



**Max-Planck-Institut für Informatik**  
**Computer Graphics Group**  
**Saarbrücken, Germany**

# **Deformable Shape Matching**

eingereicht von **Art Tevs**

Dissertation zur Erlangung des Grades  
der Doktorin/des Doktors der Ingenieurwissenschaften  
der Naturwissenschaftlich-Technischen Fakultäten  
der Universität des Saarlandes

MPI Informatik  
Campus E1.4  
66123 Saarbrücken

Saarbrücken, 10. August 2011

**Betreuender Hochschullehrer - Supervisor**

Prof. Dr. Hans-Peter Seidel,  
MPI Informatik, Saarbrücken, Germany

**Gutachter - Reviewer**

Prof. Dr. Hans-Peter Seidel,  
MPI Informatik, Saarbrücken, Germany

Dr. Michael Wand,  
Universität des Saarlandes & MPI Informatik, Saarbrücken, Germany

Dr. Ivo Ihrke,  
Universität des Saarlandes, Saarbrücken, Germany

Prof. Leonidas J. Guibas,  
Stanford University, Stanford, CA, United States

**Dekan - Dean**

Prof. Dr. Holger Hermanns,  
Universität des Saarlandes, Saarbrücken, Germany

**Kolloquium - Examination**

*Datum - Date*

21. Dezember 2011

*Vorsitzender - Chair*

Prof. Dr. Philipp Slusallek,  
Universität des Saarlandes, Saarbrücken, Germany

*Prüfer - Examiners*

Prof. Dr. Hans-Peter Seidel,  
MPI Informatik, Saarbrücken, Germany

Dr. Michael Wand,  
Universität des Saarlandes & MPI Informatik, Saarbrücken, Germany

Dr. Ivo Ihrke,  
Universität des Saarlandes, Saarbrücken, Germany

*Protokoll - Reporter*

Dr. Kiran Varanasi,  
MPI Informatik, Saarbrücken, Germany

### **Eidesstattliche Versicherung**

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, Dezember 22. 2011

---

(Art Tevs)

### **Declaration of Consent**

Herewith I agree that my thesis will be made available through the library of Computer Science Department.

Saarbrücken, Dezember 22. 2011

---

(Art Tevs)



---

## Abstract

Deformable shape matching has become an important building block in academia as well as in industry. Given two three dimensional shapes  $A$  and  $B$  the deformation function  $f$  aligning  $A$  with  $B$  has to be found. The function is discretized by a set of corresponding point pairs. Unfortunately, the computation cost of a brute-force search of correspondences is exponential. Additionally, to be of any practical use the algorithm has to be able to deal with data coming directly from 3D scanner devices which suffers from acquisition problems like noise, holes as well as missing any information about topology.

This dissertation presents novel solutions for solving shape matching: First, an algorithm estimating correspondences using a randomized search strategy is shown. Additionally, a planning step dramatically reducing the matching costs is incorporated. Using ideas of these both contributions, a method for matching multiple shapes at once is shown. The method facilitates the reconstruction of shape and motion from noisy data acquired with dynamic 3D scanners. Considering shape matching from another perspective a solution is shown using Markov Random Fields (MRF). Formulated as MRF, partial as well as full matches of a shape can be found. Here, belief propagation is utilized for inference computation in the MRF. Finally, an approach significantly reducing the space-time complexity of belief propagation for a wide spectrum of computer vision tasks is presented.

---

## Kurzfassung

Anpassung deformierbarer Formen ist zu einem wichtigen Baustein in der akademischen Welt sowie in der Industrie geworden. Gegeben zwei dreidimensionale Formen  $A$  und  $B$ , suchen wir nach einer Verformungsfunktion  $f$ , die die Deformation von  $A$  auf  $B$  abbildet. Die Funktion  $f$  wird durch eine Menge von korrespondierenden Punktepaaren diskretisiert. Leider sind die Berechnungskosten für eine Brute-Force-Suche dieser Korrespondenzen exponentiell. Um zusätzlich von einem praktischen Nutzen zu sein, muss der Suchalgorithmus in der Lage sein, mit Daten, die direkt aus 3D-Scanner kommen, umzugehen. Bedauerlicherweise leiden diese Daten unter Akquisitionsproblemen wie Rauschen, Löcher sowie fehlender Topologieinformation.

In dieser Dissertation werden neue Lösungen für das Problem der Formanpassung präsentiert. Als erstes wird ein Algorithmus gezeigt, der die Korrespondenzen mittels einer randomisierten Suchstrategie schätzt. Zusätzlich wird anhand eines automatisch berechneten Schätzplanes die Geschwindigkeit der Suchstrategie verbessert. Danach wird ein Verfahren gezeigt, dass die Anpassung mehrerer Formen gleichzeitig bewerkstelligen kann. Diese Methode ermöglicht es, die Bewegung, sowie die eigentliche Struktur des Objektes aus verrauschten Daten, die mittels dynamischer 3D-Scanner aufgenommen wurden, zu rekonstruieren. Darauf folgend wird das Problem der Formanpassung aus einer anderen Perspektive betrachtet und als Markov-Netzwerk (MRF) reformuliert. Dieses ermöglicht es, die Formen auch stückweise aufeinander abzubilden. Die eigentliche Lösung wird mittels Belief Propagation berechnet. Schließlich wird ein Ansatz gezeigt, der die Speicher-Zeit-Komplexität von Belief Propagation für ein breites Spektrum von Computer-Vision Problemen erheblich reduziert.

I have not failed. I've just  
found 10,000 ways that won't  
work.

---

*Thomas A. Edison (1847-1931)*





# CONTENTS

---

<b>I</b>	<b>Introduction</b>	<b>19</b>
<b>1</b>	<b>Problem Statement and Background</b>	<b>27</b>
1.1	Shape matching . . . . .	29
1.1.1	Shape definitions . . . . .	31
1.1.2	Allow deformations . . . . .	32
1.1.3	Topological noise . . . . .	35
1.2	Deformable matching and inference problems . . . . .	36
1.2.1	Belief Propagation . . . . .	38
1.3	Applications for deformation matching . . . . .	40
1.4	Thesis outline . . . . .	42
1.5	List of Publications . . . . .	43
<b>2</b>	<b>Related Work</b>	<b>45</b>
<b>II</b>	<b>Shape Matching</b>	<b>57</b>
<b>3</b>	<b>Pairwise Shape Matching by Randomized Sampling</b>	<b>63</b>
3.1	Input Data and Preprocessing . . . . .	64
3.2	Feature Points . . . . .	64
3.3	RANSAC Subgraph Extraction . . . . .	65
3.4	Handling Topological Noise . . . . .	67
3.5	Tangent-space optimization . . . . .	68
3.6	Approximation of Geodesics . . . . .	70
3.7	Inserting Dense Secondary Features . . . . .	71
3.8	Ranking of Matching Alternatives . . . . .	73
3.9	Implementation and Evaluation . . . . .	73
3.10	Discussion . . . . .	76
<b>4</b>	<b>PLANSAC: Plan First Before Execute</b>	<b>79</b>
4.1	Isometry invariant matching . . . . .	81

4.2	Shape matching by random sampling . . . . .	82
4.3	The planned sampling algorithm . . . . .	84
4.4	Results . . . . .	88
4.5	Discussion . . . . .	95
<b>5</b>	<b>Animation Reconstruction</b>	<b>97</b>
5.1	Overview . . . . .	100
5.1.1	Problem Statement . . . . .	100
5.1.2	Data Structures . . . . .	101
5.1.3	Robust Intrinsic Matching . . . . .	103
5.1.4	Pipeline Overview . . . . .	104
5.2	Landmarks . . . . .	105
5.2.1	Continuous Landmark Tracking . . . . .	105
5.2.2	Connecting Broken Tracks . . . . .	107
5.3	Intrinsic Charting . . . . .	108
5.3.1	Building Single-Frame Charts . . . . .	108
5.3.2	Merging Two Charts Given Landmarks . . . . .	109
5.3.3	Resampling a Chart . . . . .	111
5.3.4	Detecting Apparent Topology Changes . . . . .	113
5.4	Extrinsic Matching . . . . .	115
5.4.1	Pairwise Local Matching . . . . .	115
5.4.2	Animation Fitting . . . . .	117
5.5	Reconstruction Pipeline . . . . .	118
5.5.1	Building i-Charts . . . . .	118
5.5.2	Outer Loop: Building p-Charts . . . . .	120
5.5.3	Final Optimization . . . . .	121
5.6	Results . . . . .	122
5.6.1	Synthetic Tests . . . . .	122
5.6.2	Real-world Scanner Data . . . . .	123
5.7	Discussion and Limitations . . . . .	128
5.8	Conclusions . . . . .	129
<b>6</b>	<b>Part Summary</b>	<b>131</b>

<b>III</b>	<b>Inference Problems</b>	<b>133</b>
<b>7</b>	<b>Shape Matching as an Inference Problem</b>	<b>141</b>
7.1	Model definition . . . . .	141
7.1.1	Sum-Product Belief Propagation . . . . .	145
7.2	Implementation . . . . .	145
7.2.1	Exploiting parallelism . . . . .	146
7.3	Partial symmetry detection . . . . .	148
7.4	Results . . . . .	149
7.5	Symmetry breaking . . . . .	151
7.6	Conclusion . . . . .	152
<b>8</b>	<b>Wavelet Belief Propagation</b>	<b>155</b>
8.1	Wavelet representation . . . . .	155
8.2	Wavelet Product Integrals . . . . .	157
8.3	Adaptive Approximate Summation . . . . .	159
8.4	Conversion to the Wavelet Domain . . . . .	163
8.5	Evaluation and Applications . . . . .	166
8.6	Applications . . . . .	166
8.7	Baseline methods . . . . .	167
8.8	Results . . . . .	168
8.9	Summary . . . . .	171
<b>9</b>	<b>Part Summary</b>	<b>175</b>
<b>IV</b>	<b>Epilogue</b>	<b>177</b>
9.1	Discussion and future work . . . . .	182



# LIST OF FIGURES

---

1.1	3D scanner examples . . . . .	28
1.2	Surface scan from multiple views . . . . .	29
1.3	ICP illustration . . . . .	30
1.4	Illustration of isometric deformation . . . . .	33
1.5	Naïve descriptor matching fails . . . . .	34
1.6	Topological noise . . . . .	36
1.7	Example of a Bayesian network . . . . .	37
1.8	Passing a message with belief propagation . . . . .	40
1.9	Matching for biology . . . . .	41
2.1	Deformation ICP by Li et al. [77] . . . . .	46
2.2	Spectral matching . . . . .	47
2.3	PROSAC and RANSAC matching . . . . .	49
2.4	Template-based animation reconstruction . . . . .	53
3.1	RANSAC-matching pipeline overview . . . . .	64
3.2	Tangent space optimization . . . . .	69
3.3	Approximated geodesics . . . . .	71
3.4	Secondary features . . . . .	72
3.5	Matching results (guy) . . . . .	74
3.6	Matching results (arm) . . . . .	74
3.7	Matching results (face) . . . . .	75
3.8	Synthetic dataset results (horse) and (dragon) . . . . .	76
3.9	Synthetic dataset illustrates robustness . . . . .	77
3.10	Ambiguous matching of a star dataset . . . . .	77
3.11	Ambiguous matching of a finger . . . . .	78
4.1	Landmarks computed for a human face . . . . .	80
4.2	Wave-descriptor illustrated . . . . .	82
4.3	Descriptor examination . . . . .	84
4.4	Entropy distribution of the most descriptive points . . . . .	86
4.5	Illustration of landmark stability . . . . .	87

## List of Figures

---

4.6	Comparison between FPS and Entropy minimization . . . . .	88
4.7	Hand data set matching results . . . . .	90
4.8	Examination of matching of featureless objects . . . . .	91
4.9	Comparison to previous work . . . . .	93
4.10	Multiframe matching application . . . . .	94
4.11	Database matching application . . . . .	95
4.12	Matching results for data sets in different poses . . . . .	96
5.1	Teaser animation reconstruction . . . . .	98
5.2	Pipeline overview of animation cartography . . . . .	104
5.3	ICP tracking . . . . .	106
5.4	Hand dataset and continous tracking . . . . .	108
5.5	Matching probabilities and the variance of unreliable matches . . . . .	109
5.6	Graph merging . . . . .	111
5.7	Resampling merged charts . . . . .	112
5.8	Illustration of improved detection of topology change . . . . .	114
5.9	Topology consistency cut on the puppet dataset . . . . .	116
5.10	Space time diagram of i-chart consutrction . . . . .	119
5.11	Effect of robust matching score . . . . .	121
5.12	Comparison to the algorithm of Wand et al. [152] . . . . .	122
5.13	Results for “Abhijeet”, “Kicker” and “Woman” datasets . . . . .	124
5.14	Results for “Saskie”, “Puppet” and “Face’ data sets . . . . .	125
6.1	Marginal distribution in an optical flow example . . . . .	139
7.1	NMI of two spin images . . . . .	142
7.2	Global and local symmetry detection in a star data set . . . . .	143
7.3	Graphical model network illustration . . . . .	144
7.4	Datastructure to increase performance of LBP on GPU . . . . .	148
7.5	Results of different data sets . . . . .	153
7.6	Cuboctahedron dataset . . . . .	154
7.7	Lantern dataset . . . . .	154
8.1	2D Haar-wavelet basis function as a tensor product . . . . .	158
8.2	Adaptive matrix vector product illustration . . . . .	162
8.3	Teddy dataset . . . . .	169
8.4	Synthetic experiment for 1D wavelet belief propagation . . . . .	170

## List of Figures

---

8.5	Evaluation of time complexity . . . . .	171
8.6	Performance evaluation of different BP implementations . . . . .	172
8.7	BeanBag dataset results . . . . .	173
8.8	City hall dataset results . . . . .	174





# LIST OF TABLES ---

3.1	Performance table for different data sets matched with RANSAC-like approach . . . . .	76
4.1	Timings and errors achieved for different data sets with PLANSAC approach . . . . .	89
5.1	Statistics for processing the individual data sets with our animation cartography approach . . . . .	127
7.1	Matchig results, timings and memory consumption of symmetry-detection employing Belief Propagation . . . . .	149



## **Part I**

# **Introduction**



# INTRODUCTION

---



Shape matching and in particular *deformable* shape matching is an important building block for a lot of research fields like computer vision, computer graphics, biology as well as an indispensable tool in industry like movies, games, robotic, quality assessment, etc. Given two three dimensional shapes  $A$  and  $B$  a deformation function, which applied to the source shape  $A$  aligns it with target shape  $B$ , has to be found. A simple example of this deformation function would be a rigid body transformation to align the same shape in different poses. However, this thesis deals with a more complex problem where the shape is allowed to deform.

This thesis presents several solutions for the *isometric* shape matching problem. Isometric shape matching is based on the assumption that on-surface distances, also known as *intrinsic distances*, between any pair of points remain con-

---

stant regardless of the deformation. For example the distance between two fingers measured along the shortest path over the hand remains the same even if the hand is bent. In general, the deformation function is represented by a set of correspondences between the points on the source shape and the target shape. Hence the problem of deformable shape matching boils down to the problem of estimating these correspondences.

There are two main challenges when trying to estimate the correspondences: The first one is complexity: For two shapes  $A$  and  $B$  consisting of  $N$  and  $M$  points respectively, there are  $\mathcal{O}(N \cdot M)$  corresponding point pairs. However, only small subset of them make sense, as for example it does not make sense to put a forefinger into the correspondence to the ring finger. To find a meaningful correspondence subset of the size  $k$  one need  $\mathcal{O}\left(\binom{N \cdot M}{k}\right)$  trials, which is not efficiently solvable using a naïve brute-force search. The second challenge is data representation. To be of any practical use the correspondence estimation algorithm must be able to deal with data coming from 3D scanner devices. Such data suffer from noise, acquisition holes and missing geometric topology information, i.e., points are unstructured and unconnected. This thesis makes several contributions to overcome the shortcomings, as summarized in the next paragraphs.

### **RANSAC: randomized correspondence estimation**

In Chapter 3 a correspondence estimation algorithm is shown which is capable of solving the deformable shape matching problem on data sets coming directly from the three dimensional acquisition devices. In order to deal with the typical data acquired by a scanning device, the method uses a RANSAC-like search for correct correspondences. RANSAC [43] is an approach to estimate a model given a set of measuring points including large amount of outliers. In case of the approach presented in the chapter, the correspondence matching model is a set of meaningful correspondences which respects the isometry criterion. Since the model is self-consistent, i.e., there is no pair of points which violates the isometry criterion, it is enough to find a small set of meaningful correspondences which restrict the further search to a tiny subspace. Furthermore a method which is able to compute correspondences using a robust importance sampling routine to bias the randomized search towards a suitable solution is shown. The presented technique might be also applied to other graph matching related problems.

---

The corresponding paper: “Isometric Registration of Ambiguous and Partial Data” has been published in the proceedings of *Computer Vision and Pattern Recognition, CVPR’09* [141].

### **PLANSAC: towards simplification of the matching problem**

In Chapter 4 the matching problem will be investigated deeply and a method which computes a *plan*, i.e., a set of surface points to be used by the randomized correspondence estimation, as shown in Chapter 3, is presented. This plan minimizes the effort, i.e., the number of random trials required to estimate the right correspondence model and hence to compute the deformation function. A plan is represented by a set of *landmarks* - points distributed over the shape’s surface. These landmarks are automatically placed in such a way, that the complexity of the shape matching process is reduced. Currently, no definition for the complexity of matching two shapes exists. Experiments with different types of shapes reveal that the number of landmark points distributed over the shape hints of a possible characterization for the shape complexity. PLANSAC results suggest that it is worthwhile to analyze the actual problem first in order to reduce its complexity.

This contribution titled “Intrinsic Shape Matching by Planned Landmark Sampling” has been published in the proceedings of *Eurographics 2011* [140].

### **AnimRec: matching multiple shapes at once**

Using the definition of landmarks and the RANSAC-like correspondence estimation algorithm a solution for *animation reconstruction* is presented in Chapter 5. Here, the thesis provides a method to reconstruct the motion and the shape of a moving and isometrically deforming object from a set of partially observed data as provided, e.g., by dynamic 3D scanning devices. The main challenge here, is the missing data in the spatial and temporal domain. It is common that when performing a dynamic scan of a human face it can be acquired only partially, e.g., when scanning a face both ears are not visible by the scanning device at the same time. It is assumed that there is no a priori information about the given scanned object. This is known as *template-free* matching and although it reflects the real-world scanning scenario very well it creates additional challenges, since there is no guidance path in the form of a template the algorithm can follow. In Chapter 5 a solution to this problem is presented, which will be applied to several data sets

---

acquired with dynamic 3D scanners.

The corresponding publication: “Animation Cartography - Intrinsic Reconstruction of Shape and Motion” has been submitted and is subject to minor revisions in the *ACM Transactions on Graphics*, 2012 [139].

### **Selfmatching for symmetry detection**

Chapter 7 takes a look on the problem of shape matching from another perspective. The problem of shape matching is redefined as a Markov random field (MRF) where each node models a point on the source shape. Here, the goal is to find the best solution over the label set, which is described by the target points. The alternative formulation of the shape matching problem has the advantage of being able to find partial matches as well as multiple matches at once. This is also known as *symmetry detection*, where the goal is to find parts of an object which are similar. Several experiments will be shown where isometrically deformed parts would be identified in the object itself as well as on another object. Additional advantage of the MRF formulation, as shown in this thesis, is its ability to vary one parameter to guide the algorithm to find either more local or more global similarities. The inference computation of the MRF graph is performed with a message passing algorithm: *loopy belief propagation* [105].

This approach titled “A Probabilistic Framework for Partial Intrinsic Symmetries in Geometric Data” has been published in the conference proceedings of *International Conference on Computer Vision, ICCV'09* [71].

### **Wavelet space for speeding up belief propagation**

Chapter 7 also shows that the number of states each node in the MRF graph can take is very large. A big disadvantage of belief propagation when handling a very large state space is that one need to “manage” a huge state vector for every node in the MRF graph. Hence one iteration step of belief propagation is not only space but also very time consuming. In case of grids with a clique of size 2, the number of operations for one iteration step is  $\mathcal{O}(nk^2)$ , where  $k$  is the number of states and  $n$  the number of nodes. In Chapter 8 a method for making this complexity data-dependent is shown. This is achieved by moving the whole inference computation into the wavelet domain and hence compressing the state vector. The computation is performed solely in the wavelet domain, dramatically reducing not only the space



---

but also the time-complexity of loopy belief propagation. The benefit of *wavelet belief propagation* is presented on several computer vision tasks, such as image matching and optical flow. The use of wavelet based belief propagation offers not only theoretical but also a practical performance advantages over the existing methods. The results achieved in this chapter might be of wide applicability for research fields beyond the scope of computer graphics or computer vision.

The corresponding paper “Wavelet Belief Propagation for Large Scale Inference Problems” has been published in the proceedings of *Conference on Computer Vision and Pattern Recognition, CVPR’11* [72].



---



# Problem Statement and Background

---

In the last thirty years personal and industrial computers evolved to machines able to perform precise computations with tremendous speed. This progress moved the research towards new fields and provided the opportunity to pursue algorithms which were not possible before. *Computer vision*, for example, has its roots back to late 1970s when researches began to learn about human's perception by emulating this process on the computer.

In general, computer vision works on images which are used for different computer vision tasks, such as stereo reconstruction, image analysis, image registration, image matching etc. Generally, computer vision and computer graphics has a lot of common problems. Computer graphics is mainly used as a term describing anything which is produced on the screen by soft- or hardware. The problem solution presented in this thesis is related to both, computer vision and computer graphics, namely: *deformable shape matching*: Given a pair or a set of three dimensional shapes, a deformation required to align one shape to another has to be found. Various applications exists in the industry handling with this type of problem. An overview of possible applications is given later in this chapter.

Three dimensional shapes can be represented in several ways. Typical representations are *triangle meshes*, i.e., sets of triangles describing the surface of the shape, and *point clouds*. A shape representation by a triangle mesh is widely used in research as well as computer graphics related industries, such as in games, the movie industry, visualization software and so on. Due to their nature triangle meshes approximate the shape's surface continuously, since the surface is represented by small local triangular patches, whereas a point cloud is a set of unstructured, unconnected three dimensional points distributed over the shape's surface. Hence, compared with triangular meshes or any other polygon-like representation, point clouds just approximate the surface by a set of points instead of a locally flat

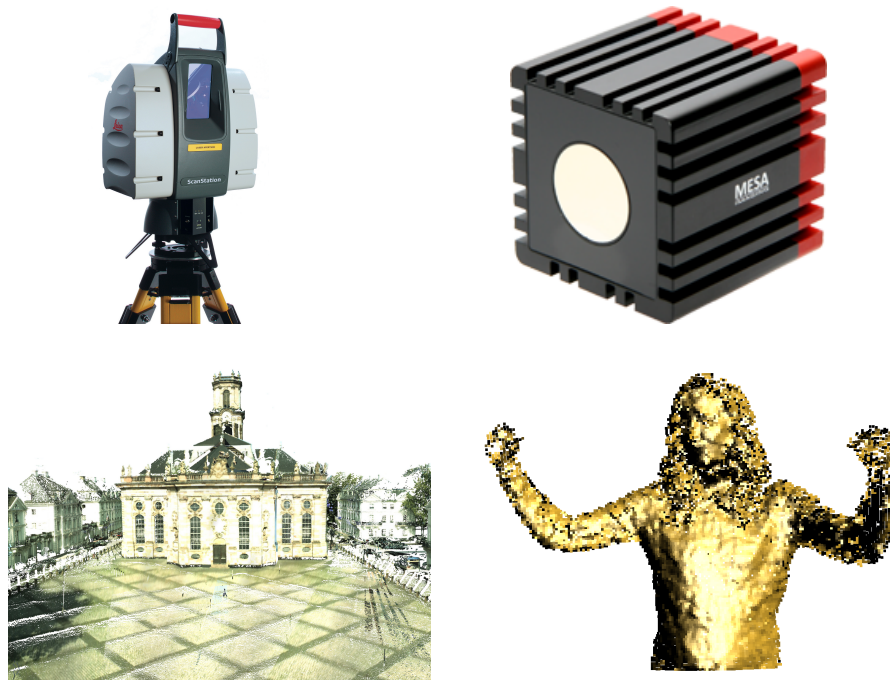


Figure 1.1: (*Top*) Typical 3D scanners based on the time-of-flight principle. Left: a Leica HDS-3000 scanner [73] usually used to scan architectural environment. Right: Swiss Ranger 4000 [85] used for real-time in-door scanning. (*Bottom*) Point clouds acquired with the presented devices. Left: Church as a static object scanned from different positions. Color information acquired with a simple optical camera was overlaid within actual 3D point set. Right: Woman performing movements scanned by the 2D time-of-flight principle. The quality is rather low but more or less typical for todays available hardware.

surface patches. A point cloud can be obtained, for example, by sampling a given shape with a 3D scanner device.

3D scanner devices sample an object point by point with different techniques. Time of flight, for example, measures the time required for a light pulse to reflect from the scanned surface, whereas stripe pattern scanners measure the deformation of a pattern projected on the object. Bernardini and Rushmeier [11] give a small overview over existing 3D scanning techniques in their report. Figure 1.1 shows two typical representatives of these types of scanner devices and their output.

This thesis deals with both representations of geometry: triangle meshes as well as point clouds. A lot of research work can be found in the literature concern-

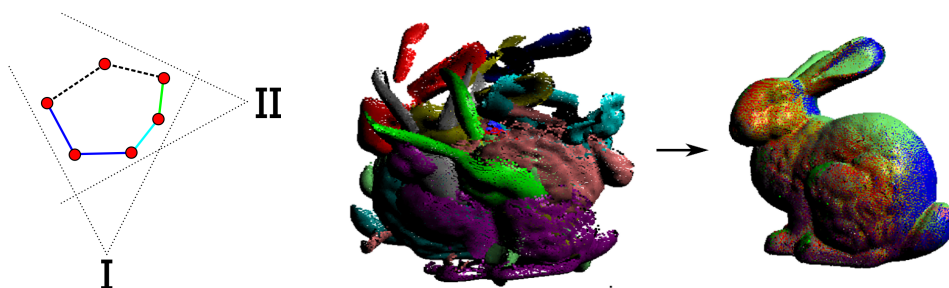


Figure 1.2: *(left)*: An object is scanned from different views, I and II. The surface seen from view I and II is marked blue and green respectively. The surface seen from both views is marked cyan. *(right)*: real-world example of different parts acquired by the scanner of the Stanford bunny object and the final result after registration, courtesy of Bokeloh [15].

ing the reconstruction of a triangulated surface from a point cloud [1, 17, 52, 142]. In general, processing in one or the other domain has its own benefits and pitfalls. Most of the results and algorithms presented in this thesis are applicable in both domains. However, since point clouds best reflects the reality of today’s available scanning hardware, the algorithms presented here are mostly focused on this representation.

## 1.1 Shape matching

In order to understand intuitively the “deformation” of shape, imagine you are wearing a shirt. A shirt has in most cases a flat shape. However, as soon as you don the shirt, it undergoes strong deformation. The deformation bends the shirt’s surface so that it aligns with your body. This deformation is exactly the piece of lacking knowledge this thesis tries to eliminate.

There are a lot of applications in computer graphics, computer vision, chemistry and further research fields which utilize shape matching approaches. One such application is the registration of multiple scans. Here, several scans of one object are combined into one large scan covering the entire object’s surface. This is a very common three dimensional acquisition process in order to acquire the full surface of an object.

Figure. 1.2 illustrates the purpose of shape matching for registration. As one can see, the view of the scanner I, covers the object one would like to acquire only



Figure 1.3: Illustration of the ICP algorithm. First, closest points for every point on the source (blue) are identified on the target (red). Next, the estimated transformation is applied and the whole process is repeated if desired.

partially. If one places an additional scanner II the object can be acquired from another side; however, the object can still not be covered fully at once. Hence a technique is required which is capable of reconstructing the whole object after several scans. Approaches from the set of *rigid shape matching* techniques could be used in order to align each part accordingly. Rigid matching is a process of computing the transformation matrix which aligns one piece of geometry to another. Coming back to the example with the shirt, rigid matching just computes the transformation, as rotation or translation, of the shirt to align it to the body. The actual surface deformation is not estimated, hence the shirt will keep its original flat form.

A very common approach to align two pieces of geometry in a rigid fashion is *ICP*, iterative closest point [12]: First for every point on the surface the closest point on the target surface is found. Next the transformation parameters such as rotation and translation are estimated using a cost function, which is based on the distance between the closest points. Finally the transformation is applied and the whole process is repeated until a certain convergence criterion is fulfilled, e.g., the residual transformation becomes negligible. Figure 1.3 illustrates the ICP approach. ICP can only be applied when enough overlay is available in the different pieces of geometry. It is one of the rigid shape matching approaches and is successfully used nowadays even in industrial applications because of its simplicity, robustness and efficiency. One main drawback of ICP is the requirement for an initial alignment. The initial alignment is often achieved by setting initial correspondences manually and (unfortunately) became quasi standard<sup>1</sup> in the industry. “Unfortunately”, because in the registration process there is still a manual intervention step, which prevents fast and highly automated processes to be executed. Although the techniques presented in this thesis are developed for non-rigid de-

<sup>1</sup>This statement has been confirmed to me personally by several companies while I was visiting Euromold’10.

formations, they could also be used for an automatic correspondence estimation making the manual intervention obsolete.

Object detection is another useful application for shape matching. This problem has its roots in the two dimensional optical character recognition (OCR): An image without any semantics is analyzed in order to identify characters or full text phrases [9]. Similar to this, three dimensional shape matching can be used to identify objects in a scan by comparing a scan to a set of predefined shapes from some database and hence providing semantics to the scanned geometry. Indeed, to teach a computer to understand the environment is still an impossible task. One of the important steps in this direction is the ability to determine and to differentiate the objects “seen” by the machine.

In the last years the computer vision community developed a lot of algorithms and methods to learn and to understand the environment seen by a computer. This dissertation contributes to this development with the shown approaches. In particular the thesis concentrates on *deformable shape matching*. The difference to the rigid shape matching approaches is the ability to register two pieces of geometry even if they both undergone non-rigid deformation.

### 1.1.1 Shape definitions

Given this short introduction, let me first provide some definitions of the terms used further in this thesis. These terms build the basis for the algorithms presented in this dissertation and hence they are important for further understanding.

**Manifold:** A manifold  $\mathcal{M} \subset \mathbb{R}^3$  is defined as an embedded 2D surface in 3D space, also known as *2-manifold*. In the following it is assumed that  $\mathcal{M}$  is at least  $C^1$  differentiable in every point.

**Geodesic:** A *geodesic* is a shortest differentiable curve  $\gamma : [0, t_0] \rightarrow \mathcal{M}$  which connects two points  $\gamma(0) = a \in \mathcal{M}$  and  $\gamma(t_0) = b \in \mathcal{M}$  along the shortest path of a manifold. The length of the curve is defined as:

$$L(\gamma) = \int_0^{t_0} \|\dot{\gamma}(t)\| dt,$$

where  $\dot{\gamma}(t) = \frac{\partial \gamma}{\partial t}(t)$ . In other words, the length of a curve on a manifold is defined as an integral of the lengths of infinitesimal *tangents*, i.e., first order derivative vectors of the curve.

**Intrinsic distance and distance metric:** A *distance metric* is a function  $d_{\mathcal{M}} : (\mathcal{M} \times \mathcal{M}) \rightarrow \mathbb{R}$  which defines the distance between any two points on a manifold. In case of three-dimensional *Euclidian space* the distance function is a simple vector norm, i.e.,  $d_{\mathbb{R}^3}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ , where  $\mathbf{x} \in \mathbb{R}^3$  and  $\mathbf{y} \in \mathbb{R}^3$ . The distance is the length of the shortest path connecting two points. The over-the-surface distance, i.e., *intrinsic distance*, in a manifold is defined as an infimum over all possible paths connecting two points on a manifold. Let  $\Omega_{pq}$  be a set of all differentiable curves on a manifold connecting points  $p \in \mathcal{M}$  and  $q \in \mathcal{M}$ . Thus the distance  $d_{\mathcal{M}}(p, q) = \|p - q\|_{\mathcal{M}}$  is defined as:

$$d_{\mathcal{M}}(p, q) = \inf_{\gamma \in \Omega_{pq}} L(\gamma).$$

For two points  $p$  and  $q$  which are not connected by any path in the manifold, the distance is set to  $d_{\mathcal{M}}(p, q) = \infty$ .

**Discrete manifolds** A *discrete manifold*  $M$  is a sampled manifold  $\mathcal{M}$  with a finite  $\epsilon_s$ -resolution. This means that for every point of the original  $\mathcal{M}$ , one point in the discrete manifold exists at geodesic distance of at most  $\epsilon_s$ . The distance metric for a discrete manifold is denoted as  $d_M$ . In order to approximate the distance metric over a discrete manifold a graph  $G = (M, E)$  encodes the manifold surface. An edge  $e \in E$  between points  $p, q \in M$  is included whenever  $p$  is among the  $k$ -nearest neighbors or lies at the Euclidian distance  $l \cdot \epsilon$  of  $q$  or vice versa. This thesis deals with a  $k$ -neighborhood where  $k = 20$ . In the following, a triangle mesh, if exists, is used directly as  $G$ , whereas for point clouds,  $G$  is computed by a  $k$ -neighborhood.

### 1.1.2 Allow deformations

Equipped with these definitions, let me formalize the concept of deformable shape matching. In general, deformable shape matching means, that given two manifolds,  $\mathcal{M}_S$  and  $\mathcal{M}_T$ , one would like to find the deformation function  $f : \mathcal{M}_S \rightarrow \mathcal{M}_T$  which maps every point of  $\mathcal{M}_S$  to the corresponding point of  $\mathcal{M}_T$ . In other words, one would like to find the deformation required to align both pieces of geometry. In general, the deformation function is described by a set of corresponding point pairs

$$\mathcal{C} = \{(x_s, x_t) | x_s \in \mathcal{M}_S, f(x_s) = x_t \in \mathcal{M}_T\}$$



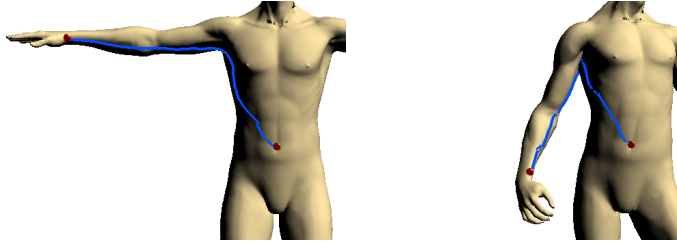


Figure 1.4: Isometric deformations preserve intrinsic distances for every point of the shape. The intrinsic distance between the two marked points is equal.

of both source and target shapes.

A number of variants of this problem exists, such as finding rigid, isometric (i.e., distance preserving), conformal (i.e., angle preserving), or more general mappings between two surfaces. In particular this thesis deals with the *isometric* mapping, which means that the intrinsic distance between any pair of points is preserved regardless of the applied deformation. Formally, given the deformation function  $f : \mathcal{M}_S \rightarrow \mathcal{M}_T$  and a distance metric  $d_{\mathcal{M}_S}$  and  $d_{\mathcal{M}_T}$  of a manifold  $\mathcal{M}_S$  and  $\mathcal{M}_T$  respectively the isometry criterion is defined as:

$$\forall p \in \mathcal{M}_S \text{ and } \forall q \in \mathcal{M}_S : d_{\mathcal{M}_S}(p, q) = d_{\mathcal{M}_T}(f(p), f(q)).$$

In other words, the isometry preserves the intrinsic distances of a shape regardless of its deformation. Figure 1.4 illustrates this situation.

An attentive reader would ask himself: *Why the isometry assumption?* Isometric mapping is a common assumption made in the field of shape matching [6, 22, 55, 84, 113, 114]. Indeed, in the daily life one can see a lot of validation of this criterion. The shirt you wear does not stretch much or do not shear along tangential direction; however, its surface area stays, in most cases, the same. Another example is the skin, which reacts painful if applying strong non-isometric deformation to it<sup>2</sup>. In general, isometric mapping models the deformation in our environment very well.

---

<sup>2</sup>Of course there are skin areas on human's body which react non-isometrically, i.e., face; however due to the limited scanner resolution the non-isometry, in most cases, can be well approximated with isometry criterion

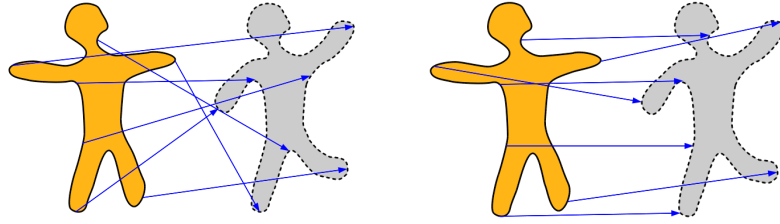


Figure 1.5: Shapes are matched by computing correspondence pairs between source and target shape. Simple descriptor comparison might fail (left) due to strong self-similarity of the object. Hence additional regularization prior is used, i.e., isometry constraint (right).

### Matching

Let me first introduce a concept of *descriptors*. A descriptor is a function or a vector  $\mathbf{D} : \mathcal{M} \rightarrow \mathbb{R}^n$  which locally and in the best case uniquely describes any point of a manifold. Given two descriptors for a point  $p \in \mathcal{M}$  and  $q \in \mathcal{M}$  they are said to be identical with respect to the descriptor if  $\|\mathbf{D}(p) - \mathbf{D}(q)\| \leq \tau_D$ , where  $\tau_D$  is a descriptor similarity threshold. In general, a descriptor can encode any information about the local neighborhood around a point, e.g., differences in multi-scale representations of the data [79], local curvature maps [44] or spin images [60].

Thus, given that concept of descriptors, a naïve shape matching approach could be executed as following: Compute first the descriptor vectors  $\mathbf{D}(p)$  and  $\mathbf{D}(q)$  for every point  $p \in \mathcal{M}_S$  and  $q \in \mathcal{M}_T$ . Then for every two points with a low descriptor distance  $\Delta = \|\mathbf{D}(p) - \mathbf{D}(q)\|$  considers them as matched. If there exists a match between a large set of points, then the full object is registered to the other.

However, this simple matching approach might fail with a high probability, as shown in Figure 1.5. Although simple descriptor matching is capable of finding corresponding points between source and the target shape, aligning the source shape with the target shape based on the matching, as shown in Figure 1.5, results in a strong misalignment between both shapes. Instead an additional term, a *regularization* term, is required, to constrain the solution. In case of algorithms shown in this thesis, the regularization term preserves the isometry constraint.

Given the isometry constraint, a matching function  $f$  is a suitable one if it satisfies the isometry criterion  $d_{\mathcal{M}_S}(p, q) = d_{\mathcal{M}_T}(f(p), f(q))$  for any point pair  $(p, q) \in \mathcal{M}_S \times \mathcal{M}_S$ . Hence a function  $f : \mathcal{M}_S \rightarrow \mathcal{M}_T$  which minimizes the

energy functional:

$$E(f) := \int_{\mathcal{M}_S} \int_{\mathcal{M}_S} |d_{\mathcal{M}_S}(p, q) - d_{\mathcal{M}_T}(f(p), f(q))| \, dpdq, \quad (1.1)$$

has to be found. In case of a discrete representation of the manifolds  $\mathcal{M}_S$  and  $\mathcal{M}_T$  the estimated deformation function  $\hat{f} : M_S \rightarrow M_T$  has to minimize:

$$E(\hat{f}) := \sum_{p, q \in M_S} |d_{M_S}(p, q) - d_{M_T}(\hat{f}(p), \hat{f}(q))|. \quad (1.2)$$

A discrete deformation function  $\hat{f}$ , which is represented by a set of corresponding point pairs of  $M_S$  and  $M_T$  can be computed naïvely by trying out all possible correspondence combinations. This results in exponential cost since each correspondence subset need to be checked. Hence more appropriate algorithms are required for fast and robust correspondence estimation which will be presented in this thesis.

### 1.1.3 Topological noise

One of the big problem which often occurs in the real-world scans are acquisition holes. In general, missing data yields a corrupted surface distance metric  $d$ . The invariant that a geodesic distance is the shortest path between two points became invalid and need to be treated properly. This problem is called *topological noise* and occurs very often in practice when dealing with data coming out of acquisition device such as 3D scanner. The main reason for this type of problem are occlusions and highly reflective surfaces. In figure 1.6(a) one can see that the geodesic distance between two points changes in presence of acquisition holes.

Another type of topological noise occurs when approximating the manifold by a discrete set of points: In the data produced by scanner devices every point is treated independently, i.e., there is no information about the neighbor structure, *topology*, between the points. In this case, one has to approximate the connectivity of the sampled manifold. As stated in Section 1.1.1, the connectivity is approximated by a  $k$ -nearest graph structure. Unfortunately, this introduces two main problems:

First, the discretization disturbs the distance metric and any smooth geodesic is approximated by zig-zag paths, which introduces systematic deviations. Figure 1.6(b) shows a simple example of a manifold (yellow) being discretized by a finite set of points (red). There exists methods to perform a better approximation

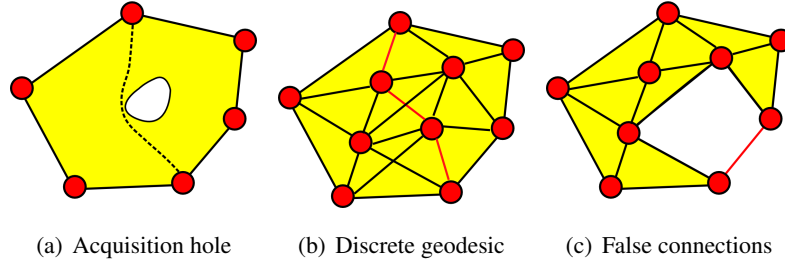


Figure 1.6: (a) The geodesic distance between two points changes dramatically on presence of acquisition holes. (b) Discretization of the manifold disturbs the geodesics. (c) Bad sampling resolution introduces false connections, also known as contacts.

of geodesics on point clouds [82] as well as on triangle meshes [67] rather than a simple Dijkstra [36] shortest path computation. However, these methods often assume certain characteristics of the input data. In order to be more general the Dijkstra shortest-path is preferred in all approaches presented in this dissertation. Additionally, using this representation consistently the systematic errors affect all geodesic paths in the same way so that they remain directly comparable, which is sufficient for the approaches shown in this thesis.

Second, in case of approximating the connectivity with a  $k$ -nearest graph, *false connections* or *contacts*, might be introduced. This effect appears when due to the sampling resolution  $\epsilon_s$  a neighbor point can not be distinguished from a point on a larger intrinsic distance. Formally, two points  $p \in M$  and  $q \in M$  build a false connection if  $d_{\mathbb{R}^3}(p, q) < \epsilon_s$  and  $d_M(p, q) > d_{\mathbb{R}^3}(p, q)$ . Figure 1.6(c) illustrates the effect of false connections.

## 1.2 Deformable matching and inference problems

Part III of this thesis introduces a formulation of deformable shape matching besides the point-to-point correspondence formalization utilizing Markov random fields (MRF). A Markov random field [68], which is very similar to *Bayesian network* as introduced next, is a graph representation of a random experiment. Let me first provide necessary definitions for a proper formulation of the matching problem as MRF.

Let  $X_i, i = 1, \dots, n$  be random variables taking values  $x_i$  in a state space  $\mathcal{X}$ . The

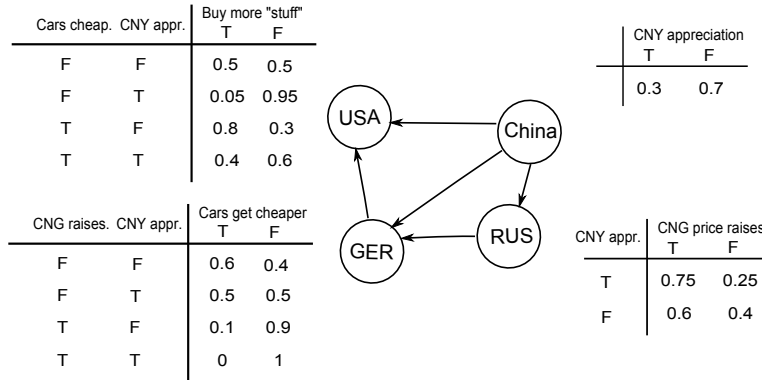


Figure 1.7: Example of a Bayesian network describing the progress of US economy, i.e., the ability of citizens to consume, with respect to China’s currency (CNY) appreciation, CNG (compressed natural gas) price raising by Russia and German’s car export pricing. In general, as one can see US economy suffers if China’s currency is appreciated, since German cars will raise in price too.

conditional dependency between these variables can be modeled by a directed and acyclic (i.e., contains no cycles) graph  $G = (V, E)$ , where  $V$  is a set of nodes (or vertices) and  $E$  is a set of directed edges  $E \subseteq \{(i, j) | i, j \in V, i \neq j\}$ . This graph is called a *Bayesian network* and models the conditional dependency between a set of random variables  $(X_i)_{i \in V}$  which takes a value  $x_i$  from a label set  $\mathcal{X}$ . A contrived illustration, shown in Figure 1.7, describes a dependency graph of a fictive economy<sup>3</sup>.

The joint probability distribution in a Bayesian network is defined as [116]:

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{v \in V} p(X_v = x_v | \text{parents}(X_v)), \quad (1.3)$$

where  $\text{parents}(X_v)$  is a set of the states of the parent nodes of a node  $X_v$ . The joint probability of the model shown in Figure 1.7 is hence computed as:  $P(U, G, C, R) = P(U|G, C)P(G|R, C)P(C)P(R|C)$ , where for the sake of simplicity the events “US citizen consume more”, “CNG price raises”, “China currency appreciates” and “German cars get cheap” are denoted as  $U$ ,  $R$ ,  $C$  and  $G$  respectively. Given this example and the condition probability rule  $P(A|B) = \frac{P(A, B)}{P(B)}$  an answer to the question: “What is the probability that US citizens will

<sup>3</sup>Please note that this network is only for illustration purpose and does neither reflect any actual state of economy of involved countries, nor it models the rather complex global economy dependency.

consume more ( $U = T$ ) if the price for CNG grows ( $R = T$ ) ?” can be computed as following:

$$P(U = T | R = T) = \frac{P(U = T, R = T)}{P(R = T)} = \frac{\sum_{C, G \in \{T, F\}} P(U = T, G, C, R = T)}{\sum_{U, G, C \in \{T, F\}} P(U, G, C, R = T)}.$$

The answer results in  $\sim 36\%$  in this example.

