Stony Brook University

Coupling Techniques for Dense Surface Registration: A Continuous-Discrete Approach

A Dissertation Presented

by

YUN ZENG

to

The Graduate School

in Partial Fulfilment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2012

Stony Brook University
The Graduate School

Yun Zeng

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Dimitris Samaras - Dissertation Advisor
Associate Professor of Computer Science

Joseph S.B. Mitchell - Chairperson of Defense
Professor of Applied Mathematics

Xianfeng David Gu - Dissertation Co-Advisor
Associate Professor of Computer Science

Nikos Paragios - External Member
Professor of Applied Mathematics and Computer Science
Ecole Centrale de Paris, Ecole des Ponts - ParisTech

This dissertation is accepted by the Graduate School

Charles Taber
Interim Dean of the Graduate School

Abstract of the Dissertation

Coupling Techniques for Dense Surface Registration: A Continuous-Discrete Approach

by

YUN ZENG

Doctor of Philosophy

in

Computer Science

Stony Brook University

2012

In this dissertation, we propose new approaches to the surface registration problem by coupling continuous geometry-based methods and combinatorial graph-based methods. On the one hand, geometry-based methods explore the intrinsic properties of the surfaces to simplify the search of correspondences among surfaces undergoing very large deformations. However, these methods are usually based on certain ideal assumptions on the qualities of the input, such as noise-free or no occlusions. Hence they are usually sensitive to uncertainties from the input that are common in real-world data. On the other hand, graph-based methods are better at dealing with uncertainties due to their statistics nature. Nevertheless, without exploring the geometric properties of the surface, discrete graph-based methods usually suffer from discretization error and high computational complexity. Thus, by exploring the relation between the two approaches, we show that our new approaches deal with surface registration problems under very challenging situations. To this purpose, three distinct approaches are explored in this thesis that achieves dense surface registration in different scenarios.

In the first approach, we cast the surface registration into a high-order graph matching problem, through the minimization of an energy function based on multiple measurements of geometric/appearance similarities and deformation priors. Our method takes advantage of conformal mapping based method which derives a closed-form solution to dense surface matching. To this end, we design an efficient way to select a finite number of matching candidates for each point of the source surface based on the a sparse set of correspondences, which naturally induces an efficient two-stage optimization approach for the dense surface registration problem. In the sparse matching stage, the high-order interactions among a sparse

set of feature points on the surfaces are used to encode the isometric deformation error using conformal mapping. In the dense registration stage, the high-order interactions of a dense set of sampling points are considered to encode the isometric deformation error as well as orientation consistency. Meanwhile, we also propose the first solution to the high-order graph matching problem that solves partial matching. Our method is validated through a series of experiments demonstrating its accuracy and efficiency, notably in challenging cases of large and/or non-isometric deformations, or meshes that are partially occluded.

In the second approach, we propose a graph-based formulation for tracking surfaces in a sequence. In order to deal with noises in the input, we propose a robust metric for the cost of matching arbitrary correspondences, which is defined as the lowest feature differences across this set of matchings that cause the particular correspondence to match. We show that for surface tracking applications, the matching cost can be efficiently computed in the conformal mapping domain. Such a matching cost is then integrated into a complete probabilistic tracking framework that enforces spatial and temporal motion consistencies, as well as error drifts and occlusions. Compared to previous 3D surface tracking approaches that either assume isometric deformations or consistent features, our method achieves dense, accurate tracking results, which we demonstrate through a series of dense, anisometric 3D surface tracking experiments.

In the third approach, we accurately characterize arbitrary deformations between two surfaces and propose a high-order graphical model for the surface registration problem. From Riemannian geometry, the local deformation at each point of a surface can be characterized by the eigenvalues of a special transformation matrix between two canonically parameterized domains. This local transformation is able to characterize all the deformations (*i.e.*, diffeomorphisms) between surfaces while being independent of both intrinsic (parametrization) and extrinsic (embedding) representations. In particular, we show that existing deformation representations (*e.g.*, isometry or conformality) can be viewed as special cases of the proposed local deformation model. Furthermore, a computationally efficient, closed-form solution is derived in the discrete setting via finite element discretization. Based on the proposed deformation model, the shape registration problem is formulated as a high-order Markov Random Field (MRF) defined on the simplicial complex (e.g., planar or tetrahedral mesh). An efficient high-order MRF optimization algorithm is designed for such a special structured MAP-MRF problem, which can be implemented in a distributed fashion and requires minimal memory. Finally, we demonstrate the speed and accuracy of the proposed approach in the applications of shape registration and tracking.

Keywords: Computer Vision, Shape Registration, Surface Tracking, Shape Deformation Model, Conformal Geometry, Graphical Models, High-order Markov Random Fields, High-order Graph Matching

To my parents

Contents

List of Figures

List of Notations

$\mathcal{G}$	graph
$\mathcal{T}$	tree graph
$\mathcal{V}$	set of vertices
$\mathcal{E}$	set of edges (u, v)
$\mathcal{H}$	set of hyperedges 2^V
u, v, w, s, t	graph nodes
i, j, k, l, m	graph labels
C, S	cliques of graph $\mathcal{G}$
$\mathcal{X}_u$	set of labels for node $u \in \mathcal{V}$
x_u	variable at node u
$M_{uv;i}$	belief propagation message from node u to node v for label i
$\mathbb{M}$	marginal polytope
$\mathbb{L}$	locally consistent marginal
$p(\cdot)$	probability distribution
$H(p)$	entropy of distribution p
$A(\theta)$	log partition function or cumulant function
$\mathbb{I}_{u;i}[x_u]$	indicator variable
$\mu_{u;i}, \mu_{uv;ij}$	marginal or min/max-marginal ($E(\mathbb{I}_{u;i}(x_u))$)
	marginal or min/max-marginal
$\theta(\cdot)$	exponential parameter
$\tau_{u;i}, \tau_{uv;ij}$	relaxed indicator variable

Acknowledgments

First of all, I would like to give my deepest gratitude to my dissertation supervisors, Prof. Dimitris Samaras and Prof. Xianfeng David Gu, who have offered me great support and guidance from which I can benefit in life long time. In so many seminars and meetings it was Dimitris who constantly pointed out my weakness in making logical arguments and conveying clear ideas, giving me an alarm bell that will keep ringing for a very long time. Thanks to David, many of the mysterious things that mathematicians and physicists are pursuing are now demystified. It is a rare chance for an engineering student to comprehend the value of those treasures made by the deep mental works of great figures.

I would like to gratefully and sincerely thank Prof. Nikos Paragios for his guidance, support, and patience. Thanks to Prof. Nikos Paragios, I have spent a marvelous summer in one of the most beautiful cities in the world – Paris, where my passion for graphical models had been rekindled by the active research going on there. Most importantly, such a passion has been extended and strengthened for the rest of my PhD life, by my close collaboration with Prof. Paragios' student, my dear friend, Chaohui Wang, who shares a lot of common interests with me. It is hard to imagine without Chaohui, how it could be possible for me to accomplish the numerous exciting projects. In research our ways of working are often complementary to each other, hence we often achieved high efficiency. Moreover, his candid criticisms and reminders have made me a better person. I feel extremely lucky to have met such a great collaborator.

I would also like to thank my other collaborators during my PhD study. During my early stage of PhD, it was out of the collaborations with Wei Zeng that led me to to the field of geometric shape matching. Her persistence in doing research had encouraged me to go through many tough steps along the way of learning geometry. As a veteran member of our computer vision group, Yang Wang has shared with me a lot of valuable experiences in doing research through the numerous projects that we have worked together. He has exemplified to all the new members in our group what an outstanding PhD should be like. I am also indebted to my

previous thesis advisors Prof. Qunsheng Peng and Prof. Wei Chen, who encouraged me to explore my own interests for my master degree research, leading to the essential part of the groundwork of this dissertation.

There are many good fellows in Stony Brook University who have made my PhD life less boring. I just cannot help mentioning these names that brought me great joy and support. The senior members include Hongyu Wang who shared a lot of his PhD experiences to help me avoid making mistakes, and Xiaotian Yin whose perfectionism inspired me to always keep an eye on beautiful things in life. It was lucky for me to come to Stony Brook with a squad of very hilarious people with whom I made countless dumplings and threw numerous parties. Among them are Dengpan Zhou, Wei Xu, Yao Chen, Zhitao Li, Shang Yang, Senlin Liang, Wei Hu and their partners (Jiansong Chen, Jieruo Liu, Feng Zhang, Xin Li, *etc*). In anticipation of a possible very long PhD life, though unfortunately this turns out to be not true, I have also enjoyed the companies of many once newcomers whose "new" points of view often refreshed my thoughts. Among them are Ziyi Zheng, Ruirui Jiang, Min Zhang, Jun Yuan, Tingbo Hou, Yifan Peng, Przemyslaw Szeptycki, Yingying Wu, *etc.*.

At last, many thanks go to those great teachers whose classes have had long-lasting influences to my research and even life. To name a few, they are Liangcai Chu, Yue Chen, Yong He, Ker-I Ko and Joe Mitchell. Moreover, there are also numerous great authors whose books have helped me go through enormous puzzles or even despairs. These great authors include Dimitri P. Bertsekas [23, 214], Theodor Frankel [78], T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Steinetc [56], *etc.*. Furthermore, there are also numerous inspiring figures, including S. Arora, E. Demaine, C. H. Papadimitriou, B. Chazelle, just to name a few, whom I encountered in different occasions. Knowing that these people are doing much more important things than I am, the memories of these encounters always encourage me to pursue a deeper understanding of a possible bigger background behind my research.

Chapter 1

Introduction

> “A good description of the problem is half the solution.”
>
> –Anonymous

Understanding 3D data has become increasingly important in computer vision due to the rapid developments in 3D scanning techniques [3, 18, 34, 99, 152, 202, 271, 297]. As a prerequisite step for most 3D shape understanding tasks (*e.g.*, expression recognition, pose estimation, shape retrieval) and knowledge-based shape manipulation tasks (*e.g.*, deformation transfer, shape compression), accurately finding the correspondences between two shapes (**shape registration**) usually determines the performance of the subsequent applications. Although the problem of 3D shape registration has been studied since the beginning of computer vision, it remains challenging to find accurate correspondences among surfaces undergoing *large non-rigid deformations*, *partial occlusions* and/or being *corrupted by noise*. In this dissertation, we develop a series of new techniques for solving the 3D shape registration problem to tackle these difficulties.

A motivation behind our new approaches to the surface registration problem is to explore the *connections / discrepancies between continuous approaches and discrete approaches*, ◁ which is an active interdisciplinary field that involves geometry, probability / statistics, graph theory, optimization and computational complexity. In physics, it has been shown that many continuous geometric concepts have their discrete counterparts [100] and seeing the world under a discrete point of view often provides us with new insights of the reality [172]. However, not all aspects of the continuous world can be seen through a discrete lens. In complexity

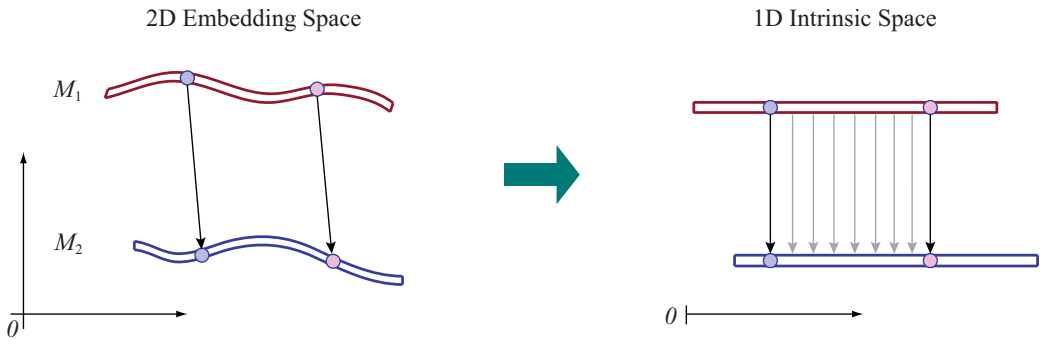


Figure 1.1: Intrinsic space based method illustrated in 1D case.

theory, evidences show that elusive discrepancies exist between the continuous world and the discrete world [50, 122, 168]. Such discrepancies are also closely related to the boundaries of computations [10]. Understanding such connections / discrepancies can not only allow us to realize "what we can do?", but also help us decide "how we can do?". For example, in computer vision, the discrete counterparts of numerous continuous models (*e.g.* level sets functions [49, 176, 186, 207]) have been discovered, resulting in significant speedups and improved solutions (*e.g.*, graph-based solutions to level sets energies [32, 90, 290, 291]). On the other hand, despite their promising performances in terms of speed and optimality, these discrete approaches often suffer from loss of accuracy in their solutions (known as the metrication error [127]). Subsequently, solutions that combine the benefits of both continuous and discrete methods are proposed which achieve speedup, optimality and accuracy (*e.g.*, [148, 198, 200, 253, 286]), though with the sacrifice of expressive power. Motivated by such a trend in computer vision, in this dissertation, we explore the connections between the continuous approaches and the discrete approaches, and we develop algorithms that overcome the limitations of many of the previous approaches for the surface registration problem.

Specifically, for the problem of surface registration, the continuous geometry-based methods explore the intrinsic properties of the problem, which often yield simple yet elegant solutions. For example, a fundamental idea in Riemannian geometry is that certain properties of the shape can be calculated based on the relations among points (via differentiation) without knowing the exact coordinate of each point. By applying such an idea to surface registration, it then becomes feasible to find correspondences between two surfaces undergoing large deformations due to its obliviousness to the extrinsic representations of the surfaces. Moreover, if the deformations among surfaces preserve certain local features (*e.g.*, the Riemannian metric), the degree of freedom of the mapping between the intrinsic representations of two surfaces can be very small, allowing us to only search for a few correspondences in order to determine the mapping for every point on the whole surfaces. Informally, such **Intrinsic space-based**

methods can be represented as

$$\begin{aligned}\text{Intrinsic Space-based Methods } &= \text{Intrinsic Space Representation} + \\ &\quad \text{Transformations among Different Representations} \end{aligned} \tag{1.1}$$

Nevertheless, there are often strict limitations in applying these ideas. For example, most intrinsic methods are usually based on certain ideal assumptions on the input, such as noise-free, no occlusions, *etc*. This is expected since these geometric methods are derived from differentiations which assumes the objects they deal with are sufficiently smooth[1]. Without mechanisms that control the robustness of these geometry-based methods, their performance can degrade significantly when presented with real-world data.

On the other hand, **graph-based methods** are better at dealing with uncertainties due to their statistics nature. Intuitively, graph-based methods perceive the world as discrete, individual entities with connections among them. By sending information among these individual entities through their connections, it is then possible to conduct *inference* on some true *states* of the world, *i.e.*,

$$\begin{aligned}\text{Graph-based Methods} &= \text{Entities (Nodes)} + \text{Connections (Edges)} + \\ &\quad \text{Information Exchanges (Reasoning over true states of the world).} \end{aligned} \tag{1.2}$$

An important property of graph-based methods is that they are not restricted to any specific application domain, rather it is *a skeleton that allows us to effectively encode our own world model, by giving meanings to what the entities represent and how they are related*. The information that can be processed inside a graph include the observed data or the prior knowledge about what should be a plausible state; the plausibility of any possible state can be conveniently represented by an energy function. Hence graph-based methods are suited for dealing with real-world data that are rich in noise and occlusions. Take the surface registration problem for example, since most of the real-world data (*e.g.*, 3D scanning data) are represented by discrete samples, it is more straightforward to represent the surface as a graph instead of a continuous function. The nodes of the graph can be the sampling points and the edges represent their neighborhood information. The state of each node can be defined as the location of its matching point on the target surface. Hence, finding the correspondences between two surfaces becomes finding the matching points for each discrete point on the surface. The level of noise can be inferred based on the smoothness of the deformations of each neighboring nodes. While occlusions can be inferred based on the consistencies of the features between each possible correspondence.

[1] "As far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality"– Albert Einstein, 1956

Though intuitive, it is not straightforward to directly apply the graph-based methods for the surface registration problem due to the following reasons: (i) If the number of nodes or the state space for each point on the surface is too large, it is not possible to do efficient inference for these points. Hence choosing a proper graph structure (number of nodes, matching candidates) for the surface registration problem determines the efficiency of the method. (ii) In order to take advantage of the geometry-based methods to handle large deformations, there usually involves high-order (> 2) interactions among points on the surfaces. Efficient inference for graphs with high-order interactions remains very challenging in computer vision. Thus, designing efficient inference algorithms for the special surface registration problem is necessary. (iii) In order to handle noisy input, a robust measure of the likelihood of each possible correspondence is needed for defining the energy function on the graph. However, in order to increase the robustness of a measure, an increase of the use of information from local to global is needed [43]. Hence, one must balance between the effectiveness and the complexity in choosing a measure.

In summary, in this dissertation, we develop a series of techniques to overcome the difficulties in applying the graph-based methods to surface registration. Note that although a graph-based formulation for the surface registration has been proposed before (*e.g.*, [8]), these methods did not systematically address the issues mentioned above. As a result, most of the previous works are only able to establish sparse correspondences (~ 100) between two surfaces. In contrast, all the techniques developed in this thesis aim at efficiently and accurately solving the **dense surface registration** problem through the insights in both geometry methods and graphical models, which is not possible through any of the previous graph-based techniques for surface registration. ◁

1.1 Basic ideas

In solving a real-world problem, its success often heavily relies on if we have a sufficient model describing the underlying mechanisms. Although there are countless problems in computer science, most of them actually share only a few basic ideas. Identifying these basic ideas behind those problems can not only help us present and re-examine our solutions in a high-level point of view, but also allow us to connect our methods to other solutions. More importantly, since the techniques developed in this dissertation involve the ideas from several fields, it is therefore essential for us to clarify some of the basic ideas that act as the pillars of these fields. Meanwhile, we shall also clarify how these ideas can be applied to the surface registration problem.

1.1.1 Bayesian methods

A central problem in computer vision is to deal with uncertainties, which often arise from the noise through the acquisitions of the data. Hence, a ubiquitous pattern of problem in computer vision is to infer the *underlying states* given the *observations*. If we define a *measure* of likelihood for each possible true state, *i.e.*, $f(\text{state}; \text{observation})$, then many tasks in computer vision fall into computing the most probable configuration:

$$\arg\max_{\text{state}} f(\text{state}; \text{observation}). \tag{1.3}$$

As a result, many approaches in computer vision boil down either to finding the optimal solution (optimization), or to defining a meaningful function for a specific problem (modeling and learning), or both.

Surprisingly, an overwhelming common pattern among the solutions of numerous vision problems falls into the paradigm of *Bayesian method*, which is derived from the basic probability rules[2]. In Bayes' theorem, it states that the measurement function based on the observation can be decomposed into two parts, one involving the likelihood of the observation with respect to the current state, another involving the prior knowledge of the state without considering the observations. Schematically, the Bayesian paradigm can be represented as

$$\text{posterior} = \text{likelihood (data fidelity)} + \text{prior knowledge (regularization)}. \tag{1.4}$$

As a result, most of the vision problems that solve Eq. 1.3 can be regarded as *Maximum a Posterior (MAP)* problems [27].

The above Bayesian paradigm also manifests itself in its relation to physics. In mechanics, the forces that an object bears can be divided into either *internal* or *external*. Internal forces come from the inside of the object, including spring force, magnetic force, etc. The external forces can be any forces coming from outside the object, such as friction, resistance, tension, etc. In early computer vision, such an idea has been successfully applied to physics-based models [171, 190]. For example, in the image denoising problem [88], the internal forces (prior knowledge) are usually defined using the gradients of the true image signal f (*e.g.*, the TV model $\int |\nabla f|$), while the external forces are defined to be the distance between the true signal and the input, (*e.g.*, $\int |f - I|$). Optimizations of these physics-based models often lead to solving a partial differential equation (PDE) problem ([185, 190]).

One important issue concerning applying the Bayesian method is the choice of the *weight* between the likelihood and the prior. Finding the optimal weight for a certain problem is in the domain of parameter learning [124]. In Sec. 1.1.3, we show that the interplay between the weight and the optimal solution can be understood in the idea of duality.

[2] "Probability theory is nothing but common sense reduced to calculation" – Pierre Simon Laplace, 1812

Application in shape registration

In shape registration, the true state to be inferred is the matching of each point on the target surface. The internal force (prior knowledge) can be defined as the the plausible deformation of the surface from the source to the target. The external force corresponds the fitness of the deformed surface to the observed target surface, which can be defined using the similarity in textures and other features. Hence solving the surface registration problem becomes optimizing an energy function defined in the form of Eq. 1.3. It is important to take both forces into account. For example, solely considering external similarities would easily fail if the input data is noisy or ambiguous. Determining the proper weight between internal forces and external forces remains an open problem in the surface registration literature.

1.1.2 Classifications of geometry

The expression "the geometry of ..." is frequently used to signify the deep understanding of a subject [61, 146, 147, 282]. Although intuitively geometry allows us to "see through" things, many insightful properties of a geometric object are usually achieved by complicated algebraic calculations, which are often counter-intuitive. For instance, modern physics relies heavily on the development of geometry in mathematics since many of the subjects (*e.g.*, spinors, monopoles, etc) can only be speculated through calculations [78, 194].

In mathematics, there are several branches of geometry. For example, there are *Euclidean geometry, Riemannian geometry, Conformal geometry, Projective geometry, Algebraic geometry and Hyperbolic geometry, Symplectic geometry*, etc. The classification of these geometries are based on the **invariant** group under which the properties of the geometry does not change[3]. In the language of quotient group, each geometry studies the properties of the geometric object under the quotient:

$$\frac{\text{Geometric Objects}}{\text{Equivalent Transformations}}. \tag{1.5}$$

For example, properties of Euclidean geometry (*e.g.*, Pythagoras' theorem, Thales' theorem) are invariant under rigid transformations. In Riemannian geometry, its equivalent transformation is called *isometric* transformation, meaning the preservation of Riemannian metrics.

[3] According to Felix Klein's Erlangen Program: *geometries study those properties of spaces invariant under various transformation groups*.

In Conformal geometry, it studies properties of objects that are equivalent under *conformal* transformation, which is a subclass of isometric transformation. In Projective geometry, as its name implies, the invariant transformation is the projective transformations, which is in fact independent of any metric structure. Projective geometry is very fundamental in computer vision since every two dimensional image is generated by the projective transformation.

Although the study of geometry seems to be in the domain of continuous mathematics, the investigation into more abstract geometry directly lead to the study of discrete mathematics. For example, in Algebraic geometry, the concept of *fundamental group* studies the classification of the surfaces based on the shrinkage of different loops on the surface. Such a classification of the surface can be elegantly represented as the operations (composition) on strings, which can be considered as the preliminary form of the Turing machines (the relation between the Algebraic geometry and Turing machine needs to be verified) [298].

Application in shape registration

For the problem of surface registration, applying the appropriate geometry can help significantly simplify the solution. The two major players in the literature are Riemannian geometry and Conformal geometry. In Riemannian geometry, the invariant transformation group is the *isometric* transformation. Hence by using features derived using Riemannian geometry, such as curvature and geodesics, it is then possible to register surfaces undergoing very large deformations (*e.g.*, [37]). Another geometry that has been successfully applied to the surface registration problem is the Conformal geometry. An important property of conformal geometry is that when two surfaces undergo only isometric deformation, a dense mapping between two surfaces can be established by recovering a closed-form solution with very few degrees of freedom. Throughout this dissertation, we shall show how such a property is explored in different applications and it significantly improves the performance.

1.1.3 Duality

Duality reflects the two (often complementary) sides of the same thing, which is prevailing in many areas (*e.g.*, geometry, physics, optimization, etc). Since the primal and the dual are always equivalent, one might wonder why bother with the dual side since it seems to give us

no more information than the primal side. Nevertheless, in many cases, since there may exist difficulties in obtaining the complete picture of the object from the primal side, looking at the dual side often provides us a complementary view, which is often beneficial.

Intuitively, duality arises from the interactions between two objects, as can be schematically represented as:

$$\langle \text{signifier}, \text{signal} \rangle \rightarrow \text{outcome}. \tag{1.6}$$

The outcome of such an interaction can be anything that is meaningful, depending on the specific domain of interest.

For example, a computer program can be regarded as the interaction between the data and the program, namely

$$\langle \text{program}, \text{data} \rangle \rightarrow \text{problem solution}. \tag{1.7}$$

Within a duality interpretation, a program can be equivalently viewed as a piece of data, which leads to the issue of *self-references* [224]. The study of self-references then leads to many important results in computability for computer programs (*e.g.*, the halting problem, [82, 298]). Another benefit from such a primal-dual view of a computer program is the interpretation of the computational complexity of a computer program. Traditionally, the complexity class **NP** is defined as the class of problems that have efficient verifiable solutions (programs) [4], and the class **P** is defined as the set of programs that halts in polynomial time for any input. The celebrated Millennium Prize Problem [4] asks the simple question that if $\mathbf{P} = \mathbf{NP}$? By examining such a question in a primal-dual interpretation, it is not difficult to find out that the original question only concerns about the distribution of the program. In fact, a distribution of the input can also affect the running time of the program. Such an insight directly leads to more general classes of complexity, for example, the classes of **PCP** [10].

In field of computational learning theory [117, 166], the program/data duality underlies some of its core ideas. In a machine learning task, the goal is to output an optimal hypothesis $h \in H$ based on the input data X, where H is a set of predefined hypotheses. Hence, the interaction involved in here is

$$\langle \text{learning algorithm}, \text{training data} \rangle \rightarrow \text{hypothesis } h. \tag{1.8}$$

In contrast to computational complexity, the concern of a learning task is learnability, which is intrinsically stochastic. Under the primal-dual point of view, the accuracy of a learning

[4] A decision problem / language is in NP if given an input x, we can easily verify that x is a YES instance of the problem (or equivalently, x is in the language) if we are given the polynomial-size solution for x, that certifies this fact [10].

algorithm does not only rely on the distribution of the data, but also the learning algorithm and the available hypothesis. The *probably approximately correct* (PAC) [256] is a widely used learning model that describes such combined effect of both sides.

Another form of duality, which is more familiar to the computer vision community, is the duality between a function and its input, as can be expressed as

$$\langle \text{function}, \text{input} \rangle \rightarrow \text{some number}. \tag{1.9}$$

Such form of duality is ubiquitous in mathematics. For example, if the input space is the vector space of $\mathbb{R}^n$, then all the linear functions defined in that space is another linear (functional) space, which is known as covector. The duality between a vector space and its covector space often helps us understand the same physical object from different aspects.

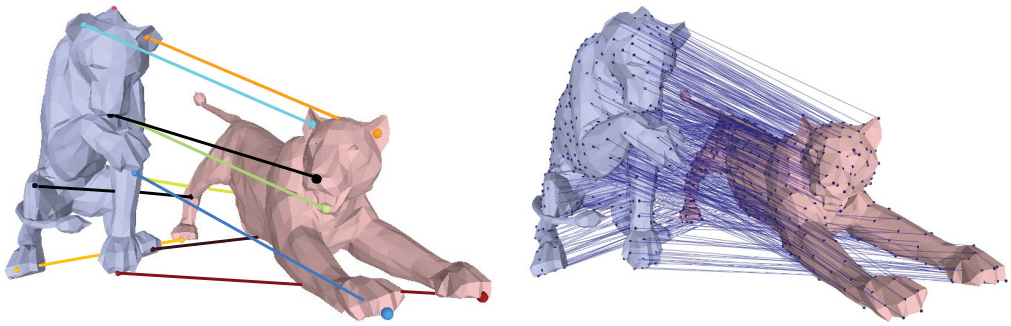
Application in shape registration

Duality is very helpful in solving optimization problems. Since almost all inference problems in Bayesian methods boil down to solving an optimization problem, duality often play a key role in efficiently finding an (near-)optimal solution. For example, when a Bayesian formulation involves discrete variables, a ubiquitous way to find the optimal solution of such formulation is to relax its variables to be continuous and the energy to be convex. However, directly solving those relaxed problem is often inefficient since there often involve thousands of millions of variables (especially in the case of dense surface registration). Even the energy is convex, a gradient descent based approach often involve a large number of iterations due to the large search space. To overcome such a difficulty, exploring the solution in the dual space of the optimization problem often involves solving a problem with a much smaller size and it can also help achieve globally optimal solutions.

1.2 Thesis overview

This thesis is mostly devoted to the developments of practical algorithms that solve the dense surface registration problem in different scenarios. This said, less focus is put on the theoretical aspects of the methods developed [5]. Nevertheless, the author does make an effort to

[5] "Analysis is inherently the domain of a mathematician while optimization belongs to the engineer" – Jon Dattorro



(a) Sparse registration (b) Dense registration

Figure 1.2: Dense non-rigid surface registration using two-stage high-order graph matching.

reveal the possible deep connections behind these methods developed, though these work do not belong to the main part of this thesis.

For every problem solved in this thesis, the following two major points have been addressed:

- **Problem formulation**: define a meaningful metric/quantitative measure for the problem, *i.e.*, what is a good solution?
- **Optimization**: search for the (near-)optimal solution based on the metric.

As a result, we have made contributions in both parts, *i.e.*, we proposed both novel formulations and novel optimization techniques for the problems we solve. More detailed overview of the contributions of this dissertation is listed as follows.

1.2.1 A high-order graph based formulation for surface registration

Our first work solves the problem of finding the dense correspondences between two surfaces undergoing large deformations (Fig. 1.2). To this end, we cast the surface registration problem into finding the optimal correspondences on a target surface for a discrete set of points on a source surface, through the minimization of an energy function based on a measurement that involves geometric/appearance similarities and deformation error. Here the deformation error involves the high-order interactions among the discrete points. Hence the surface registration is then naturally formulated as a high-order graph matching problem.

In order to solve the proposed high-order graph matching problem, we first formulate its energy function into a pseudo-boolean optimization [30], which allows us to employ an efficient order-reduction technique [106] which reduces the high-order terms into pairwise ones.

Furthermore, since the reduced pairwise pseudo-boolean optimization involves non-convex terms, we further re-formulate the original problem as the union of several sub-problems that are easier to solve. Then a meta-algorithm (known as the dual-decomposition) is applied, combining the solutions of these sub-problems to obtain an robust, often optimal solution. This is the first high-order graph matching algorithm that handles partial matching. We show that its application to the problem of sparse surface registration leads to accurate and efficient results.

However, there are obstacles in applying the above graph-matching formulation to the dense surface registration, due to its inability of being discriminative in the metric defined for sparse matching, as well as the high computational complexity and inaccuracy in the deformation prior. To overcome this, we observe that since a unique conformal mapping can be determined by fixing any three correspondences, a unique matching point for any point on the original surface can therefore be determined by any three correspondences too. If the surface is isometrically deformed, then any three correct correspondences would result in the same matching point on the target surface for any point on the source surface. Otherwise, due to noisy input and anisometric deformation of the surface, we can expect the clustering points of these matching points for any point give us the candidate matching positions. Hence, with such a candidate selection scheme, we have efficiently converted the dense surface matching problem into a combinatorial problem: selecting the optimal matching among those candidate matching points. Another advantage of our candidate selection approach is that it also allows us to detect occlusions as well.

Therefore, towards solving the dense surface registration problem, we propose an efficient two-stage optimization technique. In the first sparse registration stage, a small number of feature points are selected on the surfaces and the high-order graph matching algorithm is applied to find an initial correspondence between them. In the second dense registration stage, the results from the sparse registration stage are used to get candidate matching points for a dense set of points on the surface. Then a local high-order graph matching optimization is applied which finds the optimal registration among these selected matching points, by constraining the validity in triangulation and consistency in geometry and texture.

1.2.2 A robust metric for surface tracking and its use in a probabilistic framework

Our second work solves the problem of finding the accurate correspondences in a sequence of 3D frames (Fig. 1.3). In this problem, the deformations between two frames are small and even subtle. Contrary to our first work, whose purpose is to find the correspondence between two surfaces undergoing large deformation, the challenge in the surface tracking problem is to

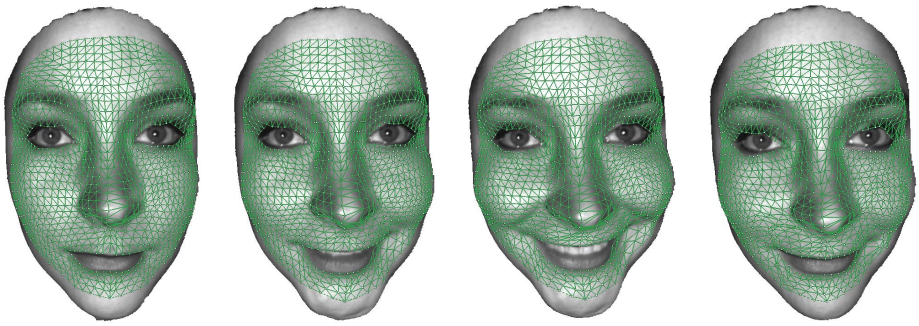


Figure 1.3: Dense surface tracking

accurately follow the subtle change of every point on the surface.

Hence, the first contribution of this work is to *define a robust metric for correspondence match*. The new metric is derived from the idea of *max-marginal* in statistics [262], which is defined as the maximal probability of multiple variables by fixing the value of a particular variable. The max-marginal essentially eliminates the dependence among the variables. This said, if we know the max-marginal of a probability function for every variable, then the *Maximal a Posterior* (MAP) solution can be easily obtained by optimizing n single variable functions where n is the total number of variables. However, in practice, the max-marginal is often difficult to compute. In this paper, again by utilizing the property of conformal mapping that every three sparse correspondences give us a closed-form solution for the matching between the whole surface, we derive an efficient way to approximate the max-marginal (in a minimization problem, it is named min-marginal more properly) for every point on the surface. Specifically, the matching cost of any particular correspondence is defined to be the lowest feature differences across the set of matchings that cause the points to correspond. Furthermore, the computation of this matching cost is implemented in a parallel hardware. In our experiments, we show that our new metric is most discriminating among the state-of-the-art local shape features.

In order to apply the idea of max-marginal to surface tracking, we construct a dense template for the 3D data in the first frame. Using our candidate selection scheme from our first work, for each vertex on the template, a number of matching points are selected in the second frame. Then we can evaluate which is the best matching among these matching candidates using our metric for any correspondence match. However, since our metric is approximately computed, finding the matching point for each vertex on the template individually can not guarantee the accuracy of every point. To remedy this, a number of priors are explored that enforces the spatio-temporal consistency of the final matching result.

