4\}}$ has $||\hat{v}_1 - \hat{v}_2||_{L_2} = ||\hat{v}_2 - \hat{v}_3||_{L_2} = 1$ and $||\hat{v}_1 - \hat{v}_3||_{L_2} = 2$, making the triangle $(\hat{v}_1, \hat{v}_2, \hat{v}_3)$ flat. Moreover, $(\hat{v}_1, \hat{v}_4, \hat{v}_3)$ is also flat implying that $\hat{v}_2 = \hat{v}_4$ and consequently $||\hat{v}_2 - \hat{v}_4||_{L_2} = 0$ contradicting the assumption that $||\hat{v}_2 - \hat{v}_4||_{L_2} = g(v_2, v_4) = 1$. Since it is impossible to find a truly isometric, or equivalently distortionless, embedding, a minimum-distortion approximate embedding should be sought which may serve as a valid initialization for an isometric correspondence algorithm [23][39].

In the sequel we observe several embedding spaces on all of which L_2 distance replaces the geodesic distance on surfaces. Note that the embeddings based on eigenanalysis are prone to arbitrary reflections due to sign ambiguities in eigenvectors and eigenfunctions.

2.5.1.1 MDS

A well-known Euclidean embedding is multidimensional scaling (MDS) which comes in classical, least-squares, and landmark forms. All MDS models aim to represent the pairwise (dis)similarity data stored in the affinity matrix as Euclidean distances in a low-dimensional space in order to make these data accessible to visual inspection and further exploration. This mapping from the affinities a_{ij} to the K-dimensional MDS configuration $\hat{\mathbf{V}}$ is achieved by the transformation function $f : a_{ij} \rightarrow d_{ij}(\hat{\mathbf{V}})$, where the particular choice of f specifies the MDS model.

Classical MDS is introduced by Gower [69] in 1966 and used extensively in computer vision/graphics applications such as shape correspondence [22][39], shape representation [9][70], and texture mapping [71]. Classical MDS essentially uses the K leading eigenvectors of the associated geodesic affinity matrix in order to transform the affinities to the K-dimensional configuration $\hat{\mathbf{V}}$ (Figure 2.9–middle). This eigenanalysis leads to a low-dimensional spectral embedding with no danger of getting stuck in local minima, a problem that least-squares MDS model exhibits while minimizing the transformation error via gradient descent or Scaling by Maximizing a Convex Function (SMACOF) optimization algorithms [72][67]. Landmark MDS (LMDS), on the other hand, embeds a large number of points by further approximating the classical MDS. Given the embedded landmark points, LMDS computes embedding coordinates for the remaining data points based on their distances from the landmark points [70] (Figure 2.9–right).

The embedding space for all MDS models is $\mathbb{R}^K$. These embeddings are all geodesic-based which poses the problem of sensitivity to local topology changes as another drawback.

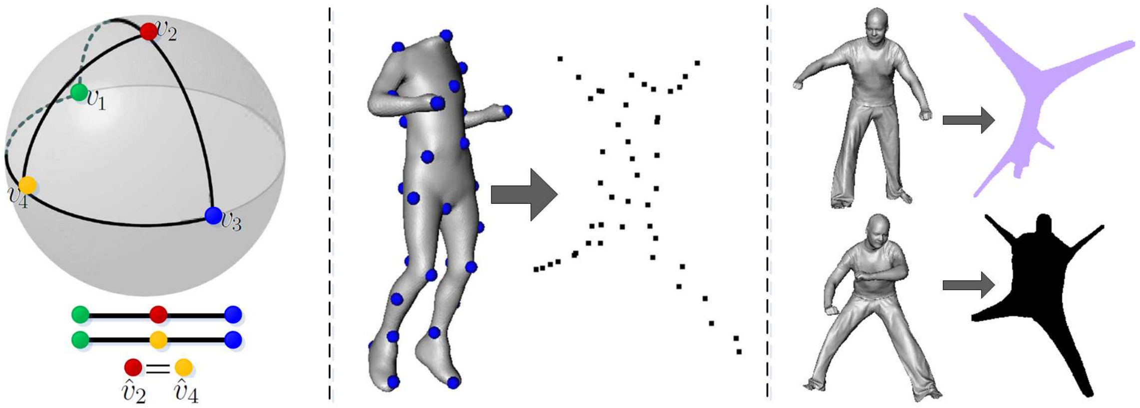

Figure 2.9: Left: Euclidean embedding ambiguously maps two distinct (red and yellow) points on the same location. Middle: Euclidean embedding of samples (blue spheres) into $\mathbb{R}^3$ via classical MDS. Right: Efficient LMDS extension embeds all vertices (top and bottom).

2.5.1.2 Laplacian embedding

The graph Laplacian encodes local geometric and topological properties of a graph or mesh which renders it convenient for describing articulated objects. The most appealing property of the Laplacian is the stability to local topology changes due to its global nature obtained by completely avoiding the use of geodesic distances. There is a combinatorial Laplacian and a discrete Laplacian both based on local neighborhoods which in turn leads to sparse matrices from which extraction of eigenvectors requires lighter computations than MDS guided by dense matrices. Such a spectral analysis on Laplacian matrix results in numerous applications such as natural vertex ordering for mesh streaming [73], calculation of number of spanning trees and connected components [74][75], and deformation-invariant embedding which is our particular interest.

Combinatorial Laplacian. The simplest combinatorial Laplacian is the umbrella operator which brings each vertex to the center of its 1-ring neighborhood via the displacement vector δ_i

$$\delta_i = \frac{1}{\sum_j w_{ij}} \sum_{v_j \in \eta_i} (w_j v_j - v_i) \tag{2.7}$$

is to be added to v_i where η_i is the set of adjacent vertices whose cardinality gives the degree d_i of v_i. Each neighbor can be weighted with $w_{ij} = 1$ for the umbrella effect or with the inverse of the distances between v_i, leading to the Laplacian smoothing operator $\mathbf{L^{smo}}$

$$L_{ij}^{\text{smo}} = \begin{cases} w_{ij}/d_i & \text{if } i \text{ and } j \text{ adjacent}, \\ -1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{2.8}$$

that yields the displacements $\boldsymbol{\delta} = \mathbf{L^{smo}}\mathbf{v}$ to smooth the original vertices $\mathbf{v}$ with. This is, however, a smoothing effect which does not lead to any kind of pose-invariant embedding. To address our isometric embedding demand, one needs to perform spectral analysis on the Laplacian matrix $\mathbf{L}$ defined similarly:

$$L_{ij} = \begin{cases} -1 & \text{if } i \text{ and } j \text{ adjacent}, \\ d_i & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{2.9}$$

that measures, when applied to $\mathbf{v}$, the local smoothness $L(v_i) = d_i v_i - \sum_{v_j \in \eta_i} v_j$ at each vertex with many other variants in the literature [76][77][78]. The eigenvalues and eigenfunctions of $\mathbf{L}$ then give the isometry-invariant embedding.

Discrete Laplacian. By properly adjusting the weights in $\mathbf{L}$ using discrete differential geometry, one not only injects more geometry in combinatorial Laplacian but also decreases the sensitivity to the peculiarities in the triangulation of the input mesh. The resulting discrete Laplacian is linked with the Laplace-Beltrami operator that appears in the wave equation and therefore referred as such [79]. Let a local criterion l be incurring large penalty if neighboring points are mapped apart by the embedding. Applying l to overlapping small neighborhoods and combining the results makes a global criterion that can be captured by the Laplace-Beltrami operator as it intuitively spreads the function it is applied to over small neighborhoods and its eigenfunctions make it possible to understand how these local neighborhoods interact and how global properties emerge from this interaction. To this effect, new $\mathbf{L}$ has the following entries:

$$L_{ij} = \begin{cases} -(\cot \alpha_{ij} + \cot \beta_{ij})/2 & \text{if } i \text{ and } j \text{ adjacent,} \\ \sum_k (\cot \alpha_{ik} + \cot \beta_{ik})/2 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

where the edge (v_i, v_j) is shared by two triangles whose angles facing (v_i, v_j) are α and β [59]. The eigenvalues and eigenfunctions of $\mathbf{L}$ then give the isometry-invariant embedding.

The embedding space for all Laplacian embeddings is $\mathbb{R}^K$. Laplacian embedding, as well as MDS embedding, are also known as spectral embeddings since the embedding coordinates are computed as eigenvectors/functions of some matrix.

2.5.2 Non-Euclidean embedding

A problem common to Euclidean embedding based techniques is that they all produce approximate and/or ambiguous solutions since they can measure deviations from isometry only approximately in the embedding space (Figure 2.9–left), which is mainly because of the line paths replacing the more general geodesic paths. Non-Euclidean embeddings relax this line constraint using more sophisticated paths between embedded vertices with the hope of decreasing the embedding distortion.

2.5.2.1 Generalized MDS

Generalized MDS (GMDS) proposes to embed source shape into the surface of the target shape [80][38], hence geodesic distances are replaced with again geodesics defined on the target surface rather than with line paths over the whole $\mathbb{R}^m$. Since the curved rooms of the target surface is more suitable than a flat space for housing a similar surface to be matched, this embedding generally performs better than the pure Euclidean embedding in the expense of minimization of a non-convex stress function that is difficult and expensive to optimize:

$$\min_{u'_1,..,u'_N \in U} \sum_{i>j} |d_V(v_i, v_j) - d_U(u'_i, u'_j)|^p \tag{2.11}$$

where $\{u'_i\}$ denote surface points which is not necessarily the discrete vertex set or subset of U. The variants of this generalized stress function to be minimized are also used by our methods within more efficient optimizers [22][24][23][25][26][27] and with special care for handling the multiple local minima scenarios that arise due to low distortions attached to true and perfectly symmetric mappings (e.g., full body flip) as well as almost symmetric mappings (e.g., bilateral flip of the legs) [25][26][27]. The non-convex optimization proposed in the original GMDS is, however, based on many geodesic distance approximations and can get stuck in a local minima. Nevertheless, the invaluable idea of embedding one shape directly into the other removes unnecessary representation errors stemming from embedding into an intermediate space.

2.5.2.2 Spherical embedding

Spherical embedding leads to smaller metric distortions compared to Euclidean embedding by changing the embedding domain from flatty $\mathbb{R}^m$ to curvy spherical space which in turn leads to better isometry-invariant representation of surfaces. This embedding parameterizes a triangle mesh onto the sphere [81][82] where parameterization requires assigning each mesh vertex a 3D position on the unit sphere such that the spherical triangles induced by the mesh connectivity are not too distorted and do not overlap. A common way to perform this embedding is to cut the closed genus-0 mesh into two pieces, parameterize each over a planar disk with a common boundary using any planar parameterization method [83][84], and then map each disk to a hemisphere to be merged along the common boundary. This embedding, however, is not popular for shape correspondence problem due to the computational load of arched distances over sphere surface as well as the difficulty of rigid surface matching in spherical spaces.

2.5.2.3 Möbius embedding

Möbius embedding conformally maps shapes with sphere topology into the the extended complex plane $\hat{\mathbb{C}}$ and covers distance-preserving isometries as a subset. When original vertices $\mathbf{V}$ and $\mathbf{U}$ are mapped to $\hat{\mathbb{C}}$ as $\hat{\mathbf{V}}$ and $\hat{\mathbf{U}}$ via, for instance, mid-edge conformal flattening [85], Möbius transformation on two point sets has a closed-form formulated as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \hat{u}_2 - \hat{u}_3 & \hat{u}_1\hat{u}_3 - \hat{u}_1\hat{u}_2 \\ \hat{u}_2 - \hat{u}_1 & \hat{u}_1\hat{u}_3 - \hat{u}_3\hat{u}_2 \end{pmatrix}^{-1} \begin{pmatrix} \hat{v}_2 - \hat{v}_3 & \hat{v}_1\hat{v}_3 - \hat{v}_1\hat{v}_2 \\ \hat{v}_2 - \hat{v}_1 & \hat{v}_1\hat{v}_3 - \hat{v}_3\hat{v}_2 \end{pmatrix} \tag{2.12}$$

given $\hat{v}_{1,2,3} \in \hat{\mathbb{C}}$ and $\hat{u}_{1,2,3} \in \hat{\mathbb{C}}$ as two triplets of correspondences defined by $v_i \to u_i$. The Möbius transformation defined by a, b, c, d then embeds each point $\hat{v}_i$ with a rational linear function $f(\hat{v}_i) = \frac{a\hat{v}_i+b}{c\hat{v}_i+d}$ to $\hat{\mathbb{C}}$ where they gain invariance against isometric deformations and hence become comparable by simple mutually closest point search on L_2 distances [31][36]. This conformal embedding that preserve angles is, however, sensitive to peculiarities of the particular triangulation and restricted to genus zero surfaces.

Chapter 3

COMPLETE SHAPE CORRESPONDENCE

We propose methods to match two semantically similar complete shapes without any missing or extra parts at coarse and dense resolutions in the presence of arbitrary scaling. Our main ingredients for the coarse correspondence are greedy optimization and EM framework [23] (Section 3.1) whereas for the dense matching we perform fast combinatorial matching in coarse-to-fine fashion [24] (Section 3.2) with special care on the multiple local minima threat [25] (Section 3.3).

3.1 Coarse Correspondence of Complete Shapes

In this section, we present our theoretically-sound EM algorithm [23] which extends our preliminary matching algorithm in [22].

3.1.1 Literature review

3D shape correspondence methods aim to find a mapping between the surface points of two given shapes, or more generally, they seek on two given shapes for pairs of surface points that are similar or semantically equivalent [17]. Shape correspondence is a fundamental problem in both computer vision and computer graphics with numerous applications such as mesh morphing [1], mesh parametrization [86], deformation transfer [14], shape registration [6], shape matching [48],[9], analysis of sequential meshes [19] and statistical shape modeling [16]. In this section we address the problem of establishing correspondence between isometric (or nearly isometric) shapes. Isometric shapes appear in various contexts such as different poses of an articulated object, models of a mesh sequence representing the motion of a human actor, or two shapes representing different but semantically similar objects (e.g., two different humans or animals).

If two shapes are perfectly isometric, then there exists an isometry, i.e., a distance-preserving mapping, between these shapes such that the geodesic distance between any two points on one shape is exactly the same as the geodesic distance between their correspondences on the other. However, two digital shapes are hardly ever perfectly isometric, even for different poses of a rigid object, due to imperfections of the modeling process and/or geometry discretization errors. Hence the goal of isometric correspondence methods existing in the literature is rather to find a mapping that minimizes the amount of deviation from isometry. A common strategy to achieve this is to embed shapes into a different (e.g., spectral) domain where geodesic distances are replaced with Euclidean distances so that isometric distortion can efficiently be measured and optimized in the embedding space [9, 39, 87, 88, 72, 77, 22]. However, since the Euclidean embedding process itself introduces a distortion [18], deviations from isometry can be measured only approximately in the embedding space. Hence all these methods mentioned above produce approximate and/or ambiguous solutions, and thus have room for improvement. To remove the approximation error, Bronstein et al. [38] propose to embed one shape to the surface of the other by using generalized multidimensional scaling (GMDS), which however requires minimization of a non-convex function that is difficult and expensive to optimize. Our main contribution is a novel shape correspondence method that minimizes the isometric distortion directly in the 3D Euclidean space, i.e., in the domain where isometry is originally defined, with a computationally efficient algorithm.

Isometry is an important clue for shape correspondence; not only since most real world deformations are isometric, but also because semantically similar shapes have similar metric structures. There are different ways of exploiting isometry for shape correspondence. One way is to minimize the deviation from isometry, indirectly in some embedding space, where Euclidean distances approximate geodesic distances. Euclidean embedding, in the context of shape analysis, can be achieved by using various techniques such as classical MDS (Multi-Dimensional Scaling) [39], [87], [22], least-squares MDS [9], [72], and spectral analysis of the graph Laplacian [77] or of the Laplace-Beltrami operator [89]. Some recent methods propose to use embeddings such as the heat kernel [33] and the canonical diffusion embedding [90], which are invariant under isometric deformations, but which are not Euclidean in the multidimensional scaling sense. Other three recent examples that can be categorized un non-Euclidean embeddings tab and can be applied to only shapes with genus zero, are based on the Möbius transformation which is used for conformal embedding of the given shapes into a canonical coordinate frame on the complex plane where deviations from isometry are approximated based on mutually closest points [31], [36], [21]. A problem common to these embedding-based techniques is that they all produce approximate and/or ambiguous solutions since they can measure deviations from isometry only approximately in the embedding space. In order to eliminate the approximation error, Bronstein et al. [38] propose to embed one of the shapes to be matched into the surface of the other via the generalized MDS, which requires minimization of a non-convex stress function. To optimize this function, they use an iterative gradient-type algorithm which however results in two major drawbacks. First, in order to avoid convergence to local minima, they employ a coarse-to-fine (multiresolution) optimization strategy [38], which may in turn yield artificial symmetric flips in the computed correspondences due to initial coarse sampling. Second, since the algorithm is based on gradient descent, it produces sub-vertex matchings which do not necessarily coincide with the initial sampling, yielding clustered correspondence samples.

The embedding process brings in distortion but reduces the correspondence problem to an alignment or point-to-point matching problem which is easier to solve. Jain and Zhang [39] for example employ the TPS-RPM (Thin-Plate Spline – Robust Point Matching) algorithm of [91] for non-rigid alignment of the embedded points. Other examples are the Hungarian algorithm employed in [72], [22], high-order graph matching in [36], graph-matching based on dual decomposition in [92], Möbius voting scheme in [31], unsupervised point clustering in [77], EM-algorithm in [87, 77], nearest point matching in [33], and association graphs in [90].

Isometric correspondence methods may incorporate local shape information into their schemes. In [31] for example, Gaussian curvature is used to find an initial set of feature points upon which the rest of the correspondence algorithm relies, whereas in [36], the extrinsic curvature and orientation information (including also texture if available) is used to augment the intrinsic global shape (isometric) information. Some methods rely mainly on local shape similarity, using descriptors such as spherical harmonics as in [28], histogram of oriented gradients as in [19], mean curvature as in [29] and shape contexts as in [30]. Local shape similarity is an important clue for shape correspondence, especially in the case of non-isometric deformations, but otherwise it is considered as less reliable than global shape information such as isometry. The methods which rely only on local geometric information may not perform well when the shapes to be matched exhibit large variations in their local geometry, or may easily confuse surface parts when there are many points that are locally similar. Hence some feature-based correspondence search algorithms include also a pruning procedure that takes into account isometric clues by enforcing geodesic consistency [29], [37], [32], [93]. In a recent work, Wang et al. [34] define an objective function in terms of local diffusion information as well as global isometric distortion, that also accounts for shape variability in a probabilistic setting. They formulate the correspondence problem as a graph labeling problem and then solve it by graph matching based on the dual decomposition technique proposed in [92]. Although this is a nice mathematical formulation, the optimization process is computationally very demanding.

Another distinction between shape correspondence methods is whether they target sparse or dense correspondence. Most embedding-based methods naturally support dense correspondence but the computational load is usually a limiting factor that can be overcome with various approximations [21]. There also exist methods which primarily aim to find a small number of feature correspondences [37], [32], [90], [93], [31]. These methods regard the sparse correspondence problem as the main challenge since, based on a sufficient number of reliable landmarks, cross-parametrization [86] or some other form of interpolation technique can always be applied to obtain a dense correspondence [1]. However, in this coarse-to-fine approach, to decide on the degree of sparsity, that would lead to a robust dense correspondence, is always a problematic issue. A particular example to sparse correspondence methods is the deformation-driven approach of [37]. In this method, shape extremities are first determined based on the average squared geodesic distance field. Then an optimal correspondence is sought between these extremities via combinatorial tree traversal by pruning the search space according to some criteria based on local shape similarity and geodesic consistency. For each candidate correspondence set, the source shape is deformed to the target based on these small number of landmarks (anchor points), and the correspondence with the smallest distortion gives the best matching. Another similar deformation-driven method is presented also in [32]. Both methods can handle large deformations but their computational cost is very high due to the repeated deformation process that they involve.

In this section we present a purely isometric method that finds an optimal correspondence between two shapes. We first evenly sample high-curvature vertices from the given shapes and then seek for a minimum-distortion mapping from one vertex set to the other. Hence our correspondence method can be regarded as sparse, though the density of the mapping can be increased as desired by adjusting the sampling rate. Since the mapping is sought between two evenly sampled vertex sets, it becomes possible to formulate minimization of the isometric distortion as a combinatorial optimization problem over the domain of all possible mappings. The resulting optimization problem can then efficiently be solved in the original 3D Euclidean space, as free of embedding errors, by using an expectation-maximization (EM) algorithm that we initialize in spectral domain. In fact our method can also be used to further improve the performance of any isometric correspondence method existing in the literature. A preliminary version of our method in this section was presented in [22], which we substantially improve mainly by solving the combinatorial optimization problem in a probabilistic setting via EM algorithm. We also note that we have very recently proposed a dense shape correspondence method [24] (Section 3.2), that minimizes the isometric distortion via combinatorial matching in a coarse-to-fine fashion. Since the goal of [24] is to achieve a dense correspondence, i.e., a matching between all vertices of two given high resolution meshes, the focus is rather on computational efficiency, and hence the method is less accurate than the one we present in this section in achieving sparse correspondence. Moreover, the dense correspondence method in [24] suffers from symmetric flips due to initial coarse sampling like the multiresolution GMDS method [38].

3.1.2 Problem formulation and overview

We address the problem of 3D shape correspondence between two (nearly) isometric shapes. We assume that each shape is represented by a manifold surface mesh of sufficiently high resolution, on which geodesic distances can easily be computed. We designate one of the shapes as *source* and the other as *target*. Let S and T denote the two sets of points sampled uniformly on the source and the target, respectively. Since the points are evenly sampled over mesh representations, we refer to them as *base vertices* (Section 3.1.3). The problem then reduces to searching for an optimal mapping from the base vertex set S to T. Note that one can find more than one optimal correspondence for symmetrical objects. We require the optimal correspondence (mapping) to have two properties; 1) to be as complete and one-to-one as possible (note that $|S| = |T|$ only in the case of perfect isometry), and 2) to minimize the deviation from isometry, i.e., the isometric distortion function defined below:

$$D_{\text{iso}}(\S) = \frac{1}{|\S|} \sum_{(s_i,t_j)\in\S} d_{\text{iso}}(s_i, t_j) \tag{3.1}$$

where $d_{\text{iso}}(s_i, t_j)$ is the contribution of the individual correspondence (s_i, t_j) to the overall isometric distortion:

$$d_{\text{iso}}(s_i, t_j) = \frac{1}{|\S'|} \sum_{(s_l,t_m)\in\S'} |g(s_i, s_l) - g(t_j, t_m)| \tag{3.2}$$

where $g(.,.)$ is the geodesic distance between two vertices on a given surface and correspondence list $\S' = \S - \{(s_i, t_j)\}$ in the most general setting, that is, unless stated otherwise. Both d_{iso} and D_{iso} take values in the interval $[0, 1]$ since the function g is normalized with respect to the maximum geodesic distance over the surface. Note that Eq. 3.1 can be seen as a variant of the generalized stress function defined in [38], that we compute between evenly sampled base vertices. Given S and T, finding the optimal correspondence $\S^*$ that minimizes the isometric distortion is actually a combinatorial problem, which we solve in a probabilistic framework using EM algorithm (Section 3.1.4). We assume that the probability of a base vertex s_i being in correspondence with t_j can be defined in terms of isometric distortion. Given these probabilities, the optimal correspondence can be estimated by maximizing a likelihood function. The probability values, hence the EM algorithm, are initialized based on the Euclidean distances between the vertices embedded into spectral domain through classical MDS (Section 3.1.5). Maximization of the likelihood function, hence minimization of the isometric distortion given in Eq. 3.1, is then achieved in the original 3D Euclidean space, for each iteration of the EM algorithm, in two steps; by using first a bipartite graph matching algorithm (Section 3.1.6.1), and then a greedy optimization technique (Section 3.1.6.2), both with polynomial time complexity (Section 3.1.7). Note that neither the number of vertices of the original source and target meshes nor $|S|$ and $|T|$ need to be exactly equal. The block diagram of the overall shape correspondence scheme is given in Figure 3.1. We demonstrate the performance of our method with experiments in Section 3.1.8, where we also compare our algorithm with two state-of-the-art techniques: the spectral method of Jain and Zhang [39] and the GMDS of Bronstein et al. [38].

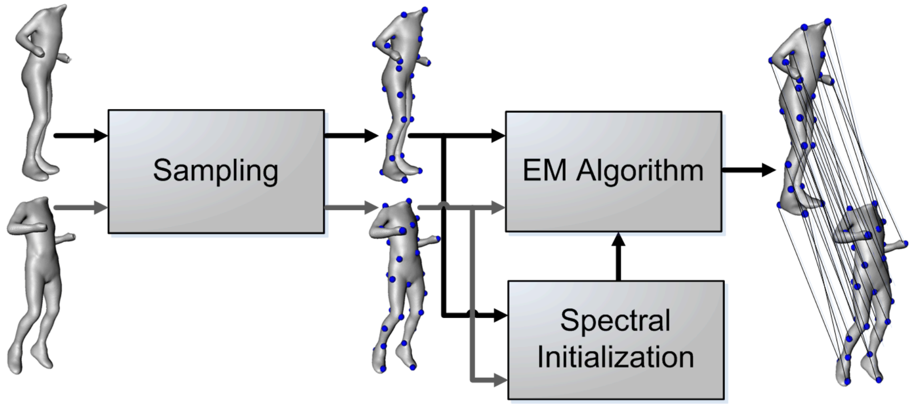

Figure 3.1: Overall isometric shape correspondence scheme.

3.1.3 Sampling

The sampling of base vertices should be as consistent as possible between source and target meshes. To achieve this goal, we sample each mesh separately, but imposing a uniformity constraint and favoring the vertices which are geometrically salient. Our sampling algorithm is as follows. We initially compute the Gaussian curvature at each vertex of the original mesh [56]. We then sort the vertices into a list in descending order with respect to their curvature values and select the top vertex of the list as the first base vertex. We launch the Dijkstra's shortest paths algorithm from this vertex and mark all the vertices lying within a distance r. The next base vertex is then picked as the first unmarked vertex in the list. When this is repeated until no unmarked vertex is left, we obtain a sampling of the surface, where the base vertices are at least at distance r apart from each other.

The sampling algorithm described above is the same as the one proposed by Hilaga et. al in [48] except that the next base vertex is selected arbitrarily from unmarked vertices in [48], whereas we select it as the unmarked vertex with the highest curvature. Hence our algorithm places the base vertices on local maxima of the Gaussian curvature and thereby generates a more consistent sampling on two shapes, which eventually improves the correspondence performance and yields more intuitive matchings. As verified in our experiments, this vertex sampling heuristic, as we call it the curvature-oriented evenly-spaced (COES) sampling, provides a good start for our algorithm. The resolution of the sampling, hence the density of the ultimate correspondence, can easily be set thanks to the intuitive parameter r, the *sampling distance*, as we will refer to it. Although COES sampling algorithm cannot in general guarantee the same sampling on two shapes, the parameter r can be set to a sufficiently small value so that all critical vertices of the given meshes (for typical shapes) are included by the sampling process. Moreover, the sampling distance r puts an upper bound to the isometric distortion of the optimal correspondence $\S^*$ that minimizes Eq. 3.1. One such upper bound is given by $D_{\text{iso}}(\S^*) \leq r$ in the case of perfect isometry.

We note that there exist in the literature several other techniques for uniform mesh sampling such as the farthest point sampling technique [94], which is commonly employed by shape correspondence algorithms such as in [38, 31]. Our COES algorithm is rather built upon the sampling method of [48] since it provides us with a convenient framework to develop an efficient saliency-based uniform sampling algorithm.

3.1.4 EM framework

Minimization of the isometric distortion in Eq. 3.1 can be formulated in a probabilistic setting as the following log-likelihood maximization problem:

$$\S^* = \arg\max_{\S} \log \mathrm{P}(\S|\mathcal{X}, \mathbf{Q}) \tag{3.3}$$

where $\mathcal{X} = (S, T)$ is the observed data, i.e., the base vertices, and $\mathbf{Q}$ is the matrix with entries q_{ij}, each representing the probability of source base vertex s_i being in correspondence with target t_j, such that $\sum_j q_{ij} = 1$. Note that since we seek for an (almost) complete correspondence, we require every $s_i \in S$ to be matched with exactly one $t_j \in T$.

The likelihood maximization problem given in Eq. 3.3 can be solved by using the EM algorithm [95]. Let $\S^{(k)}$ and $\mathbf{Q}^{(k)}$ denote the estimates for $\S$ and $\mathbf{Q}$ at iteration k. Starting from an initial estimate $\mathbf{Q}^{(0)}$, the EM algorithm alternately recomputes the expected value of $\mathbf{Q}$ and the estimate $\S^{(k)}$ through the following E-step (for Expectation) and M-step (for Maximization):

- E-step: $\mathbf{Q}^{(k)} = E(\mathbf{Q}|\mathcal{X}, \S^{(k-1)})$
- M-step: $\S^{(k)} = \arg\max_{\S} \log \mathrm{P}(\S|\mathcal{X}, \mathbf{Q}^{(k)})$

until convergence.

We assume that, for each base vertex s_i, the probability of correspondence with t_j can be modeled as a function of the resulting isometric distortion via the following Gibbs distribution:

$$q_{ij} = \mathrm{P}(t_j|s_i) = \frac{1}{T_i} e^{-\beta d_{\text{iso}}(s_i,t_j)} \tag{3.4}$$

where β is a fixed positive factor that determines the sharpness of the distribution, and T_i is a normalizing constant to be chosen such that the constraint $\sum_j q_{ij} = 1$ is satisfied for all i. The term $d_{\text{iso}}(s_i, t_j)$ is the isometric distortion due to the correspondence of s_i with t_j (see Eq. 3.2), and its expected value at iteration k can be estimated by averaging over all the correspondence pairs in $\S^{(k-1)}$:

$$\begin{aligned} d^{(k)}_{\text{iso}}(s_i, t_j) &= E(d_{\text{iso}}(s_i, t_j)|\mathcal{X}, \S^{(k-1)}) \\ &= \frac{1}{|\S^{(k-1)}| - 1} \sum_{\substack{(s_l,t_m)\in\S^{(k-1)} \\ (s_l,t_m)\neq(s_i,t_j)}} |g(s_i, s_l) - g(t_j, t_m)| \end{aligned} \tag{3.5}$$

which can then be used to compute $q^{(k)}_{ij}$ in E-step:

$$q^{(k)}_{ij} = E(q_{ij}|\mathcal{X}, \S^{(k-1)}) = \frac{1}{T_i} e^{-\beta d^{(k)}_{\text{iso}}(s_i,t_j)} \tag{3.6}$$

Hence in M-step, (assuming statistical independence for assignments of s_i) we have

$$\S^{(k)} = \arg\max_{\S} \log \prod_{(s_i,t_j)\in\S} q^{(k)}_{ij} \tag{3.7}$$

and substituting Eq. 3.6, we get

$$\S^{(k)} = \arg\max_{\S} \textstyle\sum_i \log \frac{1}{T_i} - \beta \displaystyle\sum_{(s_i,t_j)\in\S} d^{(k)}_{\text{iso}}(s_i, t_j) \tag{3.8}$$

Here the first summation is constant, hence can be ignored. Since the second term is always negative, using Eq. 3.5, the maximization in Eq. 3.8 becomes equivalent to the following minimization problem:

$$\begin{aligned} \S^{(k)} &= \arg\min_{\S} \sum_{(s_i,t_j)\in\S} d^{(k)}_{\text{iso}}(s_i, t_j) \\ &= \arg\min_{\S} \frac{1}{|\S|} \sum_{(s_i,t_j)\in\S} \frac{1}{|\S^{(k-1)}| - 1} \sum_{\substack{(s_l,t_m)\in\S^{(k-1)} \\ (s_l,t_m)\neq(s_i,t_j)}} |g(s_i, s_l) - g(t_j, t_m)| \end{aligned} \tag{3.9}$$

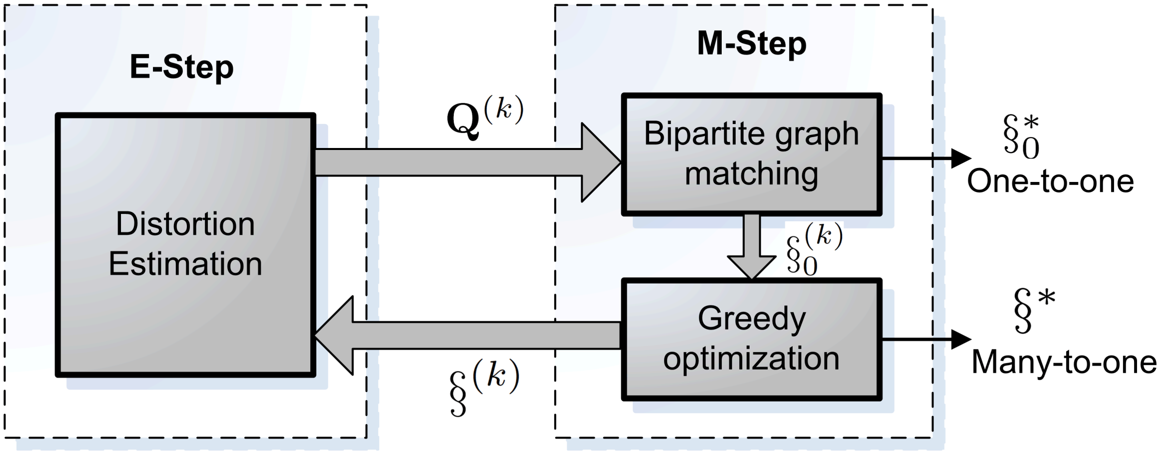

Figure 3.2: EM Algorithm.

Hence starting from an initial estimate $\mathbf{Q}^{(0)}$, or equivalently $\{d^{(0)}_{\text{iso}}(s_i, t_j)\}$, the EM algorithm repeatedly computes Eq. 3.2 based on $\S^{(k-1)}$ and minimizes the overall isometric distortion Eq. 3.1 to generate $\S^{(k)}$ (see Fig. 3.2). The EM algorithm is expected to converge to a local minimum [95], as also verified by our experiments presented in Section 3.1.8.

3.1.5 Initialization

In this section, we describe how we initialize the EM algorithm, that is, how we obtain good initial estimates for the entries of the probability matrix $\mathbf{Q}$, or equivalently for the isometric distortions $\{d^{(0)}_{\text{iso}}(s_i, t_j)\}$, defined in the previous section.

3.1.5.1 Spectral embedding

We start by computing the geodesic distances between all pairs of bases by running the Dijkstra shortest paths algorithm from each base vertex. These pairwise distances, when exposed to an exponential kernel, form a geodesic affinity matrix, $A_{ij} = \exp(-g^2(i,j)/2\sigma^2)$ for each of the base vertex sets S and T with $g(.,.)$ being the geodesic distance between two points on a given surface. We set the kernel width σ to be half of the maximum geodesic distance over the surface. Note that computation of the geodesic affinity matrix does not bring any additional load to our overall correspondence algorithm due to the Dijkstra framework that is already prepared for the COES sampling procedure.

Based on the computed geodesic affinity matrices, we transform each of the base vertex sets into an M-dimensional spectral domain by means of the classical MDS algorithm to gain invariance against rigid transformations as well as shape bending. The classical MDS, introduced by Gower [69], essentially uses the M leading eigenvectors of the affinity matrix to obtain spectral embedding. We scale each of these eigenvectors with the square root of the corresponding eigenvalue (ignoring the first eigenvector since it becomes constant due to normalization) as suggested in [39]. The scaled eigenvectors provide us with an M-dimensional spectral embedding of the base vertices, that we will denote by $\hat{S}$ and $\hat{T}$, respectively for the source and the target.

3.1.5.2 Alignment

The geodesic distances between base vertices in the original 3D space approximately correspond to L_2 distances between their M-dimensional embeddings in the spectral domain. Although the same transformation is applied to both shapes, due to arbitrary sign flips of eigenvectors, a disambiguation process is required, which tests the 2^M different possible embeddings for the best alignment. We measure the alignment of each such embedding $\hat{S}_m$ with a fixed $\hat{T}$ by means of the cost $C_m = \sum_{i=1}^{|\hat{S}_m|}(\|\hat{s}_{i,m} - \hat{t}_i\|_2)$, i.e., the sum of L_2 distances between mutually closest points in the spectral domain, where each term in the summation is the distance between some $\hat{s}_{i,m}$ and the target $\hat{t}_i$ which is closest to it. The embedding $\hat{S}_m$ producing the minimum cost C_m aligns best with $\hat{T}$. This alignment operation is visualized in Figure 3.3 for $M = 3$.

3.1.5.3 Isometric distortion

For each possible (s_i, t_j) pair, the isometric distortion $d_{\text{iso}}(s_i, t_j)$ in Eq. 3.2 can be approximated by the L_2 distance between the embedded coordinates $\hat{s}_i$ and $\hat{t}_j$, which provides us with the initial estimate that we seek for:

$$d_{\text{iso}}^{(0)}(s_i, t_j) = \|\hat{s}_i - \hat{t}_j\|_2 \tag{3.10}$$

3.1.6 Optimization

In this section, we describe the M-step of our EM algorithm, which involves minimization of the isometric distortion as given in Eq. 3.9. We solve this optimization problem in two

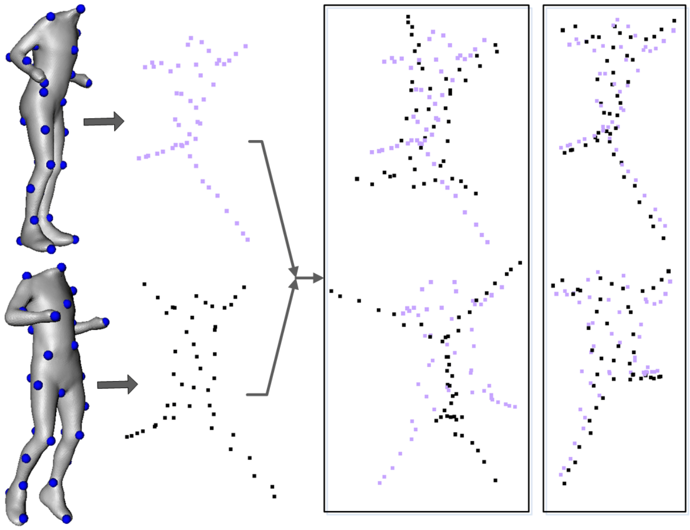

Figure 3.3: Two shapes along with their spectral embeddings (left), the alignments obtained using an arbitrary configuration of the eigenvectors (left box), and using the best aligned embedding (right box). The boxes display two different views for visual convenience.

steps. We first find a one-to-one mapping from S to T, which is close to the global optimum, via bipartite perfect matching, and then in the second step, we refine it locally via a greedy optimization algorithm that relaxes the injectivity constraint and thereby generates a many-to-one mapping. Both optimization steps are carried out in the original 3D Euclidean space.