For an undirected graph  $G$ , i.e., edges are not directed, the set of random variables  $(X_i)_{i \in V}$  is said to form a *Markov random field* (MRF) if the following *Markov properties* are hold:

$$P(X = x_1, \dots, X_n = x_n) > 0$$

and

$$P(X_i = x_i | X_j = x_j, i \neq j) = P(X_i = x_i | X_j = x_j, j \in \mathcal{N}(i)),$$

where  $\mathcal{N}(i)$  defines a *neighborhood* of  $i$ , i.e., a set of all adjacent vertices  $\mathcal{N}(i) = \{j | (i, j) \in E, i \neq j\}$ . In other words a full condition distribution of  $X_i$  depends only on its neighborhood.

In Chapter 7 a MRF will be used to formalize the shape matching and in particular deformable shape matching problem. In this case random variables describe points on the shape’s surface. Their state is represented by potential matches on the reference shape. In general, matching problems, such as shape or image matching, are described by a *Hidden Markov Model* (HMM) [110]. This is different to the introduced model: the actual state of the variable is not visible, it is hidden; however, the output of the model is directly observable. Thus every node in such MRF has a probability distribution over the states. In case of shape matching, each point has a probability distribution of being matched to a point on the reference shape.

A computation of the *inference*, or the exact “solution”, which infers the most probable values of the random variables, is NP-hard and hence can not be solved efficiently nowadays. However, there exists approximate solutions such as *loopy belief propagation* [105]. By passing messages that encode beliefs about the marginal distributions of random variables in a graphical model, the algorithm computes a self-consistent solution for the marginal distributions at each node of the graph.

### 1.2.1 Belief Propagation

Assume a MRF consisting of  $n$  random variables  $x_i$ , with  $i = 1..n$  and conditional independence encoded in a graph  $G = (V, E)$  as described before, is given. Each

variable is assumed to have a continuous,  $d$ -dimensional state space  $\Omega = [0, 1]^d \subseteq \mathbb{R}^d$ . The layout of  $G$  can be arbitrary. Furthermore, assume that the probability distribution factors into singleton node potentials  $\Phi_i$  (“data terms”) and pairwise edge potentials  $\Psi_{ij}$  (“compatibility terms”):

$$P(\mathbf{x}_1, \dots, \mathbf{x}_n) = \frac{1}{Z} \prod_{i=1..n} \Phi_i(\mathbf{x}_i) \prod_{i,j \in E} \Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j), \quad (1.4)$$

where  $Z$  is a normalization constant. The goal is to approximate the marginal distributions

$$p(\mathbf{x}_i) = \int \cdots \int_{\substack{\mathbf{x}_1 \cdots \mathbf{x}_n \in \Omega \\ j \neq i}} P(\mathbf{x}_1, \dots, \mathbf{x}_n) d\mathbf{x}_1 \dots d\mathbf{x}_n. \quad (1.5)$$

Loopy belief propagation approximates the marginal distributions  $p(\mathbf{x}_i)$  for each random variable by assigning a message  $m_{ij}$  to each edge in the graph and updates these iteratively by the following rule [156]:

$$m_{ij}(\mathbf{x}_j) = \int_{\mathbf{x}_i \in \Omega} \left( \Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \Phi_i(\mathbf{x}_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{ki}(\mathbf{x}_i) \right) d\mathbf{x}_i \quad (1.6)$$

where  $\mathcal{N}_i$  denotes the set of neighbors of node  $i$  in  $G$ . Intuitively: the message passing encodes the belief of what node  $i$  “thinks” should be the marginal distribution at node  $j$ . The *belief* function  $b_i$  of node  $i$  is given by:

$$b_i(\mathbf{x}_i) = \frac{1}{Z} \Phi_i(\mathbf{x}_i) \prod_{k \in \mathcal{N}_i \setminus j} m_{ki}(\mathbf{x}_i) \quad (1.7)$$

Belief propagation initialize all believes by the singleton potentials  $\Phi_i$  and then iteratively passes messages in the graph until convergence. The final approximations to the marginals  $p(\mathbf{x}_i)$  are the beliefs at each node. Figure 1.8 illustrates the message passing in a graph.

One of the big disadvantages of loopy belief propagation is its relatively high running time complexity, which is, in general, in the order of  $\Theta(nk^2)$  for a MRF of 2-clique size as for example used in this thesis. Here,  $n$  is the number of nodes and  $k$  the number of possible states, i.e.,  $k = |\Omega|$ .

One of the main contributions in this thesis is a method of reducing the space-time complexity of loopy belief propagation in a given MRF model. The complexity reduction, shown in Chapter 8, reduces the approximate inference computation from quadratic to sub-linear costs. One can also show easily that in worst case

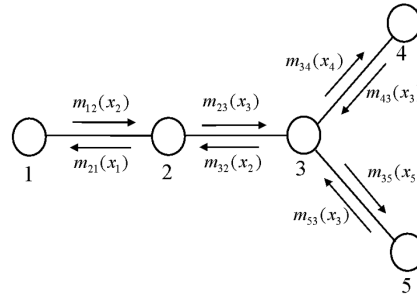


Figure 1.8: An example of the message passing algorithm [156], i.e., belief propagation. Messages representing the *belief* of a node which state its neighbor should have are passed along the graph.[Image taken from [156]]

the complexity of optimized belief propagation is equal to the computation costs without any improvements.

MRFs are used in different fields beyond computer vision, e.g., computational biology [39] to analyze the sequence of genes or speech recognition [110] to match an acoustic spectrum of a spoken word to a certain dictionary. These and many other research fields might benefit of the inference computation presented in this thesis.

### 1.3 Applications for deformation matching

One of the important applications of deformable shape matching is in improving the registration of 3D scans [26]: Even expensive 3D scanners suffer from minor calibration problems that lead to alignment problems if large models are acquired at a high spatial resolution. This can be compensated by allowing for small global deformations. An application area that requires handling much larger deformations is the acquisition of deformable objects. For example, scanning of humans, animals or other living beings usually requires multiple scanning passes from different perspectives during which the scanned subject will inevitably move in a non-rigid fashion. Hence assembling complete, high-resolution scans requires deformable shape matching. Of course one could argue that scanning by several scanner devices in parallel makes this issue obsolete. Besides the fact that multiple devices might be expensive a software solution is preferable. Additionally to achieve good scanning quality, either the scanning device need to sweep very slowly over the object (because of poor lightning conditions often existing during



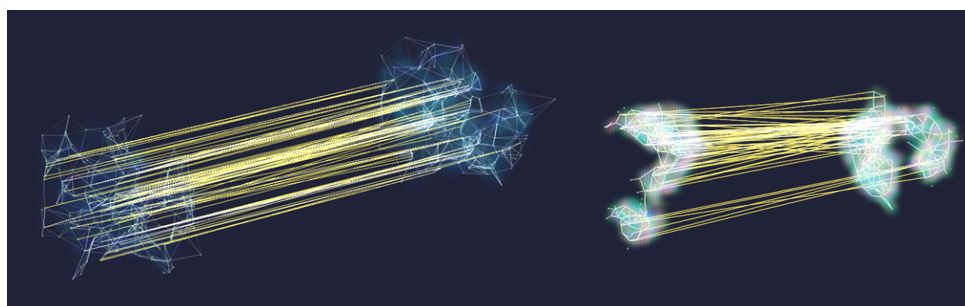


Figure 1.9: Matching of a protein experiencing folding. The correspondence estimation and hence the deformation can be used by biologists to derive additional knowledge of the reaction of proteins to different types of drugs. [Courtesy of Michael Wand]

the scanning process) or multiple scans are required in order to statistically cancel out the noise. Hence deformable shape matching could help even in the case when multiple scanner devices are available.

Although the thesis deals with geometrical structures, the solutions derived here could also be applied to usual graph matching problems. Biology is one of the further fields which could benefit from the outcomes of this work. For example a folding process of a protein can be analyzed by establishing correspondences between different temporal samples as shown in Figure 1.9. Given the ability to exactly match the protein structure over multiple time samples one can observe the folding process and hence analyze the reaction of a protein to different drugs or other external occasions.

Given the possibility of computing the deformation function for a pair of geometry pieces, it is very obvious to ask if the same technique can be applied for multiple geometry pieces at once. This already shows another application area for deformable shape matching: animation scanning and reconstruction [87, 123, 137, 153, 159]. The goal here is to capture the dynamics of a moving object in real-time. It is necessary to construct a composite object from several, probably strongly deformed poses with significant deformation. The available data in each frame of an acquired animation is usually incomplete, as typically being measured from a sparse set of views only. Therefore, the matching algorithm needs to be able to deal with partial data, as well as to be able to match portions of the surface with several large holes.

The main application field for this type of algorithms is in the film production industry. Different actions or facial expressions performed by an actor could be recorded and reconstructed applying the techniques presented in this dissertation. The reconstruction provides the artists with the actual deformation of the actor's geometry over time. This deformation can then be applied to different geometry, for example a three-dimensional representation of another actor or just any other alien-like shape.

Similarly, this also applies to reconstructing a free-viewpoint video for 3DTV. For example a soccer game or actors performing an action are captured by multiple video cameras. Using multi-stereo reconstruction approaches combined with multi-frame matching one can reconstruct the full motion of the performing entities from the data stream. The reconstruction could then be used in order to take a look at the scene from a new point of view or to apply the reconstructed motion to another virtual character. A soccer game captured with multiple cameras and translated to full 3D can for example be viewed by the viewers from any virtual point of view.

### **1.4 Thesis outline**

The thesis is grouped into two main contributions. In the first part the reader will take a look at a possible solution for the deformable shape matching problem. A solution, which is able to handle data with acquisition holes and non-rigid deformation will be shown in Chapter 3. Additionally an iterative optimization approach allowing for faster correspondence estimation compared to the previous work will be shown in Chapter 4. And finally all the ideas shown so far are combined to provide the reader with a method capable of estimating the deformation and the geometry from a sequence of strongly deformed point clouds acquired with a real-time 3D scanner, see Chapter 5.

In Chapter 7 of the thesis a small side step will be made and an alternative approach for deformation function estimation will be presented. The problem of shape matching will be defined as a Markov random field and belief propagation will be employed for the inference estimation of the MRF. And finally, in Chapter 8, a method of improving the space and time complexity of belief propagation will be shown. The improvement will make the inference estimation of the MRF with belief propagation practically resolution independent. The content of the dissertation will then be summarized in the last part of the thesis, part IV.

## 1.5 List of Publications

The following list contains all the work made public during my PhD study. The publications are separated into two list. The first list contains publications which are the basis of this dissertation and have been published in some leading venues in computer vision and computer graphics. The second list contains additional publications which have been done in the field of computer graphics, and in particular rendering, and hence represents another scope of the work that has been done during the PhD study. The main text passages of the papers from the first list have been utilized in this thesis without being tagged individually. The corresponding coauthors have permitted the use of these text passages as well as figures from the originally published papers.

A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, H.-P. Seidel: “Animation Cartography - Intrinsic Reconstruction of Shape and Motion”, *ACM Transaction on Graphics TOG’12*, 2012, to appear [139].

A. Tevs, A. Berner, M. Wand, I. Ihrke, H.-P. Seidel: “Intrinsic Shape Matching by Planned Landmark Sampling”, *Proc. of Eurographics’11*, 2011, pages 543-552, [140].

R. Lasowski, A. Tevs, M. Wand, H.-P. Seidel: “Wavelet Belief Propagation for Large Scale Inference Problems”, *Proc. of CVPR’11*, 2011, pages 1921-1928, [72].

R. Lasowski, A. Tevs, H.-P. Seidel, M. Wand: “A Probabilistic Framework for Partial Intrinsic Symmetries in Geometric Data”, *Proc. of ICCV’09*, 2009, pages 963-970, [71].

A. Tevs, M. Bokeloh, M. Wand, A. Schilling, H.-P. Seidel: “Isometric Registration of Ambiguous and Partial Data”, *Proc. of CVPR’09*, 2009, pages 1185-1192, [141].

### Additional Publications

A. Tevs, M. Wand, I. Ihrke and H.-P. Seidel: “A Bayesian Approach to Manifold Topology Reconstruction”, *Research Report, Max Planck Insitutue Informatik (January 2010)*, pages 1-23, [142].

J. Kerber, A. Tevs, A. Belyaev, R. Zayer, H.-P. Seidel “Real-time Generation of Digital Bas-Reliefs” ,*Computer-Aided Design and Applications (Special Issue: CAD in the Arts), 2010*, pages 465-478

J. Kerber, A. Tevs, R. Zayer, A. Belyaev, H.-P. Seidel: “Feature Sensitive Bas

Relief Generation” , *IEEE International Conference on Shape Modeling and Applications Proceedings (SMI'09)*, 2009, pages 148-154.

A. Tevs, I. Ihrke, H.-P. Seidel, “Maximum Mipmaps for Fast, Accurate, and Scalable Dynamic Height Field Rendering”, *Proceedings of the 2008 symposium on Interactive 3D graphics and games, I3D'08*, 2008, pages 183-190.

B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, H.-P. Seidel: “Time-resolved 3D Capture of Non-stationary Gas Flows”, *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)*, 2008, pages 132-140.

I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, H.-P. Seidel: “Eikonal Rendering: Efficient Light Transport in Refractive Objects”, *Siggraph'07: ACM SIGGRAPH 2007 papers*, 2007.

G. Ziegler, A. Tevs, C. Theobalt, H.-P. Seidel: “On-the-fly Point Clouds through Histogram Pyramids”, *11th International Fall Workshop on Vision, Modeling and Visualization (VMV'06)*, pages 137-144, 2006.

T. Stich, A. Tevs, M. Magnor: “Global Depth from Epipolar Volumes - A General Framework for Reconstructing Non-Lambertian Surfaces”, *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 916-923, 2006.

---

# 2

## Related Work

---

In the following research work related to the approaches presented in this thesis will be shown. It will be characterized and classified into several groups and will be explained how it relates to this work.

### **Shape matching:**

In general, deformable shape matching can be classified in local and global matching strategies. The algorithms presented in this thesis fall into the category of global registration methods, however for the sake of completeness local registration methods are also presented. Most of the local registration techniques are successors of the ICP (iterative closest point) [12, 88, 115] algorithm. ICP is based on a simple iterative optimization technique which is guaranteed to converge to a local minimum of some distance metric. In case of shape matching it is used in order to find the six-degree-of-freedom, i.e. rigid transformation to align one shape to each other, see Section 1.1.

There also exists deformable variants of ICP [5, 77, 117, 153], where point-to-surface distances are minimized under the regularizing assumption of elasticity. Another class of local matching techniques are piecewise ICP [26, 57]. Hereby it is assumed that the range scan is a set of several piecewise rigid parts. In general, the number of parts is small and hence not sufficient for general deformable motion. However, sparse feature points are still used for initial registration to find rough correspondences.

Sagawa et al. [117] propose a method for non-rigid registration using dense local features within the ICP framework. They define a window over every vertex of the surface and compute a convolution of an error metric in this window which has to be minimized by the particular vertex transformation. In general, they are trying to optimize the transformation of ICP with a smoothness constraint

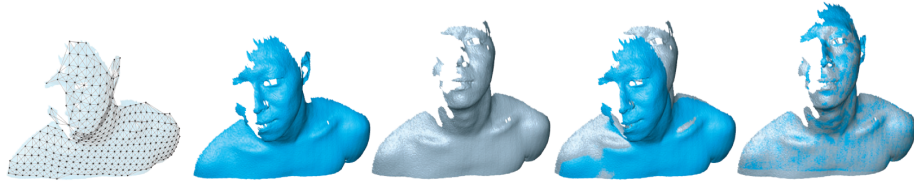


Figure 2.1: Results of the deformation ICP approach of [77]. From left to right: deformation graph, source, target, initial alignment and the final deformation. The initial alignment corresponds to the result achievable with rigid ICP. The final deformation maximizes the region of overlap and the spatial coherency of deformation while minimizing the registration error. [Image taken from [77]]

over the neighborhood of a vertex. Other deformable matching techniques include optimization with deterministic annealing [32] and Chang et al.’s technique [30] based on graph cuts. Chang and Zwicker [30, 31] provide global and robust local matching strategies for articulated, piecewise rigid models.

Matching with sparse features is another widely used concept for shape registration. Between two sets of sparse points distributed over the source and target surface correspondences are computed. Gelfand et al. [46] computes a set of unique descriptors on source and target shape and using a branch-and-bound algorithm pick an optimal set of correspondence between the descriptors to drive a consecutive ICP registration. Memoli and Sapiro [83] compare two manifolds represented by point clouds using an iterative *Farthest Point Sampling* (FPS) [91] algorithm which computes an optimized covering by minimizing an approximate Gromov-Hausdorff-Distance between source and target. Sagawa et al. [118] propose a method of matching sets of feature points based on a color histogram describing the textural neighborhood around the feature point.

Sahillioğlu and Yemez [119] search for correspondences between two shapes by detecting coarse correspondence of local patches. Finally they optimize the mapping in two steps: first transforming the mapping into the spectral domain and minimizing the error by bipartite graph matching and second, an iterative greedy algorithm in Euclidean space is used to minimize the isometric error.

Bronstein et al. [22] examine the problem of embedding surfaces into each other isometrically using a numerical optimization scheme (generalized multi-dimensional scaling). Li et al. [77] propose a semi-global matching technique that is more robust in convergence by numerically optimizing matching weights. The

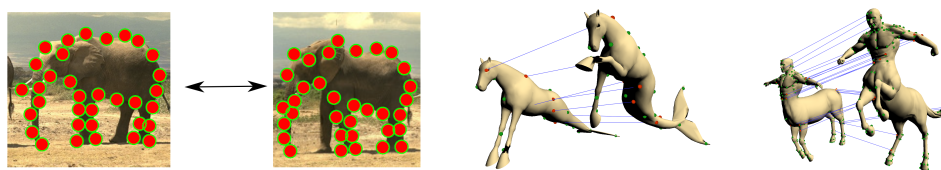


Figure 2.2: (*Left*): Matching results between two different images of an elephant achieved with the spectral matching algorithm [74], original image take from the paper. (*Middle*), (*Right*): Spectral matching algorithm applied on a 3D shape of seahorse and centaur data set [23].

technique of Li et al. is based on local matching but increases its robustness by modeling correspondences explicitly as latent variables and optimizing over them.

A simple and very effective global matching algorithm for pairwise constraints has been proposed by Leordeanu and Hebert [74]. It relaxes the quadratic assignment of pairwise matches to an eigenvalue problem. This algorithm can be applied to deformable matching using feature detection to define keypoints and the preservation of geodesic distances between pairs of features as pairwise validation criterion. The approach is used for example in [2, 16, 55] for global shape matching and is currently probably because of its simplicity the most frequently used state-of-the-art technique. The main drawback of this algorithm is that the pairwise validation criterion (sets of geodesic distances) have to be specified upfront. Therefore, a small number of topological problems that reroute some of the geodesics can drastically impact the matching results. In Chapter 3 the method of Leordeanu and Hebert will be compared to our method on real-world data which suffers from topological noise.

Huang et al. [55] examine the propagating of dense correspondences by geodesic landmark coordinates, and Ahmed et al. [2] employ Laplacian diffusion of the landmarks for dense correspondence estimation. Alternative formulations that have been proposed include optical-flow-like correspondence propagation [87, 147] for densely sampled time sequences and for sequence merging in [146], where Varanasi and colleagues compute a dense deformation field by a Laplacian diffusion from a sparse set of matched points.

These two papers [55] and [2] introduce landmark coordinates for deriving dense matches from the coarse matches returned by spectral matching [74]. Tung and Matsuyama [144] as well as Zhang et al. [158] use a set of automatically se-

lected landmarks which are coarsely sampled to estimate dense matches. Their method of detecting landmarks is based on extremal points of a geodesic integral function [51]. Landmarks are placed on such extrema to decrease the error introduced by approximate geodesic distances. In typical situations landmarks are placed on the shape's extremities. In general, however, these points are not sufficient to uniquely determine the mapping. Hence in case of [144] the authors propose to increase the subset of landmarks slightly by randomly sampling additional points on the surface. In general, this can not guarantee a sufficient subset of landmarks.

A related proposal has been made by Bronstein et al. [24], who trade-off Euclidean and intrinsic distances for validating matches. They show significant improvements over intrinsic-only matching criteria but the technique is not able to handle general cases of strong deformations, where Euclidean matching becomes very unreliable. Collaborating with Mahmoudi and Sapiro same authors [25] propose diffusion distances, which are more robust to a certain amount of topological noise, as this distance measure is sensitive to the cross section of interconnections rather than just the reachability in the case of geodesic distances. However, large scale artifacts such as big acquisition holes or false connections in a large area also change diffusion distances significantly. Unfortunately, these problems are common in the application area of this thesis dealing with real-world data.

Ruggeri et al. [113] compute a set of anchor-points of a shape by thresholding critical points of the Laplace-Beltrami [112] operator. These anchor points are located on geometrically meaningful regions of the shape and are invariant with respect to isometrics. Additionally, they sample reference points by farthest point sampling [40]. Although the idea of using the Laplace-Beltrami operator for placing anchor points is different to curvature-based approaches, i.e., feature detection, the authors are still comparing two shapes by the intrinsic distances between these reference points.

Bradley et al. [19] propose a technique specifically designed for garment capture that uses specific properties of such data sets to control the boundary conditions of a cross parameterization algorithm, thus establishing valid correspondences.

Kim et al. [66] show an algorithm capable of computing dense correspondences between two non-isometrically deformed shapes. Their method computes an optimal blend between several intrinsic maps such as Heat-Kernel maps [10, 62], con-



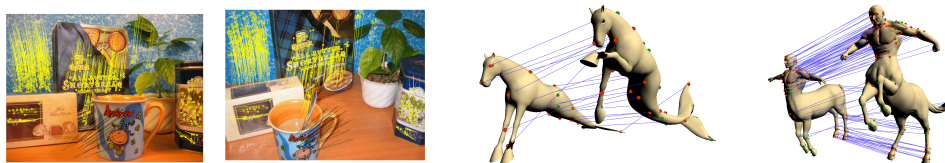


Figure 2.3: (Left): Two images matched with PROSAC [33], original image from the paper. (Middle), (Right): Seahorse and centaur 3D datasets [23] matched with RANSAC-like method as shown in Chapter 3.

formal maps [65, 80] etc., hence allowing to find correspondences even between two shapes that underwent strong non-isometric deformations.

And very recently, Sharma et al. [125] present a method of computing dense correspondences. The authors assume that a set of sparse correspondences is already given. This can either be computed by spectral matching or with approaches as presented in this thesis. Given sparse correspondences, a heat kernel is evaluated on already matched points providing information over the matching on a small, local space around the sparse correspondences. The information from the heat-kernel is then used in order to iteratively add new, dense correspondences by performing a seed-growing which propagates the seed correspondences to nearby vertices.

Kaick et al. [145] provides a good overview over the state-of-the-art shape matching techniques.

### **Robustness by randomness:**

The techniques presented in this thesis are based on randomized correspondence estimation. Hence I would like to provide an overview over the work done in this area.

Fischler and Bolles [43] describe a *hypothesis-and-check* concept called RANSAC (random sample consensus) which robustly fits a model to observed data. Several correspondence matching algorithms based on RANSAC have since been published in the literature. Chum and Matas [33] show a progressive random consensus, PROSAC, in order to find correspondences between two images (which is very similar to shape matching). The idea is to use linear order on the correspondence set defined by a similarity function of correspondences. The solution samples are drawn from a top-ranked set of correspondences. The convergence time improves over RANSAC, since the probability of choosing bad correspondences

is low. Torr and Zisserman [143] introduced MLESAC, which is very similar to RANSAC. A solution which maximizes a likelihood rather than the number of inliers is computed. The scoring function used by the authors is a truncated estimator which penalizes outliers in the same way as RANSAC, but scores inliers according to how well they fit the data.

A technique from the xSAC family presented in Chapter 3 will be applied to solve the intrinsic shape matching problem. It is motivated by the forward search algorithm of Huang et al. for rigid shape matching [56]. The technique is also related to the randomized matching algorithm by Memoli et al. [83]. In general: first an initial set of correspondences is randomly sampled. Starting at random source points, corresponding target points are chosen with probability proportional to the likelihood that the match is correct. Then additional correspondences are added if they do not violate the isometric matching criterion.

### **Minimal sets of correspondences:**

A related group of research handles a more theoretical view on the shape matching problem. Here, the authors are interested in an optimal, minimal set of correspondences to fully constrain the matching results.

Mémoli and Sapiro [84] observe from a theoretical point of view how to compare isometrically two manifolds represented by point clouds without any surface reconstruction. They use the concept of *Gromov-Hausdorff distance* [48] based on geodesic distances. The authors assume that a minimal subset of corresponding points is known.

Lipman and Funkhouser [80] observe that isometry is a subset of a *Möbius group* which has a low dimensionality: For isometrically deformed shapes with spherical topology only three point-to-point correspondences are required to fully determine the matching. Although this observation made a large impact on the understanding of the complexity of the intrinsic matching problem, it is unclear how many degrees of freedom are present under noise, imperfect isometries, or shapes of general topology. For this reason, they use a voting scheme to accumulate evidence from multiple solutions.

Ovsjanikov et al. [96] show that a single corresponding point between source and target shape is sufficient if heat-kernel signatures are taken into account for determining the matching. Again, this result holds in the numerically exact case and voting among several matches is necessary for stable results with practical

data. Their work depends on an efficient compute-and-compare method of heat kernels for manifolds [10] and is based on the results of Jones et al. [63] which describe parametrization of manifolds via heat kernels and eigenfunctions of the Laplacian.

Given this line of work a question arises: Is it possible to compute a minimal set of isometry invariant correspondences between two shapes?

A possible answer to this question is presented in Chapter 4. Here the idea of shape matching introduced in Chapter 3 is extended by not just performing importance sampling on the target point of the match but also carefully planning for which source points matches should be guessed. As will be shown in the experiments, this usually leads to a reduction in the number of random guesses that are necessary to find the correct solution as well as to an improvement of the quality of a match.

The idea is based on the work of Schmid et al. [121] who introduce entropy for feature detection. They compare different feature point detectors by their *distinctiveness*. Entropy has since then been used frequently in follow up work to estimate feature saliency. The approach shown in this thesis is different from this line of work in that it not only looks at the entropy of the feature descriptors but on the remaining entropy of the full matching problem.

Related to this, Bronstein et al. [20] use a statistical measurement of the frequency-inverse document frequency known from the field of document retrieval in the area of shape matching. They measure the statistical significance of a feature point by checking how often it can be found in other shapes. Unique features describe a shape better than those found often in other shapes. The shape matching approach shown in Chapter 4 is different in that it examines features within one shape. Again, the most important difference is that the approach shown in this thesis captures the interdependence of many interrelated feature matches rather than just the statistics of a single match.

### **Animation reconstruction:**

Given a small overview over the work done in the area of matching a pair of shapes, I would like to provide an overview over research articles published about matching a whole set of shapes at once, also known as *Animation reconstruction*. Animation reconstruction is the process of recovering the motion of a deformable object from time-varying three-dimensional scanner data, typically point clouds. There exist

multiple three-dimensional scanning techniques for acquiring moving objects in real-time [18, 34, 70, 151, 154, 155, 159, 162]. The output of such an acquisition process is a time-sequence of unstructured point clouds. The measurement process does not provide any correspondence information and usually only shows a limited part of the object at a time, due to occlusions.

The reconstruction of many-frame correspondences is a non-trivial generalization: Just repeatedly performing pairwise matches, as for example shown in Chapter 3, exponentially increases the failure probability of randomized matching. In Chapter 5 this pitfall is avoided by on the one hand using a continuous tracking algorithm and, on the other hand, by explicitly assessing the matching quality, avoiding the incorporation of ambiguous information in the result.

There are number of previous methods that require the user to provide a template model that is subsequently deformed in order to match the acquired data [7, 18, 29, 35, 75, 102, 120, 159].

Several algorithms tries to solve the deformable matching problem in a time-sequence manner, where only temporally adjacent frames are combined under the assumption of spatial coherence, using deformable variants of the ICP algorithm, such as [5, 49, 137, 153]. These techniques yield good results for small inter-frame deformations. However, they are rather unstable under fast movements of the scanned object that lead to substantial differences in pose. Matching objects with strong deformations is easy if a set of guiding markers is available that specify the rough pose of the object. Using this information (often provided by manual labeling [4, 5, 103]), the shapes are roughly prealigned and afterwards a fine-scale alignment is performed by deformable ICP. A number of fully automatic techniques have been proposed for *global*, pose independent deformable matching by computing such marker sets automatically [2, 6, 22, 30, 55, 76, 80, 113, 114, 127, 135, 139].

Mitra et al. [87] perform rigid alignment between frames, assuming rather slow motion with little local deformation. The technique is elegant and very fast but cannot handle general sequences with missing data and substantial inter-frame deformation. Wand et al. [153] use deformable matching and a statistically motivated optimization scheme. The considerable computation costs have been addressed in [152], Figure 2.4, by employing a subspace deformation technique. The technique is able to compute complete template models from partial input data but, as a local optimization technique, it is sensitive to issues such as large time steps, temporarily disappearing objects, and fragmented frames. Popa et al. [108] propose

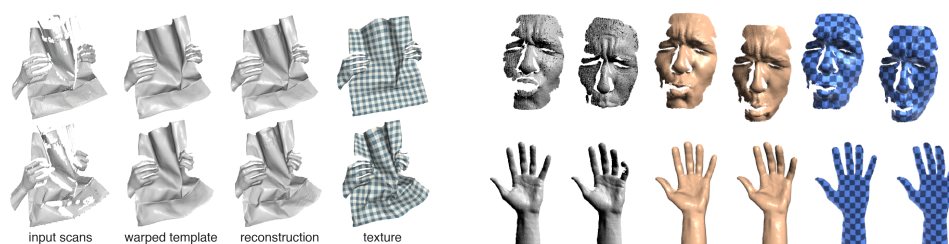


Figure 2.4: (*Left*): Template based animation reconstruction, see [75]. (*Right*): Template-free reconstruction of the shape and its deformation as shown in [152] (here images show input scans, reconstruction and applied texture on the reconstruction to indicate correspondences). [Images were taken from [75] and [152]]

an improved template-free reconstruction method based on optical flow and cross-parametrization. However, their technique cannot handle fast motion and requires additional video input for 2D feature tracking (such as provided by passive stereo acquisition systems).

Li et al. [75], Figure 2.4, use a more efficient subspace deformation technique in combination with detail transfer, which was previously examined by Bickel et al. [13] for the case of wrinkles, to obtain very good results, however, requiring a template model as input. A combination of deformable matching with Mitra et al.’s algorithm is examined in Süßmuth et al. [138]. Their work relies on having a complete shape in the first frame, again not improving on the issue of assembling the template model from partial data.

Comparable approaches have also been examined with different regularizing assumptions: Pekelny and Gotsman [107] use an articulated piecewise rigid model, segmented by the user, Sharf et al. [124] examine volume and momentum preservation as an alternative. Both are still local optimization techniques, subject to the according limitations.

A recent approach by Zheng et al. [161] aims at reconstructing the temporal correspondences of a skeleton rather than complete geometry, which can then subsequently be used as guidance information for shape alignment. The drawback is that although skeletonization improves robustness, it does not represent the full correspondence information, which cannot always be recovered reliably.

Very recently, Huang et al. [54] propose a method of reconstructing the global movement of a shape. This is performed by introducing a so called alignment graph which adds an edge to the graph if two frames of different animation sequences are

similar. The similarity is measured by computing a per-frame descriptor and comparing two frames by this descriptor. A shortest path through this alignment graph describes the global motion and alignment of every frame. The method makes strong use of deformable shape matching in order to align two frames together. Unfortunately manual user intervention is still required in order to define an optimal reference frame containing fewest topological problems. This reference frame is used in the alignment graph as a starting node to reconstruct the complete animation. The approach presented in this thesis 5 is capable of automatically generating the reference frame out of the given animation without any specific user interaction.

### **Symmetry detection by shape matching:**

In general, the problem of shape matching is strongly related to the detection of similar parts in a shape. In Chapter 7 the problem of shape matching will be defined as a Markov random field and applied to the problem of symmetry detection. Here the work is inspired by Angelov et al. [6], who formulate the deformable shape matching as a Markov random field using the geodesic distance as a pairwise consistency condition. However, their technique aims at computing a single global maximum a posteriori solution, which actually leads to ambiguity problems for matching approximately symmetric shapes, thus requiring multiple random initializations of the inference algorithm. A similar approach is employed by [127] for deformable shape matching with resilience to topological noise.

Symmetry detection using extrinsic information has been addressed by several authors. The most successful class of techniques is voting for transformations in a Hough space based on candidate correspondence pairs [89, 90, 104]. In the context of rotation and reflection symmetry detection in 2D images, the work of Park et al [101] presents an evaluation of the state-of-the-art algorithms. An algorithm for deformed lattice detection for 2D images is proposed in [100]. Extrinsic symmetry detection is taking into account extrinsic information like for example point-to-point arrangement within an object. In contrast, intrinsic symmetry detection is based on intrinsic information only, hence working with manifold properties directly. This has the strong advantage of being independent of deformation which preserve intrinsic properties.

The detection of intrinsic symmetries has received only little attention up to now: [97] present an elegant algorithm that uses eigenvectors of the Laplace-Beltrami operator on manifolds that are invariant under isometric deformations.

However, their technique is restricted to global (rather than partial) symmetries due to the global nature of their spectral approach. Intrinsic symmetry detection is also addressed by Raviv et al. [111]. The authors propose an algorithm to compute an isometric correspondence map of the shape onto itself to extract a set of symmetries using the computational framework for isometry-invariant comparison presented in [21]. Again, only global symmetries are considered. More recently Kim et al. [65] propose a method of detecting symmetric parts of an object by examining the group of Möbius transformations found in the shape, which are described in the work of Lippman and Funkhouser [80].

### **Efficient belief propagation:**

Using belief propagation for computing inference in a MRF one easily runs against space-time-complexity of this approach. As we will see in Chapter 7 belief propagation can be used to solve shape matching problem; however, its computation performance is not optimal. Belief propagation was first introduced by Pearl in 1982 [105] for tree-formed graph structures and was later shown to be a useful approximate algorithm for general graphs [106], see also Figure 1.8 for a small illustration. Since then, belief propagation has gained a lot of interest in the computer vision community. One of the main application areas for belief propagation is stereo- or multistereo-reconstruction as for example shown in [129].

Belief propagation, however, is currently pretty much infeasible for large state data [156]. There has been an effort by the research community to improve on that, e.g. by using compression or space transformations: Yu et al. [157] investigate PCA and an envelope transform for the compression in a 1D domain. The technique achieves good results but cannot be applied directly in the compression domain. Zhao et al. [160] compress the messages in the wavelet domain however each time the message is uncompressed when arriving at the node and compressed when going out. This is effective in a sensor network setting but does not speed up a non-distributed algorithm.

Minka [86] describe expectation propagation, a variant of LBP for potential functions belonging to exponential families allowing for continuous state spaces in this case. Felzenszwalb et al. [42] describe linear and near-linear time message passing algorithms for restricted types of pairwise potentials. Additionally the authors show a speed up of message update computation by FFT-based fast convolution for shift invariant potentials. Komodakis et al. [69] propose dynamic

pruning of states of low probability. However, this approach cannot handle smeared out uncertainties, but would distort such distributions by removing the uncertainty information altogether. Potetz et al. [109] approximate the solution of the MRF using a factorization approach to efficiently handle higher order potentials ( $c > 2$ ).

Sudderth et al. [133] introduce *nonparametric belief propagation* (NBP) that represents messages by samples. Fitting of Gaussian mixture models and Gibbs sampling is used for message propagation. This reconstruction from sampling leads to numerical smoothing, adding artificial uncertainty in each propagation step. For large graphs, this impacts precision. A restricted NBP for Gaussian potential functions is presented in [58]. Han et al. [50] employ mean-shift to estimate the mode and weight for the outgoing message, but this neglects uncertainty information. Following this line, Parks et. al [99] discretize the continuous state space into a grid and use only local samples to perform mean-shift on the marginals.

In Chapter 8 an improvement of the space-time complexity of belief propagation by moving all the computation to the wavelet domain will be presented. The wavelet representation is more restrictive with respect to the domain of the *state space* (restriction to hypercubes in low dimensions). Good results are obtained even for large graphical models with many random variables and very high PSNR as a reader will be able to investigate. The presented approach is aiming at representing the full probability distribution, not just the main modes as it is done with NBP.



**Part II**

**Shape Matching**



# SHAPE MATCHING ---



In this part of the thesis we will present two approaches for deformable shape matching (Chapter 3 and 4) and their application in multi-frame matching, Chapter 5. Both approaches compute correspondences for shapes that undergone (approximately) isometric deformations.

In the first chapter of this part we propose a solution for solving the isometric matching problem that makes two main contributions: First, the algorithm is robust to topological noise such as large holes or false connections, which are both observed frequently in real-world scanner data. Second, the algorithm samples the space of feasible solutions such that uncertainty in matching can be detected explicitly. A novel randomized feature matching algorithm is employed in order to find robust subsets of geodesics to verify isometric consistency.

In the second approach we look at a shape matching algorithm that uses an a-priori planning step to compute a well-distributed set of landmark points. These points are matched first in order to maximize the information gained and thus minimize the randomized sampling costs. We incorporate techniques we developed in the first approach in order to find correspondences between these landmark points. Here we make three main contributions: First, the new technique leads to a significant improvement in performance, which we demonstrate on a number of bench-

---

mark scenarios. Second, the technique does not require any feature point detection. This is often a significant limitation for models that do not show sufficient surface features. Third, we examine the actual numerical degrees of freedom of the matching problem for a given piece of geometry. Similar to the first approach our estimates take into account unprecise geodesics and potentially numerically unfavorable geometry of general topology, giving a more realistic complexity estimate.

In the last chapter of this part we incorporate the previous two approaches into a global technique in order to reconstruct the shape and motion of a deformable object from dynamic 3D scanner data, without using user provided template models. We present a system that can handle fast motion, temporally disrupted input, and can correctly match objects that disappear for extended time periods in acquisition holes due to occlusion. Our approach is motivated by cartography: We first estimate a few landmark correspondences, which are extended to a dense matching and then used to reconstruct geometry and motion. We propose a number of algorithmic building blocks: a scheme for tracking landmarks in temporally coherent and incoherent data, an algorithm for robust estimation of dense correspondences under topological noise, and the integration of local matching techniques to refine the result. We describe and evaluate the individual components and propose a complete animation reconstruction pipeline based on these ideas. We evaluate our method on a number of standard benchmark data sets and show that we can obtain correct reconstructions in situations where other techniques fail completely or require additional user guidance such as a template model.

## **Introduction and Overview**

The most common approach to compute correspondences between two shapes is to first compute a set of discriminative feature points on both shapes along with a local, isometry invariant descriptor and then try to find a matching of these features such that the pairwise geodesic distances between all corresponding pairs of feature points are preserved. Posed naively, this leads to a NP-hard quadratic assignment problem for which no sub-exponential solution strategy is known.

In this part of the thesis we utilize a randomized approach capable of determining the correspondences more efficiently than a naive approach. In particular we would like to find a suitable set of markers that could be used to guide a locally convergent deformable ICP algorithm. This is done fully automatically, without user

---

intervention and without assumptions on pose or deformation. Our only requirement is that the unknown deformation is approximately isometric, i.e., roughly preserves distances on the surface. One of the main advantages of our work is that the registration also works under topological noise, e.g., holes, badly disconnected parts, etc.

In Chapter 3, referring to the paper [141], we present an algorithm which computes the matching between automatically determined marker points, i.e. “feature” points by a randomized sampling strategy RANSAC. Here we make three main contributions:

- We propose a randomized, forward-search matching strategy that, unlike previous techniques, simultaneously estimates the correspondences and validates geodesics in an outlier-robust way.
- We employ a new tangent-space optimization algorithm to optimize the placement of feature points for maximum isometric matching, yielding better results for noisy feature positions.
- We sample the space of plausible matches, which gives us the ability to explicitly examine matching alternatives. We consider this an important building block for fully automatic matching of complex models from several pieces.

We apply our matching algorithm to a number of data sets and show reliable matching results for general pose and topologically distorted data with holes and false connections.

In the second chapter of this part, Chapter 4, referring to the paper [140], we improve the work by examining the problem of isometric matching from more generic point of view. Here we address the following two questions:

- What is the *practical complexity* of matching two isometrically deformed shapes, assuming a certain amount of uncertainty in intrinsic distance estimates?
- Can we find an algorithm that systematically collects the most relevant correspondence information to find a global shape match as quickly as possible?

To address these problems we are proposing a shape matching algorithm to which we refer as *planned* random sampling (PLANSAC): This means that instead

---

of using a randomized sampling of correspondences as demonstrated in Chapter 3, we are building a *plan* on how to choose optimized sample points, i.e. landmarks, on the source shape that maximize the information gained and therefore minimize the costs for guessing correspondences. Our key idea is to look at the entropy of the posterior distribution of possible matches to assess what are good plan points.

Finally, in Chapter 5, referring to [139], we will investigate a solution for solving the deformable matching problem for a complete time-sequence. We will utilize the concept of randomized correspondence estimation and the landmark concept in order to compute the actual shape and motion of a deformable object acquired with a scanning device. For this, we will introduce a number of algorithmic concepts which enables a solution of the problem. Every one of these components is a novel contribution on its own. We would like to emphasize the following main contributions of this work:

- We propose matching model that is robust to topological noise and that can quantify a matching uncertainty.
- We show a landmark tracking algorithm that establishes sparse correspondences fully automatically under both temporally coherent as well as under arbitrary, abrupt motion.
- We propose a final reconstruction pipeline which is capable of finding a global solution even on temporarily and spatially sparse data

The reconstruction pipeline is then applied to a large set of data acquired with different 3D scanners.

---

# 3

## Pairwise Shape Matching by Randomized Sampling

---

Let us first consider the pairwise matching problem. The algorithm proposed in this chapter starts out by computing correspondence candidates through feature matching. This results in a superset of correspondences from which we have to extract a correct subset. We also compute geodesic distances between all feature points, on both the source and the target shape, which gives us a superset of geodesics where some of those might actually be incorrect and give false cues for validating the correctness of the correspondences. We are looking for a subgraph of consistent correspondences and geodesics that is maximal in the sense that no edges can be added without exceeding an error threshold. Out of the many different solutions that may exist, we will prefer large solutions with many correspondences, each validated by many pairs of geodesic distances, as these are less likely to be spurious matches. In order to compute such solution, we use a RANSAC-like algorithm that randomly samples the solution space. For efficiency reasons, we bias the random search towards promising matching candidates by employing a suitable importance function. In order to perform this algorithm reliably, we need a criterion for the reliability of verifying geodesics, which we develop subsequently. Next, we employ a tangent-space optimization technique to compute an optimal placement of the features. After having obtained a good set of feature matches, we extend the correspondences to so far unlabeled space by inserting new secondary features derived from distances to matching correspondences. Finally, this dense feature set is also fine tuned by again applying tangent-space optimization. In an outer loop, we execute the whole matching algorithm repeatedly in order to find matching alternatives which are ranked by how well they explain the input data. In the following, we discuss all of these steps in more detail. Figure 3.1 illustrates the mentioned pipeline.

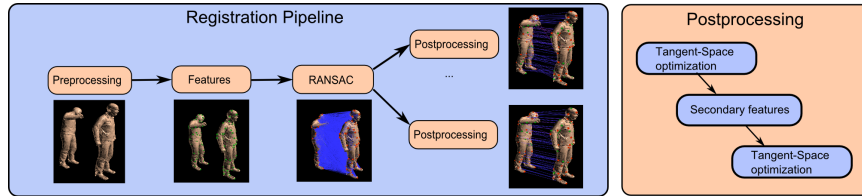


Figure 3.1: Pipeline overview of our algorithm. A pair of input point clouds is first preprocessed Sect. 3.1. Afterwards a set of feature points and a superset of all correspondences is computed, Sect. 3.2. Then a RANSAC-like search of candidate solutions is performed, Sect. 3.3. And finally each solution is being postprocessed, Sect. 3.5 and 3.7.

### 3.1 Input Data and Preprocessing

We directly work on the point-based representation as provided by a 3D scanning device. We expect two sets of 3D points  $M_S$  and  $M_T$  as input. These two point sets are discrete versions of two (unknown) smooth manifolds  $\mathcal{M}_S$  and  $\mathcal{M}_T$ , where  $S$  stands for source and  $T$  for target, however their meaning is completely interchangeable. Using the  $k$ -nearest graph, i.e., “connectivity graph”, as described in Chapter 1, we first compute normals for each data point using PCA. Please note that this graph does not need to form a valid triangle mesh.

### 3.2 Feature Points

In the next step, we compute a candidate set of surface feature points. In principle, any surface feature detection technique can be employed at this point [46, 56, 79]. In our work, we use slippage features [16]. Slippage features detect keypoints by maximizing the stability of the auto-alignment of local surface pieces in scale space, which leads to a large number of stable feature points. In the following, we will refer to the features on surface  $j \in \{S, T\}$  as  $\mathbf{x}_i^j \in \mathbf{X}^j, i \in \{1, \dots, n_j\}$ . Given a set of feature points, we next compute a local descriptor for small circular neighborhoods of each feature. We use a rather simple descriptor that just computes a histogram of mean curvature in this region. Mean curvature is computed from a quadratic moving-least-squares surface approximation (see [16] for details). Having computed feature points and descriptors, we build an initial matching graph: We connect all features on  $M_S$  and  $M_T$  with *correspondence edges* for which the



descriptors are similar up to at least a user defined threshold (we choose a conservative threshold that leads to many false positives, as we will subsequently filter the graph further). Additionally, we also connect all pairs of features  $\mathbf{x}_i^S$  in shape  $M_S$  and all pairs of features  $\mathbf{x}_i^T$  in shape  $M_T$  with *validation edges* and compute the approximate geodesic distance between these pairs. Each validation edge is tagged with this distance.

### 3.3 RANSAC Subgraph Extraction

Given the input candidate graph, we extract a consistent subset using a RANSAC-like randomized sampling algorithm. Naively, we could enumerate all possible subgraphs of our candidate set and evaluate how well it explains the data. However, this would lead to costs of  $O(2^N)$  where  $N$  is the overall number of validation and correspondence edges. By random sampling, we can examine the same search space in expected time  $O(N \cdot 2^N)$  [92], which is slightly worse. However, by augmenting the sampling density, we can obtain a more efficient solution. Obviously, it is not reasonable to try matches that are very unlikely to be correct. Therefore, we will employ importance sampling, which distorts our random sampling density in a way to yield promising matching candidates with larger probability. This will drastically reduce the expected time required to find a good match. We can further motivate the efficiency of this scheme using a heuristic argument: For the first match, we can only rely on the rather noisy descriptor matches as importance function. These matches typically have outlier rates of 80%, i.e. 4 out of 5 matches are wrong. However, once we have found at least 2-3 correct matches, this establishes a local coordinate frame in terms of geodesic distances, so that it becomes much easier to judge whether new correspondence candidates are correct. From this perspective, we only need a small number of random guesses to find a candidate set that bootstraps the matching process, rather than an exponential number in  $N$ .