Hence, the second contribution of this work is a complete probabilistic framework for

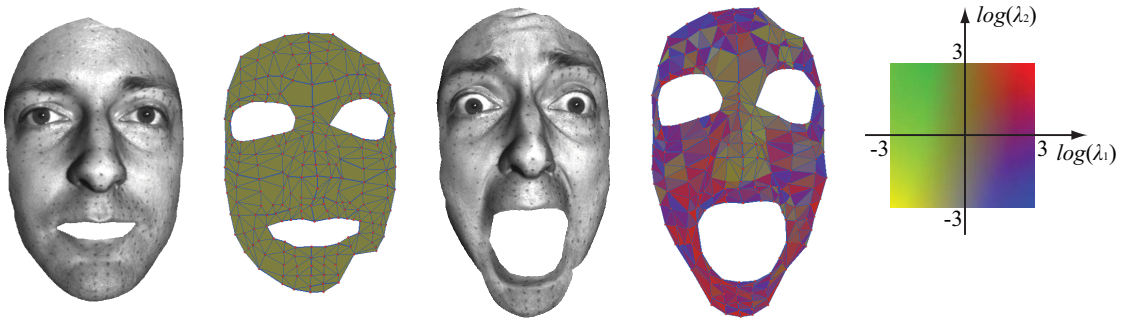


Figure 1.4: A generic surface deformation model

the surface tracking problem that considers the spatio-temporal consistency, as well as occlusions and error drifts. The metric we have defined for the correspondence match integrated in the definition of probability of any possible configuration of the dense template. First, we factorize the original multiple variable probability into the *data fidelity term* and the *spatio-temporal consistency term*. For the data fidelity term, we further factorize it into a term that regularizes the data similarity between frames and a term that constrains the consistency between current frame and all the previous frames, which essentially avoids error drifts. For the spatio-temporal consistency term, we factorize it into a spatial deformation prior term and a temporal motion consistency term, which regulate the consistency of the deformations of the template. Finally, since we only select a discrete number of matching candidates, the inference on the probabilistic framework then becomes solving a pairwise Markov Random Field (MRF) problem. The resulting energy can be efficiently solved by employing the existing MRF optimization algorithm.

1.2.3 A generic surface deformation model with a high-order MRF formulation for surface registration

In our third work, we deal with the question of accurately modeling surface deformations. Almost all the existing intrinsic space-based surface matching methods assume the deformations of the surfaces to be isometric. Nevertheless, this assumption only applies to a coarse scale in most cases. Hence, to achieve accurate surface registration, especially for the dense registration problem, an accurate surface deformation model is needed.

Our new deformation model is derived using the ideas in Riemannian geometry, where a surface is equipped with a *metric tensor* at each point. The deformation of a surface also causes the changes of the metric tensor at each point (*e.g.*, if the change is identity, then the

deformation is isometric). Accounting the changes of metric tensor allows us to quotient out large deformation as the intrinsic space space based method promises. To represent such changes, an intrinsic representation (parametrization) of the surfaces is needed. However, there are numerous ways to parameterize a surface, which are equivalent as long as the transition functions among them satisfy certain rules. Here we choose a special parametrization (the *canonical parametrization*) whose metric tensor at a particular point is the identity matrix. Under the canonical parametrization, the change of metric tensor can be represented by a symmetric matrix that is related to the Jacobian of the deformation in the parametrization domain. Then our new deformation can be accounted by the eigenvalues of such a matrix (the *canonical distortion coefficient*), which we prove to subsume the isometric and conformal deformations. Intuitively, the two eigenvalues characterize the change of an infinitesimal circle to an infinitesimal eclipse at each point.

When the input surfaces are represented by the simplicial complexes (triangular meshes), we establish the discrete counterpart of the above continuous concepts using methods in finite element analysis. To this end, we assume that the deformation of the shape in the parametrization domain be piecewise linear. Hence, the problem of computing the canonical distortion coefficient at a point becomes computing it on a facet. The canonical parametrization at a point is simply equivalent to mapping each face to the 2D domain. Within this setting, the computation of the canonical distortion coefficient for each face becomes solving linear equations with a closed-form solution.

Furthermore, we apply this deformation model to the problem of surface registration, which is formulated as a *high-order MRF* problem. In this formulation, the singleton terms correspond to the measure of correspondence matching defined in our second work and the high-order terms encode the generic deformation prior defined above. In order to solve such a special high-order MRF optimization defined on the simplicial complex, we employ the dual optimization technique [214]. By implementing the optimization algorithm in a parallel hardware, a maximal speedup of around 100 times is achieved. In our experiments, we show that our new surface registration method can significantly improve the matching accuracies of both our first work (matching surfaces with large deformations) and second work (matching surfaces with subtle change).

Bibliography

Part of Chapter 2 is based on the work presented at European Conference on Computer Vision (ECCV), 2008 [289], the workshop on 3D Face Processing in conjunction with CVPR, 2008 [288], and International Conference on Computer Vision (ICCV), 2007 [92]. In Chap-

ter 3, the unifying formulation for MRFs with high-order convex priors (Sec. 3.2.2) is based on the author's current ongoing research. Chapter 4 is based on the work presented at Computer Vision and Pattern Recognition (CVPR), 2010 [292]. Chapter 5 is based on the work presented at Computer Vision and Pattern Recognition (CVPR), 2011 [294]. Chapter 6 is based on technical report of INRIA, 2011 [293].

Here [288, 289] are out of the joint work with Wei Zeng *et.al.*, in which I participated part of the experiments; [92] is from the joint work with Prof. Xianfeng Gu *et.al.*, in which I helped part of the experiments; The work of [268, 292–294] are out of the close collaborations with Chaohui Wang *et.al.*.

Chapter 2

Groundwork – Geometry-based Methods

> “Everything is connected to everything else, ...(to be continued)”
>
> – Anonymous

In this chapter, we explain the rationale for applying **geometry-based** methods for surface matching problems as well as its connection with the **graph-based** methods. As the essential background, some basic concepts in Geometry and Topology are briefly introduced in Section 2.1. Specifically, geometric concepts based on differentiations are introduced in Sec. 2.1.1, 2.1.2, and 2.1.3. By grouping the continuous geometric objects into finite groups (topology), the connection between continuous geometry and combinatorial structures are established in Sec. 2.1.4 using the language of simplicial complexes.

Given the basic concepts in geometry and topology, we introduce one of its important branches – Conformal Geometry – and its applications in shape registration in Sec. 2.2. We emphasize here that *due to the flexibility of conformal geometry, the mapping between two surfaces can be derived in a closed-form (under certain ideal assumptions on the deformation of the surface), which is very useful for establishing dense correspondences between two surfaces*. Furthermore, to overcome the limitation of conformal geometry that it can only deal with conformal/isometric deformations, we discuss a general class of mapping between two surfaces, namely the *quasi-conformal* mapping, in Sec. 2.2.3. ◁

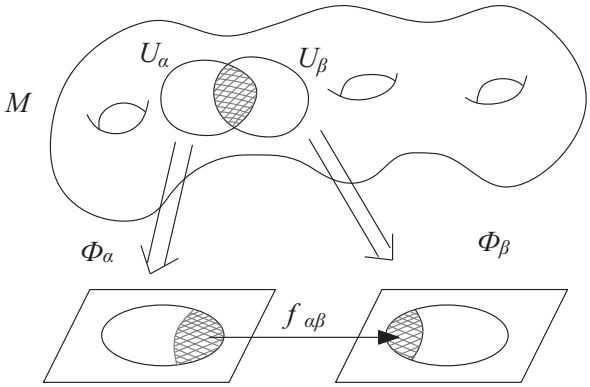


Figure 2.1: A manifold can be represented by its local charts. The transition map $f_{\alpha\beta}$ can not be arbitrary, it determines the geometric invariance that should be agreed upon among different local coordinates.

2.1 Geometry and topology

A topological n-**manifold** is a space $\mathcal{M}$ locally homeomorphic to $\mathbb{R}^n$, *i.e.*, there is a cover of open sets $\mathcal{U} = \{U_\alpha\}$ of $\mathcal{M}$ and maps $\phi_\alpha : U_\alpha \to \mathbb{R}^n$ that are continuous bijections onto their images in $\mathbb{R}^n$ with continuous inverses. Here the pairs (U_α, ϕ_α) are called **charts**. Then an **atlas** of on a manifold $\mathcal{M}$ is a collection of charts on $\mathcal{M}$ whose domains cover $\mathcal{M}$ (Fig. 2.1). Examples of manifolds include the n-sphere $\mathbb{S}^n$, the real projective n space $\mathbb{R}P^n$, *etc*.

Given the representation of a manifold using the atlas, we can study the properties of the surface *without leaving the surface*. Furthermore, if (U_α, ϕ_α) and (U_β, ϕ_β) are two charts on $\mathcal{M}$ such that $U_\alpha \cap U_\beta$ is non-empty, then the two charts are related by the **transition map**:

$$f_{\alpha\beta} := \phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta). \tag{2.1}$$

A manifold is *orientable* if all the transition maps $f_{\alpha\beta}$ preserves orientation (have determinants with positive derivative). Examples of non-orientable manifold include the projective plane $\mathbb{R}P^n$ and the Klein bottle K^2. In physics, the transition maps between two coordinate charts should belong to certain classes. For instance, in order for two observers to agree on the length/area/volume of the same object in 3D, the transition map between the two orthogonal coordinate systems that the two observers use must be a *rigid* transformation. As a more complicated example, in the theory of relativity, in order for two observers to agree on the same "length" in space-time, the transition map must be a *Lorentz* transformation [78, 194].

The atlas definition of a manifold also allows us to define the **differential** between two different surfaces. Specifically, we can define a map between two manifolds $f : \mathcal{M}^m \to \mathcal{N}^n$

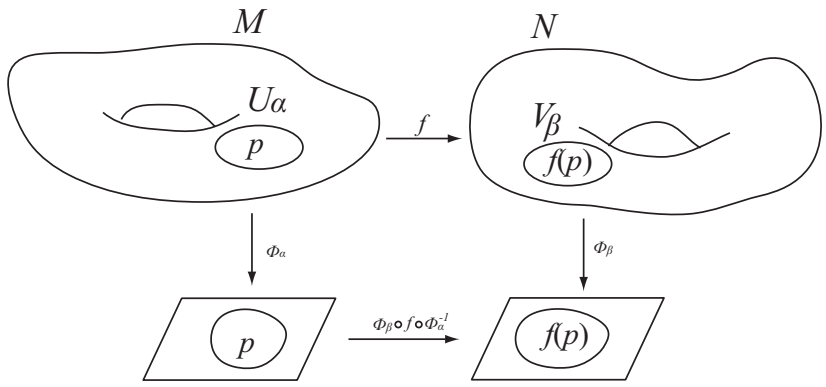


Figure 2.2: The differential between two manifolds is defined on their local charts. The mapping $\phi_\beta \circ f \circ \phi_\alpha^{-1}$ determines the type of deformations that is invariant between the local representations of the two surfaces.

using their local charts, *i.e.*, whenever f takes $p \in U_\alpha$ to $f(p) \in V_\beta \subset \mathcal{N}$, the mapping $\phi_\beta \circ f \circ \phi_\alpha^{-1}$ defines a smooth map from $\mathbb{R}^m$ to $\mathbb{R}^n$ (Fig. 2.2). Hence we can define the differential of f at p as the differential of $\phi_\beta \circ f \circ \phi_\alpha^{-1}$ from $\mathbb{R}^m$ to $\mathbb{R}^n$, which is a linear transformation mapping a vector in $\mathbb{R}^m$ to a vector in $\mathbb{R}^n$. Note that the choice of chart should be **independent** of the result of the differential, meaning that certain transformation rules must be satisfied among overlapping charts. The class of the differentials between two surfaces determines the type of deformations that are **invariant** to the local observers (Figure 2.2). Examples of such invariant deformations include isometric/conformal/affine maps (rigorous definitions of these deformations will be given later in this chapter).

One might wonder what happens if the maps are constrained to be the most general type: the diffeomorphisms. Does that mean any two surfaces can be continuously deformed from one to the other since a diffeomorphism exists between any two charts? The answer to this question is that although locally any two surfaces look the same, globally two shapes are distinguished by their **topology**. Topology is a bridge that connect continuous math with discrete math (any others?). For instance, the **Euler characteristic** is an important topological feature, which is originally defined on the planar graph by Euler and then has been shown to exist in the continuous setting in various forms, *e.g.*, the celebrated Gauss-Bonnet theorem, the fixed-point theorem or the index theory. There are also other topological features that describe more detailed structure of the surface, *e.g.*, **Homotopy** or **Homology** groups. It is important to realize the **topological obstruction** that is inherent in many real-world problems. For example, ignorance of the the existence of singular points would prevent us from constructing a everywhere smooth function on the surface [91]. Also in constructing a global conformal mapping,

it is impossible to map the whole surface conformally into 2D with positive curvature everywhere for high genus surfaces (in this case one should consider the hyperbolic space). In the following, we introduce the essential concepts for characterizing the map between two surfaces in the continuous setting.

2.1.1 Tangent space and vector field

In above, we have mentioned that the differential between two surfaces should be *independent* of their local chart representations. By "independent", we mean that if there is a transition map between two charts U_α and U_β for a quantity defined on $U_\alpha \cap U_\beta$, certain transition rules must be satisfied such that the two representations of the quantity are *equivalent* in certain sense. One of such quantities is the concept of tangent vector defined below.

Definition 1 *A **tangent vector** at $p \in \mathcal{M}$, call it $\mathbf{X}$, assigns to each coordinate patch (U_α, ϕ_α) holding p, and n-tuple of real numbers*

$$(X_\alpha^p) = (X_\alpha^1, \ldots, X_\alpha^n). \tag{2.2}$$

If $p \in U_\alpha \cap U_\beta$, and $x_\alpha = (x_\alpha^1, \ldots, x_\alpha^n)$ and $x_\beta = (x_\beta^1, \ldots, x_\beta^n)$ are the local coordinates for U_α and U_β, respectively, then we have

$$X_\beta^i = \sum_{j=1}^{n} [\frac{\partial x_\beta^i}{\partial x_\alpha^j}(p)] X_\alpha^j, \forall i \in \{1, \ldots, n\} \tag{2.3}$$

Here the matrix $c_{\alpha\beta} \equiv [\frac{\partial x_\beta^i}{\partial x_\alpha^j}]$ is the **Jacobian matrix** between the two charts U_α and U_β, as defined in Eq. 2.3.

Hence a tangent vector can be represented by different coordinates under different parametrizations as long as Eq. 2.3 is satisfied. Also at each point, all the tangent vectors form a linear space, which can be represented by a coordinate basis under any parametrization U_α.

Definition 2 *The **tangent space** to $\mathcal{M}$ at the point $p \in \mathcal{M}$ is the real vector space consisting of all tangent vectors to $\mathcal{M}$ at p. If $x_\alpha = (x^1, \ldots, x^n)$ is a coordinate system holding p, then the n vectors*

$$\frac{\partial}{\partial x^1}, \ldots, \frac{\partial}{\partial x^n} \tag{2.4}$$

*form a basis of this n-dimensional vector space and the basis is called a **coordinate basis**. We denote by $T_p\mathcal{M}$ as the tangent space at a point $p \in \mathcal{M}$.*

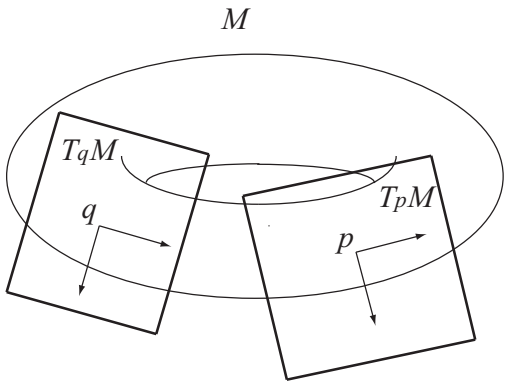


Figure 2.3: At each point, there is a linear tangent space associated to it. Each tangent vector lies in the tangent space.

Here $\frac{\partial}{\partial x^i}$ means the change of the surface along the direction of the ith axis in the local coordinate system. We denote it as an operator since we are working in the intrinsic space without knowing the specific representation of the surface. If we consider the tangent vectors at every point, we can define a **vector field** on an open set U to be the differentiable assignment of a vector $\mathbf{X}$ to each point of U; in terms of local coordinates $x_\alpha = (x^1, \ldots, x^n)$, we have

$$\mathbf{X} = \sum_{j=1}^{n} X^j(x) \frac{\partial}{\partial x^j}. \tag{2.5}$$

As in the case the surface is mapped to $\mathbb{R}$ by the function $f : \mathcal{M} \to \mathbb{R}$ and a vector at a point p on the surface, we can define a differential operator as (in local coordinates)

$$\mathbf{X}_p(f) := \sum_j [\frac{\partial f}{\partial x^j}](p) X^j. \tag{2.6}$$

One can verify that such definition is independent of the coordinate chosen if the Jacobian rule of Eq. 2.3 is satisfied.

Both the tangent vector and the tangent space are defined at a single point on the surface (Fig. 2.3). The union of all tangent spaces $T_p\mathcal{M}$ for all $p \in \mathcal{M}$ is itself a self-contained space (denoted by $T_*\mathcal{M}$), which is called the **tangent bundle** of $\mathcal{M}$. Tangent bundle itself is a manifold. A point on the tangent bundle can be described by its local coordinate system $U_\alpha \times \mathbb{R}^n$ and similar transition rules as Eq. 2.3 must be satisfied. Tangent bundle is useful when a system is characterized both by its location and speed at each point (*e.g.* Hamilton's mechanics [78, 193]).

2.1.2 Tensors, differential forms and metric tensor

So far our discussion is solely focused on the quantities of the surface without interactions among them. However, it is more intuitive to deal with physics or mathematics if we are able to compare one quantity against another. For example, one can define the length, area and volume of a geometric object. Generally speaking, a measure/interaction between two quantities can be represented in the form

$$\langle \cdot, \cdot \rangle \to \mathbb{R}. \tag{2.7}$$

Not surprisingly, in geometry, the slots are usually filled with vectors. Such a measure can be either linear or nonlinear. The linearity assumption is made on the form of the interaction that satisfies

$$\langle \cdot, v + w \rangle = \langle \cdot, v \rangle + \langle \cdot, w \rangle. \tag{2.8}$$
$$\langle v + w, \cdot \rangle = \langle v, \cdot \rangle + \langle w, \cdot \rangle. \tag{2.9}$$

Although limited, such a linearity assumption has been proven to be powerful in accurately modeling the complicated physical world [1]. Furthermore, by extending the number of objects involved in the interaction, we define the notion of multilinear interactions among n vectors as follows.

Definition 3 *Let E denote a linear nD space, a **covariant tensor of rank** r is a multilinear real-valued function*

$$Q : E \times E \times \cdots \times E \to \mathbb{R} \tag{2.10}$$

of r-tuples of vectors and the values of this function is independent of the basis in which the component of the vectors are expressed.

Again, if a tensor is defined with a geometric meaning, its value must be independent of the basis in which the components of the vectors are expressed. On the other hand, given a basis of the linear nD space E, the tensor Q can be uniquely identified by a $n \times \dots n$ matrix. The entry for the matrix is defined as

$$Q_{i_1 \dots i_n} := Q(\partial_{i_1}, \dots, \partial_{i_n}), \tag{2.11}$$

where ∂_i is the ith basis. Note that there are n^r such entries.

[1] The field of Finsler Geometry [16] studies the nonlinear structure of the geometry.

If we consider the space of all the linear functions that map E^r to $\mathbb{R}$, it is actually itself a linear space. Such a linear functional space of a multilinear product linear space E^r is the **dual space**, denoted by $\otimes^r E^*$. In convention, the dual space is often given a prefix "co". For example, the dual space $T_p^*\mathcal{M}$ to the tangent space $T_p\mathcal{M}$ is called the **cotangent space**; the dual space of a vector space is called a **covector** (or **covariant**). Given two covectors α and β of rank p and q, respectively, we can define their **tensor product** $\alpha \otimes \beta$ as follows

$$\alpha \otimes \beta(v_1, \ldots, v_{p+q}) := \alpha(v_1, \ldots, v_p)\beta(v_{p+1}, \ldots, v_{p+q}) \tag{2.12}$$

It can be seen that the functional space obtained by the tensor product is another tensor space. However, it is only a sub-space of all the rank $(p+q)$ tensor space (one has dimension n^{p+q}, the tensor product has dimension $n^p + n^q$ [2]).

Using the tensor space, one can define different quantities on the surface as long as these quantities satisfy linearity condition. In physics, one is interested in a sub-space of the covariant r-tensor space $\otimes^r E^*$ that computes the area of the parallelogram spanned by any r input vectors $\{v_1, \ldots, v_r | v_i \in E\}$ (obviously, such a subspace is a multilinear subspace of $\otimes^r E^*$ and if $\alpha \in \otimes^r E^*$, $\alpha(\ldots, v, \ldots, v, \ldots) = 0$ in each pair of entries.). By expanding the equation $\alpha(\ldots, v_r + v_s, \ldots, v_r + v_s, \ldots) = 0$, we obtain the **antisymmetry** property of such special r-tensor space, ***i.e.***,

$$\alpha(\ldots, v_r, \ldots, v_s, \ldots) = -\alpha(\ldots, v_s, \ldots, v_r, \ldots) \tag{2.13}$$

In fact, the space of r-tensors that satisfies the antisymmetry property is itself a linear subspace. Such subspace is of special interest to mathematics and physics. Formally, we define

Definition 4 *An (**exterior**) p-**form** is a covariant p-tensor $\alpha \in \otimes^r E^*$ that is antisymmetric in each pair of entries. A p-form is often denoted by $\wedge^p E^*$.*

As some special cases, a 0-form is the function $f : \mathcal{M} \to \mathbb{R}$ defined on the surface, a 1-form is simply the covector space of $T_p\mathcal{M}$, namely $T_p^*\mathcal{M}$. In contrast to the dimension n^p for a p-tensor, a p-form has dimension $\binom{n}{p}$, which is a significant smaller subspace of the original linear tensor space. Even so, many properties can be derived for the form space (known as exterior algebra).

For example, the concept of derivation can be extended for differential forms (called **exterior differentiation**), which maps a p-form to a $(p+1)$-form. We have the following theorem

[2] Note that this tensor product is in essence very similar to the factorization of graphical models as we shall introduce in the next chapter. The differences, however, are that in graphical models the variables are allowed to overlap among different lower cliques and the clique functions are not necessarily linear.

Theorem 1 *There is a unique operator, exterior differentiation,*

$$d : \wedge^p E^* \to \wedge^{p+1} E^* \tag{2.14}$$

satisfying

1. *d is additive, $d(\alpha + \beta) = d\alpha + d\beta$.*
2. *if $p = 0$, df is the usual differential of the function f*
3. *$d(\alpha^p \wedge \beta^q) = d\alpha^p \wedge \beta^q + (-1)^p \alpha^p \wedge d\beta^q$.*
4. *$d^2\alpha := d(d\alpha) = 0$, for all forms α.*

The space of tensors derived by exterior differentiation constitutes a subspace of the form space. Two special such subspaces are considered, namely the closed and exact forms. A differential form α is said to be **closed** if $d\alpha = 0$; it is said to be an **exact** form if there is another form β such that $\alpha = d\beta$. From the property of differentiation, it can be seen that any exact form must be a closed form. In physics, the concept of a closed form corresponds to the conservation of forces, and the concept of exact form corresponds to the concept of having a potential function. Hence, these two special forms are very fundamental (Q: are there any other such fundamental forms?). The study of the quotient: closed forms / exact forms, leads to the concept of *cohomology* as we shall discuss later in this section.

While the exterior differentiation lifts a p-form into a $(p+1)$-form, the **interior product**, denoted as i_v, degrades a p-form into a $(p-1)$-form, which is defined by

$$i_v \alpha^0 = 0 \qquad \text{if } \alpha \text{ is a 0-form} \tag{2.15}$$

$$i_v \alpha^1 = \alpha(v) \qquad \text{if } \alpha \text{ is a 1-form} \tag{2.16}$$

$$i_v \alpha^p(w_2, \ldots, w_p) = \alpha(v, w_2, \ldots, w_p) \qquad \text{if } \alpha \text{ is a } p\text{-form} \tag{2.17}$$

One might wonder what happens if we apply to a p-form α^p the operator $i_v \circ d$ or $d \circ i_v$. The result should still be a p-form, and one should employ the concept of **Lie derivative** $\mathfrak{L}_v$. Here we skip more detailed discussion in this direction.

Another important example of the tensor space is the **metric tensor** G that defines the inner product between two tangent vectors. A manifold quipped with a metric tensor is called **Riemannian manifold**, and Riemannian Geometry studies the property of a Riemannian manifold. Given a tangent space defined on a point $p \in \mathcal{M}$, we can define a basis $\{e_1(p), \ldots, e_n(p) | e_i(p) \in \mathbb{R}^n\}$. Hence we can define a inner product $\langle \cdot, \cdot \rangle$ that measures the linear interactions between two tangent vectors at each point, which is the metric tensor that

map $T_p\mathcal{M} \times T_p\mathcal{M}$ to $\mathbb{R}$. A metric tensor can be uniquely determined by the values of the inner product among its basis, since

$$\langle \mathbf{v}, \mathbf{w} \rangle_p = \langle \sum_i \mathbf{e}_i v^i, \sum_j \mathbf{e}_j w^j \rangle_p = \sum_i \sum_j v^i \langle \mathbf{e}_i, \mathbf{e}_j \rangle_p w^j, \forall v, w \in T_p\mathcal{M}. \tag{2.18}$$

If we define the matrix $G = (g_{ij})_p$ with entries

$$g_{ij} := \langle \mathbf{e}_i, \mathbf{e}_j \rangle_p, \tag{2.19}$$

then $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^T G \mathbf{w}$. Hence the metric tensor at a point $p \in \mathcal{M}$ can be denoted by the bilinear transformation $ds^2 : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$. In the covector space, a metric tensor (known as the covariant tensor) can also be denoted by the mapping $T_p^*\mathcal{M} \times T_p^*\mathcal{M} \to \mathbb{R}$, namely

$$ds^2 \equiv \langle dr, dr \rangle_g = \sum g_{\alpha\beta} dx^\alpha dx^\beta \tag{2.20}$$

Note that here the basis $e_i, i = 1, \ldots, n$ in Eq. 2.19 do not necessarily correspond to a coordinate basis. In the case that the basis is coordinate basis, *i.e.*, $e_i = \partial_i := \partial/\partial x^i$, we then have the metric tensor:

$$g_{ij}(x) = \langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \rangle \tag{2.21}$$

According to the change of variable rule, the transformation rule for the components of the metric between two charts should satisfy:

$$g_{ij}^\alpha = \sum_{rs} (\frac{\partial x_\beta^r}{\partial x_\alpha^i})(\frac{\partial x_\beta^s}{\partial x_\alpha^i}) g_{rs}^\beta. \tag{2.22}$$

The metric tensor allows us to measure the linear interactions between two tangent vectors. On a regular 2-manifold surface, the metric tensor is also named the **First Fundamental Form**[3], which can be defined as

$$ds^2 = \langle dr, dr \rangle = \begin{pmatrix} du & dv \end{pmatrix} \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix}, \tag{2.23}$$

Given the first fundamental form, we can compute quantities on the surface like angles between two vectors ($\frac{\langle A,B \rangle}{|A||B|}$), lengths ($|A| = \sqrt{\langle A, A \rangle}$) and areas ($dS = \sqrt{g_{11} * g_{22} - g_{12}^2} du \wedge dv$), etc.

[3]Note that the first fundamental form is not a differential form, but just a tensor product.

The tensorial functions defined above are measures for a single surface. When there is a mapping between two surfaces $\mathcal{M} \xrightarrow{F} \mathcal{N}$, these tensorial functions are changed accordingly. In the 0-form case, a function $f : \mathcal{N} \to \mathbb{R}$ is changed to a function $f \circ F : \mathcal{M} \to \mathbb{R}$. In the general n-form case, the change of tensorial function can be represented by the **pull-back** operation. The pull-back of a covariant tensor α^n is defined as follows:

$$F^*\alpha^n(\mathbf{v}_1, \ldots, \mathbf{v}_n) := \alpha^n(F_*\mathbf{v}_1, \ldots, F_*\mathbf{v}_n). \tag{2.24}$$

Here F^* is the pull-back operator on the differential form α:

$$F^* : \wedge^n\mathcal{N} \to \wedge^n\mathcal{M}. \tag{2.25}$$

As an example, we consider the metric tensor of Eq. 2.23. If a function maps $F : (U, V) \to (u, v)$, we have

$$F^*(ds^2) = \frac{\partial u}{\partial U}\frac{\partial v}{\partial U}dU^2 + (\frac{\partial u}{\partial U}\frac{\partial v}{\partial V} + \frac{\partial u}{\partial V}\frac{\partial v}{\partial U})dUdV + \frac{\partial u}{\partial V}\frac{\partial v}{\partial V}dV^2 \tag{2.26}$$

In elasticity, the difference $F^*(ds^2) - (dS^2)$ is defined as the *Lagrange deformation tensor*, where $dS^2 = dU^2 + dV^2$.

2.1.3 Curvatures and geodesics

As we have shown, the first fundamental form (metric tensor) allows us to measure angles, lengths or areas on the surface that involve first derivatives. These measurements only require the knowledge of the metric tensor at each point without knowing the exact location of the surface in its extrinsic space, namely they are intrinsic measures. There are also other quantities that we can measure intrinsically. Before discussing these intrinsic quantities, we first introduce the concept of intrinsic derivative.

First of all, let us define the **intrinsic derivative** or the **covariant differential** of a vector field $\mathbf{X}$, with respect to another vector field v, to be the tangent part of the change of $\mathbf{X}$ by moving it along v, namely

$$\nabla_{\mathbf{X}}v := \frac{d\mathbf{X}(\gamma(t))}{dt} - \langle\frac{d\mathbf{X}(\gamma(t))}{dt}, \mathbf{N}\rangle\mathbf{N}|_{t=0}. \tag{2.27}$$

Here $\gamma(t)$ is the curve whose tangent is v at $t = 0$. It is not difficult to show that the following relations holds for a covariant differentiation

$$\nabla_{\mathbf{X}}(a\mathbf{v} + b\mathbf{w}) = a\nabla_{\mathbf{X}}\mathbf{v} + b\nabla_{\mathbf{X}}\mathbf{w} \tag{2.28}$$

$$\nabla_{a\mathbf{X}+b\mathbf{Y}}\mathbf{v} = a\nabla_{\mathbf{X}}\mathbf{v} + b\nabla_{\mathbf{Y}}\mathbf{v} \tag{2.29}$$

$$\nabla_{\mathbf{X}}(f\mathbf{v}) = \mathbf{X}(f)\mathbf{v} + f\nabla_{\mathbf{X}}\mathbf{v} \tag{2.30}$$

If we let $e = (e_1, \ldots, e_n)$ be a frame of n linearly independent smooth vector fields on the surface such that $\mathbf{X} = \sum_i e_i X^i$, then we have

$$\nabla_{\mathbf{X}}(e_k v^k) = \sum_j \{\sum_i X^j e_i \omega^i_{jk} v^k + X^j e_j(v^k)) e_k\}. \tag{2.31}$$

Here ω^i_{jk} is defined by

$$\nabla_{e_j} e_k = e_i \omega^i_{jk}, \tag{2.32}$$

corresponding to the *Christoffel symbols* when e is a coordinate frame. Note that the operator $\nabla_{\mathbf{X}}\mathbf{v}$ sends two vectors to $\mathbb{R}$, and it is linear with respect to $\mathbf{X}$ (but not $\mathbf{v}$). Therefore we can think of it as a 1-form acted on $\mathbf{X}$ and such 1-form can be defined by

$$\nabla \mathbf{v}(\mathbf{X}) := \nabla_{\mathbf{X}}\mathbf{v} = \sum_i e_i \{dv^i + \sum_{jk} \omega^i_{jk} \sigma^j v^k\}(\mathbf{X}), \tag{2.33}$$

where σ^j is the dual basis of $\mathbf{e}$ defined by $\sigma^i(e_j) = \delta^i{}_j$. Also in here we use the notion of **vector-valued p-form** that sends vectors to vectors, ***i.e.***

$$\alpha(v_1, \ldots, v_p) := \sum_i e_i \alpha^i(v_1, \ldots, v_p). \tag{2.34}$$

Furthermore, if we define the **connection 1-form** by

$$\omega^k{}_j := \sum_r \omega^k_{rj} \sigma^r, \tag{2.35}$$

then we have

$$\nabla e_j = \sum_k e_k \otimes \omega^k{}_j \tag{2.36}$$

In vector forms, one has

$$\nabla \mathbf{e} = \mathbf{e} \otimes \omega, \tag{2.37}$$

where ω is the $n \times n$ matrix of connection 1-forms $(\omega^i{}_j)$. Under this notation, if $\mathbf{v} = \mathbf{e}v$ is a vector, we have

$$\nabla \mathbf{v} = \mathbf{e} \otimes (dv + \omega v), \tag{2.38}$$

which is vector-valued 1-form. Intuitively, $\nabla \mathbf{v}(\mathbf{X})$ describes the change of a tangent vector $\mathbf{X}$ when moved an infinitesimal distance along a given direction $\mathbf{v}$. Simply using the information

provided by $\nabla \mathbf{v}$ does not give us much information about the shape of the surface. In order to measure how much the surface $\mathcal{M}$ deviates from being Euclidean, the following **Riemannian curvature tensor** is defined:

$$R(\mathbf{X}, \mathbf{Y}) := \nabla_{\mathbf{Y}} \nabla_{\mathbf{X}} - \nabla_{\mathbf{X}} \nabla_{\mathbf{Y}} + \nabla_{[\mathbf{X},\mathbf{Y}]}, \tag{2.39}$$

where $[\mathbf{X}, \mathbf{Y}]$ is the Lie bracket. Intuitively, it measures the amount of deviation of a vector by "walking" (rigorously, one should parallel displace a vector) along the directions $\mathbf{X} \to \mathbf{Y} \to -\mathbf{X} \to -\mathbf{Y}$. If the surface is flat, such number should always be zero. Alternatively, one can characterize the curvature of a surface by the **curvature** 2**-forms**, which is defined by

$$\theta := d\omega + \omega \wedge \omega. \tag{2.40}$$

It is derived by the 2nd-order covariant differentition $\nabla\nabla\mathbf{e} = \nabla(\mathbf{e}\omega) = \mathbf{e}(d\omega + \omega \wedge \omega) = \mathbf{e}\theta$, where $\theta^i{}_j = d\omega^i{}_j + \sum_k \omega^i{}_k \wedge \omega^k{}_j$. One can verify that $\theta = 0$ if and only if $R(\mathbf{X}, \mathbf{Y}) = 0$ for all $\mathbf{X}$ and $\mathbf{Y}$.

Given the covariant differentiation, a parametrized curve $\gamma : [0, t] \to \mathcal{M}$ is a **geodesic** curve if $\nabla_T T = 0$ at every point, where T is the tangent of the curve γ.

2.1.4 Complexes and homology

Complexes, reflects the mathematical abstraction that a complex system is composed of basic building blocks. For example, the cell complex is an abstraction of the smooth manifold surfaces. Algebraic properties (*e.g.*, topology) of the surface can be studied under the discrete setting using complexes. There are various types of complexes introduced for different purposes, such as chain complexes, cell complexes, Vietoris-Rips complexes [259], $\check{C}$ech complexes. In this dissertation, we only introduce the chain complexes that is closely related to the discrete representations of surfaces. The language of complexes allows us to directly link many continuous geometric theories to their discrete counterparts.