3.1.6.1 Bipartite graph matching

At each iteration k of the EM algorithm, we create a complete bipartite graph G on which the minimum-weight perfect matching is sought. The base vertices S and T form the disjoint vertex sets of G which is made complete by connecting every vertex of one set to every vertex of the other with edges weighted by $w_{ij} = d_{\text{iso}}^{(k)}(s_i, t_j)$, specifying the isometric cost of matching s_i with t_j. Since the cardinalities of the disjoint sets must match for a perfect matching, if $|S| \neq |T|$, we introduce virtual vertices with connector edges of ∞ weights. Note that the numbers of base vertices are almost equal for a given pair of isometric shapes but need not be exactly the same due to possible deviations from isometry. Hence at the end of the optimization process, some base vertices in T may be left unassigned (we designate source and target so that $|S| \leq |T|$). We employ the efficient Blossom V algorithm [96] to solve the bipartite graph matching problem, hence the minimization in Eq. 3.9. Bipartite perfect matching results in a one-to-one (but not necessarily onto) mapping from S to T, that we denote by $\S_0^{(k)}$.

3.1.6.2 Greedy optimization

The greedy optimization starts with the one-to-one mapping found via bipartite perfect matching of the first step, $\S = \S_0^{(k)}$, and iteratively improves $\S$ by relaxing the injectivity constraint imposed on it. The algorithm traverses the current $\S$ and replaces each time the current pair (s_i, t_j) with some (s_i, t_n) provided that this replacement decreases the isometric distortion. The accumulation of these greedy decisions, each of which considers a local improvement, eventually leads to a local optimum on D_{iso} as we re-traverse $\S$ until convergence, i.e., until D_{iso} no longer improves.

For replacement of (s_i, t_j), the algorithm considers a small set of candidates t_m. This candidate set, denoted by T', is formed by the target base vertices (plus their base neighbors), that have been matched with the base vertices in the closed neighborhood of s_i. For each such t_m, $d_{\text{iso}}(s_i, t_m)$ is computed, and (s_i, t_j) is replaced with the pair (s_i, t_n) yielding the minimum distortion, provided also that this replacement brings in some improvement over the current one, i.e., if $d_{\text{iso}}(s_i, t_n) < d_{\text{iso}}(s_i, t_j)$. In addition to this potential replacement concerning s_i, we also consider the current match s_i^+ of the selected t_n by replacing (s_i^+, t_n) with (s_i^+, t_j) if $d_{\text{iso}}(s_i^+, t_j) < d_{\text{iso}}(s_i^+, t_n)$. Note that the function d_{iso} is evaluated each time using the current correspondence list $\S$. The greedy optimization algorithm is illustrated on an example in Fig. 3.4 and given in pseudocode in Fig. 3.5.

The greedy algorithm relies on the assumption that the initial correspondence found in the first step is reasonably good, which is so thanks to the spectral matching that serves as a good starting point. This assumes that for most of the samples on the source shape, at least a base neighbor is initially matched close to its optimum on the target. For the remaining samples for which this assumption does not hold, EM iterations are expected to resolve the problem as described next.

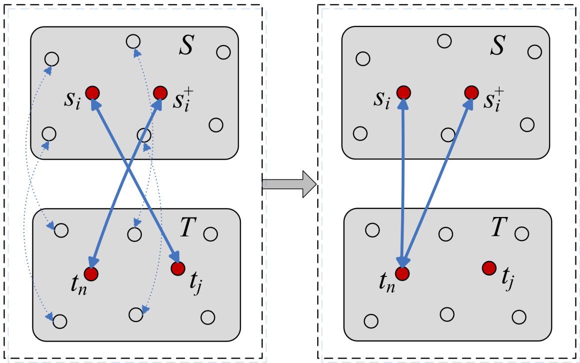

Figure 3.4: Illustration of a local improvement by the greedy algorithm. Suppose each base vertex (filled circles) has 4 neighbors (empty circles) and (s_i, t_n) is the ground-truth correspondence. When $(s_i, t_j) \in \S$ is in process, the target base vertices matched with the base neighbors of s_i (pointed by the dashed arrows) are tested (the base neighbors of these targets are also tested, though not illustrated in the figure). Since the target base t_n with minimum distortion is different than t_j in this case, the correspondence pair (s_i, t_j) is replaced with (s_i, t_n) after checking $d_{\text{iso}}(s_i, t_n)$ vs. $d_{\text{iso}}(s_i, t_j)$. Not shown above, the pair (s_i^+, t_n) is also considered next for its possible replacement with (s_i^+, t_j).

```
Input: $\S_0^{(k)}$, one-to-one, Output: $\S^{(k)}$, many-to-one
$\S = \S_0^{(k)}$
For each pair $(s_i, t_j) \in \S$
    $T' = $ GetCandidateTargets$(s_i, t_j)$
    minD $= \infty$
    For each $t_m \in T'$
        If $d_{\text{iso}}(s_i, t_m) <$ minD          $t_n = t_m$; minD $= d_{\text{iso}}(s_i, t_m)$
    If $d_{\text{iso}}(s_i, t_n) < d_{\text{iso}}(s_i, t_j)$        $(s_i, t_n)$ replaces $(s_i, t_j)$ in $\S$
    Let $s_i^+$ be the match of $t_n$, i.e., $(s_i^+, t_n) \in \S$
    If $d_{\text{iso}}(s_i^+, t_j) < d_{\text{iso}}(s_i^+, t_n)$        $(s_i^+, t_j)$ replaces $(s_i^+, t_n)$ in $\S$
Until convergence (until $D_{\text{iso}}$ no longer improves)
$\S^{(k)} = \S$
GetCandidateTargets(Pair $(s_i, t_j) \in \S$)
$T' = \emptyset$
For each $s_p \in W(s_i)$ where $W(s_i)$ is the closed base neighborhood of $s_i$
    $T' = T' \cup \{W(t_r) \mid (s_p, t_r) \in \S\}$
Return $T'$
```

Figure 3.5: The greedy optimization algorithm.

3.1.6.3 EM iterations

At each iteration k of the EM algorithm, the M-step first generates a one-to-one mapping from S to T via bipartite perfect matching. This mapping, that we denote by $\S_0^{(k)}$, is then iteratively refined by the greedy algorithm so as to produce a many-to-one mapping, denoted by $\S^{(k)}$. At the E-step of the next iteration, the bipartite matching process uses $\S^{(k)}$ to compute estimates for isometric distortion via Eq. 3.5, and thereby generates a one-to-one mapping $\S^{(k+1)}$ in a globally consistent manner. The final output of the EM algorithm at convergence, $\S^*$, can hence be one-to-one or many-to-one upon choice, depending on at which step the algorithm is stopped. We will provide experimental results for both cases in Section 3.1.8.

3.1.7 Computational Complexity

Let V denote the number of vertices in the original mesh (source or target, whichever has more vertices), and N the number of base vertices sampled on the target ($N = |T|$, recalling that $|S| \leq |T|$). The COES sampling process involves curvature-based sorting with $O(V \log V)$ complexity, followed by the actual base vertex sampling procedure in again $O(V \log V)$ time since the shortcut Dijkstra shortest paths for each base sum up to one single Dijkstra algorithm spanning V vertices. The geodesic affinity matrix is computed in $O(NV \log V)$ time. The eigenanalysis for embedding into M-dimensional spectral domain is $O(N^2)$, followed by $O(2^M N^2)$ operations for alignment, where $M \leq 6$. As for the EM algorithm, the greedy optimization demands $O(N^2)$ time since the computation of d_{iso} can be performed in linear time for each pair in $\S$. The bipartite matching part of the M-step employs the Blossom V algorithm with $O(N^2 \log N)$ complexity for a given $N \times N$ cost matrix which is created in the E-step by computing a d_{iso} value for each entry in linear time. The EM framework hence demands $O(N^3)$ work per iteration until D_{iso} converges, which takes no more than 5 iterations in our experiments. Under the valid assumption of $N \ll V$, the overall complexity is then given as $O(NV \log V)$, which can be regarded as quite efficient when the other methods in the shape correspondence literature are considered. For example, this is almost equivalent to the computational complexity of the method presented in [39], but better than GMDS [38] with $O(V^2 \log V)$ complexity, than the method in [31] with $O(N^4 \log N + V^2 \log V)$ complexity and than the method in [37], which involves combinatorial tree traversals and repeated shape deformations.

3.1.8 Experiments

3.1.8.1 Datasets

We have conducted experiments on four different types of datasets. The first consists of two mesh sequences, *Jumping Man* [97] and *Dancing Man* [98], both originally reconstructed from real scenes and each representing the real motion of a human actor. The second dataset is the *Horse Gallop* which is a computer generated synthetical mesh sequence [14]. The original meshes of these sequences are all uniform and given at high resolution with fixed connectivity, hence we have the ground-truth dense correspondences in all three cases. The third dataset is the *Ballerina* from the 3D segmentation benchmark [99], which contains five different poses of the same human model, each represented with a mesh model of arbitrary connectivity. The last dataset is the Nonrigid World shape database [38], which contains various animal and human mesh models with different poses, where each object has approximately 3K vertices with arbitrary connectivity.

3.1.8.2 Evaluation metrics

We measure the performance of our shape correspondence scheme in terms of deviation from ground-truth as well as isometric distortion. To quantify isometric distortion, we use the average distortion measure D_{iso} defined in Eq. 3.1, and also a maximum distortion measure that we denote by $d^{\dagger}_{\text{iso}}$:

$$d^{\dagger}_{\text{iso}} = \max_{(s_i,t_j)\in\S} d_{\text{iso}}(s_i, t_j), \tag{3.11}$$

where d_{iso} is the isometric distortion function given in Eq. 3.2. Similarly we compute average and maximum ground-truth correspondence errors respectively by (whenever the ground-truth correspondence is available),

$$D_{\text{grd}} = \frac{1}{|\S|} \sum_{(s_i,t_j)\in\S} g(t_i, t_j), \tag{3.12}$$

and

$$d^{\dagger}_{\text{grd}}(s_i, t_j) = \max_{(s_i,t_j)\in\S} g(t_i, t_j), \tag{3.13}$$

where each (s_i, t_i) stands for a ground-truth correspondence pair.

N	$D_{\text{grd}}, d^{\dagger}_{\text{grd}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
40	$1.99r', 4.29r'$	$1.34r', 3.13r'$
80	$0.58r', 2.92r'$	$0.56r', 1.85r'$
160	$0.46r', 2.11r'$	$0.42r', 1.28r'$
320	$0.38r', 1.13r'$	$0.31r', 0.84r'$

Table 3.1: One-to-one mapping performance for varying number N of base vertices on *Horse Gallop* sequence, where r' denotes the sampling distance for $N = 80$.

3.1.8.3 Parameter setting

The correspondence that we find between the sampled base vertices can be regarded as sparse, which can though be expanded to a dense correspondence via cross-parametrization or some other form of interpolation techniques [1]. However, it is always an issue to decide on the optimal sampling that would render the dense matching phase as efficient and robust as possible. The related sampling distance parameter r of our algorithm is the only parameter that we set manually. Our current strategy to set this parameter is to select a sufficiently small value so that all the critical vertices of a given mesh for typical shapes are included by our COES sampling procedure.

In Figure 3.6, we demonstrate, on a pair of shapes from the Horse Gallop sequence, a degenerate case when the number of base vertices is significantly low, e.g., $N = 10$. The problems due to such degenerate cases are usually fixed by increasing the base resolution, e.g., to $N = 40$, hence by a decrement on the sampling distance r. The smaller the parameter r, the less the deviation from the ground-truth during base vertex sampling, and hence the more accurate the correspondence obtained, as also verified by the quantitative results given in Table 3.1. In the table, we observe that the isometric distortion and the correspondence error both decrease as the sampling distance gets smaller, in the expense of some execution time: $1.5, 2.1, 4.0$, and 10.5 seconds for $N \approx 40, 80, 160$, and 320, respectively (recall the $O(NV \log V)$ complexity of our algorithm).

3.1.8.4 Results

We display some visual examples from the computed correspondences through Figures 3.7-3.26, and provide quantitative results in Tables 3.1 and 3.2. In all our experiments, the

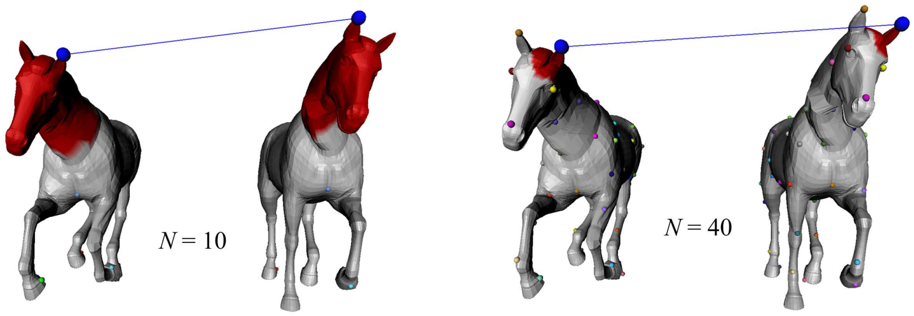

Figure 3.6: Two correspondence results on a pair of meshes from the *Horse Gallop* sequence, with $N = 10$ (left) and with $N = 10$ (right). The horse has two critical vertices on the tips of the ears, sampled alternately on the source and on the target for $N = 10$. Hence it is impossible to obtain the ground-truth left-to-left (or right-to-right) ear match at this resolution. The region painted (in red) on each shape represents the surface patch with radius r around the sampled (blue) base vertex. The mismatch problem is resolved when the number of base vertices to be matched is increased to $N = 40$.

dimension of the spectral domain is set to be $K = 6$, and the number of base vertices is about 80 unless stated otherwise.

In Fig. 3.7, we display one-to-one matching results on a shape pair from the Horse Gallop sequence for varying resolutions, hence for varying N. We observe that, while the obtained correspondences satisfactorily match the ground-truth at all resolution levels, the matching precision increases as the number N of base vertices increases. In Figures 3.8 and 3.9, we provide the visual correspondence results obtained on sample shape pairs, respectively from the Dancing Man and Ballerina sequences. Fig. 3.10 displays the correspondence results across two mesh sequences, i.e., on two shapes representing two different humans, which are hence only nearly isometric. We observe that the one-to-one mapping obtained is very accurate despite the missing head of the Jumping Man. In all these figures, two base vertices with the same color indicate a correspondence pair, whereas the worst matchings, with respect to isometric distortion and ground-truth correspondence error (whenever available), are highlighted with bold red and green lines, respectively. Note also that some target bases may not be drawn in the figures if not matched with any source base vertex.

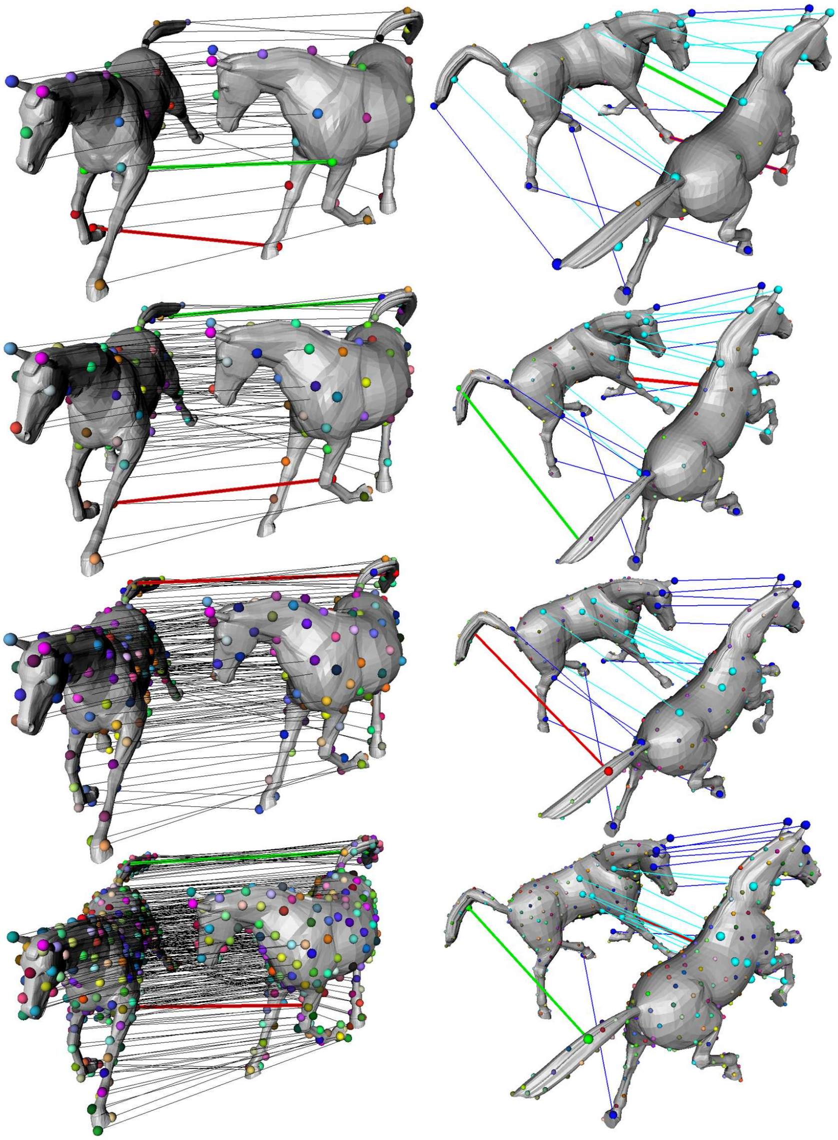

Figure 3.7: (From top to bottom) Correspondence results on *Horse Gallop* with $N = 40, 80, 160, 320$, where each row displays the shape pair from two different views. Bold green and red lines indicate the worst matches with respect to ground-truth error and isometric distortion, respectively. Note that the two worst matches coincide for $N = 160$, where red overwrites green. (Left) All available correspondence pairs, each indicated with a line segment drawn between two spheres of the same color at both ends, (right) best-10 matches with respect to isometric distortion and 10 matches for top-10 vertices of the curvature-sorted vertex list, highlighted in cyan and blue, respectively (cyan may overwrite blue). This color scheme applies to the subsequent figures in this section as well.

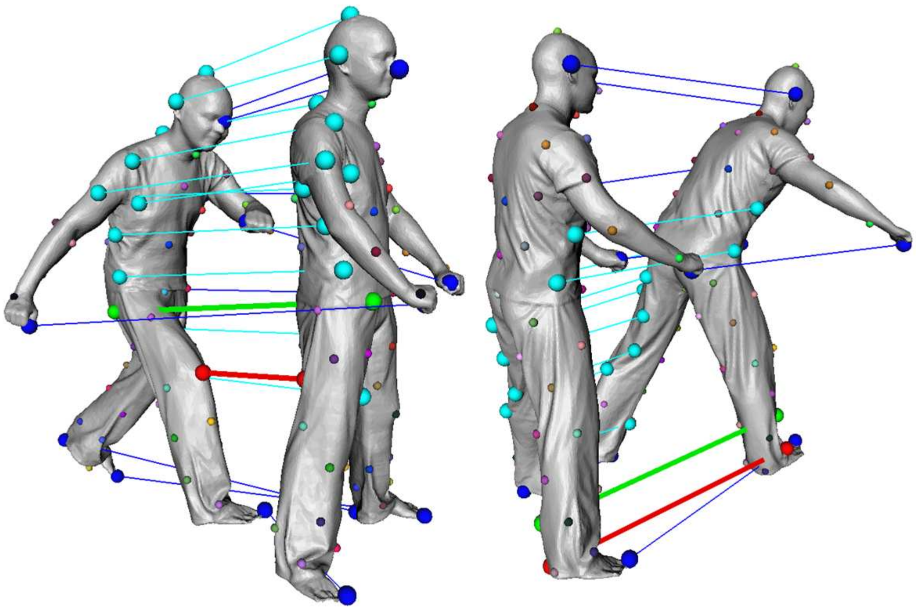

Figure 3.8: The final one-to-one mappings obtained for two different shape pairs from *Dancing Man*.

In Fig. 3.11, we demonstrate the outputs obtained at different stages of our correspondence algorithm on a Jumping Man pair and on a hybrid Jumping Man - Dancing Man pair. For each pair, we display the initial one-to-one mapping $\S_0^{(0)}$ (the output of the first bipartite matching process), the final one-to-one mapping $\S_0^*$ (the output of the last bipartite matching process at convergence), and the final many-to-one mapping $\S^*$ (the output of the last greedy optimization process at convergence). Note the improvements obtained through the different stages of the correspondence algorithm. Fig. 3.26 demonstrates these improvements on a particular example, i.e., on a Ballerina - Jumping Man hybrid pair which contains local but severe non-isometries due to the fingers of the Ballerina shape (note that the Jumping Man model does not have any fingers). Hence the Ballerina shape has more base vertices sampled on the hands as compared to the Jumping Man, and since the initial one-to-one mapping forces every source base vertex to match a different target base, the result severely deviates from the ground-truth. Our final many-to-one mapping however correctly assigns all base vertices of the fingers to the single base vertex sampled on the

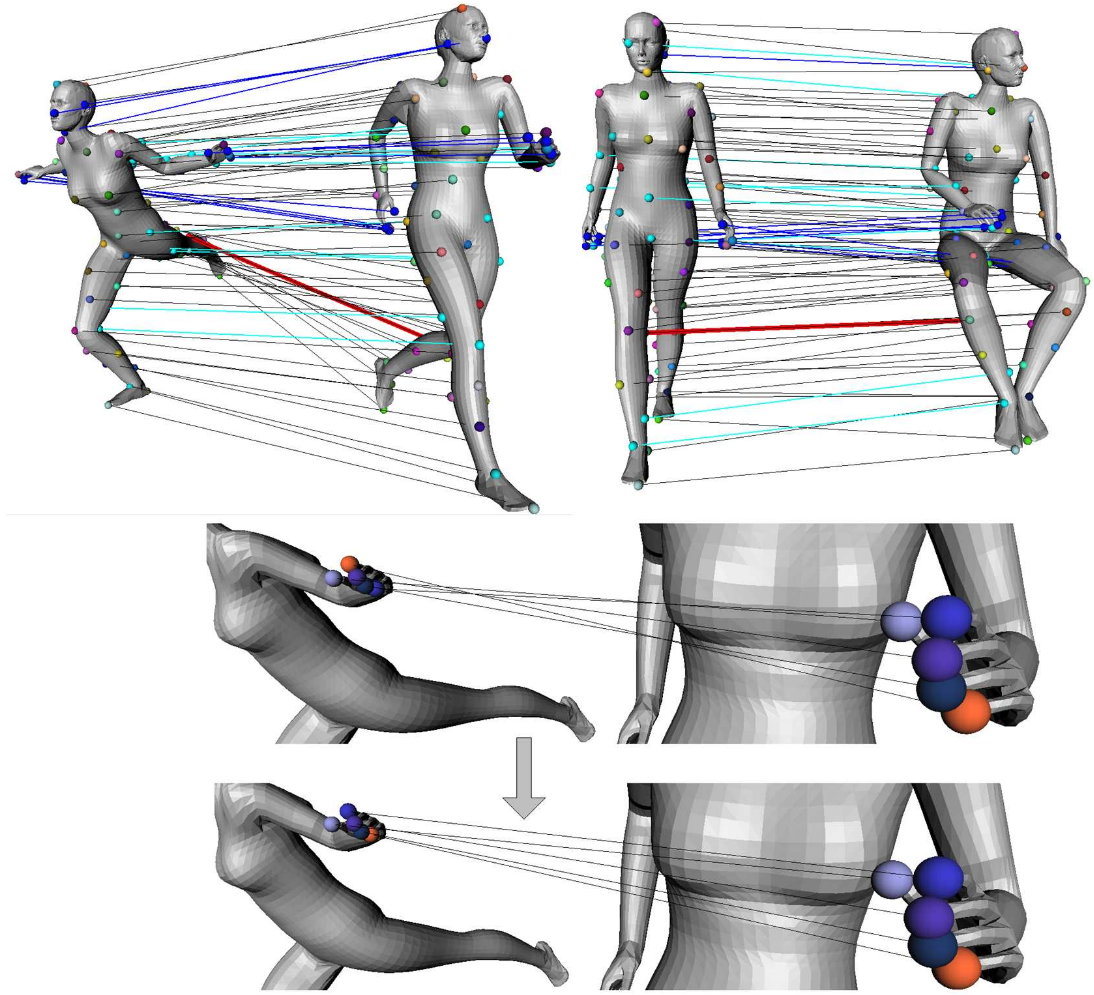

Figure 3.9: The final one-to-one mappings obtained for two different *Ballerina* pairs (top row). Note that $N \approx 80$ base vertices include all high-curvature points of a hand, which are then accurately matched (bottom row, left pair shown in zoom). Observe how the initial mismatches on the fingers ($\S_0^{(0)}$, middle row) are healed by our EM-based optimization framework ($\S_0^{*}$, bottom row).

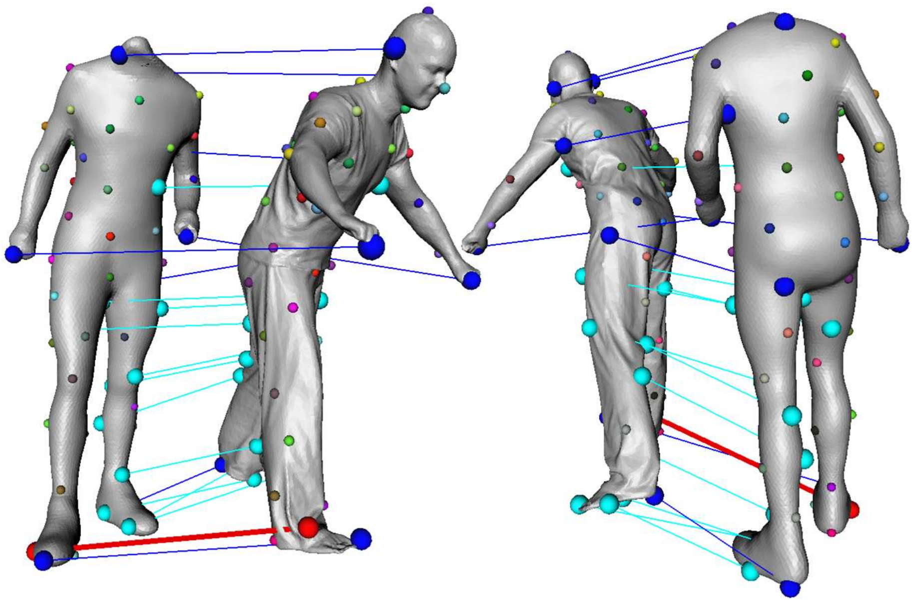

Figure 3.10: The final one-to-one mapping obtained for a *Jumping Man-Dancing Man* hybrid pair, displayed from two different views.

corresponding hand of the Jumping Man.

Table 3.2 provides, on different datasets, the values that the performance measures take at different stages of our algorithm, for $\S_0^{(0)}$, $\S_0^*$, and $\S^*$ (see also Fig. 3.11). The average-based performance measures, D_{grd} and D_{iso}, and the maximum-based measures $d^{\dagger}_{\mathrm{grd}}$ and $d^{\dagger}_{\mathrm{iso}}$, are each computed over 10 different runs of the algorithm on 10 different pairs, except one pair for *Dog-Wolf* from the Non-rigid World database and five pairs for *Ballerina*. Each run matches two spatially-apart poses of the articulated object in the corresponding sequence. We note that, in some cases, especially at low resolution matchings when N is small, the spectral alignment procedure (Section 3.1.5.2) may fail to resolve the sign ambiguities of the eigenvectors due to the symmetries in the object shapes. The results given in Table 3.2 exclude such cases in order not to artificially burst the ground-truth correspondence errors. All the performance measures are provided as a factor of the sampling distance r for better interpretation of the errors.

The results given in Table 3.2 can be assessed also by considering the input quality, i.e., the isometric deviations inherent in the original datasets. To this effect, we have measured

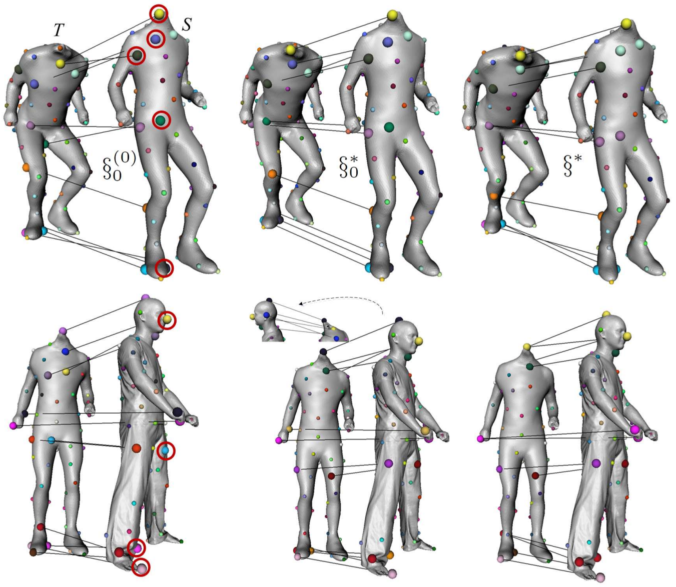

Figure 3.11: (Left) The initial one-to-one mapping $\S_0^{(0)}$, (middle) the final one-to-one mapping $\S_0^*$, and (right) the final many-to-one mapping $\S^*$, for an isometric (top) and a nearly isometric pair of shapes (bottom). Some base vertices from the source shapes, for which the final correspondences improve significantly as compared to the initial, are each marked with a red circle.

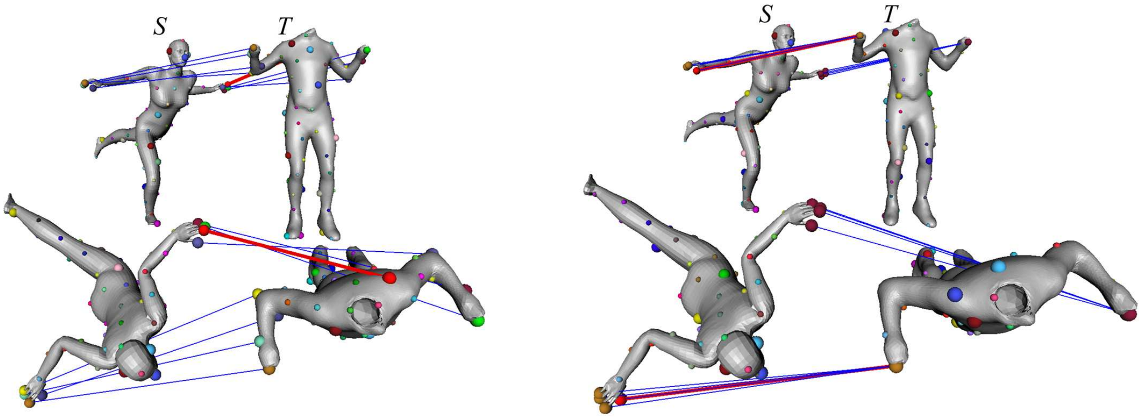

Figure 3.12: (Left) The initial one-to-one mapping $\S_0^{(0)}$, and (right) the final many-to-one mapping $\S^*$ obtained for a *Jumping Man-Ballerina* hybrid pair, displayed from two views.

the isometric distortion within and across our test sequences by computing a D_{iso} value, given the manual match of 10 critical vertices on each shape pair, and obtained the following distortion values (from the most isometric to the least): 0.030 for Jumping Man, 0.034 for Ballerina, 0.036 for Dancing Man, 0.039 for Horse Gallop, 0.046 for Jumping-Dancing pair, 0.047 for Jumping Man-Ballerina pair, and 0.060 for the Dog-Wolf pair. Each of these values can be interpreted as the isometric distortion of the semantical mapping that can be established between the corresponding shapes, i.e., as a measure of how isometric the given shapes to be matched are. In Table 3.2, we observe that the isometric distortion inherent in a dataset puts roughly an upper bound to the performance, hence a lower bound to the D_{iso} value that can be achieved by our method. We see that, for each dataset, our one-to-one mapping performance achieves a D_{iso} value which is close to this bound, and even exceeds it in the many-to-one setting.

The execution time of our implementation is mainly dominated by the number of vertices in the original meshes due to geodesic distance computation. On a 6GB 2.53GHz 64-bit workstation, the overall algorithm takes, for $N \approx 80$ base vertices, 2.1, 5.6, 8.0, 2.0, 6.0, 3.5 and 1.5 seconds, respectively on Horse Gallop ($V = 9$K), Jumping Man ($V = 16$K), Dancing Man ($V = 20$K), Ballerina ($V = 6$K), Jumping Man - Dancing Man pair ($V = 20$K), Jumping Man-Ballerina pair ($V = 16$K), and Dog-Wolf pair ($V = 3.4$K). We note that the portion of the execution time that belongs to geodesic computation varies between 77 and 90 percents on different datasets. This suggests that our shape correspondence method can be made even more efficient by using faster geodesics computation algorithms such as the one in [50].

Comparison with the spectral method. The spectral method of Jain and Zhang [39] is one of the state of the art techniques for embedding-based non-rigid shape correspondence. The method generates as output a many-to-one mapping from source to target. In Fig. 3.13, we compare our many-to-one mapping results with the many-to-one mappings obtained via this spectral method on two distinct cases: on an isometric shape pair from the Ballerina sequence, and on the non-isometric Dog-Wolf pair from the Nonrigid World database, which contains two shapes with similar structure but distorted isometry. We observe that our method outperforms the spectral method significantly in both cases, the worst matchings being significantly better as well as the correspondences being generally more accurate.

	Initial one-to-one, $\S_0^{(0)}$		Final one-to-one, $\S_0^*$		Final many-to-one, $\S^*$		Many-to-one of [39]	
Pair	$D_{\text{grd}}, d^\dagger_{\text{grd}}$	$D_{\text{iso}}, d^\dagger_{\text{iso}}$	$D_{\text{grd}}, d^\dagger_{\text{grd}}$	$D_{\text{iso}}, d^\dagger_{\text{iso}}$	$D_{\text{grd}}, d^\dagger_{\text{grd}}$	$D_{\text{iso}}, d^\dagger_{\text{iso}}$	$D_{\text{grd}}, d^\dagger_{\text{grd}}$	$D_{\text{iso}}, d^\dagger_{\text{iso}}$
Horse Gallop	$0.73r, 7.71r$	$0.64r, 2.75r$	$0.58r, 2.92r$	$0.56r, 1.85r$	$\mathbf{0.35}r, \mathbf{2.17}r$	$\mathbf{0.29}r, \mathbf{0.87}r$	$0.55r, \mathbf{2.17}r$	$0.50r, 1.49r$
Jumping Man	$0.57r, 2.45r$	$0.49r, 0.91r$	$0.52r, 2.45r$	$0.28r, 0.54r$	$\mathbf{0.40}r, \mathbf{1.92}r$	$\mathbf{0.24}r, \mathbf{0.47}r$	$0.52r, 3.98r$	$0.38r, 0.68r$
Dancing Man	$0.58r, 7.48r$	$0.48r, 2.55r$	$0.44r, 1.88r$	$0.28r, 0.82r$	$\mathbf{0.38}r, \mathbf{1.22}r$	$\mathbf{0.23}r, \mathbf{0.60}r$	$0.44r, 1.41r$	$0.27r, 0.77r$
Ballerina	n/a	$0.50r, 1.46r$	n/a	$0.43r, 1.02r$	n/a	$\mathbf{0.26}r, \mathbf{0.63}r$	n/a	$0.46r, 1.25r$
Jumping-Dancing	n/a	$0.51r, 1.03r$	n/a	$0.46r, 1.12r$	n/a	$\mathbf{0.37}r, \mathbf{0.59}r$	n/a	$0.47r, 1.03r$
Jumping-Ballerina	n/a	$1.05r, 3.32r$	n/a	$0.67r, 2.00r$	n/a	$\mathbf{0.46}r, \mathbf{1.14}r$	n/a	$0.64r, 1.50r$
Dog-Wolf	n/a	$0.73r, 2.38r$	n/a	$0.69r, 1.62r$	n/a	$\mathbf{0.51}r, \mathbf{0.79}r$	n/a	$0.65r, 1.20r$

Table 3.2: Quantitative performance analysis of our shape correspondence method in comparison with the spectral method of [39]. The best (lowest) performance values on each row are given in bold. In all cases the sampling distance r is approximately 0.07 (the maximum geodesic distance on the mesh is taken as 1.0), which yields about $N = 80$ base vertices.

While our method correctly matches all the shape extremities, the tips of the left ears on the Dog-Wolf pair are not for example correctly matched with the spectral method. We have obtained the results of the spectral method by using the Matlab code made publicly available by the authors of [39]. In the comparison tests, we have run their code on the same sets of base vertices (about 80) and the same geodesic affinity matrices that we have used for our algorithm.