In order to implement this strategy concretely, we employ the following algorithm: Start by selecting a random correspondence  $m_{i,j} = (\mathbf{x}_i^S, \mathbf{x}_j^T)$  according to an importance sampling density  $p(m_{i,j})$ . Initially, this density is chosen to be proportional to the descriptor matching score. In our case we use a gaussian distribution over the descriptor distances:

$$p(m_{i,j}) = \frac{1}{Z} \exp\left(-\frac{(\mathbf{D}_{\mathbf{x}_i^S} - \mathbf{D}_{\mathbf{x}_j^T})^2}{2\sigma_d^2}\right), \quad (3.1)$$

where  $\mathbf{D}_{\mathbf{x}_i^S}$  and  $\mathbf{D}_{\mathbf{x}_j^T}$  are the descriptor vectors of points  $\mathbf{x}_i^S$  and  $\mathbf{x}_j^T$  respectively and  $\sigma_d$  is a factor how strict we penalize descriptor differences.  $Z$  is a normalization factor.

Then iteratively add more correspondences using an augmented sampling density that takes the geodesic distances to existing features into account. These distances will not be exact because of slightly non-isometric deformations, as well as noise in the feature location and distance estimation process. For simplicity, we assume that the aggregate of these errors, for a *correct* match, yields a Gaussian distribution. The standard deviation of this distribution could be calibrated by user-labeled example matches from the data source considered; we set this parameter manually. If we already have several correspondences in our candidate sets for the correct subgraph, we have to check multiple geodesics. We assume that the error affecting the geodesic lengths is independent for each geodesic. While this is certainly not exactly the case, it is a reasonable approximation in practice. In this model, we can compute the probability of length distortion. Let  $d_{kj}^T = d_{M_T}(\mathbf{x}_k^T, \mathbf{x}_j^T)$  denote the actual geodesic distance observed on the target shape between feature points  $\mathbf{x}_k^T$  and  $\mathbf{x}_j^T$  and analogous  $d_{ki}^S = d_{M_S}(\mathbf{x}_k^S, \mathbf{x}_i^S)$  is the geodesic distance between feature points  $\mathbf{x}_k^S$  and  $\mathbf{x}_i^S$  on the source shape which are already detected as being in correspondence to the feature points on the target shape. The probability density for the match  $m_{i,j}$  being correct is thus given by:

$$p(m_{i,j}|C) = \frac{1}{Z} \prod_{k=1}^{|C|} \exp\left(-\frac{(d_{ki}^S - d_{kj}^T)^2}{2\sigma^2}\right) \quad (3.2)$$

Here,  $C$  is a set of already matched feature points pairs between the shape  $M_S$  and  $M_T$ . This formula is directly used to choose the next match to be added to our current subgraph: We compute the matching likelihood, given by this formula, and use the corresponding probability density for importance sampling. For the length differences, we assume that the geodesic lengths are correct in the first shape and randomly distorted in the second shape, with standard deviation increased by  $\sqrt{2}$ <sup>1</sup>, which is equivalent to having noise in both shapes. For importance sampling, we generally exclude any matches that receive a very low matching probability (typically, below 1%). Therefore, the iteration terminates automatically when no more reasonable matches are available. As an alternative to pure importance sampling,

---

<sup>1</sup>The increase in the variance of the distortion of the geodesics by 2 results in the  $\sqrt{2}$  increase in the standard deviation

we can also use only the best (maximum a posteriori) match once a sufficient number (typically 3-5) of base correspondences have been sampled. In our experiments, this leads to better results in comparison to the purely random algorithm.

This presented correspondence estimation algorithm is similar to PROSAC [33] and the forward search of [56]: it changes the sampling distribution during the execution of a single draw in order to avoid sampling non-useful matches as much as possible. This is a major key to efficiency, as the algorithm automatically restricts the search space during the sampling. The randomized algorithm will still need exponential time, in worst case, to find the absolutely best discrete match with least intrinsic distortion. However, solutions that are correct within a small factor of sampling and noise accuracy can be found with only a small number of trials, and this is what we are looking for. In this case, the randomized algorithm is usually exponentially faster when applied to densely sampled surfaces.

## 3.4 Handling Topological Noise

The algorithm outlined above yields very good results, comparable to state-of-the-art matching algorithms such as spectral geodesic matching [55, 74] (see Figure 3.9). However, this may fail once the geodesics become unreliable. As mentioned in the conclusions of [55], one needs to be able to handle geodesics in an outlier robust way, which is hard to incorporate in standard techniques such as pairwise spectral validation. For our RANSAC-like matching algorithm, however, this is rather easy to achieve: In a plane, two different points are sufficient to find unique geodesic coordinates for all other points up to mirroring along the line defined by the coordinate points, and three non-collinear points define a unique barycentric coordinate system. On curved manifolds, the situation might be more complex. We will take a look on this complexity in Chapter 4. In practical cases, it is highly unlikely to have two different points with the same geodesic distances to a larger number of feature points (say, to 10 different points). Therefore, we do not need to guarantee full geodesic consistency but it is sufficient to have a large enough witness set that proves the correct match, while further geodesic inconsistencies can be regarded as outliers due to topological noise. Now, we augment our RANSAC matching loop as follows: In computing the matching probabilities (3.2), we do not use all geodesics but determine the subset of geodesics that does not show

too large deviations in the distances and use only this subset for validation, i.e.  $|d_{ki}^S - d_{kj}^T| < \tau$ , where  $\tau \approx 3\epsilon_s$ . In order to make this reliable, we demand to have at least  $k$  such geodesics, with  $k$  typically in the range of 5-10. This means that out of  $n$  geodesics,  $n - k$  are allowed to be outliers. We refer to this as *outlier robust matching*, since we increase the robustness against outliers in the geodesic domain.

### 3.5 Tangent-space optimization

Feature positions are not exact as they rely on the ability of the feature detection algorithm to find well-defined spots. We usually cannot afford to use a very firm threshold on the well-constraintness of the feature position because this would remove too many features, limiting the scope of the matching. In addition, we may have wrong matches to actually different but nearby features that still meet our geodesic validation criterion. By moving these features slightly we can obtain a much better matching. Assume we are given  $k$  reference points and one more point  $\mathbf{y} \in M$  with geodesic distances  $\mathbf{d} = (d_1, \dots, d_k)$ . At  $\mathbf{y}$ , we form a two-dimensional tangent space coordinate system of orthogonal vectors  $(\mathbf{u}, \mathbf{v})$  that are orthogonal to the surface normal  $\mathbf{n}(\mathbf{y})$ . Let  $\mathbf{y}^{(uv)}$  be the coordinates of  $\mathbf{y}$  in tangent space. We consider the partial derivatives of the position of  $\mathbf{y}^{(uv)}$  by the length of the geodesics. Up to first order, we obtain:

$$\Delta \mathbf{y}^{(uv)}(d_1, \dots, d_k) \doteq \underbrace{\left( \frac{\partial \mathbf{y}^{(uv)}}{\partial d_1} \dots \frac{\partial \mathbf{y}^{(uv)}}{\partial d_k} \right)}_{=: \nabla \mathbf{d}} \Delta \mathbf{d} \quad (3.3)$$

where  $\Delta \mathbf{d}$  is the displacement in geodesic coordinates. We get a linear map  $\nabla \mathbf{d}$ , i.e. Jacobian matrix, that transforms small displacements in geodesic coordinates into displacements in spatial coordinates. In order to compute the gradients  $\partial \mathbf{y}^{(uv)} / \partial d_i$  we consider a spherical intrinsic neighborhood  $N(\mathbf{y})$  of points and compute geodesic distances to the reference point for all points in the point cloud. This comes at no additional cost, as this has to be done during the Dijkstra algorithm anyway. Afterwards, we compute a least-squares fit of a linear model to the resulting distance values (with Gaussian weighting window with standard deviation proportional to the radius of  $N(\mathbf{y})$ ). This means, we compute the coefficients  $a_0, a_1, b$  of a linear model  $\langle \mathbf{a}, \Delta \mathbf{x} \rangle + b$  that fits the observed distances best in a weighted-least-squares sense. The vector  $\mathbf{a}$  then yields our gradient approximate

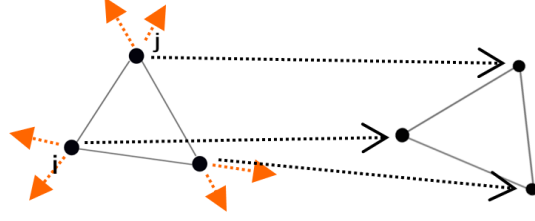


Figure 3.2: Illustration of the proposed tangent space optimization technique. Dashed arrows represent the correspondences, and the edges between the nodes the validation edges. Red arrows represents the surface gradients  $\mathbf{g}_{i,j}^{(k)}$  in the  $uv$ -plane over the points. By a small movement of every point along its gradient directions we can decrease the numerical matching error.

in tangent space. Using this first order approximation, we can setup a quadratic objective function that locally describes how well the length of matching geodesics is balanced. We then optimize this function, move the points in tangent direction, project them back on the manifold using an MLS surface approximation [3], and recompute new estimates for geodesic distances and their gradients in every step, leading to a fairly efficient Gauss-Newton-type iteration. Figure 3.2 illustrates the proposed tangent space optimization method.

We obtain the quadratic objective function by expressing the change of length of the geodesics in terms of gradients of the geodesics with respect to the point position. In the following, we use  $d_{i,j}^{(k)}$  to denote the intrinsic distance between two feature points  $\mathbf{x}_i^{(k)}$  and  $\mathbf{x}_j^{(k)}$  on surface  $k \in \{S, T\}$ . Correspondingly, we use  $\mathbf{g}_{i,j}^{(k)} := \partial \mathbf{x}_i^{(k)} / \partial d_{i,j}^{(k)}$  to denote the gradient of the geodesic distance with respect to the tangent space ( $uv$ -) coordinates of a feature point  $\mathbf{x}_i^{(k)}$ . The notation means that the gradient is measured at feature  $i$  on surface  $k$  for a geodesic that connects feature  $i$  to feature  $j$ . Thus, the distance between two feature points can expressed in terms of the distances on the reference shape<sup>2</sup> as:

$$d_{i,j}^T = d_{i,j}^S + \mathbf{g}_{i,j}^S \cdot \delta_i^S + \mathbf{g}_{j,i}^S \cdot \delta_j^S$$

which yields the following objective function:

$$\arg \min_{\delta_i^S \in \mathbb{R}^2} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n (d_{i,j}^S + \mathbf{g}_{i,j}^S \cdot \delta_i^S + \mathbf{g}_{j,i}^S \cdot \delta_j^S - d_{i,j}^T)^2 \quad (3.4)$$

<sup>2</sup>In this case the reference shape is the source shape, however their meaning is interchangeable

This is a quadratic objective function in the unknown tangential displacements  $\delta_i^S$  and  $\delta_j^S$ . Formulating the objective function as energy minimization problem:

$$E(\Delta) = \sum_{i=1}^n \sum_{j=1}^n (d_{i,j}^S + \delta_i^S \mathbf{g}_{i,j}^S + \delta_j^S \mathbf{g}_{j,i}^S - d_{i,j}^T)^2,$$

we can find the minimum if we set the first derivative of the energy function to 0. The first derivative of a complete energy functional is:

$$\nabla E(\Delta) = \sum_{k=1}^n (\nabla_{\delta_k^S} (E(\Delta))),$$

where  $\nabla_{\delta_k^S} (E(\Delta))$  is a derivative of  $E(\Delta)$  in respect to the  $\delta_k^S$  variable.

After a simple algebraic reformulation the quadratic objective function, Equation 3.4, reduces to the following linear equation system:

$$\forall k : \sum_{i=1, i \neq k}^n \left[ (\mathbf{g}_{k,i}^S \mathbf{g}_{i,k}^{S^T}) \delta_i^S + (\mathbf{g}_{k,i}^S \mathbf{g}_{k,i}^{S^T}) \delta_k^S \right] = - \sum_{i=1, i \neq k}^n (d_{i,k}^S - d_{i,k}^T) \mathbf{g}_{k,i}^S,$$

where  $\mathbf{g}^{S^T}$  is a transposed gradient vector.

We solve the resulting linear system using SVD, which is robust to degenerate cases (numerically small absolute singular values are not inverted). We then move the feature points tangentially according to the computed displacements in three-dimensional space. Afterwards, we project each point back on a surface approximation obtained from a moving least square fit (with quadratic basis functions and Gaussian weights). Then, geodesics are recomputed and the scheme is iterated until it converges, i.e., only small changes in energy occur. We move the point by at most the surface sample spacing  $\epsilon_{\text{sampl}}$  in order to prevent missing the surface in the projection step. In order to make the scheme symmetric, we alternate between optimizing the feature positions on surface  $M_S$  and  $M_T$  in each iteration. In practice, this symmetric approach shows to be significantly more robust and accurate than a one-sided optimization, moving feature points only on one surface.

### 3.6 Approximation of Geodesics

Each iteration of the tangent-space optimization step as described in the previous section requires a recomputation of intrinsic distances  $d_{i,j}^{(k)}$  and corresponding gradients  $\mathbf{g}_{i,j}^{(k)}$  that involves running Dijkstra's algorithm for each feature point, which



Figure 3.3: Error of approximated intrinsic distances, blue = 0% and red = 100%. The image shows the *maximum* error over all geodesics between each point and all other points of the point cloud. The error is measured in respect to the non-downsampled versions of the shapes. *Horse*: 8431 original points, 1355 sample points - max error is 5% of the bounding box size. *dragon*: 20002 original points, 1598 sample-points - max error is 10%.

is quite costly. In order to solve this problem we use an algorithm to precompute an approximation to the geodesic distance and the corresponding gradients between any two points on the point cloud surface: First, we compute sampling points  $y_n$  from the original point cloud. We downsample the point cloud with respect to the curvature by a Poisson-disc sampler with radius inversely proportional to the curvature. For the downsampling factor we have a trade-off between the absolute error and preprocessing time we are willing to tolerate. Figure 3.3 shows the absolute error achieved with typical parameter settings.

Having the data points we compute graph distances between the sample points using the original, full resolution graph for distance estimation. Next, we interpolate intrinsic distance  $d_{i,j}$  and gradient  $\mathbf{g}_{i,j}$  between any points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  on the surface. First we determine the  $k$  nearest data points  $\mathbf{y}$  around both points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  respectively where  $k$  is usually 4. Next, we place Gaussian basis functions around each sample point with radius (standard deviation) proportional to the point spacing and obtain the interpolated distance by a partition-of-unity interpolation: For each source point, we compute a partition-of-unity interpolation of all distances to the destination and interpolate these again using a second partition-of-unity interpolation in the destination domain.

### 3.7 Inserting Dense Secondary Features

The robust criterion for geodesic validation can also be employed to insert new secondary feature points with correspondences into our model. This is desirable

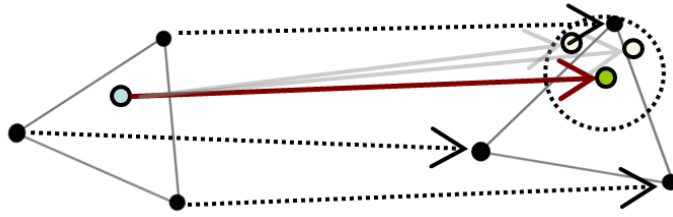


Figure 3.4: Illustration of the secondary features insertion. A position of the point from the source shape is interpolated on the target shape as described in 3.6. In order to be able to interpolate, valid correspondences have to exist in a given radius. Next in a small Euclidean area around the interpolated position we look for a point which minimizes the matching score Equation 3.2.

as the feature detection stage often fails to yield a sufficient number of feature points to cover the surface densely enough. Inserting new correspondences is easy: We first pick a random surface point out of the sample points  $\mathbf{y}$  as described in the previous section. Then, we go through all points of the target surface and pick the best matching one. We apply exactly the same matching criterion as for regular features described previously: We require a minimum number of geodesic distances to be correct within tolerance and that this subset is stable. We add the best matching of those correspondences to our correspondence set. This is iterated until all points are taken or no more reliable matches are found. In order to speed up this brute force search algorithm, we try to reduce the search area: First, we find all already matched features within a small neighborhood of  $r \cdot \epsilon_{\text{sampl}}$  of the secondary feature point (typically:  $r = 8$ ). We compute the Euclidean distances and perform a partition-of-unity interpolation of the corresponding points, interpolating the target positions in Euclidean space, which we then project back on the target surface. We now examine all target points within distance  $r \cdot \epsilon_{\text{sampl}}$  using a breadth-first search on the connectivity graph. Only if this search yields no result, we fall back to the expensive brute force search. In practice, this yields a substantial speed-up. Figure 3.4 illustrates the described technique.<sup>3</sup>

<sup>3</sup>We refer those readers who would like to know more effective approaches for dense correspondence estimation from a set of sparse correspondences to Chapters 4 and 5. In the first case, the idea of computing dense features from the set of sparse features is simplified to the case of just a vector comparison. This can be achieved due to the theoretical fundament that we create in developing the shape matching approach of Chapter 4. In Chapter 5 a more sophisticated method derives a set of dense features by incorporating the *bijection* criterion, since any matched shape has to match in the



### 3.8 Ranking of Matching Alternatives

The algorithm described so far will produce a single isometric match of the two input surfaces. In case of multiple valid solutions, it is a matter of chance which solution will be output. For equally plausible results (similarly isometric, same amount of features and area covered), the algorithm will output all these results with an equal probability. This randomization is actually useful to gain more information about the matching of the two shapes involved. We run the algorithm several times in an outer loop: Instead of using only the best initial feature match from the inner RANSAC loop, we keep the  $k$ -best matches (typically  $k = 10$ ). We run the whole pipeline for the first match. For the next best feature match, we then determine whether it is contained in the previous solution. The problem here is that feature positions might be different so that we cannot directly compare the results. Therefore, we interpolate the matching results using a partition-of-unity Gaussian interpolation with window radius  $2\epsilon_{\text{sampl}}$  and compare the obtained interpolated correspondences with the feature correspondences. Points without feature correspondences in their vicinity are treated as unmatched area and are ignored in the comparison. Only if there is a substantial disagreement, we compute a new solution from the initial feature match. Otherwise, we dismiss the match and go on examining the next best. This is repeated until all matches have been compared. In the end, we rank the results by counting the number of overall features (initial and secondary features) matched, which corresponds to matched area, multiplied by the number of geodesics that support each of the validated features. If the numbers are equal, the error in the deviation of geodesic distances is used as secondary sorting criterion. This output is a sampled description of the matching ambiguity present for the two shapes, with the “most likely” reconstruction listed first.

### 3.9 Implementation and Evaluation

We tested our algorithm on an Intel Core2 CPU with 2GHz and 2GB RAM with different synthetic and real world data sets. Table 3.1 summarizes the timings and statistics. The most expensive step is the precomputation of geodesic distances, which is necessary for feature optimization. If we restrict ourselves to matching original features only (without optimization), we obtain running times in the range

---

other direction too.

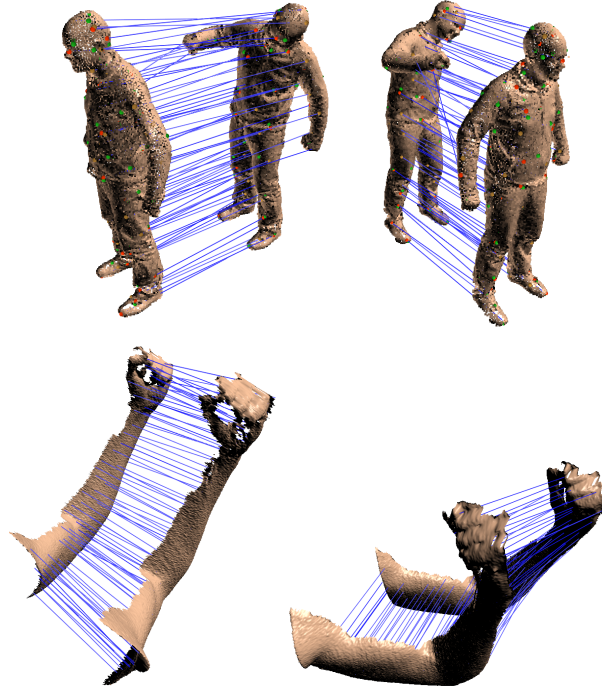


Figure 3.5: Matching results: (Top) guy data set 75 features (35 secondary features) were registered. (Bottom) arm-dataset registered 84 feature (56 secondary features).

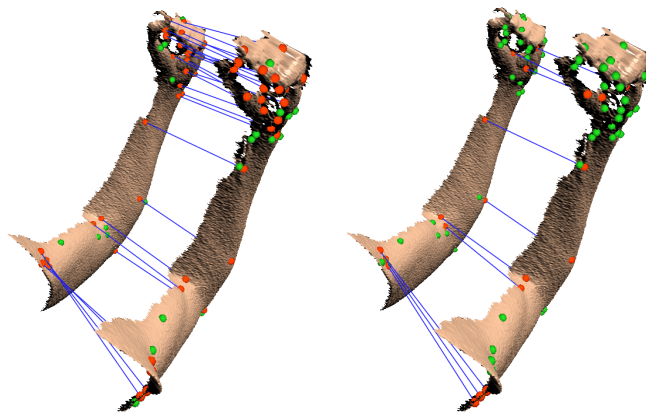


Figure 3.6: Arm data set matched with (left) our algorithm (no secondary features) and (right) spectral matching. Red colored points represents features for which a correspondence was found. Green colored points are features with no correspondence.

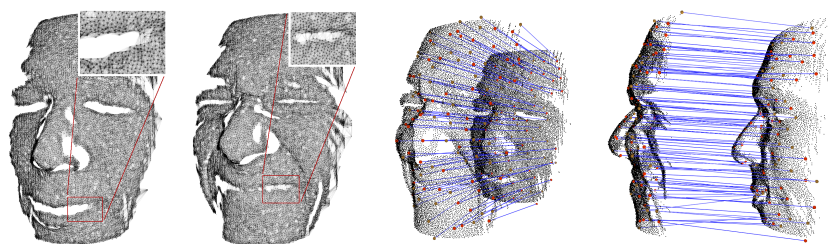


Figure 3.7: Faces data set and its registration. The dataset represents a hard case with different connectivity, i.e., topological noise, due to the misconnection in the mouth region.

of about one minute, again dominated by computing geodesics.

**Synthetic data:** First, we apply our algorithm to two well known 3D models, horse and dragon, which we sample synthetically to simulate scanner data. Figure 3.8 shows the matching results along with a comparison to ground truth. The horse data set is particularly hard to match because of the skinny and mostly featureless legs. In addition, the model is perfectly symmetric under left/right-mirroring in its rest pose. Because of the larger similarity between the right and left legs of the horse, our algorithm actually prefers the mirrored solution as best match. Figure 3.9 shows a registration of an artificial data set. The two frames of the data set are equal except for the holes added in the second frame. We were able to match more features enabling the outlier robustness than with spectral matching [74] or with a simple RANSAC matching. For a fair comparison, matching was performed without tangent-space optimization and with no additional secondary features.

**3D scanner data:** We examine three real world 3D scan data sets (original raw data up to downsampling): “guy”, “arm”, and “face”. All three suffer from both geometrical and topological noise problems. Figure 3.5 shows the result for the arm data set. Please note the big acquisition hole on the palm as well as the fingers touching in only one scan. Our algorithm was able to register 84 features. Figure 3.6 shows a side-by-side comparison of our algorithm with spectral matching, which does not capture features in regions of connectivity variation. To compare both algorithms neither tangent space optimization was performed nor secondary features were added. Results for the face data set are shown in Figure 3.7. Again, in the mouth and eye regions, false connections occur. We were able to register 103 features (49 secondary features) in around 8 minutes. The example shows one

data set	$ \mathbf{X}^S  \cdot  \mathbf{X}^T $	sample pts	precomp.	registr.	$ C $	score
dragon	4760	3196	40m13s	7m2s	95	7.43
horse	7740	1233	9m20s	11m49s	100	6.95
guy	2646	868	9m4s	3m40s	75	5.01
arm	1440	1974	16m13s	4m38s	84	0.67
face	7238	1920	17m30s	8m10s	103	4.82

Table 3.1: Test data sets (name, number of candidate correspondences, number of sample points for approximating geodesics, precomputation and registration time, number of finally matched correspondences  $|C|$ , matching score, i.e. sum of log-likelihoods in Equation 3.2.

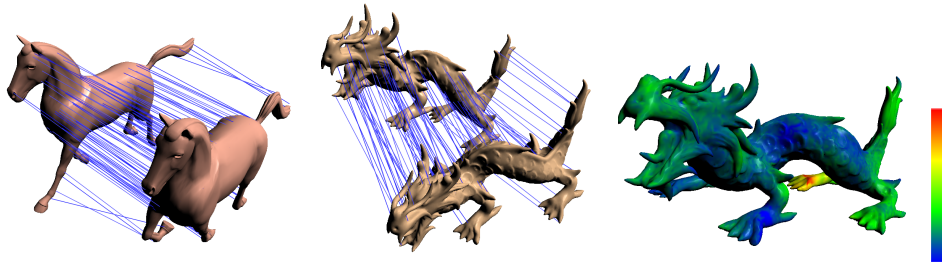


Figure 3.8: Synthetic data set: horse and dragon. (Left) 100 correspondences while matching the horse data set were found. (Right) dragon data set matched with 95 features (48 secondary) with a maximal error (red) of 11%.

limitation that our technique shares with all isometry-based matching techniques: The portion of the face above the upper lip is not deforming isometrically (at least not as portrait at scanner resolution), so that the upper lip matches a bit too low. As this does not appear as a spurious outlier, but a consistent shift, the registration does not yield the intuitively expected result. For the rest of the face, we obtain reliable results. Figure 3.11 shows results for partial matching with ambiguities - we match the forefinger of a scanned hand to the full hand model. Among the 8 best matches are matches to two different fingers.

### 3.10 Discussion

We have shown a global deformable matching approach based on a novel RANSAC-like randomized sampling algorithm. The algorithm works for general data sets solely assuming approximately isometric deformations. The algorithm is

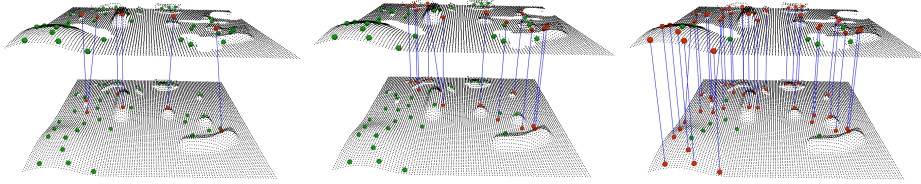


Figure 3.9: Synthetic dataset illustrating robustness against topological noise. From left to right: spectral matching algorithm (6 matches), our algorithm without outlier robustness (12 matches) and with outlier robustness enabled (33 matches) respectively. No optimization step and no secondary features were added.  $|\mathbf{X}^S| \times |\mathbf{X}^T| = 1400$

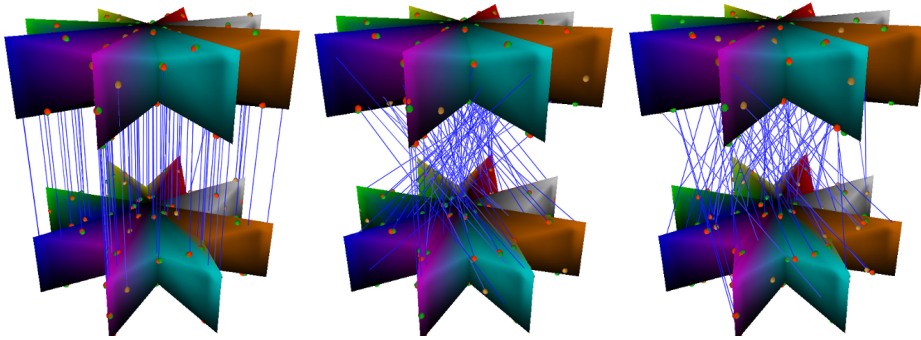


Figure 3.10: 3 solutions of matching the synthetic Star data set to itself. Due to the symmetry the matched solutions have low error with respect to geodesic distances.

robust to topological noise and unlike previously known techniques, our approach is able to output matching alternatives by sampling the space of plausible solutions. This might be an important tool in multi-part matching situations with ambiguous pairwise matches, such as animation sequence reconstruction.

One of the limitations of the approach presented here is the limitation to feature points only. Hence, the RANSAC matching times as well as the overall matching quality depend highly on the ability to place feature points well. Here, a question arises: “What if we were able to find an optimal feature point placement in order to solve the isometric matching problem best?”. In the next chapter we will study the answer to this question.

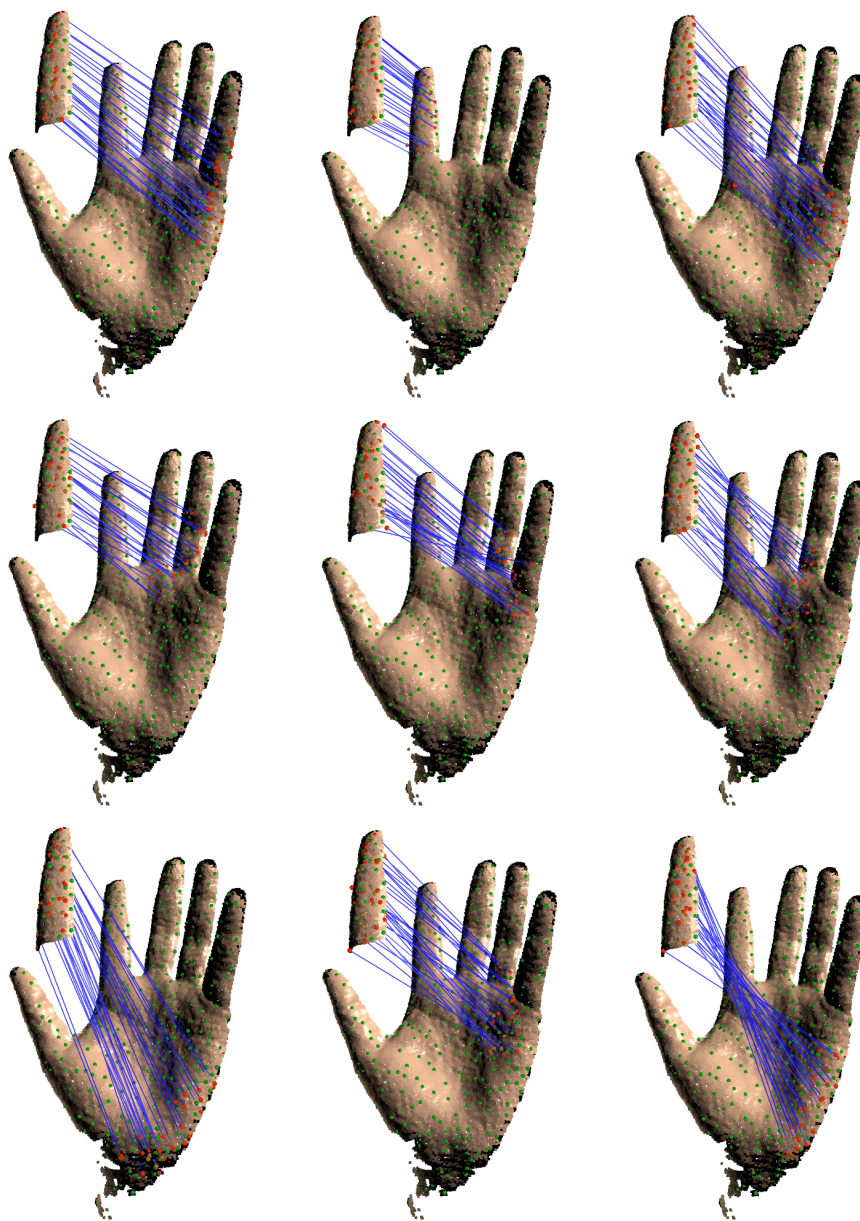


Figure 3.11: First 9 solutions of a finger-hand matching. Due to the symmetrical structure there exist ambiguous solutions which were found with our algorithm.

---



## PLANSAC: Plan First Before Execute

---

The method derived in the previous chapter helps to understand how to compute correspondences between a pair of shapes. However, as already pointed out, the problem of pairwise matching is not that well suited if limited to feature-based approaches. The main disadvantage of feature-based approaches is the potential lack of features on a shape, because keypoint detectors are designed for certain data characteristics, such as local bumps in the surface [46, 79]. These might not be present in sufficient number on some models, thus preventing successful matching. Additionally, feature point detectors depend in most cases on extrinsic information, Chapter 3. These, however, are not invariant in respect to the isometry. Changes in these properties reflect directly in the placement and detection quality of feature points. Imagine two frames of a hand where in one frame the fingers are globed into a fist whereas in the other frame they are separated. In this case a frame with a flat hand is less wrinkled than a globed fist and hence has less prominent, extrinsic, shape features.

In this chapter we will present a new method of pair-wise shape matching which does not rely on feature points. In contrast, our technique fully automatically detects a set of optimized reference points for matching, adapting automatically to the characteristics of the input. In general, we are interested in the question how many of these reference points we need for successful matching and how we should distribute them on the surface in order to solve the isometric matching problem between two shapes in the numerically best way.

Similar problems have already gained some attraction from the research community. Lipman et al. [80] prove that three point correspondences are sufficient to fix an isometry on topological discs and spheres, and, Ovsjanikov et al. [96] strengthen the result: By analyzing global shape properties encoded in heat-kernel functions, a single correspondence is sufficient to fix an isometry if the shape and

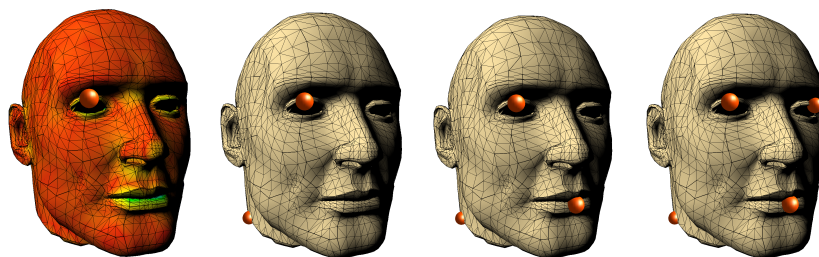


Figure 4.1: Landmarks computed for a human face, with one landmark added at each step. The first image shows the distribution of uniqueness (i.e. from blue: most unique, to red: most similar to other) of the first landmark. Further points build up a plan: Each further point is chosen to be of maximal utility for solving the isometry matching problem.

the point meet certain conditions (implying in particular the absence of continuous sets of self-isometries). These results establish that the intrinsic complexity of finding isometric matches is rather low because the problem does not allow for a large number of degrees of freedom. While these remarkable results solve much of the problem in theory, the practical complexity is still unclear: In real-world scenarios, we have to deal with input data that is undergoing only approximately isometric deformations, there are numerical precision issues, and the intrinsic distances are not reliable due to topological noise.

For example, if an input shape consists of two pieces connected with a narrow tunnel of very small diameter, fixing intrinsic distances on one piece will in practice not fully constrain the matching on the second piece even if this was the case if we had a perfectly isometric deformation and infinite numerical resolution. Consequently, the practical implementations of [80,96] employ voting involving multiple correspondence sets in order to obtain stable results in practice. For real-world scenarios, it remains unclear how much correspondence information is necessary to fully constrain the matching problem.

**Our contributions:** Our key idea is to look at the entropy of the posterior distribution of possible matches to assess what are good points we need to distribute over the surface. Starting from an empty set of prior knowledge, we incrementally add points that minimize the entropy of the posterior distribution of potential matches and thereby minimize the effort of finding a new piece of information in the next round. A landmark coordinate scheme [55, 96] is then employed to immediately



extend the solution to dense correspondences. The set of points computed in this way is called a *plan*. The plan can be pre-computed given only the source shape. It can then be reused for matching to a large number of target shapes. This leads to almost real-time matching performance in situations where a template shape is matched against a large collection of input data.

Our algorithm leads to significant savings in sampling costs compared to random sampling (Chapter 3). It provides also an insight on the practical complexity of near-isometric shape matching. Based on ideas in this chapter Ovsjanikov and colleagues [98] developed a so called *condition number* of a shape which maps the complexity or difficulty of matching different types of shapes to a single number and hence makes the difficulty of shape matching comparable.

## 4.1 Isometry invariant matching

In the following, we assume that we are given two manifolds  $\mathcal{M}_S$  and  $\mathcal{M}_T$ . Both are equipped with a distance metric  $d_{\mathcal{M}}(\cdot, \cdot)$  that measures the intrinsic distances, see Chapter 1 for a more detailed definition. We assume that  $\mathcal{M}_S$  and  $\mathcal{M}_T$  are isometric to each other, i.e., there exists a bijective function that maps the two shapes onto each other that is an isometry with respect to the intrinsic metric. Our task is to find such a mapping.

In order to solve this problem efficiently, we first resample the surfaces to a  $\epsilon$ -resolution and refer to the discrete version of  $\mathcal{M}$  as  $M$ , where  $|M| = n_s$ . For resampling, we use a standard uniform Poisson disc sampler. Furthermore, we assume that pointwise connectivity information is given which is for example the case for triangle meshes. Hence we can compute intrinsic distances as shortest paths along the mesh surface as shown in Chapter 1. For the distance computation, we use the full resolution mesh. In general, geodesic distances are very sensitive to topological changes. One could stabilize the distances by using heat kernel distances [62], which in particular are robust against slight topological noise, however we relinquish this possibility.

We pose the approximate intrinsic matching problem as an optimization problem, finding a function  $\hat{f} : M_S \rightarrow M_T$  that minimizes the Equation 1.2 presented in Chapter 1. Finding the exact minimum of Equation 1.2 could lead to a hard combinatorial problem. However, differences in correspondence at the level of the sampling resolution do not matter to us as we use the discretization only as a

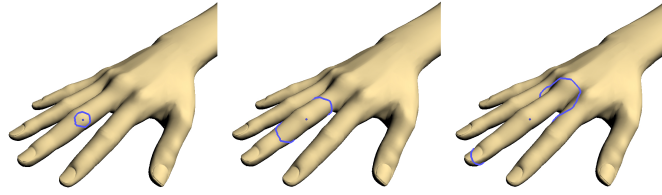


Figure 4.2: Our descriptor measures the lengths of the isocurves of increasing distance: the curves spread like a “wave” over the surface. The length of a given isocurve is highly dependent on the local shape and encodes local properties nicely.

tool for representing the surface. Therefore, we usually set a fixed error bound of  $c \cdot \epsilon$  with  $c \approx 2..3$  and consider a solution correct if it meets this precision bound. The final optimum could then be computed by a locally convergent numerical optimization [55], as shown for example in Section 3.5. Thus, this precision limit is no restriction in practice. Correspondingly, we focus in the following on finding an approximate solution in this sense as quickly as possible.

## 4.2 Shape matching by random sampling

We can now formulate the global matching algorithm. The algorithm is oriented on the RANSAC sampling strategy as presented in Chapter 3 and is reminded here shortly: We start by picking a random match between points from  $M_S$  and  $M_T$ . All pairs  $(\mathbf{x}, \mathbf{y}) \in M_S \times M_T$  are considered as candidate matches with a probability proportional to

$$P_{descr}(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\left(-\frac{(\mathbf{D}_x - \mathbf{D}_y)^2}{2\sigma_d^2}\right). \quad (4.1)$$

$\sigma_d$  is a user parameter that describes the standard deviation of matching descriptors (thus depending on model noise) and  $Z$  is a normalization factor. We use the same descriptor matching score as already shown in Equation 3.1.

As a descriptor we incorporate the intrinsic “wave” descriptor as described in [140]. In general the idea is to measure the length of isocurves, i.e. points of the same intrinsic distance  $\rho$ , marching from a start point. The length is normalized by dividing by  $2\pi\rho$ , which is the value expected for a flat (i.e., developable) piece of surface. The descriptor  $D_x(\rho)$  for each point  $\mathbf{x} \in \mathcal{M}$  is the function that maps  $\rho$  to the normalized length of the corresponding isocurve. Figure 4.2 illustrates the

isocurves and Figure 4.3 illustrates the descriptor signatures for an example model. In our implementation, we sample 16 values at constant intervals of  $\rho$ .

The actual correspondence matching is performed by random draws of possible correspondence candidates, taking the previous information into account: Points on the source shape  $M_S$  are chosen randomly. The likelihood for picking target points from  $M_T$  is weighted by the likelihood that the intrinsic distances are preserved: If one correspondence has been fixed, and a second source point has a distance  $d$  on the source shape, only points with a distance of roughly  $d$  on the target shape are accepted, again following a Gaussian error model. Given we already have  $k$  correspondences  $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_k, \mathbf{y}_k)\}$ ,  $(\mathbf{x}_i, \mathbf{y}_i) \in M_S \times M_T$ , the likelihood for one more correspondence  $(\mathbf{x}, \mathbf{y}) \in M_S \times M_T$  being correct is given by a Gaussian error model:

$$P_{dist}(\mathbf{x}, \mathbf{y} | \mathbf{x}_1, \mathbf{y}_1, \dots, \mathbf{x}_k, \mathbf{y}_k) = \frac{1}{Z} \prod_{i=1}^k \exp\left(-\frac{(d_{M_S}(\mathbf{x}, \mathbf{x}_i) - d_{M_T}(\mathbf{y}, \mathbf{y}_i))^2}{2\sigma_g^2}\right). \quad (4.2)$$

The equation is equal to the Equation 3.2 from Chapter 3. Again,  $\sigma_g$  is a user parameter that describes the standard deviation of matching geodesic distances, thus modeling how “non-isometric” the model can be, either due to numerical imprecision or due to imperfections of the pieces of input geometry themselves. In order to obtain the final sampling probability, this likelihood is multiplied with  $P_{descr}$  to account for local descriptor matches, giving the sampling distribution:

$$P_{samp}(\mathbf{x}, \mathbf{y} | \mathbf{x}_1, \mathbf{y}_1, \dots, \mathbf{x}_k, \mathbf{y}_k) = P_{descr}(\mathbf{x}, \mathbf{y}) \cdot P_{dist}(\mathbf{x}, \mathbf{y} | \mathbf{x}_1, \mathbf{y}_1, \dots, \mathbf{x}_k, \mathbf{y}_k). \quad (4.3)$$

Sampling is continued until no more matching pairs are found that do not exceed the maximum error threshold or until all points are matched. Differently to the previous approach we will use plan points instead of feature points, as explained in the next section, to limit the number of samples.

In the following, we refer to a set of matched points  $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_k, \mathbf{y}_k)\}$  as a solution set  $\mathcal{R}$ . This set sparsely models an isometric deformation  $\hat{f}_{\mathcal{R}}$  for the complete shape  $M_S$ . In order to estimate dense correspondences, Equation 4.2 is evaluated with the computed correspondences  $\mathcal{R}$  as previous matches. For every point  $\mathbf{x} \in M_S$ , we compute the probability distribution  $P_{dist}(\mathbf{x}, \mathbf{y} | \mathcal{R})$  on all  $\mathbf{y}$  of  $M_T$ . Inspired by [2, 41, 55], the point  $\mathbf{y}$  yielding the maximum likelihood is chosen as a dense match to  $\mathbf{x}$ .

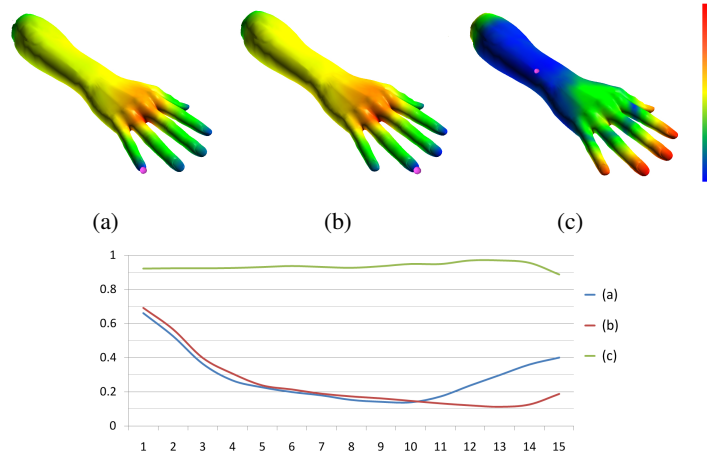


Figure 4.3: *Top row*: descriptor differences - blue is zero and red is the maximum observed difference. *Bottom row*: descriptor signature: intrinsic distance  $\rho$  is mapped to the x-axis and the normed length of the resulting isocurve to the y-axis. The curves corresponds to the points shown in the top row. The points at the two fingertips have similar curves up to the radius of  $\rho = 10$ . The signature on the arm is, however, entirely different.

To compare different solution sets in their ability to characterize the deformation, we evaluate the corresponding deformation function  $\hat{f}_{\mathcal{R}}$  by computing  $E(\hat{f}_{\mathcal{R}})$ , Equation 1.2. We do this by using the dense correspondence estimates described above. The solution score is then computed as:

$$E(\hat{f}_{\mathcal{R}}) = \frac{E(\hat{f})}{|\mathcal{R}|},$$

where  $E(\hat{f})$  is the deformation error, Equation 1.2.

Intuitively, the score  $E(\hat{f}_{\mathcal{R}})$  measures the average error introduced by the isometry represented by the solution set. The whole sampling scheme is iterated until either a solution set with a score below a given threshold  $c$  is found (typical value  $1.5\epsilon \leq c \leq 3.0\epsilon$ ) or the maximum number of trials (typically 100) is reached.

### 4.3 The planned sampling algorithm

So far the basic matching algorithm is equal to the approach described in Chapter 3 and now we can start to design a planning scheme that executes the algorithm more

rapidly by avoiding uninformative samples. We call our strategy “PLANSAC”: PLANned SAmple Consensus, referring to the fact that the algorithm plans ahead in terms of what samples are best to test in order to gain information as quickly as possible. The key idea is to look at  $P_{samp}$  (Equation 4.3) and try to make the distribution as clearly determined as possible by using a semi-solution set  $\{(\mathbf{x}_1^p, \cdot), \dots, (\mathbf{x}_K^p, \cdot)\}$  that we refer to as a *plan* for the matching step. The plan points  $\mathbf{x}_i^p$  are computed such as to maximize the information gained by matching them against a target shape. In order to measure the information content, we look at the *entropy* [122] of the final distribution.