Simplicial complex

Let S be a discrete set and 2^S be its power set, an **abstract simplicial complex** is a collection of finite subsets of S, $X \subset 2^S$ satisfying the closeness property: for each $\sigma \in X$, all subsets of σ are also in X. Each element $\sigma \in X$ is called a **simplex**, or rather a k**-simplex** if $|\sigma| = k + 1$. Given a k-simplex σ, its **faces** are the simplices corresponding to all subsets of $\sigma \subset S$. For example, in a graph, the **edges** are 1-simplices of the graph. The simplicial complexes can be considered as generalizations of the concept of *graphs*, namely graphs are simplicial

complexes without simplices of dimension higher than one. However, if we allow a graph to have edges with more than 2 nodes (hyper-edges), a simplicial complex become another term for a (hyper-)graph.

The definition of simplicial complex above is rather abstract. We can associate with an abstract k-simplex a geometric meaning by assigning a coordinate to each of its elements. The **standard Euclidean k-simplex** (Fig. 2.4) can be defined to be the space:

$$\Delta^k = \{x \in [0,\infty)^{k+1} : \sum_{i=0}^{k} x_i = 1\}. \tag{2.41}$$

Here Δ^k can be regarded as being composed of copies of Δ^j for $j < k$ via restricting them to subspaces of $\mathbb{R}^{k+1}$. More generally, a **k-simplex** σ_k can be realized as the non-degenerate convex hull of $k+1$ vertices in an n-dimensional space, if we associate each element in σ_k with a coordinate in $\mathbb{R}^n$, namely $(v_0, \dots v_k) \in \mathbb{R}^n$ with $n \geq k$, which is represented as

$$\sigma_k = \{x \in R^n | x = \sum_{i=0}^{k} \alpha^i v_i \text{ with } \alpha^i \geq 0 \text{ and } \sum_{i=0}^{k} \alpha^i = 1\}. \tag{2.42}$$

A k-simplex can also be realized into a k-manifold: $\sigma_k : \Delta_k \to \mathcal{M}^k$. Although the definition of Eq. 2.42 is independent of the order of the elements $v_i, i = 0, \dots, k$, in many situations such order does matter. One example is the directed graph where the edge is a 1-simplex; another example is the orientation of the face of a surface, which is determined by the even or odd permutations of $(v_0, v_1 \dots, v_k)$ from a canonical order.

In order to see the relation between a simplicial simplex X and its realizations as standard k-simplices, we define the k-**skeleton** of X, $k \in \mathbb{N}$, to be the quotient space:

$$X^{(k)} = (X^{(k-1)} \bigcup \coprod_{\sigma:\dim\sigma=k} \Delta^k)/\sim, \tag{2.43}$$

where $\sim$ is the equivalence relation that identifies faces of Δ^k to the corresponding faces of σ in $X^{(j)}$ for $j < k$. By this construction, we can see that $X^{(0)} = S$ is a discrete set; the 1-skeleton $X^{(1)}$ is a space homeomorphic to a collection of vertices and edges. It is easy to see that $X^{(k)} \supset X^{(k-1)}$ and $X = \cup_{k=0}^{\infty} X^{(k)}$.

Chain complex

Given a k-simplex $\sigma_k = (v_0, v_1 \dots, v_k)$, we can define its **boundary** operator $\partial\sigma_k$ as follows:

$$\partial\sigma_k := \sum_{i=0}^{k} (-1)^i (v_0, \dots \widehat{v_i} \dots, v_k) = \sum_{i=0}^{k} (-1)^i \sigma_{k-1}^{(i)}, \tag{2.44}$$

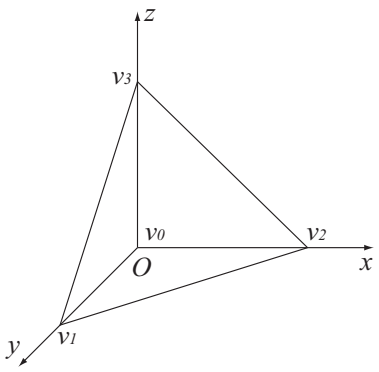


Figure 2.4: A standard Euclidean k-simplex.

where $\sigma_{k-1}^{(i)}$ denotes the ith face of σ_k and $\partial\sigma_0 = 0$. From the definition of the boundary operator, we can derive the following important lemma.

Lemma 1 *The boundary of a boundary is null:* $\partial_{k-1} \circ \partial_k = 0$ *for all* k.

Note that $\partial\sigma_k$, as a sum of $(k-1)$-simplices with coefficients ± 1, is no longer a simplex. Here the coefficients $\{+1, -1\}$ denote the orientation of each of the $(k-1)$-simplices, and itself forms a group. Extending this idea, given a coefficients in the abelian group G, we can define a k**-chain** to be the sum of a finite number of k-simplices

$$c_k = g_1\sigma_k^1 + g_2\sigma_k^2 + \ldots + g_r\sigma_k^r, g_i \in G. \tag{2.45}$$

The collection of all the k-chains with coefficients in G form a k**-chain group**, denoted as $C_k(\mathcal{M}^n; G)$ if a k-chain is realized by a n-manifold $\mathcal{M}^n$ and $C_k(\mathcal{G}, G)$ if a k-chain is realized by a graph $\mathcal{G}$.

The boundary operator for a k-chain can be defined by

$$\partial \sum_r g_r\sigma_k^r = \sum_r g_r \partial\sigma_k^r. \tag{2.46}$$

Thus if the coefficient group G is a field, the boundary operator should be a *linear transformation*. If the number of all k-simplices that form a k-chain is n_k and the number of all $(k-1)$-simplices that form a $(k-1)$-chain is n_{k-1}, then the boundary operator for a k-chain can be denoted by a $n_{k-1} \times n_k$ matrix. The collection of chains and boundary maps is assembled into a **chain complex**:

$$\ldots \to C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \ldots C_0 \xrightarrow{\partial_0} 0 \tag{2.47}$$

For example, for a single n-simplex $C_n = \Delta^n$, it is not difficult to verify that its chain complex C_k has dimension $\binom{n+1}{k+1}$.

Homology

Now we consider the classification of k-chain groups. To this purpose, we should first define certain equivalent relations. First of all, a **homomorphism** of an abelian group G into an abelian group H is a map $f : G \to H$ such that $f(g + g') = f(g) + f(g')$ for all $g, g' \in G$. It can be seen that for any mapping between two manifolds $F : \mathcal{M} \to \mathcal{N}$, a homomorphism $F_* : C_k(\mathcal{M}; G) \to C_l(\mathcal{N}; G)$ can be defined by

$$F_*(g_1\sigma_k^1 + \ldots + g_r\sigma_p^r) := g_1(F \circ \sigma_k^1) + \ldots + g_r(F \circ \sigma_p^r). \tag{2.48}$$

Using this construction, we can define the **boundary homomorphism** $F_* = \partial : C_k(\mathcal{M}; G) \to C_{p-1}(\mathcal{M}; G)$, for the boundary operator as follows

$$\partial \sum_r g_r\sigma_k^r := \sum_r g_r \partial\sigma_k^r \tag{2.49}$$

In order to classify a k-chain groups $C_k(\mathcal{M}; G)$, we consider the **kernel** and **image** with respect to the boundary homomorphism (one possible intuition is from the property that $\partial^2 = 0$). To this end, we define a k**-cycle** to be a k-chain whose boundary is 0, namely

$$Z_k(\mathcal{M}; G) := \{z_k \in C_k | \partial z_k = 0\} = \text{ker } \partial : C_k \to C_{k-1}. \tag{2.50}$$

When G is a field, Z_k is a vector subspace of C_k, resembling the kernel space of a linear transformation with respect to ∂ (*i.e.*, if a linear mapping is denoted by $f : x \to Ax$, its kernel space is the set $\{x|Ax = 0\}$).

Similarly, we define the k**-boundary** of a k-chain to be those k-chains that are the boundaries of some $(k + 1)$-chains, *i.e.*,

$$B_k(\mathcal{M}; G) := \{\beta_k \in C_k | \beta_k = \partial c_{k+1}, \text{for some } c_{k+1} \in C_{k+1}\} = \text{Im } \partial : C_{k+1} \to C_k. \tag{2.51}$$

Obviously, since $\partial^2\beta = 0$, $B_k \subset Z_k$. And B_k is a subspace of Z_k. Hence their quotient group Z_k/B_k itself forms a subspace. The k**th homology group** of a chain complex, can therefore be defined as

$$H_k(\mathcal{M}; G) := \frac{Z_k(\mathcal{M}; G)}{B_k(\mathcal{M}; G)} = \frac{\text{ker } \partial_k}{\text{im } \partial_{k+1}} = \frac{\text{cycles}}{\text{boundaries}}. \tag{2.52}$$

The class of homology groups $\{H_i(\mathcal{M}; G)\}$ is called **homology classes**.

Here we give two important examples of homology classes. The homology class $H_0(\mathcal{G}; \mathbb{F})$ of a graph $\mathcal{G}$ is the number of connected components, and $H_1(\mathcal{G}; \mathbb{F})$ is the number of independent cyclic chains of edges (cycles). For a 2-manifold $\mathcal{M}$ (connected compact orientable with genus g), we can derive $H_0(\mathcal{M}; \mathbb{F}) = 1$, $H_1(\mathcal{M}; \mathbb{F}) = 2g$, $H_2(\mathcal{M}; \mathbb{F}) = 2$.

2.1.5 Harmonic forms and Hodge decomposition

Recall that in Sec. 2.1.2 we introduced the concept of *dual space* to a vector space V as the linear functionals $V \to \mathbb{R}$. Since a k-chain $C_k(\mathcal{M};\mathbb{R})$ can be considered as a vector space, we can therefore define its dual space, namely the k-**cochain** $C^k(\mathcal{M};\mathbb{R}) := C_k^*(\mathcal{M};\mathbb{R})$. A k-cochain is a linear functional $f : C^k \to \mathbb{R}$.

Also given a linear transformation $A : V \to W$, we can associate to it a **adjoint map** $A^T : W^* \to V^*$, defined by

$$(A^T(f))(v) = f(A(v)), \forall v \in V. \tag{2.53}$$

Note that the adjoint map is in analogy to the pull-back function defined in Sec. 2.1.2 for the differential forms. The diagram below illustrates the relations among dual spaces and adjoint maps.

$$\begin{array}{ccc} V & \xrightarrow{\quad A \quad} & W \\ \Big\updownarrow \text{dual} & & \Big\updownarrow \text{dual} \\ V^* & \xleftarrow[\quad A^T \quad]{} & W^* \end{array}$$

As a chain is associated with a boundary operator ∂, a cochain can be associated with a **coboundary** δ, which is defined as the adjoint of ∂, such that $\delta\alpha(c) = \alpha(\partial c), \forall \alpha \in C^k$ and $c \in C_k$. Thus, if a boundary operator ∂_k sends C_k to C_{k-1}, its corresponding coboundary operator δ^{k-1} sends C^{k-1} to C^k. Also from $\partial \circ \partial = 0$, we obtain $\delta \circ \delta = 0$.

Similar to the chain complex, we can define the **cochain complex** as the collection of cochains and coboundary maps, *i.e.*,

$$C^0 \xrightarrow{\delta^0} C^1 \xrightarrow{\delta^1} \dots \xrightarrow{\delta^{k-1}} C^k \tag{2.54}$$

An important consequence of considering the cochain complex is we can define the analogy of the homology class, namely the **cohomology** class:

$$H^k = \frac{\ker \delta^k}{\text{im } \delta^{k-1}} = \frac{\text{cocircles}}{\text{coboundaries}}. \tag{2.55}$$

In the case of $H^1(\mathcal{M}^2, \mathbb{R})$, the cocircles are the functional α^1 defined on the edges of the surface such that for any face $F \subset \mathcal{M}^2$, $\alpha^1(\partial F) = 0$. And the coboundaries are the functional on $\mathcal{M}$ whose integration along any close curve on the surface are zero. Although the dimension of H^1 is the same as H_1, it is calculated based on a different aspect from the homology.

For the relationship between homology and cohomology, we have the following theorem.

Theorem 2 ***(Universal Coefficient)****. For $\mathbb{F}$ a field, we have $H^n(\mathcal{M};\mathbb{F}) \cong H_n^*(\mathcal{M};\mathbb{F})$.*

Although cohomology can be determined by homology and certain algebraic properties of the coefficient, considering the dual space often simplifies the algebraic difficulty as in the case of differential forms.

Theorem 3 ***(Poincaré duality)****. For $\mathcal{M}$ a manifold, there is a natural isomorphism $H_k(\mathcal{M};\mathbb{F}_2) \to H^{n-k}(\mathcal{M};\mathbb{F}_2)$. In addition, if $\mathcal{M}$ is compact, $H_k(\mathcal{M};\mathbb{F}_2) \cong H^{n-k}(\mathcal{M};\mathbb{F}_2)$.*

Harmonic forms

As we have shown, the coboundary δ^k maps C^k to C^{k+1}. If C^k is defined in a vector space and there exists a metric (inner product) for such vector space, the **adjoint** of this mapping, is another linear map that sends C^k to C^{k-1}. With an abuse of terms, such adjoint mapping is referred to as the **boundary** operator.

We can define the **laplacian** $\Delta : C^k \to C^k$ by

$$\Delta_k = \delta^*\delta + \delta\delta^* = (\delta + \delta^*)^2 \tag{2.56}$$

A cochain α is called **harmonic** iff $\Delta\alpha = 0$, *i.e.*, $\delta^*\alpha = \delta\alpha = 0$.

Hodge decomposition

One can think of the harmonic form α as the smoothest differential forms defined on the surface. In fact, the **Hodge decomposition** states that any k-form can be split into three components:

$$\beta^p = \delta\alpha^{p-1} + \delta^*\gamma^{p+1} + \mathfrak{h}^p, \tag{2.57}$$

where $\mathfrak{h}^p$ is harmonic. Since $d^2\alpha = 0$ and $\delta\delta\gamma = 1$, the Hodge decomposition actually states that any k-form can be decomposed into a closed form, a co-closed form and a harmonic form.

To summarize, in this section we have introduced the concepts in Riemannian geometry and Algebraic topology. Riemannian geometry has provided us with a number of useful metrics (*e.g.*, curvature, geodesics) to define meaningful measures for the plausibility of any mapping between two surfaces, while Algebraic topology allows us to connect continuous structures with discrete (combinatorial) structures. In the following, we shall explain how these concepts can be applied to the surface registration problem.

2.2 Conformal geometry and surface matching

As discussed in Sec. 1.1.2, geometry allows us to study the properties of the shape that are invariant under different equivalent transformations. With the concepts introduced in the previous section, now we can rigorously define these transformations. In Riemannian geometry, the equivalent transformation is the *isometric mapping* which is defined below.

Definition 5 ***(Isometric mapping)*** *A diffeomorphism $f : \mathcal{M} \to \mathcal{N}$ between two surfaces equipped with Riemannian metrics g and $\tilde{g}$ is an isometry if for all $p \in \mathcal{M}$ and all $v_1, v_2 \in T_p\mathcal{M}$ we have*

$$\langle df_p(v_1), df_p(v_2)\rangle_{\tilde{g}} = \langle v_1, v_2\rangle_g \tag{2.58}$$

The surfaces $\mathcal{M}$ and $\mathcal{N}$ are then said to be isometric.

Many real-world deformations has proven to be isometric or near isometric, such as the deformations of clothes or faces [37, 211]. Since the isometric mapping simply requires that the Riemannian metric between two surfaces be preserved, any features that are defined using the Riemannian metric can be used for defining the measure of the mapping between two surfaces. For example, one can define the similarity in *Gaussian curvature* (*i.e.*, $|R(p) - R(f(p))|$) to compare the plausibility of any pair of matchings $p \to f(p)$, or the similarity in *geodesic distances* (*i.e.*, $|d(p,q) - d(f(p), f(q))|$) to define the plausibility of any two pairs of matchings $(p,q) \to (f(p), f(q))$. However, these measures are only local measures and it is difficult to find an efficient way to recover the mapping f for the entire surface using these measures. Hence, when the number of matching points goes large (*e.g.*, > 100), computational complexity becomes an issue.

In contrast, **Conformal geometry** (also known as **Riemann Surface**) provides us an elegant way of recovering the transformation f by only knowing a few correspondences, if the surface is topologically equivalent to a sphere. Mathematically, the equivalent transformation in Conformal geometry is the conformal mapping which can be defined as follows.

Definition 6 ***(Conformal mapping)*** *A diffeomorphism $f : \mathcal{M} \to \mathcal{N}$ between two surfaces equipped with Riemannian metrics g and $\tilde{g}$ is called conformal map if for all $p \in \mathcal{M}$ and all $v_1, v_2 \in T_p\mathcal{M}$ we have*

$$\langle df_p(v_1), df_p(v_2)\rangle_{\tilde{g}} = \lambda^2(p)\langle v_1, v_2\rangle_g \tag{2.59}$$

where $\lambda^2(\cdot)$ is a nowhere-zero differentiable function on $\mathcal{M}$. Conventionally, conformal mapping can also be represented as

$$\tilde{g} = e^{2\mu}g, \tag{2.60}$$

where $\mu(p) = \log(\lambda(p))$.

Geometrically, conformal mapping is a map that preserves the angle between any two tangent vectors whereas isometric mapping preserves not only the angles but also the lengths of any tangent vector. Hence conformal mapping is less "rigid" than isometric mapping, and therefore more flexible. Two surfaces are **conformally equivalent** if there exists a conformal mapping between them. Such an equivalence class has been developed into a new geometry, namely the *Riemann surface* [73], which is rich in many useful properties. ◁

In the next section, we introduce the important properties in Riemann surface, justifying its usefulness in surface matching. Specifically, the following arguments are established:

1. Conformal mapping allow us to map any two-manifold into one of three canonical domains: the unit disk, the sphere or the hyperbolic plane (The Uniformization theorem).
2. If there exists an isometric mapping between two surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$, then if we map them to their canonical domain $\mathcal{C}$, the set of bijective mappings that map $\mathcal{C}$ to itself can be represented in a closed form solution.
3. Such closed-form solution has very low degrees of freedom, hence, *the mapping between the two surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$ can be represented in a closed form* . ◁

2.2.1 Riemann surfaces

In Riemannian geometry, a manifold $\mathcal{M}$ is equipped with a Riemannian metric g, such that the quantities (lengths, curvatures) are preserved by the transition map (Eq. 2.1). Such set of mappings can be characterized by the isometric mapping (Def. 5). Similarly, in Riemann surface theory, a manifold is equipped with atlases whose transition maps are characterized by the conformal mapping (Def. 6) between two $\mathbb{R}^2$ domains. Such mappings can also be conveniently represented in the complex domain $\mathbb{C}$, where each point $(x, y) \in \mathbb{R}^2$ is represented by a point $z = x + iy \in \mathbb{C}$. The set of conformal mappings among such complex domains are characterized by the *holomorphic* function which is defined as follows

Definition 7 ***(Holomorphic or Analytical Function)*** *A complex function $f : \mathbb{C} \to \mathbb{C}$ is holomorphic function if the* Cauchy-Riemann *equation is satisfied:*

$$\frac{\partial f}{\partial \overline{z}} = 0 \tag{2.61}$$

where

$$\frac{\partial}{\partial z} := \frac{1}{2}(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}), \qquad \frac{\partial}{\partial \overline{z}} := \frac{1}{2}(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}). \tag{2.62}$$

Given the definition of holomorphic functions for the complex plane, we can now define conformal structure for surfaces. Let us define the **complex chart** for a topological space [78] $\mathcal{M}$ to be a homeomorphism $\phi : U \to V$, where $U \subset \mathcal{M}$ is an open set and $V \subset \mathbb{C}$ is an open set in the complex plane. We say that two complex charts are **compatible** if either $U_1 \cap U_2 = \emptyset$ or the transition map

$$f_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta) \tag{2.63}$$

is holomorphic[4]. Hence, a **complex atlas** on $\mathcal{M}$ is a collection of pairwise compatible complex charts whose domain cover $\mathcal{M}$. A **complex (or conformal) structure** on $\mathcal{M}$ is a maximal complex atlas on $\mathcal{M}$ or, equivalently, an equivalence class of complex atlases on $\mathcal{M}$. Formally, we can define the Riemann surface as follows:

Definition 8 ***(Riemann surface)*** *A Riemann surface is a topological space $\mathcal{M}$ with a complex structure.*

A Riemann surface can be also studied algebraically. For example, if we define a polynomial function

$$P(z, w) = w^n + p_{n-1}(z)w^{n-1} + \cdots + p_1(z)w + p_0(z), \tag{2.64}$$

its graph

$$S = \{(z, w) \in \mathbb{C}^2 | P(z, w) = 0\} \tag{2.65}$$

can be considered as a surface rather than as a multi-valued function of z [243] and a Riemann surface can be regarded as the graph of a multi-valued complex-analytic function locally. An interesting question in Riemann surface is: Can any surface be given the structure of a Riemann surface (equip it with conformal structure)?Not surprisingly, the answer is that any *orientable* surface can be equipped with a conformal structure. Hence, any real-world surfaces can be studied using Riemann surface theory. ◁

An important example in Riemann surface is the **Riemann sphere**, whose bijective correspondence to the domain $\mathbb{C} \cup \{\infty\} = \mathbb{P}^1$ is illustrated in Fig. 2.5. In this case, $\mathbb{P}^1$ is understood as a Riemann surface by regarding the graph $z^{-1} = w$ as a local coordinate near ∞. In general, to study holomorphic maps with target space $\mathbb{P}^1$, the *meromorphic function* can be defined as follows:

[4]Following the property of holomorphic functions, it can be derived that if $f_{\alpha\beta}$ is a transition function between two compatible charts, then its derivative $\frac{\partial f}{\partial z}$ is never zero.

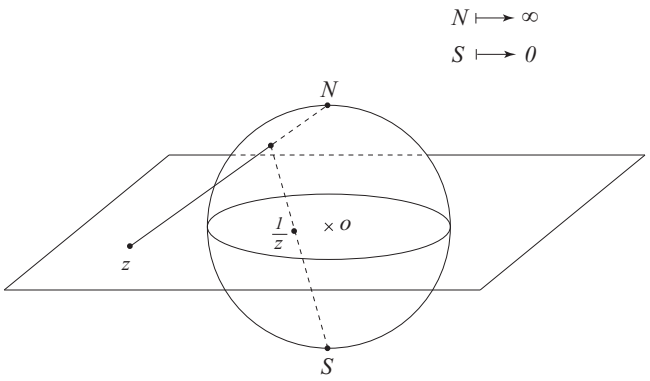


Figure 2.5: An illustration of the Riemann sphere

Definition 9 ***(Meromorphic function)*** *A function $f : U \to \mathbb{C} \cup \{\infty\}$ ($U \subset \mathbb{C}$ is open) is called meromorphic if it is holomorphic at every point where it has a finite value, whereas, near every point z_0 with $f(z_0) = \infty$, $f(z) = \phi(z)/(z - z_0)^n$ for some holomorphic function $\phi(\cdot)$, defined and non-zero around z_0.*

Here the positive number n is the *order of the pole* at z_0. It can be proven that every meromorphic function on $\mathbb{P}^1$ can be represented as a rational function $R(z) = p(z)/q(z)$ where $p(\cdot)$ and $p(\cdot)$ are polynomial functions. The *degree* of $R(z)$ is defined as $\max(\deg p(z), \deg q(z))$. Here $\deg(\cdot)$ denotes the degree of a function as we shall define in the following (Equ. 2.66).

In order to characterize the mapping between two Riemann surfaces, the first step is to characterize the local holomorphic functions between them. Let $f : \mathcal{M} \to \mathcal{N}$ be a non-constant holomorphic map between two compact, connected Riemann surfaces, with $f(r) = s$ and f not constant near r. Then, given any mapping $\psi : V_s \to \mathbb{D}$ of a sufficiently small neighborhood of $s \in \mathcal{M}$ with the unit disc $\mathbb{D}$, sending s to 0, there exists an analytic identification $\phi : U_r \to \mathbb{D}$ of a suitable neighborhood U_r of r such that the following diagram commutes: ◁

$$\begin{array}{ccc} U_{\mathcal{M}} & \xrightarrow{\quad f \quad} & V_{\mathcal{N}} \\ \downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle \psi} \\ \mathbb{D} & \longleftarrow & \mathbb{D} \end{array}$$

$$z \longmapsto z^n$$

That is, locally, f looks like the map $z \mapsto z^n$. It can be proven that this number n does not depend on the choice of neighborhoods and is called the **valency** of f at r, denoted as $v_f(r)$.

Valency is useful in characterizing the mapping f. It can also be proven that the number

$$\deg(f) := \sum_{r \in f^{-1}(s)} v_f(r) \tag{2.66}$$

is independent of the choice of point $s \in \mathcal{M}$, and is called the **degree** of the map f. The order of zeros and poles of a meromorphic function between two surfaces are related by the topology of the surface, specifically, we have the following

Theorem 4 ***(Riemann-Hurwitz formula)*** *Let $f : \mathcal{M} \to \mathcal{N}$ be a non-constant holomorphic map between compact connected Riemann surfaces, then the* total branching index *b of f is*

$$b := \sum_{s \in \mathcal{M}} \sum_{r \in f^{-1}(s)} (v_f(r) - 1) = \sum_{s \in \mathcal{N}} (deg(f) - |f^{-1}(s)|). \tag{2.67}$$

The following Riemann-Hurwitz formula holds

$$\chi(\mathcal{M}) = deg(f)\chi(\mathcal{N}) - b, \tag{2.68}$$

where $\chi(\mathcal{M}) = 2 - 2g(\mathcal{M})$ is the Euler characteristic.

In another word, the zeros and poles of a meromorphic function determines the topology of its graph, which can be calculated analytically. Furthermore, a more interesting question is that if one can find a holomorphic function that map arbitrary $\mathcal{M}$ to arbitrary $\mathcal{N}$ whose Euler characteristics satisfy Eq. 2.68. In his dissertation, Riemann partially addressed such problem, known as the Riemann mapping theorem:

Theorem 5 ***(Riemann mapping theorem)*** *If U is a non-empty simply connected open subset of the complex number plane $\mathbb{C}$, then there exists a biholomorphic (bijective and holomorphic) mapping f from U onto the open unit disk*

$$\mathbb{D} = \{z \in \mathbb{C} || z| < 1\} \tag{2.69}$$

Later on, the Riemann mapping theorem has been generalized to the context of Riemann surfaces, leading to the more general Uniformization theorem as stated below:

Theorem 6 ***(Uniformization theorem)*** *Any simply connected Riemann surface is conformally equivalent to one of the three domains: the open unit disk $\mathbb{D}$, the complex plane $\mathbb{C}$, or the Riemann sphere $\mathbb{P}^1$. In particular it admits a Riemannian metric of constant curvature.*

The uniformization theorem guarantees that any two-manifold can be conformally mapped to one of the three canonical domains: $\mathbb{D}$, $\mathbb{C}$ or $\mathbb{P}^1$. One consequence the uniformization theorem tells us is that any surface can be equipped with **isothermal coordinates**, whose metric is represented as $g = e^{2\lambda}(dx^2 + dy^2)$. This would allow us to endow an Euclidean coordinate to a surface with whose metric is conformal to the original surface. ◁

Another consequence of the uniformization theorem is its application in registering two surfaces that are isometrically deformed. To this end, we have the following result ◁

Theorem 7 ***(Isometry and conformal mapping)*** *If two surfaces are isometrically deformed from one to the other, then their canonical domain must only differ by a conformal mapping*[5].

To see this, let us assume that $f : \mathcal{M} \to \mathcal{N}$ is an isometric deformation and by conformally mapping $\mathcal{M}$ and $\mathcal{N}$ to their canonical domains, each of the two surfaces is equipped with an isothermal coordinate: $g_{\mathcal{M}} = e^{2\lambda_1}(dx_1^2 + dy_1^2)$ and $g_{\mathcal{N}} = e^{2\lambda_2}(dx_2^2 + dy_2^2)$. Since f is isometric, we have $g_{\mathcal{M}} = f^* g_{\mathcal{N}}$ is identity, *i.e.*, $(dx_1^2 + dy_1^2) = e^{2(\lambda_2 - \lambda_1)}(dx_2^2 + dy_2^2)$. Thus, the mapping between (x_1, y_1) and (x_2, y_2) is conformal.

Hence, in order to study the mapping between two surfaces undergoing isometric deformations, we only have to study the mapping between their canonical domains. The mapping that maps one canonical domain to itself is represented by the *automorphism* as defined below:

Definition 10 ***(Automorphism)*** *An **automorphism** of a region S of the complex plane is a one-to-one, conformal mapping of S to itself.*

$$\mathit{Aut}(S) = \{\textit{analytic bijections} : f : S \to S\}. \tag{2.70}$$

Certainly all the automorphisms from S to S form an automorphism group.

As we have shown above, any conformal mapping (holomorphic function) can be represented as a polynomial function $p(z)$. Since automorphism requires the mapping to be one-to-one, *i.e.*, $v_r(f) = 1, \forall r \in \mathcal{M}$, if f is defined in an open set of $\mathbb{C}$, locally the mapping must be a linear function $p(z) = az + b$. Furthermore, if we consider the domain $\mathbb{C} \cup \{\infty\}$, the automorphism is then in the form of a meromorphic function $(az + b)/(cz + d)$. In fact, this class of automorphism is characterized by the celebrated **Möbius transformation** which is defined as

$$M(z) := \frac{az + b}{cz + d}, \quad a, b, c, d \in \mathbb{C}, ad - bc = 1. \tag{2.71}$$

[5]Rigorously speaking, the use of conformal mapping for surface registration handles the broader class of conformal deformation between two surfaces and isometric deformation is only a subclass of the conformal deformation.

Furthermore, when the domain S is the unit disk $\mathbb{D}$, it can be shown that the automorphism that maps $\mathbb{D}$ to $\mathbb{D}$ has the following form ([179]):

$$M(z) = e^{i\phi}(\frac{z-a}{\bar{a}z-1}), \tag{2.72}$$

where $\phi \in \mathbb{R}$ and $a \in \mathbb{C}$. Loosely speaking, Möbius transformation in Conformal geometry can be regarded as the analog of rigid transformation in Euclidean geometry. ◁

2.2.2 Discrete structure of conformal mapping

Thus far, we have discussed the continuous theories of conformal mapping. In order to compute the conformal mapping of any discretely represented surfaces (*e.g.* triangular meshes), we need to clarify the discrete meanings of the continuous concepts introduced above. **Discrete differential geometry** (DDG) is an active area in computer vision and graphics [29]. The purpose of DDG is to search for the appropriate discretizations (discrete counterparts) of the geometry theories which preserve the the fundamental properties such as invariance to certain transformation groups. Examples include discrete Gaussian curvature [205], discrete Morse theory [54], discrete connections [58], discrete Harmonicity [196] and discrete vector fields [76]. In order to find the discrete counterparts of many of those continuous concepts, certain *equivalence assumptions* are usually made. Hence, the same geometric concept might have multiple discrete counterparts proposed. Here, we introduce and compare several approaches to establish the discrete equivalence of conformal mapping. Note that an important measure of the soundness of different discretizations is to see if *the proposed approach induces approximation error*, namely if the proposed discretization leads to any difficulty in the computation. ◁

In the discrete setting, the most common representation of a surface is the triangular mesh (chain complex). As we have shown in Def. 6, the conformal mapping between two surfaces is related by their metrics. Intuitively, for a chain complex, its discrete correspondence of the continuous metric can be defined as the lengths of the edges for each face, coinciding with the fact the metric tensor measures 1D distances (Equ. 2.23). Given this discrete counterpart of the metric, we can define the discrete conformal equivalence between two surfaces $\mathcal{M}$ and $\mathcal{N}$ as follows [161]:

Definition 11 ***(Discrete conformal equivalence)*** *Two discrete metrics l and $\tilde{l}$ on $\mathcal{M}$ and $\mathcal{N}$ are discretely conformally equivalent if, for some assignment of numbers u_i to the vertices v_i , the metrics are related by*

$$\tilde{l}_{ij} = e^{(u_i+u_j)/2} l_{ij} \tag{2.73}$$

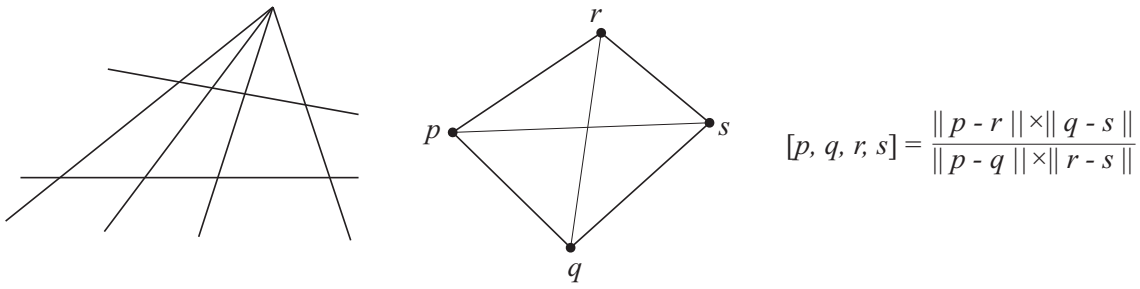


Figure 2.6: Invariant of a conformal transformation of $\mathbb{R}^n \cup \infty$: the cross-ratio of Euclidean distances.

One can verify that if we apply the Möbius transformation to the vertices of the mesh, the new transformed mesh is also discretely conformally equivalent to the original mesh.

Another way to characterize conformal mapping, which turns out to be equivalent to the definition 11, is by considering the length cross ratios: The cross ratio between two triangles $\triangle_{pqr}$ and $\triangle_{qsr}$ is defined as (Fig. 2.6)

$$[p, q, r, s] = \frac{|p - r| \cdot |q - s|}{|p - q| \cdot |r - s|}. \tag{2.74}$$

The relation between the cross ratio and the discrete conformal equivalence defined above can be established as follows:

Proposition 1 *Two meshes are discretely conformally equivalent if and only if their length cross ratios are the same.*

Given the definition of cross ratio, we can define the Möbius transformation for n-D as

Definition 12 ***(Möbius group)*** *A Möbius group $Mb(n)$ is defined to be the group of all transformations of $\mathbb{R}^n \cup \infty$ that preserve the distance cross-ratio $[p, q, r, s]$.*

It can be easily seen that a map is a Möbius transformation if and only if it preserves cross-ratios. Above we have established the discrete meaning of conformality. In the following, we introduce different approaches to computing the conformal mapping (flattening) for a surface. All of these approaches involve minimization of an energy function defined on the discrete representation of the surface. In the *curvature flow* based approach, the energy function is the curvature of the surface. In the *finite-element* based approach, the energy function is the discrete Dirichlet energy. Similarly, in the *differential 1-form* based approach, the energy function is the discrete Harmonic energy.

Curvature flow based discretization

Given the definition of discrete meaning of conformal mapping, one can conformally map a surface onto the 2D domain (conformal parametrization), where the curvature at every point is zero. To achieve this, the first step is to relate the curvature at each point to the discrete metrics of the surface. And the change of the metric can be represented by the change of the values of u_i at each point i of Equ. 2.73. In the work of [230], such flattening process is formulated as a convex minimization problem. Another approach for solving conformal parametrization is to use **circle patterns** [233]. Instead of representing the discrete metric by the lengths of edges, circle patterns consider that a circle is associated to each vertex. Under such a formulation, solving for the conformal flattening problem becomes solving for the radius of the circle at each point. However, the energy functions for the flattening process are often non-linear [107, 118], undermining the use of circle patterns for conformal flattening.