The last column of Table 3.2 presents the average and maximum-based performance measures for the spectral method in comparison to ours. These measures are computed over the shape pairs for which none of the two methods results in a symmetric flip. We observe that our many-to-one mapping $\S^*$ outperforms the results of the spectral method significantly in all settings for all datasets. Another interesting observation is that our one-to-one mapping $\S_0^*$ is performance-wise almost on a par (even slightly better in some occasions) with the many-to-one output of the spectral method.

We note that, although the complexity of the spectral method as presented in [39] is $O(V^2 \log V)$ against our $O(NV \log V)$ complexity, it can be reduced to $O(NV \log V)$ by using the Nyström method which however introduces further approximation to the Euclidean embedding process [100].

Comparison with GMDS. We have conducted experiments on the Nonrigid World shape database to compare our method with the GMDS method of [38]. For GMDS, we have used the publicly available Matlab code which uses multiresolution optimization [38].

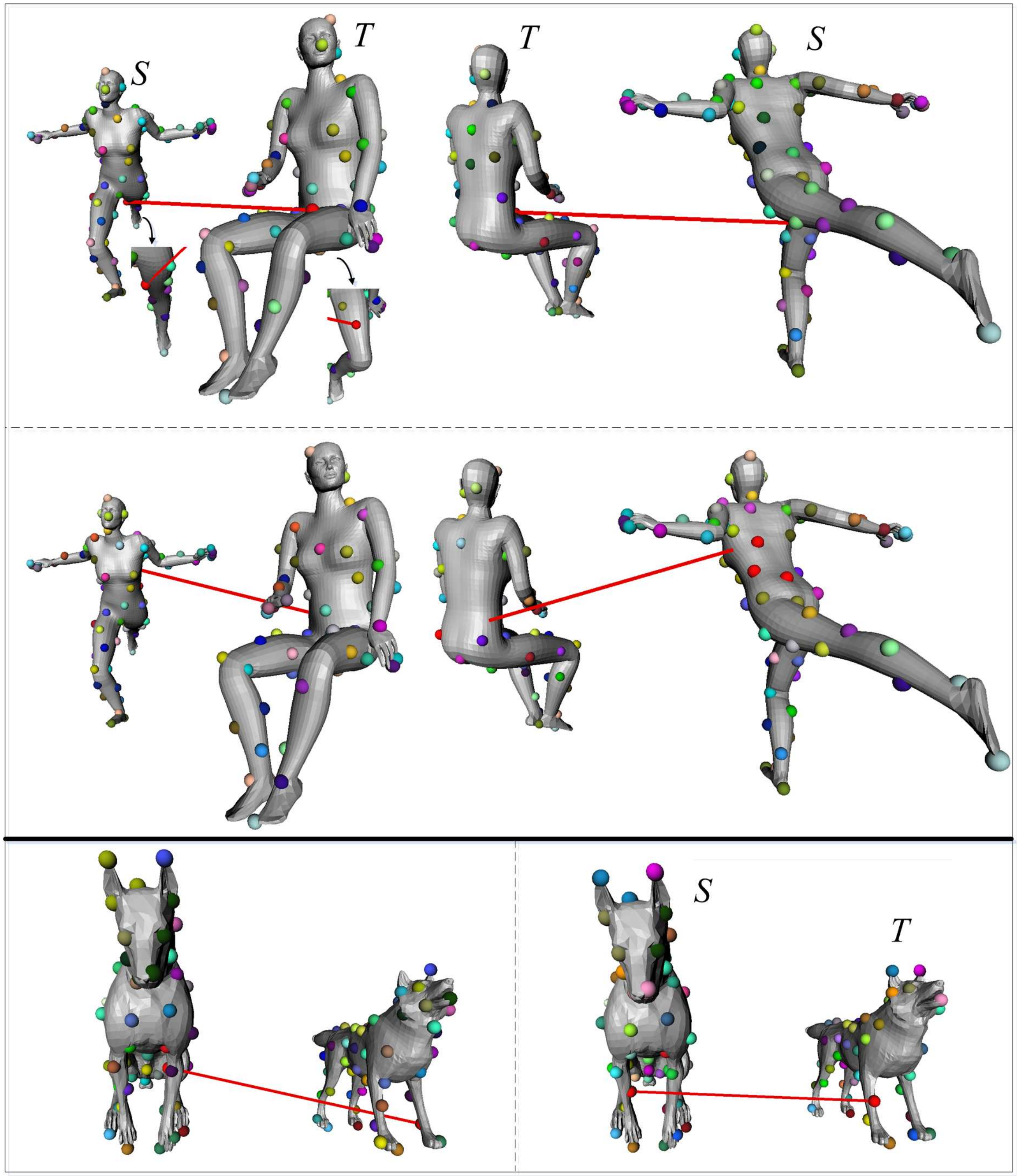

Figure 3.13: The many-to-one mappings on the *Dog-Wolf* hybrid pair from the Nonrigid World database (bottom) and on a *Ballerina* pair (top); obtained with our method (right) and with the spectral method of [39] (left).

The part of the Nonrigid World database that we have used contains mesh models of 9 cats, 11 dogs, 3 wolves, 17 horses, 24 female figures, and two different male figures containing respectively 15 and 20 poses. We have excluded some shape classes from the experiments; the gorillas having disconnected shape components, the shark which has only one mesh model in its class, and the lions on which the Matlab code for GMDS has crashed.

We have evaluated the performances in three different categories: 1) within human models, 2) within animal models, and 3) across human models. In the experiments belonging to the first and second categories, for each model, we pick a random model from the same class and then compute the correspondence in between, e.g., we match each dog to another dog. In the third category, for each human object, we pick a random model from each different human class and then compute the correspondence. We note that both methods can result in symmetric flips, though the symmetry problem with the GMDS method is more severe due to the coarse initialization step which is based on only 8 samples in the available implementation. In our experiments, while our method matched about 40% of the shape pairs over the three categories as free of symmetric flip errors, this ratio was only 27% in the case of GMDS. We note that almost all the shapes in the Nonrigid World database have intrinsic symmetries (though not perfect), and the number of symmetric flips tend to decrease with our method as we increase the number of samples, which was about 80 in these experiments. In the comparison tests, we included only the pairs that could be matched as free of symmetry problems by both methods.

Like the spectral method, the GMDS method generates as output a many-to-one mapping from the source shape to the target. Hence also in this case, for comparison we have used our many-to-one mapping results. The GMDS method has its own sampling procedure, and the publicly available code can be set so as to generate the same number of vertices that our method produces. Thus in our experiments for comparison with GMDS, the sampled vertices on a given mesh are different but the same in number.

We present the quantitative results of the comparison tests in Table 3.3. We observe that our method significantly outperforms GMDS across humans category, and is on a par with it for the isometric categories (recall also that the complexity of GMDS is $O(V^2 \log V)$ against our $O(NV \log V)$ complexity). Note that the ground-truth correspondence information is not available with the Nonrigid World database. We have excluded across animal models,

	Many-to-one (GMDS)	Many-to-one (our method)
Category	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$
Across humans	$0.625r, 1.701r$	$\mathbf{0.407}r, \mathbf{1.158}r$
Within humans	$\mathbf{0.314}r, 1.139r$	$0.379r, \mathbf{0.885}r$
Within animals	$\mathbf{0.358}r, \mathbf{1.036}r$	$0.405r, 1.231r$

Table 3.3: Quantitative performance of our shape correspondence method in comparison with the GMDS method on the Nonrigid World dataset.

as a forth possible category, from our experiments for comparison with GMDS since in that case some shape pairs to be matched would have severe global non-isometries that both methods, which are essentially isometric, would fail to handle. The three categories that we have experimented on correspond to (nearly) isometric instances of the correspondence problem with the following isometric distortion values: 0.035 for across humans, 0.028 for within humans, and 0.030 for within animals (or equivalently, $0.500r$, $0.400r$, and $0.428r$, respectively, where $r = 0.07$). Recall however that the Dog-Wolf pair for which we have presented visual and quantitative results above is from the Nonrigid World database.

We visually demonstrate the comparison tests on three examples in Fig. 3.14. In the figure, we see that the vertices matched with GMDS are not as evenly distributed as they are in our case, resulting in clustered correspondences. Note, for instance on the top row, that the samples clustered on the inner part of the left leg of one model have been matched to the samples clustered on the outer part of the left leg of the other model. This is mainly due to the fact that the GMDS algorithm is a gradient-based iterative optimization process that produces sub-vertex matchings which do not necessarily coincide with the initial sampling. We also note that, while computing the resulting isometric distortion value for a GMDS output, we round sub-vertex coordinates to their nearest vertices on the surface mesh. The effect of this rounding process to the computed distortion values is however negligible since the mesh models in the Nonrigid World database are almost uniform and at relatively high resolution. We also observe from the visual comparison that our method can match the salient points of a shape more successfully than GMDS (e.g., ear tips of the cats), thanks to our saliency-based sampling.

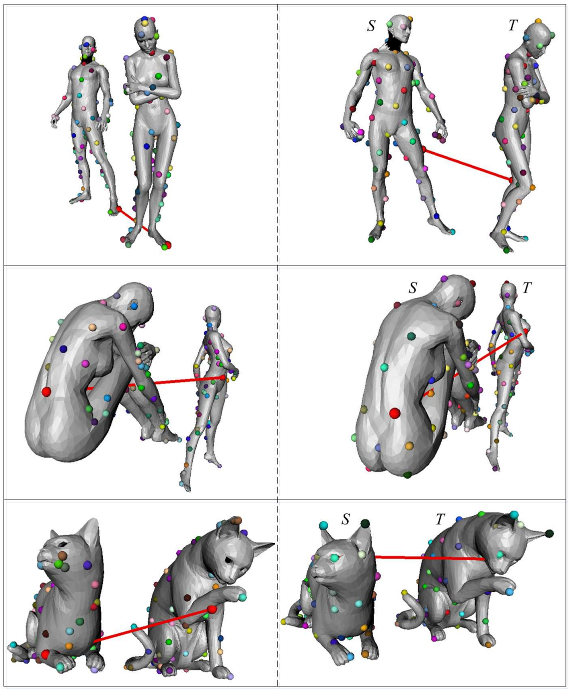

Figure 3.14: Many-to-one mappings obtained with GMDS (left) and with our method (right) for sample pairs selected from across humans (top), within humans (middle), and within animals (bottom) categories of the Nonrigid World dataset. Our worst match between the cat models (red line) falls inside the skull, hence is not visible. The vertices matched with our method are always evenly-spaced and at prominent regions such as finger or ear tips unlike GMDS matches. Note that correspondences tend cluster with GMDS in all cases.

3.1.9 Conclusion

We have proposed an isometric shape correspondence method that minimizes the isometric distortion in the original 3D Euclidean space by using EM algorithm. We have conducted experiments on various datasets, and our findings can be summarized as follows:

- Our method performs well not only on isometric shapes, but also on pairs of shapes which are nearly isometric, such as mesh representations of two different humans or animals, i.e., for shapes of the objects which are different but semantically and/or structurally similar.

- The COES sampling algorithm that we propose solves the joint sampling problem by evenly sampling high-curvature vertices from both shapes using a sufficiently small sampling distance.

- When the shapes to be matched are given in terms of evenly sampled surface points (base vertices), any isometric correspondence technique in the literature can be used to initialize our EM-based method. In that sense, our method can be used to further improve the results of any embedding-based isometric method, such as [39], that possibly suffers from approximations and/or ambiguities in the embedding domain.

- Our method generates two optimal mappings at a time from one shape to the other, one injective and the other many-to-one, the former having generally higher isometric distortion than the latter, especially in the presence of severe non-isometries between the shapes.

- Our method is computationally very efficient, and can be made even more efficient by using faster algorithms for computation of geodesic distances.

- Our method can be applied to any pair of isometric shapes with arbitrary genus. However, it cannot handle the deformations which include topological changes that would modify geodesic distances drastically.

One limiting factor for our method is that it has been designed to work on isometric shapes, though the experiments show that our algorithm works well also for nearly isometric shapes. Yet, our algorithm may not produce good correspondence results when two shapes contain severe global non-isometries such as some of the hybrid animal shape pairs in the Nonrigid World database. Another limitation of our method is due to the problem of symmetric correspondences, which is inherent to all isometry-based correspondence methods. Hence, if two shapes to be matched include intrinsic symmetries, then our method may fail to find the correct matching and result in symmetric flips. To resolve the symmetry problem, explicit symmetry information is needed and one can resort to methods that can detect global intrinsic symmetries such as [101, 89]. Our tracking mechanism that alleviates the symmetry problem [25] (Section 3.3) can also be adapted to this work that can be regarded as multiresolution thanks to the sampling distance parameter.

3.2 Dense Correspondence of Complete Shapes

In this section, we present coarse-to-fine combinatorial matching for dense isometric shape correspondence [24].

3.2.1 Literature review

We append the following literature review on dense correspondence to the review in Section 3.1.1 that already applies to this section where we describe an isometric matching algorithm.

An important problem in shape correspondence is how to achieve dense correspondence, i.e., a matching between all vertices of two given high resolution meshes. The main bottleneck in achieving dense correspondence is the computational complexity of the existing algorithms. Most embedding-based methods naturally support dense correspondence but their computational load is usually a limiting factor. To the best of our knowledge there exist only two methods in the literature [21, 24], that can achieve this in $O(V \log V)$ time, where V is the number of vertices in the given meshes to be matched. While [21] introduces various approximations, [24] to be described in this section may suffer from symmetric flip issue that is inherent to multiresolution matchers. Note that the problem in the latter is mostly handled with the application of our generic symmetric flip solver [25] (Section 3.3).

One of the best performant dense matching algorithms in the literature appears to be the BIM (blended intrinsic maps) method proposed in [21]. The approach is to search for a continuous blend of multiple low-dimensional maps which can be explored via Mobius transformations. By combining these conformal maps with weights varying smoothly over the surface, a space of maps is obtained, which can be searched in polynomial-time. Thanks to blending, non-isometric deformations are handled better with this method as compared to other Möbius transformation based methods such as in [31, 36]. However, the BIM method introduces various approximations to achieve dense correspondence in $O(V \log V)$ time. First, the isometric distortion of each blending map is estimated on the extended complex plane to which the surfaces to be matched are transformed (flattened) via mid-edge uniformization, hence with some embedding error. Second, geodesic centroids of the blending maps are approximated with Euclidean distances. Third, confidence weights of the blending process are computed over all vertices of a dense surface mesh via interpolation.

These approximations may lead to inaccurate matches (see Fig. 4.24 for an example). Moreover the method has restrictions on topology, suffering from non-delaunay triangles during the process of mid-edge uniformization and also requiring zero-genus meshes as input.

In this section, the isometric dense correspondence problem is solved by using a multiresolution strategy with the following contributions achieved. First, we propose a dense shape correspondence method which is computationally efficient. We minimize the isometric distortion directly in the 3D Euclidean space, i.e., in the domain where isometry is originally defined, by using a coarse-to-fine combinatorial search algorithm. Our method does not require any initialization and aims to find an accurate solution in the minimum-distortion sense for perfectly isometric shapes. Second, our shape correspondence method is based upon a coarse-to-fine joint sampling technique that incrementally samples evenly-distributed salient vertices from a given mesh at increasing levels of detail.

3.2.2 Problem statement and overview

Our goal is to establish a dense correspondence between two given isometric (or nearly isometric) shapes. We assume that each shape is represented by a manifold surface mesh of sufficiently high resolution, on which geodesic distances can easily be computed. We designate one of the shapes as *source* and the other as *target*. Let S and T denote the vertex sets of the source and the target meshes, respectively. Let also that a mapping $\S : S \rightarrow T$ (or a relation in the most general setting) is given. We then measure the isometric distortion D_{iso} as defined in Eq. 3.1. The problem can then be formulated as a combinatorial search over all possible mappings so as to minimize the isometric distortion function given in Eq. 3.1:

$$\S^* = \arg\min_{\S} D_{\text{iso}}(\S) \tag{3.14}$$

Since S and T consist of vertices sampled on isometric (or nearly isometric) shapes, we require the optimal mapping $\S^*$ to establish a full correspondence between S and T so that every $s_i \in S$ is related to some $t_j \in T$, and likewise every $t_j \in T$ is related to some $s_i \in S$. If we further assume that the shapes are perfectly isometric and represented by uniform triangle meshes, then $\S^*$ is constrained to be a one-to-one correspondence with $|S| = |T|$, and can be found by evaluating Eq. 3.1 for $N!$ different possible mappings, where

$N = |S| = |T|$. This combinatorial solution however has factorial complexity and hence computationally intractable for typical values of N.

Motivated by the simplicity of this optimal but expensive solution, we provide a coarse-to-fine shape matching algorithm based on combinatorial search. The idea is to reduce the search space by exploiting the fact that the optimal mapping $\S^*$ maps nearby vertices on the source shape to nearby vertices on the target. This suggests that shape matching can be performed on a patch-by-patch basis in a coarse-to-fine fashion. To this effect, we incrementally sample evenly-distributed vertices from the surfaces of both shapes at increasing levels of detail (Section 3.2.3). In parallel to this sampling process, at each level of detail, we match the sampled vertices by combinatorial search on a patch-by-patch basis (Section 3.2.4), using Equations 3.1 and 3.14. Our algorithm is built upon the basic assumption that the shapes to be matched are perfectly isometric. However two digital shapes (or their mesh representations) are hardly ever so in practice. Moreover $|S|$ is not usually equal to $|T|$, hence the correspondence $\S^*$, which is optimal in the minimum-distortion sense, is a many-to-many mapping. We note that the final output of our correspondence algorithm is also a many-to-many mapping.

3.2.3 Coarse-to-fine sampling

In this section, we describe our coarse-to-fine algorithm that we use to sample *base vertices* from the mesh representations of the given shapes. We sample S and T separately, at increasing levels of detail, by using a sequence of decreasing *sampling radii* $\{r^{(k)}\}$ such that $r^{(k)} > r^{(k+1)}$ for $k = 0, 1, ..., K$. Let $S^{(k)}$ denote the set of base vertices sampled from S at level k. We require the sets $\{S^{(k)}\}$: i) to be incremental such that $S^{(k)} \subset S^{(k+1)} \subseteq S$, and ii) to consist of samples as evenly distributed as possible on the shape surface. Note that the same notation applies also to T.

In order to obtain a consistent joint sampling between two shapes, we use an importance sampling strategy which prioritizes salient vertices (shape extremities and/or high curvature points) during selection of the samples. We initially sort the vertices of S into a list in descending order with respect to their saliency values and mark them all as base vertex candidates. The sampling algorithm starts, at the coarsest level $k = 0$, by selecting the top vertex of the list as the first base vertex. We launch the Dijkstra's shortest paths algorithm from this vertex and unmark all the vertices lying within a distance $r^{(0)}$. Then, the next base vertex is picked as the first marked vertex of the list. When this is repeated until no marked vertex is left, we obtain a sampling of the surface, where the base vertices are at least $r^{(0)}$ apart from each other. The same sampling procedure is applied to each level k by initializing the sampling with $S^{(k)} = S^{(k-1)}$, where $S^{(-1)} = \emptyset$. At the beginning of each level k, we mark all the vertices but the ones which are within a distance $r^{(k)}$ from the base vertices inherited from the previous level. We then pick the first marked vertex in the list and unmark all the vertices within a distance $r^{(k)}$. When this is repeated until no marked vertex is left, we obtain the base vertex set $S^{(k)}$ such that the samples are at least $r^{(k)}$ apart from each other. The maximum level of sampling is obtained when all the vertices in S are picked as base vertices.

Each base vertex $s_i \in S^{(k)}$, sampled as described above, defines a surface patch around itself at level k. This patch, that we denote by $S_i^{(k)}$, includes all the vertices within a distance $r^{(k)}$ from the base s_i, as illustrated in Fig.3.16. The algorithm also ensures that each vertex is covered by the patch of at least one base vertex. Hence, the resulting patches partition the shape surface at each level into overlapping regions of approximately equal size such that $S^{(k)} = \bigcup_i S_i^{(k)}$. The sampling algorithm is given as pseudocode in Fig. 3.15.

Our sampling algorithm is built on a sampling technique used in [48]. As described above, we extend this technique, which is uniresolution and which selects the base vertices evenly but in an arbitrary manner, so as to have a multiresolution sampling algorithm which prioritizes the selection of salient vertices. We measure saliency by the integral of geodesic distance function of [48] at the initial coarsest level, and by the Gauss curvature at all other levels. Hence the base vertices are placed on local maxima of Gauss curvature at every level except that they are selected from the shape extremities at the initial level.

3.2.4 Correspondence algorithm

We establish the vertex correspondence between the source shape S and the target T, at increasing levels of detail, in parallel to our coarse-to-fine sampling algorithm described in the previous section. Hence our shape correspondence algorithm produces a sequence of correspondences, $\{\S^{(k)}\}$, which is refined as the level k increases, and at the maximum level possible, we expect to get the optimal dense correspondence: $\S^* = \S^{(K)}$. In the sequel, we

```
Input: Vertex set $S$, sampling radii $\{r^{(k)}\}$ s.t. $r^{(0)} > r^{(1)} > ... > r^{(K)}$
Output: Base vertex sets $\{S^{(k)}\}$ s.t. $S^{(0)} \subset S^{(1)} \subset ... \subset S^{(K)}$
$S^{(-1)} = \emptyset$; $k = -1$;
Sort $S$ in descending order w.r.t. saliency;
Iterate on level $k$
    Mark all vertices in $S$ as base vertex candidates;
    $S^{(k)} = S^{(k-1)}$;
    For each base $s_i \in S^{(k-1)}$
        Unmark all the vertices within patch $S_i^{(k)}$;
    Repeat
        Let $s_j$ be the first marked vertex in $S$;
        $S^{(k)} = S^{(k)} \cup \{s_j\}$;
        Unmark all the vertices within patch $S_j^{(k)}$;
    Until no marked vertex is left
Until all vertices in $S$ are sampled (or maximum level $K$ is achieved)
```

Figure 3.15: The coarse-to-fine sampling algorithm.

explain how we obtain each $\S^{(k)}$, i.e., the correspondence at each level k.

3.2.4.1 Patch-based combinatorial matching

Recall from Section 3.2.3 that each base vertex set $S^{(k-1)}$, or equivalently $T^{(k-1)}$, partitions the surface of the shape S (or T) into overlapping patches of approximately equal size $\{S_i^{(k-1)}\}$ (or $\{T_i^{(k-1)}\}$). Suppose that each such patch at level $k-1$, say $S_i^{(k-1)}$, contains a fixed number M of base vertices from the next level k, i.e., from $S^{(k)}$. If we designate $S_i^{(k-1)}$ as the parent of these k^{th} level base vertices, then our coarse-to-fine sampling procedure can be thought of as a process in which the surface of the shape S is recursively subdivided into M smaller patches as levels proceed. Note that, while a newly sampled base at level k (or its patch) may have multiple parents, a base vertex inherited from level $k-1$ has only one parent patch.

Given $\S^{(k-1)}$, the correspondence $\S^{(k)}$ can be obtained on a patch-by-patch basis, i.e., for every $(s_i, t_j) \in \S^{(k-1)}$, by matching the k^{th} level base vertices inside the patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$, respectively, as illustrated in Fig. 3.17. The M base vertices of two corresponding

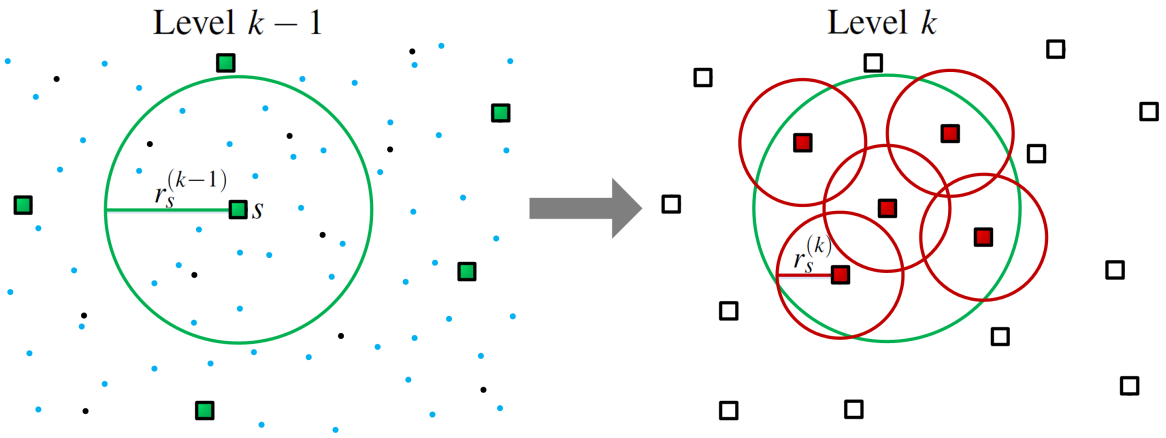

Figure 3.16: Coarse-to-fine sampling. Small blue dots represent the original vertex set S. (Left) Green points are the base vertices sampled at level $k-1$, where the base s_i defines the patch $S_i^{(k-1)}$ with radius $r^{(k-1)}$. (Right) The newly sampled base vertices at level k (in black), along with the ones inherited from the previous level (in green), constitute the base vertex set $S^{(k)}$ at level k, where each s_j defines a patch $S_j^{(k)}$ around itself with radius $r^{(k)}$. The patch $S_i^{(k-1)}$ is the parent of the base vertices that it covers at level k.

patches, say $S_i^{(k-1)}$ and $T_j^{(k-1)}$, can be matched by combinatorial search provided that M is sufficiently small. To this effect, we evaluate each time the isometric distortions of $M!$ possible one-to-one mappings $\S$ from the base vertices in $S_i^{(k-1)}$ to the base vertices in $T_j^{(k-1)}$ via Eq. 3.1 and pick the one with the least distortion. We denote this minimum-distortion patch-to-patch matching by $\S_m^{(k)}$. Recall that, to be able to compute $D_{\text{iso}}(\S)$ in Eq. 3.1, we need to set a correspondence list $\S'$ to be used in Eq. 3.2. The list $\S'$, which includes $\S$ by default, should be global enough to avoid mismatches due to local symmetries, yet local enough to perform well on details, hence we augment it with the matchings of the parent patches, whenever available: $\S' = \S \cup \bigcup_n \S_n^{(k-1)}$, where each $\S_n^{(k-1)}$ denotes the correspondence at level $k-1$ between the parent patches of s_i and t_j (note that a base vertex may have multiple parents).

At each level k, we set the value of the sampling radius $r^{(k)}$ based on the area of the largest patch at level $k-1$, denoted by $A^{(k-1)}$. We use the ad-hoc formula, $r^{(k)} = 0.6\sqrt{A^{(k-1)}/\pi}$, which ensures that the number M of bases sampled within a parent patch is about 5 (or more generally, $M \leq 6$), and that in turn allows us to match the resulting

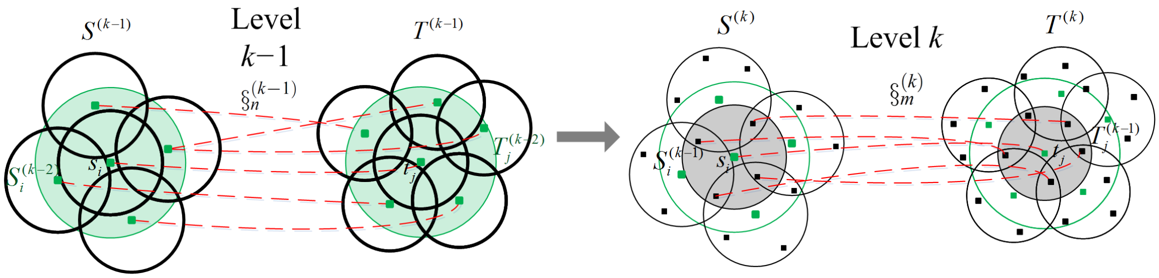

Figure 3.17: Coarse-to-fine patch-based combinatorial matching. At level k, the base vertices inside the (grey) patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$ are matched by combinatorial search. At level $k-1$, the matching $\S_n^{(k-1)}$ is one-to-many since the (green) patches $S_i^{(k-2)}$ and $T_j^{(k-2)}$ have different number of bases. Note that $\S_n^{(k-1)}$ is included in $\S'$ while matching the patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$ at level k.

patches via combinatorial search.

Since two shapes are never perfectly isometric, two patches to be matched may indeed have different number of base vertices sampled, say M and M' and $M' > M$, as it is also the case in in Fig. 3.17. In such cases we have $\binom{M'}{M}M!$ different possible one-to-one mappings to evaluate for combinatorial matching. Once the minimum-distortion one-to-one mapping is found via combinatorial search, there remain unmatched vertices in one of the patches, to which we assign residual matches in order to assure that the whole surface is covered by the correspondence algorithm as depicted in Fig. 3.17. To achieve this, each unmatched vertex in one patch is paired with all the vertices in the other one by one, and the pair that minimizes the isometric distortion is picked as the residual match. Hence each patch-to-patch matching at level k results in a mapping $\S_m^{(k)}$, which can be one-to-one, many-to-one or one-to-many, depending on the number of base vertices sampled in each patch.

3.2.4.2 Correspondence merging

At the end of each level k, we merge the patch-to-patch correspondences $\{\S_m^{(k)}\}$, obtained via combinatorial search as explained in Section 3.2.4.1, into one global correspondence $\S^{(k)}$ that covers the whole surface. Since patches partition the shape surface into overlapping regions, the union of these matchings, $\S^{(k,0)} = \bigcup_m \S_m^{(k)}$, gives an initial correspondence that needs to be simplified. The merging process at level k is carried out in the following three steps, that involve elimination of redundant multiple matches and trimming of the outliers (as illustrated in Fig. 3.18):

- **Step 1:** For every base vertex $s_i \in S^{(k)}$, we keep only one correspondence pair, the one with the minimum distortion among all $(s_i, t_j) \in \S^{(k,0)}$, which gives us $\S^{(k,1)}$, the output of the first step of the merging algorithm. We compute each isometric distortion $d_{\text{iso}}(s_i, t_j)$ via Eq.3.2 with the setting $\S' = \S^{(k,0)}$. Note that $\S^{(k,1)}$ is many-to-one.

- **Step 2:** To every isolated base vertex $t_j \in T^{(k)}$ with no correspondence $s_i \in S^{(k)}$ s.t. $(s_i, t_j) \in \S^{(k,1)}$, we assign the base $s_n \in S^{(k)}$ that yields the minimum distortion among all $(s_n, t_j) \in \S^{(k,0)}$, which gives us $\S^{(k,2)}$, the output of the second step. We compute each isometric distortion with the setting $\S' = \S^{(k,1)}$.

- **Step 3:** We replace every outlier $(s_i, t_j) \in \S^{(k,2)}$ for which $d_{\text{iso}}(s_i, t_j) > 2 \cdot D_{\text{iso}}(\S^{(k,2)})$, with (s_i, t_n) that yields the minimum distortion among all $t_n \in T^{(k)}$. We compute each isometric distortion with the setting $\S' = \S^{(k,2)}$. If the removal of an outlier creates an isolated target base, then step (2) is repeated. The output of this last step gives us the final correspondence $\S^{(k)}$, which is a many-to-many mapping.

3.2.4.3 Overall algorithm

The overall correspondence algorithm is composed of three basic tasks at each level of detail: sampling, combinatorial matching and merging. A critical parameter of the algorithm is the sampling radius $r^{(k)}$, which determines at each level k the number M of bases sampled within each parent patch. Recall that we set $r^{(k)}$ based on the area of the largest patch at level $k-1$, $A^{(k-1)}$, by using the ad-hoc formula given in Section 3.2.4.1, which ensures a sufficiently small value of M for combinatorial matching, $M \leq 6$. Hence in practice, we have two sampling radii at each level, $r_S^{(k)}$ and $r_T^{(k)}$, one for the source shape S and one for the target T, which have close but different values (since S and T are assumed to be nearly isometric). We initialize the area parameters, as $A_S^{(-1)} = A_S$ and $A_T^{(-1)} = A_T$, where A_S and A_T denote the surface areas of the source and the target, respectively. The overall correspondence algorithm can then be written as pseudocode in Fig. 3.19.

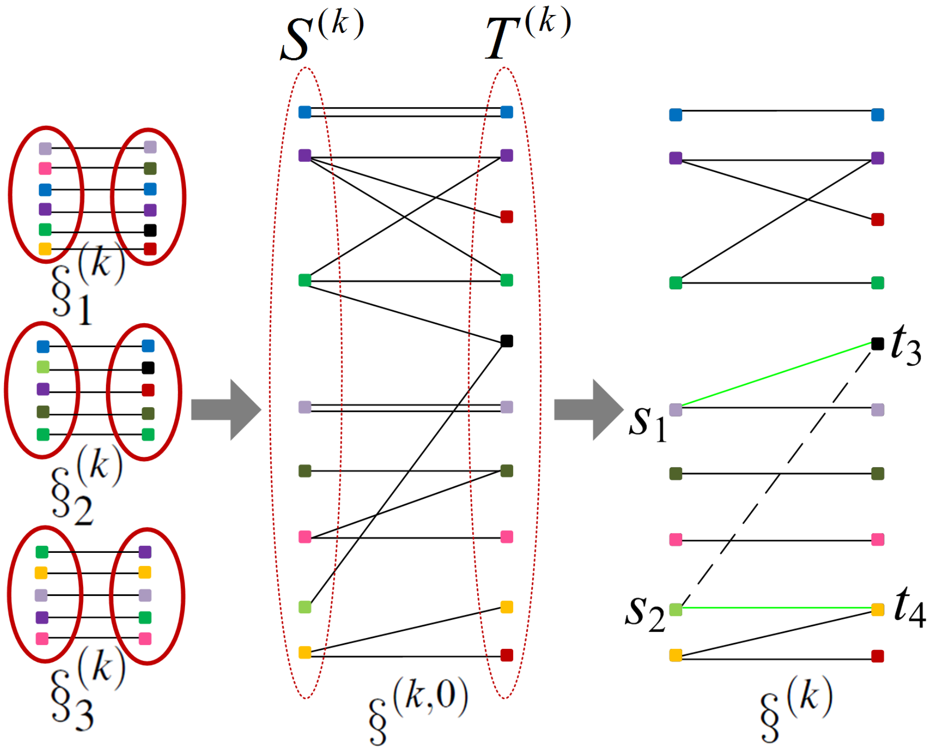

Figure 3.18: Merging of patch-based correspondences, $\S_1^{(k)}$, $\S_2^{(k)}$ and $\S_3^{(k)}$, into one global mapping $\S^{(k)}$ at level k. The illustration considers only three patches for demonstration purposes. Points from different patches with the same color correspond to the same base vertex. Note that some redundant matches are eliminated and the outlier (s_2, t_3) is replaced with (s_1, t_3) and (s_2, t_4). The resulting $\S^{(k)}$ covers every base vertex on the source shape as well as on the target.

3.2.5 An insight to why the algorithm works

We now show that, under certain conditions, a coarse-to-fine matching algorithm, such as ours, can be used to gradually localize an accurate dense correspondence as the level of detail increases, based on the following ”inclusion assertion”:

Inclusion Assertion: Suppose S and T are sampled at sufficiently high resolution from two perfectly isometric surfaces. Let $(s_i, t_i) \in \S^*$, where $\S^*$ is the optimal correspondence that minimizes $D_{\text{iso}}(\S)$, i.e., $D_{\text{iso}}(\S^*) = 0$. Suppose also that $\S^{(k)}$ is the optimal correspondence at some level k, and that the optimal solution is unique, which exempts us from ambiguities

```
Input: Vertex sets $S$ and $T$
Output: Dense Correspondence $\S^{(K)} : S \to T$
Iterate on level $k$ (initially $k = 0$)
    Set the sampling radii $r_S^{(k)}$ and $r_T^{(k)}$ based on the largest patch
        areas at the previous level, $A_S^{(k-1)}$ and $A_T^{(k-1)}$;
    Compute the base sets $S^{(k)}$ and $T^{(k)}$ by the sampling
        algorithm (Section 3.2.3);
    For each $(s_i, t_i) \in \S^{(k-1)}$
        Match $k^{\mathrm{th}}$ level base vertices in patch $S_i^{(k-1)}$ and patch $T_i^{(k-1)}$
            via combinatorial search, that gives $\S_m^{(k)}$ (Section 3.2.4.1);
    Merge patch-to-patch correspondences $\{\S_m^{(k)}\}$
        into $\S^{(k)}$ (Section 3.2.4.2);
Until all vertices in $S$ and $T$ are matched;
```

Figure 3.19: The dense correspondence algorithm.

due to symmetries. Then,

$$\forall i, j \quad (s_i, t_j) \in \S^{(k)} \to t_i \in T_j^{(k)}$$

where $T_j^{(k)}$ is the patch with radius $r^{(k)}$ centered around t_j at level k.

We will not formally prove this assertion, but rather give an outline of its justification, that will provide us with an insight into why and under what conditions our coarse-to-fine correspondence algorithm works. First note that $D_{\mathrm{iso}}(\S)$ is locally a slowly changing convex function around its optimal point. Hence the mapping $\S^{(k)}$, which is optimal at level k, will assign each base vertex s_i to some base t_j as nearest to the optimal t_i as possible. Moreover $S^{(k)}$ and $T^{(k)}$ consist of evenly-spaced base vertices sampled from perfectly isometric shapes, such that each vertex in S and T is included by the patch of at least one base vertex with radius $r^{(k)}$. Based on these, we can state the following: $\forall i, j \ \ (s_i, t_j) \in \S^{(k)} \to g(t_i, t_j) \leq r^{(k)}$, where $g(.,.)$ is the geodesic distance function, which directly implies the assertion. The inclusion assertion is illustrated in Fig. 3.20 and demonstrated on a pair of isometric shapes in Fig.3.21.