Entropy is a scalar value that represents a statistical measure of the randomness of a discrete random variable  $Y$  with possible values  $\{y_1, \dots, y_n\}$  and probability distribution  $p$ . It is defined as:

$$H_p(Y) = \sum_{i=1}^n p(y_i) \log \frac{1}{p(y_i)}, \quad (4.4)$$

where  $p(y_i)$  is the probability of  $y_i$  and we define  $p \log p := 0$  for  $p = 0$ . Intuitively, entropy can be understood as the amount of information contained in a random process described by the probability function. The entropy is maximal for a uniformly distributed probability. In this case we cannot take any advantage of the model described by the random process since every output of the process is equally likely. For a probability function with very few peaks, on the other hand, the entropy takes a low value. In this case the distribution carries a high amount of information. In our case, this means that the correspondences are known more precisely.

Our goal is now to find a set of points  $\mathbf{X}^p = \{\mathbf{x}_i^p | \mathbf{x}_i^p \in M_S, i = 1..K\}$  that we use as a plan for a matching step. We begin with an empty set  $\mathbf{X}^p$  and first add a most *discriminative* point  $\mathbf{x}_k \in M_S$ : This a point with a descriptor [140] that leads to a minimal entropy when compared to all other points. We define a descriptor entropy of a point  $\mathbf{x}_k \in M_S$  as  $H_{p_k^d}(M_S)$ . The probability function  $p_k^d$  used for the entropy computation we define as:

$$p_k^d(\mathbf{y}) = P_{descr}(\mathbf{x}_k, \mathbf{y}). \quad (4.5)$$

It computes the probability of a given point  $\mathbf{x}_k$  to be equal to any other point  $\mathbf{y} \in M_S$ . Obviously, a point  $\mathbf{x}_k$  that has a very dissimilar descriptor to other points has a low entropy  $H_{p_k^d}(M_S)$ . In other words, the chosen point is most discriminative among all others, see Figure 4.4 for an example.

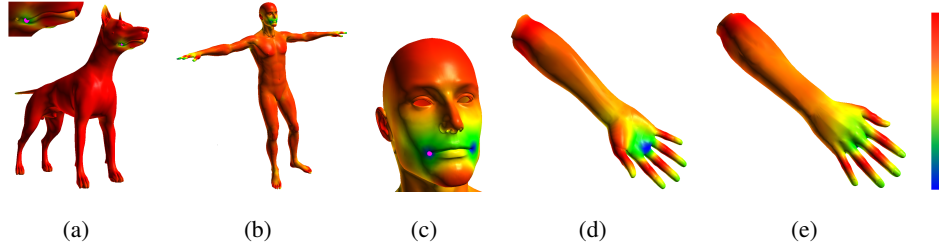


Figure 4.4: Entropy distribution of the most descriptive points, see Section 4.3. Regions with minimal entropy are marked blue, regions with maximal entropy are shown in red. (a), (b) The most descriptive point for the men and dog data sets is in the region of the corner of the mouth. (c) Zoomed view on human’s head with most descriptive point shown in purple. (d), (e) Hand data set: the most descriptive point is on the palm in the area of the middle finger.

Having selected the first point, we start to add points to  $\mathbf{X}^p$  by searching for points with most information content with respect to both descriptor matching as well as distances to previous points. We model this by checking how well shape  $\mathcal{M}_S$  matches onto itself. We do this by assuming a point  $\mathbf{x}_k \in M_S$  is the one we look for, i.e.  $\mathbf{X}_k^p = \mathbf{X}^p \cup \{\mathbf{x}_k\}$  and computing the matching entropy  $H_{p_k^m}(M_S \times M_S)$  of all possible correspondence pairs  $(\mathbf{x}, \mathbf{y}) \in M_S \times M_S$ . The probability function  $p_k^m$ , describes how well a particular pair  $(\mathbf{x}, \mathbf{y})$  is localized by the current candidate plan set  $\mathbf{X}_k^p$  and is set to:

$$p_k^m(\mathbf{x}, \mathbf{y}) = P_{dist}(\mathbf{x}, \mathbf{y} | \mathbf{X}_k^{p'}), \quad (4.6)$$

where  $\mathbf{X}_k^{p'}$  is a set of trivial matching pairs, i.e.  $\mathbf{X}_k^{p'} = \{(\mathbf{x}_i^p, \mathbf{x}_i^p) | \mathbf{x}_i^p \in \mathbf{X}_k^p\}$ . The trivial matching pairs model the self-matching of shape  $\mathcal{M}_S$ . Given that, we compute:

$$\arg \min_k \left( H_{p_k^m} + \lambda_d H_{p_k^d} \right), \quad (4.7)$$

where  $\lambda_d$  is a factor modeling how much of the descriptor entropy should be considered when searching for optimal points. A typical value is  $\lambda_d = 0.1$ . The additional descriptor entropy models the choice on the next more discriminative  $\mathbf{x}_k$  if the choice is intrinsically ambiguous. The point  $\mathbf{x}_k$  which minimizes the sum of entropies is added to our sampling set  $\mathbf{X}^p$ .

The model of matching  $\mathcal{M}_S$  to itself is supported by the fact that we have assumed that the source surface is isometric to any target surface. Therefore, and

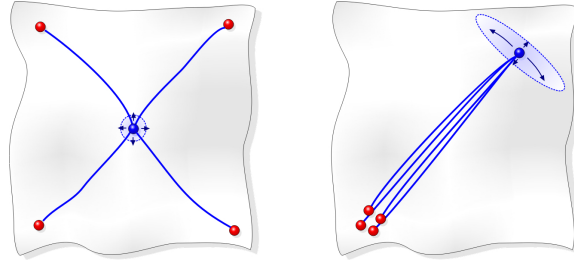


Figure 4.5: Blue regions schematically indicate the area of uncertainty given noisy geodesic distances. Left: the subset of red points defines the parametrization of the shape well. Right: A poorly chosen subset increases the uncertainty in the parametrization.

since we are using a descriptor which is completely intrinsic, the plan developed by our planning algorithm is not affected by isometric deformations.

Finally, we stop building the plan if all remaining points are fixed in their position and can be located precisely by their intrinsic distance to all points of  $\mathbf{X}^p$ . Hence, for every remaining point  $\mathbf{x}_i \in M_S$  we estimate the variance of  $P_{dist}(\mathbf{x}_i, \mathbf{y} | \mathbf{X}^p)$  over all  $\mathbf{y} \in M_S$ . The variance is computed using weighted principal component analysis (PCA) [61]: We assign a weight  $w_i(\mathbf{y}) = P_{dist}(\mathbf{x}_i, \mathbf{y} | \mathbf{X}^p)$  to every point  $\mathbf{y}$  and perform a PCA. The largest eigenvalue describes the maximum radius of the uncertainty in the localization, see Figure 4.5. If the maximal uncertainty radius for all source points  $\mathbf{x}$  is smaller than a specified threshold  $c \cdot \epsilon$ , typically  $c = 2$ , we stop adding plan points. We can now guarantee that our plan can find at least one isometric solution where every point is determined up to a standard deviation of  $c \cdot \epsilon$ .

It should also be noted that this incremental algorithm is not optimal; the best choices could be made by looking at all  $K$ -tuples simultaneously. However, computing this requires  $\mathcal{O}(n_s^K)$  time, which is very expensive even for moderate numbers of plan points  $K$ . Despite the lack of guaranteed optimality, our greedy approximation yields very good results in practice.

As we will see in our experiments, Section 4.4, our planning algorithm prefers points widely spread over the surface. This behaviour is similar to the well known *Farthest Point Sampling* (FPS) [91] which can be thought as a baseline method to compare to. However, in contrast to FPS, our planning step guarantees, up to a certain localization error, that the landmarks chosen by our algorithm are numer-

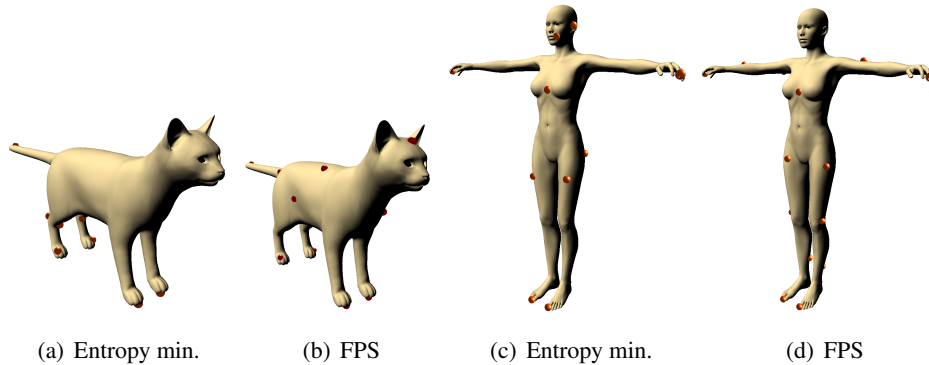


Figure 4.6: (a) and (b): Distribution of landmark points for entropy minimization criterion (10 points) and FPS (13 points) for the cat dataset. (c) and (d): Distribution of 13 landmark points computed with our approach and 18 points with FPS for the Victoria dataset. The sets of landmarks localize every point on the surface up-to  $2\epsilon$  in each of the experiments.

ically optimal with respect to isometric matching. In other words our algorithm spreads the landmarks over the surface in such a way that it minimizes the entropy of the matching uncertainty, while FPS is distributing points evenly over the surface. For simple shapes like a sphere both approaches yield similar results, however for more complex shapes the entropy minimization approach computes a smaller set of landmark points than FPS. Figure 4.6 shows a comparison of the landmark point distribution generated by our approach and by farthest point sampling.

## 4.4 Results

We evaluate our algorithm by performing shape matches on a number of benchmark models. For the evaluation, we use the standard TOSCA [23] data set. The data set consists of triangle meshes with approximately 50,000 vertices each. In addition, we have constructed a few further synthetic benchmark scenes to examine specific properties of our algorithm.

For efficiency reasons we down-sample the input meshes. Source shapes have a minimum point spacing of 4-5% of the longest bounding box size. For the target shapes, we use a higher resolution of around 1-2% of the bounding box size to provide sufficient numerical degrees of freedom for the match. The intrinsic wave descriptor is evaluated on the down-sampled source points however propagating



data set	$ \mathbf{X}^P $	$ M_S $	$ M_T $	$\#it$	$E(\hat{f}_{\mathcal{R}})$	$t_p$	$t_m$
hand	5	164	797	15	$1.25\epsilon$	175	0.658
dog	9	182	1341	2	$1.5\epsilon$	432	0.088
			1522	8	$2.0\epsilon$		0.340
			1394	12			0.572
cat	7	183	1221	8	$2.5\epsilon$	282	0.298
			1274	10			0.467
			1355	15			0.765
centaur	10	211	1055	12	$1.6\epsilon$	320	0.731
			909	16			0.853
			998	30			1.830
seahorse	9	126	572	1	$1.0\epsilon$	19	0.060
			550	1			0.045
			549	2			$1.25\epsilon$

Table 4.1: Results summarized for different data sets.  $|\mathbf{X}^P|$  is the number of points in the sampling plan,  $|M_T|$  is the number of points on the target shape for different poses (order as in Figure 4.12).  $\#it$  is the number of iterations required to find a matching solution with an error less than or equal to  $E(\hat{f}_{\mathcal{R}})$  averaged over 50 PLANSAC runs.  $t_p$  is the time required to build a plan set  $\mathbf{X}^P$ ,  $t_m$  is the average overall matching time. Timings are given in seconds for an Intel Core-2 Xenon 2.8 GHz platform (plan-build step was parallelized over 3 cores).

waves over the full resolution model. This preserves the *descriptiveness* of the down-sampled points. In the following, we refer to this target mesh sample spacing as  $\epsilon$  and specify all accuracy results with respect to this number.

**Pairwise matching results:** Table 4.1 summarizes the matching results and timings on the employed data sets, please refer to it for the following discussion. The dog data set, Figures 4.12(a)-4.12(d), represents a case of very descriptive geometry. Note that the average number of iterations required to find a solution is very low, since here we benefit from well-matched descriptors. Figure 4.12(d) shows a case where the deformation in the input data does not preserve isometric distances well. Our algorithm, however, is still able to find correct correspondences. Figures 4.12(e)-4.12(h), cat data-set, show a case where the shape has a low descriptive information. The number of iterations increases as does the localization error of the solution. The centaur data set, Figures 4.12(i)-4.12(l), is the most complex shape that we encountered in our tests. This also shows in the number of iterations that are required to obtain a suitable match. The seahorse data set, Fig-

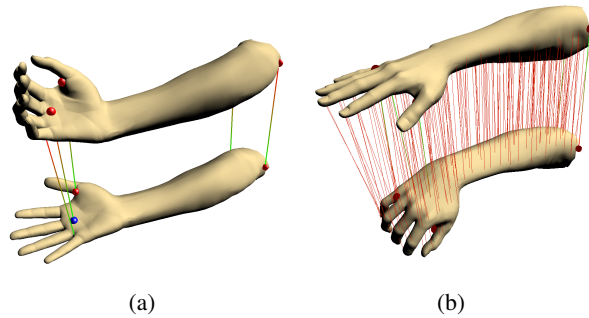


Figure 4.7: (a) Matching results of the hand data set. The most discriminative point is colored blue. (b) Dense correspondences. The solution error threshold is  $1.5\epsilon$ .

ures 4.12(m)-4.12(p), on the other hand, has very well distributed and descriptive features that explain most of the geometry very well. This, in conjunction with the relatively low resolution, which is due to the low resolution of the initial model from the data base, enables very fast matching results. Our algorithm is able to find the solution in only a single iteration most of the time. In almost all shapes the most discriminative point was found in the area of the mouth corner. This is due to the fact that these shape models contain throat geometry which is itself very descriptive. Another observation is that landmarks are placed preferably on the shape's extremities.

Figure 4.8 shows our synthetic data-sets. Please note that all used shapes are featureless, however since our technique neither relies on intrinsic nor extrinsic features, we are still able to compute a valid plan set and can guarantee to find a valid isometry.

**Examining the Practical Matching Complexity:** In the beginning of this chapter we posed the question of matching complexity in the case of numerical ill-conditioning, e.g. due to approximate geodesics. We would like to provide a partial answer to this question from a practical point of view. Figure 4.8 shows a data set consisting of one, two and three spheres connected through a narrow tube. In theory the number of computed plan points required to isometrically parametrize a sphere is three, which we also found numerically as shown in Figure 4.8(a). For two connected spheres, however, not any 3-tuple of points is sufficient to fix an isometry in practice even through the topology it is still equivalent to a sphere. Fixing intrinsic distances on one piece does not constrain the matching on the second piece in practice. It follows that for a numerically stable match more plan

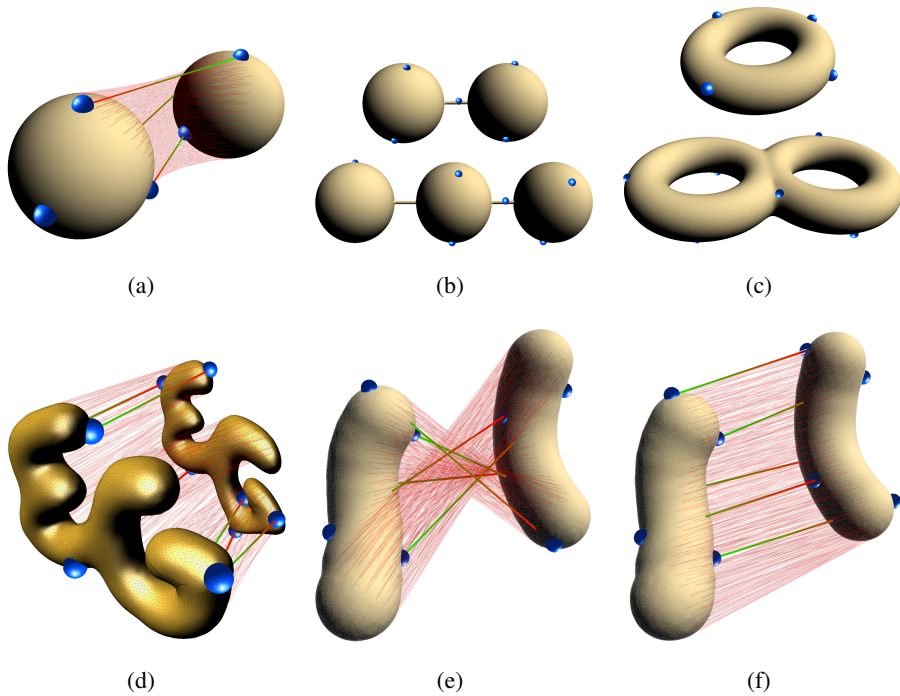


Figure 4.8: A number of featureless objects which can be matched with our method. (a) Match of a simple sphere against itself. Note that the compute plan as numerically chosen by our algorithm consists of 3 points and is thus consistent with theoretical results [80]. (b) Set of plan points  $\mathbf{X}^p$  (marked blue) detected for synthetic data sets consisting of two and three spheres connected by a thin tube. The number of required plan points for stable matching is 5 and 7 respectively. (c) shows the plan points for the torus (5, one invisible) and the double torus (6). (d) EG-Logo as a meta-blob matched against a slightly deformed version of itself. The matching error is  $2\epsilon$  or 8% of the bounding box size. (e) Synthetic peanut data set consisting of slightly deformed meta-blobs. The solution error is  $1.5\epsilon$  or 3% of the bounding box size. Although the solution appears incorrect, we have an intrinsically correct match. Extrinsically, the object was flipped along the horizontal axis and mirrored along the vertical axis. (f) Extrinsically plausible match for the peanut data set.

points are needed. As seen in Figure 4.8(b) our algorithm automatically identifies this instability and computes two additional points on every piece of geometry. Our experiments show that numerical ill-conditioning can increase the practical matching complexity towards the case of two unconnected objects, where 6 plan points would be required, three for each sphere.

A similar observation can be made for surfaces of higher genus. We tested our algorithm on a torus and double-torus shape as well. The required plan points are 4-5 for the torus, where 5 points lead to more stable matching results. The double-torus requires at least 6 plan points. In practice, we also observe differences between shapes: For example, the centaur requires 10 landmark points to guarantee a matching of reasonable accuracy while the cat requires only 7 to achieve the same accuracy.

**Comparison to previous work:** We compare our method to the approach presented in the previous chapter as also to a state-of-the-art feature-based technique, spectral matching [74]. For fair comparison, the RANSAC method, Chapter 3, is used with the same maximum number of iterations (100) as the new plan-based algorithm. We compare the performance on a subset of the models from Table 4.1 (seahorse and centaur). Figure 4.9 shows the correspondences computed on the first pose of the seahorse and centaur data sets. Since both feature-based techniques do not provide dense correspondences, we compute their matching scores  $E(\hat{f}_R)$  on the sparse correspondence set only. As can be seen, RANSAC outperforms spectral matching in the number of computed correspondences, while it is unclear which method provides a better matching accuracy. Note however, that our proposed PLANSAC scheme has a significantly higher accuracy of up to 4 times while computing a dense set as compared to only sparse matches in the competing methods.

Additionally, if only the matching phase is considered, i.e. all pre-processing is excluded, our technique is able to outperform both algorithms in computation time as well. The computation time required by spectral matching is 1.2 seconds for the seahorse and around 45 minutes for the centaur data set. For RANSAC, these numbers are 2.5 seconds and around 22 minutes, respectively. In contrast, once the plan is known, our method dramatically reduces computation cost to a fraction of the time required by both techniques, i.e. 70ms for the seahorse data set and 740ms for the centaur data set.

**Application Scenarios:** As our first scenario we consider a sequence matching

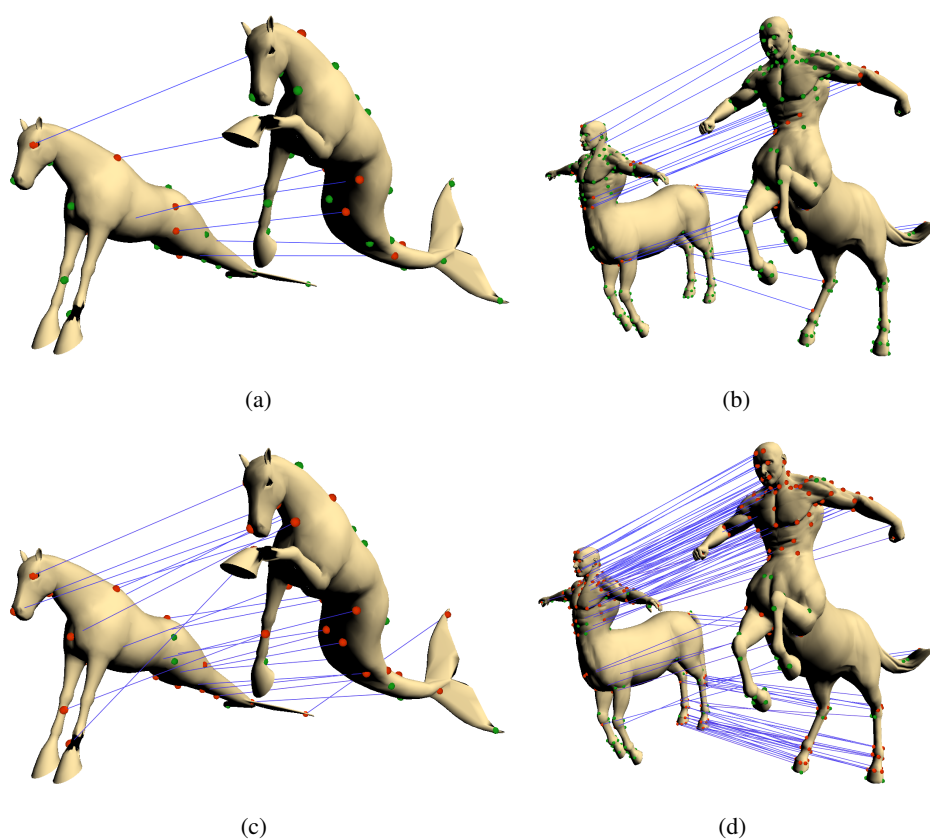


Figure 4.9: (a) and (b) show the matching for the first pose of the seahorse and centaur data sets computed with spectral matching [74]. The localization error  $E(\hat{f}_{\mathcal{R}})$  is  $6.5\epsilon$  (7 correspondences) and  $4.5\epsilon$  (26 correspondences) respectively. (c) and (d) shows the matching computed with RANSAC matching [141]. The localization errors of the RANSAC results are  $4.3\epsilon$  (22 correspondences) and  $5.6\epsilon$  (121 correspondences). In contrast, our algorithm achieves a localization error of only  $1.0\epsilon$  (seahorse) and  $1.6\epsilon$  (centaur) while matching the shapes densely, see Table 4.1. The corresponding plan matches are shown in Figure 4.12 (j) and (n).

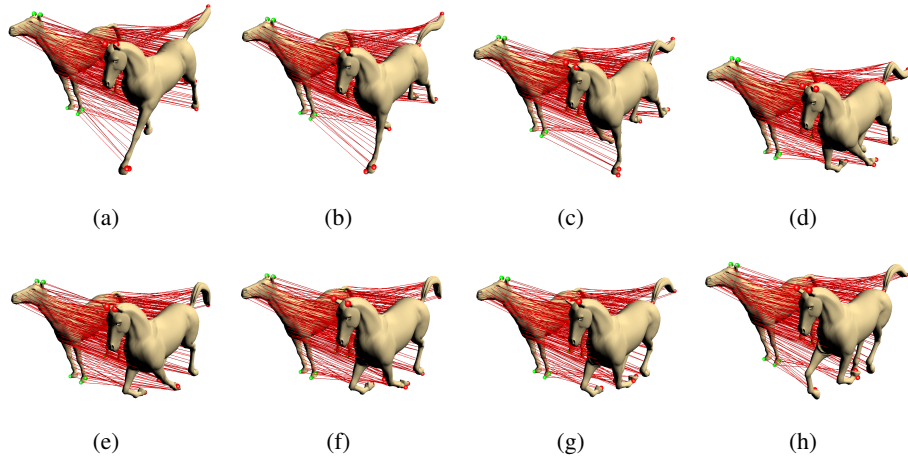


Figure 4.10: (a)-(h) Dense correspondences shown for the galloping horse data set [134]. The first 8 out of 50 frames of the data set are shown. The computation time is 52 seconds for 50 frames, the average number of PLANSAC iterations required to match the template against one frame is 4.

application. Figure 4.10 shows a running horse sequence [134] where the dense correspondences are computed by matching the first frame against all other frames. The matching of one frame against all 50 frames only takes about 52 seconds, i.e. about 1 second per frame. Please note that we do not perform any extrinsic check if a computed match is one of the symmetric solutions, therefore the solution set also contains symmetrically equivalent correspondence matches. The median average number of required PLANSAC iterations is one, their mean number is 4. There are only four frames where more than 10 iterations are required.

As a second scenario we consider a database-like query application. For this scenario we construct a small database consisting of all models in all poses shown in Figure 4.12 normalized to exhibit a uniform mean intrinsic distance. Without the latter, the task would be trivial since the models exhibit widely different scales and are thus distinguishable by their mean intrinsic distance alone. Figure 4.11 shows the results of our initial experiments. To perform the query, we match the query shape against all shapes in the database. The matches are then sorted by their matching score  $E(\hat{f}_{\mathcal{R}})$ . The probability of the best scored shape being the correct one is greater than 95% (here we ignore the match of the given template to itself) over 100 trials. We emphasize that these results are preliminary and only hint at the potential use of our algorithm in database applications.

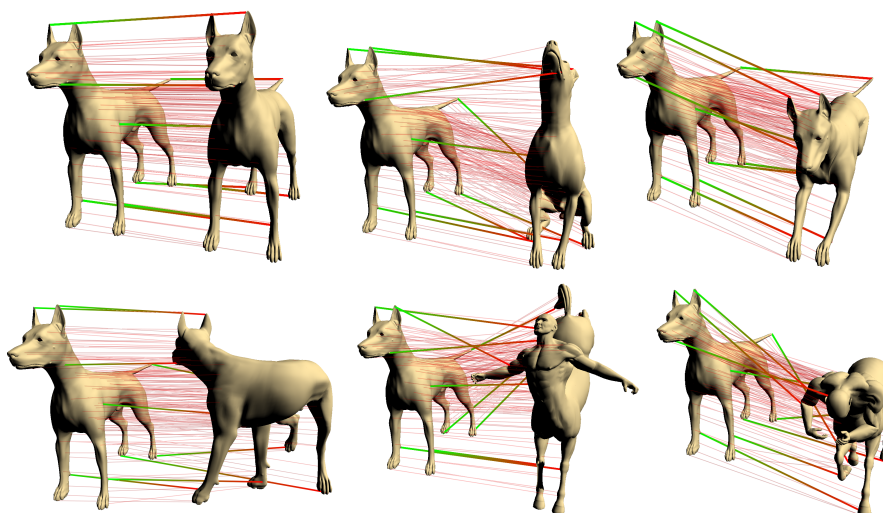


Figure 4.11: Results of an exemplary database query. Results are sorted (from left to right) by their matching score  $E(\hat{f}_{\mathcal{R}})$ .

## 4.5 Discussion

In this chapter we developed a shape matching algorithm based on the novel concept of entropy-based planned random sampling. It enables very fast and reliable matching between similar or different shapes, while simultaneously removing the need for explicit feature detection. Moreover, the algorithm automatically adapts to the input characteristics and chooses an optimized sampling strategy for any given object. Numerical experiments hint at the possibility of characterizing a shape's intrinsic complexity via the number of plan points necessary to parametrize its surface in a stable way. In general, we believe that computing a most optimal matching plan can answer the more general question of: "What is the intrinsic complexity of a shape?"

Although we based our approach on the algorithm developed in Chapter 3, we have not taken into account data containing topology noise. Thus to apply the entropy-based matching algorithm to real-world scanner data we still require additional work. In the next chapter we will take a look of applying the knowledge gathered in previous two chapters to use it for matching multiple frames of scanned data.

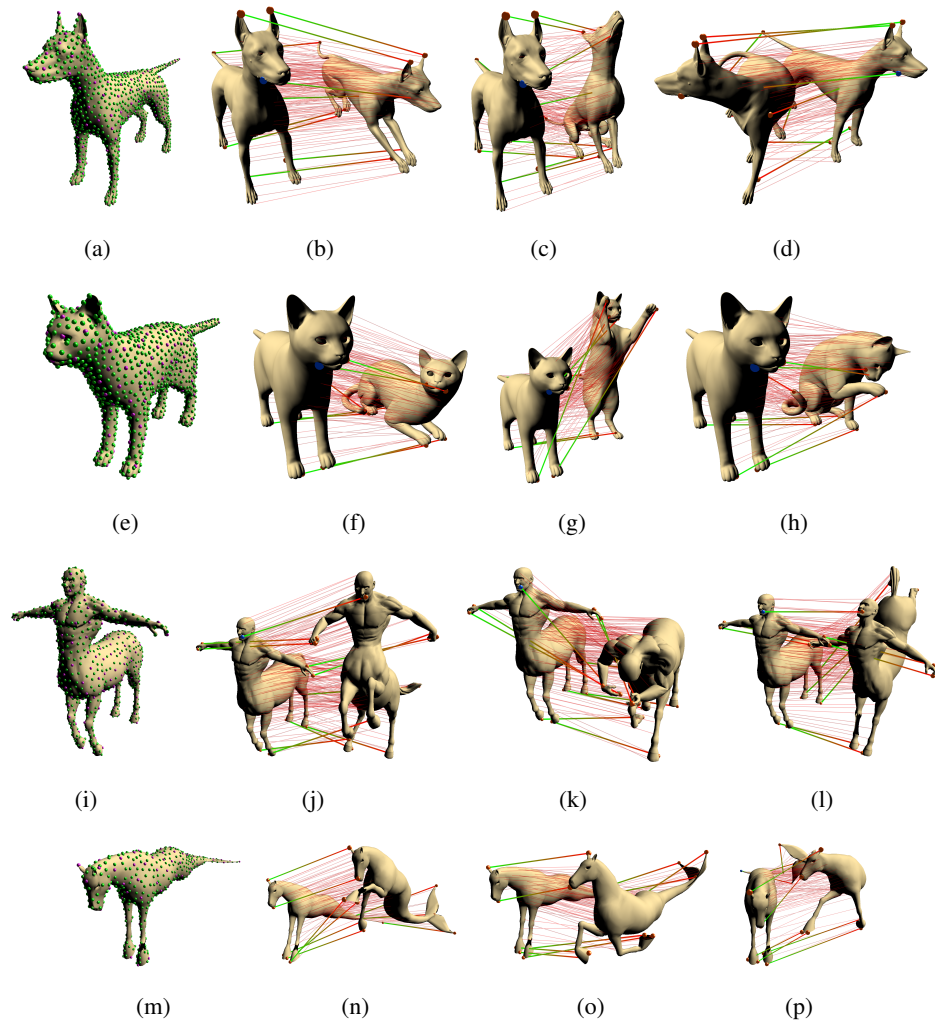


Figure 4.12: Matching results for data sets in different poses. (a), (e), (i) and (m) show the discretization of the data. Purple points mark the discretization of the source shape, green points that of the target shape (in the pose of the source shape). The order of the poses is arranged in order of increasing numerical matching complexity, i.e. the average number of iterations required to find a solution.



---



## Animation Reconstruction

---

The two previous chapters investigated a randomized forward search algorithm for matching two shapes. Additionally, we introduced the concept of landmark coordinates. In this chapter we will combine these two ideas in order to propose a solution for the problem of animation reconstruction.

Animation reconstruction is a reconstruction of the shape and motion of a deformable object from dynamic 3D scanner data, without using user provided template models. Unlike previous work that addressed this problem, we do not rely on locally convergent optimization but develop a system that can handle fast motion, temporally disrupted input, and can correctly match objects that disappear for extended time periods in acquisition holes due to occlusion. To be able to do so, we incorporate the algorithm developed in Chapter 3 which enables us to find a matching between two shapes which underwent strong (isometric) deformation.

We first estimate a few landmark correspondences, which are extended to a dense matching and then used to reconstruct geometry and motion. In order to achieve our goal we propose a number of algorithmic building blocks: a scheme for tracking landmarks in temporally coherent and incoherent data, an algorithm for robust estimation of dense correspondences under topological noise, and the integration of local matching techniques to refine the result. We describe and evaluate the individual components and propose a complete animation reconstruction pipeline based on these ideas. We evaluate our method on a number of standard benchmark data sets and show that we can obtain correct reconstructions in situations where other techniques fail completely or require additional user guidance such as a template model.

Some techniques have been proposed already to solve the problem of animation reconstruction [87, 107, 138, 152, 153]. However, these approaches employ local numerical optimization to align parts of the object incrementally: The final

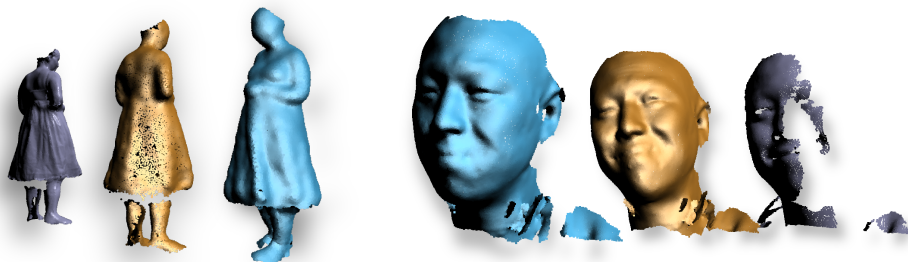


Figure 5.1: Animation reconstruction recovers template model (blue) as well as its motion over time (yellow) from dynamic point cloud data containing topological noise (gray).

shape is inferred by a deformable alignment of the geometry in time sequence order. If some of the alignments yield an incorrect result, neither the shape of the deformable object nor the correspondences are reconstructed correctly. In practice, alignment problems are frequently observed. They are caused by fast object movement or vanishing geometry that reappears in later frames in a different pose. Local alignment is not able to handle these situations correctly. The problem obviously becomes much easier if the user provides additional information, such as a template model [7, 29, 35, 75, 102, 120, 159]. Nevertheless, numerical tracking can still fail so that manual user intervention becomes necessary. Furthermore, the fixed template restricts the expressiveness of the model, prohibits topology changes, and makes an acquisition of general scenes tedious.

The approach presented in this chapter, however, is a template-free technique that is able to assemble shapes from partial scans more robustly and under more general motion than previous methods. The main idea is motivated by cartography, where any point on a map is measured with respect to a relative distance to a landmark. We first track the location of a few landmark points, which we subsequently use to compute dense correspondences, assuming that the deformation of the object is approximately isometric. The output of the algorithm is a *chart* that covers the complete original object. It encodes the intrinsic structure of the reconstructed manifold and dense correspondences to the data points. This intrinsic reconstruction does not yet provide concrete geometry. Therefore, we combine the intrinsic manifold charting with a state-of-the-art extrinsic reconstruction scheme [152] that computes actual geometry. By initializing this local numerical optimization

scheme with charted correspondences, we obtain much more reliable results.

In order to perform the charting, a number of algorithmic building blocks are necessary, each of which represents a novel contribution: First, we propose a scheme to track salient landmark points. The algorithm automatically detects temporal discontinuities and resorts to a global feature matching algorithm of Chapter 3 to provide landmark correspondences also in general settings. The second component is the intrinsic charting algorithm that extends the sparse landmark correspondences to dense matches and stitches together partially overlapping charts. Finally, we design a matching pipeline that iteratively performs tracking and chart merging to chart complete animation sequences.

A key challenge in all three steps is that we have to deal with partial data, due to occlusion artifacts. Therefore, intrinsic distances are not reliable. Similarly, the apparent topology of the input data might change, for example if a person temporarily rests his hands touching the body. We account for these problems by employing a novel robust matching model, which can handle such topological noise and furthermore quantify the uncertainty under noisy input.

We describe and evaluate the separate building blocks of the algorithm as well as a complete animation reconstruction pipeline that is composed of these components. In experiments with well known benchmark data sets, we show that the new reconstruction pipeline can handle more general input data than previous work.

In summary, the main contributions made in this chapter, are:

- A matching model that is robust to topological noise and that can quantify the matching uncertainty.
- A landmark tracking algorithm that establishes sparse correspondences fully automatically under both temporally coherent as well as arbitrary, abrupt motion.
- A charting algorithm that computes dense correspondences from sparse landmark tracks, thereby assembling multiple partial charts into one common reconstruction.
- Finally, a complete animation reconstruction pipeline that is significantly more robust than previous techniques. In particular, it can, for the first time, handle abrupt motion and occluded objects that reappear in very different pose without user input.

## 5.1 Overview

### 5.1.1 Problem Statement

Let us first define the problem we are solving and describe the input data we are expecting.

**Input data assumptions:** The goal of our method is to reconstruct a manifold and its motion from partial observations. Formally, we assume that there has been an original  $\mathcal{M}$  that underwent a time-variant motion  $f_t : \mathcal{M} \rightarrow \mathbb{R}^3$ .  $t \in [1, T]$  is the time parameter. Each  $f_t$  is assumed to be injective and differentiable, and each  $f_t(\mathcal{M})$  denotes a deformed version  $\mathcal{M}^t$  of the original manifold  $\mathcal{M}$ .

As already stated in the previous chapters we equip manifolds with an intrinsic metric  $d_{\mathcal{M}}(\cdot, \cdot)$  that measures the shortest geodesic distance between pairs of points. We assume that the deformation  $f_t$  is approximately isometric for each fixed  $t$ . This means that

$$\forall t \in \{1..T\} : d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) - d_{f_t(\mathcal{M})}(f_t(\mathbf{x}), f_t(\mathbf{y})) \leq \eta_{\sigma_f}, \quad (5.1)$$

where  $\eta_{\sigma_f} \sim N(0, \sigma_f)$  is an error that is normally distributed with standard deviation  $\sigma_f$  and mean zero. In other words, we assume that the original deformation, even before measurement, has not been perfectly isometric but that there might have been errors that are in the range of  $\sigma_f$ .

A 3D scanner only yields a partial, sampled representation. We assume that the scanner operates at regular time steps  $t \in \{1, 2, \dots, T\}$  and for each time step, yields a finite set of sample points  $D_t \subset \mathbb{R}^3$ . We denote the individual points by  $\mathbf{x}_t^{(i)}, i = 1, \dots, n_t$  and the collection of all input data by just  $D$ . To simplify further processing, we assume that parts of objects that have actually been acquired have been sampled with a sample spacing of at most  $\epsilon_s$ . Areas with lower sampling density are discarded during preprocessing. Furthermore, we assume that all of  $\mathcal{M}$  at some point has been observed with sufficient sampling density (or equivalently, we only try to reconstruct what we have observed).

Additionally, we build a  $k$ -neighborhood graph  $G = (M, E)$  to approximately encode the metric of each  $\mathcal{M}^t$ , see Chapter 1. The graph distance  $d_{M^t}(\cdot, \cdot)$  (i.e., the shortest path in the graph) between two arbitrary points serves as an approximation of the original geodesic distance  $d_{\mathcal{M}^t}(\cdot, \cdot)$  and is computed by the shortest path algorithm of Dijkstra [37].

Obviously, the discrete approximation will distort the distance measure.

Smooth geodesics are approximated by zig-zag paths in graphs, which introduces systematic deviations. However, we use this representation consistently for data and all (partial) reconstructions. The systematic errors therefore affect all geodesic paths in the same way so that they remain directly comparable, which is sufficient for our application. Nevertheless, the discretization also causes additional quantization noise in distance estimates. Accordingly, we adapt the expected error of intrinsic distances  $\sigma_f$  to be at least in the range of  $\epsilon_s$  (in practice, we use  $3\epsilon_s$ ). Hence,  $\sigma_f$  in the following describes the magnitude of both modeling as well as representation noise.

**Reconstruction Tasks:** We consider two reconstruction tasks: A *full geometric reconstruction* and the *reconstruction of a chart* of the data. The full geometric reconstruction is the ultimate goal: We want to reconstruct  $\mathcal{M}$  and  $f$ . Because of acquisition holes, this involves an interpolation of  $f$  in areas of missing data. Wand et al. [152, 153] propose a variational model that can find plausible interpolations by employing physically motivated prior assumptions on shape and motion. However, their model is non-linear and non-convex and cannot be globally optimized.

In order to compute a suitable initialization for such methods, we propose to perform a simpler reconstruction task first, the reconstruction of a chart of the data. Here, we only reconstruct the shape  $\mathcal{M}$  (up to isometries) and correspondences between  $\mathcal{M}$  and (most of) the data points  $D$ . This means, we either encode for each data point  $\mathbf{x}_t^{(i)} \in D$  its preimage  $f^{-1}(\mathbf{x}_t^{(i)}) \in \mathcal{M}$ , or mark it as unknown, in case the reconstruction was not able to interpret the data point. We can then use this shape and the correspondences to the data points as fixed boundary conditions to stabilize a locally convergent reconstruction. For efficiency, our actual pipeline will not compute explicit correspondences for each single data point but rather use a coarse cloud of correspondence samples that covers the data points in order to encode the correspondence information, as detailed in the next subsection.

### 5.1.2 Data Structures

**Sampled manifolds, extrinsic view:** Sometimes, we want to be able to give an embedding of a sampled manifold  $M$  in  $\mathbb{R}^3$ . This is trivial to encode - we just store for each graph node  $m_i \in M$  an additional position vector  $\mathbf{x}_0(m_i) \in \mathbb{R}^3$ . Following [152], we call this embedding of the chart an *urshape*. We denote the urshape of  $M$  by  $X_0(M)$ . Please note that urshapes are not unique but any isometric deformation  $f(X_0(M))$  is again a valid urshape.

**Charts:** A chart combines a (partial) reconstruction of a manifold with correspondences to data points. This means, a chart is a sampled manifold  $M$ , and for each  $m_i \in M$  we store a list of 3D positions of where node  $m_i$  would be located in each data frame. We denote these positions by  $\mathbf{x}_t(m_i) \in \mathbb{R}^3$ , where  $t$  covers a non-empty subset of time steps  $t \in 1, \dots, T$ . If the embedding is unknown at a specific time  $t$ , we mark  $\mathbf{x}_t(m_i)$  as unknown.

The definition of a chart has been chosen to account for a later technical problem: To limit computational costs, we will not be able to include every data point into the chart. Therefore, we allow for using a coarse set of points to represent  $M$  and store correspondences implicitly, by storing 3D positions for  $m_i \in M$ . Each  $m_i$  will form (partial) tracks that move over time but, in general, will not exactly coincide with data points but rather cover the data points. Please also note that a chart does not necessarily have a full geometric embedding, as the temporal coverage might be sparse and different at every node. However, as we will see later, our final pipeline will actually maintain a fully embedded urshape  $X_0(M)$  for every chart in order to interface with the extrinsic reconstruction.

**Landmark coordinates:** Some of the chart points are *landmark points*. These points are special as they correspond to features of the input data that we were able to recognize and track over time. As any other chart point, the spatial location of landmarks might not be known for the full time sequence  $t = 1..T$  but only for a (non-empty) subset. Given a set of landmarks  $L = \{l_1, \dots, l_n\} \subseteq M$ , we define the *landmark coordinates*  $\mathbf{d}_L(m)$  of an arbitrary node  $m \in M$  as the vector of intrinsic distances between  $m$  and the landmark points:

$$\mathbf{d}_M^{(L)}(m) = [d_M(l_1, m), \dots, d_M(l_n, m)]^T \quad (5.2)$$

The intrinsic distances are approximated by graph distances in  $M$ .

The main idea of our algorithm is that if two charts share a number of landmarks, we can compute dense correspondences for the remaining chart points by comparing landmark coordinates. As we have shown in Chapter 4 points of the same landmark coordinates are easily matchable. However, the main challenge here is to do this in a way that is robust to topological noise, unlike a trivial implementation as in Chapter 4. For this, we introduce a robust probabilistic matching model in the next subsection.

### 5.1.3 Robust Intrinsic Matching

A central problem of our approach is to determine correspondences between points from different charts. Let  $M_A$  and  $M_B$  be two charts that share a set  $L$  of landmarks for which we know the correspondences. We now want to compute where a point  $a \in M_A$  could correspond to in  $M_B$ . For this, we compute a probability distribution over all points of  $M_B$  that quantifies the likelihood of  $a \in M_A$  matching a point  $b \in M_B$  (denoted as  $a \sim b$ ):

$$\Pr(a \sim b|L) = \frac{1}{Z} \prod_{j=1}^{|L|} \left( \lambda \cdot e^{-\frac{(\mathbf{d}_{M_A}^{(L)}(a)[j] - \mathbf{d}_{M_B}^{(L)}(b)[j])^2}{2\sigma_j^2}} + (1 - \lambda) \right), \quad (5.3)$$

$\mathbf{d}_{M_A}^{(L)}(a)[j]$  and  $\mathbf{d}_{M_B}^{(L)}(b)[j]$  are the  $j$ th component of the landmark coordinate vector of  $a$  and  $b$ , respectively, and the term  $1/Z$  is just the normalization constant. The equation is related to Equation 3.2, however, in this case we incorporate a robustness component which ensures a non-zero probability distribution even in case of very bad configuration, i.e., if the exponent vanishes. Equation 5.3 models the matching problem by considering the geodesic distance to each landmark point independently. For each connection, we assume a normal-distributed error in case that the geodesic is correct. However, it may happen that acquisition holes or pseudo connections (“closing mouth”) distort the geodesics such that the distance is arbitrarily wrong. In this case, we do not have any information about the correct distance so that we resort to a uniform distribution. The parameter  $\lambda$  is the probability for geodesics being correct. We use a global constant failure probability of 10%, i.e.,  $\lambda = 0.9$ .

In practice, we can make the model more robust by limiting the product to take into account only nearby landmark points (in a geodesic sense) for each model point. Limiting the influence helps because the likelihood of geodesic path being distorted increases with distance to the point considered. In our implementation we use the 5 nearest landmarks.