Once an energy function is defined on the surface, either using discrete metric or circle patterns, the next step is to solve for the conformal flattening problem. A popular approach is to consider **curvature flow** on the surface [107]. The idea of curvature flow is to solve the heat equation on the curvature of the surfaces, due to the fact that for a flat surface the curvature at each point on the surface is constant and the stable solution of the heat equation is also constant everywhere.

Specifically, if we denote $\tilde{K}(p)$ as the target curvature for each point p on the surface, the heat equation of curvature flow can be formulated as

$$\frac{\partial K(p,t)}{\partial t} = \Delta_p(K(p,t) - \tilde{K}(p)), \tag{2.75}$$

where Δ is the cot-Laplace operator. At the steady state, the target curvature at each point p would equal to the target curvature $\tilde{K}(p)$. Hence by setting the target curvature at each point to be a constant and proper boundary conditions, the curvature flow can map the surface onto the 2D domain. Note that if the genus of the input shape is different from zero, one needs to set the target curvature such that the Gauss-Bonnet theorem is satisfied [63]. When the genus of the surface is high (> 1), in order to map the surface onto the 2D plane, one needs to consider the hyperbolic metric [288].

The idea of curvature flow has also been used by changing the metrics on the surfaces. For example, the **Metric Scaling** approach ([19]) evolves the edge lengths of the surface by solving the following equation:

$$e^{2u}\tilde{K} = K + \Delta u. \tag{2.76}$$

Here Δ is the Laplace-Beltrami operator and e^{2u} is the conformal factor as defined in Equ. 2.60. When the target curvature is zero, the problem of conformal parametrization becomes solving

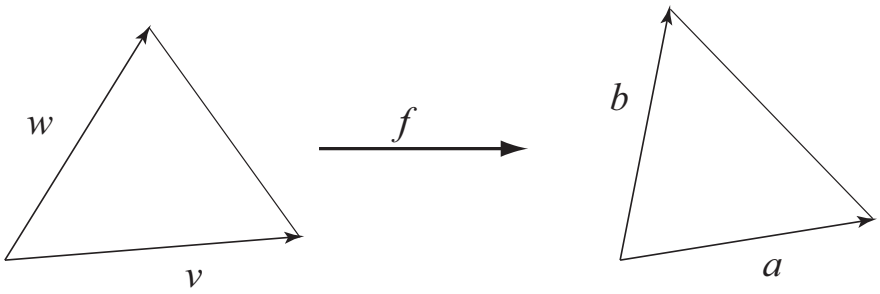


Figure 2.7: The discrete minimal surface [196] assumes the mapping between two triangulations to be linear.

the poisson equation:

$$\Delta u = -K. \tag{2.77}$$

Although this approach only requires solving a linear equation, the resulting mapping is not necessarily a flat metric. In contrast, [230] minimizes an energy that guarantees the solution is a flat metric.

One issue with the curvature flow based approaches, either using circle patterns or discrete metrics for the representation of the mesh, is the boundary boundary conditions. By setting the boundary of the surfaces to different shapes, the resulting conformal mapping could be very different. In the problem of surface registration, if the two surfaces to be matched have consistent boundary conditions, one can use those curvature flow based approaches for conformal flattening. However, in many situations, the boundary conditions are hard to be consistent or defined. In the following, we introduce two alternative numerical approaches for conformal flattening that are less sensitive to the boundary conditions.

Harmonic energy based discretization

The discrete Harmonicity based conformal flattening methods assume the surface be piecewise linear [35]. Given such an assumption, one can hence compute continuous properties of the surface such as the Laplace-Beltrimi operator. The problem of conformal flattening then becomes minimizing a certain kind of energy [196]. Specifically, given a mapping between two surfaces $f : \mathcal{M} \to \mathcal{N}$, the determinant of the Jacobian matrix of the mapping characterizes the change of area. More specifically, in the case of 2-manifold surfaces, if $(\mathcal{M}, \phi_\alpha)$ and $(\mathcal{N}, \phi_\beta)$ are parameterized by local coordinates (u, v) and (x, y) respectively[6], then their

[6]The mapping between the two parametrization domains can be represented as $(x(u, v), y(u, v)) := \phi_\beta \circ f \circ \phi_\alpha^{-1}$

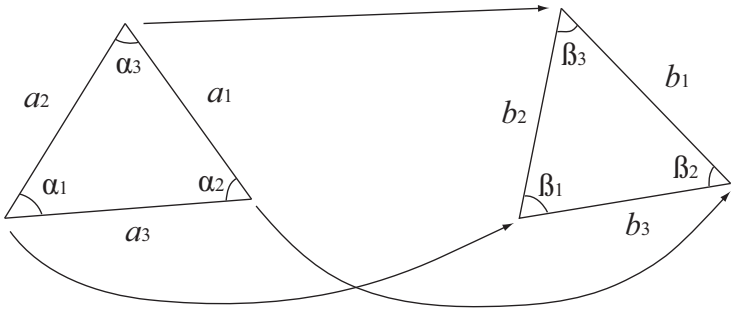


Figure 2.8: In the finite element approach for conformal parametrization, a face is considered as the basic element to represent the continuous concepts such as the Laplace-Beltrami operator. In contrast, the curvature flow based approaches consider all the triangles projected on to one particular vertex.

Jacobian matrix is defined by

$$J_{\alpha\beta} := \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial u} \\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} \end{pmatrix} \tag{2.78}$$

It can be proven that $J_{\alpha\beta}$ is independent of parameterization[7]. The *minimal surface problem* then asks what is the mapping f that minimizes the area change, *i.e.*,

$$\min_f A(f) := \int_{\mathcal{M}} |J_{\alpha\beta}| \tag{2.79}$$

Since the Jacobian at each point ($\sqrt{x_u * y_v - x_v * y_u}$) is nonlinear, directly solving this problem is not straightforward. Alternatively, [196] proposed to solve the following *Dirichlet energy*:

$$E_{\mathcal{D}}(f) := \frac{1}{2} \int_{\mathcal{M}} |\nabla f|_g^2, \tag{2.80}$$

where $|\nabla f|_g^2 = traceg(\partial f, \partial f)$. It can be shown that $\forall f$, $A(f) \leq E_D(f)$, and we have

$$E_C(f) := E_D(f) - A(f) = \frac{1}{2} \int_{\mathcal{M}} |\frac{\partial f}{\partial \bar{z}}|^2. \tag{2.81}$$

Here the energy $E_C(f)$ is called the *conformal energy* of the map f. In the least square conformal maps (LSCMs) discretization [149], a similar conformal energy is considered.

In the discrete setting, by assuming that the surface is piecewise linear [35] (*e.g.*, triangular meshes), it can be shown that the Dirichlet energy can be represented for each triangle as follows.

[7] See Chap 6 for more details.

Lemma 2 *Assuming the mapping between two triangles* $\triangle_1$ *and* $\triangle_2$ *is linear, the* Dirichlet energy *of the mapping is given by*

$$E_{\mathcal{D}}(\triangle_1 \mapsto \triangle_2) = \frac{1}{4}\sum_{i=1}^{3} \cot \alpha_1^i \cdot a_2^i, \tag{2.82}$$

where α_1^i *is and angles of* $\triangle_1$ *and* b_2^i *is the corresponding side lengths in* $\triangle_2$ *(Fig. 2.8).*

Hence, the minimizer of the Dirichlet energy (Eq. 2.80) can be obtained by solving a linear equation in the following form

$$E_{\mathcal{D}}(f) = \frac{1}{2} u^t L_D u. \tag{2.83}$$

Similarly, the area of the mapping can be represented in the discrete setting as

$$A(f) = \frac{1}{2} u^t A u. \tag{2.84}$$

Hence the conformal energy can be expressed in a quadratic form

$$E_C(f) = \frac{1}{2} u^t L_C u, \tag{2.85}$$

where $L_C = L_D - A$ is a $2V \times 2V$ matrix.

In solving such a conformal energy (Eq. 2.85), certain constraints must be imposed to obtain a unique solution. For example, both [196] and [149] require fixing at least two points in the parametrization domain in order to obtain a unique conformal flattening. In [175], by constraining the parametrization u to be *normalized*, *i.e.*, $u^t u = 1$, a spectral method for computing conformal maps can be derived. Nevertheless, these approaches only solve a linear system, hence they search for the conformal mapping in a linear subspace. Hence the resulting conformal mappings often cannot avoid approximation errors. However, by properly discretizing the surface to define the continuous concepts, it can be shown that certain discretization leads to zero approximation errors (*e.g.*, [161, 196]).

Holomorphic 1-form based discretization

Both the above two approaches for conformal flattening requires deforming the original surface to the 2D plane in order to obtain a conformal parametrization of the surface. As an alternative approach – the Holomorphic 1-form based methods [93, 94] – computes a field and its dual on the surface such that it is conformal. It makes use of the differential forms (Sec. 2.1.2) and the celebrated Riemann Roch theorem. Formally, let ω denote the 1-form defined on the surface. According to Riemann surface theory [73], conformal gradient fields $\omega + \sqrt{(-1)} * \omega$ have the following properties:

- *closedness* ω and $*\omega$ are closed, meaning the curlix of ω and $*\omega$ are both zero.
- *harmonity* ω and $*\omega$ are harmonic, meaning that the Laplacian of both ω and $*\omega$ are zero.
- *duality* The cohomology class of ω and $*\omega$ can be determined by the values of their integration along the homology basis e_i.
- *conjugacy* $*\omega$ is orthogonal to ω everywhere.

According to Hodge theory, given $2g$ real numbers $c_1, c_2, \ldots, c_{2g}$, there is a unique real gradient field ω with the first three properties, because each cohomology class has a unique harmonic gradient field ω. These properties for ω can be formulated as the following equations:

$$\begin{cases} d\omega = 0 & \text{closedness} \\ \Delta\omega = 0 & \text{harmonity} \\ \int_{e_i} \omega = c_i, i = 1, 2, ...2g & \text{duality} \end{cases} \tag{2.86}$$

The equations $\int_{e_i} \omega = c_i, i = 1, 2, \ldots, 2g$ restricts the cohomology class of ω. The conjugacy property can be formulated as

$$\star\omega = \overrightarrow{n} \times \omega \tag{2.87}$$

where $\overrightarrow{n}$ is the normal field on the surface, $\times$ is the cross product in R^3. This equation holds everywhere on the surface.

Now we consider the meaning of the above concepts in the discrete setting. The closedness property $d\omega = 0$ means the integration of ω along any simple closed curve (which bounds a topological disk) is zero. Then for each face $[u, v, w]$, the equation for closedness can be approximated by the following linear equation:

$$\omega(\partial[u, v, w]) = \omega[u, v] + \omega[v, w] + \omega[w, u] = 0 \tag{2.88}$$

The harmonity property $\Delta w = 0$ can be formulated using the well known cotangent weighting coefficients (Lemma 2). For any vertex u, the Laplacian of ω on u is zero, hence the equation for harmonity can be formulated as:

$$\Delta\omega(u) = \sum_{[u,v]\in M} k_{u,v}\omega[u, v] = 0 \tag{2.89}$$

$$k_{u,v} = -\frac{1}{2}(\cot\alpha + \cot\beta) \tag{2.90}$$

where α, β are the angles against the edge $[u, v]$ (Fig. 2.8).

The duality property $\int_{e_i} \omega = c_i$ can be implemented simply by summing up all the forms on the edge $e_i = \sum_{j=1}^{n} [u_{j-1}, u_j]$ where $u_0 = u_n$,

$$\int_{e_i} \omega = \sum_{j=1}^{n} \omega[u_{j-1}, u_j] = c_j. \tag{2.91}$$

Once we have computed ω, we can compute $\star\omega$ by using the discrete Hodge star operator, which is defined as follows. Suppose $\{d_0, d_1, d_2\}$ are the oriented edges of a triangle T, their lengths are $\{l_0, l_1, l_2\}$, and the area of T is s, then the discrete wedge product $\wedge$ is defined as

$$\int_T \omega \wedge \tau = \frac{1}{6} \begin{pmatrix} \omega(d_0) & \omega(d_1) & \omega(d_2) \\ \tau(d_0) & \tau(d_1) & \tau(d_2) \\ 1 & 1 & 1 \end{pmatrix} \tag{2.92}$$

The star wedge product $\star\wedge$ of $\wedge$ and τ on smooth surfaces is defined as follows:

$$\int_M \omega^* \wedge \tau = \int_M \omega \wedge^* \tau = \int_M \omega \times^* \tau \cdot \overrightarrow{n}, \tag{2.93}$$

where ${}^*\tau$ is obtained by rotating τ about the normal $\overrightarrow{n}$ on the tangent plane at each point of M. The discrete star wedge product on meshes is defined as

$$\int_T \omega^* \wedge \tau = UMV^T, \tag{2.94}$$

where

$$M = \frac{1}{24s} \begin{pmatrix} -4l_0^2 & l_0^2 + l_1^2 - l_2^2 & l_0^2 + l_2^2 - l_1^2 \\ l_1^2 + l_0^2 - l_2^2 & -4l_1^2 & l_1^2 + l_2^2 - l_0^2 \\ l_2^2 + l_0^2 - l_1^2 & l_2^2 + l_1^2 - l_0^2 & -4l_2^2 \end{pmatrix} \tag{2.95}$$

and

$$U = (\omega(d_0), \omega(d_1), \omega(d_2)) \tag{2.96}$$

$$V = (\tau(d_0), \tau(d_1), \tau(d_2)). \tag{2.97}$$

Therefore, once we know a set of basis of Harmonic forms $\{\omega_1, \omega_2, \omega_3, \ldots, \omega_{2g}\}$, suppose $\star\omega = \sum_{i=1}^{2g} \lambda_i \omega_i$, we can found out λ_i's by solving the following linear system

$$\int_M \omega_i \wedge \star\omega = \int \omega_i \star \wedge\omega, i = 1, 2, \ldots, 2g, \tag{2.98}$$

which reduces to solving a linear equation in the discrete setting,

$$W\Lambda = B. \tag{2.99}$$

Here W has entries $w_{ij} = \sum_{T\in M}\int_T \omega_i \wedge \omega_j$, Λ has entries λ_i, and B has entries $b_i = \sum_{T\in M}\int_T \omega_i \star \wedge\omega$. See [93, 94] for more details on the Holomorphic 1-form based approach for conformal flattening. In graphics, Holomorphic 1-form has been applied for image editing [142] and geometry processing [285].

2.2.3 Quasi-conformal mapping

Conformal Geometry assumes the surfaces be endowed with conformal structures. Although in many cases this is approximately true (e.g. surfaces undergoing isometric deformation), in general this is not necessarily the case. One reason that conformal Geometry has been widely applied in many applications is its good approximation to many real world problems [2, 41] [8]. Despite of this, in situations that accuracies are important, such as registering the details of the surface, it is necessary to explore beyond the scope of conformal mapping.

In mathematics, the prefix "quasi" is used for studying a concept that is out of its original meaning. Intuitively, the field of **quasi-conformal** mapping provides the quantitative measure for the deviation of the geometry from the conformal structure [5]. Historically, quasi-conformal mapping arises from the question if one can find a mapping from a square to a rectangle that is "closest" to conformal mapping. In the context of surface registration, since the deformations between surfaces are usually not perfectly conformal, the language of quasi-conformal mapping allows us to describe the geometry with more accuracy. In this section, we briefly introduce the ideas of quasi-conformal mapping.

Intuitively, quasi-conformal mapping is less rigid than conformal mapping (conformal mapping is less rigid than isometric mapping). Let $w = f(z), w, z \in \mathbb{C}$ be a homeomorphism from one region to another in the complex domain $\mathbb{C}$. At any point z_0, it induces a linear mapping:

$$dw = \frac{\partial f}{\partial z}dz + \frac{\partial f}{\partial \bar{z}}d\bar{z}. \tag{2.100}$$

As we have shown, if the mapping is conformal (holomorphic), we have $\frac{\partial f}{\partial \bar{z}} = 0$. Otherwise, the following ratio

$$D_f = \frac{|\frac{\partial f}{\partial z}| + |\frac{\partial f}{\partial \bar{z}}|}{|\frac{\partial f}{\partial z}| - |\frac{\partial f}{\partial \bar{z}}|} \tag{2.101}$$

[8] According to a personal eyewitness, conformal mapping has be applied during world war II for the design of airplanes by the mathematician Hermann Kober [2].

can be used for measuring the extent to which the mapping is deviated from conformal (here we assume the mapping is orientation preserving so that $|\frac{\partial f}{\partial z}| - |\frac{\partial f}{\partial \bar{z}}| > 0$). Furthermore, if we define

$$d_f = \frac{|\frac{\partial f}{\partial \bar{z}}|}{|\frac{\partial f}{\partial z}|} < 1, \tag{2.102}$$

then $D_f = (1 + d_f)/(1 - d_f)$. Here d_f is called the **Beltrami coefficient**.

Hence, we can define quasi-conformal mapping according to the value of the Beltrami coefficients.

Definition 13 ***(Quasi-conformal mapping)*** *A mapping $f : \mathbb{C} \to \mathbb{C}$ is said to be quasiconformal if D_f is bounded. It is K-quasiconformal if $D_f \leq K$*

In practice, if we know the mapping f between two surfaces, the Beltrami coefficient can be calculated by the Jacobian of the mapping at each point. That is, the relation between the Jacobian and the function f is as follows:

$$|f_z|^2 - |f_{\bar{z}}|^2 = J. \tag{2.103}$$

If we do not constrain the mapping between two surfaces except its continuity, there can be infinitely many mappings and any mapping is a quasi-conformal mapping. Hence, one important issue with the application of quasi-conformal mapping theory is that it is somehow overly flexible – the Beltrami coefficient is defined for every point on the surface. Given two surfaces, one can define the distance between them as the *closest* quasi-conformal mapping that maps one to the other. One can define different quantitative measures for the meaning of closeness. For example, one can define

$$dist(\mathcal{M}_1, \mathcal{M}_2) := \inf_{f:\mathcal{M}_1 \to \mathcal{M}_2} \sup_{p \in \mathcal{M}_1} D_f(p), \tag{2.104}$$

as the quasi-conformality between the two shapes. However, finding the proper optimization method to compute the closest quasi-conformal mapping between two surfaces remains a challenging problem.

2.3 Conclusion

In this chapter, we introduced the foundation of geometric methods for surface registration. The idea of manifold is to study the geometry in its local charts, regardless of the extrinsic representation of the surface. By properly defining the transition maps between different charts,

one can study the geometry under different invariance groups. On the local charts, one can also define geometric features such as vector fields, tensors and differential forms, *etc*. Under the umbrella of Riemannian Geometry, where a metric tensor is defined for each point on the surface, one can define useful features of the surface such as curvatures and geodesics. Curvatures and geodesics are the fundamental tools in modern physics. In the application of surface registration, Riemannian Geometry has been successfully applied under isometry assumption, achieving meaningful metrics between two surfaces.

In contrast, Conformal Geometry disregards part of the information in Riemannian metric (*i.e.* the scale information) and is thus less "rigid" than Riemannian Geometry. This flexibility of Conformal Geometry sees its advantage in many applications. For example, it is found ([248]) that the transformation among many species in the world are actually very close to conformal mapping (e.g., the profiles of fishes, the skulls of human). In the problem of surface registration, a very useful consequence of applying Conformal Geometry is that it allows us to establish the dense correspondences between two shapes in a closed-form. We shall see in this dissertation that such a theoretical guarantee can be implemented in different scenarios of the surface registration problem.

If we completely disregard the metrics endowed to each surface, then the geometry can be classified by their topology. In topology, a good abstraction of surfaces is the complexes, which represents the surface by discrete points and their combinations. The introduction of complexes allows us to connect continuous geometry to the discrete graphs. Especially, the chain complex is a good abstraction of the continuous surface. However, a graph can be much more complicated than the chain complex, indicating that not all the discrete structures have their continuous counterpart (however the reverse might be true). Another way that indicates the complexity of graph is by comparing the tensor functions versus the potential functions defined for graphs. Almost every quantity studied in geometry is studied using linear tensor, whereas the potential function in a graphical model is not necessarily linear. Hence, in the next chapter, we shall systematically study the various discrete structures.

Chapter 3

Groundwork – Graph-based Methods

> “ ..., (continued) hence everything can be explained by a unified view. ”
>
> – Anonymous

In the discrete setting, we first introduce some basic concepts of graphical models in Section 3.1. Then in Section 3.2 we discuss the techniques for Maximum a Posterior (MAP) inferences on Markov Random Fields (MRFs).

We show that *convextiy* plays an important role in identifying tractable structures. In Sec. 3.1.1, we introduce the concept of submodularity, which is the discrete counterpart of traditional concept of convexity. The study of optimization on submodular functions are drawing increasing attention [66, 74]. In Sec. 3.2.1, we introduce the theoretical background of convex geometry. Finally, we derive a new unified framework for graphical models with convex priors in Sec. 3.2.2.

The *indicator function* is a very useful tool in studying combinatorial structures.In Sec. 3.1.1 we show that the conversion from a probability set function to a probability density function can be done by defining proper indicator functions. In Sec. 3.2.1 we show that indicator function can be used for establishing the equivalence between submodular functions and convex functions. In Sec. 3.2.2, the indicator function (more specifically, the upper level sets function) plays a key role in deriving a unified representation of the convex Markov Random Fields.

3.1 Graphical models

Among the geometry theories discussed in the previous chapter, what are of interest are the structures/representations of the geometry (*e.g.*, vector fields, geodesics, complexes) and the functions/measures defined on them (*e.g.*, metrics, curvatures, forms) – the interactions between the structures and the their measures give us tons of results that reveal the interesting properties of the geometric object. In contrast, **probability** defines a measure on the multiple possible configurations of the same structure. At the first glance, there seems to be no obvious connection between the two fields. However, careful examinations would reveal some common characteristics that they share, *e.g.*, both fields involve the interactions between structure and functions. In the following, we shall give the clue that *the structures studied by probability is somehow much more complicated than the structures of geometry*. This is established by showing that the simplest structure in probability – the independent system – can fully characterize the simplicial complex structure of geometry. Another way to see this is that in geometry, almost all the quantitative measures (metric tensor, curvature, Lagrangian, *etc*) only involve linear functions, whereas in the discrete setting there are far more non-linear functions. Furthermore, the complexity of the structures in probability is characterized by the notion of *conditional independence*. The field of **graphical models** [124] focuses on studying the conditional independence by the connective structures among the variables of the probability function. ◁

3.1.1 Probability, independence system and submodularity

Mathematically, the configurations of a random experiment are defined as a nonempty collection $\mathfrak{A}$ of subsets of Ω such that $\mathfrak{A}$ is closed under countable *union*, *intersection* as well as under *complementation*. A *measurable space* is a pair $(\Omega, \mathfrak{A})$, where Ω is a space and $\mathfrak{A}$ is a σ-field. A *probability set function* is defined on a measurable space as follows

Definition 14 ***(Probability set function)*** *If* $\Pr(X)$ *is defined for a measurable space* $(\Omega, \mathfrak{A})$*, and if*

(i) $\Pr(X) \geq 0$*,*

(ii) $\Pr(X_1 \cup X_2 \ldots) = \Pr(X_1) + \Pr(X_2) + \ldots$*, where* $X_i \cap X_j = \emptyset, i \neq j$*,*

(iii) $\Pr(\Omega) = 1$

then $\Pr(\cdot)$ *is called the probability set function of the outcome of the random experiment.*

Given this definition, we can further define the *conditional probability* that event X occurs given that event Y occurs to be

$$\Pr(X|Y) := \frac{\Pr(X \cap Y)}{\Pr(Y)}. \tag{3.1}$$

Another important property of the probability function is the *law of total probability*, which states that if $X_1, X_2, \ldots, X_n$ are mutually disjoint and $\bigcup_{i=1}^{n} X_i = \Omega$, then

$$\Pr(Y) = \sum_{i=1}^{n} \Pr(Y \cap X_i) = \sum_{i=1}^{n} \Pr(Y|X_i) \Pr(X_i) \tag{3.2}$$

The *sum rule* (Eq. 3.2) and the *product rule* (Eq. 3.1) are the two basic operations on probability set functions, in contrast to the differentiation operator for continuous functions.

The notion of *independence* plays an important role in characterizing the probability set function. If $X = X_1 \cap \ldots \cap X_n$, then the event $\{X_1, \ldots, X_n\}$ are *independent* if $\Pr(X) = \prod_{i=1}^{n} \Pr(X_i)$. Hence, independence allows us to *factorize* the probability function into simpler ones. Obviously, if X is independent, any of its subsets are also independent. This closeness property allows us to define an **independence system** as a collection of unordered subsets of $\mathcal{I}$ that is *independent*, given a finite collection of objects $\mathcal{I} = \{x_\alpha\}$. Any such system must be closed with respect to restriction (any subset of an independent set is also independent) and thus defines a simplicial complex: the **independent complex** is the abstract simplicial complex (recall Sec. 2.1.4) on vertex set $\mathbf{I}$ whose k-simplices are collections of $k+1$ independent objects[1]. ◁

A number of interesting questions arise concerning characterizing an independence system, for example,

(i) Given the probability function defined for a measurable space $(\Omega, \mathfrak{A})$, can we efficiently find the maximal independence system inside $\mathfrak{A}$?

(ii) Is there an efficient algorithm to decide if a probability function is independent?

(iii) Is there an efficient way to approximate a probability function with an independent system.

A generalization of the independence system is the *matroid*, which was first introduced by H. Whitney in 1935 [281] as an abstraction of the linear dependence structure of a set of vectors. A matroid is formally defined as below:

[1] See [110] for the study of simplicial complex and graphs with independence structure, whose primary focus is their topological properties.

Definition 15 *A measurable space $(E, \mathcal{I})$ is called **matroid**, if the following holds*

(i) $\emptyset \in \mathcal{I}$

(ii) (Closeness/hereditary property) $I_1 \subset I_2 \Rightarrow I_1 \in \mathcal{I}$

(iii) (Exchange/augmentation property) $I_1, I_2 \in \mathcal{I}$, $|I_1| < |I_2| \Rightarrow \exists e \in I_2 - I_1, I_1 \cup \{e\} \in \mathcal{I}$.

Each $I \in \mathcal{I}$ is called an independent set of matroid $(E, \mathcal{I})$.

Note that the first two properties guarantee that the matroid is also an independent system. An immediate example of a matroid is the trees in a graph $\mathcal{G}$. The structure of matroid is often seen in algorithms or theoretical computer science [189] and it is closely related to a class of efficiently solvable combinatorial optimization problems. An independent set which is maximal in $\mathcal{I}$ with respect to set inclusion is called a *base* of the matroid, denoted as $\mathcal{B}$.

Another way to characterize a matroid is by defining a function on the set system $(E, \mathcal{I})$, known as the *rank function* which is defined as:

Definition 16 *A **rank function** $r : A \to \mathbb{R}$ satisfies the following*

(i) for each subset A of S, $0 \leq r(A) \leq |A|$;

(ii) if $A \subset B$, then $r(A) \leq r(B)$

(iii) for any A and B, $r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$.

It is easy to see that if we define $r(A) = |A|$, the first property guarantees non-negativeness and monotone nondecreasing properties of the rank function and $r(\emptyset) = 0$. The second property guarantees that taking a superset does not decrease the rank of a set. The third property of the rank function is also called the *submodularity* property:

Definition 17 *Consider a set E and a real function $f : 2^E \to \mathbb{R}$. f is **submodular**[2] if for any two subsets $X, Y \subset E$,*

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \tag{3.3}$$

Equivalently, submodularity can be characterized by the *marginal value* defined as

$$\Delta_Y f(X) = f(X \cup Y) - f(X). \tag{3.4}$$

And f is submodular if and only if for any $X \subset E$ and $a, b \in E \setminus X$, $\Delta_a f(X) \geq \Delta_a f(X \cup \{b\})$. In the following, we shall show that the submodular function can be regarded as the

discrete analogy to the convex functions . Also submodularity is important in combinatorial optimization, the following lemma shows one of its properties: ◁

Lemma 3 *Let $f : 2^E \to \mathbb{R}$ be an arbitrary submodular function on a subset $\mathcal{I} \subset 2^E$. The set of all minimizers of f given by*

$$\mathcal{I}_0 := \{X | X \in \mathcal{I}, f(X) = \min\{f(Y) | Y \in \mathcal{I}\}\} \tag{3.5}$$

satisfies $\forall X, Y \in \mathcal{I}_0$, we have $X \cup Y, X \cap Y \in \mathcal{I}_0$.

In many real-world problems, an event is often endowed with some quantitative values (*e.g.*, temperature, grade, pixel color). In such situations, probability can also be defined using **Random Variables**. If an event consists of n random variable $\{X_1, X_2, \ldots, X_n | X_i \in I_i\}$, then the probability of such an event can be denoted as $P(X_1, \ldots, X_n)$. In order for the properties of the probability set function to hold, the following constraints must be satisfied:

$$\text{Sum Rule: } \sum_{x \in I_x} P(X = x) = 1 \tag{3.6}$$

$$\text{Marginal Rule: } P(x) = \sum_{y \in I_y} P(x, y), \forall x \in I_x. \tag{3.7}$$

Using the conditional probability rules, a probability with multiple random variables can be factorized as

$$P(X_1, \ldots, X_n) = P(X_1)P(X_2|X_1) \ldots P(X_n|X_1, \ldots, X_{n-1}). \tag{3.8}$$

Here we can see that the probability function can describe the interaction/dependence among multiple variables, which introduces a rich structure to be explored.

To summarize, in this subsection, we established the connection among independence system, matroid and submodularity. We have shown that an independence system is the simplest structure for studying the probability function, and submodularity exhibits a more general but tractable structure. The elements involved in an independent system present no interaction among them. In the following, we check a generalized form of the independence, namely the *conditional independence*.

3.1.2 Conditional independence, Markov random fields and Gibbs distribution

If we relax the independence assumption, by simply requiring the most succinct factorization of a probability function, then such factorization can be studied under the umbrella of

[2]The notion of convexity or submodularity is different from the linear system in the sense that it is used in characterizing minimizers.

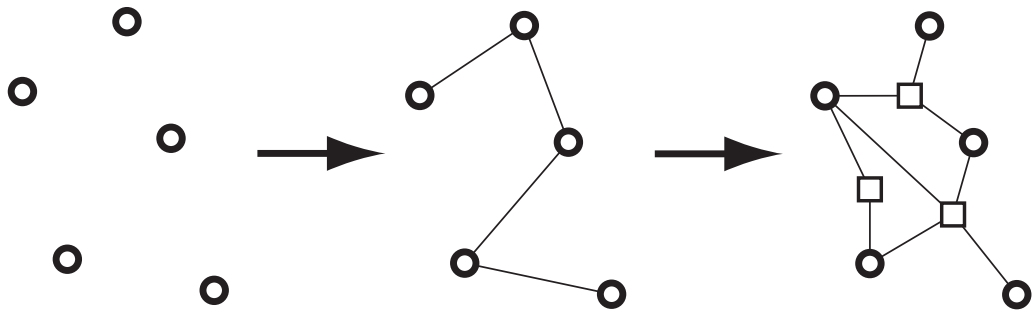


Figure 3.1: The independence among a graph increases.

graphical models. In this subsection, we introduce the theoretical background of graphical models.

Definition 18 ***(Conditional independence)*** *X is conditionally independent of Y given Z, written $X \perp\!\!\!\perp Y|Z$, if*

$$P(x, y|z) = P(x|z)P(y|z). \tag{3.9}$$

Conditional independence allows us to represent the coupling among different random variables. Furthermore, if we denote each variable as a node in a graph, the dependency among nodes can be succinctly represented by the (hyper-)edges among them (Fig. 3.1. Formally, a **graphical model** is a probabilistic model for which a graph denotes the conditional independence structure between random variables [1]. There are two types of graphical models: one is defined on directed acyclic graph, known as *Bayesian network*, and another is undirected, known as *Markov Random Field* (MRF). In the following, we will focus on the theories of the undirectly graphical model due to its close relevance to our surface registration approach[3].

Markov random fields

Before defining the MRF, we first introduce the concept of *Markov property*. Intuitively, it states that the value of a variable is only related to the values of its neighboring variables.

Definition 19 ***(Markov property)***. *For any variable i, the distribution X_i is conditionally independent of the rest of the graph given just the variables that lie in the neighborhood of the*

[3]A directed graphical model can be equivalently converted to an undirected one using the moralization method [124].

vertex, i.e.,

$$\Pr(X_i|X_{V\setminus i}) = \Pr(X_i|X_{\mathcal{N}_i}). \tag{3.10}$$

More generally, for any disjoint subsets X, Y, Z of V such that Z separates X from Y in the graph, it holds that

$$X \perp\!\!\!\perp Y|Z \tag{3.11}$$

It can be seen that this independence can be computed from the graph. Formally, we define MRF as follows:

Definition 20 ***(Markov random field)*** *The collection of random variables $X_1, \ldots, X_n$ is a Markov random field with respect to a neighborhood system on $\mathcal{G}$ if and only if the following two conditions are satisfied.*

(i) $\Pr(x) > 0$*, for all* $x \in X_1 \times X_2 \ldots, \times X_n$

(ii) The Markov property as defined in Def. 19 is satisfied for every node.

Note that condition (i) actually rules out those MRF-like energies with hard constraints (*e.g.*, [267, 292]). Although it is possible to use soft constraints to penalize unlikely configurations, since the energy constructed is usually non-submodular, optimal configurations are often NP-hard to infer.

Next, we relate the probabilistic density function for an MRF to the more general energy functions. In solving real-world problems, it is more natural to define an energy without considering its probability meaning, although many energies can be interpreted as defining a distribution (*e.g*, the least square energy can often be equivalently regarded as defining a Gaussian distribution [27]). A nice property of MRF is that it actually covers a wide variety of energy functions.

Definition 21 ***(Gibbs distribution)*** *Given a graph $\mathcal{G}$ and a neighborhood system $\mathcal{N}$ defined on it, a probability function defined on the set of configurations Ω is called Gibbs distribution if it has the form*

$$\Pr(x_1, \ldots, x_n) = \frac{1}{Z} \exp(-\frac{E(x)}{T}), \tag{3.12}$$

where

(i) The partition function Z *is defined as*

$$Z = \sum_{x \in \Omega} \exp(-\frac{U(x)}{T}). \tag{3.13}$$

(ii) T *is a constant which is known as the "temperature" in statistical mechanics.*

(iii) $E(x)$ *is the "energy" function of the form*

$$U(x) = \sum_{c \in \mathcal{C}} V_c(x), \tag{3.14}$$

where $\mathcal{C}$ *is the set of cliques in* $\mathcal{G}$*.*

The following theorem establishes the connection between the Gibbs distribution and the MRF functions [65, 98].

Theorem 8 ***(Hammersley-Clifford).*** *X is Markov random field with respect to a neighborhood system $\mathcal{N}_{\mathcal{G}}$ on the graph $\mathcal{G}$ if and only if it is a Gibbs random field with respect to the same neighborhood system.*

The H-C theorem is powerful in that it tells us that we can define a wide variety of energy functions of form 3.14 and they can all be equivalently interpreted as solving the undirectly graphical model (MRF) problem.