In the case of perfect isometry, the inclusion assertion basically suggests that, based on the optimal correspondence $\S^{(k-1)}$ at level $k-1$, a patch-based matching algorithm, such as ours, can be used to find the optimal correspondence $\S^{(k)}$ at level k. Starting from the coarsest level $k = 0$, our algorithm aims to maintain the optimality at each level k by patched-based combinatorial matching and merging, assuming the optimality of the correspondence found at the previous level. The algorithm is expected to eventually converge to the optimal correspondence $\S^*$ as levels proceed, since at the last level there will be only one vertex left in each patch.

The inclusion assertion, hence our algorithm, discards the symmetry problem which is however inherent to all isometry-based correspondence methods. We note that, since our algorithm is coarse-to-fine, even though the shapes to be matched are not symmetric as a whole, symmetric flips may occasionally arise at the initial level due to coarse sampling, which can then effect further levels of matching. Moreover, since real shapes are usually only nearly isometric, the initial sampling based on extremities or Gaussian curvature with particular choices of the sampling radius can lead to failure cases. That means, the optimal correspondence $\S^*$ ideally given by Eq. 3.14 may not always match the dense correspondence $\S^{(K)}$ that our algorithm produces.

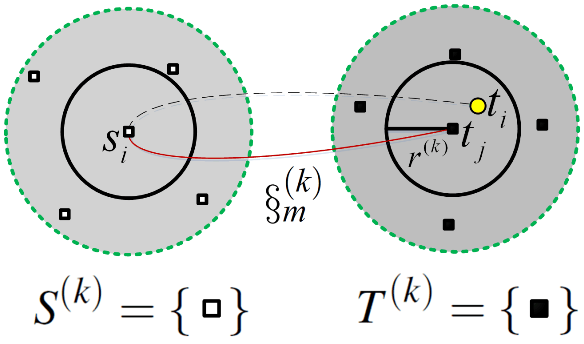

Figure 3.20: Illustration of the inclusion assertion at level k. The optimal correspondence t_i of s_i is included in the patch defined by t_j, whenever $(s_i, t_j) \in \S_m^{(k)}$.

3.2.6 Computational complexity

Sorting the vertices of the original mesh w.r.t. saliency prior to the sampling process takes $O(N \log N)$ time where $N = \max(|S|, |T|)$. Our correspondence algorithm then proceeds

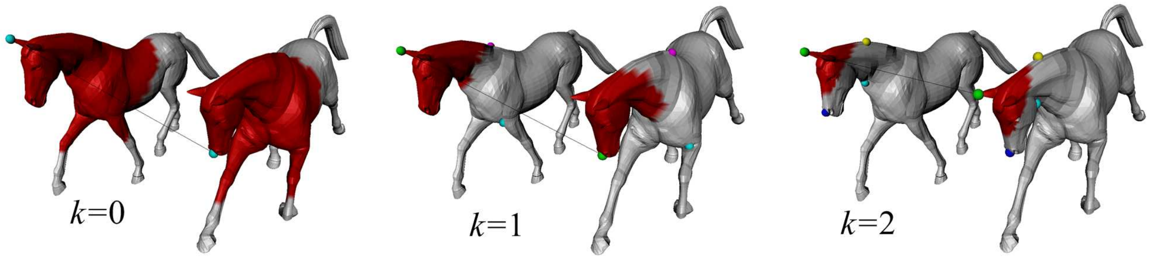

Figure 3.21: Demonstration of the inclusion assertion. Note how the correspondence between the right ears of two horses is refined as levels proceed. At levels $k = 0$ and $k = 1$, the correspondence is far from the optimal, yet the optimal target vertex is included at each level by the patch of the matched base, as required by the inclusion assertion. At level $k = 2$, the left ears are correctly matched. The corresponding patches are painted in red at each level.

with the following computations:.

- *Coarse-to-fine sampling.* We sample about $M = 5$ base vertices within each parent patch. Restricted to the patch to be sampled, Dijkstra's shortest paths algorithm, when applied to all active patches, takes $O(\sum_{k=1}^{K} M^k \cdot \frac{N}{M^k} \log_M \frac{N}{M^k})$ time, where M^k and $\frac{N}{M^k}$ can be interpreted as the number and the size of current patches, respectively, and $K = \log_M N$ is the maximum level of detail that can be achieved. The summation can be expanded as $N \log_M 1 + N \log_M M + .. + N \log_M N$, yielding a total of $O(N \log N)$ time complexity.

- *Patch-based combinatorial matching.* Each patch pair is matched in constant time $O(M!)$ since there are $M!$ mappings to be evaluated via D_{iso} that can be computed also in constant time. Hence all patch pairs at all levels are matched in $O(\sum_{k=1}^{K} M^k \cdot M!)$ time, which unfolds to a total of $O(N \log N)$ complexity since $M^K = N$.

- *Merging.* Given a global initial mapping $\S^{(k,0)}$ of size M^k at the end of each level k, its evenly-spaced subset that contains E matches ($E \ll N$) is used as the list $\S'$ to be traversed in the computation of d_{iso} (see step 1 in Section 3.2.4.2). Hence, the total time complexity can first be written as $O(\sum_{k=1}^{K} 3 \cdot M^k E)$, where the term $M^k E$ is due to d_{iso} computations in the three-step merging algorithm. As before, the total complexity then reduces to $O(N \log N)$ with a proper choice of E such as 100.

The complexity analysis above reveals $O(N \log N)$ overall time complexity which scales well to large meshes, especially when compared to other isometric shape correspondence algorithms in the literature, such as $O(N^2 \log N)$ complexity of [39] and [38], and $O(N^4 \log N)$ complexity of [31].

3.2.7 Experimental results

We have tested our algorithm on several shape datasets: 1) on various mesh sequences and 2) on Nonrigid World database [38]. The mesh sequences are namely *Ballerina* [99], *Jumping Man* [97] and *Horse Gallop* [14], each containing different poses of an articulated object. While *Jumping Man* and *Horse Gallop* are given as fixed-connectivity mesh sequences sampled uniformly at high-resolution with 16K and 9K vertices, respectively, the *Ballerina* sequence exhibits severe non-uniformity as well as different number of vertices and connectivity among its 5 available different poses with around 6K vertices. On the other hand, the part of the Nonrigid World database that we use contains mesh models of 9 cats, 11 dogs, 3 wolves, 17 horses, 24 female figures, and two different male figures, containing 15 and 20 poses, where each object has approximately 3K vertices with arbitrary connectivity.

We measure the performance of our shape correspondence algorithm in terms of deviation from the ground-truth as well as isometric distortion. To quantify isometric distortion, we use the average distortion measure D_{iso} defined in Eq. 3.1, and also a maximum distortion measure that we denote by $d^{\dagger}_{\text{iso}}$ (Eq. 3.11). Similarly we compute average and maximum ground-truth correspondence errors respectively by (whenever the ground-truth correspondence is available) D_{grd} (Eq. 3.12) and $d^{\dagger}_{\text{grd}}$ (Eq. 3.13).

In the figures where shape correspondences are displayed, the worst matchings, with respect to isometric distortion $d^{\dagger}_{\text{iso}}$ and the ground-truth correspondence error $d^{\dagger}_{\text{grd}}$ (whenever available), are highlighted with bold red and green lines, respectively. In Fig. 3.22, we display the dense correspondence obtained on a shape pair from the Ballerina sequence, and zoom on the fingers to show the accuracy and smoothness of the mapping obtained. Similarly, dense correspondences established between Horse Gallop pairs as well as hybrid Wolf-Horse pairs are given in Fig. 3.23.

In Fig. 3.24, we display the correspondences obtained at increasing levels of detail for

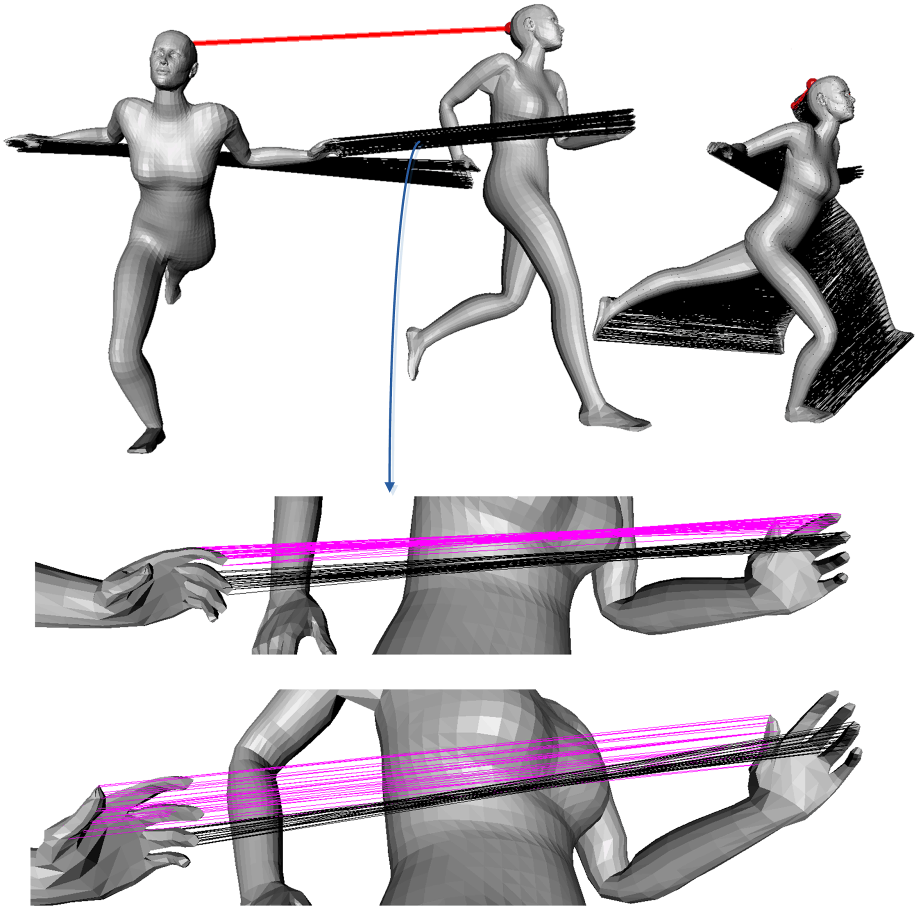

Figure 3.22: Dense correspondence for a Ballerina pair ($\S^{(K)}$ with $K = 11$), zooming on the fingers of the left hands. The worst match w.r.t isometric distortion, as measured by $d^{\dagger}_{\text{iso}}$, is indicated with a bold red line.

a Ballerina pair and a Dog-Wolf pair from the Nonrigid World. We observe that the correspondences are improved and refined as levels proceed, and eventually we obtain a very accurate correspondence, even on the Dog-Wolf pair which contains severe non-isometries as compared to the articulated Ballerina pair. As a general rule, the mapping becomes denser and hence the isometric distortion is expected to decrease as levels proceed. This is verified by the plot given in Fig. 3.25, except that there is a consistent small increase in distortion from level $k = 0$ to $k = 1$, which is due to accurate joint sampling of shape extremities at the initial level. In Fig. 3.25, we plot the surface coverage of matched points as well as the

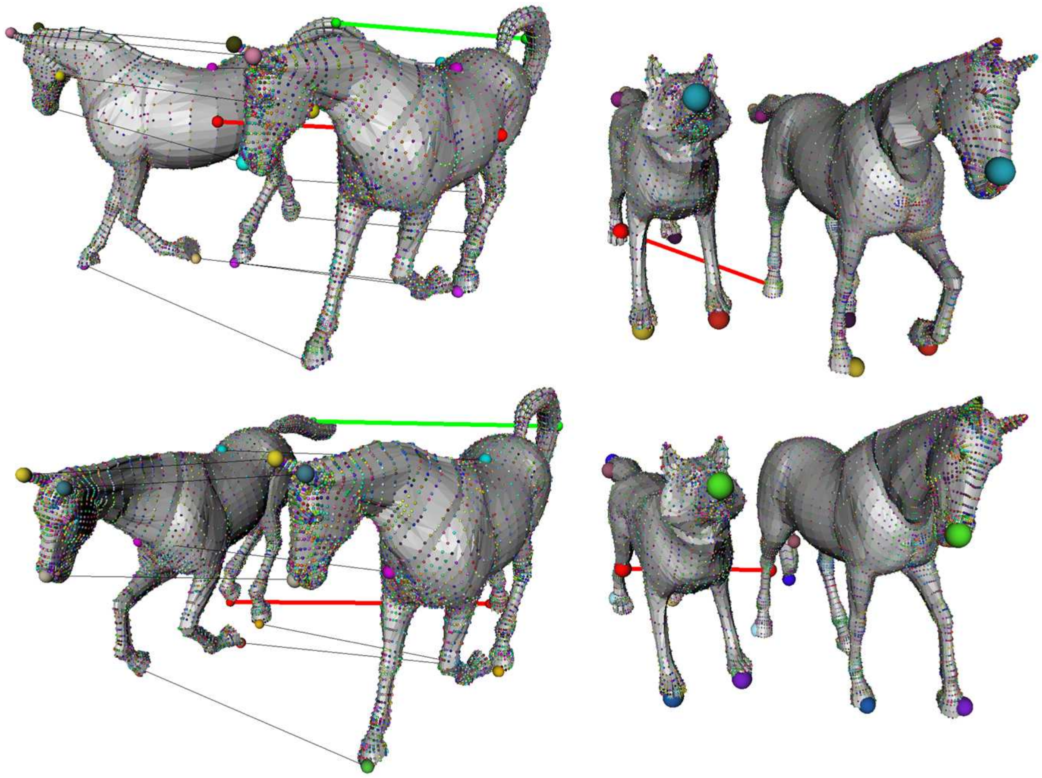

Figure 3.23: Dense correspondences obtained on a Horse Gallop pair (left) and on a Horse-Wolf pair (right). The match with the worst ground-truth error is indicated with a green line for the Horse pair.

isometric distortion for varying k, computed by averaging the results over 6 sample pairs (whenever available).

The isometric quality of the input is an important factor for the performance of our algorithm. To measure isometric quality, we assign each dataset an isometric distortion value, denoted by ξ, which is computed as the value of D_{iso} for the manual one-to-one matching of 10 shape extremities: (from the most isometric to the least) 0.034 for Ballerina, 0.039 for Horse Gallop, 0.047 for the Jumping Man-Ballerina pair, 0.060 for the Dog-Wolf pair, and 0.064 for the Wolf-Horse pair. Although any pair can be regarded as only nearly isometric from a rigorous point of view, the Ballerina and Horse Gallop sequences, having higher input quality, provide better outputs when compared to the other hybrid pairs, as observed in Table 3.4 as well as in Fig. 3.25. Nevertheless, the hybrid pairs still yield high-

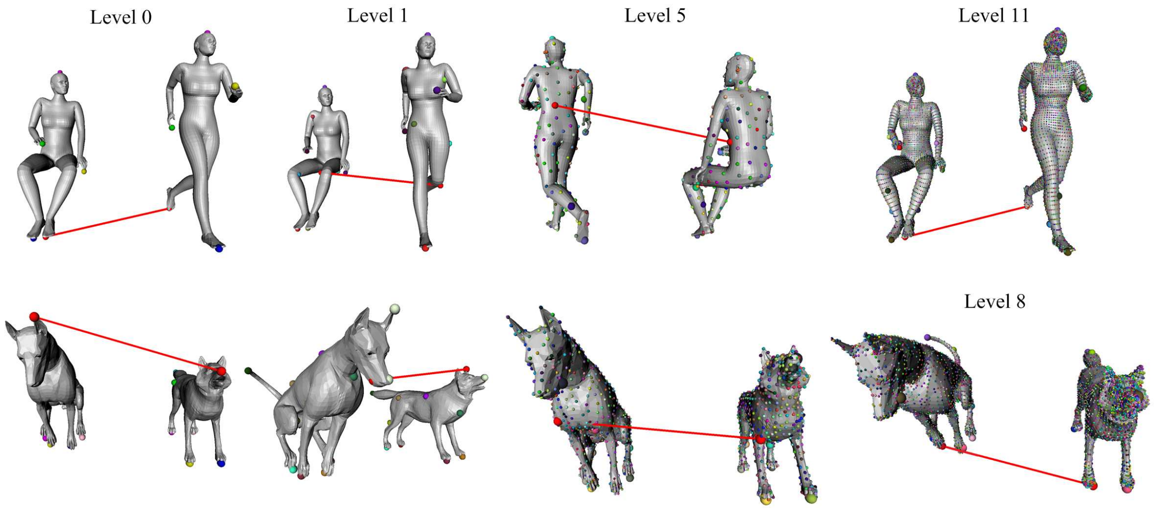

Figure 3.24: Correspondences obtained at increasing levels of detail for a Ballerina pair (top) and for the Dog-Wolf pair (bottom).

quality results despite significant non-isometries that they contain, e.g., due to the fingers of the Ballerina matched with the hand of the Jumping Man (with no fingers) as shown in Fig. 3.26, or due to the scale differences between the corresponding parts of the Wolf and Dog shapes such as tails, legs, and faces (Fig. 3.24).

We compare our isometric shape correspondence method with two state-of-the-art techniques: the spectral method of [39] and the generalized multidimensional scaling (GMDS) method of [38]. We have obtained the results of these two methods by using the Matlab codes made publicly available by the authors. Since neither the spectral method nor the GMDS scales well to large meshes, in the comparison experiments we have stopped our coarse-to-fine algorithm at an intermediate level ($k = 4$, $\sim$ 200 base vertices for the spectral method and $k = 5$, $\sim$ 300 base vertices for GMDS). For comparisons with the spectral method, we have run their code on the same set of base vertices that we have used for our algorithm. The GMDS method however has its own sampling procedure, and the publicly available code can be set so as to generate the same number of vertices that our method produces. Hence in the latter case, the sampled vertices are different but the same in number.

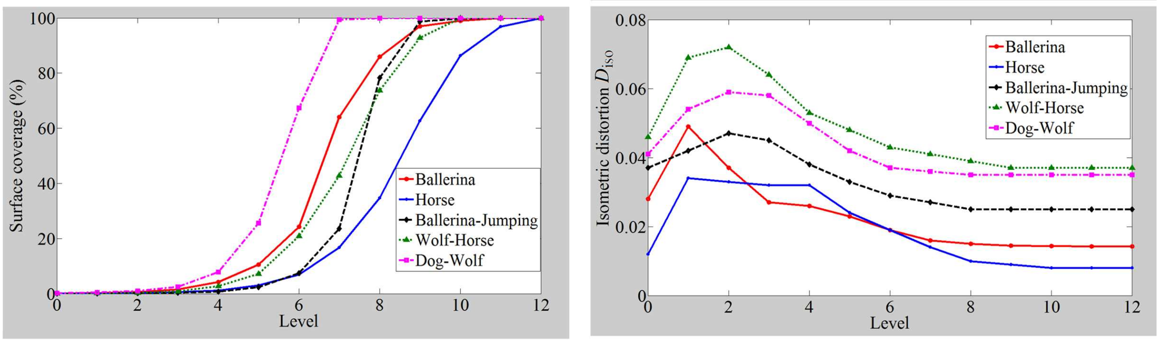

Figure 3.25: The surface coverage (left) and isometric distortion (D_{iso}) (right) of the correspondence obtained at each level k for different shape datasets.

All the performance measures provided in Table 3.4 for comparison with the spectral method are each computed and averaged over 10 different runs of the algorithm on 10 different pairs, except one pair for the *Dog-Wolf* and 5 pairs for *Ballerina*, where each run matches two spatially-apart poses of the articulated object in the corresponding sequence. In Table 3.5, we provide the results of the comparison of our method with GMDS on the Nonrigid World database. In this case, we evaluate the average performance in three different categories: 1) within human models, 2) within animal models, and 3) across human models. In the first and second categories, all possible shape pairs are tested such that a human or animal model is paired up only with models from the same class, e.g., a dog to another dog, whereas in the third category, a human object is matched to a model if and only if it represents a different human. We note that all three methods can result in symmetric flips since they are all isometry-based. Hence when comparing two methods, we have included only the pairs that can be matched as free of symmetry problems by both methods in consideration. Note also that, in our case, if the value of M_0, i.e., the number of extremities at the coarsest level, is increased in the range $5 \leq M_0 \leq 9$, the number of symmetric flips tend to decrease.

The spectral method, as well as the GMDS method, generates as output a many-to-one mapping from the source shape to the target while our method produces a many-to-many correspondence. Hence to make our result compatible, for comparison we have used the many-to-one mapping, $\S^{(k,1)}$, that the first step of our merging algorithm generates (see

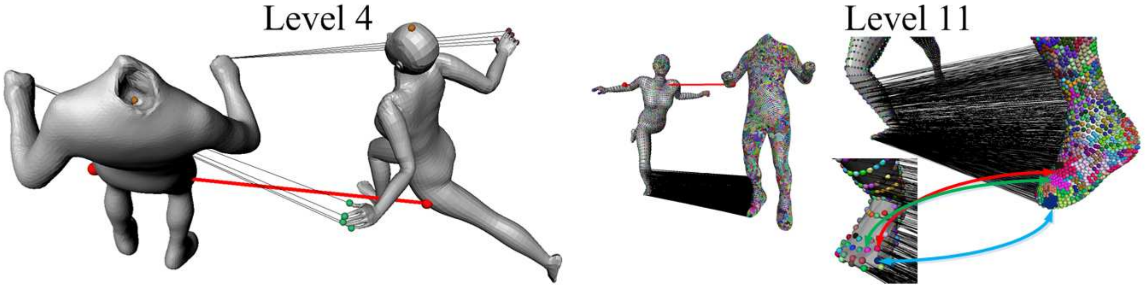

Figure 3.26: Correspondence results on a Jumping Man - Ballerina hybrid pair. At level $k = 4$, the fingers of the Ballerina are mapped to a single vertex on the hand of the Jumping Man as desired (left). Clustered matchings are observed on the Jumping Man at the last level, which is as expected since its mesh representation is significantly denser than the Ballerina mesh (right).

Section 3.2.4.2). In Table 3.4, we observe that our many-to-one mapping, $\S^{(k,1)}$, outperforms the results of the spectral method for all datasets, and that the improvement is more significant on more isometric Ballerina and Horse Gallop pairs. In the table, we also provide the performance results for the dense mapping, $\S^{(K)}$, obtained with our method, which are significantly better than the results obtained at intermediate levels. Note that, on all datasets, the isometric distortion of our final many-to-many dense mapping is lower than the corresponding input distortion ξ. We have the ground-truth correspondences only for the Horse Gallop sequence, hence we compare the two methods in this case also w.r.t their ground-truth error performances. Our method outperforms the spectral method with $D_{\text{grd}} = 0.039$ and $d^{\dagger}_{\text{grd}} = 0.102$ against $D_{\text{grd}} = 0.062$ and $d^{\dagger}_{\text{grd}} = 0.195$, when the results are averaged over 10 different runs of the algorithm on 10 different pairs. In Fig. 3.27, we visually compare the performance of our algorithm with the spectral method on a Horse Gallop pair. We observe that our method outperforms the spectral method significantly in this case, the worst matching being significantly better with our method as well as the correspondences being generally more accurate. While our method correctly matches all the shape extremities, the tips of the ears on the Horse are not for example correctly matched with the spectral method.

When compared with the GMDS method, we observe in Table 3.5 that our method is performance-wise on a par in terms of isometric distortion, even slightly better in some cases (recall that the complexity of GMDS is $O(N^2 \log N)$ against our $O(N \log N)$ complexity). This is visually demonstrated on two examples in Fig. 3.28. In the figure, we see that the matched vertices with GMDS are not as evenly distributed as they are in our case. This is mainly due to the fact that the GMDS algorithm is an iterative optimization process that produces sub-vertex matchings which do not necessarily coincide with the initial sampling. Hence while computing the resulting isometric distortion value for a GMDS output, we round sub-vertex coordinates to their nearest vertices on the surface mesh. The effect of this rounding process to the computed distortion values is however negligible since the mesh models in the Nonrigid World database are almost uniform and at relatively high resolution. We also observe from the visual comparison that our method can match the salient points of a shape more successfully than GMDS (e.g., ear tips of the cats), thanks to our saliency-based sampling.

	Many-to-one [39]	Many-to-one (our method)	Many-to-many (our method)
Pair	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
Ballerina	.023, .054	.016, .038	.014, .024
Horse Gallop	.021, .040	.013, .026	.008, .015
Jumping-Baller.	.041, .069	.035, .058	.028, .051
Dog-Wolf	.050, .105	.044, .075	.036, .055
Wolf-Horse Gal.	.055, .087	.048, .079	.037, .066

Table 3.4: Quantitative performance of our method in comparison with spectral method of [39].

	GMDS	Our method
Pair	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
Within animals	.022, .087	.021, .096
Within humans	.018, .076	.020, .079
Across humans	.026, .068	.038, .082

Table 3.5: Quantitative performance of our method in comparison with the GMDS method of [38] on Nonrigid World database.

Finally we note that, on a 6GB 2.53GHz 64-bit workstation, the overall execution time of our implementation, needed to match all the vertices so as to obtain a dense correspondence, is 38, 103, 110, 281 and 337 seconds for Dog-Wolf, Ballerina, Wolf - Horse Gallop, Horse Gallop and Jumping Man-Ballerina pairs, respectively.

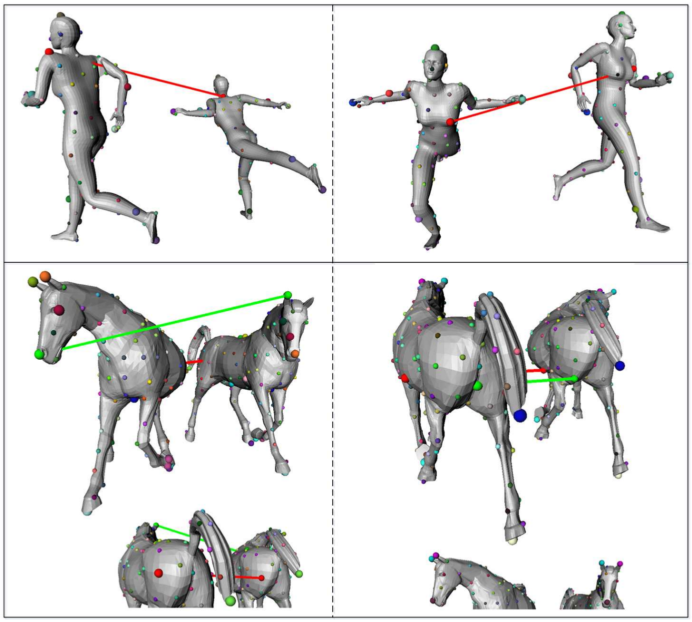

Figure 3.27: Many-to-one mappings obtained with the spectral method [39] (left) and with our method at level $k = 4$ (right) for a Ballerina pair (top) and a Horse Gallop pair (bottom). Green and red lines indicate the worst matches w.r.t. ground-truth and isometric distortion, respectively.

3.2.8 Conclusion

We have proposed a dense isometric shape correspondence method based on coarse-to-fine sampling and combinatorial matching. Our findings can be summarized as follows:

- Our method is computationally very efficient with $O(N \log N)$ complexity, and hence scales well to large meshes.

- The dense correspondences that we obtain are very accurate since the isometric distortion is minimized in the original 3D Euclidean space, as free of approximation errors that embedding-based methods usually suffer from.

- Our method performs well not only on isometric shapes, but also on pairs of shapes which are nearly isometric, such as mesh representations of two different humans or animals, i.e., for shapes of the objects which are different but semantically and/or structurally similar.

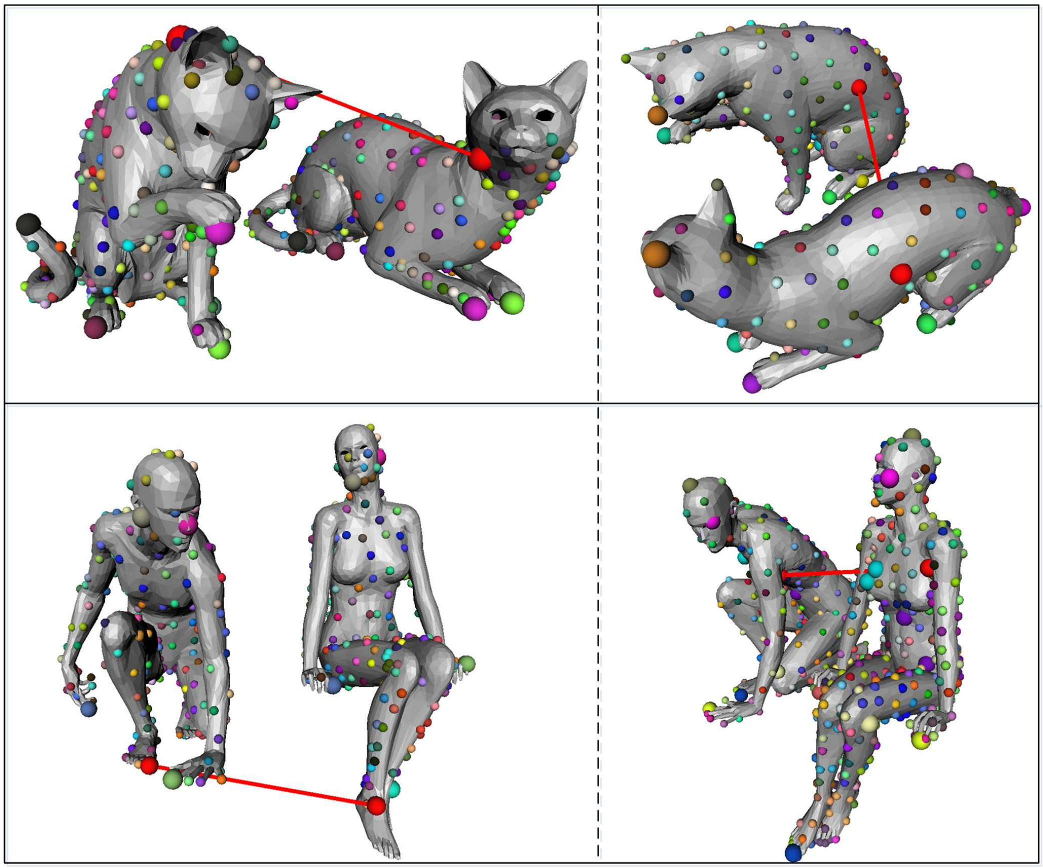

Figure 3.28: Many-to-one mappings obtained with the GMDS method [38] (left) and with our method at level $k = 5$ (right) for a within-animal pair (top) and for a across-human pair (bottom).

- Our method produces accurate correspondences at different levels of detail thanks to our coarse-to-fine joint sampling algorithm.

A shortcoming of our method is due to the symmetry problem, which is inherent to all isometry-based correspondence algorithms. Moreover, since our algorithm is coarse-to-fine, even though the shapes to be matched are not symmetric as a whole, symmetric flips may occasionally arise at the initial level due to coarse sampling. This also relates to the difficult problem of finding a reliable initial correspondence as discussed in Section 3.2.1, which we plan to address as future work. We also plan to evaluate the performance of our method with larger-scale comparisons to more recent methods, using benchmark datasets such as in [21]. These two plans are all addressed in our recent work [25] (Section 3.3).

3.3 Symmetric Flip Problem With A Solution

In this section, we introduce the symmetric flip problem that is inherent to multiresolution isometric shape matching algorithms along with a solution [25] verified on two such matchers that work in coarse-to-fine fashion, namely [24] (Section 3.2) and GMDS [38].

3.3.1 Introduction

We add the following literature review on symmetric flip problem to the dense correspondence literature review in Section 3.2.1 that already applies to this section where we describe a dense isometric shape matching algorithm.

A shape that exhibits intrinsic symmetries yields to more than one optimal solutions while matching it to its articulated pose or to itself. One can exploit the latter, i.e., self-matching, to detect intrinsic symmetries or self-similarities [89, 102, 103].

Whether known a priori or not, intrinsic symmetries cause the so-called symmetric flip problem especially severely for multiresolution shape matchers if precautions are not taken. In [24], the isometric dense correspondence problem is solved by using a multiresolution strategy. A minimum distortion mapping is searched in the original 3D Euclidean space via coarse-to-fine combinatorial matching, hence as free of embedding errors. A coarse-to-fine approach is employed also in [38, 34] to speed up the optimization process and to improve accuracy. A major drawback of coarse-to-fine shape correspondence algorithms in general is the symmetric flip problem due to few number of samples to be matched at coarse levels (e.g., left arm/leg is matched to right arm/leg between two human shapes). This section essentially addresses this initial symmetric flip problem by extending the previous work presented in [24] (Section 3.2). Our key idea is to keep track of all the optimal solutions, which may be more than one due to symmetry especially at coarse levels, throughout denser levels of the shape matching process. This becomes possible thanks to the coarse-to-fine combinatorial matching process involved in the method described in [24], which can sort out all possible mappings with respect to their isometric distortions. We test the performance of our method with extensive experiments on several shape benchmarks in comparison to two state of the art methods mentioned above, namely the GMDS method [38] and the BIM (blended intrinsic maps) method [21], where the former needs further interpolation to achieve dense correspondence yet the latter has several drawbacks due to various approximations they use. First, the isometric distortion of each blending map is estimated on the extended complex plane to which the surfaces to be matched are transformed (flattened) via mid-edge uniformization, hence with some embedding error. Second, geodesic centroids of the blending maps are approximated with Euclidean distances. Third, confidence weights of the blending process are computed over all vertices of a dense surface mesh via interpolation. These approximations may lead to inaccurate matches (see Fig. 4.24 for an example). Moreover the method has restrictions on topology, suffering from non-delaunay triangles during the process of mid-edge uniformization and also requiring zero-genus meshes as input.

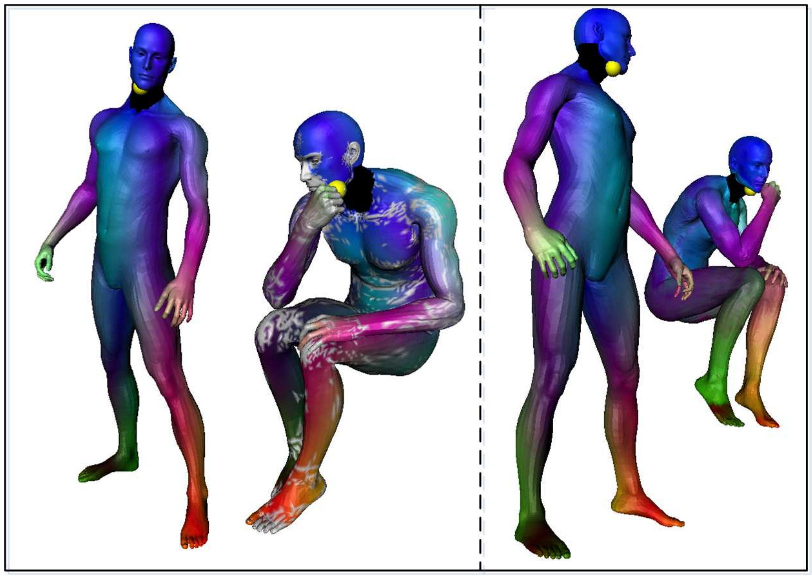

Figure 3.29: Our method vs. BIM. The yellow vertex on the face of the source human model is mapped to the hand of the target by BIM (left) and to the face by our method (right). Note also the unmatched target vertices (in grey color) on the BIM result, which do not exist in our case.

The correspondence algorithm in this section has two main contributions. First, it addresses the symmetric flip problem which is actually inherent to all coarse-to-fine correspondence algorithms, and extends the dense isometric shape correspondence method proposed in [24] in that respect. Our key idea of tracking symmetric flips can also be considered as a meta-approach which can be applied to other multiresolution shape matching algorithms such as [38], as we demonstrate by experiments. Second, the extended method is tested on several shape benchmarks and compared with two state of the art techniques.

3.3.2 Coarse-to-fine combinatorial matching

The method proposed in [24] (Section 3.2) provides an isometric dense shape correspondence algorithm based on combinatorial search. We will refer to this method as C2FCM (coarse-to-fine combinatorial matching). The idea is to reduce the search space by exploiting the fact that the optimal correspondence maps nearby vertices on the source shape to nearby vertices on the target. Hence shape matching is performed efficiently on a patch-by-patch basis in a coarse-to-fine manner.

The C2FCM algorithm is composed of three basic tasks at each level of detail: sampling, combinatorial matching and merging (see Fig. 3.30). Each of the shape surfaces to be matched is initially regarded as a root patch on which evenly-spaced high-curvature M base vertices are sampled. Each of these base vertices defines a surface patch around itself with a sampling radius. Once the samples, hence the patches, are matched via combinatorial matching by evaluating the isometric distortion of all $M!$ possible mappings, the matched parent patches are recursively subdivided into smaller sub-patches as levels of detail increase, by further sampling M base vertices on each parent. At each level, the samples on each pair of corresponding patches are separately matched via combinatorial matching and then merged into one single correspondence. The number M is chosen to be small enough ($M \approx 5$) so that combinatorial matching becomes possible. We note that the algorithm does not require the initial set of samples to include all the shape extremities; as the levels of detail proceed, the samples corresponding to high-curvature vertices are gradually populated and the correspondence accuracy is improved to the extent that the precision of the current level allows, as explained in detail in [24]. At the finest level, a complete dense correspondence, which is optimal in the minimum-distortion sense, is obtained, that matches every vertex in either of the shapes with at least one vertex on the other. Hence the final correspondence is usually a many-to-many mapping. Although the algorithm is built upon the basic assumption that the shapes to be matched are perfectly isometric, the experiments conducted show that it performs well also on nearly isometric shapes. We note that the method can handle input meshes with arbitrary genus.