This model is similar to the robust RANSAC approach presented in Chapter 3, where only the  $k$ -best geodesics are considered for matching. However, our new model provides some important improvements: It provides a continuous probability density that describes the likelihood of matching point  $a$  on  $M_B$ . If landmark points are chosen in a good configuration, the density is more sharply peaked than for landmark points in a bad configuration (with geodesics almost parallel, see

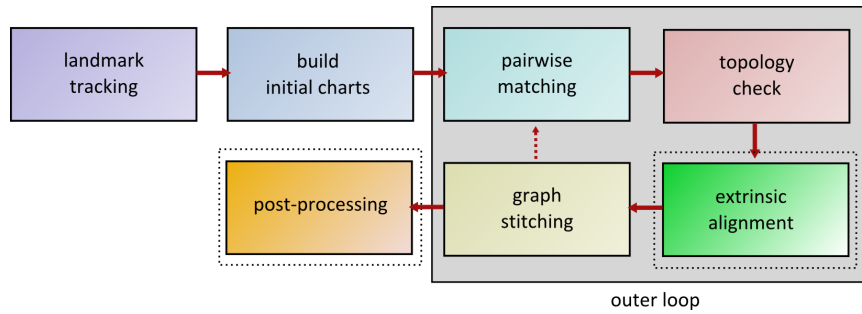


Figure 5.2: Overview of the animation cartography pipeline. Landmark tracks are used to build initial charts (i-charts), which are recursively combined in a pairwise chart merging loop. The final result is used to initialize a numerical bundle adjustment algorithm for post-processing. Dotted blocks indicate extrinsic matching components based on previous work of [152].

Figure 4.5 for example).

The probability density does not only encode the maximum likelihood match, but we have the complete distribution that quantifies the uncertainty. In particular, we examine the variance of  $\Pr(a \sim b|L)$  w.r.t. to  $\mathbf{x}_t(b)$  in order to determine how *certain* a match is. If the variance is high, the match is not reliable. Please note that the variance automatically increases if the outlier probability  $1 - \lambda$  increases. In this case, more correct landmark matches are required to reduce the variance again (the uniform density “floor” of the distribution converges to zero with  $(1 - \lambda)^{|L|}$ ). Furthermore, if the error in the normal distribution is large, combining multiple landmark correspondences reduces the variance because multiplying the Gaussians will lead to a more peaked distribution. Another important improvement over the previous model is that we do not need to fix a constant  $k$  of reliable geodesics but we can use the more natural formulation that geodesics have a certain failure probability; the resulting uncertainty is automatically taken into account, including the case that even some of the  $k$  best matching geodesics could be wrong.

#### 5.1.4 Pipeline Overview

The full animation reconstruction pipeline consists of a number of components. We will discuss each individual component separately in the next Sections (5.2, 5.3 and 5.4) and the composition of the full pipeline afterwards, in Section 5.5. Here, we give a brief overview for orientation (see also Figure 5.2).



The reconstruction starts by *landmark tracking*. In this step, the input data is examined for feature regions and a KLT-like tracking scheme [81], adapted to 3D geometry, is used to find landmark tracks. Out of these tracks, initial charts are built, which we refer to in short as *i-charts*. Our algorithm usually extracts a number of such *i-charts* that end when tracks discontinue due to abrupt motion or occlusion. Therefore, the next step matches disconnected *i-charts* to general partial charts of the animation, which we refer to in short as *p-charts*. The matching step involves an additional topology check to remove incorrect edges from the sampled manifolds and an extrinsic refinement step to improve the matching precision, followed by a graph stitching step that connects the two manifold representations (charts). This scheme is iterated in an outer loop until a full chart of the complete animation is obtained. Finally, we input the full chart as boundary conditions in a standard numerical optimization to obtain the final results; we use the method of [152].

In the following three sections we opt for an isolated discussion of each individual pipeline component for two reasons: First, it makes it easier to structure the rather complex reconstruction system. Second, several of the individual components might be useful as algorithmic primitives in other geometry processing contexts so that it is valuable to look at them separately.

## 5.2 Landmarks

Landmarks are the key concept for solving the reconstruction problem because they allow us to characterize dense correspondences between surfaces by fixing only a small number of landmark correspondences. This reduces the combinatorial complexity of the matching problem to a level that makes the reconstruction feasible.

### 5.2.1 Continuous Landmark Tracking

The first component of our reconstruction pipeline is a tracker for *continuous* landmark tracks. It gets the temporal sequence of raw data  $D$  from the scanner as input. The task is to (1) identify feature regions, (2) track features over time, and (3) recognize when tracks end due to incoherent motion.

We solve the first problem (1) by running slippage analysis [45]. It looks at every frame  $D_t$  of the data and determines for each point  $\mathbf{x}_t^{(i)}$  whether a region

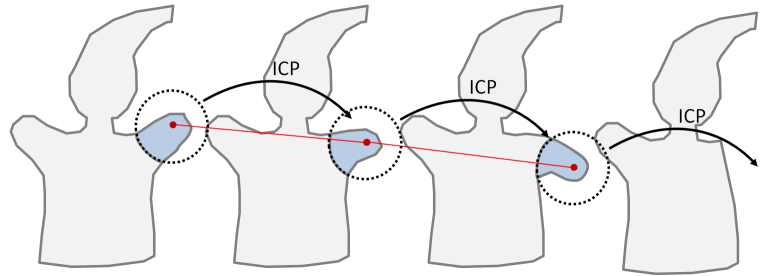


Figure 5.3: We align small patches of points to successive frames via ICP to generate tracks. A track is stopped when ICP fails to compute a stable result.

of radius  $r$  around  $\mathbf{x}_t^{(i)}$  can be stably aligned to itself under a rigid motion (in practice, we use  $r = 10\%$  of the bounding box size of the object). For flat areas, for example, the alignment is unstable because the patch could just slip along the plane. We keep only the unslippable regions and perform a coarse  $r$ -sampling to distribute *feature points* uniformly. Again, we use a Poisson-disc algorithm to obtain a good uniform distribution.

The main tracking step (2) is performed by simple rigid ICP (Figure 5.3): We extract the  $r$ -neighborhood of each feature point and align it to the next frame using point-to-plane ICP, always initialized to the (known) position of the previous frame. If the algorithm converges, we align the same geometry again to the next frame, and iterate until the alignment diverges. The landmark track is given by the trajectory of the center of the aligned region (the feature point) over time. Divergence is determined (3) by not converging to a fixed point within 32 iterations or by a translational motion by more than  $r$  within one frame (which is likely to be wrong, because there was no initial overlap of the geometry with the new target).

We start new tracks automatically: For each new frame, we recompute the non-slippable regions and insert a new point whenever it is not  $r$ -covered with feature points that are being tracked.

This scheme could be considered a geometric variant of the well-known KLT feature tracker for images [81]. It works quite effectively in our situation because scanned data usually contains a lot of coherent motion (but not everywhere) with small motion between frames. Locally, within a small spatial and temporal environment, the motion is usually almost rigid. Our scheme does not lead to perfect results but might create both false negatives and positives, which have to be handled by the robust matching model.

### 5.2.2 Connecting Broken Tracks

Usually, the tracking algorithm is not able to cover the whole animation sequence but rapid motion or occlusion disrupt some or all of the tracks. Therefore, we need an algorithm to reconnect broken tracks.

We consider two point clouds  $A, B \subset \mathbb{R}^3$ . They might already have a small number of landmark tracks  $L$  in common, but the set  $L$  can be empty. If a few continuous tracks are present, we include these as initialization, so that the algorithm finds the correct solution more quickly and more reliably. We now form candidate landmark correspondences by connecting each landmark node of  $A$  to every other node in  $B$  (here: landmarks and ordinary nodes). From this set, we have to find a consistent subset.

In order to extend the tracks which might have been stopped for example due to rapid motion, we employ the forward search algorithm as described in Chapter 3. For this we create a set of all possible candidate correspondences between the tracks on frame  $A$  and all other ordinary points on frame  $B$ . As mentioned before, the set of continuous tracks  $L$  acts as a initialization for the RANSAC-like algorithm. The whole forward search/RANSAC loop is iterated multiple times (typically, 100 trials), and the result with the largest number of established matches is used as the final result.

Figure 5.4 shows an example of rapid motion. Here, a very large temporal gap between two frames breaks the track continuation as described before. Employing the RANSAC-like search algorithm to continue the tracks continues the broken tracks to the other frames.

In principle, we could just always apply this algorithm to find landmark tracks, omitting the continuous tracking phase altogether. However, RANSAC-based matching might fail with a small probability. Therefore, several independent matching operations have success probability that declines exponentially with the number of matches. By making use of temporal coherence, we can make our algorithm substantially more robust, or in other words, dramatically reduce the computational costs (because the number of RANSAC-rounds would have to be increased exponentially to make up for this).

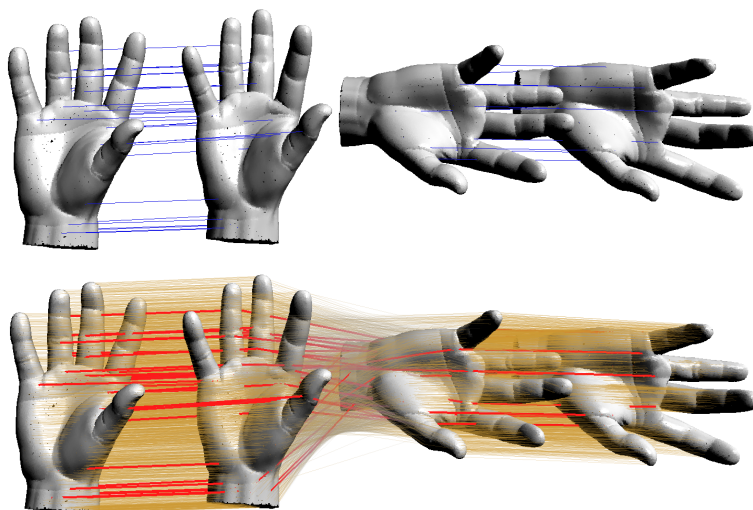


Figure 5.4: Applying landmark tracking to the hand data set (synthetic). Upper row: The blue tracks are obtained by continuous tracking; they end automatically at the abrupt turn in the middle. Bottom row: The situation is recognized automatically and the RANSAC algorithm establishes additional landmark correspondences. The orange lines indicate the final dense correspondences.

### 5.3 Intrinsic Charting

We now assume that we know landmark correspondences and turn our attention to the problem of establishing dense correspondences among charts, and subsequently merging these into compound reconstructions. We look at a number of different algorithmic steps: Creating a single frame chart from scratch (as initialization), merging two charts given landmarks, and checking the connectivity of merged charts. Afterwards, we use these more elementary algorithms to formulate higher level algorithms that build i-charts and p-charts.

#### 5.3.1 Building Single-Frame Charts

We build initial charts (i.e. just sampled manifolds) for a single frame directly from data  $D_t$ : In order to limit the computational costs, we resample  $D_t$  to an (Euclidean) sample spacing of  $\epsilon_s$ , using a Poisson-disc sampler. This yields the vertices of a sampled manifold  $M_t$ . Afterwards, we form the graph  $G_t$  by building a  $k$ -nearest neighbor graph ( $k$  is typically set to 20) on  $M_t$ , with respect to Euclidean distances. We then also use the Euclidean distance of the points as edge length. For

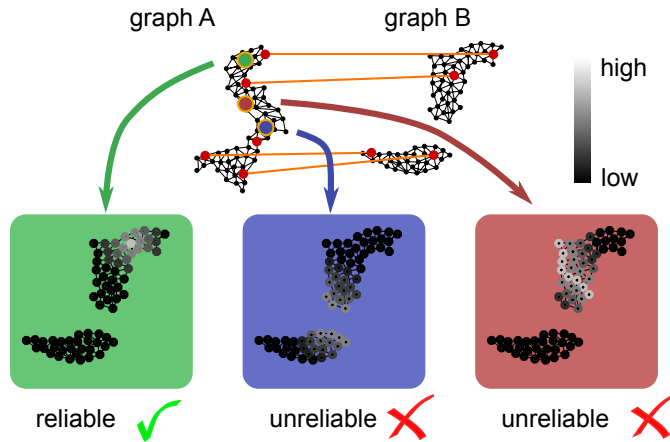


Figure 5.5: Matching probabilities (schematic drawing): For the green point (left) the variance is low so that the match is accepted as reliable. The matching point for the blue point would be located in the hole of chart  $B$ , leading to a high variance that indicates an unreliable match (for large holes, a uniform distribution remains). The red point finally, has a proper neighborhood, however, the landmark coordinates do not constrain the match sufficiently. Again, this leads to a high variance and the match is detected to be not reliable.

a smooth manifold, this is a first-order approximation of the true (but unknown) distances, which is sufficiently accurate for the sampling resolution we employ. Afterwards, we remove all vertices and edges in connected components with fewer than 100 points in order to delete small outliers patches and undersampled data.

### 5.3.2 Merging Two Charts Given Landmarks

Let us assume that we have two charts  $M_A$  and  $M_B$  and a set of landmarks  $L$  that the two charts have in common. Our goal is now to compute dense correspondences and then stitch together the charts accordingly to form a single sampled manifold.

**Probabilistic Correspondences:** We go through all points of  $a \in M_A$  and compute a probability distribution  $Pr(a \sim b|L)$  for all points  $b \in M_B$  according to Equation 5.3. If the landmarks are placed well to constrain the matching point and if redundant landmark coordinates are all consistent, a single narrow peak indicates the expected position. If only a small number of inconsistent distances are present, this scheme still leads to one pronounced maximum. In case of insufficient or completely inconsistent information, we obtain a spread-out distribution with high

variance, which can be detected (see Figure 5.5).

We use the variance of the distribution of the matching score as a reliability measure for the correctness of a match. We assume that  $M_B$  has an extrinsic embedding  $X_0(M_B)$  and annotate each point  $x_b \in X_0(M_B)$  with the probability  $Pr(a \sim b|L)$ . Then, we compute the mean and covariance of this distribution in 3D by a PCA analysis. As uncertainty criterion, we look at the largest eigenvalue of the PCA. In our implementation, we consider matches unreliable if this value is larger than  $3\epsilon_s$ . Unreliable correspondences will be excluded from the output.

We can further reduce the risk of wrong correspondences if we perform a *bijective consistency check*. Intuitively, we aim at establishing correspondences that are valid either way, whether matching from  $A$  to  $B$  or vice versa. In our probabilistic framework, this is realized by constructing a probability in graph  $A$  that the matched point in  $B$  matches back to the region around the original point in  $A$  with a high probability. Computationally, we importance sample the matching distribution in  $B$  to determine a set of potentially matching points. We then determine their matching probability on  $A$ . Assuming statistical independence of the individual potential matches, we multiply their distributions in  $A$  to obtain a probability density that the match is bijective. If the original point in  $A$  has a high probability of being matched back, we accept the match, otherwise it is discarded.

**Point-to-point correspondences:** Finally, we need to convert the matching densities to actual point-to-point correspondences. The simplest way to do so is the *nearest-neighbor* approach. We just connect  $a \in M_A$  to the point  $b \in M_B$  that maximizes  $Pr(a \sim b|L)$ . This is simple and robust but comes with an error of  $\mathcal{O}(\epsilon_s)$ . Another option is an *extrinsic* approach: we assume that  $M_A$  and  $M_B$  both have an extrinsic urshape  $X_0(M_A)$  and  $X_0(M_B)$ . We then use the nearest neighbor estimates to initialize an extrinsic optimization that aligns the two urshapes by pairwise deformable matching (see Section 5.4). From the urshapes, we recompute a new sampled manifold, as described below:

**Graph merging:** Having two aligned urshapes, we can easily recompute a new sampled manifold. We just connect each point to its extrinsic  $k$ -nearest-neighbors (in a Euclidean sense) in the overlaid urshapes.

We need to avoid connecting parts that accidentally have a similar Euclidean position but are actually far away in an intrinsic sense. This can happen because the extrinsic optimization does not perform any collision detection, Section 5.4. We therefore do not consider all points as candidates for the  $k$ -nearest neighbors

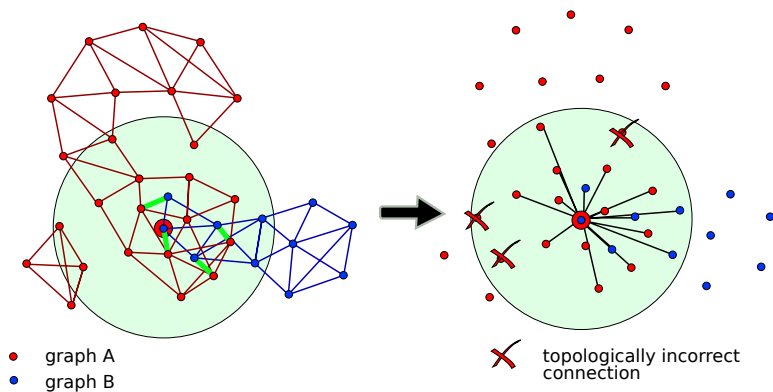


Figure 5.6: Graph merging: The blue and the red graph are merged. Points within the  $c \cdot \epsilon_s$  search range (light green) are potential neighbors of the center point. We exclude points that are not reachable by walking on the joint red and blue graph without leaving the search range. Nearest neighbor correspondences (green) can be used as “bridges” to walk from blue to red and back.

but only those that can be reached by a short walk along the graph of the sampled manifold: We allow only points within a Euclidean distance  $c \cdot \epsilon_s$  (for  $k = 20$  we use accordingly  $c = 3$ ), walking on the graph edges of  $M_B$  and  $M_A$  and using the nearest neighbor correspondences between  $M_A$  and  $M_B$  as “bridges” to switch between the graphs (see Figure 5.6 for an illustration).

In summary, this allows us to merge two charts into a single one if we find suitable landmark correspondences. It might fail to recognize corresponding parts if the landmarks are unable to reliably identify the dense matches. However, as part of the output, the algorithm will mark these regions and not provide correspondence information. Furthermore, as described above, the algorithm needs an extrinsic urshape for both charts in order to compute an accurate matching. Otherwise, a nearest neighbors solution is possible but it introduces a small error in each operation so that iterative merging would become inaccurate over time. A purely intrinsic solution might be an interesting way to go, in order to reduce the introduced error, however we have not implemented it for our pipeline.

### 5.3.3 Resampling a Chart

The next operation we need to provide is to reduce the complexity of a chart by resampling. The motivation for this is that we will need to perform many chart

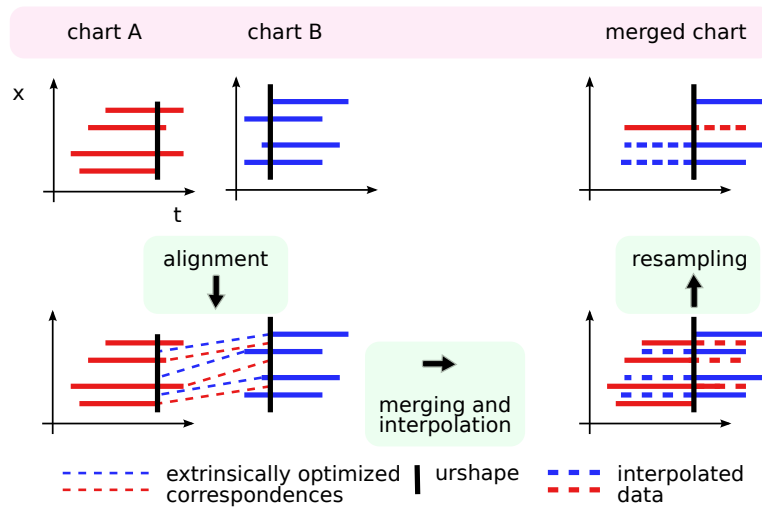


Figure 5.7: Resampling merged charts: We first merge two charts  $A, B$  (upper left) by nearest neighbor matching (lower left). The match is refined by extrinsic, numerical alignment, which leads to partially represented correspondences (lower right). When we resample the representation, we need to perform neighborhood-based interpolation to retain this information (upper right).

merging operations that will constantly increase the sampling density in overlapping parts, which at some point becomes a problem in terms of computational costs.

Resampling itself is very easy: We just use the Poisson-disc sampler to remove nodes from the graph that are still covered by nearby nodes within an extrinsic distance of no more than  $\epsilon_s/2$ . The remaining challenge is to maintain the correspondence information between the chart and the actual data. At this point we need to remember that charts encode correspondences by attaching sets of extrinsic positions of points to which they correspond. Therefore, removing chart points deletes valuable correspondence information (see Figure 5.7 for an illustration of this problem).

We propose again an extrinsic scheme to counter this problem by *interpolation*: We keep the original chart  $M$  and chart  $M'$  resampled to a sample spacing of  $\epsilon_s$ . Each  $m' \in M'$  is also a node in the original  $M$ . We look at all neighbors  $N_{\epsilon_s} \subseteq M$  of  $m'$  that are located within an (intrinsic) distance of  $\epsilon_s$ . For each time step  $t$  that is covered by the chart, we then retrieve their extrinsic embeddings. If we find at least three such points, we compute a local tangent space by fitting a least squares-



optimal rigid alignment  $\mathbf{T}$  of the points at time  $t$  to the corresponding urshape points [53]. We then estimate the correspondence of  $m'$  at time  $t$  as  $\mathbf{T}^{-1}(x_0(m'))$ , i.e., by just transforming the urshape point back to the corresponding tangent space.

A special situation occurs for landmark nodes. Since landmarks carry global matching information that is valid across different charts, these nodes cannot be deleted, moved or interpolated to different positions in the graph. Hence we just copy the landmark nodes into the resulting chart and their position does not need to be interpolated over the temporal support.

This scheme performs a first order accurate interpolation which yields satisfying results for the dense sampling we are employing in our implementation. The scheme could easily be implemented intrinsically, without having an extrinsic urshape, using the intrinsic distances as weights for the tangent space approximation but this is not necessary for our pipeline.

### 5.3.4 Detecting Apparent Topology Changes

If we use the chart merging algorithms described above to assemble a more comprehensive chart from simpler ones, we are still facing a major problem: It might happen that the *apparent topology* of the chart changes, for example if the mouth closes in a face scan. Charts build from closed mouth data have an incorrect metric structure and incorrect topology: They do not show a hole in the mouth region and the distance between the lips is too short. If we merge charts for the open and the closed mouth, this will cause problems. We therefore need to detect this situation and adapt the graph of the chart accordingly.

Given that scenario we have one invariant to detect such mistakes: the intrinsic distance between corresponding points must always be larger or equal to the largest extrinsic distance that has ever been observed, i.e. for any two points  $p$  and  $q$ ,  $\forall t : d_{\mathcal{M}}(f_t(p), f_t(q)) \geq d_{\mathbb{R}^3}(f_t(p), f_t(q))$ . This criterion is used in [153] to build a straightforward “edge-stretch” test: It just checks if extrinsic embeddings of points connected by a common edge violate this invariant, and if so, delete the edge. This works in practice but it is not very robust; it requires a delicate trade-off of elasticity and edge-stretch tolerance. We adopt this basic idea and also propose an improved, more robust algorithm.

We can implement the basic stretch test easily by just comparing the Euclidean embedding (correspondences to data) of neighboring nodes at all time steps. Because the extrinsic correspondences are stored only sparsely, we also have to resort

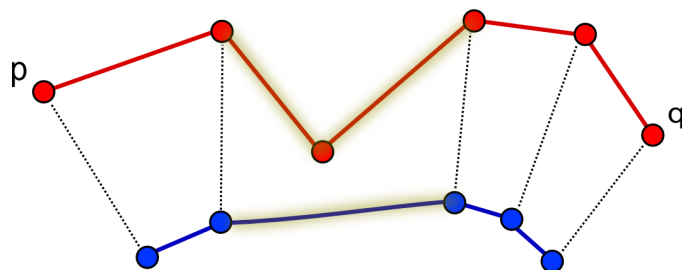


Figure 5.8: An illustration of our improved detection of topology change. Both frames  $A$  (blue) and  $B$  (red) have correspondences established between the points. The Euclidian distance between point  $p$  and  $q$  in frame  $B$  is in this example larger than the geodesic distance in frame  $A$ . Hence we look for a sub-path which violates this invariant. A vote is placed for every edge in each such sub-path. An edge which contains the most number of votes is then removed. Greenish edges indicates the subpath which collects the most votes in this example.

to interpolation from neighbors at both endpoints (as described in the previous paragraph) to make this robust. However such a basic test requires that for every two neighbour nodes their corresponding points in other frames are also direct neighbours. If for example due to discretization noise the correspondences fluctuate around several points, which are not direct neighbours, then we would never find out that there must be a connectivity cut between these two nodes.

Hence we propose an improved version of connectivity cut which looks at the problem from a more global perspective: The main idea is to look at shortest intrinsic paths and all of their known temporal correspondences in Euclidean space. If we find a sub-path for which the endpoints are at a larger Euclidean distance than the geodesic distance of the path, we know that a part of this path violates our invariant, hence one or more edges on that path must be deleted. In order to search for those paths, we compute the geodesic paths between all pairs of nodes in the graph and compare the Euclidean and the intrinsic distance for all time steps. If we find a violation (intrinsic distance being too small), we walk inwards along the path until we find the smallest interval that still violates the distance criterion. We also stop shrinking the interval if correspondences are not known. This usually does not yet give us the desired result because the intervals in which the error occurs can be quite large. We therefore perform a voting scheme in order to identify edges which are responsible for the violation. Each edge gets a vote if it shows up in a path that

violates the distance criterion. After that, we determine the set of all edges that obtained a maximum number of votes and delete them from the graph. Then we iterate this scheme until no more violating paths are found. Computing all pairs of paths is obviously too slow. We therefore restrict the search to paths of a bounded length and use only a subsample of starting points instead of all points. Figure 5.8 illustrates our approach on a simplified example.

This strategy is more robust in finding problematic connections than the edge-stretch test. However, due to the greedy deletion algorithm, it might still delete a larger set of edges than absolutely necessary. Due to subsampling, it is also possible to miss smaller topological problems. Nevertheless, our experience is that the improved strategy is significantly more robust than the previous technique. An example of the topological consistency filter on the hand puppet data set [75] is shown in Figure 5.9; the simple edge-stretch test fails here.

## 5.4 Extrinsic Matching

Finally we would like to take a small overview over the extrinsic matching algorithm which is used in some parts of our reconstruction pipeline. This is only a very short summary of the previous work of Wand et al. [152, 153]; we refer the reader to the original papers for implementation details. We describe the previous work first so that the difference of our new approach becomes clear. Furthermore, we will use the numerical approach only for refinement. This approach is very common in optimization: we first use a coarse global optimization algorithm to estimate a good initialization for a more precise (but not globally optimal) local optimization scheme.

### 5.4.1 Pairwise Local Matching

We use this extrinsic deformation routine in order to align the extrinsically matched charts of frame  $A$  and  $B$ . The main idea of the extrinsic matching algorithm is to compute a deformation field  $f : A \rightarrow \mathbb{R}^3$ , where  $A$  is a point cloud, that minimizes a matching energy:

$$E_{match}(f) = E_{dist}(f(A), B) + E_{elastic}(f) \quad (5.4)$$

$E_{match}$  combines two energy functions: The first,  $E_{dist}$ , measures the distance of point cloud  $B$  from the deformed  $f(A)$ . It sums up the point-to-plane distance

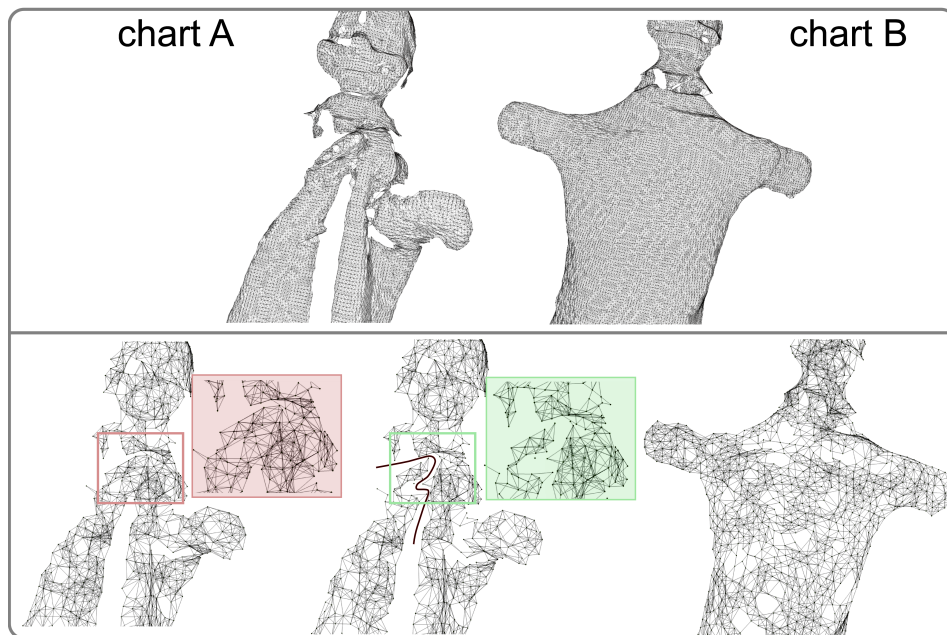


Figure 5.9: A difficult case for chart merging: the two charts of the puppet data set have a very different connectivity, i.e., different topology. In input data (top row) on the left, the hand of the puppet is merged with the body. On the right, the puppet is fully visible and hand and body are disconnected. The lower row shows the chart connectivity graph before performing the topology consistency check (left), after clean-up (middle) and for the second chart (right). Note how the hand got disconnected from the body in the left chart while the connectivity of the chart on the right is unaffected.

between points from  $f(A)$  and points from  $B$ . In order to support partial matching reliably, a number of heuristics are employed, such as checking the angle of the correspondence vectors to the surface normals. The energy  $E_{elastic}$  penalizes the elastic energy of the deformation field  $f$ , trying to keep it as-rigid-as possible. In the optimum, minimal bending and stretching is introduced while still matching the data well. The two terms are usually weighted to control the trade-off. We use the elastic subspace matching model of [152], but several other choices are possible, see for example the seminal work of [5, 49].

### 5.4.2 Animation Fitting

The pairwise matching model of Equation 5.4 is extended in Wand et al. [152, 153] to a global animation fitting approach that fits animation sequences with multiple frames to data. For this, an augmented energy function is employed:

$$E_{anim}(f) = E_{dist}(f(M), D) + E_{elastic}(f) + E_{temp}(f) \quad (5.5)$$

It now operates on a whole animation sequence. It computes the distance to the data at all frames (summation over time) and it also sums up the elastic energy in all time steps. Furthermore, it adds a new term  $E_{temp}$  that takes into account the temporal behavior of the time-dependent motion field  $f$ . It penalizes acceleration such that smooth motion is preferred. This energy can be optimized using partially initialized data, where some correspondences  $f_t(m), m \in M$  are not known. The method first fixes the known correspondences and fills in the missing data and then performs a global energy minimization. This interpolates missing data in a temporally coherent fashion and distributes the remaining error globally. Another way to view this is as a numerical bundle adjustment to improve the reconstruction accuracy.

Once again, it is very important to stress that this optimization is only reliable if the model is suitably initialized. In particular, the data term is highly non-convex so that model parts covered by data need to be prepositioned close to matching data points. We use the existing technique because of its ability to interpolate missing data and because the numerical optimization, as a continuous method, does not suffer from precision limitations (unlike some of our intrinsic algorithms, as discussed next). There is a small inconsistency, though: The extrinsic methods assume elastic deformations (minimizing bending and stretching), while the intrinsic methods

assume isometry (minimizing stretching only). For the target data, this is an acceptable restriction: The stricter assumption of elastic behavior is a reasonable regularizer, as validated extensively in previous work [49, 75, 138, 152, 153].

## 5.5 Reconstruction Pipeline

We now use the building blocks developed in the previous section to setup a complete animation reconstruction pipeline. We divide the algorithm into three conceptual stages: *i-chart building*, *p-chart merging*, and *final optimization*.

### 5.5.1 Building i-Charts

As a first step of our algorithm, we run the continuous landmark tracking to determine a set of landmark tracks  $L$ . Afterwards, we first build a separate single frame chart for every frame of the input. From this correspondence information, we build initial charts (i-charts). This is done by merging single frame charts using the continuous landmarks tracks  $L$ .

We first select a subset of starting frames to build the initial charts (our current implementation uses every tenth input time step as starting frame). For each starting frame, we build one i-chart. We first fix the landmark set  $L$  to the landmark sets that overlap the starting frame. We then walk both forward and backward in time and use the chart merging algorithm to merge the data into a larger i-chart. For this step, however, we ignore the stitching of the graphs and use only the reference frame as chart's urshape. This provides us with temporal correspondence information of the chart and a suitable urshape (i.e. does not contain any "artificial" errors which might be introduced with our stitching pipeline).

In each merging step, we exclude unreliable correspondences, and also exclude newly starting tracks that were not continuously present from the starting frame. Therefore, the amount of area covered will typically decrease with time distance to the starting frame. When the ratio of matched nodes to the number of nodes in the base frame falls below a threshold (we use 40% in all our results) we stop the temporal extension of the i-charts. Finally, we equip the newly created chart  $M_t$  with an urshape  $X_o(M_t)$ ; we just use the starting frame. Figure 5.10 summarizes the process.

We have designed this procedure to make sure that an i-chart does not contain the same piece of geometry twice at different positions in the chart: We never

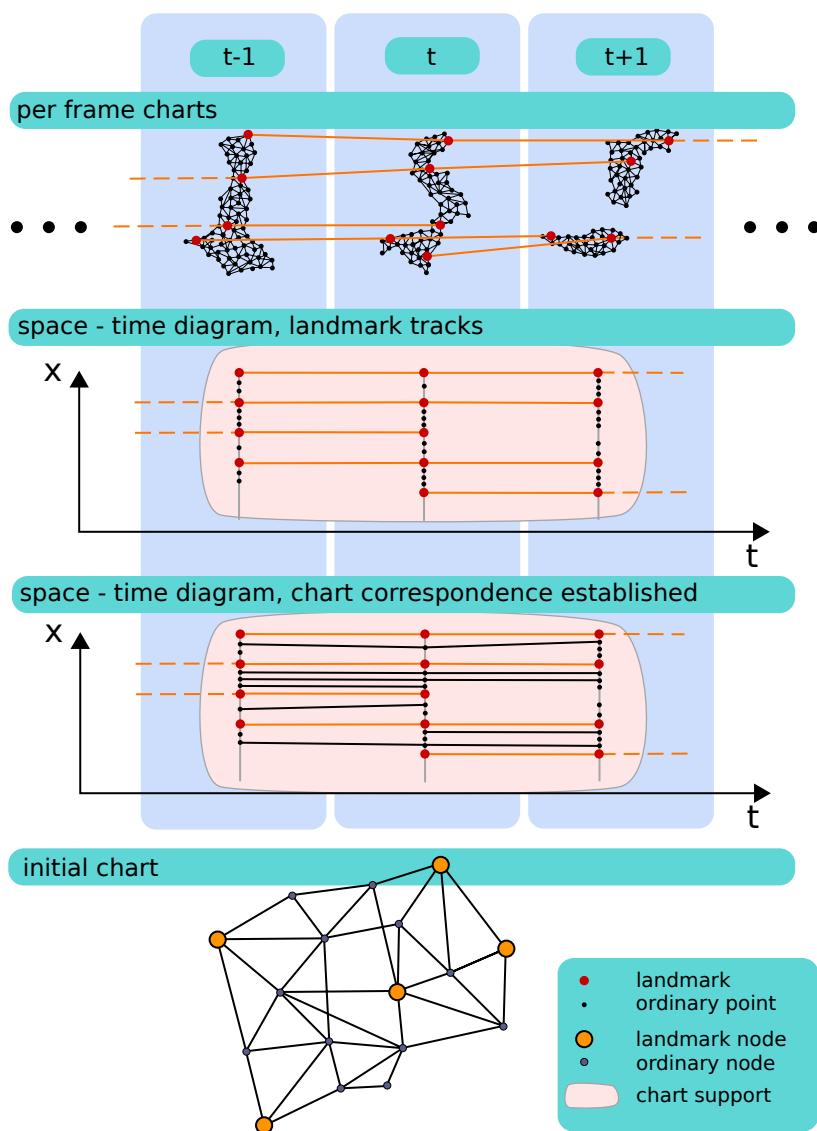


Figure 5.10: i-chart construction as space time diagram (x-axis: time, y-axis: spatial location). We first build single frame charts and track landmarks (first row). The second row shows a block of tracked landmark points. These are used to perform robust matching of landmark coordinates to establish correspondence between ordinary points (third row). The resulting chart is represented as a time independent graph that encodes the intrinsic structure (last row). Each node stores correspondences to the raw data (not shown).

include any area in an i-chart that could already have been represented elsewhere within the same chart, but where we would not yet have been able to recognize this fact. This is guaranteed by not introducing new landmarks and by only collecting reliable correspondences. Therefore, the coverage of i-charts is typically still fragmented. Patching these fragments together is the task of the next step, *p-chart merging*.

### 5.5.2 Outer Loop: Building p-Charts

We now build p-charts by stitching together separate i-charts as well as p-charts that have already been generated earlier during this process. The stitching is done by using the global matching. It first tries to establish landmark correspondences. If a sufficient number of matches is found, the two charts are assembled by chart merging, followed by subsampling. The topology check is performed before graph merging in order to reduce the error accumulation which might be introduced by merging two urshapes with false connections.

Merging by global matching has a certain risk because the RANSAC, Chapter 3, matching algorithm might fail to give good results with some small probability. We can minimize the risk by using good matching candidates first. Each i-chart and newly generated p-chart is kept in a pool of matching candidates. In order to decide on which pairs to match first, we use the following score:

$$w_{score} = \lambda_1 w_{overlap} + \lambda_2 w_{match} + \lambda_3 w_{common}, \quad (5.6)$$

$w_{overlap}$  is the temporal overlap of the charts, i.e. the number of overlapping frames of the two charts normalized by their maximum length.  $w_{common}$  is the normalized number of common landmarks in both charts.  $w_{match}$  is the average number of matched nodes during all previous i-chart or p-chart merging operations, thus quantifying how well the matching worked out in the history of this chart. The weight parameters are set to  $\lambda_1 = 3$ ,  $\lambda_2 = 2$  and  $\lambda_3 = 1$ , putting most emphasize on overlap. This heuristic scoring encourages the merging of charts that actually do overlap and are not likely to be bad matches. In addition, we also monitor the outcome of a match. Chart merging is considered a failure if only a small number of correspondences have been established in relation to the overall number of nodes (in practice, we use 30% as threshold). In case of failure, the p-chart is not added to the pool and only one of the two participating charts is kept. We keep the



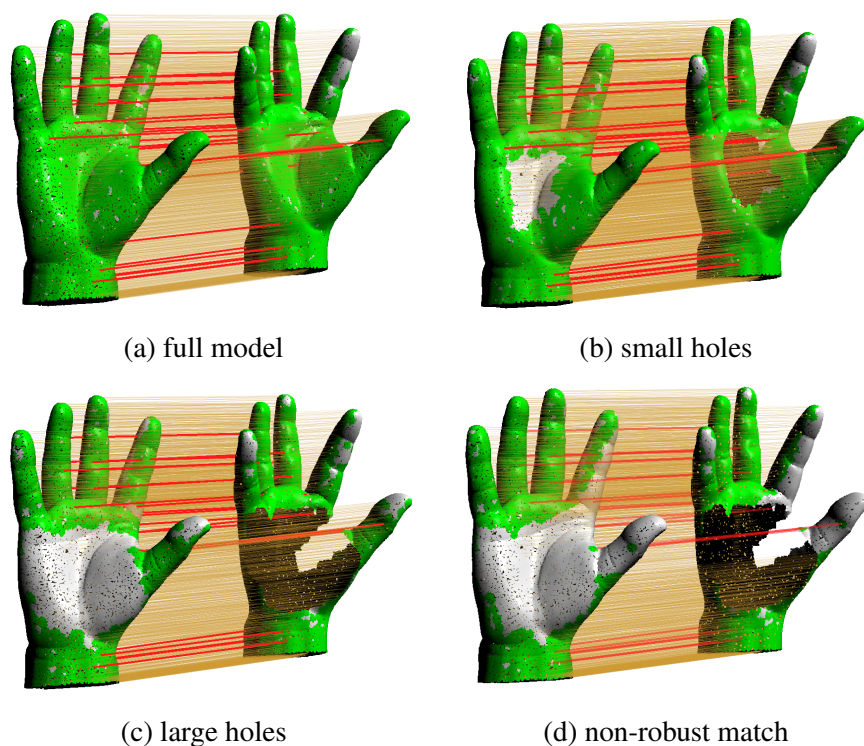


Figure 5.11: Effect of robust matching: Matching a (synthetic) hand model with a simulated acquisition hole. The results (a)-(c) use robust matching so that still large areas are covered with reliable correspondences (green). In the non-robust result in the lower right (d), significantly more area outside the hole region cannot be matched.

“better” one judging by Equation 5.6 (omitting the overlap which is not defined for a single chart).

### 5.5.3 Final Optimization

The outer loop described above is run until only one chart is left in the pool, which is the final reconstructed chart, and the primary output of our method.

We use this chart to drive a final numerical bundle adjustment according to the animation fitting algorithm, Section 5.4.2. This yields a full motion sequence where the urshape of the final chart is deformed to plausibly fit all of the data and move smoothly over time for frames or parts where no data is available.

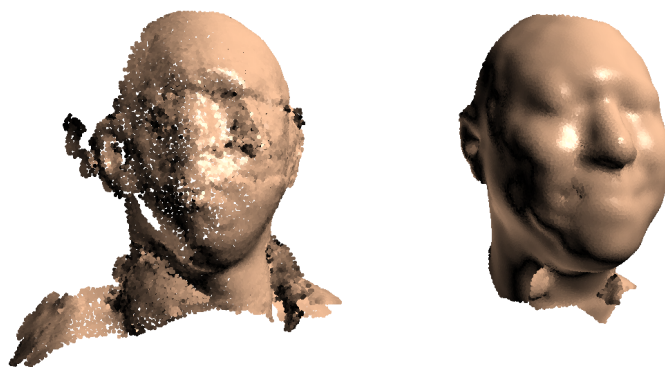


Figure 5.12: Comparison of the algorithm of Wand et al. [152] (left) and our reconstruction (right) for the “Face 2 shuffled” data set. In the case of Wand et al., large deformations between individual frames prevent a proper alignment of the data.

## 5.6 Results

We evaluate our algorithm on a number of data sets. We use the “Saskia”, “Abhijeet” and “Kicker” data sets of Vlastic et al. [151], which have been acquired using a photometric stereo approach. We also include the “Face” and “Puppet” data sets of Li et al. [75], which have been acquired with the motion-compensated structured light acquisition method of [154]. Finally, we also include the “woman” a dataset that we acquired ourselves using a Swissranger SR4000 [85] time-of-flight depth camera. In addition to the original data sets we create a shuffled version of the “Face” data set by deleting subsequences of frames and re-arranging the remaining data blocks. This data set is specifically designed to test the performance of our broken track continuation technique, Section 5.2.2. In addition, we also use a synthetic data set of a gesticulating hand, created in Poser 7, to separately evaluate the two main new pipeline stages, landmark tracking and robust charting. A brief summary is shown in Figure 5.14.

### 5.6.1 Synthetic Tests

We first examine the two most important algorithmic components of our pipeline separately before we test the complete pipeline. Figure 5.11 shows a hand model in two different poses with an increasing amount of missing data. Green area indicates that the variance of the matching distributions indicates a reliable match. The robust matching model is able to find reliable matches for most of the non-hole

area and does not create false-positives. If we turn off the robustness, the coverage is substantially reduced.

Figure 5.4 shows a tracking result on a hand sequence that in the middle undergoes an abrupt motion. The landmark tracks are correctly interrupted at this point and the RANSAC matching is invoked to build new landmark correspondences. Finally, the dense chart merging is used to obtain globally consistent dense correspondences.

### 5.6.2 Real-world Scanner Data

The different real-world data sets (see Figure 5.14) present a number of challenges to our reconstruction pipeline. For the “Saskia” data set, the legs of the person often appear to be connected to the skirt, giving evidence for a different topology than the correct interpretation of the legs being separately moving objects. In addition, we have significant amounts of missing data in the leg region. Another difficulty is presented by the arms moving upwards in the beginning of the sequence. Since the scanner is not able to resolve the gap between the arms and the body the surface seems to undergo elastic deformation. Even though this violates our model our algorithm is able to process the data. Also note how the legs are reconstructed as separate entities, see Figure 5.1 (left). This is only possible using the improved version of the topology changes 5.3.4 detection approach. The basic variant proposed in earlier work fails to recognize the individual parts.

The main challenge in the “Face” data set, which has previously been used for template-based animation reconstruction, is presented by large amounts of missing data, due to the single camera scanner setup. Important features of the head, such as both ears, are never present simultaneously in any of the input frames. The nose is often visible from one side only. Nevertheless, our algorithm successfully assembles a complete urshape of the person, including both ears and a closed nose surface, See Figure 5.1 (right) and Figure 5.14. In addition, the neck region appears to be disconnected for a large part of the sequence. Our algorithm is able to correctly connect the neck to the head. A small artifact remains: The data set contains a few small disconnected outlier patches (collar of the shirt) that are attached to the main figure in the reconstruction. Here, the available data is insufficient to handle these pieces correctly.

The “Puppet” data set is an example for a strongly deformable object. It has also previously been used for template-based reconstruction [75]. The data set



Figure 5.13: Original data (left) and reconstruction with parameterization of the “Abhijeet”, “Kicker” and “Woman” data sets for different poses.



Figure 5.14: Original data (left) and reconstruction with parameterization of the “Saskia”, “Puppet” and “Face” data sets for different poses.

is challenging due to its strongly changing apparent topology (hand connected to body), see Figure 5.9. The incorrect topology even persists for as many as 40 out of 100 frames. Even with large amounts of missing data in the folds and widely varying apparent surface topology we recover the complete sequence. Again, using the improved algorithm to resolve the topology is essential; the previous algorithm leads to incorrect reconstructions.

The “Face shuffled” data set shows a test sequence for our landmark continuation strategy. We cut the original “Face” sequence into 5 blocks of 10,29,29,10, and 28 data frames each. In between these contiguous blocks of data we deleted 12,10,8, and 5 frames of the original data frames. They are present as empty frames in the modified data set, disrupting landmark tracking altogether. As shown in Figure 5.12 (right), our algorithm is still able to reconstruct a complete template model and its motion over the full sequence, even interpolating the missing frames with plausible information. For comparison, we show a result computed with the sequential alignment algorithm of Wand et al. [152], Figure 5.12 (left), which, as expected, is not able to perform a useful reconstruction for this type of incoherent motion.