Conditional independence and graph structure

Finally, to complete the introduction of MRF, we address the problem of graph structure and probabilistic density functions. It turns out that not all distribution functions can be represented as a graph and vice versa. The following definitions characterize two different classes of graphs.

Definition 22 ***($\mathcal{D}$-map)*** *A graph (directed or undirected) is said to be a $\mathcal{D}$-map ('dependencies map') for a distribution if every conditional independence statement of the form $X \perp\!\!\!\perp Y|Z$ for sets of variables X, Y, and Z that is satisfied by the distribution is reflected in the graph. Thus, a completely disconnected graph having no edges is trivially a $\mathcal{D}$-map for any distribution.*

Definition 23 ***($\mathcal{I}$-map)*** *A graph (directed or undirected) is said to be a $\mathcal{I}$-map ('independencies map') for a distribution if every conditional independence statement of the form $X \perp\!\!\!\perp Y|Z$ for sets of variables$X$, Y, and Z that is expressed by the graph is also satisfied by the distribution. Thus, a completely connected graph is trivially a $\mathcal{I}$-map for any distribution.*

A graph that is both $\mathcal{D}$-map and $\mathcal{I}$-map for a distribution is called a $\mathcal{P}$-map ('perfect map'). Not all the distributions have $\mathcal{P}$-maps.

3.1.3 Information theory and exponential families

In the seminal work of Shannon [216], the **entropy** is defined as as a lower bound on the number of bits needed to transmit the state of a random variable. Quantitatively, this can be defined by [57]

$$H[p] := -\sum_{x \in I_x} p(x) \log p(x) = E_p(\log \frac{1}{p(X)}) \tag{3.15}$$

for a random variables X with a probability mass function $p(\cdot)$. Entropy allows us to measure not only the uncertainty of a random variable, but also the dependency among different variables. To this end, the **mutual information** between two random variables X and Y is defined by

$$I(X;Y) := \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3.16}$$

Furthermore, we have the following properties for mutual information:

$$I(X;Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y) \tag{3.17}$$

$$I(X;X) = H(X) \tag{3.18}$$

Hence, if two variables x and y are independent, $I(X;Y) = 0$.

Mutual information is also useful in measuring the loss of information among different representations. If a family of probability function $\{f_\theta(x)\}$ is indexed by parameters θ, then a function $T(X)$ is said to be a **sufficient statistic** relative to the family $\{f_\theta(x)\}$ if X is independent of θ given $T(X)$, or equivalently

$$I(\theta; X) = I(\theta; T(X)) \tag{3.19}$$

Furthermore, if $T(X)$ is sufficient statistic of X, the **exponential family** of distributions is defined as

$$p(X) = h(X) e^{\theta^T T(X) - A(\theta)}. \tag{3.20}$$

Here $A(\theta)$ is the *log partition function* or *cumulant function*. Most familiar statistical families turn out to belong to the exponential family [101]. The log partition function plays an important role in characterizing the statistics of X. For example, the mean and variance of X can be computed from $A(\theta)$ by

$$\mu := E_\theta(T(X)) = \{\frac{\partial A(\theta)}{\partial \theta}\} \tag{3.21}$$

$$\sigma := \text{Var}_\theta(T(X)) = \{\frac{\partial^2 A(\theta)}{\partial \theta^2}\}. \tag{3.22}$$

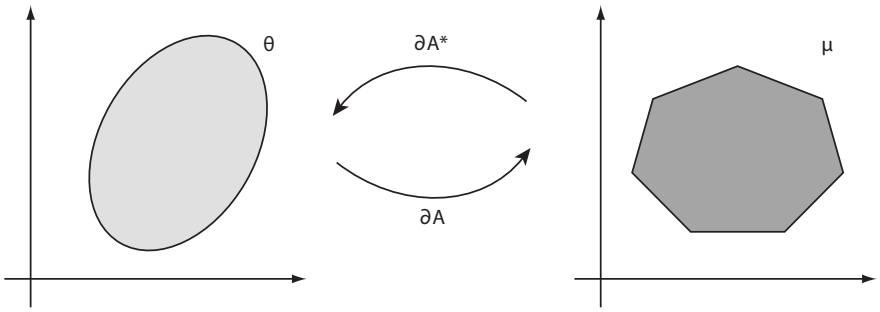


Figure 3.2: The duality between the parameter space θ and the mean parameter space μ.

In fact, Equ. 3.21 gives us a mapping between the parameter space θ and the mean value space μ. Moreover, $A(\theta)$ is always a convex function. Hence by defining the conjugate dual function (more detailed discussions on conjugate dual are given in the following section)

$$A^*(\mu) := \sup_{\theta}\{\langle \mu, \theta \rangle - A(\theta)\} \tag{3.23}$$

a duality between the parameter space θ and the mean value space μ can be established (Fig. 3.2). Intuitively, such duality allows us to study the exponential family in its mean parameter space, which often significantly simply the problem.

Finally, we discuss on the connection between the entropy and the exponential families. In fact, the exponential family can be regarded as a family of distributions that achieve the maximal entropy given a predefined mean values:

$$\max_{p} \; H(p) \tag{3.24}$$

$$\text{s.t.} \quad E_p(T(X)) = \mu \tag{3.25}$$

It can be proven that the class of distributions that achieve the above maxima have the form [162]:

$$p(x) \propto \exp\{\theta^T T(X)\} = \exp\{\langle \theta, T(X) \rangle\}, \tag{3.26}$$

On the other side, for each mean parameter μ, if we denote by $\theta(\mu)$ the parameter that satisfies

$$E_{p_{\theta(\mu)}}(T(X)) = \mu, \tag{3.27}$$

then we have the following relationship between entropy and the dual function $A^*(\mu)$ defined in Equ. 3.23:

$$A^*(\mu) = -H(p_{\theta(\mu)}). \tag{3.28}$$

Such elegant relationships among the entropy, parameter space, mean parameter space, established by conjugate duality, lay the foundation of graphical models.

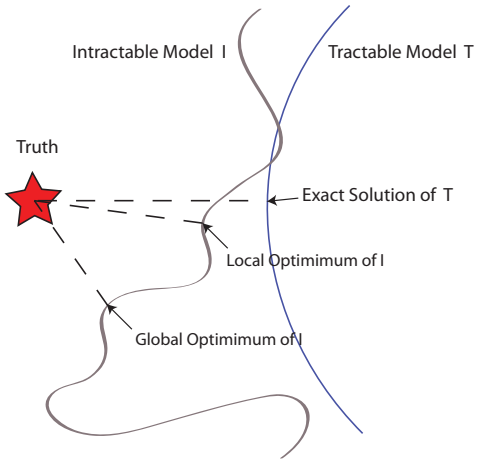


Figure 3.3: An illustration of the tractable v.s. intractable models. Although tractable models gives us the optimal solution, their model powers are usually less than those intractable models. Hence one can hope that even the suboptimal solutions of the intractable models can allow us better approach the truth than those tractable models. However, there is also a risk that the suboptimal solutions are not as good as the optimal solutions of the tractable models.

3.2 Inference on Markov Random Fields

So far, we have introduced the basic concepts of graphical models. In applying graphical models to real-world problems, two types of inference are most commonly met: computation of marginal and maximum a posterior (MAP)[4]. The computation of marginal distribution requires the computation of the log partition function (Eq. 3.20), which is often intractable in the general case ($\#P$-complete). The computation of MAP is also an NP-hard problem, and even the approximation becomes very difficult when the size of the cliques becomes large. In fact, exact inference algorithms for arbitrary graphs have already been well studied, such as the junction-tree algorithm [144, 145]. Its complexity is exponential to the size of the maximum clique. In the case that the graph is a tree, both MAP and marginal can be computed in polynomial time.

Although exact inference is intractable, since graphical models is able to model a wide variety of real-world problems, it is often highly desired to do inference on graphs with loops. Also, intractability corresponds to the worst-case input, many problems from real-world applications can be tackled very effectively[124]. Thus, approximation algorithms are also widely studied in inference. An illustration for the necessity of investigating approximate algorithm

[4]There are also other tasks such as M-best solutions [79]

is shown in Fig. 3.3.

There are many approximate inference algorithms, such as belief propagations [191, 192], or graph cuts based methods [33, 135]. It has been shown that both approaches can be connected to each other [125, 242]. In particular, linear programming plays an important role in designing efficient algorithms [125, 262, 279]. Currently, almost all the efficient algorithms are based on the idea of **relaxation**. With different relaxations, the original problems are often solved in the dual space, which can effectively avoid local optimum in the original space (but not the dual space). It has also been shown that when the gap between the relaxed solution and the original problem can be decreased to be arbitrary small if we increase the number of relaxations [132, 226–228, 263, 278]. However, the relaxations would become exponentially large as the gap approaches to zero.

In this section, we first discuss the theoretical background for MRF optimization techniques, namely the convex geometry [23, 31, 59, 231], and in the end we provide a novel unified formulation for MRF problems with convex energy.

3.2.1 Convex geometry and optimization

Convex function plays an important role in optimization since *any locally optimal solution implies globally optimal solution*. A set $\mathcal{C}$ is said to be *convex* if for all $x, y \in \mathcal{C}$ and $0 \leq \mu \leq 1$, $\mu x + (1-\mu)y \in \mathcal{C}$. From its original definition, convexity is a continuous concept due to the continuous variable μ involved in the definition. Similarly, a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to be **convex** if

$$\mu f(x) + (1-\mu) f(y) \geq f(\mu x + (1-\mu)y) \tag{3.29}$$

for all $x, y \in \mathbb{R}^n$ and $0 \leq \mu \leq 1$.

A convex function can be converted to a convex set by using the *epigraph*. The **epigraph** of a function $f : X \to [-\infty, +\infty]$, where $X \subset \mathbb{R}^n$ is a subset of $\mathbb{R}^{n+1}$:

$$\text{epi}(f) := \{(x, w) | x \in X, w \in \mathbb{R}, f(x) \leq w\}. \tag{3.30}$$

It can be shown that if the epigraph of a function f is convex, then f is convex.

There are many ways to recognize a convex function such as the use of the first and second derivatives of the function. The following proposition is useful in our later derivation of a unified convex MRF formulation.

Proposition 2 *Let $f_i : \mathbb{R}^n \to (-\infty, +\infty]$, be given convex functions for $i \in I$, where I is an*

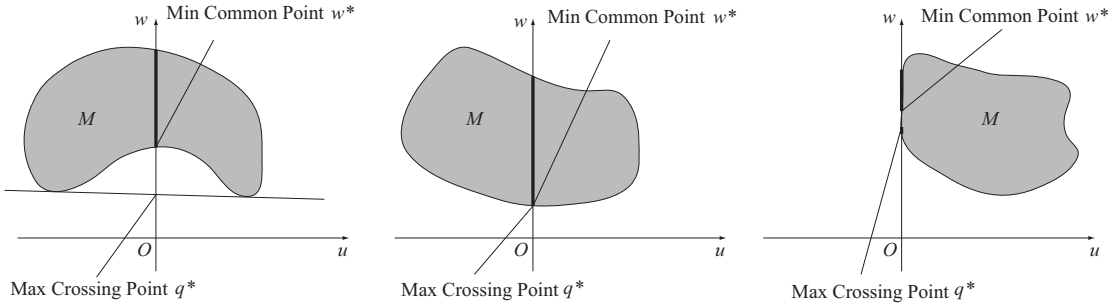


Figure 3.4: The min common/max crossing duality represents the most basic idea of duality in convex geometry. The above shows three different situations for the relationship between min common and max crossing points for a set M. When the min common and the max crossing points meet (middle), the two problems become equivalent to each other. Otherwise there is a gap between them and $q^* < w^*$.

index set, then the following function is also convex

$$g(x) = \sup_{i \in I} f_i(x), \tag{3.31}$$

where $g(\cdot)$ is defined on $\mathbb{R}^n \to (-\infty, +\infty]$.

Dualities in convex geometry

One of the primary goals of optimization is to locate the minimum of a function $f(\cdot)$ in some domain $\mathcal{C}$:

$$\min_{x \in \mathcal{C}} f(x) \tag{3.32}$$

This problem can be intractable if either the domain $\mathcal{C}$ or the function f is intractable, or both. Hence in practice, most work on the optimization problem are constrained in the tractable cases. One of the most general tractable cases is *convexity*. In particular, the idea of duality plays an important role and can be represented in multiple ways. In the following, we briefly list the various dualities studied in convex geometry.

(i) Min common/max crossing duality

Similar to the duality in classic geometry, the duality in convex optimization also involves interactions between two counterparts. Such interactions are often represented by the act of hyperplanes on the boundary of the convex set. By using hyperplanes, one can define one-to-one correspondences between the primal and the dual problems. One reason for exploring the

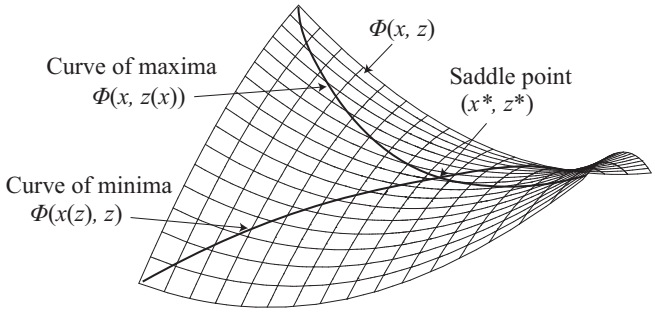


Figure 3.5: Saddle point.

duality is that the dual problem is usually easier to solve than the primal one. However, for non-convex problem, the dual problem is not necessarily equivalent to the primal. Hence one is interested in situations when the primal and dual solutions meet, or how far the dual problem is from the primal one (approximation solution). One of the most basic forms of duality is the idea of min common/max crossing, which can be illustrated in Fig. 3.4. Given a set M, the *min common* problem is

$$\min \quad w \tag{3.33}$$

$$\text{s.t.} \quad (0, w) \in M \tag{3.34}$$

and the *max crossing* problem is

$$\max_{\mu} \quad \{\inf_{(u,w)\in M} \{w + \mu' u\}\} \tag{3.35}$$

$$\text{s.t.} \quad \mu \in \mathbb{R}^n \tag{3.36}$$

Let w^* and q^* denote the optimal solution of the min-common and the max crossing problems, respectively, the following theorem establishes their relationship.

Proposition 3 ***(Weak Duality Theorem)*** *For the min common and max crossing problems we have*

$$q^* \leq w^* \tag{3.37}$$

When $q^* = w^*$, solving the max crossing is equivalent to solving the min common problem.

(ii) Saddle point and minimax theory

A direct application of the min-common/max-crossing duality is in the proof of minimax theory. Given a function $\phi : X \times Z \mapsto \mathbb{R}$, where X and Z are nonempty subsets of $\mathbb{R}^n$ and

$\mathbb{R}^m$ respectively, we consider the following two problems:

$$\sup_{z\in Z}\inf_{x\in X}\phi(x,z) \tag{3.38}$$

and

$$\inf_{x\in X}\sup_{z\in Z}\phi(x,z). \tag{3.39}$$

It is easy to see that

$$\sup_{z\in Z}\inf_{x\in X}\phi(x,z)\leq\inf_{x\in X}\sup_{z\in Z}\phi(x,z) \tag{3.40}$$

The minimax problem asks under what conditions the following equality holds:

$$\sup_{z\in Z}\inf_{x\in X}\phi(x,z)=\inf_{x\in X}\sup_{z\in Z}\phi(x,z) \tag{3.41}$$

A pair of vectors $(x^*,z^*)\in X\times Z$ is called a saddle point of ϕ if

$$\phi(x^*,z)\leq\phi(x^*,z^*)\leq\phi(x,z^*),\forall x\in X,\forall z\in Z \tag{3.42}$$

If saddle point exists, the minimax equality holds. Some sufficient conditions can be proven using the min common/max crossing duality [23]. The minimax theory is useful in applications such as Lagrangian duality, game theory [261], *etc*.

(iii) Polar cones and polyhedron

In discrete optimization, the discrete solution can often be represented as a set of finite number of points in $\mathbb{R}^n$, namely $\{a_1,\ldots,a_r\}$. In order to search for the optimal solution among such a set, it is often convenient to convert the discrete set to be continuous. One way to represent a discrete set by a continuous set is through the concept of **convex hull**. In another way, we can define the *cone* of a set $\{a_1,\ldots,a_r\}$ to be

$$\text{cone}(\{a_1,\ldots,a_r\})=\{x|x=\sum_{j=1}^{r}\mu_j a_j,\mu_j\geq 0,j=1,\ldots,r\} \tag{3.43}$$

To study the duality of a cone, we define the **polar cone** for a nonempty set C as

$$C^*:=\{y|y'x\leq 0,\forall x\in C\} \tag{3.44}$$

Theorem 9 ***(Polar cone theorem)*** *For any nonempty cone C, $(C^*)^*$ = cl(conv(C)). Here $cl(\cdot)$ denotes the closure of a set.*

Using the polar cone, one can establish the duality between a set of points and a set of half planes. A subset P of $\mathbb{R}^n$ is a **polyhedral set** (or polyhedron) if it is nonempty and has the form

$$P = \{x | a_i' x \leq b_i, i = 1, \ldots, r\} \tag{3.45}$$

where a_i are vectors in $\mathbb{R}^n$ and $b_i \in \mathbb{R}$. To see the benefits of exploring cone geometry and its duality, the following proposition reveals an interesting relationship

Proposition 4 ***(Minkowski-Weyl Representation)*** *A set P is polyhedral if and only if there is a nonempty finite set $\{v_1, \ldots, v_m\}$ and a finitely generated cone C such that $P = conv(\{v_1, \ldots, v_m\}) + C$,* i.e.*,*

$$P = \{x | x = \sum_{i=1}^{m} \mu_i v_i + y, \sum_{i=1}^{m} \mu_i = 1, \mu_i \geq 0, i = 1, \ldots, m, y \in C\}. \tag{3.46}$$

Also one is interested in the conditions for continuous solution to coincide with integer solutions:

Theorem 10 ***(Unimodularity and integer solution)*** *Let P be a polyhedral set of the form*

$$P = \{x | Ax = b, c \leq x \leq d\}, \tag{3.47}$$

where A is an $m \times n$ matrix, b is a vector in $\mathbb{R}^m$, and c and d are vectors in $\mathbb{R}^n$. Assume that all the components of the matrix A and the vectors b, c, and d are integer, and that the matrix A is totally unimodular. Then all the extreme points of P have integer components.

Finally, we give a condition for a function f to be a polyhedral set.

Proposition 5 *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a convex function. Then f is polyhedral if and only if dom(f) is a polyhedral set and*

$$f(x) = \max_{i=1,\ldots,m} \{a_i' x + b_i\}, \forall x \in \textit{dom}(f). \tag{3.48}$$

Here dom$(\cdot)$ is defined as

$$\text{dom}(f) := \{x | f(x) < \infty\}. \tag{3.49}$$

(iv) Lagrange duality

The concept of polar cones can be applied to the optimization of a function f over a convex subset X of $\mathbb{R}^n$.

Proposition 6 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. A vector x^* minimizes f over a convex set $X \subset \mathbb{R}^n$ if and only if there exists a subgradient $d \in \partial f(x^*)$ such that*

$$d'(x - x^*) \geq 0, \forall x \in X. \tag{3.50}$$

Equivalently, x^* minimizes f over a convex subset X of $\mathbb{R}^n$ if and only if

$$0 \in \partial f(x^*) + T_X(x^*)^*. \tag{3.51}$$

$$\begin{aligned} \min \quad & f(x) && (3.52)\\ \text{s.t.} \quad & h(x) = 0, g(x) \leq 0, && (3.53) \end{aligned}$$

where $h(x) = (h_1(x), \ldots, h_m(x))$ and $g(x) = (g_1(x), \ldots, g_r(x))$.

We define the active constraints

$$A(x^*) = \{j | g_j(x^*) = 0\} \tag{3.54}$$

The set of Lagrange multipliers

$$M := \{(\lambda, \mu) | \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(x^*) + \sum_{i=1}^{r} \mu_i \nabla g_i(x^*) = 0, \mu \geq 0, \mu_i = 0, i \in A(x^*)\} \tag{3.55}$$

Proposition 7 ***(Optimality Conditions)*** *Let x^* be a local minimum of problem 3.52. Then there exist scalars $\mu_0^*, \lambda_1^*, \ldots, \lambda_m^*, \mu_1^*, \ldots, \mu_r^*$, satisfying the following conditions:*

(i) $-(\mu_0^* \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) + \sum_{i=1}^r \mu_i^* \nabla g_i(x^*)) \in T_X(x^*)^*$

(ii) $\mu_i^* \geq 0$ *for all* $i = 0, 1, \ldots, r$

(iii) $\mu_0^*, \lambda_1^*, \ldots, \lambda_m^*, \mu_1^*, \ldots, \mu_r^*$ *are not all equal to* 0.

The lagrangian function

$$L(x, \lambda, \mu) = f(x) + \lambda' h(x) + \mu' g(x) \tag{3.56}$$

The *dual function* is defined by

$$q(\lambda, \mu) = \inf_{x \in X} L(x, \lambda, \mu) \tag{3.57}$$

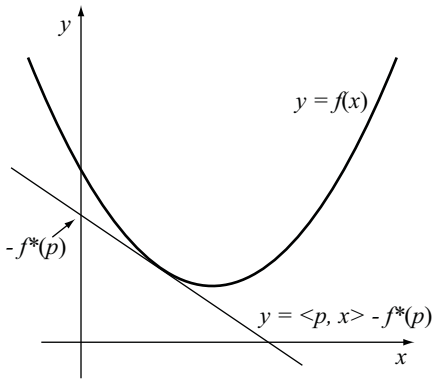


Figure 3.6: The Conjugate duality.

The dual problem is

$$\max \quad q(\lambda, \mu) \tag{3.58}$$

$$\text{s.t.} \quad \lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^r, \mu \geq 0 \tag{3.59}$$

The relation between the dual problem and the max crossing problem can be established by the following set

$$\begin{aligned} M =& \{(v, u, w) | x \in X, h(x) = v, g(x) \leq u, f(x) \leq w\} \\ =& \{(v, u, w) | \text{for some} (\bar{v}, \bar{u}, \bar{w}) \in S, \bar{v} = v, \bar{u} \leq u, \bar{w} \leq w)\} \end{aligned} \tag{3.60}$$

(v) Conjugate duality and Fenchel duality

The *convex conjugate* $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ of a function f is defined by

$$f^*(p) := \sup\{\langle p, x \rangle - f(x) | x \in \mathbb{R}^n\} \tag{3.61}$$

The conjugate function is useful in another duality, namely the Fenchel duality, which is useful in many optimization problems. The Fenchel duality considers the following problem:

$$\min_x f_1(x) - f_2(x) \tag{3.62}$$

$$\text{s.t. } x \in X_1 \cap X_2. \tag{3.63}$$

Here $f_1(\cdot)$ and $f_2(\cdot)$ are functions on X_1 and X_2, respectively. If we apply the Lagrangian duality to the equivalent problem:

$$\min_{y,z} f_1(y) - f_2(z) \tag{3.64}$$

$$\text{s.t. } z = y, \quad y \in X_1, \quad z \in X_2 \tag{3.65}$$

we have the dual problem as

$$\min_{p} \quad f_1^*(p) + f_2^*(p) \tag{3.66}$$

$$\text{s.t.} \quad p \in \Lambda_1 \cap \Lambda_2. \tag{3.67}$$

Here Λ_1 and Λ_2 are the finite range of the conjugate function f_1^* and f_2^*. Note that this is the original form of the dual-decomposition framework ([24, 25, 134, 225]) that is popular in MRF optimization.

Optimization

The above dualities allow us to convert the original optimization problems into their (often approximate) tractable / convex forms. Once we have the tractable problem to solve, algorithms for optimization can be applied. Although there are numerous techniques for solving an optimization problem, the basic idea is always to decrease the energy step by step [214], *e.g.*, the gradient descent algorithm. In particular, when the original problem is combinatorial, one should consider how to achieve integral solution from the continuous solution. For example, the min cut/max flow algorithm [6] is to solve the dual problem of min cut by maximizing the flow in each step. Once the maximal flow is achieved, the integral solution is obtained by seeing if there exists unsaturated flow to the source or the sink. In general, special design of algorithms can achieve guaranteed solution within certain range of the global optimality [258].

Submodularity and convex functions

In combinatorial optimization, one often considers the optimization of a set function $f : 2^E \to \mathbb{R} \cup \{\pm\infty\}$. The tractability of such a combinatorial problem can be characterized by the concept of submodularity as discussed in Sec. 3.1.1. The relationship between submodularity and convexity can be formulated in terms of the *Lovász extension*. The conversion from a set function $f : 2^E \to \mathbb{R} \cup \{\pm\infty\}$ to a continuous convex function $\hat{f} : \mathbb{R}^E \to \mathbb{R} \cup \{\pm\infty\}$ can be done as follows

(i) For each $v \in \mathbb{R}^E$, we index the components of v in non-increasing order as $\{v_1, v_2, \ldots, v_n\}$, where $v_1 \geq v_2 \ldots v_n$ and $n = |E|$.

(ii) Also, we define $V_j := \{v_1, \ldots, v_j\}$ for $j = 1, \ldots, n$ and the characteristic vector of a

subset $S \subset E$ defined by

$$\mathcal{X}_S(v) = \begin{cases} 1 & \text{if } v \in S \\ 0 & \text{otherwise .} \end{cases} \tag{3.68}$$

Then we have

$$v = \sum_{i=1}^{n-1}(v_i - v_{i+1})\mathcal{X}_{V_i} + v_n\mathcal{X}_{V_n} \tag{3.69}$$

(iii) Given the linear combination for any vector $p \in \mathbb{R}^n$, we can define a linear interpolation of the discrete set function $f : 2^E \to \mathbb{R} \cup \{\pm\infty\}$ as

$$\hat{f}(p) := \sum_{i=1}^{n-1}(v_i - v_{i+1})f(V_i) + v_n f(V_n), \tag{3.70}$$

which is known as the **Lovász extension** $\hat{f}$ of f.

The relationship between a submodular function and its Lovász extension is established as follows [80]:

Theorem 11 *A set function f is sub-modular if and only if its Lovász extension $\hat{f}$ is convex.*

Hence, we conclude that submodular functions are discrete analogues of convex functions. ◁

Besides linear programming, there are also other relaxation methods, such as quadric programming for the pairwise MRFs [13, 139, 140].

3.2.2 Discrete MRFs with high-order convex priors – unifying continuous and discrete approaches

In above, we discussed the relaxation approach for general MRF energy functions. In this section, we will discuss the case when each potential function ϕ_a in Equ. 7.2 is convex, or submodular. We will make use of the property that a convex function can be represented by the upper envelop of half planes (Prop. 2 and 5).

Our new formulation of high-order MRF with convex priors (Sec. 3.2.2) is largely motivated by the development of globally optimal solutions for continuous PDEs. For example, the total variation (TV) functional [207] is defined as

$$E^{TV}(u) = \alpha \int |u - u_0|^2 + \mu \int |\nabla u|. \tag{3.71}$$

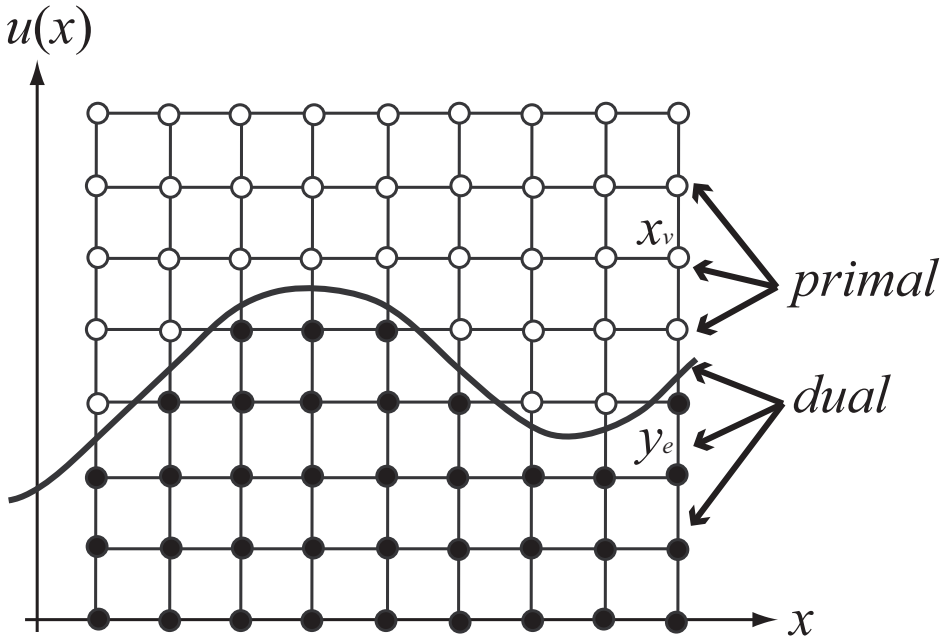


Figure 3.7: The idea of lifting and duality illustrated in the 1D case. The function $u(x)$ is represented implicitly via the characteristic function ϕ which is 1 under the curve and 0 above (Equ. 3.72). The convex relaxation is obtain by relaxing the range of the characteristic function to be within $[0, 1]$ and considering the vector fields in the lifted space. In the discrete setting, solving the original variational problem amounts to finding the optimal value $x_v \in [0, 1]$ on the discrete grid, which can also be constructed via graph-based models.

There are both *continuous* and *combinatorial* (graph-based) models that are formulated for solving the TV functional. It is shown that the TV functional is closely related to certain types of MRF problem, *e.g.*, the convex multilabel problem [200] or the continuous maxflow/mincut problem [9, 235]. Also the TV functional has been employed to solve certain variants of the Mumford-Shah (MS) functional (*e.g.*, [181, 234]) to achieve globally optimal solutions. When the energy is convex (*e.g.*, the TV functional or the continuous cut problem [235]), both continuous and combinatorial approaches are able to find the globally optimal solution via the dual optimization technique.

Continuous approach

The techniques involved in achieving the global optimal solutions of both MS and TV functionals rely heavily on the *upper level set* function $\mathbf{1}_u : \Sigma = \Omega \times \Gamma \rightarrow \{0, 1\}$, which is defined

by

$$\phi_u(x,y,t) \equiv \mathbf{1}_u(x,y,t) = \begin{cases} 1, \text{if } t < u(x,y), \\ 0, \text{otherwise.} \end{cases} \tag{3.72}$$

Here $\Gamma \subset \mathbb{R}$. u can be recovered from its upper level set ϕ_u by the *layer cake formula*:

$$u(x,y) = t_{min} + \int_\Gamma \phi_u(x,y,t)dt. \tag{3.73}$$

If we relax the range of ϕ_u to be within $[0,1]$, the relaxed feasible set of ϕ_u becomes

$$\mathcal{C} = \{\phi : \Sigma \to [0,1] | \phi(x,t_{min}) = 1, \phi(x,t_{max}) = 0\}.$$

This relaxation allows us to treat the optimization problem in a continuous manner.

In order to relate the upper level set function ϕ to the TV model or the MS model, the following minimization problem is considered [187, 286]:

$$\min_{\phi_u \in \mathcal{C}} \{E(u) = \int_\Omega \psi(\nabla \phi_u)\}, \tag{3.74}$$

where ψ is a function defined over the gradient of ϕ.

Primal-dual principle: The *duality* used in here is closely related to the cone theories introduced above. Recall that the *polar cone* of a nonempty set C is given by $C^* = \{y | \langle y, x\rangle \leq 0, \forall x \in C\}$. Consider the set $C_\psi = \{(x,w) | x \in \Sigma, w \in \mathbb{R}, \psi(x) \leq w\}$. The dual variable of ψ is defined by the vector fields (vectors associated to each $x \in \Sigma$) $W_\psi = \{\mathbf{p} | \mathbf{p} \in \Sigma, (\mathbf{p}, -1) \in C_\psi^*\}$, which is also known as the Wulff shape in [187].

The idea of dual optimization is to consider the vector fields $\mathbf{p} \in W_\psi$. When the function $\psi(\cdot)$ is convex (which is the case for the TV functional and the continuous max-flow), it can be shown that C_ψ is also convex so we have $\psi(x) = \max_{\mathbf{p} \in W_\psi} \langle x, \mathbf{p}\rangle$.

So the optimal solution of the original problem is achieved via the minimax (primal-dual) problem

$$\min_u E(u) = \min_{\phi \in \mathcal{C}} \max_{\mathbf{p} \in W_\psi} \{\int_\Sigma \langle \nabla \phi, \mathbf{p}\rangle dv\} \tag{3.75}$$

When there exists discontinuities on u (*e.g.*, the piecewise-smooth MS model), we can still apply the above primal-dual framework to obtain a globally optimal solution of ϕ by adding more complicated constraints over the vector fields ([7, 199]). However, in this case the discontinuities have been treated implicitly. So it is not clear how to obtain a segmentation

simply out of the numerical solution of the upper level set function 3.72 which only encodes the information about u.

To solve the variational problem 3.74 numerically, the space Σ is usually discretized into regular Cartesian grid with vertices $\mathcal{V}^{\Omega} \times \mathcal{K}^{\Gamma}$ and edges $\mathcal{E}$ (Fig 3.7). The goal now is to solve for the 0-1 variables on the nodes of the grid $\mathbf{x} = \{x^C_{p,k} | p \in \mathcal{V}^{\Omega}, k \in \mathcal{K}^{\Gamma}\}$ with the discrete version of the constraints C, *i.e.*, $C_d = \{x^C_{p,k} | x_{p,k_{max}} = 0, x^C_{p,k_{min}} = 1, x^C_{p,k} \in \{0,1\}, \forall p \in \mathcal{V}^{\Omega}, k \in \mathcal{K}^{\Gamma}\}$. The gradient becomes a linear operator A_{ev} that maps vertices to edges as defined in [90]. The dual domain W_{ψ} becomes constraints W for each edge. Solving the primal-dual problem of 3.75 amounts to solving the following problem,

$$\min_{x \in C_d} \max_{y \in W_p} E(\mathbf{x}|\mathbf{y}) = \langle A_{ev}\mathbf{x}, \mathbf{y} \rangle = \sum_{p \in \mathcal{V}^{\Omega}} \sum_{k \in \mathcal{K}^{\Gamma}} \langle A^i_{ev}\mathbf{x}, y_{p,k} \rangle.$$

The reason that continuous method is parallelizable is that the energy can be decomposed into similar local kernels. So it is amenable to GPU implementation. In the Appendix B, we derive a similar formulation for the primal-dual principle of pairwise MRF optimization.