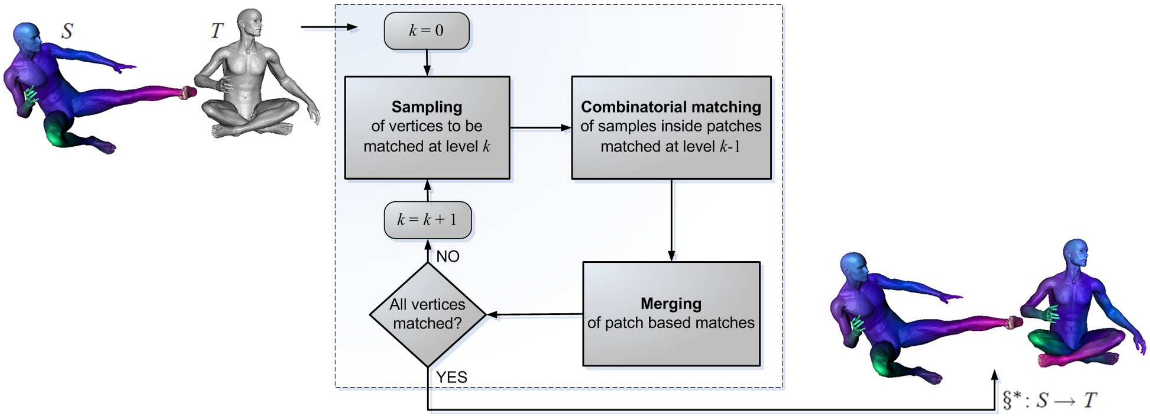

Figure 3.30: Block diagram of the C2FCM algorithm [24].

3.3.3 Symmetric flip problem

Given a mapping $\S : S \rightarrow T$ between two vertex sets S and T sampled from source and target shapes, respectively, we measure its isometric distortion D_{iso} via Eq. 3.1. The correspondence problem can then be formulated as a combinatorial search for an optimal $\S^*$ over all possible mappings, that minimizes this isometric distortion function. One key observation here is that one can find more than one optimal mapping for symmetrical objects: the true correspondence plus the flipped versions. In the case of coarse-to-fine approach, the symmetric flip problem becomes even more severe due to few number of samples to be matched at coarse levels; a coarsely sampled version of a shape may sometimes appear to be symmetric although the shape is not symmetric when considered at a finer resolution as illustrated in Fig. 3.31. Especially the initial level of the matching process in the C2FCM algorithm may exhibit this configuration, and therefore is liable to symmetric flips which cannot be recovered as levels proceed. We note that all purely-isometric shape correspondence methods existing in the literature actually suffer from this symmetric flip problem regardless of the correspondence density since shapes may naturally exhibit intrinsic symmetries. In the coarse correspondence method of [35] for instance, the symmetric flip problem is cast to a sign ambiguity problem associated with the eigenfunctions of the Laplace-Beltrami operator to explicitly seek for all possible mappings which are equally or approximately optimal in terms of isometric distortion.

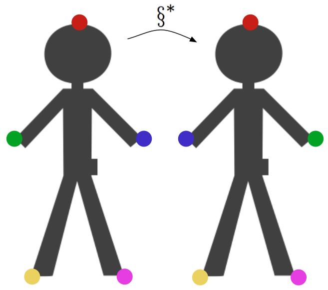

Figure 3.31: Symmetric flip problem due to coarse sampling: Although the shapes to be matched are not perfectly symmetric, the sampled vertices at the first level appear to be symmetric in terms of geodesic distances. This may yield a flipped coarse correspondence result as shown, which is actually one of the four optimal solutions.

In the C2FCM algorithm, symmetric flips may theoretically appear at any level since base vertices at each level are matched patch by patch, where each patch consists of only about $M = 5$ samples. However, since the C2FCM method expands the correspondence list $\S'$ used in computation of the distortion in Eq. 3.1 at each intermediate level with the known correspondence pairs computed at the previous level, the chance of having flipped correspondences is greatly reduced at intermediate levels, as illustrated in Fig. 3.32. This expansion is not possible at the first level, and the correspondence list $\S'$ in Eq. 3.1 can thus simply be set to $\S - \{(s_i, t_j)\}$.

3.3.4 Tracking symmetric flips

The discussion in Section 3.3.3 basically suggests that by avoiding the symmetric flip problem only at the initial level, we can greatly reduce the chance of having a flipped dense correspondence. We propose to resolve any such ambiguities due to coarse sampling by tracking all the optimal (or near-optimal) solutions of the first level (optimal in the minimum distortion sense) throughout intermediate levels of the shape matching process. This is possible thanks to the combinatorial matching algorithm that can sort all possible mappings with respect to their distortion values. The algorithm is described in the sequel.

We start by launching the C2FCM method at the first level (level 0), i.e., via combi-

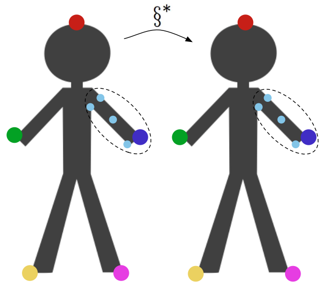

Figure 3.32: The C2FCM method handling symmetric flip problem at intermediate levels. Suppose that the mapping obtained at the first level is free of symmetric flips. Then at the next level, while matching the base vertices inside the highlighted patches (cyans and blue), the isometric distortions are computed by taking into account also the correspondences obtained at the previous level (other matching colors), which usually resolves the matching ambiguities due to symmetry.

natorial matching of the initial M samples. While the original algorithm would pick and pursue only the minimum-distortion mapping of the first level, we keep track of top-N minimum-distortion mappings until some level K_1, where the number of matched vertices is sufficiently large, which occurs typically at level 4 with about 250 vertices. Hence for each of these N initial mappings, we separately pursue the C2FCM algorithm and compute the corresponding denser mapping at level K_1. The choice of N is automatic since we choose it as the first significant jump in the ascendingly sorted sequence of distortion values $D_1, D_2, .., D_L$ of all possible mappings $\S_1, \S_2, .., \S_L$, respectively, where distortions are computed via Eq. 3.1 with $L = M!$. The first instance on this sorted distortion sequence, where the condition $D_{N+1} - D_N > \delta$ holds, reveals the desired jump index N which is typically 4 for human shapes. We set the threshold δ as twice the average of the first 10 distortion difference values: $\delta = 2(\sum_{l=2}^{11}(D_l - D_{l-1}))/10$.

This tracking process along with the distortion curve is illustrated in Fig. 3.33–left for a pair of human shapes. Among the four mappings which are tracked until level $K_1 = 4$, the blue one ($\S_2$) wins the race, having the least isometric distortion at this intermediate level, and is to be maintained on its own until the final level that gives the dense map. Note that, with the original C2FCM method, the mapping $\S_1$ (in green) would continue alone and eventually converge to a flipped result in the absence of any other flipped competitors. We also observe that $\S_1$ yields flipped hands and legs, whereas $\S_4$ (cyan) has correct hands but flipped legs. The mapping $\S_6$ (orange), that is not actually tracked due to jump condition, would yield hand-to-head matching and flipped legs. Another demonstration of the tracking process is given in Fig. 3.33–right for a pair of glass shapes, where N comes out to be 2, a common value for the glasses class.

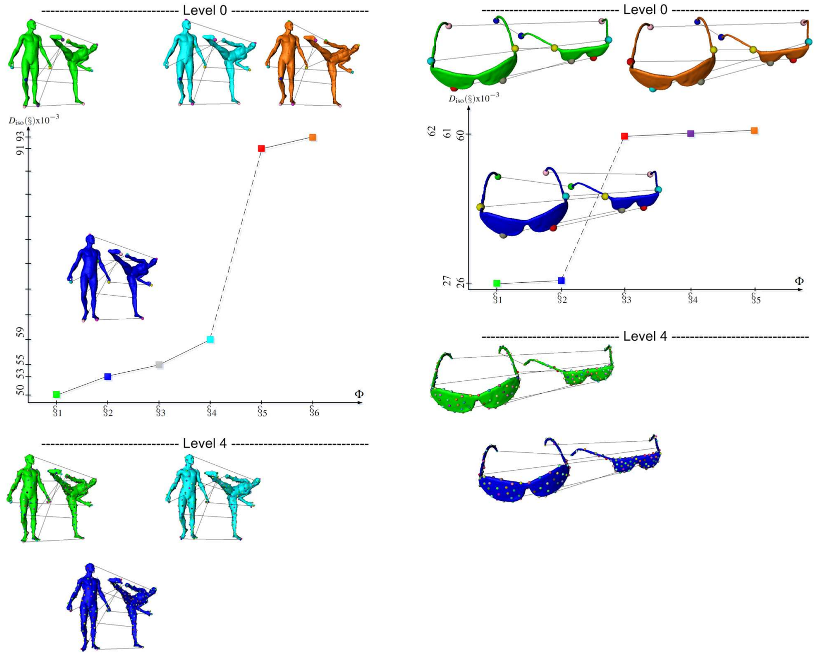

Figure 3.33: (Left) The distortion plot for the top 6 mappings at the first level along with the visualizations of $\S_1, \S_2, \S_4$ and $\S_6$ (Top). The mappings that $\S_1, \S_2$ and $\S_4$ lead to at level $K_1 = 4$ (Bottom). Spheres of matching colors and lines indicate the correspondence pairs (all lines are not drawn at level 4). The mapping $\S_6$ (orange) is not tracked since its distortion value appears after the first significant jump. The mapping $\S_3$ is actually tracked but not shown in the figure for visual convenience. (Right) Similar symmetric flip tracking for a pair of glasses shapes with the same layout except three out of top 5 mappings at level 0 shown for visual convenience.

3.3.5 Computational complexity

The complexity of the original C2FCM method is $O(V \log V)$, where V is the number of vertices in the original meshes to be matched [24]. In our case, running the C2FCM method N times for tracking purposes up to some level K_1, which is typically $K_1 = 4$, i.e., not up to the finest level where the dense map is obtained, incurs no additional asymptotic cost since N is usually a small number varying based on the intrinsic symmetries of the shapes to be matched, which is for example typically 4 for human shapes. Initial sorting of $M!$ mappings for jump detection also comes free of asymptotic cost for $M \leq 8$ but nevertheless, for practical usage, we perform the sorting only on mappings with isometric distortion below a threshold value (0.15 in our experiments). This significantly reduces the number of mappings (typically to $M' \approx 300$) to sort with no accuracy loss as the true mapping to be tracked will almost certainly be in the top-M' least distorted mappings. In our experiments, it has always remained within the top 6.

Our overall $O(V \log V)$ complexity outperforms that of other isometric dense shape correspondence algorithms in the literature, such as $O(V^2 \log V)$ complexity of [39] and GMDS [38], $O(V^2)$ complexity of HKM [33], $O(V^2 \log V + Y^4 \log Y)$ complexity of Möbius Voting [31]. Note that, GMDS and Möbius Voting can achieve a coarse correspondence between Y feature points which can later be interpolated into a dense one with no additional asymptotic cost. Blended intrinsic maps of [21] is faster than these methods with $\max(O(V \log V), O(W^2))$ complexity with W being an upper bound on the number of maps considered.

3.3.6 Experimental results

We have employed a comprehensive test suite consisting of five well-known 3D shape benchmark datasets: TOSCA [18], Watertight [104], SHREC'11 [105], SCAPE [106], and Non-rigid World [38]. Our results are also compared with two state-of-the-art techniques, namely GMDS [38] and Blended Intrinsic Maps (BIM) [21] by running their publicly available codes.

We provide symmetric flip ratios resulting from each dataset in Tables 3.6, 3.9, 3.11 and 3.13. We also demonstrate numerical (Tables 3.7, 3.8, 3.10, 3.12, and 3.14) and visual (Figures 3.34-3.38) quality of our final dense maps in comparison with the two methods. We measure isometric quality of the final maps by D_{iso} using Eq. 3.1, whereas ground-truth distortion is measured by D_{grd} (Eq. 3.12) whenever ground-truth correspondence pairs are available as (s_i, t_i) $\forall i$. We also employ two additional worst-case distortion measures, the maximum ground-truth distortion $d^{\dagger}_{\text{grd}}$ (Eq. 3.13) and the maximum isometric distortion $d^{\dagger}_{\text{iso}}$ (Eq. 3.11). As usual, in computation of the performance measures, we use normalized geodesic distances so that the max geodesic distance on a shape surface is taken as 1.0.

As a convention that applies to all figures, for each dataset and a particular shape class, we display the mapping on the pair for which the method whose $d^{\dagger}_{\text{iso}}$ (or $d^{\dagger}_{\text{grd}}$) is worse than the other method for that particular class; a † sign emphasizes this interesting mapping. The same pair is also shown with the mapping computed by the other method – expected to be better. In all figures, the dense mapping is visualized by transferring source vertex colors to the target through computed correspondences.

Finally, there is also a downsampled version of each dataset that we use, with meshes of size at most 5K to be able to run the GMDS method which does not scale well to large meshes.

3.3.6.1 Dataset: TOSCA

The original TOSCA dataset is explored under three classes: within animals (11 cats and 8 horses), within humans (two different males with 12 and 20 poses each, and 12 females), and across humans (using the three human classes above). The within-animal results are obtained by matching every cat (or horse) to a different randomly selected cat (or horse). The same applies to within-human results. Likewise, in the across-human experiments, each human model in the dataset is matched to another model randomly selected from a different human class.

In Table 3.6 - rows 2-4, we see for instance that $100\% - 38\% = 62\%$ of across-human tests start with an initial symmetric flip at level 0, which means that if we had employed the original C2FCM algorithm, then 62% of the resulting dense maps would suffer from being symmetrically flipped. With our method, on the other hand, only $100\% - 78\% = 22\%$ of all correspondence results is symmetrically flipped at the end. Another interpretation is that 62% initial symmetric flips reduce to just 22% in the final result, an improvement of 40%. The BIM method [21] is 95% free of symmetric flip errors for this class, and outperforms our method for all other classes as well in this set as far as the symmetric flip problem is concerned.

Dataset: TOSCA	% of all results		
	A	B	C
C2FCM	59	61	38
Our method	69	68	78
BIM	100	100	95
C2FCM	37	60	67
Our method	58	75	83
GMDS	37	54	39

Table 3.6: Percentage values for dense correspondence results without symmetric flips, obtained with different methods. A: Within animals, B: Within humans, C: Across humans. Last three rows show the results on downsampled versions of classes A to C.

For isometric and ground-truth distortions in this dataset, we observe in Table 3.7 that $d^{\dagger}_{\text{iso}}$ and $d^{\dagger}_{\text{grd}}$ values favor our method for within and across humans as those classes may include pairs with touching surface parts, e.g., hand-to-breast, which are confused by BIM that approximates geodesic centroid as a weighted Euclidean centroid (see Fig. 3.34 and also Fig. 4.24). As for the average values D_{iso} and D_{grd}, our method is on a par with the results of BIM.

Class	Our method		BIM	
	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{grd}}, d^{\dagger}_{\text{grd}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{grd}}, d^{\dagger}_{\text{grd}}$
A	.018, .081	.030, .226	.018, .071	.018, .155
B	.015, .052	.033, .202	.013, .266	.021, .417
C	.019, .065	n/a	.017, .068	n/a

Table 3.7: Isometric and ground-truth distortions obtained on TOSCA with our method as compared to BIM. A: Within animals, B: Within humans, C: Across humans.

We run the GMDS method of [38] on a downsampled version of TOSCA database, using random test pairs chosen in the same way as described previously. The number of vertex samples to be matched on the decimated models is set to be 150, a parameter of the publicly available code. The 150 correspondence pairs, which are obtained with GMDS for each shape pair, are then used to interpolate a dense map as described in [21].

The resulting symmetric flip ratios (Table 3.6 - rows 5-7) as well as average isometric distortion values (Table 3.8) all favor our method in this dataset when compared to

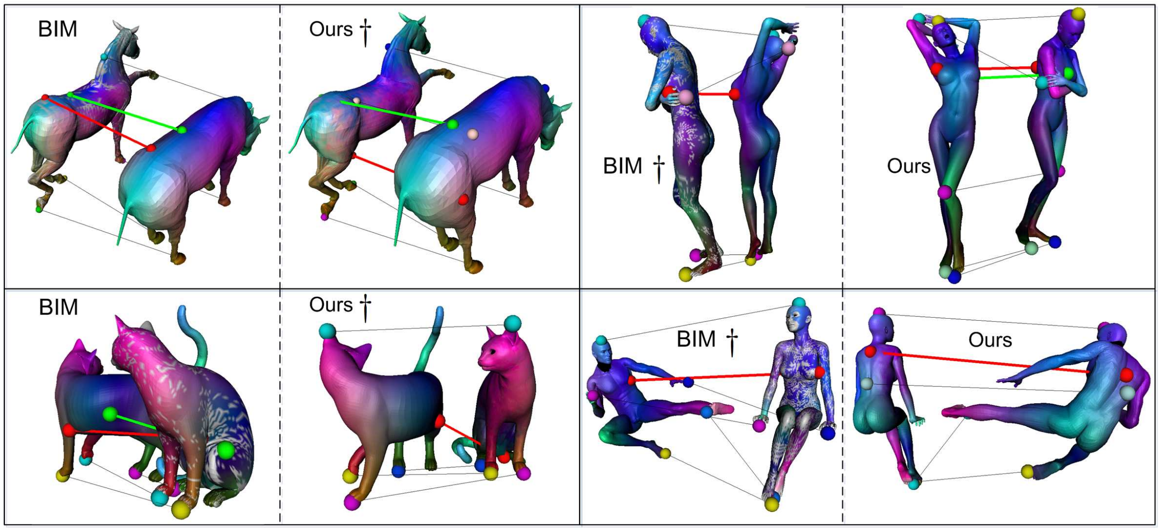

Figure 3.34: Dense correspondence results on TOSCA, obtained with our method vs. BIM. The red and green lines indicate the worst matches w.r.t. isometric and ground-truth distortions, respectively. The worst matchings are in general better in our case when compared to BIM. Notice for example hand-to-breast and breast-to-arm matches (third column) as well as the unmatched regions shown in grey for BIM results. In addition to transferred colors, some lines ending with spheres of matching colors are used to enhance visuals. When red and green lines overlap, only red is shown. This whole representation scheme applies to the subsequent figures of this section as well.

GMDS. Our maximum distortion values, however, are slightly worse on human test pairs. Nevertheless, we see that 150 vertices matched with GMDS are not evenly distributed on the shape surfaces, which results in clustered correspondences (Fig. 3.35), hence one of the major shortcomings of the GMDS method. This is mainly due to the fact that the GMDS algorithm employs a gradient-based iterative optimization process that produces sub-vertex matchings which do not necessarily coincide with the initial sampling. We also note that, while computing the resulting isometric distortion value for a GMDS output, we round sub-vertex coordinates to their nearest vertices on the surface mesh. The effect of this rounding process to the computed distortion values is negligible since the mesh models in this, and all other datasets are almost uniform and at relatively high resolution. We also observe from the visual comparison that the GMDS matching often misses the salient points of a shape (e.g., ear tips of the cats).

	Our method	GMDS
Class	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
A	.024, .045	.017, .051
B	.018, .049	.022, .043
C	.020, .081	.026, .057

Table 3.8: Isometric distortions obtained on TOSCA downsampled with our method as compared to GMDS. A: Within animals, B: Within humans, C: Across humans.

3.3.6.2 Dataset: Watertight

The part that we have used from the original dataset consists of human, glasses, chair, teddy bear, hand, fish, and armadillo classes of cardinality 20 each. We have omitted ant and octopus classes which are too symmetric to handle in the sense that the shapes in these classes do not contain sufficient clues to resolve ambiguities due to symmetry even at denser resolution levels. We also exclude the remaining 10 classes since they exhibit severe non-isometries, such as airplanes and four-leggeds, while our method, as well as GMDS, has been designed to work on isometric shapes. In these experiments, an object fetched from a class is matched to a random object from the same class.

Concerning symmetric flips (Table 3.9 - rows 2-4), our method is as good as BIM which cannot however handle the chair class with nonzero genus objects and is slightly worse than

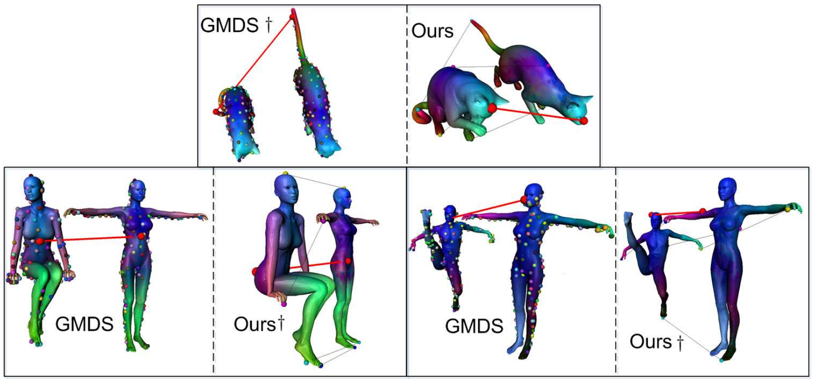

Figure 3.35: Dense correspondence results on TOSCA collapsed, obtained with GMDS (left) and with our method (right). Generating 150 correspondence pairs for GMDS are shown as spheres of matching colors.

our method on isometric distortion measurements (Table 3.10 - column 2-3).

As the flow of the transferred colors shows (Fig. 3.36 - top), our dense map is not always as smooth as the one generated by BIM which interpolates the feature correspondences in the extended complex plane by nearest neighbor search to generate the conformal maps to be blended. These smooth conformal maps are obtained at the expense of leaving a fair amount of target vertices unmatched (grey) even after blending. Our final map, on the other hand, is always onto, i.e., no vertices are left unmatched.

Dataset:	% of all results						
Watertight	A	B	C	D	E	F	G
C2FCM	28	63	30	50	50	69	60
Our method	**89**	**88**	70	**67**	**88**	**88**	**90**
BIM	**89**	**57**	n/a	**83**	**100**	**85**	**89**
C2FCM	61	29	69	42	50	43	37
Our method	**83**	**71**	66	**50**	**83**	**86**	**75**
GMDS	**28**	**40**	n/a	**33**	**50**	**71**	**25**

Table 3.9: Percentage values for dense correspondence results without symmetric flips, obtained with different methods. A: Humans, B: Glasses, C: Chairs, D: Teddy bears, E: Hands, F: Fishes, G: Armadillos. Last three rows show the results on downsampled versions of classes A to G.

Downsampled Watertight models, when fed into GMDS in the way described in Section

	Our method	BIM	Our method	GMDS
Class	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
A	.022, .109	.035, .121	.021, .053	.031, .079
B	.008, .020	.063, .248	.007, .015	.034, .147
C	.024, .109	n/a	.022, .022	n/a
D	.032, .108	.048, .225	.029, .081	.032, .119
E	.034, .138	.046, .167	.027, .049	.029, .134
F	.033, .140	.061, .192	.024, .079	.024, .069
G	.030, .185	.049, .193	.031, .098	.039, .135

Table 3.10: Isometric distortions on Watertight dataset, obtained with different methods. A: Humans, B: Glasses, C: Chairs, D: Teddy bears, E: Hands, F: Fishes, G: Armadillos. Fourth and fifth columns give the results on downsampled versions of classes A to G.

3.3.6.1, are significantly more prone to symmetric flips than our method (Table 3.9 - rows 5-7). Initial GMDS matchings suffer from the clustering issue (Fig. 3.36 - bottom), and when interpolated into a dense map, produce worse results as compared to ours in terms isometric distortion (Table 3.10 - column 4 and 5). Note also that, although a small topological noise connecting index and middle fingers does not fail our method (see Fig. 3.36), noisy connections that alter geodesic distances more severely may cause instabilities that are not expected in diffusion-based methods such as [33].

3.3.6.3 Dataset: SHREC'11

We employ SHREC'11 dataset in order to see to what extent our algorithm tolerates noise and holes of various sizes. Due to zero genus constraint of the BIM method, only noise and shotnoise classes are used for the corresponding comparison suite. The GMDS public code, on the other hand, crashes on all test cases but the shotnoise. The results summarized in Table 3.11 and Table 3.12 generally show that both of our competitors are more sensitive to noise as compared to our method (see Fig. 3.37 as well). Among five different shotnoise levels tested, we fail to achieve stable results for the forth and fifth levels (in increasing order) which possess severe shotnoises.

We give our results on microholes, holes, and viewing classes (which cannot be obtained with BIM and GMDS) in the format of (D_{iso}, $d^{\dagger}_{\text{iso}}$), as (.013, .059), (.022, .058), and (.016, .068), respectively (Fig. 3.37 - bottom).

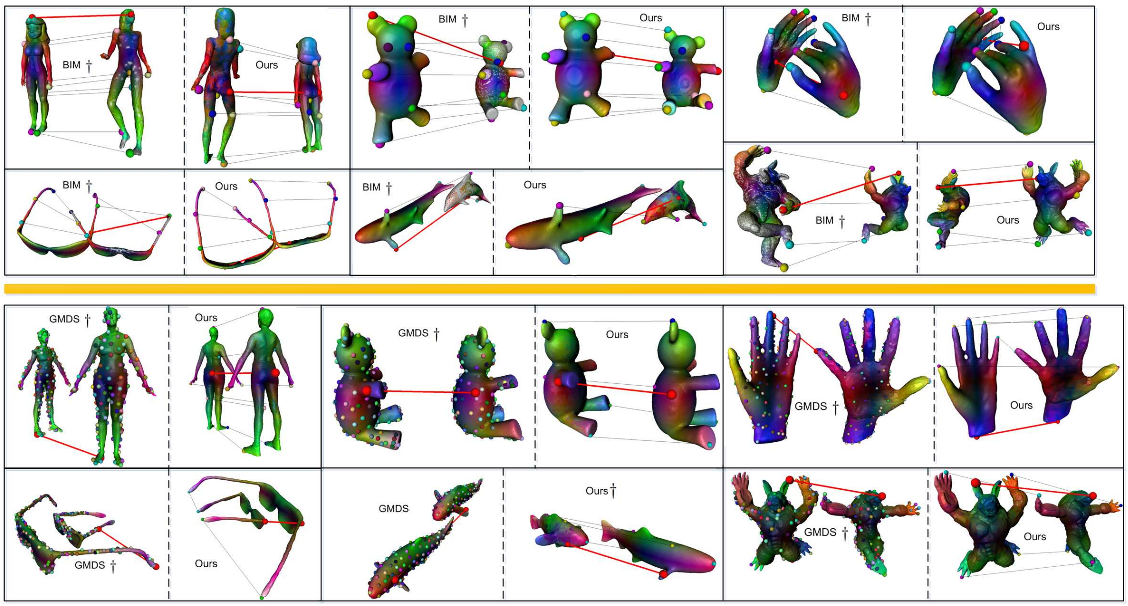

Figure 3.36: Dense correspondence results on Watertight dataset, obtained with our method vs. BIM (top) and vs. GMDS (bottom, separated by yellow line).

3.3.6.4 Dataset: SCAPE

Each SCAPE model represents the articulated motion of a human actor reconstructed from range data. We match each SCAPE model to a random model from the remainig 71 shapes. Although the BIM method handles symmetric flips better than our method in this dataset (Table 3.13), the mid-edge uniformization phase of their conformal maps to be blended suffers from non-delaunay triangles on this raw data, hence leaving them slightly behind us in distortion performances (Table 3.14) when coupled with weighted Euclidean centroids

Dataset: SHREC'11	% of all results		
	A	B	C
C2FCM	20	87	54
Our method	**80**	**94**	**62**
BIM	**50**	**38**	n/a
GMDS	n/a	n/a	**23**

Table 3.11: Percentage values for dense correspondence results without symmetric flips, obtained with different methods. A: Noise, B: Shotnoise, C: Shotnoise on downsampled set.

	Our method	BIM	GMDS
Class	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$
A	.019, .066	.013, .079	n/a
B	.015, .042	.032, .108	n/a
C	.010, .066	n/a	.026, .074

Table 3.12: Isometric distortions on SHREC'11, obtained with different methods. A: Noise, B: Shotnoise, C: Shotnoise on the downsampled set.

approximation of geodesic centroids for efficiency. The GMDS method is again slightly worse than our method in terms of symmetric flips (Table 3.13) and distortion measurements ($(D_{\text{iso}}, d^{\dagger}_{\text{iso}}) = (.025, .086)$ of our method vs. (.028, .103) of GMDS) mainly due to their clustered generating matches.

Dataset: SCAPE	% of all results	
	A	B
C2FCM	56	24
Our method	**60**	**61**
BIM	**92**	n/a
GMDS	n/a	**42**

Table 3.13: Percentage values for dense correspondence results without symmetric flips, obtained with different methods. A: SCAPE, B: SCAPE downsampled.

Our method		BIM	
$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{grd}}, d^{\dagger}_{\text{grd}}$	$D_{\text{iso}}, d^{\dagger}_{\text{iso}}$	$D_{\text{grd}}, d^{\dagger}_{\text{grd}}$
.019, .097	.045, .308	.027, .254	.039, .306

Table 3.14: Isometric and ground-truth distortions on SCAPE dataset, obtained with our method in comparison to BIM.

3.3.6.5 Application to GMDS

In addition to C2FCM [24] on which our symmetric flip tracking approach has been verified thus far, we now show how well this method, as a meta-approach, extends to another multiresolution isometric shape matching algorithm, namely GMDS [38], whose optimization method is completely different from C2FCM. To this end, we first run the public GMDS

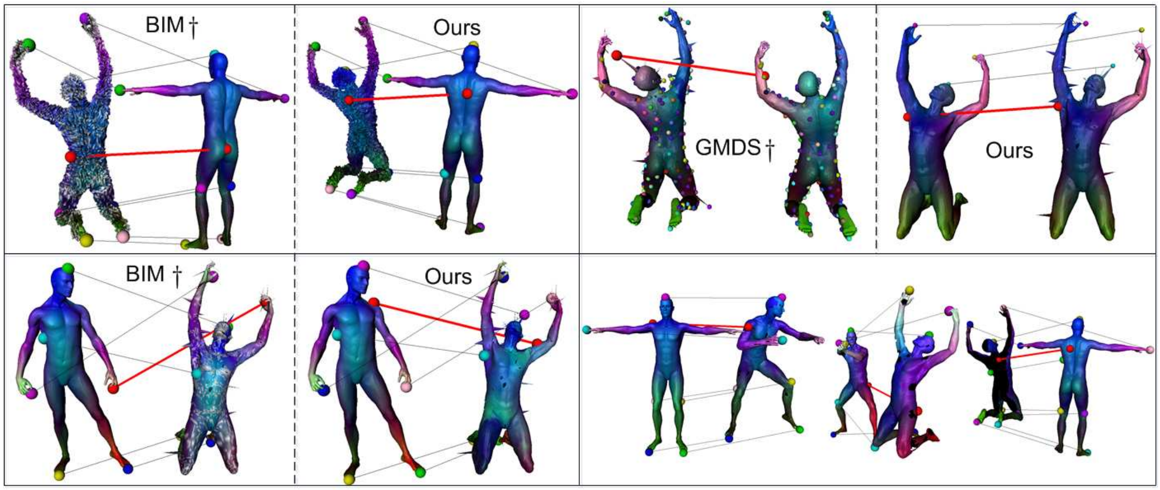

Figure 3.37: Dense correspondence results on SHREC'11. Sample pairs for noise, shotnoise, and downsampled shotnoise classes (top to bottom), obtained with our method (right) and with BIM and GMDS (left). Our result samples for microhole, hole, and viewing classes are displayed separately at the bottom (left to right).

code on a part of the Nonrigid World database that contains 8 cats, 6 centaurs, and 17 horses (for within animals class) as well as 10 male and 23 female figures (for within and across humans classes), where the classes are formed in the same way as described in Section 3.3.6.1. We then slightly modify the original public code so as to initialize GMDS with the qualified maps generated at the first level of our method where qualification is based on the jump detection scheme described in Section 3.3.4. Amongst the qualified maps that are tracked until a relatively dense GMDS level that matches 250 vertices, we select the one yielding the minimum GMDS distortion as the final map. In Table 3.15, we provide the symmetric flip ratios on the final maps, which clearly favor the latter GMDS modified by our tracking method (last row). We depict the tracking process in Fig. 3.39 along with the corresponding distortion curve, where the number of tracked (qualified) maps is 8, a common value for centaurs having three different pairs of limbs.

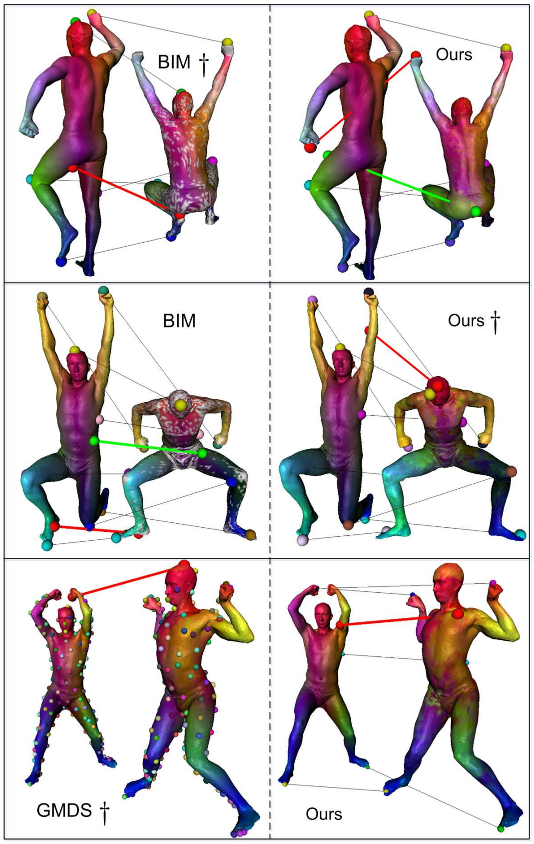

Figure 3.38: Dense correspondence results on SCAPE. BIM yields the worst $d^{\dagger}_{\text{iso}}$ (top row, red line), our method the worst $d^{\dagger}_{\text{grd}}$ (middle row, green overlapped by red), and GMDS the worst $d^{\dagger}_{\text{iso}}$ (bottom row) on SCAPE downsampled.

3.3.6.6 Timing

With $O(V \log V)$ time complexity, our algorithm scales well to large meshes such as TOSCA and SHREC'11 humans of 53K vertices, TOSCA animals and Watertight armadillos of 28K vertices, and SCAPE humans of 13K vertices. The tracking process for symmetric flips (Section 3.3.4) is quite fast; on the largest pair of 53K vertices, a 2.53GHz 64-bit workstation takes 10 seconds to evaluate $M!$ distortions of maps of sizes M each via D_{iso} (Eq. 3.1) for $M = 8$ and sort about 300 of them. This is followed by about 48 seconds tracking of top-N (typically ≤ 4) mappings until level K_1 (typically 4). The original C2FCM method then continues to bring the qualified mapping to a dense map in 1578 seconds. For a SCAPE

Dataset: Nonrigid World	% of all results		
	A	B	C
GMDS	19	42	47
GMDS modified with tracking	61	70	67

Table 3.15: Percentage values for correspondence results without symmetric flips, obtained with the original and modified GMDS methods. A: Within animals, B: Within humans, C: Across humans. Last three rows show the results on decimated versions of classes A to C.

pair with around 13K vertices, these evaluation-sorting, tracking, and dense mapping times become 2, 13, and 391 seconds, respectively, whereas for a downsampled shape pair, e.g., from TOSCA, with 5K vertices, the respective execution times are 1, 4, and 66 seconds.

3.3.6.7 Discussion

We summarize our comparative experimental findings as follows:

- In terms of symmetric flip ratio, our method is always better than GMDS which is, as a coarse-to-fine solution, prone to symmetric flips at the initial levels with yet no care taken. While BIM performs as good as our approach on Watertight models, it is worse for SHREC'11 models that are exposed to various types of noises. The results on TOSCA and SCAPE models favor BIM as the conformal maps being blended are area-preserving at every point which proves useful in distinguishing mapping the front of a human to the back.

- In isometric distortion performance, which goes parallel to ground-truth performances (whenever ground-truth correspondences are available), our method outperforms GMDS for all datasets mainly because of the clustered matchings of the latter, that are interpolated into a dense map. The BIM method, on the other hand, is on a par with ours on TOSCA and slightly worse for all other datasets concerning average distortions as the conformal maps to be blended induce embedding errors, and more importantly confidence weights of the blending process are computed over all vertices of a dense surface mesh via interpolation. Recall that our algorithm is embedding and interpolation free. Moreover, the BIM yields unstable results on the datasets that include shapes

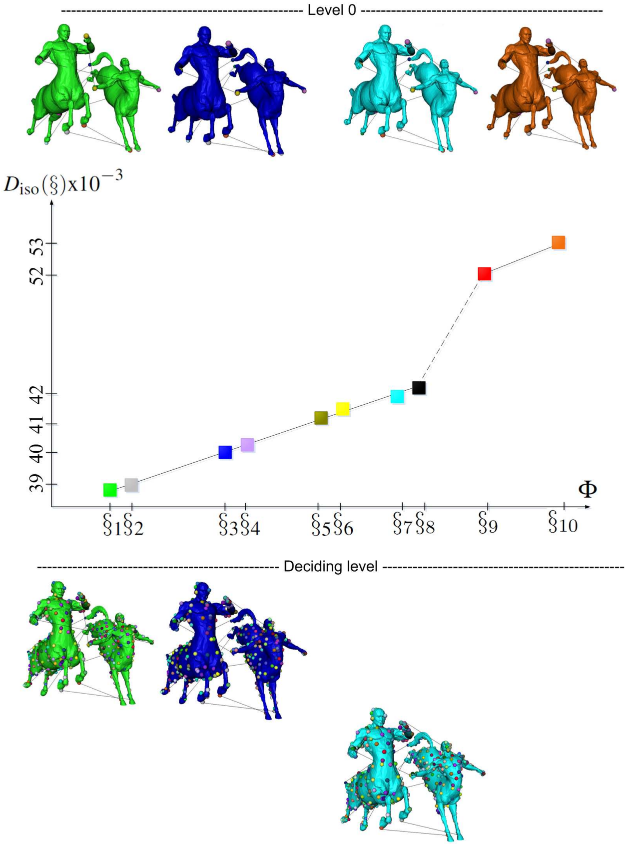

Figure 3.39: (Top) The distortion plot for the top 10 mappings at the first level along with the visualizations of $\S_1, \S_3, \S_7$ and $\S_{10}$. (Bottom) The GMDS mappings that $\S_1, \S_3$ and $\S_7$ lead to at the dense GMDS level where the map generated by $\S_7$ (cyan) wins the race having the least distortion and $\S_{10}$ (orange) is not even tracked.

with touching surface parts since Euclidian centroid is replaced with the geodesic one, with worst distortion behavior significantly behind ours.