The “Abhijeet” data set is particularly challenging. The topology is ambiguous and the geometry shows systematic low-frequency artifacts. Parts of the arm are displaced by more than the diameter of the arm itself, and incorrect sheets of surface show up, probably due to the photometric acquisition approach that cannot estimate depth reliably in this highly occluded situation. The situation is particularly bad for the first 20 frames, where the arms are merged into the body and drag large sheets of phantom geometry with them when disconnecting. When we omit these frames, we obtain qualitatively correct results for the remaining 94 frames, with stable correspondences. The main artifact is that the arms in some frames twist and squeeze. However, the data supporting this is rather weak and already outside the scope covered by our matching model. We therefore believe that this example shows quite well both the limitations of our matching model itself as well as the robustness of the computational pipeline.

The “Woman” data set represents a stress test and partial failure case for our approach. The time-of-flight data is extremely noisy, which is a major challenge for the landmark tracking algorithm. In addition, the apparent topology is again constantly changing, including a full connection of the arms with the upper body in the beginning of the sequence. We obtain only sparse tracking information so

data set	Saskia	Face	Face shuffled	Puppet
# frames in seq.	116/116	200/200	141/106	100/100
avg. # data points / frame	20500	10300	10300	9800
avg. # ord. points	1250	1300	1300	1250
avg. # landmarks	82	66	66	86
comp. time i-charts	5h	3h	2h	3h30min
comp. time outer loop	3h	1h20min	1h	1h40min
comp. time post-proc.	30min	1h	1h	25min

data set	Abhijeet*	Kicker*	Woman*
# frames in seq.	92/112	20/20	100/100
avg. # data points / frame	20000	16000	8300
avg. # ord. points	1800	2000	3000
avg. # landmarks	81	79	34
comp. time i-charts	4h30min	1h18min	21min
comp. time outer loop	1h	19min	2h
comp. time post-proc.	12min	8min	4min

Table 5.1: Statistics for processing the individual data sets.

that our algorithm was not able to reconstruct dense correspondences reliably over the body for all frames but some data remains uncharted. Hence, the final optimization produces a smooth interpolation that in some parts does not follow the data. The quality of the reconstructed geometry is low; the correspondence noise does not permit resolving high frequency details in the final reconstruction, but the result is qualitatively correct. For such kind of very low-resolution data, additional cues such as a simultaneously recorded video with color information is probably necessary to permit better reconstructions.

In Table 5.1, we show statistics of algorithm run times and other characteristic data for the different sequences. The first row shows the number of reconstructed frames versus the available data frames. Note that for the “Face shuffled” data set more frames are reconstructed than are present in the original data. The second row shows the average number of data points per frame of the input sequence, while row three shows the number of nodes in a typical chart. The average number of detected landmarks per frame is shown in row four. Note that this number varies widely over frames. Finally, the computation times for the different steps of our algorithm

are given. The computations were run on a 2xQuadCore Intel Xeon X5550 with 2.67 GHz. Datasets marked with a “\*” were computed on 2xHexaCore Intel Xeon X5650 with 2.67 GHz.

## 5.7 Discussion and Limitations

As shown by the example scenes, our algorithm is able to handle more general input data that could not be reconstructed automatically by previous techniques. Not relying on temporal coherence is an important step for practical applications. Although scanners are available that scan at very high frame rates, the fact that geometry often vanishes in acquisition holes and reappears in a different pose is a strongly limiting factor in practice to previous algorithms. We can also show that the proposed algorithm is quite robust. Even for data sets with strong noise or artifacts outside our modeling assumptions, we still obtain qualitatively correct results.

As most complex reconstruction systems, our method has a number of parameters. However, we were able to fix most of these parameters for all of the data sets, as described in the text. We only adapted the sampling resolution  $\epsilon_s$  to minimize the computational costs. In addition, we have increased the number of minimal evidence landmarks in the robust matching scores from 5 to 6 in the “Abhijeet” data set as this lead to slightly better results. Finally, we have adapted the regularizer weights for stiffness and acceleration penalty in the final numerical optimization for best visual (aesthetic) impression.

Our method still has a number of limitations that require further research: A problem is the handling of “unreliable” data. Our current pipeline dismisses this data in the construction of initial i-charts but during p-chart merging, we currently do not delete uncharted data because this could reintroduce large holes in the charts but rather rely on extrinsic alignment to match these pieces. This problem can be addresses by a good scheduling of the merging operations, which are commutative but not associative: the order in which pairs are merged matters. The current heuristic tries to minimize the negative impact by aiming at large overlap, but better orders (possibly including options for backtracking from bad matches) might exist.

A second issue is the detection of topological changes. Although we can handle more scenes than previous techniques, we still encounter problems in some



situations. In particular, if large acquisition holes and topological changes coincide, this can lead to incorrect results where the local connectivity is not resolved correctly. An example is the face scan from [153]. Our technique cannot resolve the opening of the mouth because of large acquisition holes opening up around the lips simultaneously with the opening of the mouth. The missing area is too large to be handled by even robust intrinsic matching. In this case, a purely extrinsic technique has an advantage over our approach.

Finally, the combination of elastic and isometric matching is sometimes a limiting factor: for objects with very strong deformations, this introduces a bias towards rigidity, leading to insufficient bending. A purely intrinsic formulation of the charting could probably reduce these problems. However, this seems to be a minor issue in practice that can usually be resolved by reducing the strength of the elastic regularizer appropriately.

### 5.8 Conclusions

In this chapter we investigated a global optimization technique for animation reconstruction from dynamic point cloud sequences as produced by dynamic range scanning devices. The method is based on the concept of cartography and uses an intrinsic framework for a more reliable and robust matching of partial deformable shapes in vastly different poses. Iteratively applying this technique automatically yields a completed template model, its motion over the course of the acquired sequence and a consistent parameterization. Our technique uses a landmark tracking scheme that uses temporal coherence if available but can fully automatically resort to the efficient randomized global matching algorithm of Chapter 3 if required by the data. We can thus recover from scanner shortcomings such as large scale occlusion and we can handle fast motion in the scene. We also improve the robustness in detecting topological changes. Overall, we are able to process sequences under significantly more general conditions than previous work.





## Part Summary

---

In this part of the dissertation we have shown different approaches for solving the shape matching problem for pairwise matching and for matching of complete animation sequences. In Chapter 3 we introduced a randomized search strategy which is capable to identify a subset of correspondences between two shapes with respect to isometric deformation. We also saw that applying a robust estimator score while importance sampling the potential correspondences enables us to perform a fast correspondence estimation between a pair of shapes. The main advantage of this approach is the ability to do global registration without any a priori knowledge of the initial pose of the object or any initial alignment.

Based on our knowledge obtained during the development of this algorithm, we developed an intrinsic shape matching algorithm as presented in Chapter 4. We investigated the problem of shape matching from a numerical point of view only. We presented an approach which is capable of computing an optimal subset of surface points, to which we refer as landmarks. These landmarks represent reference points which are used in order to “address” every other point on the surface. Since we are looking for correspondence estimation between two shapes which underwent nearly isometric deformation, we concluded, that if we find a matching between these landmarks, we can automatically find a match for every other point on the surface. The previous work done in this area indicates that very few landmarks are sufficient. However, this results are only valid in ideal cases. Our approach shown in Chapter 4 shows the numerical way to find a number and the placement of these landmark points, for practically robust matching of two shapes.

Based on the concept of landmark coordinates and equipped with the possibility of matching two point clouds acquired by a scanning device we introduced an application, animation reconstruction, in Chapter 5. Animation reconstruction is

a reconstruction of the shape and its motion from a set of a dynamic set of point clouds. Using the knowledge of robust pairwise point cloud matching, Chapter 3, we were able to find a global solution for the animation reconstruction problem even if there are large temporal gaps in the set of point cloud frames. The concept of landmark coordinates, Chapter 4, was extended in the Chapter 5 in order to find dense correspondences, which allows for an intrinsic shape merging/charting. This enables us to reconstruct the shape as a temporal “puzzle” where all the parts are combined together to retrieve the final shape.

In general, we have shown that the problem of shape matching for three dimensional shapes has a valid algorithmic and also a theoretical solution. We were able to reduce the problem of matching two shapes isometrically to the question of how to place the important points on a surface, so that the matching problem becomes efficiently solvable. Using the RANSAC approach for correspondence estimation, we were able to find the right solution in a fraction of the time that would be required by a brute force search.

In the next part of the thesis we will take a look at a correspondence estimation approach through an optimization framework known as *belief propagation*. Thus showing a different way of formulating the shape matching problem and its solution.

## **Part III**

# **Inference Problems**



# INFERENCE PROBLEMS ---



In the previous chapters we described several approaches to compute correspondences between a pair (or a set) of shapes. The described algorithms estimate a suitable matching between a set of points which are distributed over the surface of the shapes. In general, we can classify these kind of algorithms as an *intermediate representation* shape matching algorithms. Thus, given a shape, we compute first an intermediate representation of the shape, in our case a set of landmark or feature points, and finally perform a matching between them.

In this part of the thesis, we take a step into the direction of matching shapes without an intermediate representation. Hence we formulate the problem of shape matching as a direct dense correspondence estimation process. We study this problem from the perspective of symmetry detection, i.e. finding parts on a single shape which are symmetric to each other. This problem is related in its formulation to the regular shape matching problem since it can be solved by computing all the matches of the shape to itself.

In Chapter 7 we present a novel algorithm based on a conceptually simple and straightforward probabilistic formulation of partial shape matching which is itself represented by a Markov random field. This way, we obtain a probability distribution over all possible intrinsic matches of a shape to itself, which reveals the

---

symmetry structure of the object. Rather than examining this on an exponentially sized distribution directly, which is infeasible, we approximate marginals of this distribution using sum-product loopy belief propagation (LBP) and show how the symmetry information can subsequently be extracted from this condensed representation. We incorporate a parallel implementation on graphics hardware, which enables us to extract symmetries of deformable shapes efficiently.

Finally, we leave the problem of shape matching and take a closer look at the belief propagation algorithm itself, in particular at its performance. Loopy belief propagation is a powerful tool for approximate inference in Markov random fields. The generality of the algorithm is particularly appealing: Unlike other approximate inference techniques, LBP is mostly independent of the structure of the Markov random field and the probabilistic potentials used in the model. The algorithm can be applied to graphs with general clique sizes, arbitrary potentials, and general label sets.

However, for problems with large state spaces, the runtime costs are often prohibitively high. In order to solve large state space problems with LBP we encode all beliefs, marginals, and messages in a wavelet representation, which can represent the probabilistic information much more compactly. We operate solely in the wavelet domain without leaving the space for any intermediate computations. This yields an output-sensitive algorithm where the running time depends mostly on the information content rather than discretization resolution. We apply the new technique to typical problems with large state spaces such as image matching and wide-baseline optical flow where we observe a significantly improved scaling behavior with discretization resolution. For large problems, the new generic technique is significantly faster than even an optimized spatial domain implementation.

## Introduction

Our model is inspired by the work of Anguelov et al. [6] who formulated the shape matching problem as a Markov random field using geodesic distances as a pairwise consistency condition. Intuitively, the model enforces two conditions: (1) a correspondence should point to pieces of geometry that within a local neighborhood match the source geometry, (2) neighboring correspondences should behave in a consistent way. For isometric shape matching this means that pairs of correspondences should preserve intrinsic distances. Enforcing this condition strictly only



---

allows for global shape matching as we have analyzed in the previous chapters. However, we would like to go one step further and detect *symmetric* parts, i.e. pieces of a shape which are similar to each other. This kind of problem is closely related to the global deformable shape matching. The additional challenge here is that we are looking for partial self-matching, by registering a shape to itself.

The detection of symmetry patterns in data sets is a very useful tool for a large number of applications, such as compression, noise reduction, object recognition or abnormality detection. Although the recognition of symmetries is a seemingly effortless task for human perception, the algorithmic extraction of symmetries in digital data sets is a very challenging task. It receives a lot of attention in computer vision and graphics research [21, 89, 90, 97, 100, 101, 104, 111].

In order to achieve the goal of symmetric part detection with our probabilistic model, we employ truncated potentials that yield piecewise consistent solutions that describe partial shape matches. This model describes a probability distribution over all possible assignments of source to target geometry. This probability space is of exponential size, and it would be very hard to find actual symmetry matches. The critical observation for our method is that this extremely large space is usually mostly empty: For realistic 3D models, only very few sets of correspondences exist that actually have a significant non-zero probability. Therefore, it is possible to compute a solution in a projection to a much smaller subspace and still retain all the relevant information. We implement this idea by computing the marginals of distributions for each source point, i.e. the distribution over all correspondences of a single point, marginalized over all assignments to the other correspondence variables. This “averages” and thus overlays all potential matches. In a final postprocessing step, we then extract the actual symmetries from this condensed representation.

In general, we make three main contributions in the first chapter of this part:

- Our method is based on a conceptually simple probabilistic formulation of partial shape matching. The actual matching algorithm is then derived from this formal model.
- Due to the generality of our framework we are able to detect intrinsic partial symmetries. This means that symmetric parts of a deformable object (for example the hands of a human) are recognized independent of the actual pose as long as the deformation is approximately isometric.
- We compute marginals using loopy belief propagation on the graphics hard-

---

ware to reduce the computational effort

We apply our method on several standard 3D models and extract partial symmetries, in general, deformed poses. Hence demonstrating that the theoretical model indeed leads to a practical solution for symmetry detection in more general settings than previous approaches.

As we will see towards the end of the Chapter 7 the size of the inference problem becomes very large for practical problem sizes. The number of *states* (or *labels*) which every variable in the Markov random field can take is in the order of thousands. For  $n$  random variables, each of the variables can be in  $k$  different states with potentials of order (clique size)  $c$ , each iteration of the basic algorithm needs  $\Theta(nk^c)$  running time. For large state spaces, this is often prohibitively expensive, even in the case of pairwise MRFs ( $c=2$ ). Due to this space-time complexity belief propagation is infeasible for very large state spaces.

In general, this “restriction” is also unfavorable for MRFs with continuous states. Such MRFs are frequently used in applications such as shape and image matching, where labels for example represent positions in a 2D image. Representing such distributions accurately with a grid of discrete bins often leads to high costs despite coarse resolutions.

However, if take a closer look at the marginal distributions within such algorithms as shape matching, optical flow, stereo reconstruction, etc., the marginals computed with these approaches typically show a lot of coherence so that a full histogram representation is waste. For example, in image matching applications, we obtain sharply peaked initial beliefs at salient features and rather uniform, uninformative beliefs within uniformly colored regions. The LBP algorithm then distributes this knowledge under some regularizer. This successively creates more peaked beliefs until only a few final sharp peaks are left. Figure 6.1 shows such typical marginal distributions before and after belief propagation is applied.

In the second and last chapter of this part of the thesis, we examine a more compact representation of marginals, which leads to a novel belief propagation algorithm that scales better to state spaces discretized at a high resolution. Instead of the original histograms, we operate in a space of linearly transformed histograms. A natural choice for the basis of such a message-passing space is a wavelet basis because wavelets are able to efficiently encode both low frequency distributions, referring to high uncertainty, as well as sharply peaked distributions. Uncertain knowledge is easily captured by a few low frequency wavelet coefficients. On

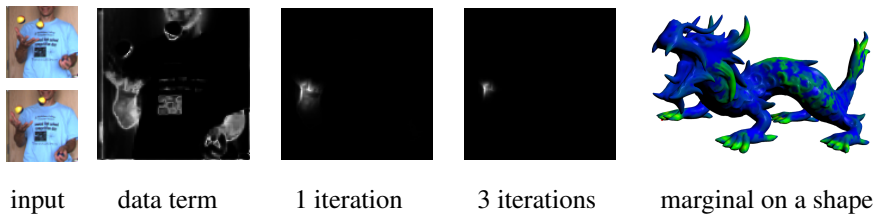


Figure 6.1: Marginal distributions in an optical flow and a symmetry detection by shape self-matching (Chapter 7) application. Marginals are computed with the loopy belief propagation algorithm. Typical marginals are usually sparse and consist of some smooth regions with a few salient peaks distributed over the surface.

the other hand, a Dirac delta pulse represented at resolution  $\mathcal{O}(\epsilon)$  can be encoded with  $\mathcal{O}(\log \frac{1}{\epsilon})$  coefficients, as shown in Chapter 8. Because of the linearity of the transform, mixtures of such cases can also be handled.

In order to work with this representation, there is a major technical obstacle: While the transformation and compression of individual messages and beliefs into the wavelet domain is straightforward, the message passing algorithm itself cannot be easily formulated in the wavelet domain. A trivial solution would be to transform the message back to the spatial domain at each iteration step. However, this would avoid the computational advantages of the encoding. The other obvious alternative is using  $n$ -ary wavelet products for message evaluation, which in a naïve implementation leads to exponential costs with respect to  $n$ .

In general, we make the following main contributions:

- We present an algorithm which employs recent results on efficient evaluation of  $n$ -ary wavelet product integrals [94, 136] in order to perform the direct algebraic product evaluation in linear rather than exponential time.
- We develop a wavelet-space encoding and a hierarchical scheduling scheme for efficient computation of the integration over pairwise potentials required in each LBP iteration
- We show that we can formulate a resolution independent version of LBP that does not require any a priori discretization of the domain.

For evaluation, we apply the algorithm to image matching, which is closely related to shape matching, and optical flow tasks that require large label spaces.

---

We compare the implementation to a simple histogram-based implementation of LBP as well as an optimized version. The observed scaling behavior of the new technique is significantly better than that of both spatial domain algorithms. Computational costs grow significantly weaker with discretization resolution, and for large label spaces, the new algorithm outperforms the spatial domain implementations.



# Shape Matching as an Inference Problem

---

## 7.1 Model definition

First, let us remind on the actual problem definition. Given two manifolds  $\mathcal{M}_S$  and  $\mathcal{M}_T$  and two point subsets  $M_S \in \mathcal{M}_S$  and  $M_T \in \mathcal{M}_T$ , we seek a deformation function  $f : \mathcal{M}_S \rightarrow \mathcal{M}_T$ , which maps every point  $s \in \mathcal{M}_S$  to every point  $t \in \mathcal{M}_T$ . The subsets  $M_S$  and  $M_T$  denote the discrete, sampled versions of the source and target shape respectively. As already stated in the introduction, in this chapter we are focusing on detecting symmetrical parts of a single object, hence in this case  $\mathcal{M}_S = \mathcal{M}_T$ . However, the model described in this chapter is directly applicable to the case of two different shapes.

In our practical implementation we obtain the point sets  $M_S$  and  $M_T$  by a Poisson-disc sampling of both source and target surface, where the target surface receives a higher resolution (typically: a factor of 4) in order to be able to adjust the matching positions reliably. In the following we will refer to  $\mathcal{M}_S$  and  $\mathcal{M}_T$  as original shapes, i.e. input shapes which were not sub-sampled, and  $M_S$  and  $M_T$  as the sub-sampled versions.

In contrast to the point-to-point mappings as presented in Part II of the thesis, a whole distribution over the function  $f$  is computed. This stores matching probabilities for every possible target point. Similar idea has been already used in Chapter 5, however this time the distribution is computed in a global way incorporating also the distribution of neighbor points.

Each point in  $M_S$  represents a node in a graph or, more precise in probabilistic inference terminology, a hidden node associated with a state variable. The state variables are the set of points  $M_T$  and are denoted as  $f_s = f(s), s \in M_S$  to indicate the label/state assigned to the node  $s$ . As in [6], adjacent points on the surface are connected with each other to preserve geometric compatibility between corre-

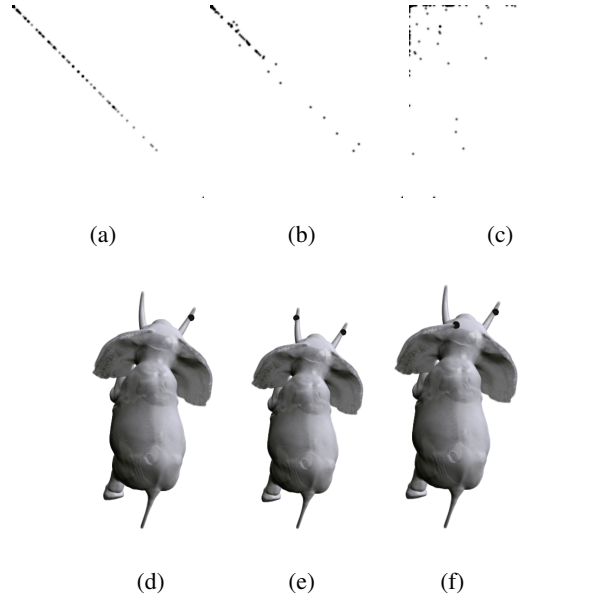


Figure 7.1: (a) NMI between 2 identical spin images around one point. The point is shown in (d). (b) NMI between spin images around similar point surroundings. The two points are shown in (e). (c) NMI between spin images of different points surroundings. The corresponding points are shown in (f).

spondences. We consider only pairwise interactions between neighboring nodes. A set of neighbor nodes for each node  $s$  is denoted as  $N_s$ . This results in a joint probability distribution of the form

$$P(f) = \frac{1}{Z} \prod_{s \in M_S} \Phi_s(f_s) \prod_{q \in N_s} \Psi_{sq}(f_s, f_q) \quad (7.1)$$

The probability that a source point corresponds to a target point is given by the single potential  $\Phi_s$ . For the single potential we use the normalized mutual information (NMI) [149] between two corresponding spin images [60]. Spin images acts as descriptors in our case. NMI is a similarity measure based on entropy [122]. Given  $p_I(i)$  which is the probability that a discrete random variable  $I$  has value  $i$  the entropy is defined, as already introduced in Chapter 4 as:

$$H(I) = - \sum_i p_I(i) \log p_I(i) \quad (7.2)$$

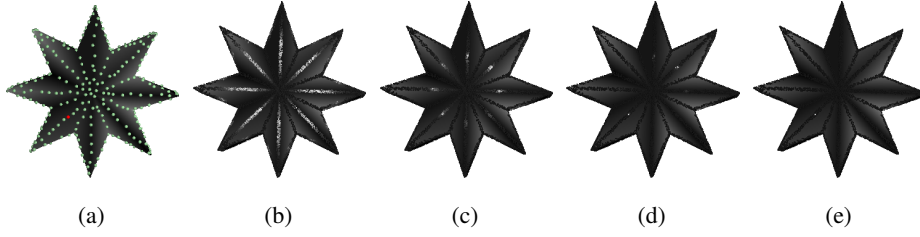


Figure 7.2: Example of parameter settings. (a) The set of sampled points  $M_S$  on the source side and the selected source point in red for which the belief is shown on the target side on the subsequent images. The brighter the spots the better the source point matches the corresponding target points. (b) A probability distribution without any smoothness term ( $\lambda = 0$ ). (c) and (d) Parameter settings ( $\lambda = 1, \rho = 0.2$ , respectively  $\lambda = 5, \rho = 0.1$ ) for more globalized symmetry patterns. At the other end of scale, (e) shows a marginal distribution which has exactly one peak ( $\lambda = 5, \rho = 0.05$ ).

For two discrete random variables  $I_1$  and  $I_2$  the entropy of their joint distribution is defined as

$$H(I) = - \sum_{i_1, i_2} p_{I_1, I_2}(i_1, i_2) \log p_{I_1, I_2}(i_1, i_2) \quad (7.3)$$

The more similar the distribution of the random variables  $I_1$  and  $I_2$  is, the lower the joint entropy is compared to the individual entropies. The NMI is at a maximum when both distributions are equal. It is defined as

$$NMI(I_1, I_2) = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)} \quad (7.4)$$

Figure 7.1 illustrates an example of NMI distribution for different points selected on a shape. In order to reject outliers in the single potential, we employ a robust function [78]

$$\Phi_s(f_s) = \frac{1}{1 + \frac{(1-NMI)^2}{\gamma}} \quad (7.5)$$

The smaller  $\gamma$  the higher the rejection rate. In our experiments we set  $\gamma$  to 0.01. Other similarity metrics can be employed for the single potentials like curvature maps [44], or shape index (computed from principal curvatures) [38].

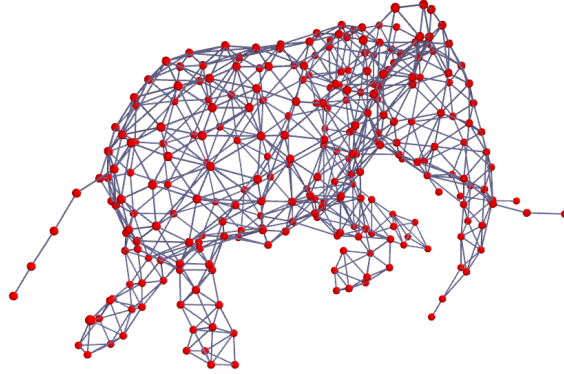


Figure 7.3: Example of network over the uniformly sampled 271 points of the elephant dataset 7.1.

For the pairwise potential  $\Psi_{sq}(f_s, f_q)$ , we also use a robust potential where the probability of an assignment for the two neighboring points  $s \in M_S$  and  $q \in M_S$  is defined to preserve locally isometric deformations. It is encoded by the difference in the geodesic distances between the source neighboring points  $d_{M_S}(s, q)$  and the assignment  $d_{M_T}(f_s, f_q)$ .

$$\begin{aligned} \psi_{geometric} &= \exp \left\{ -\lambda (d_{M_S}(s, q) - d_{M_T}(f_s, f_q))^2 \right\} \\ \Psi(f_s, f_q) &= \max(\psi_{geometric}, \rho^2) \end{aligned} \quad (7.6)$$

$\lambda$  controls the smoothness of the distribution in the assignment space. The higher the value the more influential becomes the pairwise potential enforcing geometrical consistency. For partial symmetry detection it is important to allow for multiple peaks in the marginals while ensuring geometrical consistency. This is controlled by the parameter  $\rho$  which enables to have different probability values for neighboring source points (i.e. allow for discontinuities). It takes values between  $[0.1..1.0]$ . In our experiments we set  $\rho$  to 0.4. In the literature the parameter is called the truncation parameter in the negative log likelihood space [148]. The level of discretization on the target side has to be linked with  $\lambda$  since it defines the allowed variation from a perfectly isometric mapping.

By varying  $\lambda$  and  $\rho$  we can extract symmetries at different levels of resolution. For instance, global symmetries, i.e. shape matching, can be detected at a coarser resolution and hence  $\lambda$  should be set high. Fine symmetries can be detected at a high resolution and in this case  $\lambda$  should be smaller. See Figure 7.2 for examples



of parameter settings.

### 7.1.1 Sum-Product Belief Propagation

We shortly recall the sum-product belief propagation algorithm for Markov random fields. For each node in the graph the marginal probability in the correspondence space is computed. This is achieved by passing messages around the graph in a defined neighborhood  $N$  of each node. The graph is computed in the following way: around each sampled point on the surface, a region-growing on the full resolution mesh is performed. If two regions meet, both sampled points are connected by an edge. Figure 7.3 shows an example of a network configuration. The messages from a node to its neighboring nodes are passed either in parallel, hence all nodes pass their messages in parallel, or in sequential order. Goldberger et al. [47] analyze the benefits of a serial order message updating compared to parallel. In general, serial order converges two times faster than parallel updating. Since for a serial update rule the memory requirements are smaller than for the parallel order (i.e. no need to hold the message vector of the previous time step in memory) and it converges in the general case more quickly, we use serial order message updating. The message update rule is computed as follows:

$$m_{s \rightarrow q}(f_q) = \sum_{f_s \in M_T} \left( \Psi_{sq}(f_s, f_q) \Phi_s(f_s) \prod_{n \in N(s) \setminus q} m_{n \rightarrow s}(f_s) \right), \quad (7.7)$$

where  $m_{n \rightarrow s}$  is the message sent from the  $s$ ' neighbor node  $n$  to  $s$ . After  $t$  iterations the belief vector is computed for each node

$$b_q(f_q) = \Phi_q(f_q) \prod_{n \in N(q)} m_{n \rightarrow q}(f_q), \quad (7.8)$$

where the belief value  $b_q(f_q)$  is a probability of how well point  $q \in M_S$  matches the point  $f_q \in M_T$ .

## 7.2 Implementation

In order to reduce the complexity, we choose  $M_S$  to be much smaller in size than  $M_T$ . The level of surface discretization determines also the symmetry scale detection, i.e. a denser sampling results in finer symmetries. In our experiments,  $\lambda$  is set to 5,  $\rho$  to 0.4 and we run 10 iterations. We have tested our framework on

two rigid models (cuboctahedron and star) and on three deformable shapes (male figure, elephant and dragon). Furthermore, to test the ambiguous registration, two deformed frames of a lantern model have been used.

The messages are initialized with uniform probabilities, i.e.  $1/|M_T|$ , so that the probabilities for a node  $s$  sum up to 1. At the first iteration the order of passing is the order of vertex numbering and at all other iterations we generate a random path. One iteration consists in a forward pass to compute the probabilities when arriving at the node and a backward pass computing the final probabilities in the current iteration (in the backward pass the order is reversed). The complexity of the message computation is quadratic in the number of labels and the message passing is linear in the number of vertices for the pairwise MRF (Equation 7.7). In order to shorten the computation time, we have implemented a GPU version of our algorithm as described next.

### 7.2.1 Exploiting parallelism

Belief propagation is known to be well suited for a parallel implementation. Several researchers have implemented BP either on a large scale CPU cluster [14] as a support vector regression solver, or on a GPU [28] to solve stereo vision problems. In this work we also exploit the GPU parallelism to compute the sum-product belief propagation.

For our GPU implementation we use an nVidia GeForce GTX280 with CUDA environment, which provides 1GB of video memory and double precision support. The messages sent between nodes are stored as a 2D array of doubles in the CUDA memory space. The number of processed nodes and labels is limited by the available video memory. For example the dragon2 data set, which has the dimensions of  $|M_S| = 259$  and  $|M_T| = 11837$  labels occupies 832 MB including a small amount of additional memory space for temporary data structures. The memory consumption doubles if parallel message update is used, since the data structure for  $m_{p \rightarrow q}^t(f_q)$  and  $m_{p \rightarrow q}^{t-1}(f_q)$  need to be held in memory. As stated before we use sequential message update schedule in order to process huge number of nodes and labels with a smaller memory footprint. In this case the convergence time of the algorithm becomes almost one half of the time needed when updating the messages in parallel as [47] shows, which provides an additional benefit. The GPU implementation is outlined in the following pseudocode.

The message from node  $s$  to  $q$  for each label  $f_q$  is computed in one thread,

**GPU Procedure 1** Compute BP in parallel over  $f_q$ **for all nodes  $s$  do**    compute  $z_s = \Phi_s(f_s) \prod_{n \in N(s) \setminus q} m_{n \rightarrow s}(f_s)$  (in parallel over  $f_s$ )    compute  $m_{s \rightarrow q}(f_q) = \sum_{f_s \in M_T} (\Psi_{sq}(f_s, f_q) \cdot z_s)$     normalize  $m_{s \rightarrow q}(f_q)$ **end for**compute  $\prod_{n \in N(q)} m_{n \rightarrow q}(f_q)$ compute  $b_q(f_q)$ 

however with different computation kernels. Threads are arranged in blocks and are scheduled by the hardware to different multi-processors on the GPU. Threads arranged in warps are executed in parallel. The CUDA guide [95] gives a more detailed information about this architecture. One of the main bottlenecks in CUDA applications is a non-optimized access to the memory. To solve this issue, Silberstein et al. [126] proposes to use a block’s shared memory to implement a software managed cache for the sum-product computation. However this strategy is not applicable for our datasets which exceed the available shared memory size. Hence we arranged the used datasets in a coalescent way (see Figure 7.4), so that read and write accesses performed by each thread can be executed more efficiently. Coalescent access is achieved, when multiple threads in a block access memory at sequential addresses, taking full advantage of GPU’s memory bandwidth. In case where this was not possible or where many threads access nearby areas in the look-up tables (i.e.  $\Phi_p$ ), we have used CUDA’s texture interface to incorporate hardware cached access for the data. See CUDA’s guide [95] for a detailed description of performance optimization techniques.

We support variable neighborhood size which differs from previous belief propagation implementations on the GPU [28]. However, this increases the branching complexity of the implementation of the message update rule, Equation 7.7. As an improvement, we precompute the product of the incoming messages into a node  $s$  as shown in the algorithm “GPU Procedure 1” before the main message update computation. The parallelization is performed over all labels  $f_q$  and  $f_s$ , hence each thread from a two-dimensional thread block (see [95]) computes the product of incoming message and the single potential  $\Phi_s$  for all possible  $f_q$  and  $f_s$  combinations. The result is stored in the data-structure as shown in Figure 7.4. Using this data structure the computation of all incoming messages into a node  $q$  for each label

$z_1$	$z_2$	...	$z_i$	...	$z_{n-1}$	$z_n$	$m_{N(q_i)[0] \rightarrow q_i}$
$z_1$	$z_2$	...	$z_i$	...	$z_{n-1}$	$z_n$	$m_{N(q_i)[1] \rightarrow q_i}$
$z_1$	$z_2$	...	$z_i$	...	$z_{n-1}$	$z_n$	$\vdots$
...	...	...	...	...	...	...	$\vdots$
$z_1$	$z_2$	...	$z_i$	...	$z_{n-1}$	$z_n$	$m_{N(q_i)[m] \rightarrow q_i}$
$z_1$	$z_2$	...	$z_i$	...	$z_{n-1}$	$z_n$	

Figure 7.4: Illustration of the datastructure used to speed-up the sum-product computation. The values  $z_s$ , see "GPU Procedure 1" on the previous page, are stored row wise. Messages from neighbor vertices are stored columnwise, for example:  $m_{N(q_i)[k] \rightarrow q_i}$  is the value of  $z_s$  for the neighbour  $k$  of node  $q_i$ . Thread parallelization over all labels  $f_s$  and  $f_q$  enables efficient implementation.

$f_q$  becomes a simple multiplication along a column. Afterwards we compute the actual value of the message vector for label  $f_q$  in parallel by summing up the previous result row-wise. Splitting up these two computations enables a more coalescent computation which perfectly suits for the *SIMD* (Single Instruction Multiple Data) structure of the GPU's processing units. Although we have a sum-loop within a computation kernel in the second step of the shown GPU procedures, the access is still coalescent and is performed in parallel with respect to the labels  $f_q$ , resulting in efficient read from the memory. Our GPU implementation achieves a speedup up to 700x in comparison to the single-threaded CPU implementation, see Table 7.1.

### 7.3 Partial symmetry detection

To recall, for each node or source point  $s_i$ , the belief vector defines a marginal probability distribution over all correspondences on the target pointset. In order to recognize symmetric parts of an object, we use the peaks, i.e. labels with the high probability, in the marginal distribution,  $t_i \in M_T$ , to start a region growing algorithm around them. The peaks are predetermined by a threshold, which is typically set to be in the range  $[0.5..0.75]$ , i.e. points with a smaller marginal are rejected and points which are above the threshold are added to a queue. The criteria for the growing is to consider all the neighbors  $n_i \in M_S$  in the graphical model of  $s_i \in M_S$ , and their corresponding geodesic distance  $d_{M_S}(s_i, n_i)$ . In the same time, we identify the peaks in the marginal distribution of these points,

data set	$ \mathcal{M}_S ,  \mathcal{M}_T $	$ M_S $	$ M_T $	$\max N $	time GPU	time CPU
dragon1	20002	115	6000	14	4.10s	2870.30s
dragon2	53859	259	11837	10	40.39s	27083.92s
cuboctahedron	200000	126	10888	9	16.83s	1075.45s
star	30181	479	8000	14	73.58s	28004.27s
elephant	42321	271	12722	11	55.45s	36920.53s
male figure	41492	270	8000	8	14.17s	12894.38s
lantern	16735	111	11434	8	12.32s	8604.37s

Table 7.1: Test data set name, number of points in the original source shape and target shapes ( $|\mathcal{M}_S|$  and  $|\mathcal{M}_T|$ ), number of graph vertices ( $|M_S|$ ), number of correspondences on the target side ( $|M_T|$ ), maximum neighbor number, time measured for GPU solution, time measured for single-threaded CPU (2.66 GHz and 3.25 GB RAM) solution. Timings are given for 1 iteration only. Note that for all datasets except the two deformed lantern shapes which have the same number of points, the source and target deformable shapes are identical ( $\mathcal{M}_S = \mathcal{M}_T$ )

$t_i = f_{s_i} \in M_T$  and  $t'_i = f_{n_i} \in M_T$  and their corresponding geodesic distance  $d_{M_T}(t_i, t'_i)$ . If  $|d_{M_T}(t_i, t'_i) - d_{M_S}(s_i, n_i)|$  is within an error bound  $d_{error}$  (typically  $d_{error} \in [0.001..0.15]$ ), the corresponding target point  $t'$  as also its counterpart  $n_i$  is also added to the seed queue for the region growing. All the involved geodesics have been previously computed for the belief propagation algorithm.

Iterating this algorithm until the queue is empty results in regions which are identified as being symmetric to the region marked by point  $s_i$ . This is the case as long as any pair of direct neighbor points in shape  $\mathcal{M}_S$  and their counterparts identified by higher marginals in shape  $\mathcal{M}_T$  preserve the geodesic distances.

## 7.4 Results

We applied our algorithm to several datasets, dragon, elephant, male, lantern, star and cuboctahedron.

Figures 7.5(c), 7.5(f), 7.5(i), and 7.5(m) show the marginal probability distribution for the red marked source points shown in Figures 7.5(b), 7.5(h), and 7.5(l) color encoded over the triangle mesh<sup>1</sup>. The color encoding of the probability is seen in Figure 7.5(a), where blue color stands for 0 and red for 1.0. For a contin-

<sup>1</sup>the triangle mesh is used only for visualization purposes

uous representation of the probability distribution  $P$  on the original pointset of the shape  $\mathcal{M}_T$  or on the triangle mesh, we use positive basis functions  $u$  which have a support radius  $\sigma$ :

$$P_{s_i}(\mathbf{t}) = \sum_{j=1}^{N(t)} \lambda_{i,j} u_j(\mathbf{t}), \quad \mathbf{t} \in \mathcal{M}_T, \quad (7.9)$$

where  $\sum_{j=1}^{N(t)} \lambda_{i,j} = 1$  and  $u_j(\mathbf{t}) = \exp\{-\|\mathbf{t}_j - \mathbf{t}\|^2/\sigma^2\}$ .

To evaluate the symmetry detection results, we manually segmented the recognized parts of the shapes. We then computed the percentage of points that have been correctly classified. Incorrect points are either those outside the manual segmentation or missing. The overlap is annotated in the figure captions. To check how prone to local minima the framework is, which might vary strongly for belief propagation applications, we run belief propagation several times starting from random initializations for the results datasets. Belief propagation converged to the same fixed point on the average of 7 iterations. This does not guarantee global convergence, but is a strong indicator of robustness.

Dragon2 represents a case of higher resolution of the dragon dataset, Figure 7.5(l)-7.5(o). The feet of the dragon have been recognized only partially since the back of the feet differ too much. Unfortunately, the scales of the dragon could not be accurately detected, even when the model was sampled more densely. We assume that this failure case is caused by geodetic noise, which is introduced by only approximating geodesics on a  $k$ -neighborhood relatively sparsely sampled in respect to the size of the scales. Finer sampling reflects in infeasible computation time and memory requirements, which is why we did not sample the dragon more densely.

The extraction of symmetries is influenced by two parameters: the marginal threshold and geodesic matching quality encoded in  $d_{error}$ . In case of highly deformable parts like the ears of the elephant,  $d_{error}$  can be relaxed. This parameter removes outliers from other peaks that show a local similarity to the source point. For the male figure, the hands have been recognized well.

The 12 symmetries of the vertices defining the cuboctahedron have been mutually recognized (Figure 7.6(a)). And in case of ambiguous matching, the algorithm could retrieve all four parts of the lantern (Figures 7.7(a), 7.7(b)) all of which underwent isometric deformation. This shows that the theoretical statements made in this chapter are well applicable also for deformable shape matching.

## 7.5 Symmetry breaking

The algorithm shown so far to find symmetries works reasonably well, however due to the nature of loopy belief propagation there is a potential shortcoming of the algorithm. In general, belief propagation can compute correct results only in case of absence of loops in the underlying graph structure. In our case, however, the graph structure used to approximate the geometry contains loops. In such a configuration loopy belief propagation can only yield to an approximate marginal distribution.

This problem has been already analyzed by Pearl [106] for the polytree algorithm on a Bayesian network containing loops. In presence of loops messages are circulating around the network and hence the marginal distribution described by the messages may not converge to a stable equilibrium. Same thoughts are directly applicable to belief propagation because of the message passing structure around the graph. Thus the marginal distribution can start to oscillate instead of a stable convergence. This has been observed and further analyzed by Murphy et al. [93] in the context of error-correcting codes as well as by Jansen [59] in the context of partial symmetries in two-dimensional images.

Due to the oscillation of the marginals the distribution containing several peaks, which is in our case given by the presence of symmetries in the underlying shapes, might start to “prefer” one of the peaks by assigning a slightly higher probability value. This can further be reinforced by consecutive iterations. We refer to such a scenario as *symmetry breaking* which is similar to a phenomenon known in the field of physics. Thus, higher number of iterations could “over-fit” the marginal inference and hence results in a deficient solution.

Murphy et al. [93] propose to solve the problem of oscillation by replacing messages sent at time point  $t$  with a weighted average of the message at time  $t$  and  $t - 1$ . The authors show that this simple change sufficiently reduces the chance of oscillating marginals while still resulting in same beliefs. Jansen [59] approximates the beliefs using *minimal spanning trees* instead of the graph structure and shows that the symmetry detection in 2D images benefits of these changes.

The examples shown in this chapter have not observed the symmetry breaking behavior due to a graph structures not suffering of large amount of loops as it can be for example observed in the 2D image symmetry problems [59]. In most cases the symmetric part we were looking for in the shapes are located in loop-

free regions (see Figure 7.3 for an example) which again prevents the symmetry breaking. However, it is worthwhile to analyze this problem in future work and to see if we can incorporate the concepts of Murphy et al. as well as of Jansen in order to reduce the chance of symmetry breaking further.

## 7.6 Conclusion

We have presented a probabilistic framework for identifying partial intrinsic symmetries in geometric data which can also be applied for global shape matching. It exploits the marginal distribution over all possible correspondences and gives a global localization of potential symmetries. By tracking the peaks in the marginals, we are able to extract nearby symmetric parts that can be explained by an isometric deformation. As a result, we obtain a technique to extract partial intrinsic symmetries in general data sets. One limitation of the current implementation is the noise handling. Since noise acts as a low-pass filter on the available geometric information, for stronger noise ( $> 0.3\%$  object size), we obtain smoothed marginals which introduce false positives in the "peak tracking" step. Our framework can handle only a small amount of noise. We consider this approach also to be useful for other problems besides symmetry detection. For example, shape matching algorithms are frequently troubled by ambiguities, which is in particular a problem for partial and multi-piece matching. Our technique is able to represent ambiguities explicitly so that the application algorithm can use this information and avoid problems, for example by combinatorial search over several alternatives in multi-piece matching. The requirement to sample source and target surface rather densely are the main bottleneck in terms of computational costs.

In the next chapter we will develop an improvement of the basic belief propagation algorithm in terms of spatial as also computational complexity. We will show that by transforming the belief propagation algorithm into another data representation, we can simultaneously improve on memory consumption as well as on the running time required to compute the inference result.



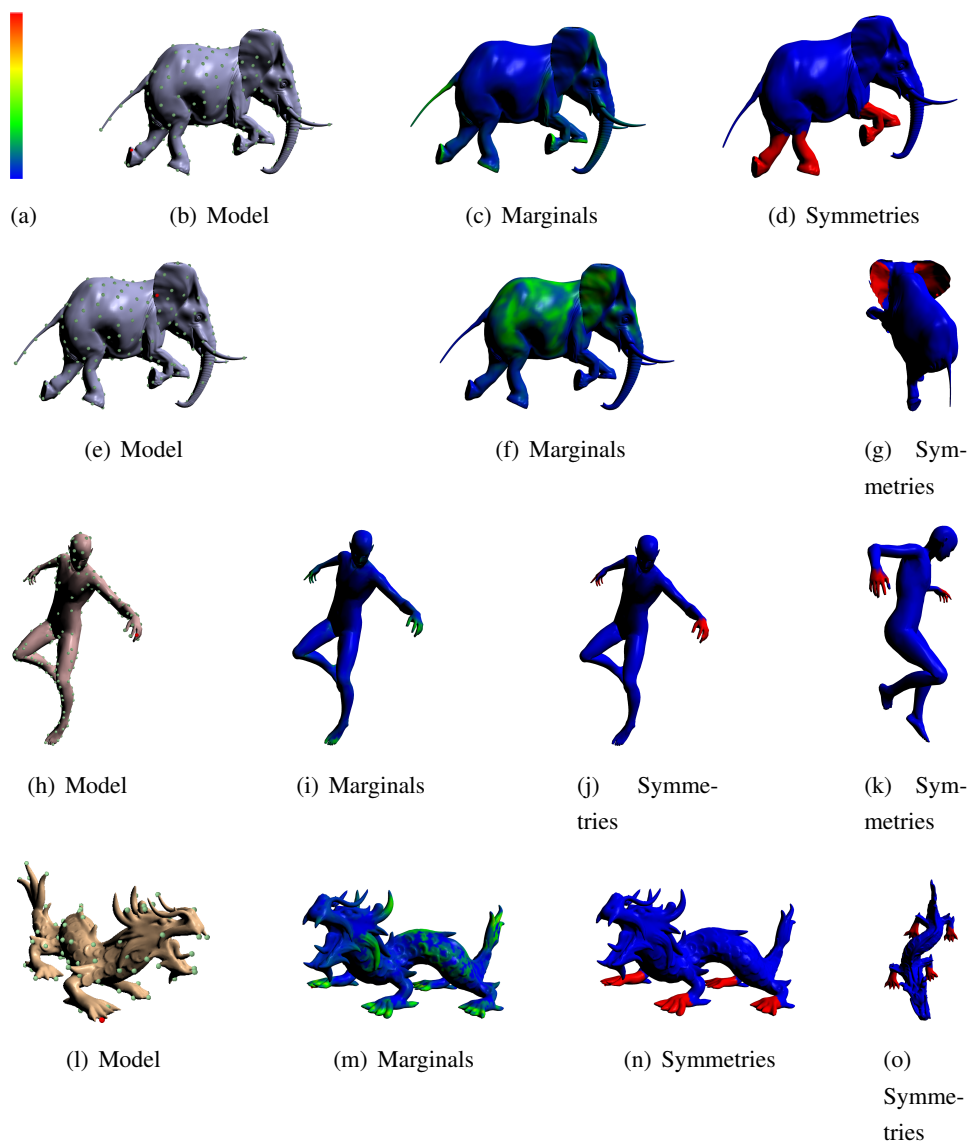


Figure 7.5: Visualization of different datasets, color encoded interpolated marginals for the corresponding selected point (in red) and symmetries. (a) Color encoding of the marginals, blue is equal to 0 and red to 1.0. (b)-(g) Elephant dataset. The overlap with a manual segmentation is 79% for the ears and 92% for the legs. (h)-(k) Male dataset. The hands have an overlap of 89% compared to the manual segmentation. (l)-(o) Dragon1 dataset. The overlap with the manual segmentation is 91%.