Combinatorial approach

Now we employ the ideas in the continuous approaches introduced above to solve the combinatorial MRF problem. Given a hyper-graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, we consider the high-order MRF problem with convex priors:

$$E(\mathbf{x}) = \sum_{v \in \mathcal{V}} \theta_v(x_v) + \sum_{e \in \mathcal{E}} \theta_e(x^e_1, x^e_2, \ldots, x^e_{|e|}), \tag{3.76}$$

where $x^e_i \in \mathcal{L} = \{1, 2, \ldots, L\}$ and $\theta_e(x^e_1, x^e_2, \ldots, x^e_{|e|}), e \in \mathcal{E}$ are discrete convex functions of $\{x^e_1, x^e_2, \ldots, x^e_{|e|}\}$. Since the function $\theta_e(x_1, x_2, \ldots, x_{|e|})$ is convex, according to Prop. 5. it can be represented as follows [123]:

$$\theta_e(x^e_1, x^e_2, \ldots, x^e_{|e|}) = \max_{k \in \mathcal{I}_e} \{\sum_{i \in e} \theta^k_{ei} x^e_i + \theta^k_{e0}\}, \tag{3.77}$$

where $\mathcal{I}_e$ is the index set for the upper envelop functions. Fig. 3.8 shows an example of using upper envelop for representing convex functions. Note that with this representation, the complexity of the function is no longer determined by the order of its energy function. Instead, its complexity is determined by the number of linear envelop functions.

In order to apply the duality principle discussed in Sec. 3.2.1, the first step is to convert the problem of 3.76 to LP. To this end, we define the indicator variables using the idea of upper

level set function ([200, 286]) as follows (Fig. 3.7):

$$\tau_{u;i} = \begin{cases} 1 & \text{if } i \leq x_u \\ 0 & \text{otherwise} \end{cases}, u \in \mathcal{V} \text{ and } i \in \mathcal{L}. \tag{3.78}$$

Note that this definition is different from the definition of Equ. 7.3. It follows that $\theta_v(\cdot)$ can be represented as

$$\theta_v(x) = \sum_{i \in \mathcal{L}} \theta_{v;i}(\tau_{v;i} - \tau_{v;i+1}) \tag{3.79}$$

where we define $\tau_{v;1} = 1$, $\tau_{v;L+1} = 0$ and $\theta_{v;i} = \theta_v(i)$. This corresponds to the continuous case of representing the zero-derivative terms (*e.g.*, the first term of Equ. 3.71) using upper level sets function. One advantage of using such indicator variables is that the complexity of an MRF problem is no longer determined by its order, as we shall see in the following. Another advantage, compared to the standard indicator variable (see Appendix A), is that the variable x_v can be conveniently represented by the equation $x_v = \sum_{i \in \mathcal{L}} \tau_{v;i}$. ◁

Therefore, the original convex MRF optimization problem (Equ. 3.76) becomes

$$\min_{\tau, \mathbf{x}} \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{L}} \theta_{v;i}(\tau_{v;i} - \tau_{v;i+1}) + \sum_{e \in \mathcal{E}} \max_{k \in \mathcal{I}_e} \{ \sum_{v \in e} \theta_{ev}^k x_v + \theta_{e0}^k \} \tag{3.80}$$

$$\text{s.t. } x_v = \sum_{i \in L} \tau_{v;i} \qquad \forall v \in \mathcal{V} \tag{3.81}$$

$$\tau_{v;i} - \tau_{v;i+1} \leq 1 \qquad \forall v \in \mathcal{V}, i \in \mathcal{L} \tag{3.82}$$

$$\tau_{v;i+1} - \tau_{v;i} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{L} \tag{3.83}$$

$$\tau_{v;i} \in \{0, 1\} \tag{3.84}$$

By introducing auxiliary variables $\{t_e\}$ for each $e \in \mathcal{E}$, the above optimization can be converted into a LP problem:

$$\min_{\tau, \mathbf{x}} \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{L}} \theta_{v;i}(\tau_{v;i} - \tau_{v;i+1}) + \sum_{e \in \mathcal{E}} t_e \tag{3.85}$$

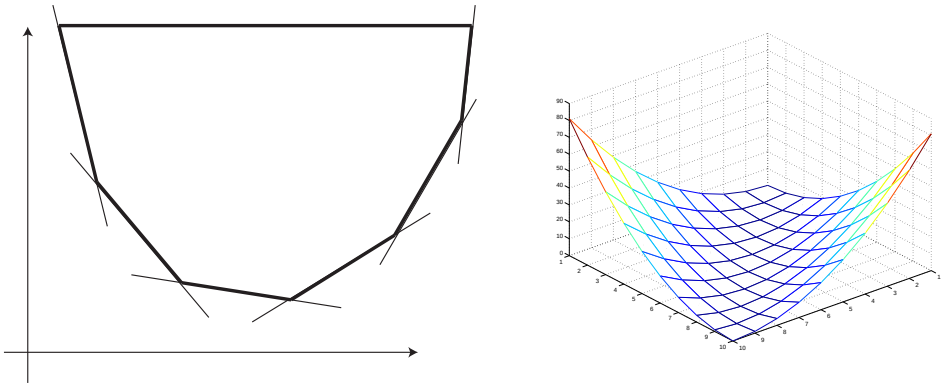
$$\text{s.t. } x_v = \sum_{i \in L} \tau_{v;i} \qquad \forall v \in \mathcal{V} \tag{3.86}$$

$$\tau_{v;i} - \tau_{v;i+1} \leq 1 \qquad \forall v \in \mathcal{V}, i \in \mathcal{L} \tag{3.87}$$

$$\tau_{v;i+1} - \tau_{v;i} \leq 0 \qquad \forall v \in \mathcal{V}, i \in \mathcal{L} \tag{3.88}$$

$$\sum_{v \in e} \theta_{ev}^k x_v + \theta_{e0}^k \leq t_e \qquad \forall e \in \mathcal{E}, k \in \mathcal{I}_e \tag{3.89}$$

$$\tau_{v;i} \in \{0, 1\} \tag{3.90}$$



(a) 1D (b) 2D

Figure 3.8: A convex function $V(X_1, X_2)$ can be represented as the upper bound of a set of linear functions $\max_i\{f_i(X_1, X_2) = a_i X_1 + b_i X_2 + c_i\}$. For a convex function, the number of linear functions is $O(|X_1||X_2|)$. This convexification does not reduce the number of the primal variables. The problem can be simplified when the convex function is in a simple form, namely with a small number of linear envelops.

Duality

Now we consider the duality of the LP-relaxation of Eq. 3.85, by introducing the dual variables

$$x_v = \sum_{i\in\mathcal{L}} \tau_{v;i} \qquad \rightarrow \text{ merge to 3.89} \tag{3.91}$$

$$\tau_{v;i} - \tau_{v;i+1} \leq 1 \qquad \rightarrow \mu^1_{v;i} \tag{3.92}$$

$$\tau_{v;i+1} - \tau_{v;i} \leq 0 \qquad \rightarrow \mu^2_{v;i} \tag{3.93}$$

$$\sum_{v\in e} \theta^k_{ev} x_v + \theta^k_{e0} \leq t_e \qquad \rightarrow m_{e;k} \tag{3.94}$$

Note that in this formulation, the number of dual variable depends on the number of upper linear envelop functions, namely $O(|\mathcal{V}||\mathcal{L}| + \sum_{e\in\mathcal{E}} |\mathcal{I}_e|)$, which is independent of the order of

the energy functions. Then the dual problem for the primal energy function Eq. 3.85 becomes:

$$\max \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{L}} \mu^1_{v;i} \tag{3.95}$$

$$\text{s.t.}(\mu^1_{v;i} - \mu^1_{v;i-1}) - (\mu^2_{v;i} - \mu^2_{v;i-1}) + \sum_{\{e|i\in e\}}\sum_{k\in\mathcal{I}_e} \theta^k_{ev} m_{e;k} \leq \theta_{v;i} - \theta_{v;i-1}, \quad \forall v, i \tag{3.96}$$

$$\sum_{v\in\mathcal{V}}\sum_{k\in\mathcal{I}_e} \theta^k_{e0} m_{e;k} \leq 0 \tag{3.97}$$

$$-\sum_{k\in\mathcal{I}_e} m_{e;k} \leq 1 \tag{3.98}$$

$$m_{e;k} \leq 0, \mu^1_{v;i} \leq 0, \mu^2_{v;i} \leq 0 \quad \forall e, k \tag{3.99}$$

Note that the number of dual constraints is in the order of $O(|\mathcal{V}||\mathcal{L}| + \sum_{e\in\mathcal{E}} |\mathcal{I}_e|)$.

Optimality conditions and integral primal solutions

For the integer programming of problem (3.85), the *complementary slackness* condition holds for the optimal solution of the primal and dual problems:

$$\mu^1_{v;i} < 0 \Rightarrow \tau_{v;i} - \tau_{v;i+1} = 1 \tag{3.100}$$

$$\mu^2_{v;i} < 0 \Rightarrow \tau_{v;i} = \tau_{v;i+1} \tag{3.101}$$

It follows that given the dual solution $\{\mu^1_{v;i}, \mu^2_{v;i}\}$ the integral primal solution can be obtained by checking if $\mu^1_{v;i} < 0$. Note that when the convex relaxation is not tight, other techniques can be applied for obtaining the integral solutions (*e.g.*, [280]).

Relation to previous algorithms

In above, we presented a new LP relaxation for the high-order MRF problem with convex priors, which is motivated by the upper level set function used in the continuous cases.

(i) Relation to the min-cut/max-flow algorithm for Graph Cuts

As a special case of Equ. 3.76, the energy function of the binary graph cuts can be formulated as

$$\min_{x_v\in\{0,1\},v\in\mathcal{V}} \{E(x) = \sum_{v\in\mathcal{V}} \theta_v(x_v) + \sum_{(u,v)\in\mathcal{E}} \theta_{uv}(x_u, x_v)\}. \tag{3.102}$$

Here the pairwise energy function $\theta(\cdot,\cdot)$ can always be represented by two upper envelop functions (Fig. 3.9). Using the upper level set based indicator variables, we have $\tau_{v;0} = 1$ and

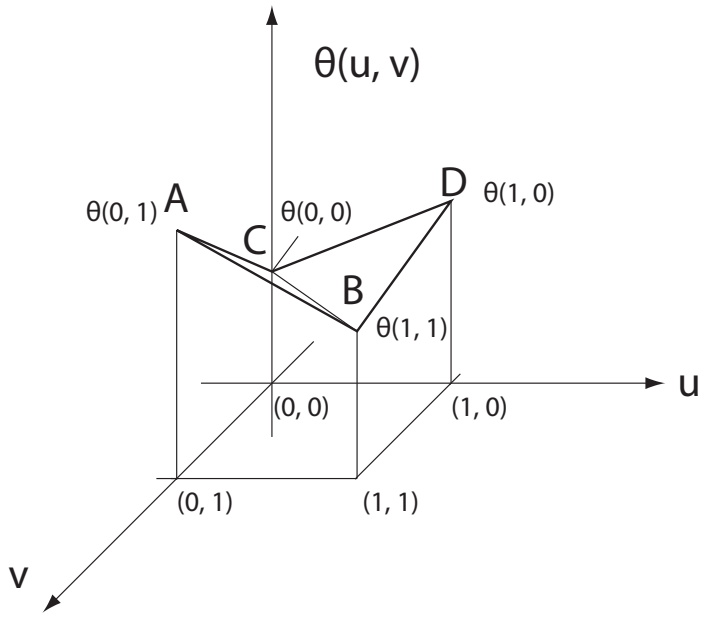


Figure 3.9: Linear upper envelop functions for the pairwise terms $\theta(\cdot, \cdot)$ in the binary graph cuts problem. Here the upper envelop functions are the planes that span ABC and BCD, respectively.

$x_v = 1 + \tau_{v;1}$. It follows that the energy function can be represented as

$$E(\tau, \mathbf{t}) = \sum_v (\theta_{v;0}(\tau_{v;0} - \tau_{v;1}) + \theta_{v;1}\tau_{v;1}) + \sum_{(u,v)} t_{uv} \tag{3.103}$$

$$= \sum_v \tau_{v;1}(\theta_{v;1} - \theta_{v;0}) + \sum_{(u,v)} t_{uv} + \text{constant}, \tag{3.104}$$

Assuming the two linear upper envelop function for each pairwise term $\theta_{uv}(x_u, x_v)$ are $a^1_{uv}x_u + b^1_{uv}x_v + c^1_{uv}$ and $a^2_{uv}x_u + b^2_{uv}x_v + c^2_{uv}$ respectively, the constraints for the graph cuts energy become

$$\tau_{v;1} \leq 1, \qquad \forall v \in \mathcal{V} \tag{3.105}$$

$$a^1_{uv}(1 + \tau_{u;1}) + b^1_{uv}(1 + \tau_{v;1}) + c^1_{uv} \leq t_{uv}, \qquad \forall (u, v) \in \mathcal{E} \tag{3.106}$$

$$a^2_{uv}(1 + \tau_{u;1}) + b^2_{uv}(1 + \tau_{v;1}) + c^2_{uv} \leq t_{uv}, \qquad \forall (u, v) \in \mathcal{E} \tag{3.107}$$

$$\tau_{v;1} \geq 0 \tag{3.108}$$

By introducing the dual variable w^1_{uv} and w^2_{uv}, it can be shown that our new LP relaxation lead to an alternative formulation of the min-cut/max-flow problem.

Graph theory	Linear algebra	Vector calculus	Topology
Functions on vertices	Vector in $\mathbb{R}^n$	Potential function	0-cochain
Edge flow	Skew-symmetric matrix in $\mathbb{R}^{n\times n}$	Vector field	1-cochain
Triangular flow	Skew-symmetric hyper-matrix in $\mathbb{R}^{n\times n\times n}$	Tensor field	2-cochain

Table 3.1: A comparison between graph structure and geometric structure.

(ii) Relation to Convex MRFs

The energy function considered in the convex MRF literature (*e.g.*, [105, 129]) takes the following form

$$\min_{\mathbf{x}} \sum_{v\in\mathcal{V}} D_v(x_v) + \sum_{(u,v)\in\mathcal{E}} V_{uv}(x_v - x_u), \tag{3.109}$$

where $V_{uv}(\cdot)$ is a convex function (*e.g.*, the discrete gradient $(x_v - x_v)^2$). In such cases, it is easy to see the the number of upper linear envelop functions for each $V_{uv}(\cdot)$ is at most $2|\mathcal{L}|$) (Fig. 3.8 (a)). Hence, the number of dual variables required for solving the convex MRF problem is $O(|\mathcal{L}||\mathcal{V}|)$.

3.3 Conclusion

In this section, we have reviewed the combinatorial structures on graphs. We have shown that the dependency in probability functions can be sufficiently characterized by the connectivity structure on a graph, each node of which represents the random variables. To conduct inference on graphs, convexity plays an important role in the search of global optimal solutions. In the continuous setting, convex optimization has been well studied as a tractable structure [23, 31, 214]. In the discrete setting, convexity also represents a class of tractable structures and is represented by the concept of submodularity [80, 177]. Furthermore, we have shown in Sec. 3.2.2 that all the exponential families with convex potential functions can be succinctly relaxed using linear programming, therefore can be efficiently solved / approximated. The complexity of marginal polytope can be characterized by the number of facets, which can be exponentially large. However, in the case of tree-structured graphs, this number turns out to be polynomial. This can be understood also from the message passing point of view [11].

The Markov property allows us to represent the level of dependency in a probabilistic system, though many of its problems are intractable (NP-hard). The junction tree can be

used to characterize such a complexity, namely the existence of loops can sometimes lead the problem to be exponentially large.

Linear programming (LP) relaxation has been proven to be a very useful tool in solving the MRF optimization problem. Besides LP, the quadric / semidefinite programming relaxation [12, 13] is also a popular approach for solving combinatorial problems. Although its applications have already been explored in MRF optimization (*e.g.*, [139, 140]), its full potential is yet to be investigated. A comparison between the combinatorial structure and graph structure can be summarized in Table 3.1.

The Physical law can be mostly described by linear / convex interactions. A nonlinear / nonconvex problem can be converted into a linear / convex one, but often with the price of being exponentially complex. It is yet to be investigated the relationship between information theory and complexity. One possible explanation for the fact that the computational power is bounded by the polynomials is that the computer is only a physical machine, so it must abide by the physical law, which is linear as we have shown in Chapter 2.

Chapter 4

Dense Surface Registration using High-order Graph Matching

4.1 Introduction

Accurately finding the correspondences between two or more surfaces (surface registration) is the prerequisite for many applications, such as 3D shape recognition, deformation transfer, and object recognition [46, 257]. Despite the increasing popularity of 3D acquisition techniques, it remains challenging to accurately register 3D shapes that undergo large, non-rigid deformations or are subject to a high level of noise. In this chapter, we propose a high-order graph based approach for the problem of surface registration that integrates multiple assumptions on the quality of any match, which handles surfaces with large, non-rigid deformations, partial mapping and/or noisy input.

In order to find the correspondences between two surfaces undergoing large deformations, directly searching in their embedding space is often inefficient or even intractable due to the large degrees of freedom in the their non-rigid deformations. A popular approach that overcomes such difficulty is to resort to intrinsic representations of the surfaces that are invariant to certain types of deformations (*e.g.*, [38]). Most prior arts that fall into such intrinsic space based methods assume that two surfaces undergo only *isometric* (*i.e.*, distance or metric preserving) deformations, which is often a rough approximation for the deformations of most real-world objects. Successful approaches for such intrinsic space based methods include

geodesic/exponential maps ([37]), conformal maps ([269, 270, 289, 295]) and diffusion maps ([188, 218]). Among these approaches, only conformal maps provide a closed-form solution for the dense surface matching problem, which is characterized by the Möbius transformation in the uniformization domain that only requires the determination of up to three correspondences between two surfaces. Moreover, it has been shown in [5] that a direct generalization of the conformal mapping, *i.e.*, the quasi-conformal mapping, is able to handle arbitrary diffeomorphisms between two surfaces.

Although conformal maps are well-suited for establishing dense correspondences between surfaces undergoing large deformations, relying on a single mapping is often sensitive to noisy input, or boundary conditions. It is therefore necessary to consider the correspondences obtained from multiple mappings and define robust metrics to handle uncertainties in the input. To this purpose, graph-based approaches ([124]) have proven to be very successful in handling noisy input and occlusions (*e.g.*, [267]) due to their statistical nature. Hence, in this paper, we cast the surface registration problem into the search of the correspondences for a discrete set of points on the surface. We show that the deformation constraints between two surfaces can be represented by the third-order interactions among these point sets. Thus finding the optimal registration becomes finding the optimal matching for a high-order graph by minimizing an energy which constrains the quality of any possible matching. To fully utilize the powers of intrinsic space-based methods and graph-based approach, we propose two formulations to solve the sparse and dense surface registration problems respectively, the combination of which allows us to achieve both efficient and accurate dense surface registration results.

In our first stage, inspired by the metric defined in [156], we formulate the surface registration problem as a high-order graph matching problem that integrates both extrinsic matching costs (texture and normal consistencies) and intrinsic deformation errors (curvature and deviation from isometry [156]). Due to the challenge imposed by solving a high-order graph matching problem, we propose the first solution to the high-order graph matching problem that handles partial matching by using the dual-decomposition technique [134]. However, the metric defined in [156] is only robust to points on the surface that are far apart. When the points to be registered go dense, any possible triplet matching would result in a small distance, which leads the metric to be no longer discriminating. Hence this formulation can only handle sparse surface registration problems.

Towards dense surface registration, from the results obtained by the sparse registration, we design an efficient matching candidate selection scheme for any point on the surface based on the fact that any three correspondences determine a unique closed-form solution to establish the mapping between two surfaces. Thanks to such candidate selection, the problem of dense surface registration becomes combinatorial and local. Finally, a local high-order graph matching algorithm is proposed to achieve accurate dense surface registration results. Note that

the idea of using multiple Möbius transformations induced by any three correspondences to achieve dense registration was first appeared in our paper [292]. Such an idea is later extended recently in computer graphics by Kim *et.al.* [120].

This chapter is organized as follows. In Sec. 4.2, we review related work on surface registration and graph matching. Sec. 4.3 overviews our two-stage hierarchical surface matching framework. Our high-order graph matching problem for surface registration is formulated in Sec. 4.5. In Sec. 4.6 we present our novel candidate selection scheme for dense surface registration. Given the candidate selection scheme, the algorithm for dense surface registration is discussed in Sec. 4.7. The implementation details and experimental results are presented in Sec. 4.8. Finally, we conclude our work in Sec. 4.9.

4.2 Literature review

4.2.1 Surface registration

Surface matching (also known as *surface registration* or *surface alignment*), whose objective is to determine meaningful correspondences between two or more surfaces, is a fundamental problem in computer vision, computer graphics and medical imaging. Using the Bayesian paradigm, one can formulate the surface registration problem as follows:

$$\arg \max_{T:\mathcal{M}_1 \to \mathcal{M}_2} P(T|\mathcal{M}_1, \mathcal{M}_2) \propto P(\mathcal{M}_1, \mathcal{M}_2|T)P(T), \tag{4.1}$$

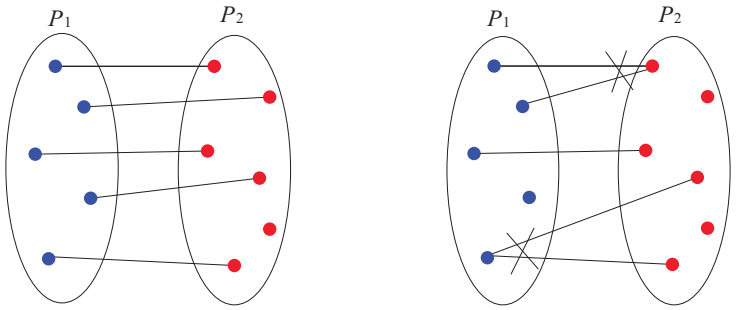
where T is the mapping between two input surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$, represented either as analytical functions, point clouds or triangular meshes. Almost all the existing approaches for surface registration make certain assumptions on the possible transformations T, or $P(T)$. Despite a large amount of literature on surface matching, it remains a very challenging problem, particularly when the surfaces undergo large, non-rigid deformations and are subject to a high level of noise. In order to handle surface matching problems under such difficult situations, it is often necessary to take into account both local feature similarities and global deformation constrains [257]. While local structures are somewhat straightforward to handle, the consideration of global structures imposes a major challenge for surface matching.

In order to impose global deformation constraints for the surface matching problem, many existing works are based on certain *rigidity* assumption on the deformations of the surface and impose rigidity as a global regularization in searching for correspondences. Assuming two representations of a shape only differ by a global rigid deformation (*i.e.*, rotation and translation, or $T(p) = \text{Rot}p + \text{Trans}$), the iterative closest points (ICP) [26] method and its variants [167, 209] has been successfully applied for shape registration with various extensions (*e.g.*,

[40, 96, 109]). ICP alternates between determining correspondences between two surfaces and computing a rigid transformation using the obtained correspondences. Obviously, such a scheme can easily get trapped in local minima and thus requires that the difference between the two surfaces are close enough to get a satisfactory matching result. Furthermore, global rigidity does not take into account bendable shapes (*e.g.*, garments or rubber bands). Hence the notion of *local rigidity* has been proposed which assumes that the deformation between two local neighborhoods of each correspondence be rigid [104] (or mathematically, $T(p) = \text{Rot}_p p + \text{Trans}_p$). Similar to ICP, an alternating optimization is usually required to optimize the objective function. Also such methods can easily stuck in local optimality. However, since the rigidity assumption is imposed onto every point, when registering two surfaces with large deformations, the degree of freedom of the deformation is also increased, leading to high complexity in computation.

As an alternative representation of the local rigidity, approaches based on geodesic distances among pairs of points on the surface have proven to be robust to large deformations [36, 37]. In Riemannian geometry, the notion of local rigidity can be characterized by the concept of *isometry* (Chap 2.2), meaning the preservation of the lengths of any infinitesimal vector between any correspondence (or $dT(\cdot)$ is orthogonal). Furthermore, a surface is isometrically deformed if and only if for any pair of points on the surface, the geodesic distance between their correspondences on the deformed surface is the same as the geodesic distance between the two points on the original surface. This property allows us to define a distance between any pair of correspondences to impose the local rigidity constrain. For example, [8] formulated the surface matching problem as a pairwise MRF optimization problem. They consider pairwise potentials between neighbor points based on the deviation of geodesic distance, and they employ the Loopy Belief Propagation algorithm for the MRF optimization.

Although pairwise distance can be robustly evaluated for surfaces with large deformations [51, 121, 173, 184, 283], it is difficult to apply such methods to handle shape matching with isometric ambiguities such as symmetry or when the data is corrupted by noise. Another successful approach for matching shapes with large deformations is the conformal mapping based method (*e.g.*, [120, 269, 270, 289, 295]). A conformal mapping is characterized by requiring that the eigenvalues of the transformation matrix $dT(\cdot)$ be equal. An important property of conformal mapping is that if two surfaces are isometrically deformed from one to the other, their correspondences only differ by a Möbius transformation in their conformal parametrization (also known as Uniformization) domains. Hence, once such a transformation is recovered, a one-to-one correspondences between the two surfaces can be established, giving us a global transformation between two surfaces. Based on such global property of conformal mapping, [269, 270, 289, 295] established dense correspondences between two surfaces by specifying a few initial feature correspondences. These approaches relied heav-



(a) An example of valid matching (b) An example of invalid mathcing

Figure 4.1: The graph matching problem can be stated as finding the one-to-one correspondence between two sets of points P_1 and P_2. (a) shows an example of valid matching. (b) is an example of invalid matching.

ily on the accuracy of the selection of the initial correspondence points. Given noisy 3D scan data with varying scale, boundaries and resolutions, the performance of the above methods might degrade as it is difficult to find reliable feature point correspondences. To remedy this, [156] proposed to find sparse correspondences between two surfaces based on a voting scheme. Since every three correspondences determine a unique Möbius transform between the uniformization domain of the two surfaces, it also determines a correspondence match between two surfaces. Hence, for each possible triplet matching, one can define a measure of the plausibility of such matching by matching among all the other points on the whole surface. This global property of the conformal mapping based method guarantees the robustness of the method with respect to local noise as well as large deformation. Besides, [156] also showed its method's ability to handle two surfaces with largely inconsistent boundaries, which is usually difficult for conformal mapping based methods.

Furthermore, since most surface deformations are not perfectly isometric, solely considering intrinsic embedding information may introduce approximation errors to the matching result. Due to the above mentioned uncertainty, [246, 247] proposed to search for the correspondences using probabilistic formulations and randomized algorithms. However, due to the need for computing geodesics on the surface, they are only able to establish a few hundred correspondences.

4.2.2 Graph matching

The graph matching problem, also known as assignment problem or simply matching problem [44], deals with the problem of how to assign n items (jobs, students, men) to m other items

(machines, tasks, women). Mathematically, the *bipartite graph matching* problem is defined on a bipartite graph $\mathcal{G} = (U, V; E)$ where U and V are two vertex sets and $E \subset U \times V$ is an edge set. A matching M is a subset of the edges such that every vertex of G meets at most one edge of the matching. Equivalently, a matching is a map $\varphi : U \to V$, such that the mapping is one-to-one, *i.e.*,

$$\varphi(i) \neq \varphi(j) \qquad \text{whenever } i \neq j. \tag{4.2}$$

One could also extend this formulation to dealing with *partial matching*, where a valid matching is defined by the mapping $\varphi' : U \to V \cup \{\emptyset\}$, such that

$$\varphi'(i) \neq \varphi'(j) \qquad \text{whenever } i \neq j \text{ and } \varphi'(i) \neq \emptyset \text{ and } \varphi'(j) \neq \emptyset. \tag{4.3}$$

When $|U| = |V|$ and every vertex of G coincides with an edge of the matching M, such matching is called a *perfect matching*, which has been widely studied since the 1930s'. In 1935 Philip Hall [97] proved the *marriage theorem* which gives a necessary and sufficient condition for the existence of a perfect matching. However, such proof is non-constructive. It had been shown by Hopcroft and Karp [102] that a perfect matching can be found in $O(m\sqrt{n})$ time where m is the number of edges. The idea of their algorithm is based on iteratively searching for an augmenting path that alternates between matched and unmatched edges. Due to the matroid structure of the matching problem, it can be shown that upon termination, this algorithm is able to find the matching with maximal size. Thus it can also be applied to finding the match M with maximal number of matching pairs, known as the *maximum bipartite matching* problem.

The above matching problems concern the (cardinality) number of matching pairs, without distinguishing the differences among different pairs. If we assign a weight to each match and ask for the optimal matching whose sum of weights are maximal, then such problems are called *weighted assignment problem*. The weight can be defined on each matching pair, or multiple matching pairs to constrain the configurations of matching. With respect to the maximal number of matching pairs that are assigned the weight, there are *linear assignment problems*, *quadratic assignment problems* and *multi-index assignment problems*.

Linear assignment problem

In the simplest case, we can define the *linear assignment problem* as follows. Let $C = (c_{ij})$ be an $n \times m$ matrix where each entry c_{ij} denotes the cost of assigning node i to node j, then the objective function of an assignment $\varphi : U \to V$ is

$$\sum_{i=1}^{n} c_{i\varphi(i)}. \tag{4.4}$$

Hence the linear assignment problem becomes finding the optimal assignment φ that minimizes Eq. 4.4.

In 1946 Easterfield proposed the first algorithm for solving the linear assignment problem which runs in $O(2^n n^2)$ time [70]. In 1955 the Hungarian algorithm was proposed by Harold Kuhn [138], which is the genesis of the network flow based algorithm that later gained widespread popularity in the combinatorial optimization community. The time complexity of the original algorithm proposed by Kuhn is $O(n^4)$ and was later improved to achieve an $O(n^3)$ running time by Dinic and Kronrod [62].

Beyond those deterministic algorithms for the linear assignment problem, there are also randomized approaches. For example, Bersekas [22] proposed a randomized dual algorithm and achieved significant speedup in the average case.

Quadratic assignment problem

Not surprisingly, the cost function of quadratic assignment problem involves a four-index cost matrix $D = (d_{ijkl})$ where $d_{ij;kl}$ denotes the cost of assigning i to k and j to l.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij;\varphi(i)\varphi(j)}. \tag{4.5}$$

Note that the linear cost can be encoded into quadric cost as well, by defining

$$d_{ii,jj} = c_{ij}, \forall i \in \{1, \ldots, n\}, \in \{1, \ldots, m\}. \tag{4.6}$$

The quadratic assignment problem was first introduced as a mathematical model for the location of indivisible economical activities by Koopmans and Beckmann [136] in 1957. In their paper, they considered a less general cost matrix where $d_{ij;kl} = a_{ik}b_{jl}$. It turns out that there are numerous real-world applications that can be modeled as quadratic assignment problems. For example, the wiring problem [232], the hospital layout problem [137], etc. However, solving a quadratic assignment problem is much more difficult than solving the linear assignment problem since it is an NP-Hard problem [210]. Nugent, Vollmann and Ruml [48] proposed a set of test instances for such problem that are widely considered to be difficult to solve. Approaches for solving the quadratic assignment problem includes decomposition, branch-and-bound, and branch-and-cut, which are all based on traditional optimization techniques. Also there are parallel algorithms and randomized algorithms such as simulated annealing or genetic algorithms developed.

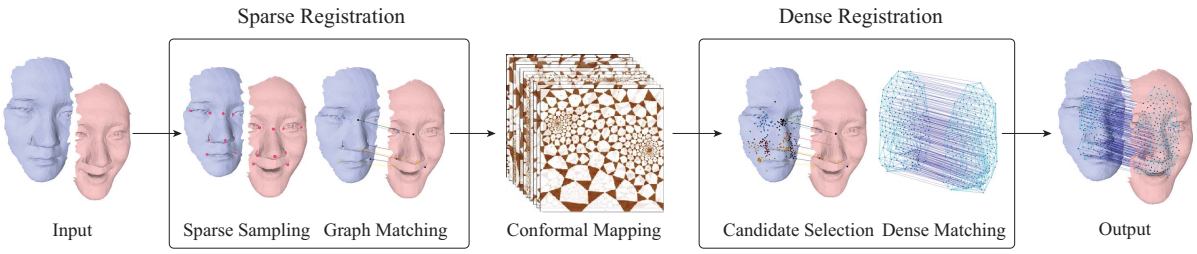


Figure 4.2: Overview of our algorithmic framework for surface registration as described in Sec. 4.3.

Multi-index assignment problem

The multi-index assignment problem, as a natural extension of linear assignment problems, were introduced by Pierskalla [195] in 1968. The dimension of the cost matrix for a k-index problem is $(nm)^k$. For example, the cost function of a 3-index problem can be represented as

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} d_{ijk;\varphi(i)\varphi(j)\varphi(k)}. \tag{4.7}$$

One can encode the linear assignment problem by defining

$$d_{iii,jjj} = c_{ij}, \tag{4.8}$$

and the quadratic assignment problem by defining

$$d_{iji,klk} = d_{jii,lkk} = d_{iij,kkl} = d_{ij,kl}. \tag{4.9}$$

Similar to the quadratic assignment problem,the multi-index assignment problem can be solved by using its linear programming formulation and employing optimization techniques based on integer programming. For example, Balas and Saltzman [15] solved the 3-index assignment problem using the branch and bound technique, which involves subgradient optimization for the relaxed dual problem.

4.3 System overview

As shown in Fig. 4.2, our surface registration system consists of two stages: sparse surface registration and dense surface registration.

4.3.1 Sparse surface registration

In the sparse feature matching stage, an initial set of sparse feature points are selected among the local maxima of Gaussian curvature [156] from the input surfaces S_1 and S_2. Using our high-order graph matching algorithm (Sec. 4.4 and 4.5), we find n_s correspondences between the two feature sets $\{p_1^1, p_2^1 \ldots, p_{n_s}^1\} \rightarrow \{p_1^2, p_2^2, \ldots p_{n_s}^2\}$. Since the matching cost involves computing the matching cost for every possible triplet matching, the complexity of encoding the graph energy grows in the order of $o(n_s^6)$. Because the accuracy of the sparse matching stage determines that of the dense matching, here we only select a small set of feature points (typically $8-15$ in our experiments) and compute all the possible triplet matching cost among them. Hence the computational cost is low in finding the sparse correspondences and computing the associated conformal maps.

4.3.2 Dense surface registration

In the dense feature matching stage, we first use the results from the sparse matching stage to obtain a set of candidates for each point on the surface (Sec. 4.6). Since every three correspondences determine a unique conformal map between two surfaces, n_s sparse matching points would give us $\binom{n_s}{3}$ candidate matching points on S_2 for any point $p \in S_1$. These points are then clustered to obtain meaningful matching candidates for each point. Such candidate selection also allows us to formulate the dense surface registration problem as a combinatorial problem. To impose constrains on the deformations between two surfaces, we propose a generic local surface deformation model defined on the triangulated graph of these points on S_1 with meaningful candidate matching points. Finally, a high-order MRF optimization is then formulated and solved to obtain the optimal dense registration result (Sec. 4.7).