- In the case of GMDS, the uniformity of the evenly-spaced initial samples is not necessarily preserved in the final correspondence output, hence matching salient shape points such as ear tips is not guaranteed. Besides, the algorithm does not support dense matching without interpolation that incurs more errors. We usually hit salient points at intermediate levels and yet support dense matching without need for any interpolation.

- Sphere topology restriction of the BIM as well as its sensitivity to peculiarities of a given triangulation, e.g., non-delaunay triangles, are avoided in our approach (see especially the experimental results on SHREC'11 and SCAPE datasets).

- The conformal maps in BIM are not usually onto after closest point matching in the embedding domain, which in turn leaves a significant amount of target vertices unmatched even after blending. Further interpolation would be required to obtain the full dense mapping that we produce, which is many-to-many. The interpolated conformal maps in BIM, however, generally leads to smoother dense maps than ours.

- Our symmetric flip tracking approach, when applied to the multiresolution GMDS method, significantly reduces the percentage of the correspondence results with symmetric flips.

- By tracking, we significantly reduce the occurrence rate of symmetric flips as an end result, yet cannot completely solve the problem. In our experiments, in almost all cases where the symmetric flip problem cannot be resolved, the true solution is included in the initial set of qualified mappings, but does not yield the minimum-distortion after tracking (further increasing the resolution of the intermediate dense level does not help either). This is mainly due to two reasons. First, in the case of perfectly isometric and intrinsically symmetric shapes, even at dense resolutions, there are actually more than one correspondence, yielding distortion values which are all close to the global minimum but possibly with some inaccuracy due to discretization and modeling errors. Second, the shapes to be matched are often only nearly isometric while the multiresolution correspondence algorithms that we have tested are designed to work on perfectly isometric shape pairs. For the rare cases where the true correspondence does not even appear in the initial set of maps to be tracked, one possibility could be to seek for local minima of the distortion function, that would reveal the approximate symmetries more accurately, as suggested in [103].

3.3.7 Conclusion

We have extended our original efficient dense isometric shape correspondence algorithm [24] (Section 3.2) so as to address the symmetric flip problem. To this effect, we couple the coarse-to-fine structure of the original algorithm with a tracking mechanism that brings some candidate initial maps into a finer resolution where their isometric distortions can be more accurately computed. Maintaining only the best one at this stage generally reduces the symmetric flips in the final correspondence, as demonstrated on five well-known benchmark 3D shape datasets in comparison with two state-of-the-art techniques. Our final dense maps are also better than or on a par with our competitors as far as the final isometric and ground-truth distortions are concerned. We perform the isometric matching in the original 3D Euclidean space wherein isometry is defined, and hence free of embedding errors. Our dense map is also interpolation-free as the combinatorial matching continues efficiently until all vertices are explicitly matched. Note also that, this tracking-based solution can be used until a desired coarseness level of correspondence depending on the application or the method to be compared with. The core idea of tracking symmetric flips (Section 3.3.4) can be adapted to other techniques affected by the same issues (Section 3.3.6.5) as well as to future multiresolution isometric shape matching algorithms. We finally note that the proposed method can be used for symmetry detection by seeking optimal (or near-optimal) maps from a shape to itself in the same spirit as other symmetry detection methods available in the literature such as [103, 89]. As future work, we plan to make the framework available for partially isometric shapes.

Chapter 4

PARTIAL SHAPE CORRESPONDENCE

We propose methods to match at coarse and dense resolutions two semantically similar shapes that may have multiple common parts at arbitrary scales as well as parts that are not similar. Since scale normalization in this scenario is not as trivial as that in complete matching, we first focus on this issue in a particular setting where one shape is the isometric (or nearly isometric) part of the other up to an arbitrary scale [26] (Section 4.1). We then handle a more general setting where shape pairs to be matched are partially isometric but both having parts which are not in common [27] (Section 4.2). For both methods, initial correspondences that are all sparse at the shape extremities are extended to dense ones where the notion of density is more relaxed in this section, i.e., not all vertices, but relatively high number of vertices. Note also that both methods support complete matching naturally.

Our main ingredient for [26] (Section 4.1) is the novel third-order scale-invariant isometric distortion measure whereas for [27] (Section 4.2) we cast votes based on the dense sampling on the areas defined by isolated triplets of samples, hence free of intervention from irrelevant parts.

4.1 Scale Normalization for Isometric Shape Matching

In this section, we address the scale problem that is inherent to isometric shape correspondence and apparent in partial shape correspondence [26].

4.1.1 Literature review

As we already know by now, isometry is an important clue in matching similar or semantically equivalent surface points from two shapes since similar shape parts usually have similar metric structures. Although shape matching can be achieved by enforcing geodesic consistencies or by searching for mappings with minimum isometric distortion, the arbitrary scale of shapes usually poses an important challenge to overcome for partial or complete shape correspondence, as we address in this section.

There are different ways of dealing with the scale problem in the partial and/or complete shape correspondence literature. Some methods simply assume that the shapes come in compatible scales, which is rather a very strong assumption, but that might sometimes prove useful especially in the case of matching 3D scan data [38, 42, 32, 29]. Other methods are feature-based and rely solely on local shape descriptors which are usually designed to be scale-invariant [28, 19, 58]. Local shape similarity is an important clue for shape correspondence, especially in the case of non-isometric deformations, but otherwise it is considered as less reliable than global shape information such as isometry. They may not perform well for example when the shapes to be matched exhibit large variations in their local geometry, or when there are many points that are locally similar.

If two shapes are isometric (perfectly, nearly, or partially) and come in different scales, in order to be able to incorporate metric similarities, the shapes have to be normalized into the same scale prior to the matching process. There exist two different approaches to achieve this. The first and simpler approach is to scale the original geometry with respect to some global intrinsic property such as maximum geodesic distance [22, 37], maximum centricity [93], or total surface area [33]. This strategy may work satisfactorily in the case of complete shape correspondence but otherwise for partial matching its success depends highly on the global similarity between the shapes. Even in the case of complete correspondence, if the given shapes are only nearly isometric, it is indeed possible that global intrinsic properties significantly deviate from one shape to the other, yielding inaccurate normalization. The second approach is to transform input shapes into a different domain where the scaling problem is implicitly handled. Euclidean embedding is one such transform which is commonly used for isometric shape correspondence, but it is rather a global technique and does not well apply to partial correspondence [39, 77]. A better alternative for partial matching is based on the Möbius transformation that can be used for conformal embedding of the given shapes into a canonical coordinate frame on the complex plane [31, 36], where deviations from isometry can approximately be computed based on mutually closest points. The Möbius Voting method in [31] considers the most general setting of the partial correspondence problem where shapes to be matched can also have uncommon (non-similar) parts. This voting-based approach can generate reliable but sparse correspondences, suffering mainly from spurious votes of the extra uncommon parts in the shape models, and hence performs poorly in terms of dense and shape extremity matching. We also note that the Möbius Voting approach is restricted to sphere topology and prone to errors due to embedding approximation.

An alternative to geodesic metric is the diffusion metric which is less accurate for measuring isometric distortion but generally considered as more robust to topological noise. Local scale differences are however difficult to handle using diffusion-based metrics. The commute-time metric for example addresses the scale problem only globally [34], hence cannot be used for the partial matching problem, i.e., in a setting where one of the shapes to be matched includes a scaled part of the other. In [107], a scale-invariant version of the heat kernel signature is used to address the part matching problem, which however requires setting a time scale parameter that itself depends on the shape scale.

In the most general setting of the correspondence problem, the ambiguity of shape scale is a local issue that cannot be resolved relying on some global shape intrinsics. This is demonstrated in Fig. 4.1. While two completely isometric shapes can be brought to the same scale trivially by normalizing their maximum geodesic distances (Fig. 4.1a), scale normalization for partially overlapping shapes (and even for hybrid shapes which are nearly isometric) is not that straightforward (Fig. 4.1b). In the latter case, the scale ambiguity can be resolved based on local geodesic distances computed on a sparse set of trusted correspondences (Fig. 4.1c), which is the approach that we follow in this paper. Alternatively, a small set of trusted correspondences can be used to define an Euclidean embedding, such as Möbius transformation in the extended complex plane [31], that implicitly handles the scale problem (Fig. 4.1d).

In this section, we consider the shape correspondence problem in the particular setting where one of the shapes to be matched is an isometric part of the other up to an arbitrary scale (note that this setting also includes the problem of complete shape matching). We address the scale problem in a combinatorial framework that minimizes a scale-invariant isometric distortion function in the 3D Euclidean space. We first sample shape extremities from the mesh representations of the given shape pair and then find a coarse map in between via combinatorial search, that also extends to dense matching. The output of the proposed

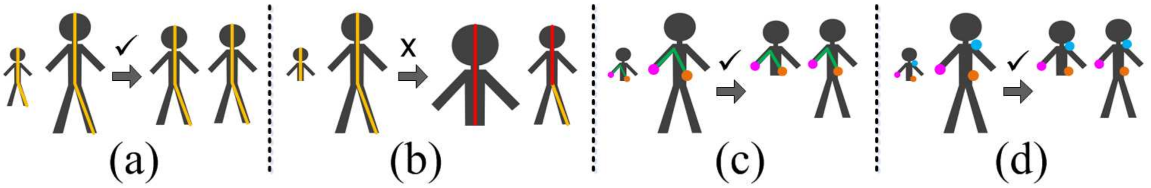

Figure 4.1: (a) Scale normalization of perfectly isometric shapes based on maximum geodesic distances (yellow paths), which does not apply to partially isometric shapes, e.g., normalized distances (red paths) are not the same on both shapes (b). Scale normalization for partial matching can be achieved based on trusted correspondences, either by normalizing (green) geodesic paths (c) or by defining an Euclidean embedding (d).

method is hence a sparse or dense optimal set of correspondences between the surfaces of the given partially or completely isometric shapes.

4.1.2 Isometric distortion

We describe two different isometric distortion measures in the sequel, $D^{(1)}_{\text{iso}}$ and $D^{(2)}_{\text{iso}}$, that can be used interchangeably in our combinatorial shape matching framework. The first measure can be considered as the one that we propose as a novel measure whereas the latter rather serves as a baseline measure to compare with the former. These measures will be compared experimentally later in Section 4.1.7.

4.1.2.1 Scale-invariant isometric distortion

Let S and T represent two sets of points sampled from the given source and target shapes, respectively. Suppose also that a mapping $\S : S \to T$ (a relation in the most general setting) is given. Then the isometric distortion of this mapping can be measured by the scale-invariant function $D^{(1)}_{\text{iso}}(\S)$ as follows:

$$D^{(1)}_{\text{iso}}(\S) = \frac{1}{|\S|} \sum_{(s_i,t_j)} d^{(1)}_{\text{iso}}(s_i, t_j), \tag{4.1}$$

where $d^{(1)}_{\text{iso}}(s_i, t_j)$ is the contribution of the individual correspondence (s_i, t_j) to the overall isometric distortion:

$$d^{(1)}_{\text{iso}}(s_i, t_j) = \frac{1}{\binom{|\S'|}{2}} \sum_{((s_a,t_b),(s_c,t_d))\in\mathcal{C}(\S')} |\rho(s_i, t_j; s_a, t_b) - \rho(s_i, t_j; s_c.t_d)| \tag{4.2}$$

with $\S' = \S - \{(s_i, t_j)\}$ and $\mathcal{C}(\S')$ denoting the set of all pairwise combinations from $\S'$. The ratio function $\rho(s_i, t_j; s_k, t_l)$ is then written in terms of geodesic distances, for a given $(s_k, t_l) \in \S$:

$$\rho(s_i, t_j; s_k, t_l) = \max\left(\frac{g(s_i, s_k)}{g(t_j, t_l)}, \frac{g(t_j, t_l)}{g(s_i, s_k)}\right) \tag{4.3}$$

where $g(.,.)$ is the geodesic distance between two surface points. This definition of isometric distortion is based on the observation that the ratios between geodesic distances on a surface remain unchanged under scaling and isometric deformations. Hence if T and S are sampled consistently from the given arbitrarily scaled (partially) isometric shapes, then one can find an optimal mapping $\S^*$ in between such that $D^{(1)}_{\text{iso}}(\S^*) = 0$. This is illustrated in Fig. 4.2.

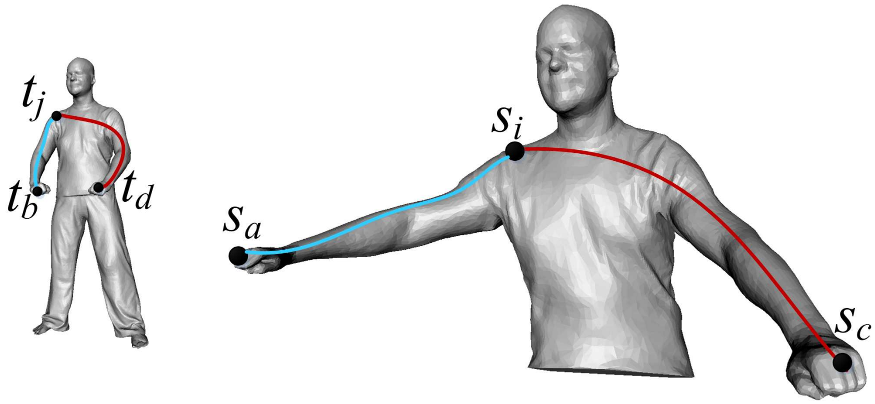

Figure 4.2: Demonstration of the scale-invariant isometric distortion measure defined in (4.1). The ratios between geodesic distances remain invariant under scaling and isometric deformation: $\rho(s_i, t_j; s_a, t_b) = \rho(s_i, t_j; s_c, t_d)$.

4.1.2.2 Isometric distortion with normalized geodesics

The isometric distortion $D^{(2)}_{\text{iso}}(\S)$ for a given map $\S : S \rightarrow T$ between point sets S and T is computed as follows:

$$D^{(2)}_{\text{iso}}(\S) = \frac{1}{|\S|} \sum_{(s_i,t_j)\in\S} d^{(2)}_{\text{iso}}(s_i,t_j) \tag{4.4}$$

where $d^{(2)}_{\text{iso}}(s_i,t_j)$ is the contribution of the individual correspondence (s_i,t_j) to the overall isometric distortion:

$$d^{(2)}_{\text{iso}}(s_i,t_j) = \frac{1}{|\S'|} \sum_{(s_l,t_m)\in\S'} |g_{\text{n}}(s_i,s_l) - g_{\text{n}}(t_j,t_m)| \tag{4.5}$$

where $\S'$ is the list of correspondences to be traversed, which is set to be as $\S' = \S - \{(s_i,t_j)\}$ unless stated otherwise. The function $g_{\text{n}}(.,.)$ is the geodesic distance between two samples, normalized by the local maximum geodesic distance, that is, the distance between the two farthest points in S (or T). Note that this definition of isometric distortion requires a consistent joint-sampling between S and T to enable reliable normalization of pairwise geodesics (see also Fig. 4.3).

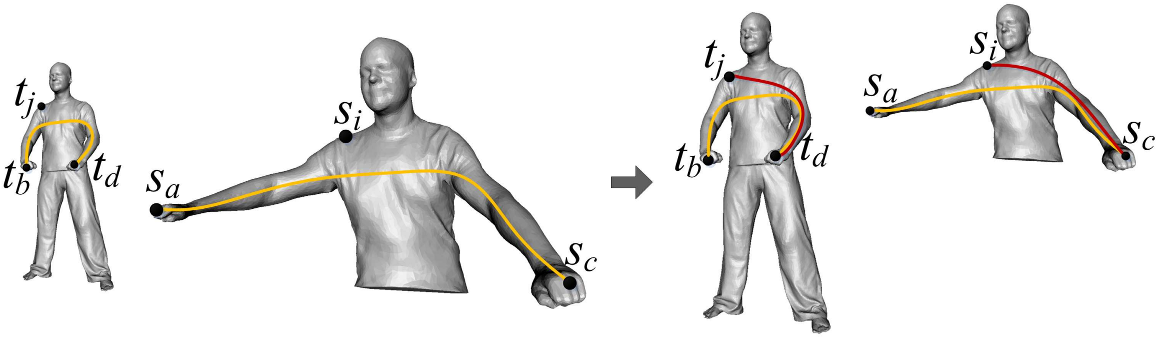

Figure 4.3: Demonstration of the distortion measure with normalized geodesics defined in (4.4). Suppose that sample sets S and T each contains three surface points to be matched. The geodesics used in computation of $d^{(2)}_{\text{iso}}(s_i,t_j)$ (red paths on the right) are normalized w.r.t. the local maximum geodesics (yellow paths on the left).

Unlike $D^{(1)}_{\text{iso}}(\S)$ that averages over all $\binom{|\S|-1}{2}$ pairs of available correspondences in $\S$ for each individual isometric distortion computation of a constituent match, $D^{(2)}_{\text{iso}}(\S)$ relies only on two farthest points to obtain locally normalized geodesics. Although this suggests a saving in the order of $|\S|$ for the time complexity of isometric distortion computation, the use of the latter measure may easily induce a failure: If the two farthest point pair in one sample set does not align well with the pair in the other, the normalization of geodesics becomes inconsistent, and the distortion measure cannot be computed reliably. Although this is less of a problem in the case of complete shape matching (when compared to partial matching), complete matching of semantically similar shapes may still be problematic since farthest point pairs defining maximum geodesics are not necessarily consistent for nearly isometric shapes, such as for Cat vs. Wolf with long and short tails, and for Gorilla vs. Human with long and short arms, being two specific examples from our experiments.

4.1.3 Feature point selection

We use shape extremities as feature points since they are the most salient points of a surface, which are easy to identify while searching for a sparse set of correspondences. To find shape extremities on a given shape, we first compute the integral geodesic distance function [48] for every vertex of the mesh representation and mark the one yielding the maximum function value which is expected to be at the most prominent tip of the shape. The farthest point sampling procedure of [63] started from this marked vertex then selects a number of samples on the surface. We set the number of samples large enough to provide sufficient coverage on the surface, and small enough for computational efficiency of the combinatorial framework that will be described in the next section. The feature vertices resulting from this sparse sampling process constitute the sets S (source) and T (target) to be matched (see Fig. 4.4).

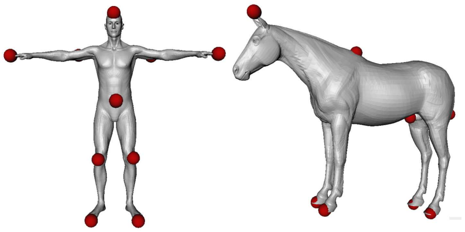

Figure 4.4: Feature vertices on two models (10 samples).

4.1.4 Combinatorial matching

Once we have the feature sets, S and T, extracted from the source and target shapes, we search for an optimal partial mapping $\S^*$ from S to T with minimum distortion. In our setting, one of the shapes, say S, is an isometric part of T up to a scale factor. However, since two shapes are never perfectly isometric, even partly, due to imperfections of the modeling process and geometry discretization errors, it is not usually possible to find a zero distortion mapping, hence the goal rather becomes minimization of the isometric distortion function, that is either $D^{(1)}_{\text{iso}}(\S)$ defined in (4.1) or $D^{(2)}_{\text{iso}}(\S)$ in (4.4). We also note that when matching a shape part with a complete model, the geodesics may slightly differ between similar surface points from one shape to the other due to the cut regions.

We minimize the isometric distortion function via combinatorial search over all possible mappings $\S$ between S and T. In order to make this combinatorial search problem tractable and to reduce the search space, we consider $M = 5$ evenly-spaced vertices from S. Since, in our setting, the shape S is assumed to be a part of T, these extreme vertices are expected to match (at least roughly) with a subset of T. Hence we need to compute the isometric distortion for $M!\binom{|T|}{M}$ different possible (one-to-one) mappings, and the mapping that yields the minimum distortion is selected as the optimal mapping $\S^*$ (see Fig. 4.5).

4.1.5 Extension to dense correspondence

We now extend the sparse correspondence $\S^* : S \to T$ between matched shape extremities to a denser map. This support for dense correspondence also allows us to alleviate the symmetric flip problem that isometric coarse correspondence methods usually suffer from (e.g., left arm/leg is matched to right arm/leg between two human shapes due to sparse sampling).

The first step towards dense matching is to bring the shapes to the same scale using the trusted $\S^*$ that provides us with the factor κ to scale the target mesh:

$$\kappa = \frac{1}{\binom{|\S^*|}{2}} \sum_{((s_a,t_b),(s_c,t_d))\in\mathcal{C}(\S^*)} \frac{g(s_a,s_c)}{g(t_b,t_d)} \tag{4.6}$$

With both shapes at the same scale, in the second step, we take evenly-spaced dense samples on each with consistent spacing. The dense sampling algorithm that we use is very similar to the one described in [22]. We set the sampling radius to $r = 0.17\sqrt{A/\pi}$, where A

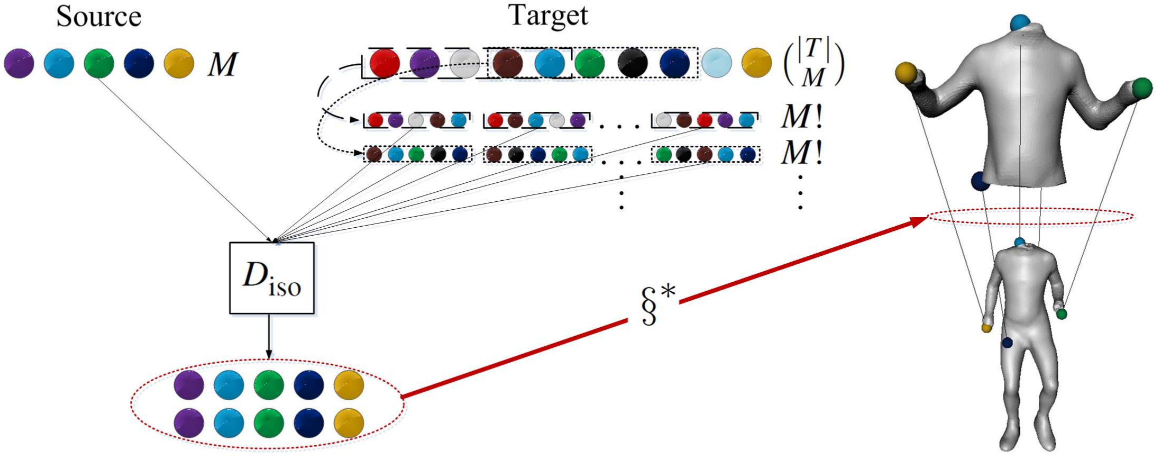

Figure 4.5: Overview of the combinatorial matching process. M extreme vertices of the source shape are matched with $|T|$ extremities from the target. Hence $M!$ possible permutations of $\binom{|T|}{M}$ different combinations are tested with the source, and the one with minimum distortion gives the optimal mapping $\S^*$ (right).

is the surface area of the target shape, that ensures sampling of about 100 dense samples. An arbitrary vertex is selected as the initial sample, and all the vertices lying within its patch of radius r are marked. The next sample is then selected arbitrarily from the set of unmarked vertices. When this is repeated until no unmarked vertex is left, we obtain a partitioning of the shape surface into samples that are at least r apart from each other. We denote the dense sample sets on source and target shapes by $\hat{S}$ and $\hat{T}$, respectively, to which we also append S and T themselves since they cover the salient extreme points. Note that the number of samples on the source shape can be significantly less than the number of samples on the target since the former is assumed to be an isometric part of the latter in our setting.

The third and last step begins with filling in a cost matrix $\mathbf{C}$ where each entry c_{ij} represents the cost of matching a sample $\hat{s}_i$ in $\hat{S}$ to a sample $\hat{t}_j$ in $\hat{T}$. We build $\mathbf{C}$ by setting $c_{i,j} = d^{(2)}_{\text{iso}}(\hat{s}_i, \hat{t}_j)$ via (4.5), based on the trusted correspondences in $\S^*$, hence $\S' = \S^*$. We then perform minimum-weight perfect matching [96] on $\mathbf{C}$, that reveals the desired dense map $\hat{\S}^*$. Since the cardinalities of the disjoint sets must match for a perfect matching, if

$|\hat{S}| \neq |\hat{T}|$, we introduce virtual vertices with connector edges of ∞ weights.

To address the symmetric flip problem, we iterate the third step above K times, each time setting $\S'$ to one of the best K sparse correspondences. Note that our combinatorial matching framework allows us to sort out the K least distorted sparse maps, $\S_1, \S_2, .., \S_K$, where $\S_1 = \S^*$. Hence we compute K dense maps $\hat{\S}_1, \hat{\S}_2, .., \hat{\S}_K$. We then promote the sparse map among the best K, which generates the dense correspondence with minimum distortion, as our final sparse correspondence output:

$$\S^* \leftarrow \arg \min_{\S_1, \S_2, .., \S_K} \hat{D}_{\text{iso}}(\hat{\S}) \tag{4.7}$$

where $\hat{D}_{\text{iso}}$ is similar to $D^{(2)}_{\text{iso}}$ except that it uses geodesics normalized by the same overall maximum geodesic over the shapes, which is the maximum geodesic of the target. In this way, the coarse correspondences with similar distortion values, which are possibly symmetrically flipped, are compared at a finer resolution where their isometric distortions can be more accurately computed. We use $K = 5$ in all our experiments, which is sufficient to differentiate the true mapping most of the time.

4.1.6 Computational complexity

We analyze the computational complexity of $D^{(1)}_{\text{iso}}(\S)$ and $D^{(2)}_{\text{iso}}(\S)$ computations, and the combinatorial part matching algorithm separately. Given a mapping $\S$ of size N, it takes $N \cdot \binom{N-1}{2}$ iterations of constant work to compute each $d^{(1)}_{\text{iso}}(s_i, t_j)$, and hence $O(N^3)$ time for $D^{(1)}_{\text{iso}}(\S)$ computation. Similarly, $D^{(2)}_{\text{iso}}(\S)$ is achieved in $O(N^2)$ time as $N \cdot (N-1)$ iterations are required.

The partial matching algorithm, on the other hand, when fed with two original meshes having V_S and V_T vertices, starts with feature point selection of $O(V \log V)$ time, where $V = \max(V_S, V_T)$. The combinatorial search algorithm then demands $\binom{|T|}{M} M!$ evaluations of the function $D^{(1)}_{\text{iso}}(\S)$ or $D^{(2)}_{\text{iso}}(\S)$, where M is the size of the sought mapping, i.e., the size of the reduced source feature set. Hence the overall algorithmic complexity is $O(\binom{|T|}{M} M! M^3)$ or $O(\binom{|T|}{M} M! M^2)$, which gives reasonable computation time in practice thanks to the feature sampling that leads to small $|T|$ values as well as the choice $M = 5$.

Establishing a dense map of size U adds another $O(V \log V)$ for dense sampling plus $O(U^2 \log U)$ time for minimum-weight perfect matching, both of which are dominated by the preceding combinatorial part matching algorithm (we set $U = 100$ in our experiments).

4.1.7 Experimental results

We have tested the performance of our shape correspondence algorithm on several shape datasets. The first three datasets are uniformly sampled fixed-connectivity mesh sequences, each representing the motion of an articulated object, which we refer to as *Horse*, *Dog*, *Wolf*, *Cat* (from TOSCA shape benchmark [18]), *Jumping Man* [97], and *Dancing Man* [98]. The fourth dataset is a reconstructed pose sequence of a human actor from the SCAPE benchmark [106], which contains 71 different non-uniformly sampled fixed-connectivity models. We have created partial models from some of these datasets by manually cropping and arbitrarily rescaling the original complete models. Representative partial models can be observed throughout the figures.

The last dataset is from the SHREC'11 shape benchmark [105]. The part that we use from this dataset includes a null (reference) shape (*Human*) plus three classes named as isometry, partial, and scaling, each containing 5 uniform high-resolution human models with arbitrary connectivity. The isometry class contains isometric deformations of the null shape and the scaling class includes models from the isometry class in different scales, whereas the partial class is composed by parts of the complete models from the isometry class. We have also used the female class from TOSCA, which contains 10 models, to match with the partial class of SHREC'11 to expand our experiments on nearly isometric shape pairs. Besides, we match *Gorilla* from TOSCA with *Human*, which verifies, along with *Cat* vs. *Wolf*, the accuracy of our algorithm on semantically similar hybrid shape pairs whose maximum geodesic distances do not coincide, a case that would fail most of the conventional isometric shape matchers.

We evaluate the quality of the obtained sparse mappings by using the distortion measure $D_{\text{iso}}^{(1)}$ or $D_{\text{iso}}^{(2)}$, depending on which measure is employed to find the given mapping. We use the distortion measure $\hat{D}_{\text{iso}}$ to evaluate the dense mappings. We also use the ground-truth distortion D_{grd} (Eq. 3.12) to measure deviation from ground-truth correspondences whenever ground-truth correspondence of the source sample on target is either known a priori or computed automatically by aligning the cropped model with its complete version. The maximum geodesic distance on the target model is normalized to 1 in order to simplify the interpretation of this measure.

We have compared our algorithm with the Möbius Voting (MV) method of [31]. For comparison tests, we have run the publicly available code of MV with its default settings of 100 samples and 1M votes, and evaluated its performance based on the $M = 5$ samples that are closest to the shape extremity samples used for our algorithm. We also compare the full dense correspondences obtained by MV to our dense correspondences. In the visualization of our results, we highlight the generating/trusted coarse correspondences with larger spheres and bold lines, and the generated dense maps with smaller spheres. Yellow lines represent the worst matches w.r.t. ground-truth distortion (or isometric distortion in case the former is not available).

Figures 4.6-4.9 display various examples from our partial correspondence results obtained by using the isometric distortion measure $D_{\text{iso}}^{(1)}$, whereas in Fig. 4.22, we visually compare the performance of our algorithm with the MV method. Since Möbius Voting algorithm is restricted to sphere topology, while our method is not, for the sake of comparison, we have patched up all the partial models at their cut regions except for standard SHREC'11 shapes. Whether cut regions are patched up or not, the geodesic information on a partial surface slightly changes with respect to its complete version, introducing some extra imperfection to the correspondence problem; yet we observe that the correspondences that we obtain using our method are very satisfactory, and generally more accurate especially at shape extremities when compared to MV (see Fig. 4.22). The MV algorithm can actually generate accurate correspondences only for a small number of correspondence pairs, those with high confidence values. However the vertices corresponding to these pairs are not generally well distributed on the surface, being arbitrarily located, as can also be observed from Fig. 4.22. We also observe that the matching results of MV at shape extremities may be very inaccurate.

We provide quantitative performance evaluation of our method for dense and extremity-based sparse matching in comparison to MV in Table 4.1, where we also compare the two isometric distortion functions that we have employed for scale normalization. We note that in addition to partial matching (first 4 rows), both methods naturally support complete matching (last 4 rows). Note also that the computation of dense distortion $\hat{D}_{\text{iso}}$, as used in Eq. 4.7, is based on trusted coarse correspondences obtained either by using the distortion measure $D_{\text{iso}}^{(1)}$ (column named as "with $D_{\text{iso}}^{(1)}$") or $D_{\text{iso}}^{(2)}$ (column named as "with $D_{\text{iso}}^{(2)}$").

For part matching experiments (Figures 4.6-4.20), we have respectively 6, 7, 7, and 5 partial models for *Jumping Man*, *Dancing Man*, *Horse*, and *SHREC11* sequences. In the test suite for each dataset, each of the C complete models is matched with P random partial models from the same class, where the pair (C, P) is $(18, 2)$, $(9, 3)$, $(7, 3)$, and $(11, 5)$, respectively. For *SHREC11* evaluation, in addition to matching the 11 models from the null shape, isometry, and scaling classes to all 5 models from the partial class, we also compute the mapping from each of the 10 female models from TOSCA to a random partial class model in *SHREC11*. In all cases, we exclude from evaluation, the shape pairs that result in symmetric flips when matched using our method and/or Möbius Voting. As a result, the performance measures given in Table 4.1 are computed over 19, 22, 15, and 39 pairs, respectively on four datasets. Note that, a denser map that captures more intrinsic geometry of the shapes tends to overcome the symmetric flip problems observed at a coarser resolution, which in turn corrects the symmetric flip between the initial extreme samples as well (Fig. 4.7-bottom).

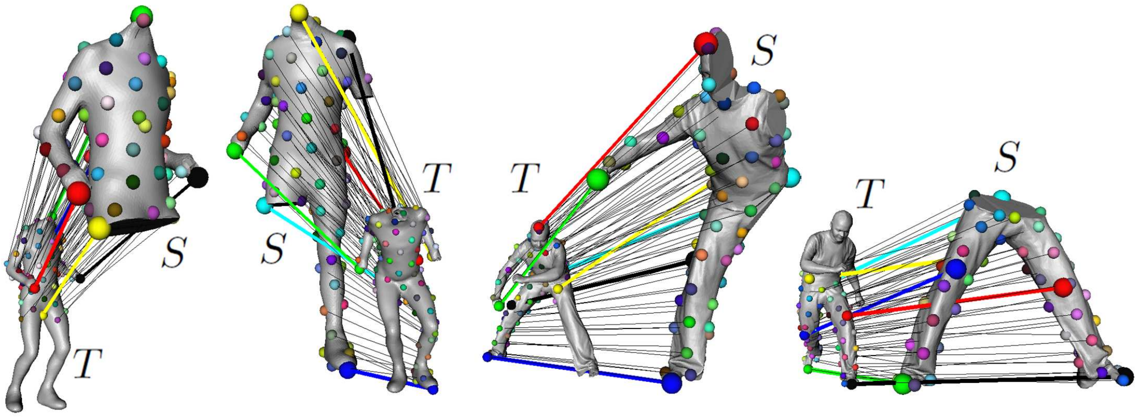

Figure 4.6: Partial correspondences between two different Jumping Man pairs at different scales (left half), and similarly for Dancing Man pairs (right half). S and T denote the partial and complete models, respectively, which applies to all subsequent figures of this section that involve partial models.

As for complete matching experiments (Fig. 4.9), each of the 9 and 10 complete models is matched with a random complete model from the same class for *Dog* and *SCAPE* sequences, respectively. We also perform complete matching across classes to address the incompatible maximum geodesic issue with 5 *Cat-Wolf* and 4 *Gorilla-Human* pairs.

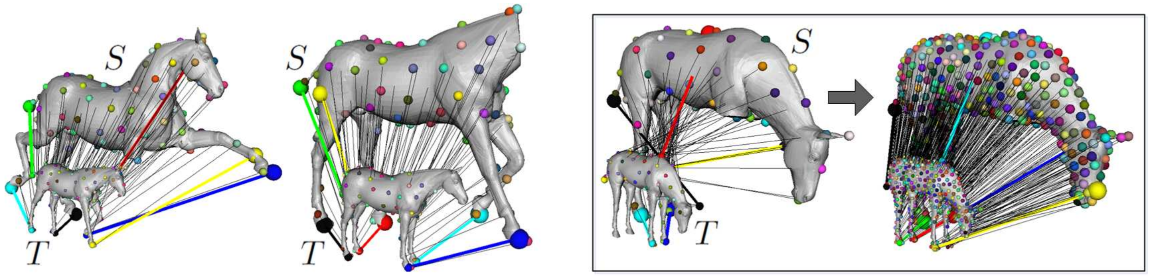

Figure 4.7: (Left) Partial correspondences obtained on two different Horse pairs. (Framed) The sparse partial mapping obtained on a Horse pair initially has a symmetric flip problem (left), which is resolved at a denser resolution (right).

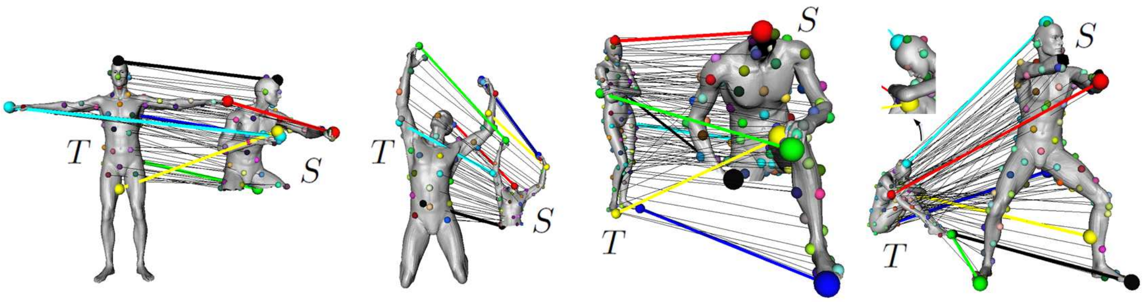

Figure 4.8: Partial correspondence on SHREC11: (Left to right) Mappings to a partial class model S from a null shape, from a scaling class shape, and from two female class shapes in TOSCA.