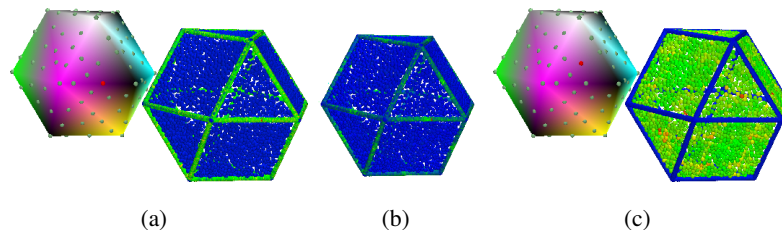


Figure 7.6: Cuboctahedron dataset. (a) On the left the selected point on an edge of the cuboctahedron. On the right the marginals computed without smoothness term ( $\lambda = 0$ ) on the point set  $M_T$ . (b) Computed marginals with  $\lambda = 5$  and  $\rho = 0.05$ . Note that the peaks are now more localized around the symmetric points on the edges. (c) Selected point on one of the faces of the cuboctahedron (left) and the marginal distribution of the point set  $M_T$  (right). The selected point matches to any other points on any other face of the object.

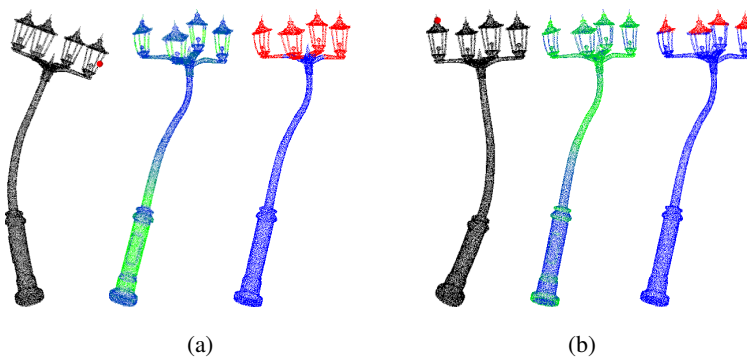


Figure 7.7: Lantern dataset. (a) Selected point marked in red (left) followed by the marginals on the point set  $M_T$  (middle) and extracted symmetries on the shape  $\mathcal{M}_T$  (right). It correctly identified the 4 ambiguous parts. Note that the segmentation stopped at the bending of the arms where the geodesic error became too large. (b) Another example on the same dataset. In this case the parameter controlling the geodesic compatibility  $d_{error}$  is set more restrictively such that only the tops are segmented.



# Wavelet Belief Propagation

---

In the previous chapter we studied belief propagation (BP) and employed this approach for symmetry detection or shape matching. We have seen that the disadvantage of BP is the high amount of memory required to store the marginals for every node in the graph. Given  $k$  labels and  $n$  nodes we have a space complexity of  $\mathcal{O}(nk)$  and even worse the time complexity is  $\Theta(nk^c)$ , where  $c$  is the clique size in the graph. However, the attractiveness of BP for typical applications such as matching, stereo reconstruction or optical flow is rooted in the fact that the marginals often show a coherent structure. Figure 6.1 shows these typical marginals.

In this chapter we will use this fact and present a new approach which reduces the space-time complexity of the belief propagation algorithm. The theory presented here is not the first step towards continuous state representation of the random variables [50, 58, 99, 133] in a graphical model; however, in contrast to these techniques, we represent the full messages in the wavelet domain. In addition we achieve very high signal-to-noise ratios with this “compression” approach.

In general throughout this chapter we will concentrate on several computer vision tasks, like optical flow or image matching. This example applications allow us a more intuitive presentation of the theory and provides easier representation of the underlying problem. Additionally, the complexity achieved with loopy belief propagation executed on a regular grid is below the one of a general graphical structure.

## 8.1 Wavelet representation

The goal of this contribution is to perform belief propagation algorithm completely in the wavelet domain. For the overview and the definition of the belief propagation approach please refer to the Section 1.2.1 of the Chapter 1.

In the following, we use a Haar-wavelet basis  $B = \{b_1, \dots, b_N\}$  with  $N = (2^k)^d$ ,  $k \in \mathbb{N}$ , i.e., the resolution in each dimension is a power of two, with a *non-standard decomposition*, see [131] for more details on wavelets. A nonstandard decomposition yields basis function of a square non-zero support in each level, see Figure 8.1(a); whereas a *standard decomposition* yields rectangular basis functions. A nonstandard decomposition enables efficient algorithms for computing products of wavelet encoded signals as shown in [94, 136]. The Haar basis consists of one constant scaling function  $b_1$ , covering the whole domain, as well as wavelets that describe local differences at different scales. The basis functions together form a  $d$ -dimensional tree, e.g. for  $d = 2$  this is a quadtree. The basis is orthonormal and all functions except from  $b_1$  integrate to zero over the full domain  $\Omega$ . We now express all marginal distributions that show up in the LBP algorithm, Section 1.2.1, as linear combinations of wavelet functions:

$$m_{ij}(\mathbf{x}) = \sum_{p=1}^N \lambda_p^{m_{ij}} b_p(\mathbf{x}) \quad (8.1)$$

$$\Phi_i(\mathbf{x}) = \sum_{p=1}^N \lambda_p^{\Phi_i} b_p(\mathbf{x}) \quad (8.2)$$

$$\Psi_{ij}(\mathbf{x}, \mathbf{y}) = \sum_{p=1}^N \sum_{q=1}^N \lambda_{pq}^{\Psi_{ij}} b_{pq}(\mathbf{x}, \mathbf{y}). \quad (8.3)$$

In the following, we will use just  $\lambda^{m_{ij}}$ ,  $\lambda^{\Phi_i}$ , and  $\lambda^{\Psi_{ij}}$ , without the indices  $p, q$ , to denote the complete vector of wavelet coefficients. Before we go on, Equation 8.3 needs some more attention: since  $\Psi$  is a  $2 \cdot d$ -dimensional function we cannot use the same basis as for the other,  $d$ -dimensional functions. At this point, we opt for a tensor product construction  $b_{pq}(\mathbf{x}_i, \mathbf{x}_j) := b_p(\mathbf{x}_i) \cdot b_q(\mathbf{x}_j)$ . In case if  $b_p$  and  $b_q$  are both 1D-basis functions, this yields a 2D standard decomposition Haar wavelet basis, and for higher dimensions, we obtain functions that are nonstandard shaped restricted to the first and last  $d$  dimensions, and standard shaped for combinations of these dimensions. Figure 8.1(b) illustrates the build of the two-dimensional standard Haar-wavelet basis function as a tensor product of two one-dimensional basis functions. Choosing the tensor product basis at this point facilitates the computation of the message passing integral, as we will see in the next step.

We now plug this representation into the message passing equation 1.6 intro-

duced in Chapter 1. After algebraic reordering, we obtain:

$$m_{ij}(\mathbf{y}) = \sum_q b_q(\mathbf{y}) \left[ \sum_{p=1}^N \lambda_{pq}^{\Psi_{ij}} \int_{\mathbf{x} \in \Omega} b_p(\mathbf{x}) R_{ij}(\mathbf{x}) \right] \quad (8.4)$$

with

$$R_{ij}(\mathbf{x}) := \left( \sum_{r=1}^N \lambda_r^{\Phi_i} b_r(\mathbf{x}) \right) \prod_{k \in \mathcal{N}_i \setminus j} \left( \sum_{l=1}^N \lambda_l^{m_{ki}} b_l(\mathbf{x}) \right) d\mathbf{x}. \quad (8.5)$$

As we see, due to our tensor product basis representation of the  $\Psi_{ij}$ , we can split the basis functions  $b_{pq}$  into two parts and directly obtain a wavelet representation for the new message  $m_{ij}$ . The coefficients of the basis are given by the expression in square brackets. We now deal with this expression in two steps: First, we evaluate the products in the term  $R_{ij}(\mathbf{x})$  (Equation 8.5) using the  $n$ -ary wavelet product integral algorithm of Sun et al. [136]. In a second step, we perform the outer summation. We show that this step corresponds to a matrix vector product and provide an adaptive approximation algorithm for this step.

## 8.2 Wavelet Product Integrals

The term  $R_{ij}(\mathbf{x})$  is a product of functions represented in the same wavelet basis  $B$ . Algebraically, this is a product of  $|\mathcal{N}_i| + 1$  sums of up to  $N$  terms each. Assuming that  $r$  of them are non-zero, a naïve expansion of this product would result in  $\mathcal{O}(r^{|\mathcal{N}_i|+1})$  terms. Clearly, this is not acceptable.

Recently, there has been considerable interest in computing such products of wavelet-represented functions in the field of three-dimensional rendering, where integrals of products of transformed functions occur for example in lighting computations. Ng et al. [94] provide a fast solution for triple product integrals, which has been extended to general products by Sun et al. [136]. In both cases, a non-standard Haar wavelet basis is used. The key observation is that the functions are orthogonal and have only limited support. Due to the quadtree-structure of the basis functions, only direct and indirect descendant nodes can cause non-zero contributions in a pointwise multiplication, and the integral over most combinations is still zero. Based on these observations, Sun et al. [136] construct an algorithm that can evaluate the product of  $m$  wavelet-represented functions with at most  $r$  non-zero coefficients each in time  $\mathcal{O}(r \cdot m)$ .

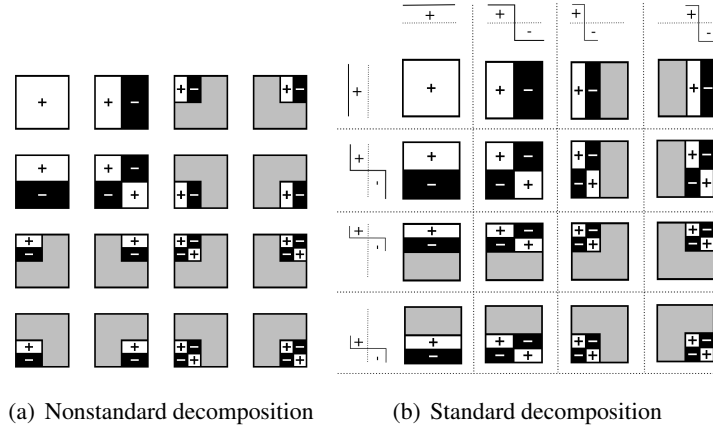


Figure 8.1: The 2D-basis functions of a Haar-wavelet obtained with a nonstandard and standard decomposition. (a) A nonstandard decomposition is defined to be of square in each level and comes directly from the definition of the Haar-wavelet decomposition [131]. (b) A  $d$ -dimensional basis function obtained with standard decomposition can be expressed as a tensor product of two  $d - 1$ -basis functions. In this example, the tensor product is build by two 1-dimensional basis functions, illustrates along horizontal and vertical axis. In case of two 2D-basis function the coupling of both squares results in a 4D domain with hyper-rectangles as a basis function. [Image taken from [130]]

We now use their wavelet product algorithm to compute the coefficients of the function  $R_{ij}(\mathbf{x})$ . Because we will always be looking at a fixed message passing step from node  $i$  to  $j$ , we will in the following omit the indices  $i, j$  to simplify the exposition. The wavelet product algorithm builds explicit hierarchies ( $d$ -dimensional quadtrees) to represent each term in the product and then incrementally merges these trees by a simultaneous hierarchical traversal. The output of the algorithm is a quadtree hierarchy that stores all non-zero wavelet basis coefficients for the product. We denote these by  $\lambda^R = (\lambda_1^R, \dots, \lambda_N^R)$ . For further details of the wavelet product algorithm, see [136].

Next, we deal with the integral in Equation 8.4: The integration over  $b_p(\mathbf{x})$  multiplied with  $R(\mathbf{x})$  is actually easy to evaluate [136]. As  $B$  is an orthonormal basis, this integral just outputs the coefficient with index  $p$  from the wavelet tree of  $R$ .

In the following, we will write the wavelet coefficients of  $\Psi$  as an  $N \times N$  matrix  $\mathbf{\Lambda}^\Psi := [\lambda_{pq}^\Psi]_{p,q}$ . In this notation, the summation in square brackets in Equation 8.4 reduces to a dot product of the wavelet coefficients  $\lambda^R$  and the  $q$ -th column of  $\mathbf{\Lambda}^\Psi$ . The full message vector  $\lambda^m$  can be thus computed as the matrix-vector product:

$$\lambda^m = \mathbf{\Lambda}^\Psi \lambda^R, \quad (8.6)$$

The task is now to evaluate this product efficiently, which we will address in the next section.

### 8.3 Adaptive Approximate Summation

A naïve multiplication leads to  $\mathcal{O}(N^2)$  running time, identical to the standard algorithm. To speed up, we make use of the special structure of this data. Both the input vector  $\lambda^R$  and the output vector  $\lambda^m$  consist of wavelet coefficients structured as  $d$ -dimensional quadrees. The support of the associated basis functions is strictly hierarchically nested. We use this property to compute hierarchical bounds on contribution of subtrees within  $\lambda^R$  and  $\mathbf{\Lambda}^\Psi$ . For  $\lambda^R$ , this is easy: In linear time, we can compute the sum of squared coefficients within the subtree of each node. Because of the orthogonality of the wavelet basis, the sum of squared coefficients values for one node is exactly the integral error that we make if we drop one such subtree completely. In the following we denote this sum as  $\lambda_\tau^R$ .

#### Hierarchical bounds

Next, we determine hierarchical bounds on  $\mathbf{\Lambda}^\Psi$ . The supports of the basis functions associated with these coefficients are also hierarchically nested. However, because of the tensor product construction, the structure is a bit more complex: The compatibility potentials  $\Psi$  are  $(2d)$ -dimensional objects. For example, in the case of a 2D state space, the  $\Psi$  is a 4D function. Each coefficient in the wavelet representation describes the coupling of a 2D square in the input and another 2D square in the output domain. Figure 8.1(b) illustrates the coupling of two one-dimensional basis functions. The same holds for the corresponding hyper-squares in arbitrary dimension. The actual bounds are easy to obtain if we know the difference between maximum and minimum absolute values of  $\Psi$  within the non-zero domain of the basis function. Section 8.4 discusses three different ways how to compute these values. In general, no wavelet coefficient can be larger in absolute value than

this bound scaled with the area (or hyper-volume in case of four dimensions) and the normalization constant of the wavelet basis. For any given function  $f(x)$  one can prove this claim as following: given an orthonormal Haar-wavelet basis, the corresponding wavelet coefficient  $\lambda_i$  is computed as [131]:

$$\lambda_i = \int_{\Omega} f(x)b_i(x)dx,$$

where  $b_i(x)$  is the Haar-wavelet basis function and  $\Omega$  is the domain of the function  $f(x)$ . This can be reformulated as:

$$\lambda_i = \int_{\Omega^+} f(x)b_i(x)dx + \int_{\Omega^-} f(x)b_i(x)dx + \int_{\Omega^0} f(x)b_i(x)dx, \quad (8.7)$$

where  $\Omega^+$  and  $\Omega^-$  are subdomains of  $\Omega$  with the Haar-wavelet basis function being positive or negative respectively.  $\Omega^0$  corresponds to the subdomain with  $b_i(x) = 0$ . Taking the normalization constant out of the basis function, the Equation 8.7 can be rewritten as:

$$\lambda_i = Z \left( \int_{\Omega^+} f(x)dx - \int_{\Omega^-} f(x)dx \right) = Z(A - B), \quad (8.8)$$

where  $Z$  is the Haar-wavelet normalization constant of the basis function  $b_i$ , i.e.,  $b_i(x) = Z \cdot b_i^0(x)$  and  $b_i^0(x)$  is the Haar-wavelet box-function, see [131]. For any function  $f(x)$  the boundary conditions are defined as:

$$\int_{\Omega'} f(x)dx \leq |\Omega'| \max_{x \in \Omega'} f(x) \text{ and } \int_{\Omega'} f(x)dx \geq |\Omega'| \min_{x \in \Omega'} f(x),$$

where  $\Omega'$  is just any subdomain of  $f(x)$ . Hence the boundaries for  $A$  and  $B$  from Equation 8.8 are:

$$A \leq |\Omega^+|f_{max} \text{ and } B \geq |\Omega^-|f_{min} \Leftrightarrow -B \leq -|\Omega^-|f_{min},$$

where  $f_{max} = \max_x f(x)$  and  $f_{min} = \min_x f(x)$ . For the Haar-wavelets it follows that  $|\Omega^+| = |\Omega^-| = N$ . Substituting this and the inequalities of  $A$  and  $B$  into Equation 8.8 we get for the absolute value of  $\lambda_i$ :

$$|\lambda_i| = |Z(A - B)| \rightarrow |\lambda_i| \leq \lambda_7^\Psi = |ZN(f_{max} - f_{min})|. \quad (8.9)$$

### Adaptive summation

Using the hierarchical bounds on  $\Lambda^\Psi$  and  $\lambda^R$ , we can now evaluate the vector-matrix product efficiently. Figure 8.2 shows the key steps. The blue tree represents



the output tree for  $\mathbf{m}$ , which is empty in the beginning. An empty output tree represents the vector of wavelet coefficients  $\lambda^m$  initialized to zero, i.e.,  $\lambda^m = [0, \dots, 0]^T$ . Since nodes which represents zero coefficients have not to exists in the wavelet quadtree representation, the initial tree is just empty. The orange colored tree encodes the input  $\lambda^R$ . A link connecting nodes of both trees represents a product operation  $\lambda^m \rightarrow \lambda^m + \lambda_{pq}^\Psi \cdot \lambda_p^R$ . We now try to restrict ourselves to the most important interactions by using a priority queue to schedule the most interesting regions of interaction first. Please note that the importance of interactions depend both on the structure of  $\Psi$  as well as on the actual function  $R$ . Intuitively, the algorithm prefers to contribute to the solution  $\lambda^m$ , only that part of missing information, i.e.,  $\lambda_{pq}^\Psi \cdot \lambda_p^R$ , which promises the largest reduction in the reconstruction error. For each link, we set the priority to the upper bound of the integral error that we would incur if we leave out this whole subtree of computation. This error is estimated by multiplying the upper error bound on the corresponding region in the domain of  $\Psi$  with the upper error bound of the corresponding region in the domain of  $R$ , i.e.,  $\tau^w = \lambda_\tau^\Psi \cdot \lambda_\tau^R$ .

In the first step, Figure 8.2(a), of the tree scheduling algorithm, the roots of both trees are linked with infinite cost. This initial link represents the product of scaling coefficients of the pairwise potential and the input tree  $\lambda^R$ , i.e.,  $\lambda_0^m = \lambda_{00}^\Psi \cdot \lambda_0^R$ . Now a pair with the largest upper bound  $\tau^w$  on the potential error is picked iteratively from the queue. Intuitively, this improves the solution by computing the most promising products first. The multiplication indicated by the linked pair is performed and the result is stored. Next, all link combinations between the children of nodes of both trees are put on the queue, Figure 8.2(b). Additionally, Figure 8.2(c), links between every child node and pair nodes are built. For these links, the parent nodes are marked as fixed, which enforces that the iteration will never subdivide these nodes but keep the connection to the fixed parent node. These links correspond to asymmetric combinations of hyper-squares of different size. In matrix notation, these links corresponds to the column-wise (fixed on  $\lambda^R$  side) or row-wise (fixed on  $\lambda^m$  side) multiplication, see Figure 8.2(d).

The stopping criterion is the relative  $l_2$  error of the current estimate: While building the message, its current square integral norm is tracked, i.e.  $c = \sum_{i \in M} (\lambda_i^m)^2$ . Here  $M$  indicates the set of output nodes whose coefficient is non-zero or in other words:  $M$  is a set of existing nodes in the wavelet quadtree. At the same time, we can track the amount of potential error left in the priority queue

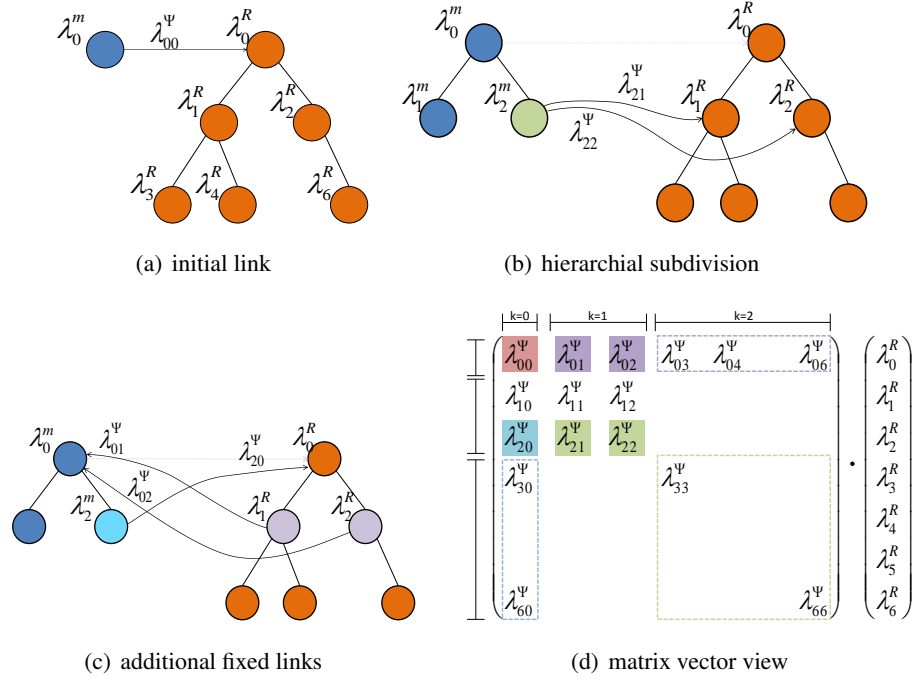


Figure 8.2: Illustration of the adaptive matrix vector product. Trees on the left (blue) are outputs, trees on the right (orange) are input. (a) Initial link on the priority queue denoting the product  $\lambda_0^m \rightarrow \lambda_{00}^\Psi \cdot \lambda_0^R$ . (b) After processing a link, new connections which indicates the product of the children of both processed input nodes, are added. In this example the links represent the products:  $\lambda_2^m \rightarrow \lambda_2^m + \lambda_{21}^\Psi \cdot \lambda_1^R$  and  $\lambda_2^m \rightarrow \lambda_2^m + \lambda_{22}^\Psi \cdot \lambda_2^R$ . (c) Additionally, “fixed” links between nodes and their counterpart’s children added. In this case, the links represents the products:  $\lambda_0^m \rightarrow \lambda_0^m + \lambda_{01}^\Psi \cdot \lambda_1^R$  and  $\lambda_0^m \rightarrow \lambda_0^m + \lambda_{02}^\Psi \cdot \lambda_2^R$ . (d) Matrix interpretation: Colors indicate the corresponding cases of (a) - (c). Non-filled rectangles indicates the further development of the tree-scheduling algorithm for the corresponding edge. For example; introducing a fixed link in (c) (purple color) the node on the output side will stay fixed while on the input side (right side) they will be propagating down. This results in an element-wise multiplication and summation along a row in the matrix vector representation. Due to illustration purpose, the scaling coefficient  $\lambda_0^\Psi$  is omitted. The level of the input tree is indicated by  $k$ .

by keeping track of the sum of the error bounds. If the ratio is smaller than a user specified constant the iteration is ended. The constant can be chosen empirically to achieve the desired approximation compared to the baseline method, see Section 8.7. Our actual implementation uses a simple absolute error stopping criterion, i.e., a link is not pushed to the queue if  $\tau^w < \tau_\epsilon$ , where  $\tau_\epsilon$  is chosen empirically.

The tree based message update algorithm presented here is just an adaptive, approximate matrix vector multiplication. It solves the matrix vector product element-wise and prefers to compute those products of elements first, which most contribute to the squared norm of the resulting vector. It is independent of the domain resolution. Solution gathering is stopped if there is no more input summands given, i.e., priority queue is empty. This is achieved when the desired signal approximation is reached. Therefore, we can work with an unbounded potential number  $N$  of wavelet coefficients and let the algorithm automatically choose a representation that in each step is accurate up to a fixed  $l_2$  error bound. In other words, the algorithm adaptively determines a hierarchical discretization on the fly rather than fixing it upfront. This automatic adaptation cannot be obtained with traditional algorithms that work on fixed grids. Non-parametric sampling methods [133] are resolution independent by design but available methods cannot guarantee strict error bounds.

For moderately sized state spaces, this property might not be necessary so that we can simplify our algorithm by performing the priority queue-driven scheduling for each row of the matrix-vector multiplication instead of globally. We refer to this approach as *pruned* WBP, indicating that we prune the solution on a row-base instead of a globally consistent error minimization. The scaling behavior of this method is, however, not as good, see Figure 8.6, since we then require at least  $\mathcal{O}(N)$  row-wise multiplications. For small-sized state spaces the runtime is quite a bit lower because the access to a sparse data structure storing the result coefficients, as in our full featured hierarchical multiplication, is avoided. The fully adaptive version uses a hash table to store coefficients while the row-wise version can use a simple array.

## 8.4 Conversion to the Wavelet Domain

The algorithm described above relies on the availability of certain information about  $\Phi_i$  and the  $\Psi_{ij}$ . So far, we have formulated this as an abstract oracle; now

we propose different implementations. The first type of information is the wavelet transform of these functions. Because the Haar-basis functions  $b_p$  (as well as the tensor-product basis function  $b_{pq}$ ) are orthogonal to each other, the transformation is given by just an inner product between the input function and a basis function. Because of the simple, piecewise constant form of the basis, this can be reduced to a sum of a small number of integrals over the input function. Again:

$$\lambda_{pq}^{\Psi_{ij}} = \sum_{\mathbf{x}_i, \mathbf{x}_j} \Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) b_{pq}(\mathbf{x}_i, \mathbf{x}_j) \quad (8.10)$$

$$= \sum_{x_i} \sum_{x_j} \Psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) b_p(\mathbf{x}_i) b_q(\mathbf{x}_j), \quad (8.11)$$

where  $b_{pq}(\mathbf{x}_i, \mathbf{x}_j)$  is a tensor product of both orthogonal basis functions  $b_p(\mathbf{x}_i)$  and  $b_q(\mathbf{x}_j)$ . Hence the main problem of computing of  $\lambda_{pq}^{\Psi_{ij}}$  reduces to the computation of an integral over a certain domain.

The second type of information we need are lower and upper bounds for  $\Psi_{ij}$ : For two quadtree-aligned squares  $\mathcal{X}, \mathcal{Y}$ , we need to be able to bound the minimum and maximum value of  $\Psi_{ij}(\mathbf{x}, \mathbf{y})$  for  $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$ .

We have evaluated three variants how to achieve both types of information in practice:

1. Full wavelet transformation of all (different)  $\Phi_i$  and  $\Psi_{ij}$  on a regular  $2^K \times 2^K$  grid for fixed  $K$ .
2. On-demand wavelet coefficients computation of  $\Psi$  utilizing its spatial symmetry.
3. Completely analytical solution.

For the first point, (1), we precomputed both the bounds as well as the wavelet transforms using a simple fast-wavelet transform as described in [131]. For large  $K$ , this is not feasible anymore, since the size of the  $\Psi$ -table becomes too large because of the quadratic growth with respect to  $\Phi$ -table. Hence we experimented with other two possibilities. (2) Given a pairwise compatibility function  $\Psi$  that is symmetric, i.e.,  $\Psi(x, y) = \Psi(y, x)$  and stationary with respect to  $\mathbf{x}$  or  $\mathbf{y}$ , i.e.,  $\Psi(x, y) = \Psi(x + t, y + t)$ , the space can be reduced by exploiting this redundancy as we will see next:

### Computing $\lambda_{pq}^\psi$ by exploiting symmetry redundancy

In case when  $\Psi_{ij}(\mathbf{x}, \mathbf{y})$  is symmetric and stationary we can compute its wavelet coefficient in worst-case running time of  $\mathcal{O}(\log n)$  for a 2D and  $\mathcal{O}(\log^2 n)$  for a 4D domain, where  $n = \sqrt{k}$  is the width of the discretization domain. The following algorithm, Algorithm 1, shows a computation of the integral in a 2D case over a given domain specified by  $x$  and  $y$  as the position and  $w, h$  as the size of the domain.

---

#### computeIntegral( $x, y, w, h$ )

---

```

if  $w = 0 \vee h = 0$  then
    return 0
end if

if  $y = 0 \wedge \text{inCache}(x, w, h)$  then
    return fromCache( $x, w, h$ )
end if

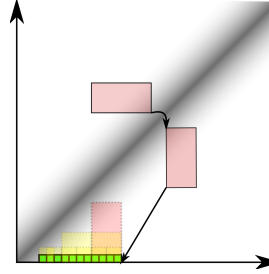
if  $y > 0$  then
    if  $y > x$  then
        swap( $x, y$ )
        swap( $w, h$ )
    end if
    return computeIntegral( $x - y, 0, w, h$ )
end if

sum = computeIntegral( $x, y, \frac{w}{2}, \frac{h}{2}$ ) + computeIntegral( $x + \frac{w}{2}, y, \frac{w}{2}, \frac{h}{2}$ ) +
      computeIntegral( $x, y + \frac{h}{2}, \frac{w}{2}, \frac{h}{2}$ ) + computeIntegral( $x + \frac{w}{2}, y + \frac{h}{2}, \frac{w}{2}, \frac{h}{2}$ )

if  $y = 0 \wedge w = h$  then
    putIntoCache(sum,  $x, w$ )
end if

return sum

```



**Algorithm 1** The figure illustrates a 2-dimensional pairwise compatibility function, which is symmetric and stationary. Each of the  $\Psi_{ij}$  functions (in this case Gauss-distribution) is sampled along the vertical axis, i.e. along its domain. A computation of a wavelet coefficients is proportional to an integration over the hyperdomain of the full  $\Psi$  function. The integral hyperrectangle  $(x, y, w, h)$  is first mirrored and projected to the x-axis. This is repeated recursively for each of the sub-rectangles (i.e. children). If cell is found in the cache, its value is used. For cells of the size  $1 \times 1$  (i.e. green cells) the function  $\Psi$  is evaluated directly. Yellow cells mark those added to the cache.

Since the extensions of the basis functions are always a power of two, the memory complexity of Algorithm 1 is bounded by  $\Theta(n \log n)$  for two-dimensional and  $\Theta(n^2 \log^2 n)$  for a four-dimensional case. It is straight forward to extend the algorithm to a four-dimensional case where one has to mirror the hyperrectangle

on the middle hyperplane and project it to the  $xy$ -plane. As caching criterion for the 4D case, one can set  $x = 0 \wedge y = 0 \wedge d_x = d_y \wedge d_z = d_w$ , where  $d_i$  is the extension of the hyperrectangle in dimension  $i$ , which is optimal with respect to the tensor product basis function  $b_{pq}$ .

Given that algorithm, however, a resolution free, i.e., independent of the resolution  $K$ , operation is not possible. Furthermore, the coefficient evaluation has a non-constant complexity and hence reduces the overall running time either. Hence in the following we assume that an analytical integration of  $\Psi$  as well as the computation of minimal and maximal values of  $\Psi$  over any interval is possible (3). We omit all precomputed tables and perform an analytic integration and bounding at runtime. An analytic solution of an integral is provided for many classes of functions  $\Psi$  (directly or through a piecewise Taylor approximation) but requires additional manual effort. In case, if no analytical solution is possible and the given pairwise potential  $\Psi$  is symmetric and stationary with respect to  $\mathbf{x}$  or  $\mathbf{y}$ , the proposed method (2) can be employed. In case, when the discretized domain has a smaller size  $K$  even a straightforward a priori wavelet transformation (1) might be sufficient. Figure 8.6 illustrates the running time complexity for different discretization levels.

## 8.5 Evaluation and Applications

### 8.6 Applications

We evaluate our method with two standard applications of LBP that typically require large state spaces: deformable image matching and wide-baseline optical flow [27, 128, 132]. In both cases, we are operating on a two-dimensional domain  $\Omega$ . As data term  $\Phi$ , we use Fourier descriptors [64] (compressed by PCA) that characterize local image content in a rotationally invariant way. The compatibility function  $\Psi$  differs for the two applications:

**Deformable image matching:** Here, we aim at preserving distances along the edges of the graph  $G$ , with increasing penalty for larger deviations. In other words, we are trying to keep the matching “as-rigid-as-possible”:

$$\Psi_{ij}^{def}(x_i, x_j) = \max \left( e^{-\beta(D_s(i,j) - D_t(x_i, x_j))^2}, \rho \right) \quad (8.12)$$

$d_s(i, j)$  and  $d_t(x_i, x_j)$  are Euclidean distances between source and target point

pairs,  $\beta$  determines how easily deformable the model is, and  $\rho$  is a truncation parameter that allows for discontinuities in the solution to make the matching robust.

**Optical flow:** Here, we use a regularizer that imposes a non-rotational invariant smoothness constraint [128]:

$$\Psi_{ij}^{Fd}(a, b) = \max \left( e^{-\beta(|a_x - b_x + d_x| + |a_y - b_y + d_y|)}, \rho \right) \quad (8.13)$$

$(a_x, a_y)$ ,  $(b_x, b_y)$  are the displacement vectors and  $(d_x, d_y)$  is one of the four stationary vectors  $(1, 0)^T$ ,  $(-1, 0)^T$ ,  $(0, 1)^T$ ,  $(0, -1)^T$ , depending on current direction of the message propagation in the MRF graph, respectively. When propagating a message from node  $i$  to node  $j$  in the MRF graph, we would like both nodes to have parallel displacement vectors, hence we need to move the highest response of the smoothness potential by a vector equal to the displacement between these both nodes, which is encoded by the stationary vector  $(d_x, d_y)$ . Again,  $\beta$  and  $\rho$  control the regularizer strength and robustness.

## 8.7 Baseline methods

We compare our approach with standard loopy belief propagation (LBP) as well with an optimized version, which we call *pruned* belief propagation (PBP). Pruned belief propagation employs a simple but very effective optimization: Many pairwise compatibility potentials  $\Psi$  are rather sparse. For a fixed  $\mathbf{y}$ ,  $\Psi(\mathbf{x}, \mathbf{y})$  is typically non-constant only for a small area  $\mathcal{A} \subseteq \Omega$ . In this case, we can restrict the evaluation of the integral in Equation 1.6 to  $\mathcal{A}$ . To account for the integral over constant  $\Psi$ , we compute the unweighted integral of  $\Phi$  over  $\mathcal{A}$  and compute the integral over  $\bar{\mathcal{A}}$  as complement to the integral over  $\Omega$ , at no extra costs. Then, we weight this value with the constant value  $\Psi$  attains in  $\bar{\mathcal{A}}$ . In practice, this easy to implement optimization typically leads to large speedups and is still exact compared to a standard belief propagation implementation. Therefore, it should be included for a fair comparison.

The comparison with WBP is based on PSNR (peak signal to noise ratio):

$$PSNR = 20 \cdot \log_{10} \left( \frac{1}{\sqrt{MSE}} \right) [\text{dB}]. \quad (8.14)$$

The mean squared error (MSE) is defined as

$$MSE = \frac{1}{kn} \sum_{i=0}^{n-1} \sum_{j=0}^{k-1} (f_i^w(x_k) - f_i(x_k))^2, \quad (8.15)$$

where  $f_i^w(x_k)$  is the belief of node  $i$  for the label  $x_k$  computed within the wavelet domain. In other words, mean-squared-error (MSE) is a sum of quadratic differences between belief vectors of the final result in the wavelet and the spatial domain. Here, we assume that our ground-truth is the solution computed by the standard belief propagation approach. A PSNR of 70dB<sup>1</sup>, which corresponds to a MSE of  $1e-7$ , is a good quality threshold for compressed marginals and will be further used in our experiments. We consider a signal as being *approximately equal* if the PSNR is more than 100dB with respect to the ground-truth.

## 8.8 Results

**Image matching:** Our first test is performing as-rigid-as-possible image matching on a strongly deformed example image. We use the first and fourth frame of the teddy image sequence [150] and additionally deformed the fourth frame by applying a spatial “sine-wave” filter, in order to stress the correspondence estimation. In this example, we use precomputed wavelet decompositions and row-based approximate summation (Section 8.3). Figure 8.3 shows the visual results. We are able to obtain a more than threefold speedup with WBP (1.25 hours), over the optimized PBP (4.8 hours), implementation at a PSNR of 95dB. The baseline standard LBP implementation would have used (estimated) 1200 hours for the same data set.

**Optical flow:** Our second test is optical flow, applied to the BeanBag data set from the Middlebury database [8]. The displacements in this experiment are in the range of 10-20% of the image size, which is a good test case for large displacement optical flow. Figure 8.7 shows the warped source to target image and corresponding optical flow. The color encoded displacement vectors computed with standard belief propagation are shown in Figure 8.7(d) and those computed with wavelet belief propagation in Figure 8.7(g). Please note that our PSNR error is computed over all belief vectors  $\mathbf{m}$  rather than the maximum label likelihood as shown in the resulting figures.

**Comparison against downsampling:** We have also performed an experiment for a 1D case (a 1D Markov chain with a 1D interval as state space) where the solution can be computed exactly. We run wavelet and standard BP at different resolutions to study how the error scales with the retained information. Our analytical

---

<sup>1</sup>Due to conversion of a signal into/from the wavelet domain we are limited by hardware double precision value representation, achieving at most a PSNR of 156dB.



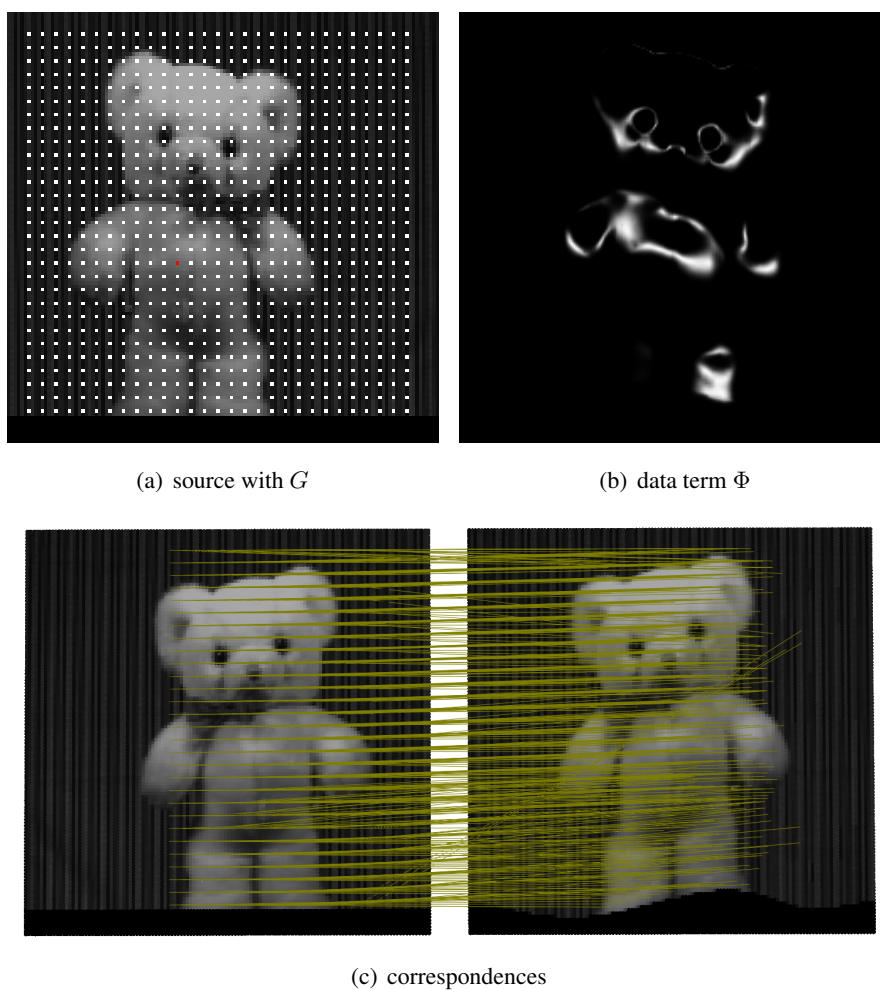


Figure 8.3: Teddy data set for the image matching experiment. (a) Source image; the  $29^2 = 841$  nodes of  $G$  are shown as white dots. State space:  $256^2 = 65536$  labels, i.e.  $128 \times 128$  resolution with half-pixel accuracy. (b) Fourier data term for the point on the teddy's belly marked in red. (c) Visualization of the correspondences.

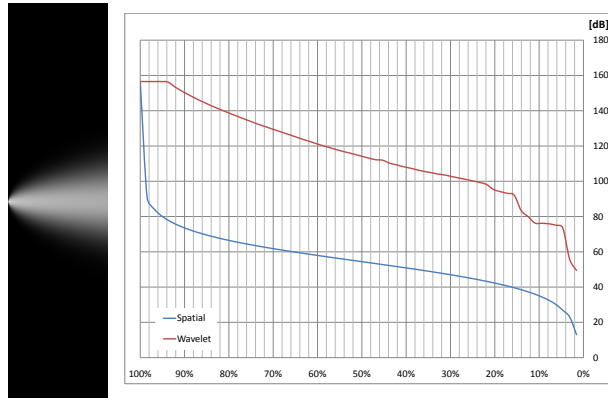


Figure 8.4: *Left*: Reference solution. *Right*: PSNR for the same percentage of coefficients, comparing LBP with simple binning (“spatial”) and a wavelet LBP algorithm.

example uses a  $\Phi$  that is a Dirac impulse for the first node and a constant function for all nodes  $i \neq 0$ . The regularizer consists of a Gaussian blur with a slight upward shift. The reference solution is computed with 1024 states. Figure 8.4 shows the error for the same number of coefficients used in the state space. The accuracy of the wavelet approximation drops much more gracefully; the PSNR is between two to four orders of magnitude better than that for simple downsampling.

**Scaling behavior:** We study one more wide-baseline optical flow example on a synthetic “lady bug” data set (see Figure 8.5). We vary the resolution of the discretization domain and compare the running times of standard LBP, pruned PBP, and wavelet BP. We use wavelet BP at 70dB and 100dB PSNR, compared against the *final results* of the reference LBP solution. For 70dB, we use both the precomputed and the analytical computation of the wavelet transform of  $\Psi$ , all other cases are precomputed. We also test the simplified row-based summation (Section 8.3), where very small entries in the precomputed table for  $\Psi$  are statically pruned by  $1e-15$ , i.e. floating point accuracy; the method is therefore referred to as “*WBP pruned*”) and achieves a PSNR of more than 110dB.

Figure 8.6 shows the running times for one iteration of belief propagation. As expected, the spatial domain implementations all scale quadratically (confirmed by a slope of 2 in the log-log plot). The row-based summation is very fast in absolute numbers, but scales slightly super-linearly (empirical slope 1.3). The full WPB implementations have the largest constant overhead factor (due to the more

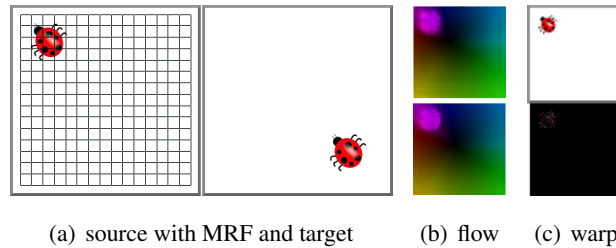


Figure 8.5: Synthetic example for time complexity evaluation. (a) Source and target images. The MRF consists of 256 nodes. (b) Flow fields computed with WBP at 70dB (top) and with PBP (bottom) at 4096 nodes and 4096 labels. The computation is 7min and 56min for WBP and PBP respectively. (c) Warped image computed with WBP at 70dB (top) and difference image to the source (bottom).

involved data structures), but show the best scaling behavior.

## 8.9 Summary

In this chapter we have introduced a new algorithm that performs sum-product belief propagation directly in a wavelet transformed representation. It represents all messages as a sparse linear combination of wavelet basis functions, which typically yields a much more concise representation than traditional histograms of labels. Our technique is general and applicable to pairwise MRFs with general potential functions. Unlike previous work our algorithm performs all operations in the wavelet domain only, thereby only requiring computational costs dependent on the information content in this representation rather than on the spatial resolution. In practice, we obtain a significantly better scaling behavior, however, at the price of a larger overhead (larger constant factor in the runtime). In absolute time, wavelet belief propagation outperforms even our optimized spatial domain base-line implementation for medium to large state spaces (depending on the chosen summation algorithm). In terms of quality, we obtain large signal-to-noise in comparison to the final marginals obtained from standard LBP.

**Limitations:** Probably the most important one is the larger constant in the runtime due to the more involved data structures. For small state spaces, the method does not offer advantages. The usage of wavelets also limits the domain of the state space; we need to be able to define an orthogonal Haar-basis on this domain. So

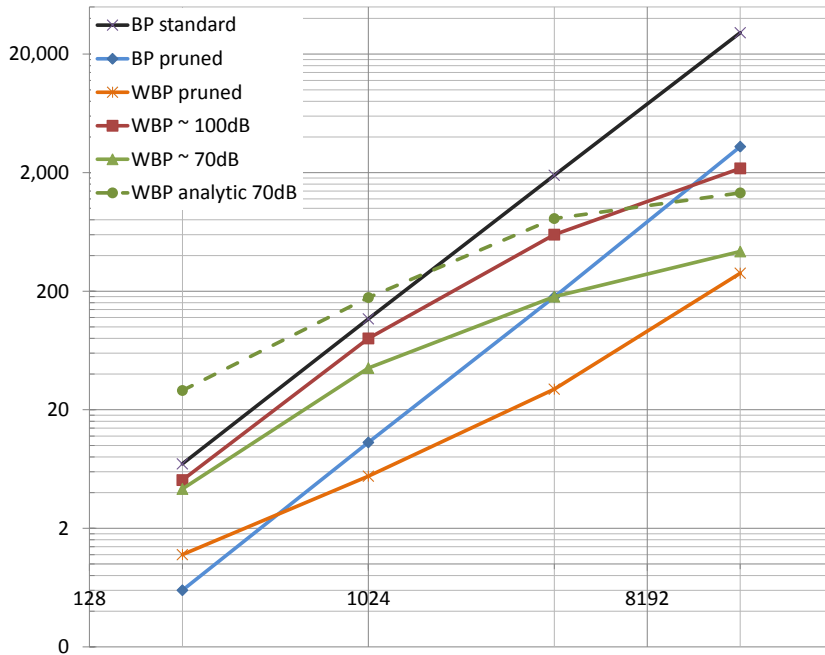


Figure 8.6: Execution time of one belief propagation iteration for different number of labels,  $16^2 = 256$  to  $128^2 = 16384$  labels. The number of nodes in the MRF is constant. Both axes of the diagram are logarithmic. Wavelet belief propagation scales differently and thus outperforms the spatial domain implementations for large state spaces.

far, we have only looked at square domains, but generalizations may be possible, i.e. by embedding an irregular domain into a square and pad the extra space with zeros. Another limitation is the use of sum-product belief propagation; the max-product algorithm cannot be supported. Finally, for very high and unlimited (purely adaptive) resolutions, some integral properties and bounds of  $\Psi$  must be derived analytically. A hybrid, partially precomputed technique with caching might reduce these issues.