4.4 A pseudo-boolean formulation of the high-order graph matching problem

Let us denote by P_1 and P_2 the set of points from two surfaces S_1 and S_2 respectively (Fig. 4.1). $P \equiv P_1 \times P_2$ denotes the set of possible correspondences. We define the boolean indicator variable

$$x_a = \begin{cases} 1 & \text{if } a = (i,j) \in P \text{ is a correspondence,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

A basic constraint is that each point in P_1 is mapped to at most one point in P_2, while for each point in P_2 there is at most one point in P_1 mapping to it. Therefore, we have the set of feasible solutions defined as,

$$\mathcal{C} = \{\mathbf{x} \in \{0,1\}^{P_1 \times P_2} | \sum_{i \in P_1} x_{i,j} \leq 1, \sum_{j \in P_2} x_{i,j} \leq 1, \\ \forall i \in P_1 \text{ and } \forall j \in P_2\}. \tag{4.11}$$

Moreover as shown in [67], in order for matching to be scale invariant a high-order (degree 3) graph matching is required as follows

$$\min_{\mathbf{x} \in \mathcal{C}} \{E(\mathbf{x}|\theta) = \sum_{a \in P} \theta_a x_a + \sum_{(a,b) \in P \times P} \theta_{ab} x_a x_b + \\ \sum_{(a,b,c) \in P \times P \times P} \theta_{abc} x_a x_b x_c\}, \tag{4.12}$$

where θ_a is the matching cost for each correspondence $a \in P$, θ_{ab} for a pair of correspondences $(a,b) \in P \times P$, and θ_{abc} for a triplet of correspondences $(a,b,c) \in P \times P \times P$.

Because the matching constraint (Eq. 4.11) makes the optimization problem of Eq. 4.12 difficult to solve, most existing works attempted to relax it (*e.g.*, [67]). In fact, the matching constraint can be reduced to pairwise terms in the energy function. We observe the following equivalence:

$$\forall i \in P_1, \sum_{j \in P_2} x_{i,j} \leq 1 \\ \text{iff } \min_{x_{i,j}} \sum_{j', j'' \in P_2, j' \neq j''} \theta^{\infty} x_{i,j'} x_{i,j''} = 0 \tag{4.13}$$

where θ^{∞} is a sufficiently large number. We use $P^{\mathcal{C}}$ to denote the set of pairs that encodes the matching constraints for all the correspondences. Thus, the general high-order matching problem can be formulated as the following pseudo-boolean optimization problem [30]

$$\min_{\mathbf{x} \in \{0,1\}^{P_1 \times P_2}} \{E(\mathbf{x}|\theta) = \sum_{a \in P} \theta_a x_a + \sum_{(a,b) \in P \times P} \theta_{ab} x_a x_b + \\ \sum_{(a,b) \in P^{\mathcal{C}}} \theta^{\infty} x_a x_b + \sum_{(a,b,c) \in P \times P \times P} \theta_{abc} x_a x_b x_c\}. \tag{4.14}$$

The above formulation is general and therefore is able to capture almost all matching scenarios (*e.g.*, partial matching) by properly defining the potentials.

Because of the positive weight θ^{∞} that encodes the matching constraint, the energy function 4.14 is nonconvex [31], and in general this is an NP-hard problem [30]. The advantage of

the pseudo-boolean formulation is that theoretically any high-order terms can be reduced into a quadratic term [30] which can be done efficiently [106]. In this paper we employ the flexible dual-decomposition technique [134] which often obtains a near optimal solution. The details of the optimization algorithm is given in Sec. 4.4.1.

4.4.1 Optimization and computational complexity

The idea of dual-decomposition is to re-formulate the original problem as the union of several sub-problems that are easier to solve [133, 134]. For the graph matching problem in Eq. 4.12, let θ denote the vector of the weights of the singleton, pairwise and triplet terms, and I denote the set of subproblems. The decomposition is represented by $E(\mathbf{x}|\theta) = \sum_{\sigma \in I} \rho_\sigma E^\sigma(x|\theta^\sigma)$, where ρ_σ is the weight for each subproblem. Then the original problem is solved by updating the parameter θ^σ of each subproblem σ that increases the energy of the dual problem. Moreover, we have the following decomposition constraint:

$$\sum_{\sigma \in I} \rho_\sigma \theta^\sigma = \theta. \tag{4.15}$$

If we can find a lower bound $\Phi_\sigma(\theta^\sigma)$ for each subproblem, *i.e.*, $\Phi_\sigma(\theta^\sigma) \leq \min_x E^\sigma(x|\theta^\sigma)$, then we can obtain a lower bound for the original problem, *i.e.*,

$$\Phi(\theta) = \sum_{\sigma \in I} \rho_\sigma \Phi_\sigma(\theta^\sigma) \leq \sum_{\sigma \in I} \rho_\sigma E^\sigma(\mathbf{x}|\theta^\sigma) = E(\mathbf{x}|\theta). \tag{4.16}$$

In particular, we decompose the original problem into the following three subproblems:

(i) a **linear subproblem** which considers only the singleton term $\sum_{a \in P} \theta_a x_a$. This linear subproblem is also known as the linear assignment problem [6].

(ii) a **higher-order pseudo-boolean subproblem** by reducing the high-order terms in 4.12 to quadratic terms [30] which can be solved by the QPBO algorithm [128]. Here we employ [106] for the reduction.

(iii) a **local subproblem** which divides the original surface into small regions and uses an exhaustive search to find the optimal solution in each small surface region.

Opposite to [252] that only considers pairwise subproblems, a high-order pseudo-boolean subproblem is introduced because of the high-order terms in Eq. 4.12. After solving the subproblems, the dual variables $\{\theta^\sigma\}$ are updated by projecting to the space that satisfies Eq. 4.15 as in [133].

For the dual-decomposition algorithm above, the most expensive step in each iteration is the max-flow computation. In our paper, we use the popular implementation in [135], whose worst case complexity is $O(mn^2|C|)$, where m is the number of edges, n is the number of vertices, and $|C|$ is the cost of minimum cut. Assuming we select $|P_1|$ and $|P_2|$ feature points from two surfaces, there are $O(|P_1|^3|P_2|^3)$ possible triplets, each represented by a high-order term in Eq. 4.12. After the reduction from the high-order terms to quadratic terms, we can significantly reduce the complexity without searching for all possible matching correspondences.

4.5 High-order graph matching for sparse surface matching

To consider multiple sources of similarity measurements, in our graph matching formulation of Eq. 4.12, the singleton terms define both appearance and geometric similarities, the pairwise terms constrain the mapping to be one-to-one and the high-order terms constrain the intrinsic deformation errors. Furthermore, the potential functions can also be learned from a training set [45].

The singleton potential

For each correspondence (i, j) we consider both the geometric and texture (if available) information to define its potential as in [249]. For simplicity, we use the Gaussian curvature $\text{curv}(i)$ at point i, which is invariant under an isometric transformation [63], and the texture value $\text{tex}(i)$ at point i. The singleton potential for a correspondence (i, j) is defined as

$$\theta_{i,j} = (\text{curv}(i) - \text{curv}(j))^2 + \lambda_0(\text{tex}(i) - \text{tex}(j))^2 \tag{4.17}$$

where λ_0 balances the weight between the curvature and the texture information. Similarly, other features can also be considered such as spin-image [108].

The pairwise potential

The pairwise potential encodes the one-to-one mapping constrain for the graph matching of Eq. 4.13, *i.e.*, for any possible matching pairs $(p_i^1, p_j^1) \in S_1$ and $(p_i^2, p_j^2) \in S_2$, we define the pairwise matching cost:

$$\theta_{ij} = \begin{cases} \theta^\infty & \text{if } p_i^1 = p_j^2 \text{ or } p_i^2 = p_j^2 \\ 0 & \text{otherwise .} \end{cases} \tag{4.18}$$

Here θ^∞ is set to be 10^5 in our experiments.

The high-order potential

According to the uniformization theorem [73], any 3D surface can be flattened conformally to a canonical 2D domain. Within such a mapping each feature point p has a parametric coordinate in the complex plane $z_p \in \hat{\mathbb{C}}$. The flexibility of this conformal mapping is represented by a Möbius transform, which can be uniquely determined by fixing any three points on the surface. Inspired by Lipman *et.al.* [156], we compute the matching score between two triplets as the deformation error based on the Möbius transform.

Given two surfaces, S_1 and S_2, for any two triplets, $(p_i^1, p_j^1, p_k^1) \in S_1$ and $(p_i^2, p_j^2, p_k^2) \in S_2$, we first recover the associated Möbius transformation $m^1(z)$ and $m^2(z)$ that map each triplet to a constant configuration $(e^{i\frac{2\pi}{3}}, e^{i\frac{4\pi}{3}}, e^{i2\pi})$. This transformation essentially equips each point in the sets P_1 and P_2 with a new coordinate in $\hat{\mathbb{C}}$. Let us denote the new coordinate for each point p as $z(p) \in \hat{\mathbb{C}}$. Similar to [156], we establish correspondences between the two sets P_1 and P_2 by searching the mutually closest point correspondences set under the new coordinates, denoted as

$$\begin{aligned}\mathcal{M}_{ijk} =& \{(p_1, p_2) | p_1 \in S_1, p_2 \in S_2, \\ & \forall\, p_2' \in S_2 \setminus \{p_2\}, |z(p_1) - z(p_2)| < |z(p_1) - z(p_2')|, \\ & \forall\, p_1' \in S_1 \setminus \{p_1\}, |z(p_1) - z(p_2)| < |z(p_1') - z(p_2)|\}\end{aligned}$$

and define the deformation error as

$$E_{ijk} = \sum_{p \in P_1, \mathcal{M}_{ijk}(p) \neq \emptyset} |z(p) - z(\mathcal{M}_{ijk}(p))|^2. \tag{4.19}$$

Then we define the Möbius matching potential as follows,

$$\theta_{ijk}^{\text{Möbius}} = \begin{cases} \frac{E_{ijk}}{|\mathcal{M}_{ijk}|^2} - 1. & \text{if } \frac{E_{ijk}}{|\mathcal{M}_{ijk}|} < \delta \\ 1/|\mathcal{M}_{ijk}| & \text{otherwise} \end{cases} \tag{4.20}$$

Here δ is lower bound value to single out unlikely correspondences (in our experiment $\delta = 0.1$). Without it the minimization problem of Eq. 4.12 would encourage as many as possible correspondences even when some of them do not match. Intuitively, if there were more matching pairs and the distances between those matching pairs were smaller, the potential would be lower.

However, considering the Möbius energy alone can introduce certain ambiguity since it encodes only isometric information. To resolve such ambiguity, we consider the Gaussian map of the surface. The Gaussian map is defined as the mapping of the normal at each point on the surface to the unit sphere [63]. The Gaussian map captures the extrinsic geometric

information of the surface. In order to avoid ambiguities in orientation, the orientation of the Gaussian maps is considered for each of the triplets. Two triplets have the same orientation if and only if the determinant of their normals have the same sign. Therefore, we have

$$\theta_{ijk}^{\text{Gaussian}} = \begin{cases} 0 & \text{if } \det(\mathbf{n}_i^1, \mathbf{n}_j^1, \mathbf{n}_k^1) \det(\mathbf{n}_i^2, \mathbf{n}_j^2, \mathbf{n}_k^2) \geq 0 \\ 1/|\mathcal{M}_{ijk}| & \text{otherwise} \end{cases} \tag{4.21}$$

Here $\mathbf{n}_i \in \mathbb{R}^3$ denotes the normal at point i, and $\det(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)$ denotes the determinant of the 3×3 matrix $[\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k]$. This is introduced as a soft constraint in our framework, because in the extreme case, the normal can reverse its orientations when the surface undergoes very large deformations. Finally, the triple potential for each possible triple matching $(p_i^1, p_j^1, p_k^1) \rightarrow (p_i^2, p_j^2, p_k^2)$ can be defined as

$$\theta_{ijk} = \lambda_1 \theta_{ijk}^{\text{Möbius}} + \lambda_2 \theta_{ijk}^{\text{Gaussian}} \tag{4.22}$$

4.6 Candidate selection for dense surface registration

The number of vertices n considered in this high-order formulation is the main computational bottleneck of our approach. In particular, when n becomes large, as in the case of dense surface registration, it is computationally expensive to solve the high-order graph matching problem. Furthermore, the optimality properties of the obtained solution degrade since the assumption of isometry is only an approximation and applies only when the features are far apart. The graph structure of the above matching problem would also be very complex if we consider all possible triplets. Several heuristic ways were considered to prune off some triplets, such as restricting the number of triangles per vertex [67]. However, because of the complexity of the problem, such pruning schemes often lead to erroneous matching results when the number of feature points is large. To reduce the computational complexity, we propose a two-stage optimization pipeline including *sparse feature matching* and *dense point matching*, as illustrated in Fig. 4.2.

Since the initial feature points are selected among the vertices and the middle points of the edges of the meshes, the matching results could be unreliable if the mesh resolution is low. To address the above issue, we consider all conformal maps induced by different Möbius transformations, which are determined by every three correspondences between two surfaces, for the **dense point matching**.

4.6.1 Candidate selection

From the sparse feature matching stage, we have a set of sparse correspondences $\{p_1^1, p_2^1 \ldots, p_{n_s}^1\} \rightarrow \{p_1^2, p_2^2, \ldots p_{n_s}^2\}$ between S_1 and S_2. Because the deformation of the surface might not be isometric, we propose a candidate selection scheme based on Möbius transforms to compensate for the approximation error. Given any three correspondence pairs, $\{p_i^1, p_j^1, p_k^1\} \rightarrow \{p_i^2, p_j^2, p_k^2\}$, the Möbius transformation can be computed efficiently in a closed form [156]. Under such a Möbius transformation, any point $p^1 \in S_1$ will be mapped to a different candidate location $c(p^1) \in S_2$. Thus, for each point in the source surface, we can compute the candidate locations in the target surface by considering all possible Möbius transformations from the feature correspondences.

One advantage of our candidate approach is robustness. If any part of the sparse matching result from Sec. 4.5 is accurate, the matching candidates given by the Möbius groups will distribute closely around the true location. Another advantage is that this voting scheme provides a fast and effective way of constraining the search space for any point on the surface.

4.6.2 Candidate clustering

Based on the above candidate locations, we want to use the underlying distribution to reduce our search space for the dense matching. It is also important that the dense matching should optimize the same objective as in the sparse matching stage. For any voting point $c(p^1) \in S_2$ of a source point $p^1 \in S_1$ that is obtained by aligning three correspondences $\{p_i^1, p_j^1, p_k^1\} \rightarrow \{p_i^2, p_j^2, p_k^2\}(i, j, k = 1 \ldots n)$, there is a cost $\theta_{ijk}^{\text{Möbius}}$ in the matching energy of Eq. 4.12. Intuitively, the lower the value of $\theta_{ijk}^{\text{Möbius}}$ and the closer the curvature and texture is, the more likely p^1 and $c(p^1)$ match. Therefore, we define the likelihood of each candidate matching $p^1 \rightarrow c(p^1)$ under the alignment of $\{p_i^1, p_j^1, p_k^1\} \rightarrow \{p_i^2, p_j^2, p_k^2\}$ as follows

$$f_{ijk}(p^1, c(p^1)) = e^{-\theta_{ijk}^{\text{Möbius}}} \tag{4.23}$$

where $\theta_{ijk}^{\text{Möbius}}$ is the Möbius matching potential in Eq. 4.20. To obtain the candidate distribution for each point $p^1 \in S_1$, we use a kernel density estimate (KDE) with the density function defined as

$$\rho(p^1, c(p^1)) = \sum_c f_{ijk}(p^1, c(p^1)) K(\frac{\|c(p^1) - c(p_c^1)\|}{h}) \tag{4.24}$$

where $c(p_c^1)$ is the center location of each kernel K in S_2 and h is the kernel bandwidth. The mean shift clustering [55] is employed to find the modes of this density. Compared to parametric representations, KDE has does not require nonlinear optimization to learn the distribution parameters.

Since we search for the modes in Eq. 4.24 on the 2D manifold instead of the 3D Euclidean embedding space, the distance function should be defined as the geodesic distance on the surface. However, as illustrated in Fig. 4.5 most of the candidate locations are close to the center, so the Euclidean distance is used in our method to simplify the mode search. To handle partial surface matching, we only select the modes with density higher than 0.1 and the closest point on the surface as the candidate matching point. If no such mode exists, we report that there is no reliable matching point. The average number of resulting candidate points in our experiments is $1-6$. So our candidate selection and clustering method can significantly reduce the search space.

Fig. 4.5, 4.6 and 4.7 show some examples of candidate matching.

4.7 Dense surface registration

Based on the matching candidates obtained for each vertex, our goal now is to find a good matching position locally for each dense point. This problem can be formulated similarly to the high-order graph matching problem defined in Sec. 4.4. Since the candidate voting scheme in Sec. 4.5 has removed the ambiguities caused by the Möbius transforms, we only need to consider the matching cost based on texture and geometric similarities defined in Eq. 4.17, as well as the orientation consistency defined in Eq. 4.21. Furthermore, the orientation consistency term can be defined locally, *i.e.*, each triangle $\triangle_{p_1p_2p_3}$ and its matched triangle $\triangle_{p'_1p'_2p'_3}$ should have the same orientation in the uniformization domain, which is known as having no flip in [219]. More specifically, for the three vertices of each triangle $\triangle_{123}$, we define the potential of matching (p_1, p'_1), (p_2, p'_2) and (p_3, p'_3) as follows

$$\theta_{123,1'2'3'} = \begin{cases} \theta^{\infty} & \text{sign}(\triangle_{123}) \neq \text{sign}(\triangle_{1'2'3'}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.25}$$

Here θ^{∞} is a sufficiently large number and $\text{sign}(\triangle_{123})$ and $\text{sign}(\triangle_{1'2'3'})$ denote the orientation of the triangle $p_1p_2p_3$ and $p'_1p'_2p'_3$, respectively, in the uniformization domain. From the candidate voting it is not guaranteed that every point has at least one matching candidate. Therefore, we remove the points without any matching candidate and obtain a triangulation for the remaining points on S_1 through the Delaunay triangulation algorithm [60] in the uniformization domain.

Suppose for each point $p \in S_1$, its matching candidates are given by $\mathcal{C}_p = \{p_i | p_i \in S_2, i = 1, 2, \ldots, n_p\}$. We define the boolean indicator variable:

$$x_p^i = \begin{cases} 1 & \text{if } p, p_i \in \mathcal{C}_p \text{ are correspondences} \\ 0 & \text{otherwise.} \end{cases} \tag{4.26}$$

Assuming that each $p \in S_1$ is matched to at most one of its candidates, we have the matching constraint:

$$\sum_{p_i \in \mathcal{C}_p} x_p^i \leq 1 \tag{4.27}$$

Therefore, the same optimization technique as described in Sec. 4.4.1 can be applied to solve the above problem.

Compared to the graph matching problem in Sec. 4.4.1, one major advantage of our local graph matching algorithm is that the number of matching candidates of each point is typically less than 6 and, therefore, the number of variables is very small. In particular, to match n points locally, there are only $O(n)$ variables and $O(n)$ triplet terms since the dense points are triangulated in the planar parametric domain.

4.8 Experimental results

In this section we evaluate our new algorithmic framework for dense surface matching. In our experiments, we match surfaces with large deformations and inconsistent boundaries (partial overlapping). The number of vertices for each mesh is in the range of $1,500 - 4,000$. With our high-order graph matching algorithm, we can find the dense matching for $60 - 90$ percent of all vertices, which is illustrated as matched/total (no. of matched vertices/no. of total vertices of the source surface) for each example. The lion data of Fig. 4.8 comes from [237] and the face and hand data are captured with texture by the 3D scanner introduced in [270]. To measure the quality of dense registration, from the Delaunay triangulation of the points on the source surface, we consider the ratio of the area of each local triangle to the area of its matched triangle. For the natural deformations (*e.g.*, expression change, stretched arms or bending figures) we experimented with, the local area is not expected to undergo abrupt change. Therefore the area ratio is expected to be close to one for every local triangle.

Matching with largely inconsistent boundaries and partial overlapping: The mid-edge uniformization algorithm allows to map the boundaries of the surface to slits and preserve the conformal structure of the surface in an exact sense. Hence it is suitable for matching partially overlapping surfaces. This property can be combined with our candidate voting scheme to determine the outliers near the boundary where the mean shift clustering returns a low score. Examples are shown in Fig. 4.11, 4.14, and 4.12. An example of significant non-overlap between the two meshes is shown in Fig. 4.2.

Matching with large deformations: Fig. 4.12 and 4.13 show results that match two surfaces undergoing a large deformation. Even when the sparse features can not all be selected

consistently (as shown in Fig. 4.13), our high-order graph matching algorithm in Sec. 4.4.1 is able to find reliable sparse correspondences (Fig. 4.13(a)) and obtain a dense surface matching result through the two-stage optimization scheme (Fig. 4.13(b)).

Comparison experiments: Fig. 4.14 shows a comparison between our algorithm and the least square conformal mapping (LSCM) approach [269]. Although LSCM can handle free boundaries, there is no theoretical guarantee that the conformal structure is preserved near the boundary and it might include self-intersections in the mapping [219]. In our comparison, we use the feature correspondences computed from the sparse matching stage to initialize the LSCM experiments. The inaccuracy of the LSCM approach can be observed in Fig. 4.14(c). In this example, although all vertices on the left mesh are matched to the right mesh, there are approximately 42 percent flipped triangles. Note that here we cannot compare directly with the results in [269] where the initial feature points were manually selected.

4.9 Conclusion

We proposed an algorithmic framework for non-rigid surface matching. In particular, a high-order graph matching formulation is used to combine the appearance and geometry similarity as well as the implicit embedding energy between deformed surfaces. Therefore, our proposed method can establish robust sparse registration between non-rigid surfaces with large deformations, partial matching and inconsistent boundaries and scales. Furthermore, a two-stage algorithm is proposed to constrain the search space through candidate selection and local graph matching, which allows to achieve dense surface registration with a sub-vertex accuracy. The method is modular both with respect to the density of points as well as the potentials used to determine optimal partial correspondences.

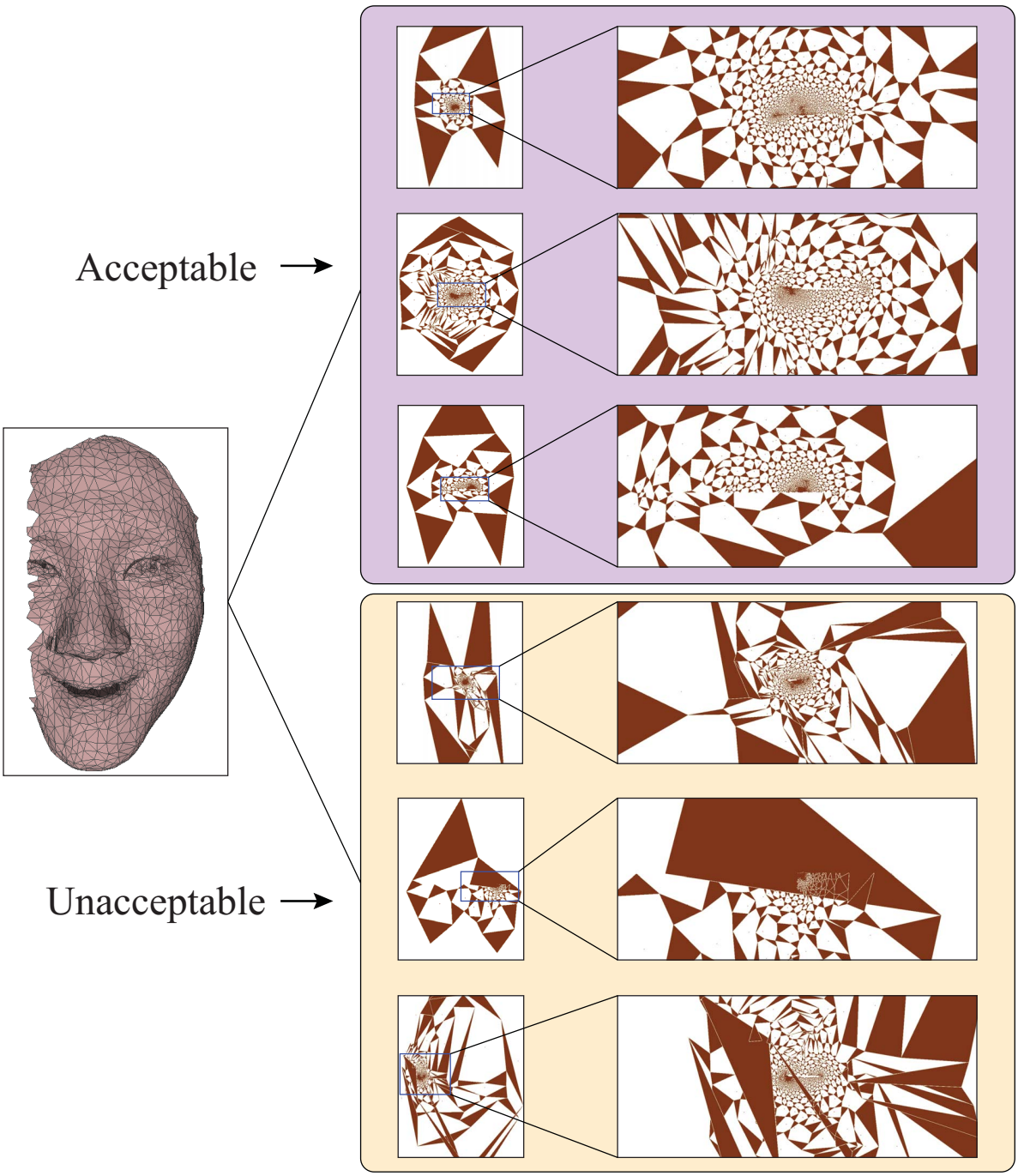


Figure 4.3: Numerical stability problem using conformal mapping.

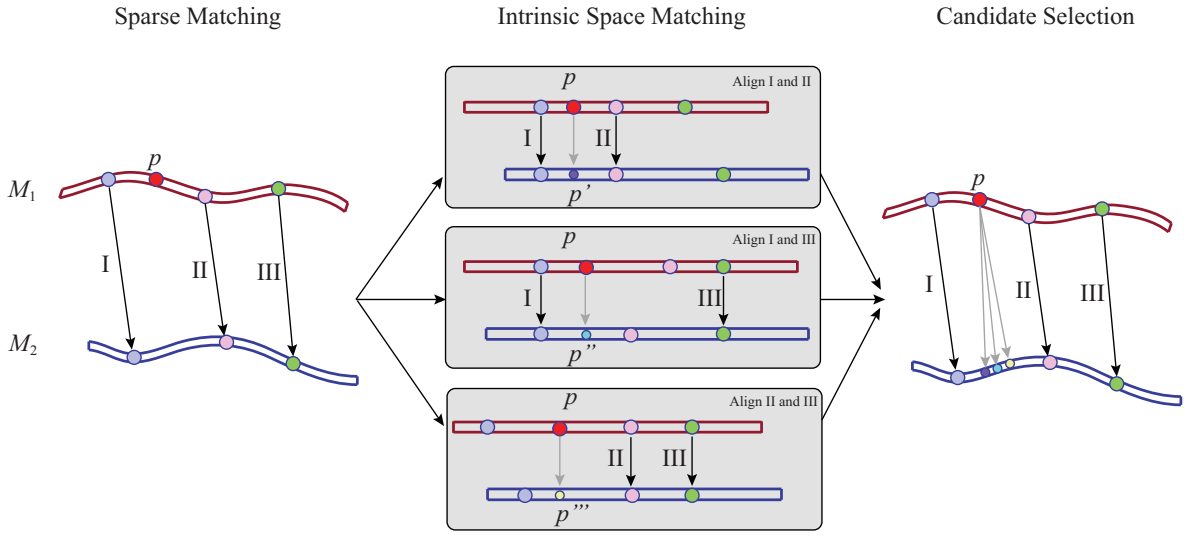


Figure 4.4: Illustration of candidate selection scheme in 2D.

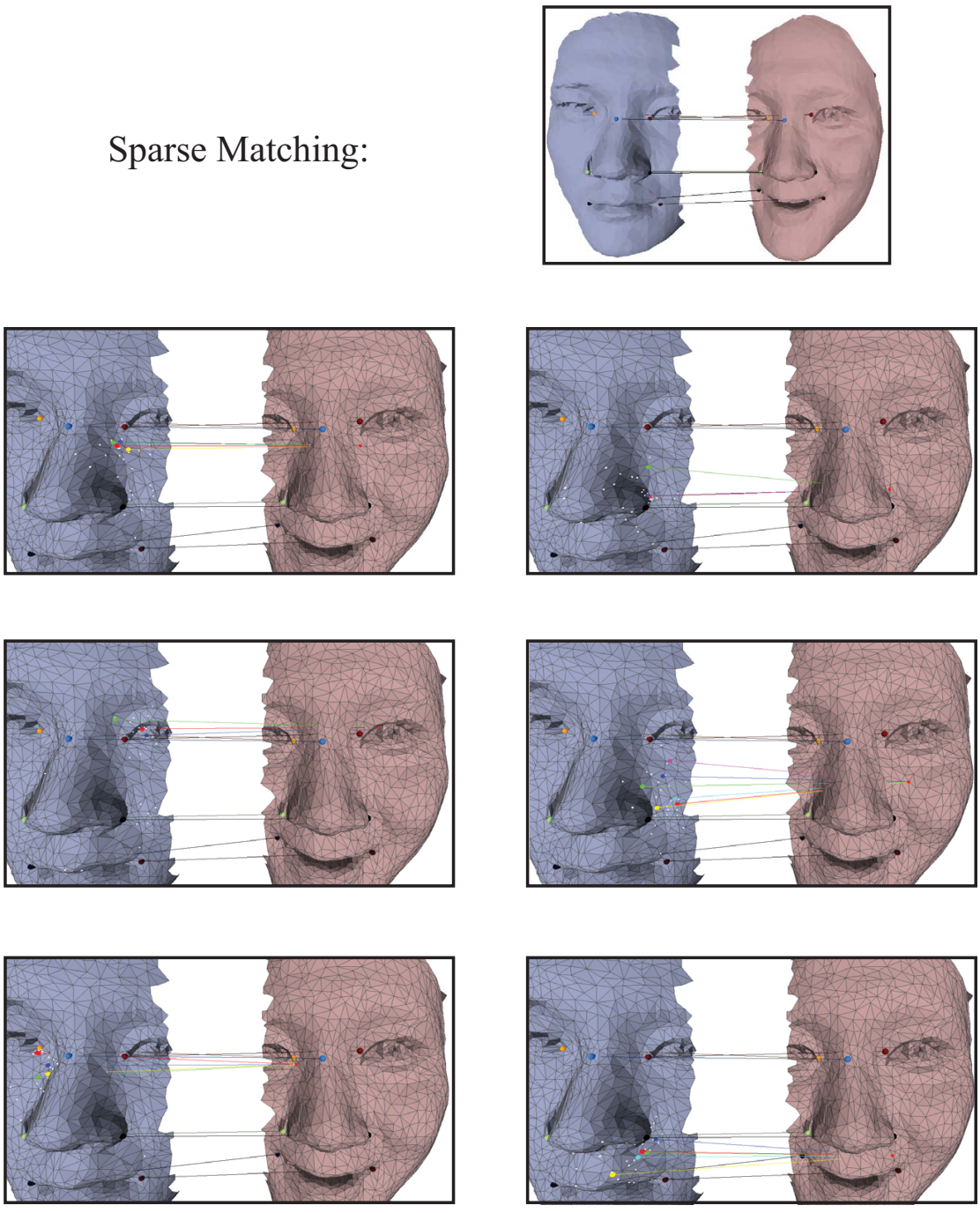


Figure 4.5: Visualization of selected candidates. (Best viewed in color)

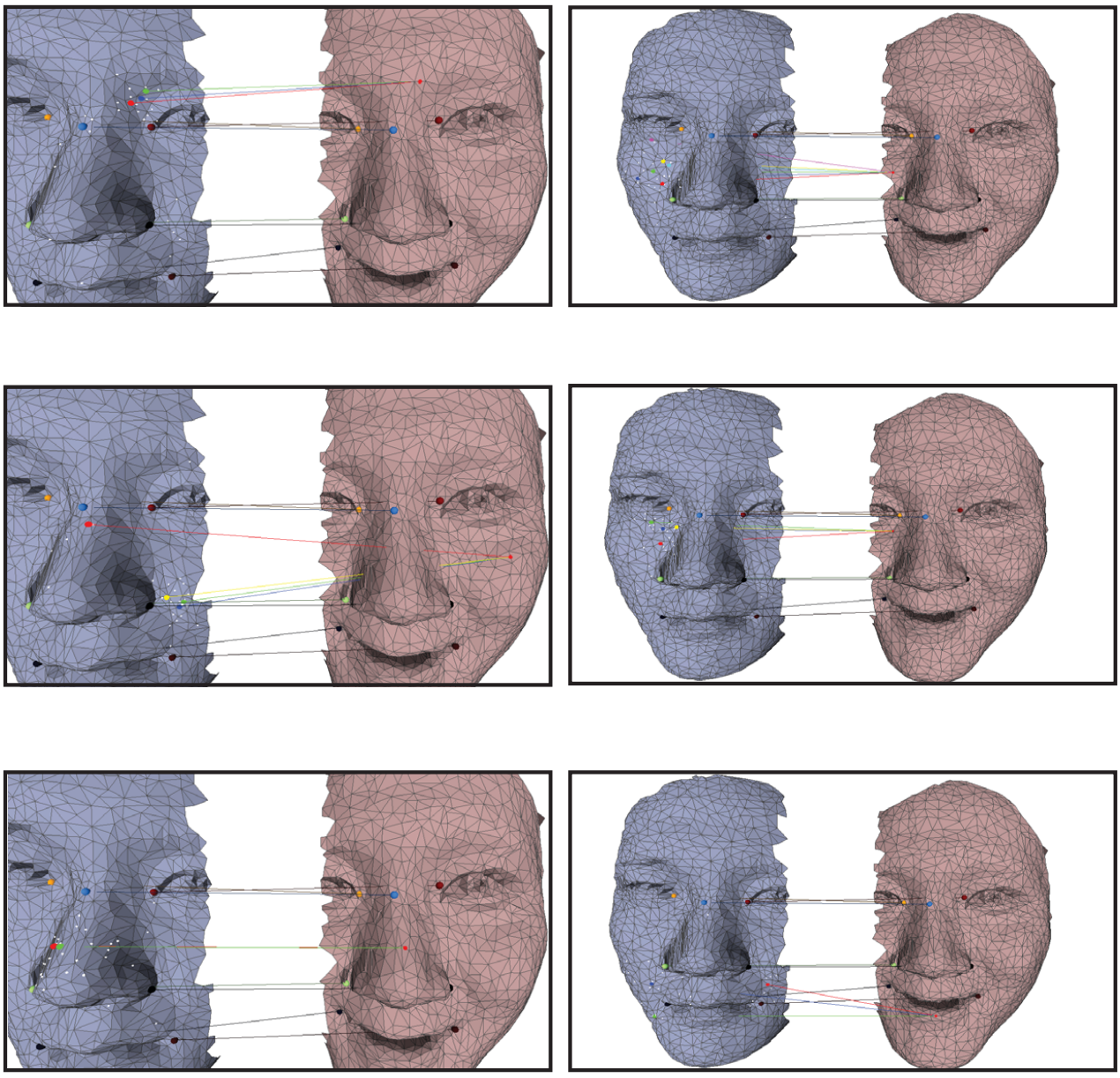


Figure 4.6: Visualization of selected candidates. (Best viewed in color)