In Table 4.1, we observe that the distortion measure $D_{\text{iso}}^{(1)}$ outperforms $D_{\text{iso}}^{(2)}$ especially for the part matching problem (first 4 rows). The performance difference here is mostly due to the inconsistency of trusted farthest point pairs used for geodesic normalization with the latter measure, which is not likely to occur in complete matching where $D_{\text{iso}}^{(2)}$ performs as good as $D_{\text{iso}}^{(1)}$ for shapes with perfectly compatible maximum geodesic distances (rows $5-6$) but falls behind for hybrid pairs (rows $7-8$). We also observe that our method outperforms the MV method in sparse extremity matching as well as dense matching. Note that MV cannot be evaluated on *SHREC11* since the models in the partial class contain holes on the surface. The ground-truth correspondence information is not available for hybrid pairs and

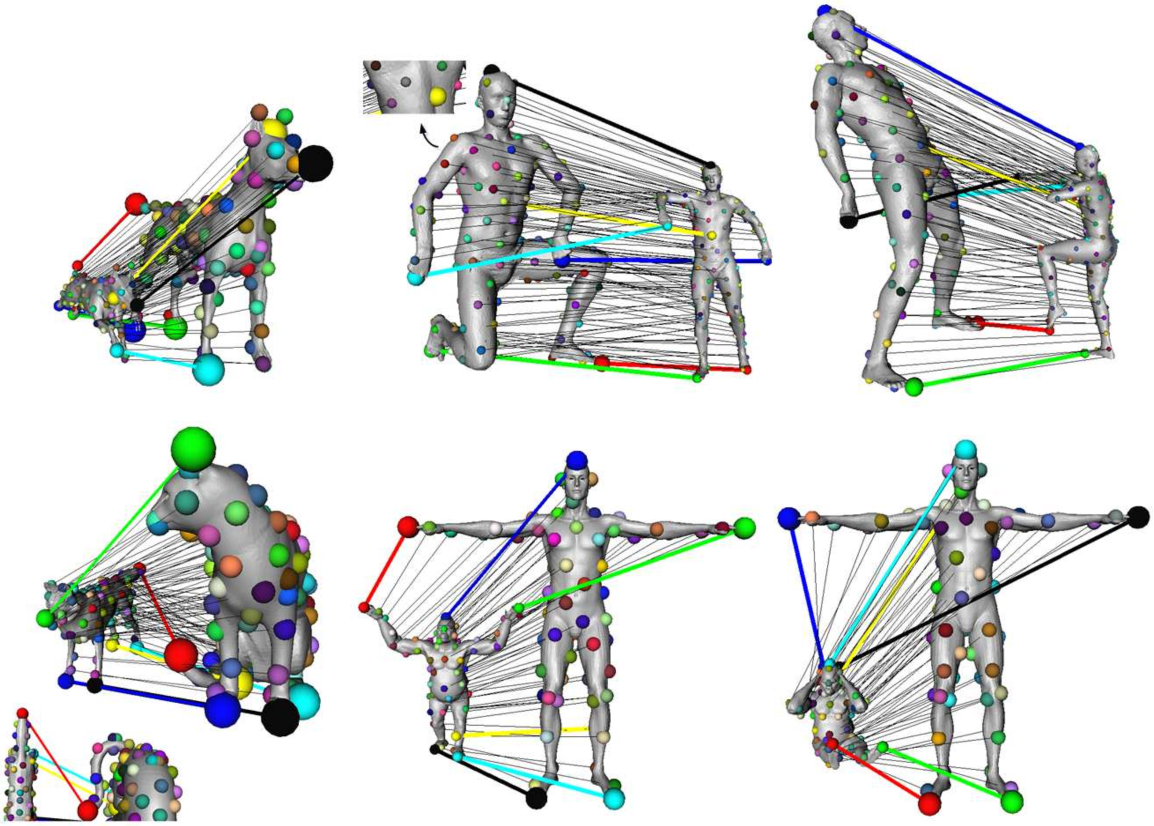

Figure 4.9: Complete correspondences between a Dog pair and two SCAPE pairs (top) as well as between a Cat-Wolf pair and two Gorilla-Human pairs (bottom).

SHREC11.

The execution times of our isometric part matching algorithm on a 2.53GHz PC is, for the highest-resolution dataset *SHREC11* of 50K vertices, 30 seconds for feature selection, followed by 0.6 seconds combinatorial matching of $M = 5$ samples and then 0.4 seconds dense matching of $U = 100$ samples. The respective seconds for the lowest-resolution dataset *SCAPE* of 12.5K vertices are 3.6, 0.3, and 0.3. Note that the cubic complexity of the promoted scale-invariant measure $D^{(1)}_{\text{iso}}$ becomes negligible compared to the quadratic $D^{(2)}_{\text{iso}}$ with $M = 5$.

When we investigate how the choice of M, i.e., the size of the sought coarse mapping, affects the output, we observe that small values, e.g., $M = 3$, may cause unstable results as the individual distortions making up $D^{(1)}_{\text{iso}}(\S)$ become less reliable, whereas with relatively

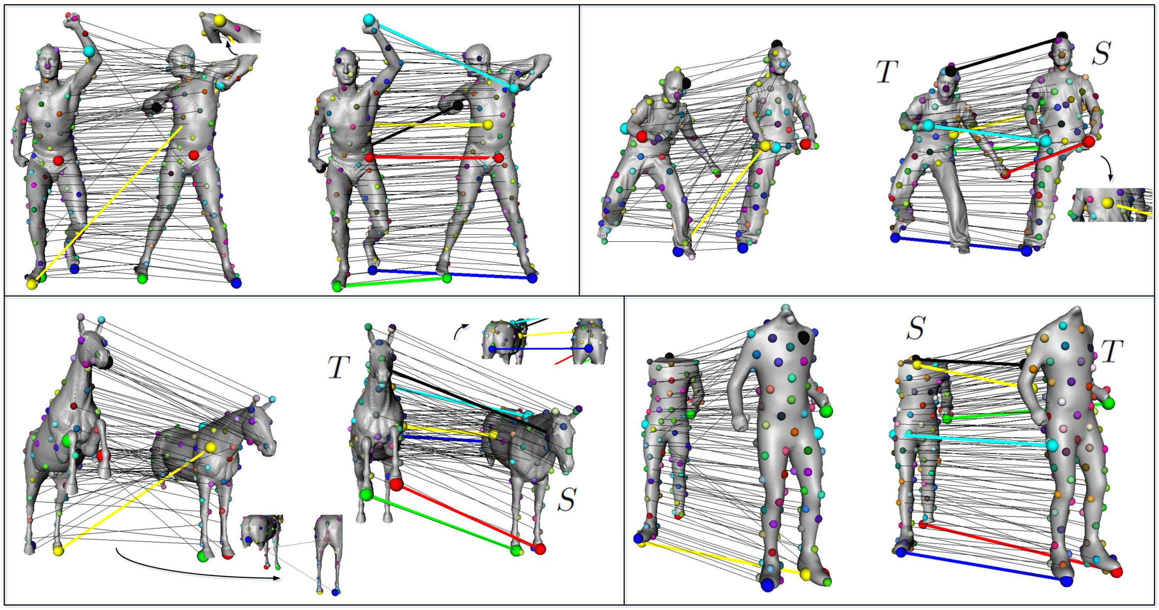

Figure 4.10: Partial correspondences obtained by our method and by MV respectively on the left and right hand sides of each box for four different datasets. MV matches corresponding to our sparse matches are shown with larger spheres, whereas smaller spheres represent the remaining matches.

high values such as $M = 7$, other than computational load concerns, inaccuracies may arise due to joint sampling that starts to produce incompatible point pairs to be matched. In Fig. 4.11, we see that although the initial samples which may contain some inconsistent points are matched as accurately as possible in all cases, the quality of the final dense correspondences may degrade due to these inconsistencies with $D_{\text{grd}} = .201$, $.058$, and $.107$ values on average over SCAPE dataset for $M = 3$, 5, and 7, respectively. We note that the proper choice of M can also be considered as model or database dependent. On the *Dog* dataset for example, which contains shapes with more extremities, the choice of $M = 5$ or $M = 7$ does not affect the performance with $D_{\text{grd}} = .037$ and $.038$ values on average, respectively (for the complete matching scenario).

	Our extremity matching		Our dense matching		MV extremity	MV dense
			with $D_{\text{iso}}^{(2)}$	with $D_{\text{iso}}^{(1)}$		
Pair	$D_{\text{iso}}^{(2)}, D_{\text{grd}}$	$D_{\text{iso}}^{(1)}, D_{\text{grd}}$	$\hat{D}_{\text{iso}}, D_{\text{grd}}$	$\hat{D}_{\text{iso}}, D_{\text{grd}}$	$D_{\text{iso}}^{(1)}, D_{\text{grd}}$	$\hat{D}_{\text{iso}}, D_{\text{grd}}$
Jumping Man (partial)	.030, .083	**.219**, **.072**	.019, .139	**.017**, **.089**	**2.465**, **.253**	**.051**, **.140**
Dancing Man (partial)	.025, .068	**.159**, **.037**	.011, .082	**.011**, **.055**	**1.965**, **.428**	**.068**, **.241**
Horse (partial)	.027, .056	**.114**, **.028**	.012, .055	.**011**, **.048**	**2.463**, **.244**	**.051**, **.121**
SHREC11 (partial)	.044, n/a	**.207**, n/a	.038, n/a	**.034**, n/a	n/a, n/a	n/a, n/a
Dog (complete)	.038, .032	**.191**, **.032**	.013, .037	**.013**, **.037**	**2.530**, **.210**	**.044**, **.101**
SCAPE (complete)	.032, .026	**.143**, **.026**	.010, .058	**.010**, **.058**	**2.744**, **.348**	**.069**, **.139**
Cat-Wolf (complete)	.077, n/a	**.372**, n/a	.039, n/a	**.027**, n/a	**3.367**, n/a	**.101**, n/a
Gorilla-Human (complete)	.079, n/a	**.376**, n/a	.041, n/a	**.021**, n/a	**3.879**, n/a	**.091**, n/a

Table 4.1: Quantitative performance analysis of our method in comparison with Möbius Voting (MV) as bold vs. bold and underlined bold vs. underlined bold.

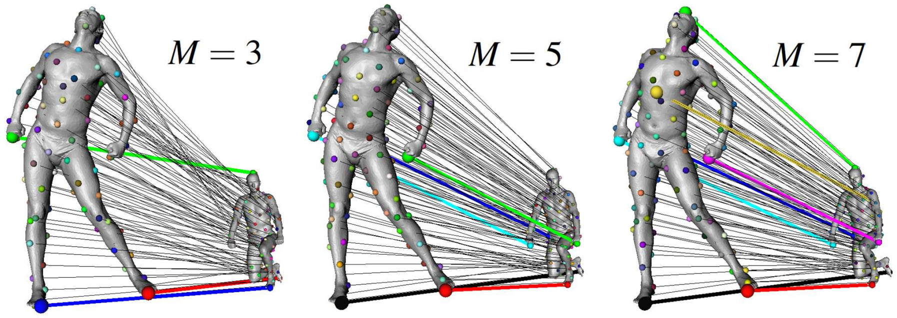

Figure 4.11: Different choices of M in creation of the initial coarse mapping on a SCAPE shape pair.

4.1.8 Conclusion

We have introduced a novel scale normalization method, in comparison with a state of the art method, that proves useful for partial/complete shape correspondence under a particular setting where one of the shapes to be matched is a scaled and isometric part of the other. Our method uses an isometric distortion measure in a combinatorial framework to establish a trusted coarse correspondence based on which the shapes are brought to the same scale, hence handling the scale problem. Dense matching then enables comparison of our approach with state of the art as well as alleviates the symmetric flip problem due to initial coarse sampling. Our isometric matching solution naturally covers the complete matching problem, where the novel distortion measure has proved to be very effective while matching hybrid pairs of semantically similar shapes whose maximum geodesic distances do not necessarily coincide, a case that would fail most conventional isometric shape matchers.

As a partial shape matcher, there is still room for improvement in this framework. We assume that the sample set on the partial model is (approximately) a subset of the complete model sample set which currently has a cardinality of 10. While this setting is sufficient when the partial model is at least the half of the complete model, as in general and in all examples of this work, the assumption breaks when the part is too small, e.g., Hand vs. Human, since in this case the complete model will probably lack most of the samples featured in the partial model. A simple solution is just to populate the target samples to cover all extremities, instead of only 10 evenly-spaced ones, which would work up to 20 samples due to combinatorial complexity.

As a scale normalization tool, we see the work complete after thorough comparisons and experiments. The most straightforward example to an application for which our method can be used in its current form is part retrieval, i.e., searching a 3D database for shapes containing a surface part given as query. We also note that the proposed scale-invariant isometric distortion measure can be embedded into other frameworks that address the partial shape correspondence problem in a more general setting, such as matching shape pairs which are partially isometric but both having parts which are not in common. That would however require the envisagement of a mechanism to eliminate the shape outliers (the uncommon parts) from the global pool of shape vertices, such as the voting approach employed in [31], which will be the topic of our further research.

4.2 Isometric Partial Matching Algorithm

In this section, we present our rank-and-vote-and-combine (RAVAC) algorithm to address the shape correspondence in the most general setting where shapes have multiple common parts at arbitrary scales as well as parts that are not similar [27].

4.2.1 Literature review

The focus of the literature review in Section 4.1.1 is on the scale issue which applies to here as well. Additionally, we review below the literature on partial shape correspondence in the most general setting.

There are few methods in the literature, that are capable to address the partial correspondence problem in the most general setting where shapes may have multiple common parts at arbitrary scales as well as parts that are not similar [31], [28], [29], [93]. All these methods rely on scale-invariant local shape descriptors except for the Möbius Voting method [31]. Note also that the methods in [29], [93] enforce geodesic consistency in addition to local shape similarity, and hence resort to global intrinsic properties for shape normalization. When compared to Möbius Voting (MV), our method has several advantages. First, we handle the scale problem inherent to partial correspondence directly in the 3D Euclidean space wherein isometry is originally defined, hence as free of embedding errors. Second, our method can produce reliable *dense* correspondences between partially isometric shapes. Third, we impose no restriction on shape topology. Last, our method generates more reliable and accurate correspondences, especially at shape extremities, and can handle shape pairs with less similarity overlap.

The Blended Intrinsic Maps (BIM) method [21] extends MV to address the complete shape correspondence problem by blending multiple complete maps generated by shape extremities spread all over the whole surface. Our method is methodologically similar in the sense that we explore multiple partial maps between shape extremities of similar parts. As a shortcoming other than sphere topology restriction, the BIM method does not support partial matching at all.

In this paper, we propose a rank-vote-and-combine (RAVAC) algorithm to find correspondences between partially and/or nearly isometric shapes, which naturally solves the complete correspondence problem as well as partial correspondence. Our algorithm collects partial isometry cues from the given shapes by considering all possible partial mappings (relations) between shape extremities and accumulates the collected information into a vote matrix which is then used to find an overall optimal partial correspondence. We first coarsely sample shape extremities from each shape which may be at arbitrary scale (Section 3.2.3). In the ranking phase, we explore the space of all possible partial maps between these extremities so as to rank all possible matchings (correspondence pairs) with respect to the isometric distortion that they yield (Section 4.2.3). In the voting phase, the qualified top-ranked matchings are subjected to a more detailed analysis at a denser resolution and assigned with confidence values that accumulate into the vote matrix (Section 4.2.4). In the combining phase, by iterating a minimum weight perfect matching algorithm based on the vote matrix, an optimal (partial) mapping between shape extremities is obtained (Section 4.2.5), which can later be extended to a denser map (Section 4.2.6). The computational complexity of the overall shape correspondence algorithm is relatively low, as analyzed in Section 4.2.7. We test the performance of our method on several datasets and benchmarks in comparison to MV and BIM, as presented in Section 4.2.8.

4.2.2 Sampling

We pick shape extremities of the given shapes by using local extrema of the integral geodesic distance function [48]. Let $\mu(v)$ denote the integral geodesic distance at vertex v. Prior to computation of μ, we apply Laplacian smoothing to each shape model to prevent samples at noisy bumps. We then initialize the sample sets with local maxima and minima of μ. The local maxima are expected to be on the tips of a given shape whereas local minima correspond to surface points which lie near the center of the shape [37]. The initial sample sets are then exposed to two steps of pruning, first of which clusters geodesically close samples into the most extreme ones where the closeness threshold is determined based on the maximum geodesic distance $g_{\max}$ on the surface. In our experiments, we have used the value obtained by dividing $g_{\max}$ with $h \in [10, 20]$ depending on the dataset. The second step of pruning removes a local maximum (minimum) v from the sample set if $\mu(v)$ is less (greater) than the average μ to cancel out redundant extremities that are not on tips (central region). The vertices resulting from this sparse sampling process constitute the sets S (source) and T (target) to be matched (Fig. 4.12).

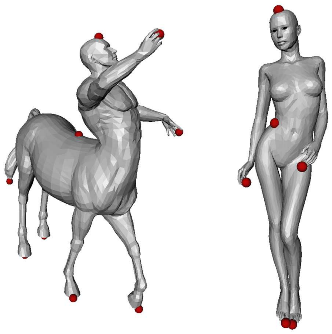

Figure 4.12: Samples (red spheres) produced on the centaur ($|S| = 9$) and girl ($|T| = 6$) models.

4.2.3 Ranking

Given a mapping $\S : S \to T$, we measure the isometric distortion D_{iso} as follows:

$$D_{\text{iso}}(\S) = \frac{1}{|\S|} \sum_{(s_i,t_j)\in\S} d_{\text{iso}}(s_i, t_j, \S') \tag{4.8}$$

where $d_{\text{iso}}(s_i, t_j, \S')$ is the contribution of the individual correspondence (s_i, t_j) to the overall isometric distortion:

$$d_{\text{iso}}(s_i, t_j, \S') = \frac{1}{|\S'|} \sum_{(s_l,t_m)\in\S'} |g(s_i, s_l) - g(t_j, t_m)| \tag{4.9}$$

where $g(.,.)$ is the geodesic distance between two vertices on a given surface and $\S'$ is the correspondence list which is by default $\S - (s_i, t_j)$. Note that Eq. 4.8 is essentially same with Eq. 3.1 except we made some rearrangement to accommodate the upcoming formulations. An optimal solution $\S^*$ can then be searched over all possible mappings so as to minimize the isometric distortion function given in (4.8), which cannot be directly applied in a partial matching setting because there is no agreed maximum geodesic distance to normalize g on source and target into $[0, 1]$ interval due to possible local scale differences. We however observe that, the individual isometric distortion of a queried match (s_i, t_j) can safely be evaluated in the absence of globally normalized geodesics using a correspondence list consisting of matches from the shape part where s_i and t_j itself resides in. We estimate the individual isometric distortion by traversing over all mappings of cardinalities 2 to 5, which does not include (s_i, t_j). We do not check beyond 5 due to efficiency reasons as well as the fact that $1 + 5$ extremities are usually sufficient to represent any given shape part. The estimate, $\tilde{d}_{\text{iso}}(s_i, t_j)$, of the individual isometric distortion of the correspondence (s_i, t_j) is then computed by

$$\tilde{d}_{\text{iso}}(s_i, t_j) = \frac{1}{4} \sum_{k \in [2,5]} \min_l \{ d_{\text{iso}}(s_i, t_j, \S_l^{(k)}) \} \tag{4.10}$$

where $\{\S_l^{(k)} |\ l = 1, 2, ..., L_k\}$ is the set of all maps of size k, not including (s_i, t_j), and $L_k = \binom{|S|-1}{k}\binom{|T|-1}{k}(k!)$. We denote this set by $\mathcal{S}^{(k)}$. Taking the minimum in (4.10) guarantees that if (s_i, t_j) is a good match and traverses a list of matches from the same shape part it resides in, then this is appreciated by selecting the lowest distortion. We then average over sets of maps with different cardinalities since maps of small size, e.g., with $k = 2$ or 3, are likely to fall in the same part as (s_i, t_j) but may exhibit symmetric flip problems, whereas mappings with large cardinalities, e.g., $k = 4$ or 5, are unlikely to be confused by flips but have the risk of including irrelevant samples from a distinct part.

4.2.3.1 Safe map generators

In Eq. 4.10, each (s_i, t_j) traverses all possible one-to-one mappings to compute the minimum distortion over $\mathcal{S}^{(k)}$. To reduce computation, we prune $\mathcal{S}^{(k)}$ so as to keep only the *potentially safe* maps, i.e., the maps between k source samples and k target samples which are expected to be from similar shape parts (see Figure 4.13).

To this end, for each k, we define a set of safe map generators, $\mathcal{G}^{(k)}$, which contains all pairs of k-tuples, one tuple from the source sample set and the other from the target, such that any map between these tuples is potentially safe. We denote each of these pairs of sample tuples by $G^{(k)} \in \mathcal{G}^{(k)}$. A pair of k-tuples is identified as a safe map generator if it satisfies the geodesic consistency condition that the average of pairwise normalized geodesics between source samples is close to that of between target samples. We normalize the geodesics with the maximum geodesic distance between the samples of the given tuple. Note that although all $k!$ mappings generated from a given $G^{(k)}$ are referred to as potentially safe, only a small portion of them are actually correct mappings between two similar parts, and hence while evaluating a query match (s_i, t_j) via Eq. 4.10, taking the minimum helps eliminating the contribution of the irrelevant partial maps.

We create the generator sets $\{\mathcal{G}^{(k)}\}$ incrementally for $k = 3, 4, 5$ (no pruning is necessary for $k = 2$). For $k = 3$, each triplet of source samples is tested with each triplet of target samples to meet the geodesic consistency condition. Among $\binom{|S|}{3}\binom{|T|}{3}$ pairwise triplet combinations, typically $20\% - 30\%$ make into $\mathcal{G}^{(3)}$ in our experiments, where the closeness threshold is set as 0.15. For $k = 4, 5$, we incrementally build $\mathcal{G}^{(k)}$ from $\mathcal{G}^{(k-1)}$. In each case, a pair of source and target samples appended to an existing generator $G_m^{(k)}$ triggers a new geodesic consistency test and typically $2\% - 4\%$ of all possible pairwise combinations are selected. Some safe map generators from $\mathcal{G}^{(3)}$ are demonstrated in Figure 4.14.

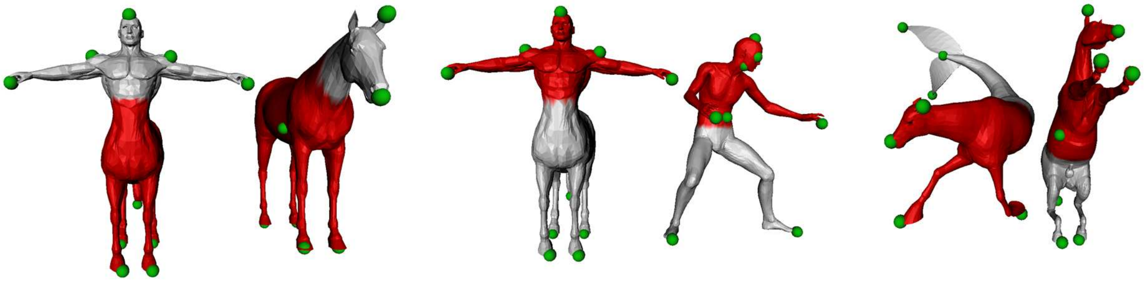

Figure 4.13: Samples (green spheres) on three different shape pairs. A safe map generator $G_m^{(k)}$ is expected to choose k samples from each common shape part (red regions). In spite of the significant structural difference due to large uncommon components (gray regions), our algorithm produces reliable correspondences.

By replacing $\mathcal{S}^{(k)}$ in (4.10) with the potentially safe one-to-one maps based on $\mathcal{G}^{(k)}$, we not only reduce the search space significantly but also increase the accuracy by excluding unexpected distortion values due to evaluation of (s_i, t_j) via (unsafe) maps that accommodates samples from irrelevant shape parts. Once the individual distortions are computed, for each source sample s_i, we rank the pairs (s_i, t_j) based on their individual distortions: We sort all possible $|T|$ different matches with respect to $\tilde{d}_{\text{iso}}(s_i, t_j)$ in ascending order and qualify only the ones with a distortion value that appears before the first significant jump in the corresponding distortion plot (we set the jump value as the sum of the first two distortion differences in the sorted list). With the qualified matches, the voting module is then ready to start, as described next.

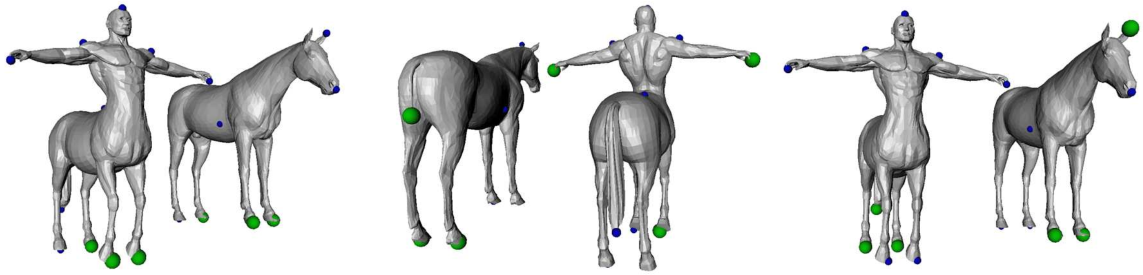

Figure 4.14: Three different triplet pairs (safe map generators) from $\mathcal{G}^{(3)}$ are indicated with large green spheres on three different views of the same shape pair. Blue spheres represent the remaining samples.

4.2.4 Voting

With the ranking of possible matches in hand, one possibility to solve the correspondence problem is to select the least distorted match for each source sample. This straightforward solution would give a (possibly many-to-one) mapping that would however suffer from symmetric flips and mismatches due to low number of extremities being matched. We therefore consult to a voting procedure which is more robust, that relies on the ranking obtained in the previous section.

The voting process considers the generator set $\mathcal{G}^{(3)}$ (the others are discarded simply due to computational reasons). Among 3! potentially safe maps generated from each $G_m^{(3)} \in \mathcal{G}^{(3)}$, only those containing the matches qualified in the ranking phase are taken into account. Each such safe map $\S_l^{(3)}$ defines two regions of interest on the given two shapes (as will be explained next), which are resampled and matched at a denser level (see Figure 4.15). The resulting isometric distortion is then used to vote up the three matches contained in this potentially safe map. This is repeated for all qualified safe maps and the resulting votes are accumulated into a vote matrix where each entry represents the confidence of a potential match between two shape extremities. In the sequel, we describe the voting algorithm in detail.

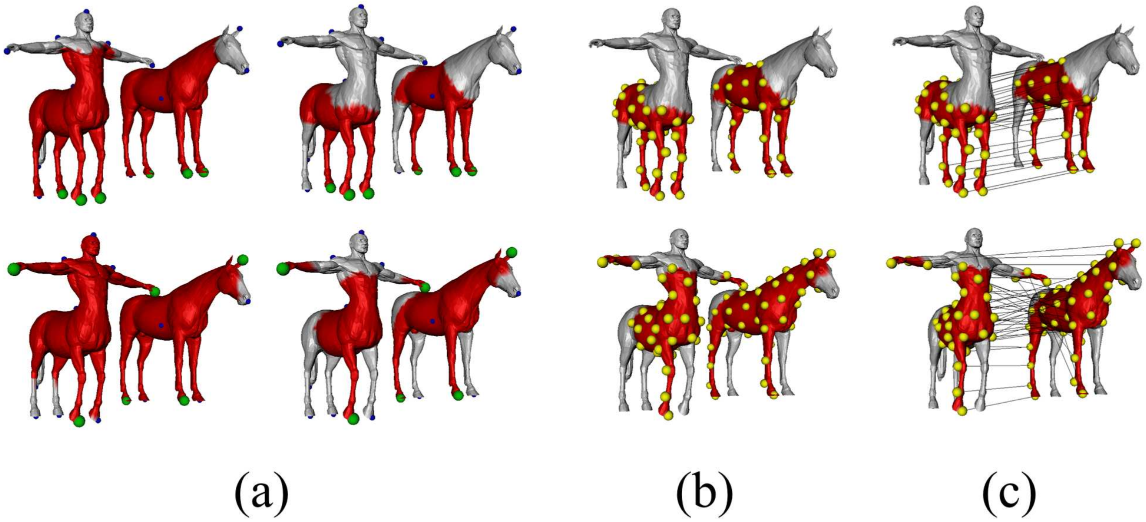

Figure 4.15: Two examples of the voting process in top and bottom rows for two different generating pairs of sample triplets from $\mathcal{G}^{(3)}$. a) Two steps that decide regions of interest (painted red), b) evenly-spaced dense samples (yellow spheres), and c) one-to-one map between them (lines) to be used in computation of confidence votes.

4.2.4.1 Finding regions of interest

Let $\S_l^{(3)}$ be a potentially safe map generated from $G_m^{(3)} = ((s_{i_1}, s_{i_2}, s_{i_3}), (t_{j_1}, t_{j_2}, t_{j_3}))$ such that $\S_l^{(3)} = \{(s_{i_1}, t_{j_1}), (s_{i_2}, t_{j_2}), (s_{i_3}, t_{j_3})\}$. The voting algorithm first brings the shapes to the same scale by scaling the target mesh with a factor $\kappa = (\frac{g(s_{i_1}, s_{i_2})}{g(t_{j_1}, t_{j_2})} + \frac{g(s_{i_1}, s_{i_3})}{g(t_{j_1}, t_{j_3})} + \frac{g(s_{i_2}, s_{i_3})}{g(t_{j_2}, t_{j_3})})/3$ based on the geodesic distance ratios between the ordered sample points, and then finds the regions of interest that these shape extremities determine (see Figure 4.15a). Let the extremity sample sets $\{s_{i_1}, s_{i_2}, s_{i_3}\}$ and $\{t_{j_1}, t_{j_2}, t_{j_3}\}$ be denoted by S_l and T_l, respectively. The region of interest on the source shape includes the source mesh vertices that are close to S_l and distant to $S - S_l$. To implement this, we mark a vertex v as a region vertex if $g(s, v) < g_{\max} \ \ \forall s \in S_l$, where $g_{\max}$ is the maximum geodesic between extremity samples in S_l (see Figure 4.15a-left). To meet the second requirement, for each maximal extremity $s' \in S - S_l$, we unmark the region vertices that are at most $g(s, s')/2$ apart from s' where $s \in S_l$ is the closest extremity to s' (see Figure 4.15a-right). The region of interest on the target shape defined by the extremity set T_l is computed likewise.

4.2.4.2 Dense region sampling

Next, we distribute evenly-spaced dense samples in the regions of interest (see Figure 4.15b). We resample and populate the region of interest on the source shape by first selecting the corresponding extremities as the first three dense samples. Given the region area A, we use the radius $r = 0.17\sqrt{A/\pi}$ to ensure evenly sampling of about 100 dense samples as follows. When an arbitrary region vertex is selected as a dense sample, all the region vertices lying within its patch of radius r are marked. The next dense sample is then selected arbitrarily from the unmarked region vertices. When this is repeated until no unmarked region vertex is left, we obtain a partitioning of the region into dense samples that are at least r apart from each other [48]. A similar evenly-sampling on the regions of interest of the scaled target mesh using the same r makes the joint dense samples as consistent as possible, especially if source and target regions are due to the compatible generating extremity samples from the same shape parts, in which case the distortion of the dense map to be computed becomes low. We denote the dense sample sets on regions due to S_l and T_l by $\hat{S}_l$ and $\hat{T}_l$, respectively.

4.2.4.3 Dense region matching

We match $\hat{S}_l$ and $\hat{T}_l$ by using a fast minimum-weight perfect matching algorithm [96], and denote the resulting dense map by $\hat{\S}_l$. To feed the algorithm, we build a cost matrix $\mathbf{C}$ where each entry c_{pq} is the isometric distortion of matching a source sample $\hat{s}_p \in \hat{S}_l$ to a target sample $\hat{t}_q \in \hat{T}_l$. We compute each c_{pq} based on the three correspondences available in the qualified safe map $\S_l^{(3)}$ by setting $c_{pq} = d_{\text{iso}}(\hat{s}_p, \hat{t}_q, \S_l^{(3)})$ via (4.9), which is expected to map $\hat{S}_l$ to $\hat{T}_l$ with low distortion if $\S_l^{(3)}$ is correct. Since the cardinalities of the disjoint sets must match for a perfect matching, if $|\hat{S}_l| \neq |\hat{T}_l|$, we introduce virtual vertices with connector edges of ∞ weights. We also enforce the three correspondences available in the map $\S_l^{(3)}$ to be preserved in the resulting dense map by setting the corresponding entries of the cost matrix to $-\infty$. Hence we guarantee that $\S_l^{(3)} \subset \hat{\S}_l$.

4.2.4.4 Vote matrix

The dense region matching process described previously is repeated for each qualified safe map $\S_l^{(3)}$ generated from $\mathcal{G}^{(3)}$, and each such matching process produces a confidence vote $\gamma_l(s_i, t_j)$ for each pair $(s_i, t_j) \in \S_l^{(3)}$. This confidence vote is computed based on the individual isometric distortion that the dense matching yields:

$$\gamma_l(s_i, t_j) = \exp(-d_{\text{iso}}(s_i, t_j, \hat{\S}_l)) \tag{4.11}$$

which produces a value in $[0, 1]$. Note that the confidence measure given above considers the global consistency of the matching between dense samples since (s_i, t_j) traverses all $\sim$100 matches in $\hat{\S}_l$ to compute d_{iso} via Eq. 4.9. (see Figure 4.16).

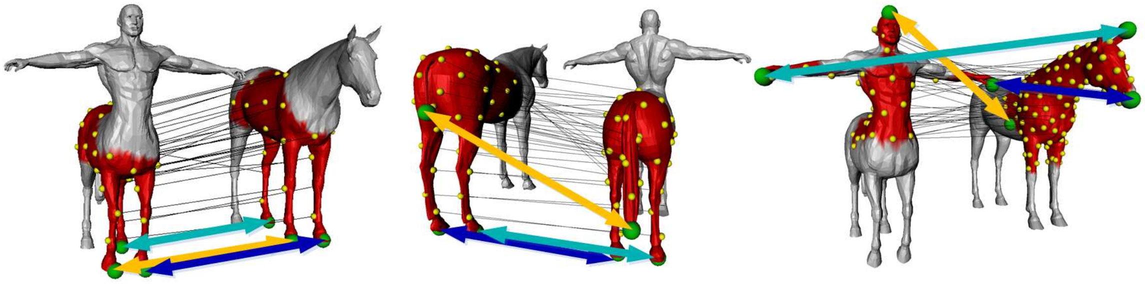

Figure 4.16: Confidence vote assignment to the matches between the extremity samples (colored bold lines) for three different safe maps shown at left, middle, and right. Confidence votes are computed by traversing a pair (s_i, t_j) over the dense matching displayed with thin black lines.

The confidence votes resulting from all dense mappings are then accumulated into a vote matrix $\mathbf{\Gamma}$, where each entry γ_{ij} eventually represents the confidence of matching a source extremity $s_i \in S$ to a target extremity $t_j \in T$. More specifically, each entry γ_{ij} is given by the average of all confidence votes that the pair (s_i, t_j) gets. We note that, to improve robustness, we discard a qualified safe map from the voting process if the target region of interest is twice larger than the source region after scale normalization since this definitely implies a bad configuration, e.g., three source samples from finger tips of a hand vs. a target triplet consisting of two hands and a head on a pair of human shapes.

4.2.5 Combining

We use the vote matrix $\mathbf{\Gamma}$ to find an optimal mapping, $\S^* : S \to T$, from the set of source extremities to the set of target extremities. We first convert the vote matrix into a cost matrix and then apply the minimum-weight perfect matching algorithm in [96], that gives us an optimal one-to-one mapping which respects confidence values globally. The cost matrix $\mathbf{C}^*$ is formed by replacing the high confidence entries in $\mathbf{\Gamma}$ with $c^*_{ij} = 1 - \gamma_{ij}$ and others with ∞. High-confidence entries are determined automatically using a procedure that is similar to the jump detection algorithm described in Section 4.2.3.1. Given a sample s_i, we sort all confidences in row γ_i to infer the average difference ζ between the consecutive sorted confidences. We then mark the entries appearing before the first significant jump, which we set to 1.5ζ, as high-confidence entries. Taking into account only high confidence entries improves the robustness of the matching algorithm.

The above perfect matching algorithm produces a one-to-one mapping that associates every source extremity with one target extremity sample. This is a desirable solution in the case of complete shape correspondence as well as for the problem of part matching. However, when the shapes are partially isometric both with parts that are not similar, some of the matches in the resulting map will clearly be outliers which distract the optimization process itself. Also, when the structural dissimilarity between the shapes is large, there is the danger of occupying a nice spot on the target shape with an irrelevant match which originates from a source sample whose counterpart does not actually exist on the target. To address this problem, we iterate the perfect matching algorithm each time removing one of the outliers. Since an outlier match is expected to have small confidence, at each iteration we remove the least-confident match by setting the corresponding entry in $\mathbf{C}^*$ to ∞ and solving the new $\mathbf{C}^*$ again and repeating these removals until convergence, i.e., when there is no jump in the confidences of the matches in the resulting one-to-one map, i.e., all consecutive confidence differences are less than 3ζ. Hence the final map that our algorithm produces is always one-to-one, but does not necessarily associate every extremity sample on the source (or target) shape with an extremity on the other; in other words, it is a one-to-one partial mapping. We summarize our overall shape correspondence algorithm (RAVAC) with the pseudocode given below.