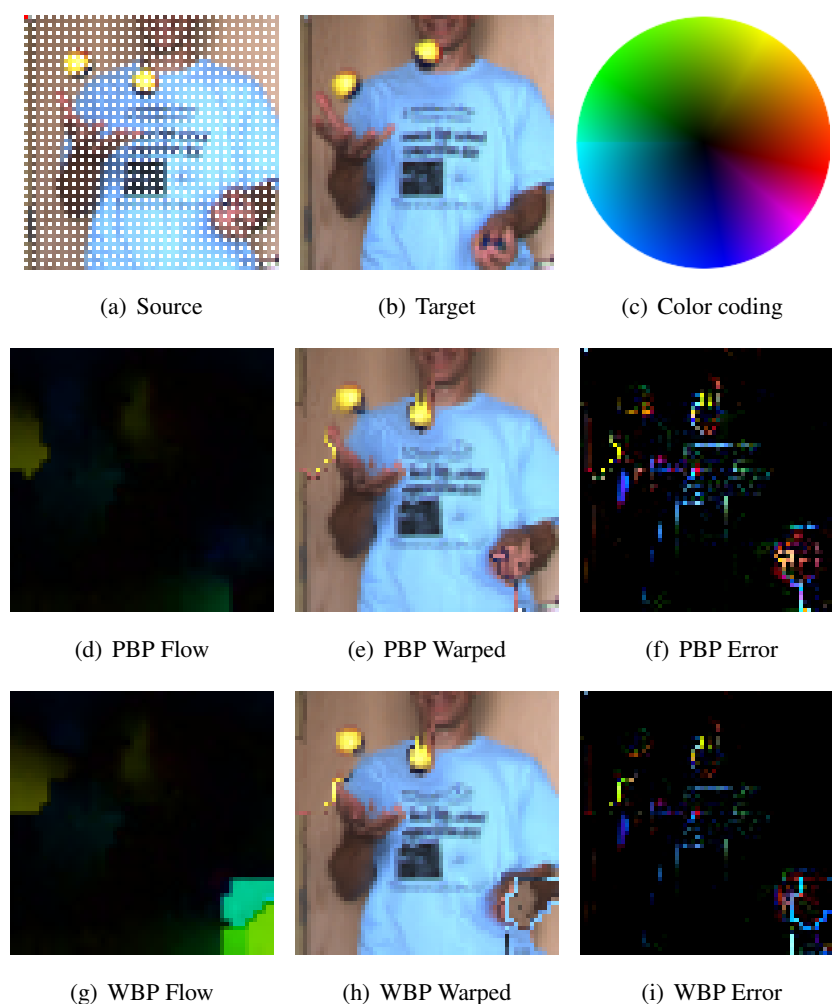


Figure 8.7: Results of the BeanBag data set. (e) and (h) Warped image computed with pruned BP and our tree-based wavelet BP. The MRF graph consists of 1024 nodes and is shown in (a). On the target side (b) we incorporate subpixel accuracy, by defining a label set of 16384 labels. (d) Flow computed with pruned BP in around 12 hours. (g) Flow computed with wavelet BP in 1.5 hours, the PSNR is 70dB. Number of iterations is 3. Please note that PSNR is computed over all marginal distributions at every node and might be different to the signal-to-noise ratio of peak marginals only used for flow encoding. This explains the large flow estimate in the right-bottom corner of the flow image. (c) Color coding used in our optical flow results. Note: The contrast in the error images has been enhanced for a more pleasant visualization.

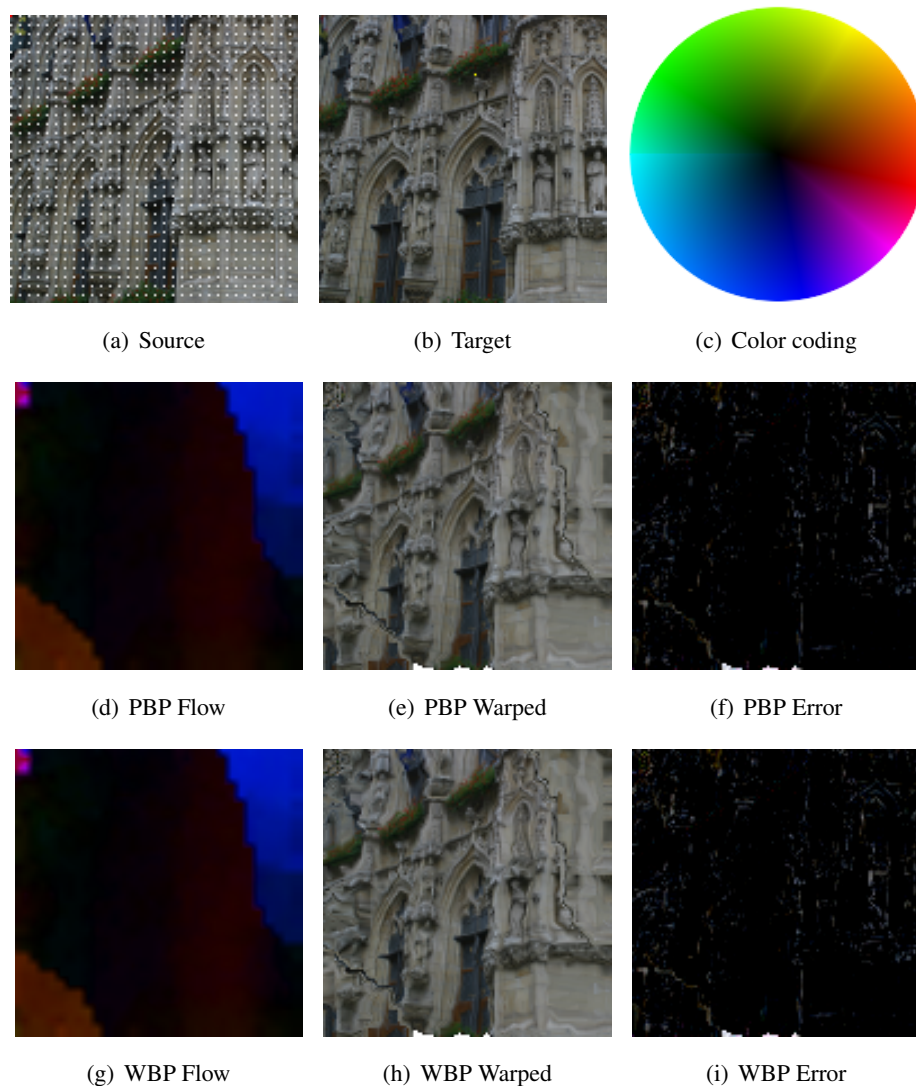


Figure 8.8: City hall Leuven data set (<http://cvlab.epfl.ch/data/strechamvs/>). (a) and (b) Source and target images. The number of MRF nodes is 1024 and the number of labels is 65536. Results were computed in 3 iterations. (d), (e) and (f) Flow, warped and difference image (i.e. difference between warped and source image) for the results computed with pruned BP. The computation time by pruned BP was 150 hours. (g), (h) and (i) Flow, warped and difference image of pruned wavelet BP (i.e. simplified row-based summation) computed in around 5 hours with a PSNR of 102dB. Note: The contrast in the error images has been enhanced for a more pleasant visualization.



## Part Summary

---

In this part of the thesis we have introduced two approaches. In the first case, Chapter 7, we have seen that deformable shape matching can be expressed as an inference problem on a Markov random field (MRF) and solved with the help of Loopy Belief Propagation (LBP). We derived a solution for symmetry detection, which is in particular the same as computing a correspondence of the shape to itself. Our results proved that the problem definition is sound and can be used alternatively to the common approach of feature point based shape matching as used in the previous part of the dissertation.

Looking at the results achieved in Chapter 7, the number of states/labels each variable in such MRF can take is becoming very large. Even worse, there rarely exists a theoretical or practical solution for solving MRFs with tremendous state spaces. In Chapter 8 we have developed an algorithm which is capable of computing an inference with LBP entirely formulated in the wavelet domain. The space-time complexity of LBP in the wavelet domain became data-driven and has in the worst-case the same complexity as standard LBP. We have seen that compared to standard and optimized versions of belief propagation, the newly developed WBP was able to achieve speedups in the order of several magnitudes. An evaluation on a synthetic examples showed that the running time complexity of wavelet belief propagation (WBP) is even sub-linear in case of a highly peaked marginal distribution, which is not that uncommon for many tasks where LBP is commonly used.

In general, we believe that the theoretical foundation made with the development of wavelet belief propagation approach will find their use in other research fields dealing with MRF optimizations.





## **Part IV**

# **Epilogue**



# EPILOGUE

---



In the first half of the thesis, Chapter 3, a correspondence estimation algorithm based on the idea of RANSAC has been shown. A robustness score, which incorporates only a subset of witness geodesics, make the shape matching algorithm resistant to topological noise. Utilizing an importance sampling function, the randomized search for correct correspondences has been biased towards a more promising solution. An additional tangential optimization technique reduces the overall correspondence error and improve the deformation function with respect to isometry. Real-world data demonstrate the robustness of the algorithm and its capability to find an approximation of the deformation function between two given pieces of geometry more densely than it was possible before. In general, the proposed algorithm is capable of computing correspondences between two geometry pieces acquired with 3D scanner devices in a robust and efficient way. This provides an important building block for solving deformable shape matching problems.

In Chapter 4, we took a closer look at the problem of shape matching and proposed an improvement of the existing RANSAC matching by incorporating an a-priori planning step. A so-called plan represented by a set of landmark points distributed over the surface has been introduced. These landmarks are automati-

---

cally placed in a way, that the numerical complexity of the shape matching process is reduced. In this case, the complexity of the shape matching is measured as the number of RANSAC samples required to compute a valid solution. Furthermore it has been empirically shown that for different types of shapes there exist different numbers and constellations of landmark points in the plan. This hints at a possible characterization of a shape's intrinsic complexity by counting the number of plan points required to reduce the problem of self-matching to a minimum. Summarized, the PLANSAC scheme, has a much wider applicability than shape matching. Many related areas in computer graphics and computer vision use random sampling based procedures for an abundance of tasks. Introducing a planning step might result in significant performance improvements similar to those demonstrated in this thesis.

In Chapter 5, using the concept of robust correspondence estimation and landmark coordinates, a method for animation reconstruction has been derived. Here, the goal is to reconstruct a shape and its motion from a given set of unordered and uncorrelated dynamic point clouds. The concept of landmark coordinates, shown in Chapter 4, has been reinforced with respect to topological noise. Incorporating the RANSAC-like correspondence estimation from Chapter 3 the algorithm is able to estimate correspondences even between frames with large temporal gaps. Point clouds acquired with real-time 3D scanners or by other scanning approaches like multiview-stereo reconstruction has been used in order to demonstrate the real-world scenarios. Although the problem of animation reconstruction is very complex, the algorithm is still able to achieve very good results without any a priori given template model. The results show that the reconstruction of shape and motion based on geometric information only, as shown in Chapter 5, is possible and acts as solid grounding for the research in this area.

In the third part of this thesis, Chapter 7, the problem of shape matching has been observed from another perspective. It has been redefined as a Markov random field (MRF) where each node describes a point on the source shape. The goal is to find the best inference over the label set, which is described by the target points. The formulation of the problem as MRF provides one with the benefit of being able to find all local, partial symmetries in the shape. This is in stark contrast to the methods shown in the previous chapters, which, if used as symmetry detectors, would only be able to find global symmetries, i.e. mappings of the complete shape onto itself. The formulation as a MRF also offers another interesting perspective:

---

just by varying one parameter the algorithm is able to find either more local or more global symmetries. The ability to find local symmetries as also global shape matchings has been demonstrated on various shapes. The inference in the MRF has been computed with loopy belief propagation (LBP). The ability of computing partial symmetries or matches in a shape provides a large step towards automatic understanding and classification of objects as seen by a machine.

One big disadvantage of any belief propagation approach is the relatively high amount of memory and running time required to estimate the inference. Since the model is “managing” a large amount of possible states for each variable, the performance drops very quickly. Equipped with the strong wish to solve that problem an approach for an optimized version of belief propagation, namely *wavelet belief propagation*, has been shown in Chapter 8. The marginal distribution and all computations have been moved into the wavelet domain. The transformation into the wavelet domain dramatically reduces the space-time complexity for specific types of problems. Experimenting with computer vision tasks, such as image matching and optical flow, the use of the wavelet based belief propagation approach facilitates not only theoretical but also a practical performance advantage over the existing methods. The advantage of an efficient inference estimation provides an important building block for other research fields which employ belief propagation.

Summarized as a simple list, the main contributions of this dissertation are:

- a RANSAC-like approach for robust correspondence estimation between a pair of point clouds.
- PLANSAC as a new concept for shape matching by incorporating a novel a-priori planning step which reduces the subsequent correspondence estimation to a minimum.
- a method known as animation reconstruction for reconstructing the shape and the motion of a moving deformable object from a set of unordered point clouds as obtained by a dynamic 3D scanner. The method is able to deal with large temporal and spatial gaps in the data.
- a redefinition of shape matching as a MRF problem and a solution using belief propagation, providing the ability to detect even partial matches as for example required for self-symmetry detection.
- an optimization of belief propagation by performing the computation in the

---

wavelet domain taking a large step towards resolution independent, i.e. continuous, belief propagation inference computation.

## 9.1 Discussion and future work

To the end, I would like to discuss the main essence of the contribution made in this dissertation and what remains to be done in the future research.

The problem of shape matching has evolved over several years. Right in the beginning, research was interested in the case of rigid shape matching. I believe that this problem can be considered as solved nowadays, since it has been widely adopted in the industry and is relatively easy to solve with ICP related approaches, assuming a good initial alignment is given. Besides that, one can imagine a simple RANSAC algorithm which estimates the initial shape alignment with respect to the reference frame by computing the transformation matrices based on a set of three or four matched descriptors.

Matching shapes, which undergo strong deformation, however, can still not be considered as solved on its own. In this dissertation I have shown several approaches for the case of isometric deformation. Here the main assumption is that the shape's surface does not strongly deform in elastic way hence preserving a distance between any two points on the surface. This assumption is well founded as discussed in the beginning of the thesis. Of course this assumption does not generalize to all instances of deformable shapes and hence can not be considered as the answer to everything. However, I believe that there exists a large set of problems which are perfectly reflected in this assumption. My personal opinion is that in the next years we have to try to understand the complexity of the shape matching problem first, before implementing any algorithm trying to solve it. The contribution proposed in Chapter 4 is one of the steps towards the comprehension of the problem. This idea has now been taken up by Ovsjanikov and colleagues [98] where the authors derive a so called *condition number* which maps the difficulty of shape matching for different types of shapes to a single number as has been hinted in this thesis.

I think, as soon as we understand the problem we are trying to solve in the possible best way, the solution will be obvious.

If we knew what it was we  
were doing, it would not be  
called research, would it?

---

*Albert Einstein (1879-1955)*





## BIBLIOGRAPHY

---

- [1] ADAMY, U., GIESEN, J., AND JOHN, M. New techniques for topologically correct surface reconstruction. In *Visualization 2000. Proceedings* (2000), pp. 373–380. [29](#)
- [2] AHMED, N., THEOBALT, C., ROSSL, C., THRUN, S., AND SEIDEL, H.-P. Dense correspondence finding for parametrization-free animation reconstruction from video. *Proc. of CVPR* (June 2008), 1–8. [47](#), [52](#), [83](#)
- [3] ALEXA, M., BEHT, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. Computing and rendering point set surfaces. *IEEE Trans. Visualization and Comp. Graphics* 9, 1 (2003), 3–15. [69](#)
- [4] ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. Articulated body deformation from range scan data. *ACM Trans. Graph.* 21, 3 (2002), 612–619. [52](#)
- [5] ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM, pp. 587–594. [45](#), [52](#), [117](#)
- [6] ANGUELOV, D., SRINIVASAN, P., PANG, H.-C., KOLLER, D., THRUN, S., AND DAVIS, J. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Proc. of NIPS* (2004), pp. 33–40. [33](#), [52](#), [54](#), [136](#), [141](#)
- [7] ANUAR, N., AND GUSKOV, I. Extracting animated meshes with adaptive motion estimation. In *Proc. of VMV* (2004), pp. 63–71. [52](#), [98](#)
- [8] BAKER, S., SCHARSTEIN, D., LEWIS, J. P., ROTH, S., BLACK, M. J., AND SZELISKI, R. A database and evaluation methodology for optical flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (oct. 2007), pp. 1–8. [168](#)
- [9] BELONGIE, S., MORI, G., AND MALIK, J. Matching with shape contexts. In *IEEE Workshop on Content-based access of Image and Video-Libraries* (2000), p. 20. [31](#)

## Bibliography

---

- [10] BÉRARD, P., BESSON, G., AND GALLOT, S. Embedding riemannian manifolds by their heat kernel. *Geometric And Functional Analysis* 4, 4 (1994), 373–398. [48](#), [51](#)
- [11] BERNARDINI, F., AND RUSHMEIER, H. E. The 3d model acquisition pipeline. *Computer Graphics Forum* 21 (2002), 149–172. [28](#)
- [12] BESL, P. J., AND MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (February 1992), 239–256. [30](#), [45](#)
- [13] BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.* 26, 3 (2007), 33–41. [53](#)
- [14] BICKSON, D., YOM-TOV, E., AND DOLEV, D. A gaussian belief propagation solver for large scale support vector machines. In *in 5th European Conference on Complex Systems* (Oct 2008). [146](#)
- [15] BOKELOH, M. Feature based global registration of point cloud. Master’s thesis, GRIS, Uni-Tübingen, 2007. [29](#)
- [16] BOKELOH, M., BERNER, A., WAND, M., SCHILLING, A., AND SEIDEL, H.-P. Slippage features. Tech. Rep. WSI-2008-03, University of Tübingen, WSI/GRIS, 2008. [47](#), [64](#)
- [17] BOLLE, R. M., AND VEMURI, B. C. On three-dimensional surface reconstruction methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 13, 1 (jan 1991), 1–13. [29](#)
- [18] BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. High resolution passive facial performance capture. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH ’10, ACM, pp. 41–51. [52](#)
- [19] BRADLEY, D., POPA, T., SHEFFER, A., HEIDRICH, W., AND BOUBEKEUR, T. Markerless garment capture. *ACM Trans. Graph.* 27, 3 (2008), 99. [48](#)
- [20] BRONSTEIN, A. M., BRONSTEIN, M. M., CARMON, Y., AND KIMMEL, R. Partial similarity of shapes using a statistical significance measure. *IPSPJ Transactions on Computer Vision and Applications* 1 (2009), 105–114. [51](#)

## Bibliography

---

- [21] BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.* 28, 5 (2006), 1812–1836. [55](#), [137](#)
- [22] BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *In PNAS* 103, 5 (2006), 1168–1172. [33](#), [46](#), [52](#)
- [23] BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Tosca dataset. <http://tosca.cs.technion.ac.il/data/nonrigid3d.zip>, 2006. [Online; accessed 20-June-2010]. [47](#), [49](#), [88](#)
- [24] BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Topology-invariant similarity of nonrigid shapes. *Intl. Journal of Computer Vision (IJCV)* 81, 3 (2009), 281–301. [48](#)
- [25] BRONSTEIN, A. M., BRONSTEIN, M. M., KIMMEL, R., MAHMOUDI, M., AND SAPIRO, G. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Int. J. Comput. Vision* 89 (September 2010), 266–286. [48](#)
- [26] BROWN, B., AND RUSINKIEWICZ, S. Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 26, 3 (2007). [40](#), [45](#)
- [27] BROX, T., BRUHN, A., PAPPENBERG, N., AND WEICKERT, J. High accuracy optical flow estimation based on a theory for warping. *In ECCV* (Prague, Czech Republic, May 2004), T. Pajdla and J. Matas, Eds., vol. 3024 of *LNCS*, Springer, pp. 25–36. [166](#)
- [28] BRUNTON, A., SHU, C., AND ROTH, G. Belief propagation on the gpu for stereo vision. *In Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 76–. [146](#), [147](#)
- [29] CARRANZA, J., THEOBALT, C., MAGNOR, M. A., AND SEIDEL, H.-P. Free-viewpoint video of human actors. *In ACM Trans. Graph.* (2003), pp. 569–577. [52](#), [98](#)

## Bibliography

---

- [30] CHANG, W., AND ZWICKER, M. Automatic registration for articulated shapes. *Computer Graphics Forum (Proc. of SGP)* 27, 5 (2008), 1459–1468. [46](#), [52](#)
- [31] CHANG, W., AND ZWICKER, M. Range scan registration using reduced deformable models. *Computer Graphics Forum (Proc. Eurgraphics)* 28, 2 (2009), 447–456. [46](#)
- [32] CHUI, H., AND RANGARAJAN, A. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* 89, 2-3 (2003), 114–141. [46](#)
- [33] CHUM, O., AND MATAS, J. Matching with PROSAC - progressive sample consensus. In *Proc. of CVPR* (2005), vol. 1, pp. 220–226. [49](#), [67](#)
- [34] DAVIS, J., NEHAB, D., RAMAMOORTHY, R., AND RUSINKIEWICZ, S. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE PAMI* 27, 2 (2005), 296–302. [52](#)
- [35] DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. Performance capture from sparse multi-view video. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 98–108. [52](#), [98](#)
- [36] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. [10.1007/BF01386390](#). [36](#)
- [37] DIJKSTRA, E. W. A note on two problems in connexion with graphs. In *Numerische Mathematik*, vol. 1. Mathematisch Centrum, Amsterdam, The Netherlands, 1959, pp. 269–271. [100](#)
- [38] DORAI, C., AND JAIN, A. K. Cosmos - a representation scheme for 3d free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1995), 1115–1130. [143](#)
- [39] DURBIN, R., CENTRE, S., EDDY, S. R., KROGH, A., AND MITCHISON, L. G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998. [40](#)

## Bibliography

---

- [40] ELDAR, Y., LINDENBAUM, M., PORAT, M., AND ZEEVI, Y. Y. The farthest point strategy for progressive image sampling. *IEEE Trans Image Process* 6, 9 (1997), 1305–1320. [48](#)
- [41] FANG, Q., GAO, J., GUIBAS, L. J., SILVA, V., AND ZHANG, L. Glider: Gradient landmark-based distributed routing for sensor networks. In *in Proc. of the 24th Conference of the IEEE Communication Society (INFOCOM)* (2005), pp. 339–350. [83](#)
- [42] FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. Efficient belief propagation for early vision. *Int. J. Comput. Vision* 70 (October 2006), 41–54. [55](#)
- [43] FISCHLER, M. A., AND BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24 (June 1981), 381–395. [22](#), [49](#)
- [44] GATZKE, T., GRIMM, C., GARLAND, M., AND ZELINKA, S. Curvature maps for local shape comparison. In *Proc. Shape Modeling International* (2005), pp. 244–256. [34](#), [143](#)
- [45] GELFAND, N., AND GUIBAS, L. J. Shape segmentation using local slip-pipe analysis. In *Proc. of SGP* (2004), pp. 214–223. [105](#)
- [46] GELFAND, N., MITRA, N. J., GUIBAS, L. J., AND POTTMANN, H. Robust global registration. In *Proc. of SGP* (2005), pp. 197–206. [46](#), [64](#), [79](#)
- [47] GOLDBERGER, J., AND KFIR, H. Serial schedules for belief-propagation: Analysis of convergence time. *Information Theory, IEEE Transactions on* 54, 3 (March 2008), 1316–1319. [145](#), [146](#)
- [48] GROMOV, M. *Metric structures for Riemannian and non-Riemannian spaces*. Birkhäuser, 1999. [50](#)
- [49] HÄHNEL, D., THRUN, S., AND BURGARD, W. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)* (2003), pp. 915–920. [52](#), [117](#), [118](#)

## Bibliography

---

- [50] HAN, T. X., NING, H., AND HUANG, T. S. Efficient nonparametric belief propagation with application to articulated body tracking. *CVPR I* (2006), 214–221. [56](#), [155](#)
- [51] HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. of SIGGRAPH '01* (2001), pp. 203–212. [48](#)
- [52] HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), SIGGRAPH '92, ACM, pp. 71–78. [29](#)
- [53] HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A* 4, 4 (1987), 629–642. [113](#)
- [54] HUANG, P., HILTON, A., AND BUDD, C. Global temporal registration of multiple non-rigid surface sequences. In *Proc. of CVPR* (2011), p. to appera. [53](#)
- [55] HUANG, Q.-X., ADAMS, B., WICKE, M., AND GUIBAS, L. J. Non-rigid registration under isometric deformations. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), SGP '08, Eurographics Association, pp. 1449–1457. [33](#), [47](#), [52](#), [67](#), [80](#), [82](#), [83](#)
- [56] HUANG, Q.-X., FLÖRY, S., GELFAND, N., HOFER, M., AND POTTMANN, H. Reassembling fractured objects by geometric matching. *ACM Trans. Graphics* 25, 3 (2006), 569–578. [50](#), [64](#), [67](#)
- [57] IKEMOTO, L., GELFAND, N., AND LEVOY, M. A hierarchical method for aligning warped meshes. In *Proc. Fourth International Conference on 3D Digital Imaging and Modeling* (2003). [45](#)
- [58] ISARD, M. Pampas: real-valued graphical models for computer vision. In *Proceedings of the 2003 IEEE computer society conference on Computer vision and pattern recognition* (Washington, DC, USA, 2003), CVPR'03, IEEE Computer Society, pp. 613–620. [56](#), [155](#)

## Bibliography

---

- [59] JANSEN, S. Symmetry detection in images using belief propagation. Master's thesis, Saarland University, Germany, 2010. [151](#)
- [60] JOHNSON, A. E., AND HEBERT, M. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (1999), 433–449. [34](#), [142](#)
- [61] JOLLIFFE, I. T. *Principal Component Analysis*. Springer, 2000. [87](#)
- [62] JONES, P. W., MAGAGIONI, M., AND SCHUL, R. Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels. In *PNAS* (2008), pp. 1803–1808. [48](#), [81](#)
- [63] JONES, P. W., MAGGIONI, M., AND SCHUL, R. Universal local parametrizations via heat kernels and eigenfunctions of the laplacian. In *Ann. Acad. Scient. Fen.* (2010), pp. 1–44. [51](#)
- [64] KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 156–164. [166](#)
- [65] KIM, V. G., LIPMAN, Y., CHEN, X., AND FUNKHOUSER, T. Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Proc. of Symposium on Geometry Processing)* 29, 5 (2010). [49](#), [55](#)
- [66] KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. Blended intrinsic maps. *Transactions on Graphics (Proc. of SIGGRAPH 2011)* (2011), to appear. [48](#)
- [67] KIMMEL, R., AND SETHIAN, J. A. Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA* (1998), pp. 8431–8435. [36](#)
- [68] KINDERMANN, R., AND SNELL, J. L. *Markov Random Fields and Their Applications*. AMS, 1980. [36](#)
- [69] KOMODAKIS, N., AND TZIRITAS, G. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing* 16, 11 (November 2007), 2649–2661. [55](#)

## Bibliography

---

- [70] KÖNIG, S., AND GUMHOLD, S. Image-based motion compensation for structured light scanning of dynamic scenes. *Int. J. of Int. Sys. Tech. App.* 5, 3/4 (2008), 434 – 441. [52](#)
- [71] LASOWSKI, R., TEVS, A., SEIDEL, H.-P., AND WAND, M. A probabilistic framework for partial intrinsic symmetries in geometric data. In *Proc. of International Conference on Computer Vision (ICCV)* (Koyoto, Japan, 2009), pp. 963–970. [24](#), [43](#)
- [72] LASOWSKI, R., TEVS, A., WAND, M., AND SEIDEL, H.-P. Wavelet belief propagation for large scale inference problems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)* (2011), IEEE, pp. 1921–1928. [25](#), [43](#)
- [73] LEICA GEOSYSTEMS. Leica geosystems introduces a new class of laser scanner: Leica scanstation. [http://hds.leica-geosystems.com/en/Press-Releases\\_5604.htm](http://hds.leica-geosystems.com/en/Press-Releases_5604.htm). [28](#)
- [74] LEORDEANU, M., AND HEBERT, M. A spectral technique for correspondence problems using pairwise constraints. In *Proc. of ICCV* (2005), vol. 2, pp. 1482–1489. [47](#), [67](#), [75](#), [92](#), [93](#)
- [75] LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.* 28 (December 2009), 175:1–175:10. [52](#), [53](#), [98](#), [115](#), [118](#), [122](#), [123](#)
- [76] LI, H., LUO, L., VLASIC, D., PEERS, P., POPOVIĆ, J., PAULY, M., AND RUSINKIEWICZ, S. Temporally coherent completion of dynamic shapes. *ACM Transactions on Graphics, TOG* (2011), to appear. [52](#)
- [77] LI, H., SUMNER, R. W., AND PAULY, M. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP)* 27, 5 (2008), 1421–1430. [13](#), [45](#), [46](#)
- [78] LI, S. Z. On discontinuity-adaptive smoothness priors in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 6 (1995), 576–586. [143](#)
- [79] LI, X., AND GUSKOV, I. Multiscale features for approximate alignment of point-based surfaces. In *Proc. of SGP* (2005), pp. 217–226. [34](#), [64](#), [79](#)



- [80] LIPMAN, Y., AND FUNKHOUSER, T. Möbius voting for surface correspondence. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 72:1–72:12. [49](#), [50](#), [52](#), [55](#), [79](#), [80](#), [91](#)
- [81] LUCAS, B. D., AND KANADE, T. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence* (1981), pp. 674–679. [105](#), [106](#)
- [82] MEMOLI, F., AND SAPIRO, G. Distance functions and geodesics on point clouds. IMA Preprint <http://purl.umn.edu/3856>, 2003. [36](#)
- [83] MEMOLI, F., AND SAPIRO, G. Comparing point clouds. In *Proc. of SGP* (2004), pp. 32–40. [46](#), [50](#)
- [84] MÉMOLI, F., AND SAPIRO, G. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. Comput. Math.* 5, 3 (2005), 313–347. [33](#), [50](#)
- [85] MESA IMAGING. <http://www.mesa-imaging.ch/>. [28](#), [122](#)
- [86] MINKA, T. P. Expectation propagation for approximate bayesian inference. In *Conf. on Uncertainty in Artificial Intelligence* (2001), pp. 362–369. [55](#)
- [87] MITRA, N. J., FLORY, S., OVSJANIKOV, M., GELFAND, N., GUIBAS, L. J., AND POTTMANN, H. Dynamic geometry registration. In *Proc. of SGP* (2007), pp. 173–182. [41](#), [47](#), [52](#), [97](#)
- [88] MITRA, N. J., GELFAND, N., POTTMANN, H., AND GUIBAS, L. J. Registration of point cloud data from a geometric optimization perspective. In *Proc. of SGP* (2004), pp. 23–31. [45](#)
- [89] MITRA, N. J., GUIBAS, L. J., AND PAULY, M. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.* 25, 3 (2006), 560–568. [54](#), [137](#)
- [90] MITRA, N. J., GUIBAS, L. J., AND PAULY, M. Symmetrization. *ACM Trans. Graph.* 26, 3 (2007), 63–. [54](#), [137](#)
- [91] MOENNING, C., AND DODGSON, N. A. Fast marching farthest point sampling for implicit surfaces and point clouds. Tech. rep., University of Cambridge, 2003. [46](#), [87](#)

## Bibliography

---

- [92] MOTWANI, R., AND RAGHAVAN, R. *Randomized Algorithms*. Cambridge University Press, 1995. [65](#)
- [93] MURPHY, K. P., WEISS, Y., AND JORDAN, M. I. Loopy belief propagation for approximate inference: An empirical study. In *In Proceedings of Uncertainty in AI* (1999), pp. 467–475. [151](#)
- [94] NG, R., RAMAMOORTHY, R., AND HANRAHAN, P. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph.* 23 (August 2004), 477–487. [139](#), [156](#), [157](#)
- [95] NVIDIA. *NVIDIA CUDA Programming Guide 2.0*. NVIDIA Corporation, 2008. [147](#)
- [96] OVSJANIKOV, M., MÉRIGOT, Q., MÉMOLI, F., AND GUIBAS, L. J. One point isometric matching with the heat kernel. In *Proc. of SGP* (2010), pp. 1555–1564. [50](#), [79](#), [80](#)
- [97] OVSJANIKOV, M., SUN, J., AND GUIBAS, L. Global intrinsic symmetries of shapes. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), SGP '08, Eurographics Association, pp. 1341–1348. [54](#), [137](#)
- [98] OVSJANIKOV, M., XING HUANG, Q., AND GUIBAS, L. J. A condition number for non-rigid shape matching. In *Eurographics Symposium on Geometry Processing (SGP)* (2011), p. to appear. [81](#), [182](#)
- [99] PARK, M., BROCKLEHURST, K., COLLINS, R. T., AND LIU, Y. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Trans. PAMI* 31, 10 (2009), 1804–1816. [56](#), [155](#)
- [100] PARK, M., COLLINS, R., AND LIU, Y. Deformed lattice discovery via efficient mean-shift belief propagation. In *European Conference on Computer Vision (ECCV)* (2008), pp. 474–485. [54](#), [137](#)
- [101] PARK, M., LEE, S., CHUN CHEN, P., KASHYAP, S., BUTT, A. A., AND LIU, Y. Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition* (Los Alamitos, CA, USA, 2008), pp. 1–8. [54](#), [137](#)

## Bibliography

---

- [102] PARK, S. I., AND HODGINS, J. K. Capturing and animating skin deformation in human motion. *ACM Trans. Graph.* 25, 3 (2006), 881–889. [52](#), [98](#)
- [103] PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. Example-based 3d scan completion. In *Proceedings of the third Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2005), Eurographics Association. [52](#)
- [104] PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. J. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics* 27, 3 (2008), #43, 1–11. [54](#), [137](#)
- [105] PEARL, J. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second National Conference on Artificial Intelligence. AAAI'82* (Pittsburgh, PA. Menlo Park, California, 1982), AAAI Press, pp. 133–136. [24](#), [38](#), [55](#)
- [106] PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, vol. 2. Morgan Kaufmann, 1988. [55](#), [151](#)
- [107] PEKELNY, Y., AND GOTSMAN, C. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum* 27, 2 (Apr. 2008), 399–408. [53](#), [97](#)
- [108] POPA, T., SOUTH-DICKINSON, I., BRADLEY, D., SHEFFER, A., AND HEIDRICH, W. Globally consistent space-time reconstruction. *Computer Graphics Forum (Proc. SGP)* 29, 5 (2010), 1633–1642. [52](#)
- [109] POTETZ, B. Efficient belief propagation for vision using linear constraint nodes. In *CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Minneapolis, MN, USA, 2007. [56](#)
- [110] RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, 2 (feb 1989), 257–286. [38](#), [40](#)

## Bibliography

---

- [111] RAVIV, D., BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Symmetries of non-rigid shapes. In *International Conference on Computer Vision (ICCV)* (2007), pp. 1–7. [55](#), [137](#)
- [112] REUTER, M., BIASOTTI, S., GIORGI, D., PATANÉ, G., AND SPAGNUOLO, M. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers & Graphics* 33 (2009), 381–390. [48](#)
- [113] RUGGERI, M. R., PATANÉ, G., SPAGNUOLO, M., AND SAUPE, D. Spectral-driven isometry-invariant matching of 3d shapes. *IJCV* 89 (2010), 248–265. [33](#), [48](#), [52](#)
- [114] RUGGERI, M. R., AND SAUPE, D. Isometry-invariant matching of point set surfaces. In *Proc. of EG 3DOR* (2008). [33](#), [52](#)
- [115] RUSINKIEWICZ, S., AND LEVOY, M. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling* (2001), pp. 145–152. [45](#)
- [116] RUSSELL, S. J., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. [37](#)
- [117] SAGAWA, R., AKASAKA, K., YAGI, Y., HAMER, H., AND VAN GOOL, L. Elastic convolved icp for the registration of deformable objects. In *Computer Vision Workshops (ICCV Workshops)* (4 2009), pp. 1558–1565. [45](#)
- [118] SAGAWA, R., OSAWA, N., AND YAGI, Y. Deformable registration of textured range images by using texture and shape features. In *3-D Digital Imaging and Modeling, 2007. 3DIM '07. Sixth International Conference on* (aug. 2007), pp. 65–72. [46](#)
- [119] SAHILLIOĞLU, Y., AND YEMEZ, Y. 3d shape correspondence by isometry-driven greedy optimization. In *Electronic Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)* (San Francisco, CA, USA, 2010), pp. 1–8. [46](#)
- [120] SAND, P., MCMILLAN, L., AND POPOVIĆ, J. Continuous capture of skin deformation. *ACM Trans. Graph.* 22, 3 (2003), 578–586. [52](#), [98](#)
- [121] SCHMID, C., MOHR, R., AND BAUCKHAGE, C. Comparing and evaluating interest points. In *Proc. of ICCV* (Jan. 1998), pp. 230–235. [51](#)

## Bibliography

---

- [122] SHANNON, C. E. Prediction and entropy of printed english. *Bell Systems Technical Journal* (1951), 50–64. [85](#), [142](#)
- [123] SHARF, A., ALCANTARA, D. A., LEWINER, T., GREIF, C., SHEFFER, A., AMENTA, N., AND COHEN-OR, D. Space-time surface reconstruction using incompressible flow. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* (2008). [41](#)
- [124] SHARF, A., ALCANTARA, D. A., LEWINER, T., GREIF, C., SHEFFER, A., AMENTA, N., AND COHEN-OR, D. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph.* 27, 5 (2008), 110. [53](#)
- [125] SHARMA, A., HORAUD, R. P., CECH, J., AND BOYER, E. Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition* (Colorado Springs, USA, 2011). [49](#)
- [126] SILBERSTEIN, M., SCHUSTER, A., GEIGER, D., PATNEY, A., AND OWENS, J. D. Efficient computation of sum-products on gpus through software-managed cache. In *Proceedings of the 22nd ACM International Conference on Supercomputing* (June 2008). [147](#)
- [127] STARCK, J., AND HILTON, A. Correspondence labelling for wide-timeframe free-form surface matching. In *Proc. of ICCV (2007)*, pp. 1–8. [52](#), [54](#)
- [128] STEINBRUCKER, F., POCK, T., AND CREMERS, D. Large displacement optical flow computation without warping. In *CVPR* (sept. 2009), pp. 1609–1614. [166](#), [167](#)
- [129] STICH, T., TEVS, A., AND MAGNOR, M. Global depth from epipolar volumes—a general framework for reconstructing non-lambertian surfaces. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)* (Washington, DC, USA, 2006), 3DPVT '06, IEEE Computer Society, pp. 916–923. [55](#)
- [130] STOLLNITZ, E. J., DEROSE, T. D., AND SALESIN, D. H. Wavelets for computer graphics: A primer, part 1. *IEEE Computer Graphics and Applications* 15 (1995), 76–84. [158](#)

- [131] STOLLNITZ, E. J., SALESIN, H., AND DEROSE, D. T. D. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers Inc., 1996. [156](#), [158](#), [160](#), [164](#)
- [132] STRECHA, C., FRANSENS, R., AND GOOL, L. V. A probabilistic approach to large displacement. In *In proc of ECCV Workshop SMVP (2004)*, pp. 71–82. [166](#)
- [133] SUDDERTH, E. B., IHLER, A. T., FREEMAN, W. T., AND WILLSKY, A. S. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition (CVPR) (2003)*, pp. 605–612. [56](#), [155](#), [163](#)
- [134] SUMNER, R. W., AND POPOVIĆ, J. Deformation transfer for triangle meshes. In *Proc. of SIGGRAPH '04 (New York, NY, USA, 2004)*, ACM, pp. 399–405. [94](#)
- [135] SUN, J., OVSJANIKOV, M., AND GUIBAS, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing (Aire-la-Ville, Switzerland, Switzerland, 2009)*, SGP '09, Eurographics Association, pp. 1383–1392. [52](#)
- [136] SUN, W., AND MUKHERJEE, A. Generalized wavelet product integral for rendering dynamic glossy objects. *ACM Trans. Graph.* 25 (July 2006), 955–966. [139](#), [156](#), [157](#), [158](#)
- [137] SÜSSMUTH, J., WINTER, M., AND GREINER, G. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proceedings of SGP 2008)* 27, 5 (2008), 1469–1476. [41](#), [52](#)
- [138] SÜSSMUTH, J., WINTER, M., AND GREINER, G. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proc. SGP)* 27, 5 (2008), 1469–1476. [53](#), [97](#), [118](#)
- [139] TEVS, A., BERNER, A., WAND, M., IHRKE, I., BOKELOH, M., KERBER, J., AND SEIDEL, H.-P. Animation cartography - intrinsic reconstruction of shape and motion. *ACM Transactions on Graphics, TOG* (2012), to appear. [24](#), [43](#), [52](#), [62](#)

- [140] TEVS, A., BERNER, A., WAND, M., IHRKE, I., AND SEIDEL, H.-P. Intrinsic shape matching by planned landmark sampling. In *Proc. of Eurographics* (2011), vol. 30, pp. 543–552. [23](#), [43](#), [61](#), [82](#), [85](#)
- [141] TEVS, A., BOKELOH, M., WAND, M., SCHILLING, A., AND SEIDEL, H.-P. Isometric registration of ambiguous and partial data. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 1185–1192. [23](#), [43](#), [61](#), [93](#)
- [142] TEVS, A., WAND, M., IHRKE, I., AND SEIDEL, H.-P. A bayesian approach to manifold topology reconstruction. Tech. Rep. MPI-I-2009-4-002, Max Planck Institute Informatik, Saarbruecken, Germany, January 2010. [29](#), [43](#)
- [143] TORR, P. H. S., AND ZISSERMAN, A. Mlesac: a new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.* 78, 1 (2000), 138–156. [50](#)
- [144] TUNG, T., AND MATSUYAMA, T. Dynamic surface matching by geodesic mapping for 3d animation transfer. In *Proc. of CVPR* (2010), pp. 1402–1409. [47](#), [48](#)
- [145] VAN KAICK, O., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. A survey on shape correspondence. In *Proc. of EG STAR* (2010), pp. 1–22. [49](#)
- [146] VARANASI, K., ZAHARESCU, A., BOYER, E., AND HORAUD, R. P. Temporal surface tracking using mesh evolution. In *Proc. of ECCV* (2008), pp. 30–43. [47](#)
- [147] VEDULA, S., BAKER, S., RANER, P., COLLINS, R., AND KANADE, T. Three-dimensional scene flow. In *Proceedings of the 7th International Conference on Computer Vision* (September 1999), vol. 2, pp. 722 – 729.
- [148] VEKSLER, O. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999. [144](#)
- [149] VIOLA, P. A., AND WELLS, WILLIAM M., I. Alignment by maximization of mutual information. *Computer Vision, 1995. Proceedings., Fifth International Conference on* (Jun 1995), 16–23. [142](#)

- [150] VISION, AND DATABASE, A. S. C. I. Teddy dataset. <http://vasc.ri.cmu.edu//idb/html/motion/teddy4/index.html>, 2003. 168
- [151] VLASIC, D., PEERS, P., BARAN, I., DEBEVEC, P., POPOVIĆ, J., RUSINKIEWICZ, S., AND MATUSIK, W. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph.* 28, 5 (2009), 174. 52, 122
- [152] WAND, M., ADAMS, B., OVSJANIKOV, M., BERNER, A., BOKELOH, M., JENKE, P., GUIBAS, L. J., SEIDEL, H.-P., AND SCHILLING, A. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.* 28, 2 (2009), 1–15. 14, 52, 53, 97, 98, 101, 104, 105, 115, 117, 118, 122, 126
- [153] WAND, M., JENKE, P., HUANG, Q., BOKELOH, M., GUIBAS, L. J., AND SCHILLING, A. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. of SGP* (2007), pp. 49–58. 41, 45, 52, 97, 101, 113, 115, 117, 118, 129
- [154] WEISE, T., LEIBE, B., AND GOOL, L. V. Fast 3d scanning with automatic motion compensation. In *Proc. of CVPR* (June 2007), pp. 1–8. 52, 122
- [155] WÜRMLIN, S., LAMBORAY, E., STAADT, O. G., AND GROSS, M. H. 3d video recorder. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2002), IEEE Computer Society, p. 325. 52
- [156] YEDIDIA, J. S., FREEMAN, W. T., AND WEISS, Y. *Understanding belief propagation and its generalizations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, pp. 239–269. 39, 40, 55
- [157] YU, T., LIN, R.-S., SUPER, B., AND TANG, B. Efficient message representations for belief propagation. *CVPR* (2007). 55
- [158] ZHANG, H., SHEFFER, A., COHEN-OR, D., ZHOU, Q., VAN KAICK, O., AND TAGLIASACCHI, A. Deformation-drive shape correspondence. *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing 2008)* 27, 5 (2008), 1431–1439. 47



## Bibliography

---

- [159] ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. Spacetime faces: High resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 548–558. [41](#), [52](#), [98](#)
- [160] ZHAO, W., AND LIANG, Y. W-lbp: Wavelet-based loopy belief propagation for wireless sensor networks. *International Conference on Sensor Technologies and Applications* (2009), 617–622. [55](#)
- [161] ZHENG, Q., SHARF, A., TAGLIASACCHI, A., CHEN, B., ZHANG, H., SHEFFER, A., AND COHEN-OR, D. Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum (Special Issue of Eurographics)* 29, 2 (2010), 635–644. [53](#)
- [162] ZITNICK, C. L., KANG, S. B., UYTENDAELE, M., WINDER, S., AND SZELISKI, R. High-quality video view interpolation using a layered representation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 600–608. [52](#)