4.2.6 Extension to dense map

The optimal coarse correspondence $\S^*$ that our RAVAC algorithm produces between sparse shape extremities can be extended to a dense map. For each mapping with cardinality three, which is a subset of $\S^*$, we densely resample and match the corresponding regions of

```
Input: Extremity sample sets $S$ and $T$
Output: One-to-one mapping $\S^* : S \rightarrow T$
———— Ranking ————
For each $s_i \in S$
  Estimate $\tilde{d}_{\text{iso}}(s_i, t_j) \;\; \forall t_j \in T$ via (4.10);
  Qualify the match $(s_i, t_k)$ for voting if $d_{\text{iso}}(s_i, t_k)$ appears
  before the first significant jump in the sorted distortion plot of $s_i$;
———— Voting ————
$\mathbf{\Gamma}$: Vote matrix with all entries $\gamma_{ij}$ initialized to 0;
$\mathcal{G}^{(3)}$: Set of all safe map generators;
For $m = 1$ to $|\mathcal{G}^{(3)}|$
  If $G^{(3)}_m = ((s_{i_1}, s_{i_2}, s_{i_3}), (t_{j_1}, t_{j_2}, t_{j_3}))$ generates
  $\S^{(3)}_l = \{(s_{i_1}, t_{j_1}), (s_{i_2}, t_{j_2}), (s_{i_3}, t_{j_3})\}$ where all pairs are qualified
    Bring meshes to the same scale by multiplying target with
    $\kappa = (\frac{g(s_{i_1}, s_{i_2})}{g(t_{j_1}, t_{j_2})} + \frac{g(s_{i_1}, s_{i_3})}{g(t_{j_1}, t_{j_3})} + \frac{g(s_{i_2}, s_{i_3})}{g(t_{j_2}, t_{j_3})})/3$;
    Set $S_l = \{s_{i_1}, s_{i_2}, s_{i_3}\}$ and $T_l = \{t_{j_1}, t_{j_2}, t_{j_3}\}$;
    Compute the regions of interest on source and target defined by $S_l$ and $T_l$;
    Spread ~100 dense samples $\hat{S}_l$ and $\hat{T}_l$ on regions of interest;
    Find the dense map $\hat{\S}_l : \hat{S}_l \rightarrow \hat{T}_l$;
    Vote up confidence of extremity match $(s_i, t_j) \in \S^{(3)}_l$ via
    $\gamma_{ij} = \gamma_{ij} + \exp(-d_{\text{iso}}(s_i, t_j, \hat{\S}_l))$;
———— Combining ————
$\mathbf{C}^* = \infty$;
$c^*_{ij} = 1 - \gamma_{ij}$ for high-confidence matches $(s_i, t_j)$;
Repeat
  $\S^*$ = minimum-weight perfect matching on $\mathbf{C}^*$
  Let $(s_a, t_b)$ be the least-confident match in $\S^*$;
  $c^*_{ab} = \infty$;
Until there is no jump in confidences of the matches in $\S^*$
```

Figure 4.17: The general shape correspondence algorithm: RAVAC.

interest. This process is repeated for all $\S_l^{(3)} \subset \S^*$, and then the resulting dense matchings are blended into one dense map, that we denote by $\hat{\S}^*$.

The process of resampling and matching the regions is the same as described in Section 4.2.4 except that this time the resampling algorithm takes into account the other overlapping regions of interest while populating its samples. The regions of interest are enforced to include the same samples in the parts where they overlap. Hence while resampling a region (Section 4.2.4.2), the dense sample set is initialized to include all the dense samples that have been so far included by some other regions of interest. This enables to accumulate a set of candidate matches on the target, $F(\hat{s}_i)$, for a given dense source sample $\hat{s}_i$ as regions of interest are matched. Let $\hat{\mathbf{t}}_j$ be the coordinate vector for the target dense sample $\hat{t}_j$. The blended coordinate

$$\mathbf{b}_i = \frac{1}{|F(\hat{s}_i)|} \sum_{\hat{t}_j \in F(\hat{s}_i)} \hat{\mathbf{t}}_j \tag{4.12}$$

then approximates the geodesic centroid of the candidate matches for $\hat{s}_i$ and provides $(\hat{s}_i, \hat{t}_k)$ as the blended dense match, where $\hat{t}_k$ is the target vertex closest to $\mathbf{b}_i$ in L_2 sense.

We note that the main computational load of this extension is on the minimum-weight perfect matching phase which is negligible when the number of dense samples to be matched is less than 500.

4.2.7 Computational complexity

Sampling N initial extremities on the input mesh with V vertices takes $O(V \log V)$ time. The ranking module demands $O(N^4)$ operations as each of N samples is tested with all $O(N^3)$ triplets to traverse a map of constant size. The voting procedure, for each qualified triplet ($O(N^3)$), generates a potentially safe map, defines regions of interest around the map ($O(V)$) and distributes $\sim$100 dense samples on them ($O(V \log V)$), which are then matched for confidence computations. The voting complexity is hence $O(N^3 V \log V)$ where the number of region vertices taken as V is generally much fewer and there is always the possibility to hash the existing regions to reuse them in constant time, which yield a much more moderate cost in practice. The final combining step performs minimum-weight perfect matching of $O(N^2 \log N)$ work about at most 10 times until convergence. The dense map extension comes without any additional complexity as the blended coordinates computed in $O(N^3 V \log V)$ time provide the closest mesh vertices in $O(NV)$ time. The overall worst case complexity is therefore $O(N^3 V \log V)$ assuming $N \ll V$.

Compared to $O(V^2 \log V + N^4 \log N)$ complexity of the MV method, our method is considered to be fast since it uses a much smaller N, e.g., 10 vs. 250. We note that the number of sample triplet pairs in the above discussion does not exceed 15K for a typical shape pair with $N = 10$ extremity samples after pruning, e.g., 25% of all $\binom{10}{3}\binom{10}{3}$ triplets. The BIM method, on the other hand, is faster with $O(V \log V)$ complexity yet not guaranteed to solve the partial matching problem.

4.2.8 Experimental results

We test the performance of our method on several shape benchmarks for complete, partial and dense correspondence problems in the presence of isometric (or nearly isometric) deformations and scale differences.

The first dataset is a subset of the Non-rigid World benchmark [38], which consists of uniformly-sampled meshes representing articulated motions of 17 horses, 6 centaurs, 6 seahorse, 21 gorillas, 4 males, and 4 females, each with ∼3.4K vertices and arbitrary connectivity. We have also created 4 partial horse models by manually cropping the original complete models. Matching horses within themselves (Horse↔Horse) is a complete matching problem on shape pairs with no structural differences, whereas establishing a map from a horse to its cropped version (Horse↔Horse-part) is basically a part matching problem where one shape is an isometric part of the other. A harder, and perhaps more realistic, partial correspondence setting involves pairs with uncommon parts such as Centaur↔Horse, Seahorse↔Horse, and Centaur↔Human pairs.

The second dataset is a part of the SHREC'11 benchmark [105]. A high-resolution mesh of a null reference male model (SHREC-null), its 5 different poses that have undergone isometric deformations (SHREC-iso), one isometric pose in 5 different scales (SHREC-sca), and 5 cropped models (SHREC-part) are represented with ∼50K uniformly-spaced and arbitrarily connected vertices.

The third and the last dataset that we employ is the SCAPE benchmark [106], which is reconstructed from a real scene, representing the real motion of a human actor in 71 meshes. Since this dataset comes with ground-truth correspondence information for all

∼12.5K vertices of the meshes, we mainly use it to evaluate the performance of our dense matching results.

Figures 4.18-4.21 display various examples from our shape correspondence results, whereas in Figures 4.22 and 4.23 we compare our algorithm with the Möbius voting (MV) method of [31], and in Figure 4.24 with the Blended Intrinsic Maps (BIM) method of [21]. In all figures concerning maps produced by our algorithm, we give the most confident 6 extremity matches in red, green, blue, black, cyan, and magenta colors, respectively, and for the subsequent matches, if exist, we use dashed black lines with spherical endpoints scaled with a radius proportional to the confidence of the correspondence pair. Unmatched samples, if exist, are represented by small red spheres. Apart from these visual evaluations, we also evaluate the performance using the distortion measure D_{grd} (Eq. 3.12) which quantifies the deviation of a given correspondence $\S$ from ground-truth correspondences as the ground-truth correspondence of a given source sample on the target is either marked manually or available a priori as in SCAPE dataset, and $g(.,.)$ is the geodesic distance function. The maximum geodesic distance on the target model is normalized to 1 to simplify the interpretation of this measure. For comparison tests, we have run the publicly available code of MV with the default settings of 100 samples and 1M votes and evaluate their correspondence results based on the samples that are closest to the extremity samples used by our algorithm. We also compare the correspondence formed by their top (most confident) 5 matches with our 5 corresponding matches. For the former case concerning extremity matches, our method outperforms MV whereas for the latter 5 matches that tend to be on non-extremities such as shape centers, we are almost on a par with them (Table 4.2). Note that, our original coarse correspondence on the extremity samples needs to be extended to a denser one as described in Section 4.2.6 in order to find our closest counterparts to the top 5 MV matches.

We have also run the public code of BIM to compare their extremity and full dense matching performances with ours on 8 Horse (from TOSCA shape benchmark [18]) and 12 SCAPE pairs for both of which the full ground-truth information is available. For the former, we follow the same strategy in MV extremity comparisons whereas for the latter we interpolate our dense map (Section 4.2.6) to a full map between all vertices as described in [21]. In extremity matching, BIM is slightly better than our method mainly because we enforce three extremity correspondences in the generating partial maps to be preserved in the resulting dense maps to be blended. The BIM method, on the other hand, blends unrestricted match candidates for a given extreme sample which renders it more flexible as exemplified via nose matches in Figure 4.24-bottom. Full dense maps of BIM are again slightly better than our interpolated counterparts (Table 4.3) yet suffer from unmatched regions. All the slight differences in favor of BIM during these complete correspondence tests can be tolerated since our main concern is the more challenging partial correspondence problem that cannot be handled by the BIM method.

For the complete correspondence tests between the Non-rigid World horse models, each complete model is mapped to a random complete model whereas the partial correspondence is evaluated by finding maps between each cropped model and 4 random complete models, hence a total of 33 pairs (Figure 4.18). Note that, we do not change any component or parameter to toggle between the complete and partial correspondence problems as the algorithm designed specifically for the partial matching problem solves the simpler complete matching problem naturally.

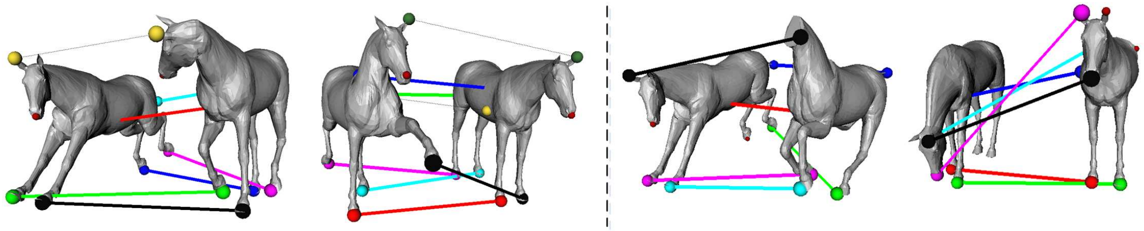

Figure 4.18: Complete shape matching between the extremities of two complete horse models from Non-rigid World (left). Maps between a horse and two cropped models as partial correspondences (right).

The other tests that we have conducted on the Non-rigid World benchmark require partial correspondence in the presence of uncommon parts that complicates the problem further. Our algorithm successfully rules out the samples representing the uncommon parts without causing any confusions on the matches concerning samples of interest from the common parts (Figure 4.19). The outlier matches between heads of Seahorse↔Horse are removed by the iterative perfect matching process in the combining phase as the confidences of these matches are relatively low compared to the matches between the feet, which is as expected since the heads of these two models are isometrically very different. In these experiments, we pick 5 centaurs and match each of them with 4 random models from horse, male, and female classes, hence a total of 60 pairs. Similarly, all seahorses are matched to 4 random horse models for another 24 pairs.

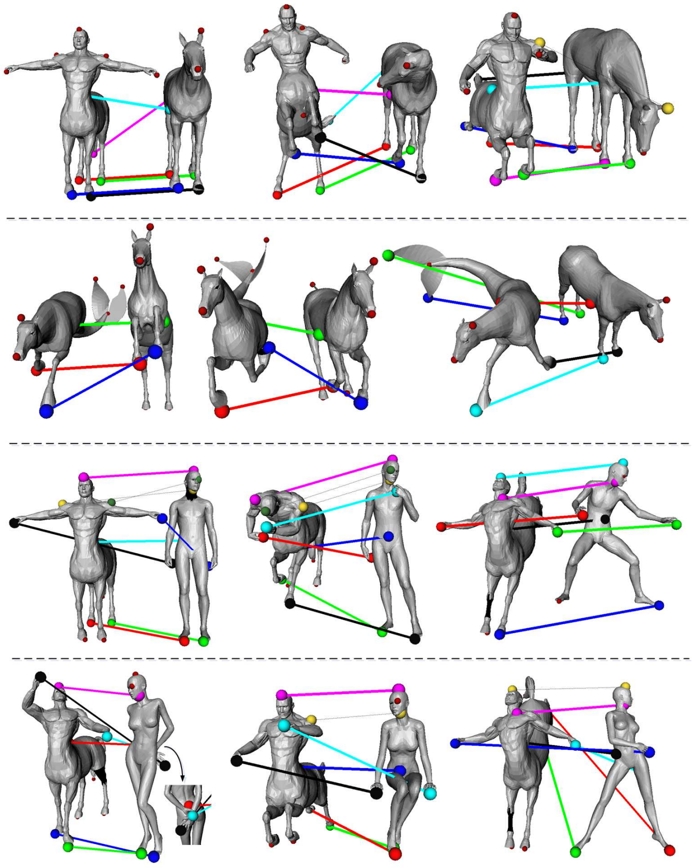

Figure 4.19: Examples of partial shape matching on three different pairs from three different shape classes in Non-rigid World, given in increasing level of difficulty from top to bottom row.

The SHREC'11 tests are performed by mapping each isometry class model to a random model from the same class. As far as the partial correspondence is concerned, each model in the partial class is matched with 3 random models from the isometry and scaling classes, hence a total of 35 pairs (Figure 4.20).

We also experiment on a low-resolution gorilla and a high-resolution male from two dif-

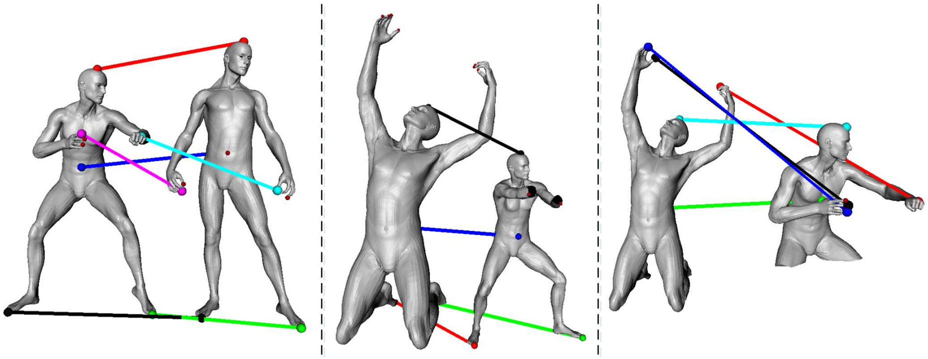

Figure 4.20: Example matchings on SHREC'11 between complete shapes of the isometry class (left). A partial model mapped to a scaling model (middle) and to an isometry model (right) via partial matching. Note the arbitrary number of samples on fingers.

ferent benchmarks to demonstrate the endurance of our algorithm not only to the difference and size of the triangulations but also to complete matching of shapes that exhibit local similarities but large deviations from global isometry (Figure 4.21). With 21 pairs obtained by matching the null shape to all gorillas, we obtain successful results as the semantically common parts accommodate sample triplets to cast votes. Since we isolate each potentially compatible triplet pair from all other samples during the voting process, the male hands can be matched to the elongated gorilla hands successfully.

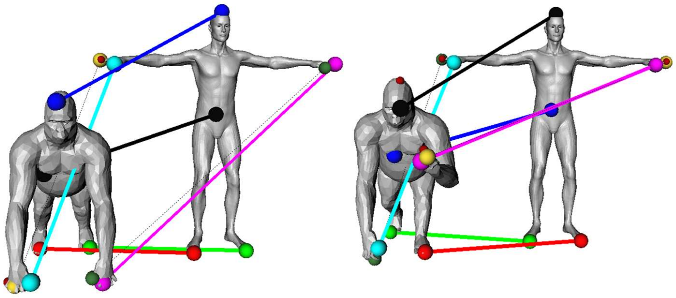

Figure 4.21: Two examples of complete shape matching between male and gorilla meshes that exhibit locally similarities but large deviations from global isometry.

Several examples for comparison with MV are demonstrated in Figure 4.22. The top 5 MV matches are highlighted by large spheres whereas their extremity matches that are closest to ours are indicated by large spheres with connecting lines. All other small spheres of matching colors represent the remaining correspondences. A similar visualization is performed for our results as well except only 5 of our dense matches, that correspond to the top 5 MV matches, are shown. We observe that whenever the dissimilarity between shapes increases, MV shows instabilities especially at the extremity matches as the mutual closest point matches in their embedding domain starts to confuse on these regions of small area. Our results, on the other hand, rely on the dense matchings obtained in the neighborhoods of the extremities, which are hence less likely to get negatively affected by irrelevant data.

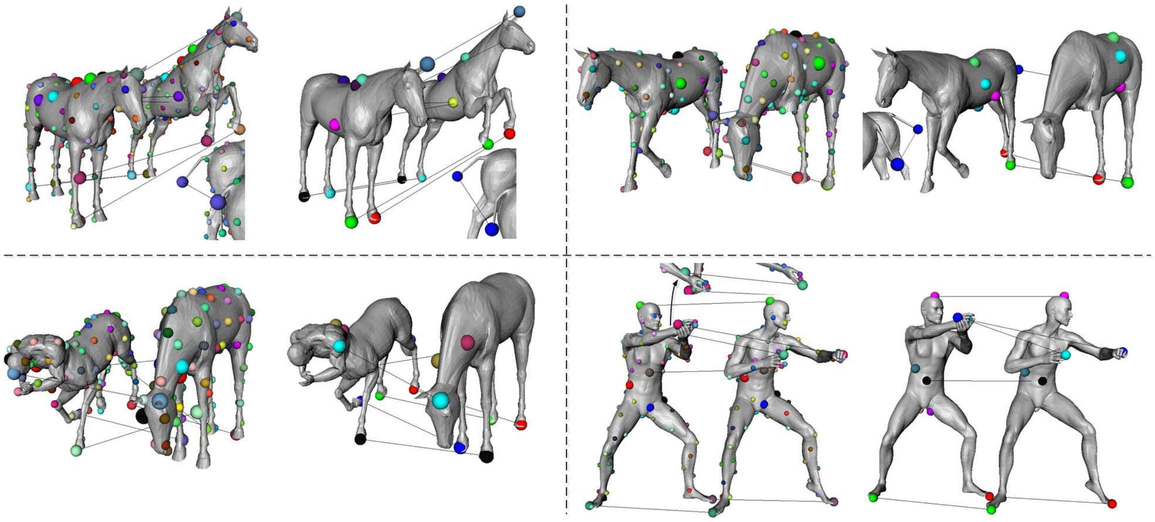

Figure 4.22: Möbius Voting (left) vs. our method (right) on four different shape classes.

We finally evaluate our dense matching described in Section 4.2.6. By establishing a dense map on SCAPE data, we can thoroughly evaluate D_{grd} over all matches, rather than on just 5 matches, thanks to the ground-truth information available. Figure 4.23 visualizes our dense map between ∼250 vertices and the MV map of the same size with thin lines, demonstrating our much smoother correspondence flow as compared to MV. The yellow bold line represents the worst individual match which suffers from outliers around extremity samples for the MV method. The quantitative evaluation favors our method as well with $D_{\text{grd}} = .153$ vs. $.044$ for MV and our method, respectively. Similarly, the worst distortions for the dense correspondences obtained are .952 and .257 in our favor.

In Table 4.2, we observe that the MV results on SCAPE database, which includes much more non-delaunay triangles than SHREC meshes, degrade as compared to its performance on SHREC, although the object types and isometric deformations applied are quite similar for these two datasets. This decrease in performance is not observed in our method which is insensitive to the peculiarities of the particular triangulation. The missing entries in the table for the rows including SHREC-part meshes with holes are due to sphere topology restriction of MV. The entry for Centaur↔Human pair is also missing since the similarity between the shapes is required to be more than 40% in the case of MV [31]. Note also that gorilla and seahorse meshes crash the public MV code.

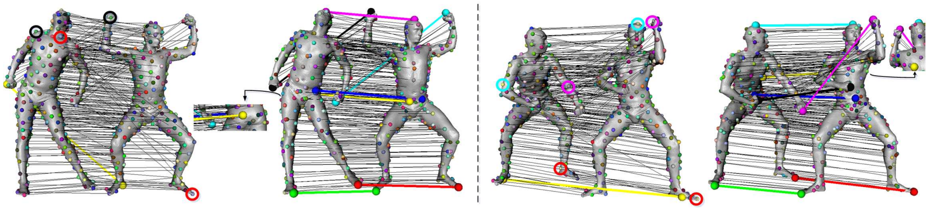

Figure 4.23: Dense maps between SCAPE models computed by Möbius Voting (left) and our method (right). Some MV matches that correspond to our extremity matches are marked with circles. Yellow lines show the worst matches. Dashed line separates two different pairs.

Finally, the execution times of our shape correspondence algorithm on a 2.53GHz PC is 150, 175, 145, 91, 557, 1216, and 22 seconds for Horse↔Horse, Centaur↔Horse, Seahorse↔Horse, Centaur↔Human, SHREC-iso↔SHREC-iso, Gorilla↔Human, and SCAPE↔SCAPE, respectively. The times are dominated by the voting module which creates and samples regions of interests whereas the fast ranking phase just demands shortest path distances between few number of extremity samples. The relatively high execution time on Gorilla↔Human is mainly due to 15 samples in matching as opposed to the typical 10 samples for the others. The fastest runs are on SCAPE↔SCAPE pairs, dealing with only ∼6 samples.

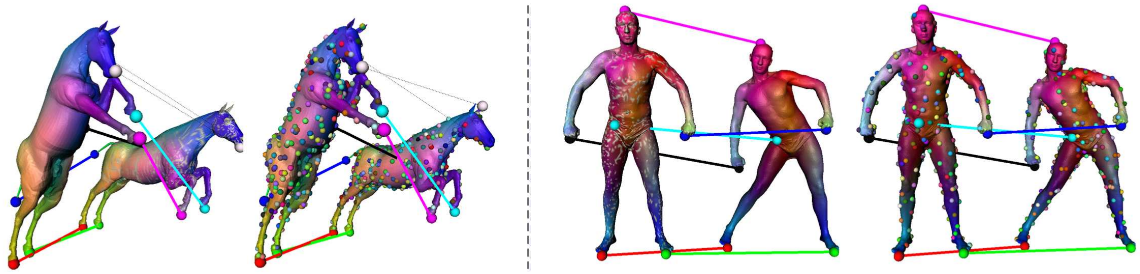

Figure 4.24: Blended Intrinsic Maps (left) vs. our method (right) on two different shape classes. The color of each source vertex is transferred to the corresponding target vertex where unmatched vertices are painted in grey. Our dense map used in interpolation is shown as spheres of matching colors.

4.2.9 Conclusion

We have presented a 3D correspondence algorithm that handles partially and/or nearly isometric shape pairs using a rank-and-vote-and-combine (RAVAC) algorithm. We summarize our findings and conclusions below:

- As an isometric shape correspondence method, RAVAC is very general and can be applied to shapes that may have multiple common parts at arbitrary scales as well as parts that are not similar.

- With RAVAC, we can establish correspondences which are partial and dense at the same time.

- The performance of RAVAC is especially good at shape extremities.

- RAVAC, designed for the partial correspondence problem, can handle the simpler complete correspondence naturally.

- RAVAC can also handle a harder variant of the complete correspondence problem that involves shapes which exhibit local similarities but also large deviations from global isometry, such as Gorilla↔Human pair, as long as semantically similar parts are represented by sufficient number of samples, which is 3 at the least.

	Extremities (MV, Our method)	5-matches (MV-top5, Our counterparts)
Dataset	$(D_{\mathrm{grd}}, D_{\mathrm{grd}})$	$(D_{\mathrm{grd}}, D_{\mathrm{grd}})$
Horse↔Horse	(.189, .028)	(.014, .066)
Horse↔Horse-part	(.281, .045)	(.039, .089)
Centaur↔Horse	(.348, .046)	(.025, .133)
Seahorse↔Horse	(n/a, .071)	(n/a, n/a)
Centaur↔Human	(n/a, .078)	(n/a, n/a)
SHREC-iso↔SHREC-iso	(.053, .003)	(.002, .044)
SHREC-part↔SHREC-iso	(n/a, .049)	(n/a, n/a)
SHREC-part↔SHREC-sca	(n/a, .051)	(n/a, n/a)
Gorilla↔SHREC-null	(n/a, .065)	(n/a, n/a)
SCAPE↔SCAPE	(.182, .004)	(.007, .045)

Table 4.2: Quantitative evaluation of our method in comparison with MV [31].

	Extremities (BIM, Our method)	Full dense map (BIM, Our method)
Dataset	$(D_{\mathrm{grd}}, D_{\mathrm{grd}})$	$(D_{\mathrm{grd}}, D_{\mathrm{grd}})$
Horse↔Horse	(.007, .024)	(.019, .037)
SCAPE↔SCAPE	(.012, .017)	(.042, .051)

Table 4.3: Quantitative evaluation of our method in comparison with BIM [21].

- RAVAC performs well on shape pairs with quite small similarity overlap, such as Centaur↔Human pair.

- RAVAC is based on well-established paradigms such as combinatorial optimization (ranking) and bipartite graph matching (voting), all guided by isometric distortion measurements performed in the original 3D Euclidean space where isometry is originally defined, hence as free of embedding errors.

- We avoid the symmetric flip problem which is common to sparse correspondence methods by incorporating information via auxiliary dense sampling and matching.

- RAVAC has no restriction on shape topology as long as geodesics can reliably be computed on the surfaces of the given shapes, as demonstrated on SHREC'11 shapes containing holes.

Whether partial or complete, our algorithm is designed for isometric shapes, though the experiments show that our algorithm works well also for nearly isometric shapes. Yet there remains future work to address the tradeoff between the accuracy of the geodesic metric currently in use and the topological noise robustness of the diffusion-based metrics to be tested.

Chapter 5

CONCLUSIONS AND FUTURE WORK

We have presented a package of novel and computationally efficient algorithms that can establish coarse and dense correspondences between two complete or partially overlapping isometric (or nearly isometric) shapes (Section 1.4). Through this journey the following contributions have been achieved.

- Four new sampling algorithms (Sections 3.1.3, 3.2.3, 4.1.3, 4.2.2).
- Isometric distortion measurement functions (Equations 3.1, 4.1, 4.10) that guide the optimizations.
- Isometric distortion optimizers based on solid paradigms (Sections 3.1.6.1, 4.1.5, 4.2.4.3, 3.1.6.2, 3.1.4, 3.2.4.1, 4.1.4, 4.2.3, 4.2.4).
- For all isometric shape correspondence algorithms, sticking with the 3D Euclidean space where isometry is originally defined as well as insensitivity to shape topology and to the peculiarities of the particular triangulation.
- The fastest computational complexity on dense matching (Section 3.2.6), and thorough complexity analyses of all algorithms
- Correspondence tracking mechanism addressing the symmetric flip problem (Sections 3.3.4, 4.1.5).
- Partial correspondence between shapes with significantly small similarity overlap, the smallest indeed to the best of our knowledge (Section 4.2).
- Correspondences that are partial and dense at the same time (Sections 4.1, 4.2).

There remains a future work to address the tradeoff between the accuracy of the geodesic metric currently in use by all of our algorithms and the topological noise robustness of the diffusion-based metrics to be tested.

As another future research direction, it may be desirable to incorporate more shapes into the process to establish or improve correspondences in between. While correspondence methods between more than two shapes can benefit from the existing pairwise methods, one also needs to consider the new parameters, the difficulties as well as the opportunities that are specific to this new problem. For correspondence generation within a mesh sequence, such an opportunity can be the injection of the Euclidean distance into the optimization process as internally isometric shapes of the consecutive frames are expected to be externally close as well. For improvement of a group of correspondence, on the other hand, one may force consistency rules such that all cycles of consistent maps return to identity, i.e., the correspondence path started from vertex v on the first shape returns to v itself after being tracked over a collection of pairwise maps.

BIBLIOGRAPHY

[1] M. Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21:173–196, 2002.

[2] M. Kilian, N. J. Mitra, and H. Pottman. Geometric modeling in shape space. *Proc. SIGGRAPH*, 26:1–8, 2007.

[3] Z. Popovic B. Allen, B. Curless. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, 2003.

[4] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *Proc. SIGGRAPH*, 23(3):870–877, 2004.

[5] P. J. Besl and N. D. Mckay. A method for registration of 3D shapes. *IEEE Trans. PAMI*, 14:239–256, 1992.

[6] D. Anguelov, P. Srinivasan, H. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. *Proc. NIPS*, 37:33–40, 2004.

[7] Y. Pekelny and C. Gotsman. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum*, 27(2):399–408, 2008.

[8] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. Seidel. Motion capture using joint skeleton tracking and surface estimation. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Trans. PAMI*, 25:1285–1295, 2003.

[10] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design*, 39(5):398–407, 2007.

[11] R. Liu and H. Zhang. Segmentation of 3d meshes through spectral clustering. *Proc. Pacific Graphics*, pages 298–305, 2004.

[12] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3d models. *Computers and Graphics (Shape Modeling International)*, 2009.

[13] V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: Constructing constrained texture maps. *Proc. SIGGRAPH*, pages 326–333, 2003.

[14] R. W. Sumner and J. Popovic. Deformation transfer for triangle meshes. *Proc. SIGGRAPH*, 23(3), 2004.

[15] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. *Proc. SIGGRAPH*, pages 69–76, 1999.

[16] B. C. Munsell, P. Dalal, and S. Wang. Evaluating shape correspondence for statistical shape analysis: A benchmark study. *IEEE Trans. PAMI*, 30(11):2023–2039, 2008.

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

[18] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes.* Springer, 2008.

[19] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. An optimization approach to improving collections of shape maps. *Computer Graphics Forum (SGP'11)*, 30(5):1481–1491, 2011.

[21] V. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *Proc. SIGGRAPH*, 30(4), 2011.

[22] Y. Sahillioğlu and Y. Yemez. 3d shape correspondence by isometry-driven greedy optimization. *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 453–458, 2010.

[23] Y. Sahillioğlu and Y. Yemez. Minimum-distortion isometric shape correspondence using em algorithm. *IEEE Trans. PAMI (to appear)*, 2012.

[24] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum (SGP'11)*, 30(5):1461–1470, 2011.

[25] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine isometric shape correspondence by tracking symmetric flips. *Computer Graphics Forum (in revision)*, 2012.

[26] Y. Sahillioğlu and Y. Yemez. Scale normalization for isometric shape matching. *Proc. Pacific Graphics (to appear)*, 2012.

[27] Y. Sahillioğlu and Y. Yemez. Partial 3d correspondence from shape extremities.

[28] T. Funkhouser and P. Shilane. Partial matching of 3d shapes with priority-driven search. *Symposium on Geometry Processing (SGP)*, 2006.

[29] A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. Isometric registration of ambiguous and partial data. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[30] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24:509–523, 2002.

[31] Y. Lipman and T. Funkhouser. Mobius voting for surface correspondence. *ACM Trans. Graph.*, 28(3), 2009.

[32] Q. Huang, B. Adams, M. Wicke, and L. Guibas. Non-rigid registration under isometric deformations. *Symposium on Geometry Processing (SGP)*, pages 1149–1458, 2008.

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

[34] C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios. Discrete minimum distortion correspondence problems for non-rigid shape matching. *Proc. Scale Space and Variational Methods*, 2011.

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

[36] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[37] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchil. Deformation-driven shape correspondence. *Computer Graphics Forum*, pages 1431–1439, 2008.

[38] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing*, 28(5):1812–1836, 2006.

[39] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. *IEEE Int. Conf. on Shape Modeling and Applications (SMI)*, pages 118–129, 2006.

[40] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Proc. Conf. Robotics and Automation*, 1991.

[41] N. J. Mitra, N. Gelfand, H. Pottman, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. *Symposium on Geometry Processing (SGP)*, pages 23–32, 2004.

[42] N. Gelfand, N. Mitra, L. Guibas, and H. Pottman. Robust global registration. *Symposium on Geometry Processing (SGP)*, 2005.

[43] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *Int. Conf. on 3D Digital Imaging and Modeling (3DIM)*, 2001.

[44] W. Chang and M. Zwicker. Automatic registration for articulated shapes. *Computer Graphics Forum*, 27(5):1459–1468, 2008.

[45] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.

[46] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (2nd ed.)*. MIT Press and McGraw-Hill, 2001.

[47] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):569–615, 1987.

[48] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. *Proc. SIGGRAPH*, 2001.

[49] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proc. National Academy of Sciences (PNAS)*, (15):8431–8435, 1998.

[50] O. Weber, Y. S. Devir, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Parallel algorithms for approximation of distance maps on parametric surfaces. *Proc. SIGGRAPH*, 27(4), 2008.

[51] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3):553–560, 2005.

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

[53] F. Fouss, A. Pirotte, J. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19, 2007.

[54] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens. Graph nodes clustering based on the commute-time kernel. *Proc. of 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2007.

[55] Y. Lipman, R. Rustamov, and T. Funkhouser. Biharmonic distance. *ACM Trans. Graph.*, 29(3), 2010.

[56] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *VisMath*, 2002.

[57] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5), 2009.

[58] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[59] R. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. *Symposium on Geometry Processing (SGP)*, pages 225–233, 2007.

[60] H. Kriegel T. Seidl M. Ankerst, G. Kastenmuller. 3d shape histograms for similarity search and classification in spatial databases. *Proc. SSD*, 1999.

[61] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.*, 24(4):249–259, 2008.

[62] M. Spagnuolo D. Saupe R. Ruggeri, G. Patane. Spectral-driven isometry-invariant matching of 3d shapes. *Intl. Journal of Computer Vision (IJCV)*, 89(2-3):248–265, 2010.

[63] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Processing*, 6:1305–1315, 1997.

[64] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[65] G. Peyre and L. Cohen. Surface segmentation using geodesic centroidal tesselation. *Proc. Int'l Symp. 3D Data Processing Visualization Transmission*, pages 995–1002, 2004.

[66] D. Nehab and P. Shilane. Stratified point sampling of 3D models. *Eurographics Symposium on Point-Based Graphics*, pages 49–56, 2004.

[67] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *Proc. Pacific Graphics*, 21:649–658, 2005.

[68] J. Tierny, J. Vandeborre, and M. Daoudi. Enhancing 3d mesh topological skeletons with discrete contour constrictions. *The Visual Computer*, 24:155–172, 2008.

[69] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, pages 325–338, 1966.

[70] V. de Silva and J.B. Tenenbaum. Global versus local methods for nonlinear dimensionality reduction. *Proc. NIPS*, pages 721–728, 2003.

[71] G. Ziegelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. Vis. & Comp. Graphics*, 8:198–207, 2002.

[72] S. Wuhrer, C. Shu, and P. Bose. Posture invariant correspondence of triangular meshes in shape space. *Int. Conf. on 3D Digital Imaging and Modeling (3DIM)*, 2009.

[73] M. Isenburg and P. Lindstrom. Streaming meshes. *IEEE Visualization*, pages 30–37, 2005.

[74] G. Kirchhoff. Uber die auflosung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer strome gefuhrt wird. *Ann. Phys. Chem*, pages 497–508, 1847.

[75] J. Feidman. Computing betti numbers via combinatorial laplacians. *Proc. 28th Sympos. Theory Comput.*, pages 386–391, 1996.

[76] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003.

[77] D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[78] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2301–2306, 2004.

[79] B. Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. *IEEE Int. Conf. on Shape Modeling and Applications (SMI)*, 2006.

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

[81] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graph.*, 22(3):358–363, 2003.

[82] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, and G. Sapiro. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):1–8, 200.

[83] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 21(3):210–218, 2002.

[84] A. Sheffer and E. Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers*, 17(3):326–337, 2001.

[85] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, pages 15–36, 1993.

[86] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. *Proc. SIGGRAPH*, pages 861–869, 2004.

[87] M. Carcassoni and E. R. Hancock. Correspondence matching with modal clusters. *IEEE Trans. PAMI*, 25(12):1609–1615, 2003.

[88] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1482–1489, 2005.

[89] M. Ovsjanikov, J. Sun, and L. Guibas. Global intrinsic symmetries of shapes. *Computer Graphics Forum*, 27(5):1341–1348, 2007.

[90] A. Sharma and R. P. Horaud. Shape matching based on diffusion embedding and on mutual isometric consistency. *Proc. Workshop on Nonrigid Shape Analysis and Deformable Image Alignment (NORDIA)*, 2010.

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

[92] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *Proc. ECCV*, pages 596–609, 2008.

[93] O. K.-C. Au, D. Cohen-Or, C.-L. Tai, H. Fu, and Y. Zheng. Electors voting for fast automatic shape correspondence. *Computer Graphics Forum (Proc. Eurographics)*, 29(2):645–654, 2010.

[94] G. Peyre and L. D. Cohen. Geodesic remeshing using front propagation. *Proc. IEEE Workshop on Variational, Geometric and Level Set Methods*, pages 33–40, 2003.

[95] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

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

[97] P. Sand, L. McMillan, and J. Popovic. Continuous capture of skin deformation. *Int. Conf. on Computer Graphics and Interactive Techniques*, 2003.

[98] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *Proc. SIGGRAPH*, 2008.

[99] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3d mesh segmentation. *Proc. SIGGRAPH*, 2009.

[100] R. Liu, V. Jain, and H. Zhang. Subsampling for efficient spectral mesh processing. *Proc. Computer Graphics International*, 2006.

[101] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser. Mobius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Symposium on Geometry Processing)*, 29(5), 2010.

[102] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. *Proc. Workshop on Nonrigid Registration and Tracking through Learning (NRTL)*, 2005.

[103] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Full and partial symmetries of non-rigid shapes. *Intl. Journal of Computer Vision*, 89(1):18–39, 2009.

[104] D. Giorgi, S. Biasotti, and L. Paraboschi. Shrec: shape retrieval contest: Watertight models track. 2007.

[105] E. Boyer, A. Bronstein, M. Bronstein, B. Bustos, and et al. Shrec 2011: robust feature detection and description benchmark. *Eurographics Workshop on 3D Object Retrieval*, 2011.

[106] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[107] J. Pokrass, A. M. Bronstein, and M. M. Bronstein. A correspondence-less approach to matching of deformable shapes. *Proc. Scale Space and Variational Methods*, 2011.