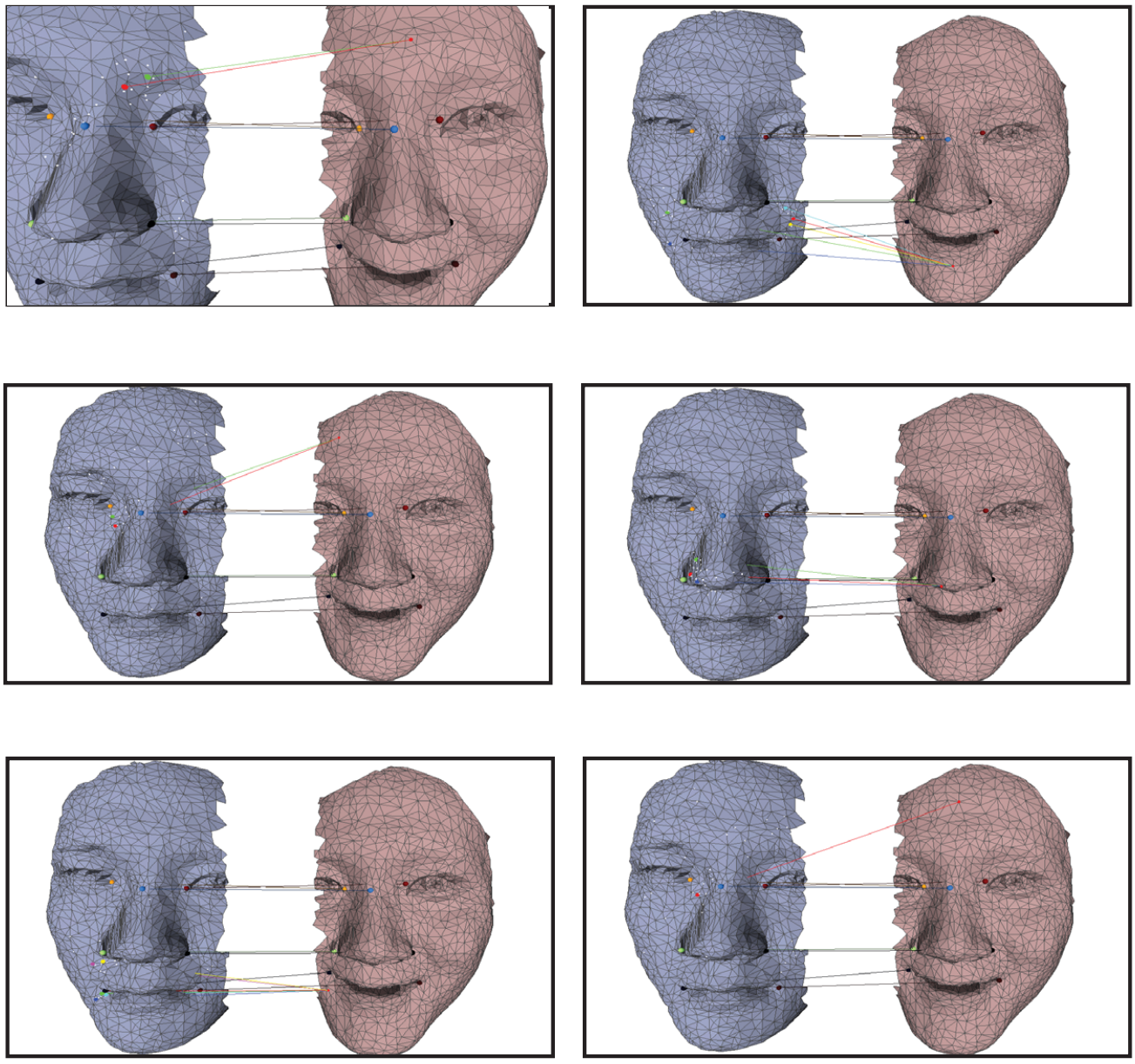
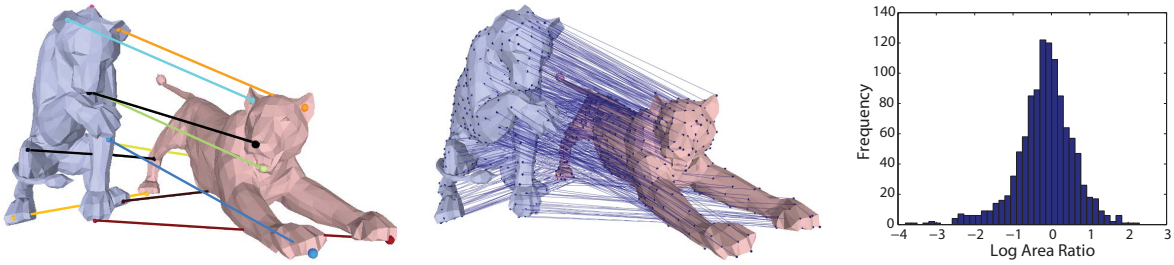
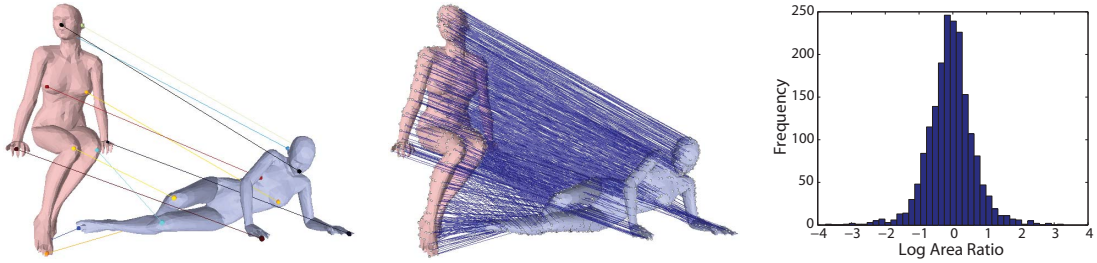


Figure 4.7: Visualization of selected candidates. (Best viewed in color)



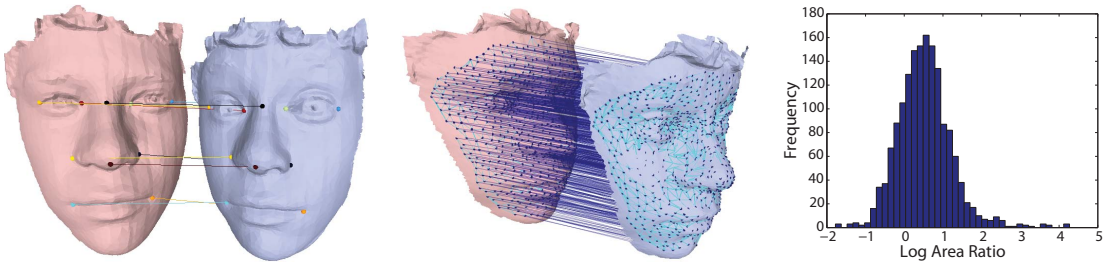
(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.8: Matching result of animal example.



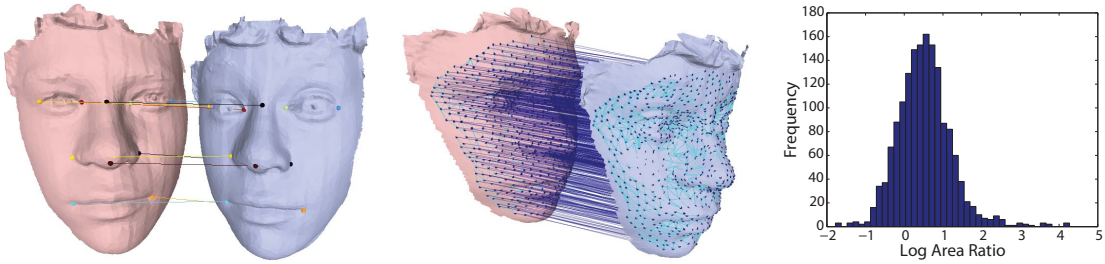
(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.9: Matching result of body example.



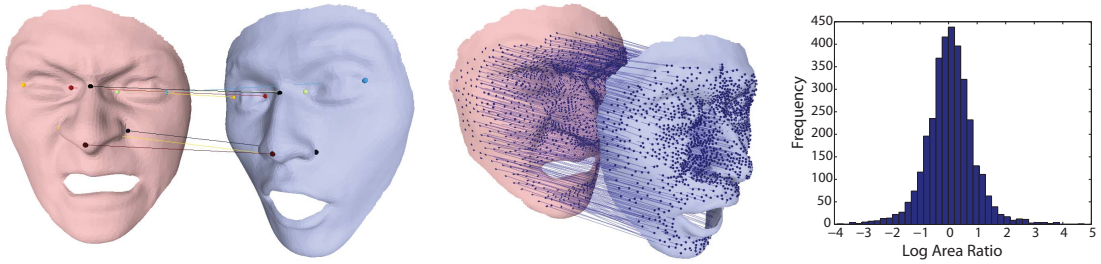
(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.10: Matching result of face example.



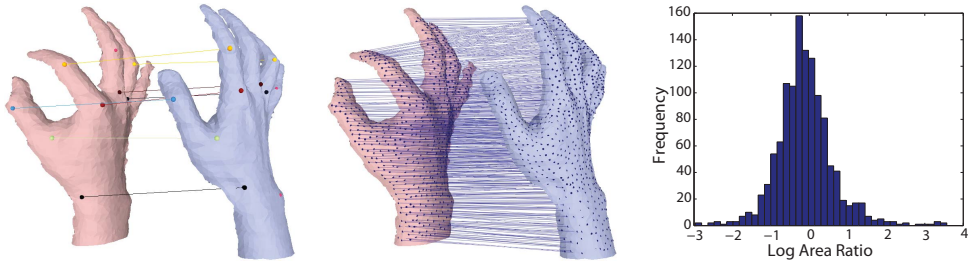
(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.11: Matching result of face example.



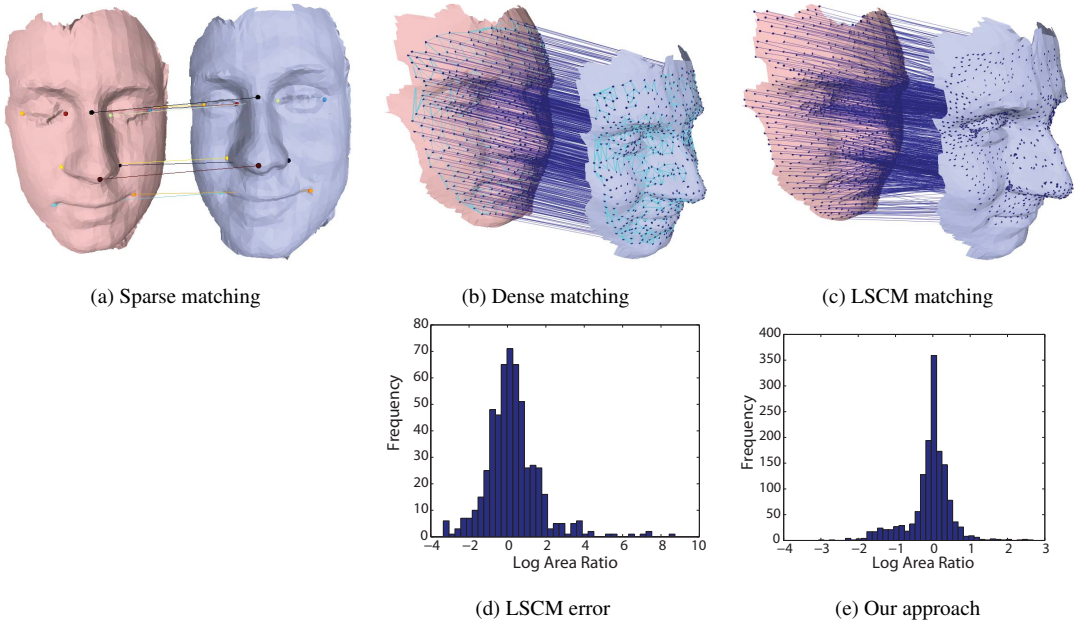
(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.12: Matching result of face with holes.



(a) Sparse registration (b) Dense registration (b) Matching area ratio

Figure 4.13: Matching result of hand example.



(a) Sparse matching (b) Dense matching (c) LSCM matching

(d) LSCM error (e) Our approach

Figure 4.14: Comparison with LSCM approach [269] for dense surface matching. (matched/total = $1455/1635$) (best viewed in color). Notice the high number of flipped triangles in (c)

Chapter 5

Intrinsic Dense 3D Surface Tracking

> “Time is not important; only life is important.”
>
> – “The Fifth Element”

5.1 Introduction

Dynamic 3D data has become increasingly popular with the advances in 3D reconstruction techniques [3, 99, 152, 215, 271, 297]. An important prerequisite for most applications is to register the 3D data among frames. For applications such as facial expression analysis/transfer [260], dense and accurate registration is highly desired for capturing subtle details. However, achieving dense, accurate registration remains challenging when there is noise, large deformations and lack of reliable features. In this paper, we address the challenging problem of tracking a deformable template from dynamic, markerless 3D data.

According to the well-known Riemann uniformization theorem [73], any simply-connected surface with a Riemannian metric can be conformally deformed onto one of three canonical spaces: the sphere, the plane and the hyperbolic disk. By using uniformization, 3D geometric problems are naturally converted to 2D ones, which in general simplifies computation. Most importantly, when the deformations between surfaces are isometric, matching between two surfaces can be greatly simplified in the uniformization domain by only searching for a few correspondences [155].

Previously, in order to match two surfaces in the uniformization domain, the work of

[269, 270, 287, 289] relied on consistent feature boundaries or points to determine prescribed sparse correspondences. Once the sparse correspondences are established, a single conformal energy is minimized to match between the whole surfaces. To get rid of the reliance on consistent feature points, Lipman *et. al.* [156] observed that when two surfaces are isometrically deformed, only three correspondences are needed to determine a unique conformal mapping which is described by a Möbius transformation. Sparse correspondences are optimized by voting from different Möbius transformations induced by different combinations of correspondences. Recently, based on the fact that every three correspondences determine a unique conformal mapping, *i.e.*, a candidate matching point for every point on the surface, Zeng *et al.* [292] formulated high-order graph matching problem to search for the optimal dense matching result based on textural and geometric similarities. Despite its success in combining multiple matching criteria to handle more than isometric deformations, the textural and geometric similarity measures for each matching candidate are only point-wise and hence sensitive to noise. Most recently, Lipman *et. al.*. [155] proposed a new distance function that compares two neighborhoods (*i.e.*, disks) between two points in the disk uniformization domain, which improves the robustness of matching cost for each candidate correspondence. Nevertheless, this distance cannot handle general surface matching when the surfaces have inconsistent boundaries or are anisometrically deformed. Comparing two neighborhoods directly in the uniformization domain is not straightforward since a disk is no longer mapped to a disk under Möbius transformations [179].

In this chapter, we define a new distance that compares the neighborhoods between any candidate matching pair when the two surfaces have inconsistent boundaries and are not isometrically deformed. Since every three correspondences determine a unique conformal mapping between the two surfaces, we can define a matching cost based on feature differences for every such possible match. However, globally searching for the best three correspondences that match the two surfaces is limited because the two surfaces can be matched only when they are isometrically deformed and have consistent boundaries. Hence, we define a cost function for a particular correspondence by the lowest feature differences across the set of transformations that cause the two points to match, which only involves searching for the correspondences of another two points on the surface. By restricting the comparison of feature differences only between the neighborhoods of the correspondence, we can handle surfaces with inconsistent boundaries or anisometric deformations. A matching cost between two neighborhoods can be therefore efficiently computed since only one conformal mapping is needed for one surface and the other conformal mappings induced from different correspondence matches are computed in a closed form.

With the above mentioned matching cost for any candidate correspondence, it is not enough to simply output such locally best match for each point due to multiple optima and numerical

errors. Therefore, regularization is necessary for a plausible result. In this paper, we formulate the surface tracking problem in a unified probabilistic inference framework that takes into account spatio-temporal consistency as well as the possibility for drift error. We show that such an inference problem can be approximated by standard MRF optimization in a discrete setting and occlusion can be appropriately handled. Combinatorial methods based on graphical models have become popular due to their capabilities of solving for more complicated deformations [85, 87, 220] and avoiding local optimal solutions [126, 135]. Besides, occlusion handling can be conveniently modeled in the same framework [236].

In summary, the primary contributions of this chapter are a robust intrinsic distance function for measuring the cost of matching two points and a unified framework for intrinsic 3D surface tracking. To achieve a robust 3D tracking system, our framework includes an intrinsic spatial deformation prior that constrains consistency in local deformations among neighboring points as well as drift and occlusion handling. Our tracking method is performed in the uniformization domain, so it is robust to large deformations and scale changes. Compared to existing tracking algorithms such as [34, 270], we do not require prescribed feature detectors and do not rely on consistent boundaries. Therefore our method is able to handle surface tracking under challenging situations as shown in our experiments with a deforming sponge. Unlike the system of Weise *et al.* [275], we do not require a pre-defined training set for PCA learning, which is important for accurately tracking both large previously unseen variations in object deformation as well as subtle but significant differences in the case of facial expression changes. Quantitative results show that our algorithm achieves a high level of accuracy.

The remainder of this paper is organized as follows. In Sec. 5.2 related work on image registration and shape tracking methods are reviewed. The new measure of matching correspondences is defined in Sec. 5.3. In Sec. 5.4 we introduce a probabilistic 3D surface tracking framework. The implementation details are given in Sec. 5.5. Experimental results and validations are part of Sec. 5.6. Finally we conclude our work in Sec. 5.7.

5.2 Related Work

In this section, we review related work on image/surface tracking.

5.2.1 Image registration

The problem of 3D shape registration shares many similarities with the 2D image registration (image alignment) problem. For example, almost all real-world 3D shapes are two-manifolds; hence those 3D shape registration problems are inherently 2D. It is therefore important to

understand the problem of 2D registration before stepping further into the 3D case.

Mathematically, an image can be formulated as the mappings $I : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}^d$ that maps each position of the image to a d dimensional feature space (intensity or color). The problem of image registration can be formulated as finding a map $T : \mathbb{R}^2 \mapsto \mathbb{R}^2$ such that certain distance measure (or object function) between the two images, denoted by $d(\{I_1, I_2, T\})$ is minimized. In the Bayesian paradigm, the distance measure can be interpreted as the probability function $P(I_1, I_2|T) \propto d(\{I_1, I_2, T\})$.

Following this formulation, the optical flow method ([103, 160]) minimizes the following energy

$$d(\{I_1, I_2, T\}) = \int_{\Omega} |I_1(T(p)) - I_2(p)|^q dp \tag{5.1}$$

By assuming the displacement between the two images (*i.e.*, $\Delta p = T(p) - p$) to be sufficiently small, one can approximate the above energy using the Taylor expansion: $I_p(p + \Delta p) \approx I_1(p) + \Delta p I_1'(p)$. There are also other definitions of the matching distance $d(\{I_1, I_2, T\})$. For example, when we want to take into account partial alignment problem, we can add a weight to each pixel. Or when we want to take into account intensity variation, we may need to add additional parameters into the distance (*e.g.*, bias and gain) [1]. Using the Bayesian methods, since $P(T|I_1, I_2) \propto P(I_1, I_2|T)P(T)$, by adding various constraints on the plausible deformation T (*i.e.*, $P(T)$), various solutions can be derived.

In the energy function $d(\{I_1, I_2, T\})$, the search space of $T(\cdot)$ is often simplified in order to facilitate efficient optimization. A common assumption on the deformation T is its smoothness, which is often added into the total energy function for minimization. An alternative way to simplify the deformation space is to model the deformation using the parameter of a few discrete points, *e.g.*,

$$T(p, C) = p + B(p)C, \tag{5.2}$$

where $C = \{c_1, \ldots, c_n\}$ is a set of control points or simply certain parameters and $B(p)$ is a set of basis functions ([14, 95, 213]). One advantage of such spline-based approaches over the optical-flow methods (based on Taylor series) is that it can take into account non-linear deformations of T (*e.g.*, affine transform [222], lens distortion, water distortion [250]). Moreover, the basis function can be also learnt from data, leading to the celebrated active appearance models (AAM) [169]. Approaches based on estimating parameters learnt from data are also categorized as *generative methods*, while approaches that directly estimate the

[1] Note that there are also other global statistics based image distance, such as mutual information based [197], or Fourier-based approaches [238]

deformation $T(\cdot)$ belong to the *discriminative methods*. Intuitively, discriminative methods can be regarded as the duality of those generative methods, or as the spectral approach. The advantage of generative methods is its robustness to local noises or information missing (occlusions), while in contrast the discriminative methods are better at handling varieties among different data. Hence one promising direction is to combine both approaches to improve the robustness of generative methods (*e.g.*, [251]).

The above methods, either discriminative or generative, all formulated the image registration problem in the continuous setting. An advantage of the continuous approaches is its accuracy. However, its optimization usually relies on good initialization. There are also approaches that solves the image registration problem in the discrete or combinatorial setting, *e.g.*, [85–87, 159, 220]. The idea the combinatorial approaches is based on the assumption that the deformation of each point can be discretely sampled. Despite the possible loss of accuracy, combinatorial methods often have the advantage of flexibility. For example, it can encode matching energy from statistically learned distribution [206] or impose deformation priors without a closed-form [119]. Besides, high-order interactions of combinatorial based approaches allows us to eliminate the variances in scale [84, 141], or projections [268].

All of those above-mentioned combinatorial approaches are discriminative methods that do not impose learned data to constrain stability. As a result, they often require sufficient sampling space and/or high computationally complexity. As an alternative approach, recently, the work of [112, 148, 254] proposed to fusion the results by different continuous methods, either by discrete optimization or by continuous optimization, to achieve better registration results. However, there are still little work on the application of generative methods for combinatorial image registration [17]. Examples of applications of generative methods include the image synthesis/completion and matching [130], *etc*.

5.2.2 Shape tracking

Using the Bayesian paradigm, shape tracking can be formulated as finding the plausible transformations between consecutive frames: $P(T_{1,2}, \ldots, T_{n-1,n}|\mathcal{M}_1, \ldots, \mathcal{M}_n)$. However, since $P(T_{1,2}, \ldots|\mathcal{M}_1, \ldots) \propto P(\mathcal{M}_1, \ldots|T_{1,2}, \ldots)P(T_{1,2}, \ldots)$, the data likelihood and the prior are not just defined on the deformation between frames, so the consistencies across frames should be considered. Hence one distinction between shape tracking and registration is that tracking requires the consideration of spatio-temporal consistencies. Another distinction is the representation of the surface, since the current 3D/depth data often contain a lot of noise/holes and topological noise.

Similar to the image registration case, the 3D shape tracking methods can be classified

as either generative or discriminative. The discriminative methods find correspondences between frames, which are often useful in shape reconstruction [34, 150, 151, 153, 265, 266]. In contrast, the generative methods are often used in real-time performance or shape animations where a pre-learned deformations can help accelerate the computational time ([154, 274, 276]).

Generally speaking, discriminative methods rely less on a generic model for the tracking target while generative methods rely more on an accurate model[2]. The simplest way to treat the surface tracking is to consider each point on the surface individually (point cloud), and the matching point in the next frame can be found by using the nearest neighbor (NN) search. The nearest neighbor search can be efficiently conducted using the KD-tree data structure [60]. To improve robustness, the average multiple nearest neighbors (kNN) can be considered to be the matching candidate [264]. However, such a simple scheme can only be applied to very limited simple cases.

In order to improve the tracking accuracy, one can either impose local or global motion constraints. By assuming the deformation between two frames to be rigid, the ICP method [26] can be applied for shape tracking and it can also handle occlusions well. Another popular global deformation constrain is isometry [270]. However, in such case, a two manifold surface must be constructed.

Whereas global deformation models can achieve closed-form solution for the deformation of each point, they are unable to model the subtle changes at each point. In contrast, local models are better at dealing with varieties among individual points. The motion consistency is often imposed among neighboring points. For example, the finite element based approaches [170] represent the surfaces as a set of connected triangular elements. Under such a representation, one can model the physics-based properties, *e.g.*, stiffness, dampness, kinetic energy, into the deformations of the surface. Another popular representation of the surface is to use free-form surfaces [75, 208], splines [239], or level sets [217].

5.3 A robust correspondence matching distance

We assume a shape is represented in a metric feature space $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$, where $\mathcal{M}$ is a compact connected and complete Riemannnian surface, $d_{\mathcal{M}} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a measure of distances between pairs of points on $\mathcal{M}$ and $f_{\mathcal{M}} : \mathcal{M} \to \mathbb{R}^n$ is the mapping of each point on $\mathcal{M}$ into the feature space (such as curvatures, texture, etc). Previously a number of metrics have been proposed for measuring the similarities between any two shapes based on geometric

[2]This makes sense since generative models should have the ability to reconstruct the original input thus it should have a much more accurate model for the target than the discriminative methods.

information only, *e.g.*, the geodesics distance [37], the diffusion distance [39] and the distances based on the conformal factor [155]. To compare two shapes $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$ and $(\mathcal{N}, d_{\mathcal{N}}, f_{\mathcal{N}})$, we denote the set of possible mappings (*e.g.*, diffeomorphism) between them as $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$. A distance between any two shapes $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$ and $(\mathcal{N}, d_{\mathcal{N}}, f_{\mathcal{N}})$ can be defined as follows:

$$d^{\mathcal{T}}(\mathcal{M},\mathcal{N}) = \inf_{t\in\mathcal{T}_{\mathcal{M}\to\mathcal{N}}} \int_{\mathcal{M}} |f_{\mathcal{M}}(x) - f_{\mathcal{N}}(t(x))| dx. \tag{5.3}$$

Such a definition resembles the sum of absolute differences (SAD) metric used in motion estimation for video coding. Nevertheless, in the context of surface registration, it provides a flexible way of handling a wide range of deformations between surfaces. For example, when the feature is the conformal factor [289] and the mappings $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ are restricted to Möbius transformations in the uniformization domain, it handles isometric or near-isometric surface matching. For surfaces with more general deformations, we may use other features such as texture and curvature [292].

In the problem of dense surface matching, a global distance function such as Eq. 5.3 does not guarantee the quality of the matchings of individual points. In this paper, we consider the following distance function for mapping any point $p \in \mathcal{M}$ to any point $q \in \mathcal{N}$:

$$d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p,q) = \inf_{\substack{t\in\mathcal{T}_{\mathcal{M}\to\mathcal{N}} \\ t(p)=q}} \int_{\mathcal{M}} |f_{\mathcal{M}}(x) - f_{\mathcal{N}}(t(x))| dx, \tag{5.4}$$

which is defined by the cost of matching the two surfaces by fixing the particular correspondence. When there is no mapping in the group $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ that maps p to q, we define the distance to be infinite.

First of all, it is easy to see that $d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p,q) \geq d^{\mathcal{T}}(\mathcal{M},\mathcal{N})$ for any $p \in \mathcal{M}, q \in \mathcal{N}$ and the lower bound is achieved when q is the mapping of p that minimizes the energy of Eq. 5.3. In the problem of dense surface registration where we want to find the correspondences of a point set $P = \{p_i | p_i \in \mathcal{M}, i = 1, \ldots, n\}$, since we have

$$d^{\mathcal{T}}(\mathcal{M},\mathcal{N}) \leq \frac{\sum_{p\in P} d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p, t(p))}{|P|}, \forall t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}}, \tag{5.5}$$

the problem of shape registration can be formulated as

$$\inf_{t\in\mathcal{T}_{\mathcal{M}\to\mathcal{N}}} \sum_{p\in P} d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p, t(p)). \tag{5.6}$$

Hence, if we can sufficiently sample a small set of matching candidates in $t(p)$ for each $p \in P$ and evaluate the distance function $d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ efficiently, the problem of finding the

correspondences of the set P can be consequently solved. In the following, we show how the distance function can be efficiently approximated in the uniformization domain. Furthermore, since the distance function of Eq. 5.4 is defined on the whole surface, when the transformation group $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ is confined to mappings with bounded area distortions, a small deviation from the true matching that minimizes Eq. 5.3 would cause the energy measure deviate significantly from the optimum, which guarantees the robustness of this distance measure.

5.3.1 Approximation in the uniformization domain

Although the distance defined in Eq. 5.4 gives us a robust way of evaluating the matching cost between points, it is in general computationally intractable to evaluate such distance function directly in the 3D embedding space since it involves searching among all possible matchings between two surfaces given a correspondence. For example, in [296] the matching cost given a few sparse correspondences is measured by deforming the whole surface to the target based on certain deformation energy which is only able to establish up to 10 correspondences. In this paper, we propose an efficient way for approximating the distance function of Eq. 5.4 by considering a mapping set $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ defined in the uniformization domain.

In order to take into account general mappings between two surfaces with inconsistent boundaries, we consider a neighborhood $N(p)$ of p and the points on its boundary $\partial N(p) = \{p_1, \ldots, p_r\}$. For each possible mapping of the neighboring points $p_1, \ldots, p_r \in \partial N(p)$, the distance function of Eq. 5.4 can be approximated by warping the neighborhood to that of the target. Directly computing such warping and comparing the feature difference for each warping is very costly. However, if we note that a mapping between the two surfaces can be computed by specifying a few feature correspondences and solving an interpolation conformal energy [269, 270, 289], we then introduce an efficient way to approximate the distance function of Eq. 5.4. This motivates us to consider the mappings of the neighborhood $N(p)$ in the uniformization domain.

Formally, we denote the uniformization (conformal mapping) of any surface $\mathcal{M}$ as $\phi_{\mathcal{M}} : \mathcal{M} \to \mathcal{U} \subset \mathbb{C}$. Also we consider the set of mappings $\mathcal{T}^{\text{UNI}}$ that is induced by specifying three correspondences between two surfaces in the uniformization domain [156]. For any point $p \in \mathcal{M}$, we define the image of a point p as $\text{Img}(p) = \{t(p) | t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}}\}$, where $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ can be arbitrary diffeomorphisms and we only require that $\mathcal{T}^{\text{UNI}} \subset \mathcal{T}_{\mathcal{M}\to\mathcal{N}}$. For any two points $p_1, p_2 \in \partial N(p)$, $q \in \text{Img}(p)$, $q_1 \in \text{Img}(p_1)$ and $q_2 \in \text{Img}(p_2)$, let us denote by $\text{Mo} : pp_1p_2 \to qq_1q_2$ as the Möbius transformation that maps (p, p_1, p_2) to (q, q_1, q_2) on $\mathcal{U}$. We

(a) First frame (b) Second frame

Figure 5.1: Two frames used for the evaluation of cost function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$.

then *approximate* the distance of Eq. 5.4 in the uniformization domain as follows:

$$d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p,q) = \inf_{\substack{p_1,p_2\in\partial N(p),q_1\in\text{Img}(p_1),q_2\in\text{Img}(p_2),\\ \text{Mo}:pp_1p_2\rightarrow qq_1q_2}} \frac{\int_{\phi_{\mathcal{M}}(N(p))\subset\mathcal{U}} |f_{\mathcal{M}}(\phi^{-1}_{\mathcal{M}}(z)) - f_{\mathcal{N}}(\phi^{-1}_{\mathcal{N}}(\text{Mo}(z)))|dz}{Area(N(p))}. \tag{5.7}$$

Here $\phi_{\mathcal{M}}(N(p))$ denotes the mapping of the neighborhood $N(p)$ to the uniformization domain and $Area(N(p))$ denotes the area of the neighborhood. When the feature is only based on geometry, (*e.g.*, the conformal factor [155] or the gaussian curvature), the distance measures the deviation from isometric deformation. However, when the feature is based on other measures such as texture, the distance measures the deviation from other deformations which are not necessarily isometric. Therefore, our definition is general and subsumes the isometric deformation as a special case.

Fig. 5.2, 5.3, 5.5 and 5.4 are examples for the evaluation of the distance function defined above. Here we select two frames with large deformations (Fig. 5.1). Since the input depth data is on the regular $m \times n$ image domain, for each point p on the first mesh, we first choose its closest point q on the second mesh and then the matching candidate is selected in a 7×7 neighborhood of a point q. The distance function is evaluated for each of the 49 matching candidates. We also compare our distance functions with other features.

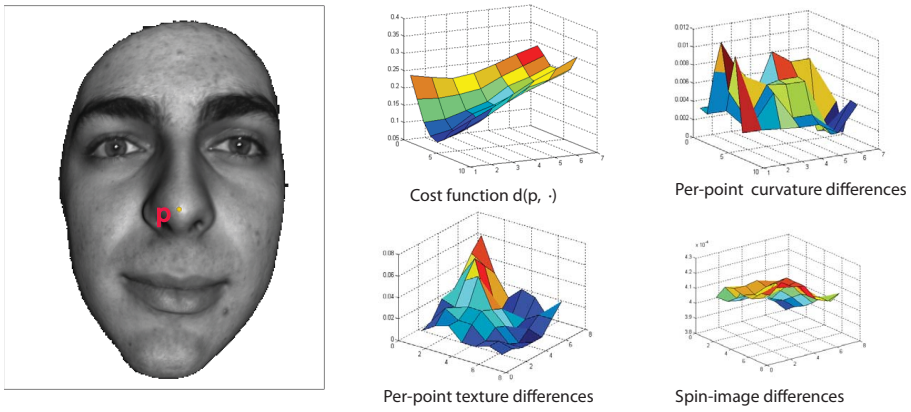


Figure 5.2: Evaluation of cost function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for the nose part.

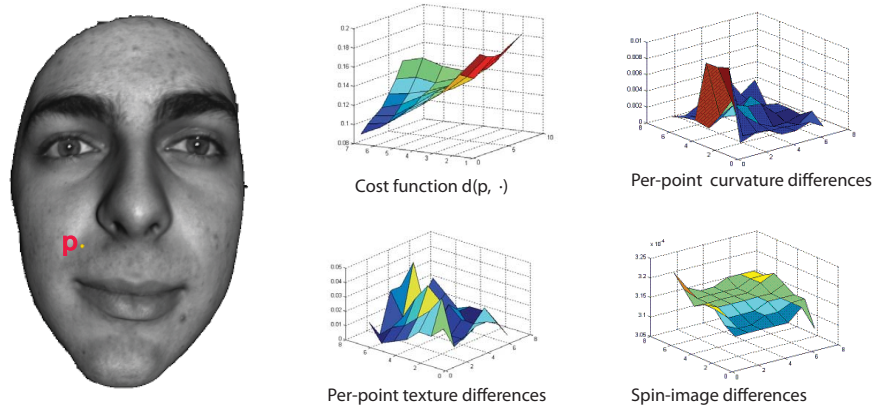


Figure 5.3: Evaluation of cost function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for the cheek part.

5.4 A probabilistic framework for surface tracking

The intrinsic distance $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ measures the likelihood of matching between individual points. However, the set of points $q \in t(p), t \in \mathcal{T}$ that achieves the minimal of $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p,q)$ is not necessarily unique. Also in the approximation of $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ numerical error is unavoidable. Therefore, regularization is necessary for a plausible result. Hence, we formulate the surface registration problem as

$$\inf_{t\in\mathcal{T}_{\mathcal{M}\to\mathcal{N}}} \sum_{p\in P} d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p,t(p)) + \mathcal{R}(t(P)), \tag{5.8}$$

where $\mathcal{R}(t(P))$ is the regularization on the registration results. In this section, we investigate the 3D surface tracking problem in a probabilistic framework which takes into account

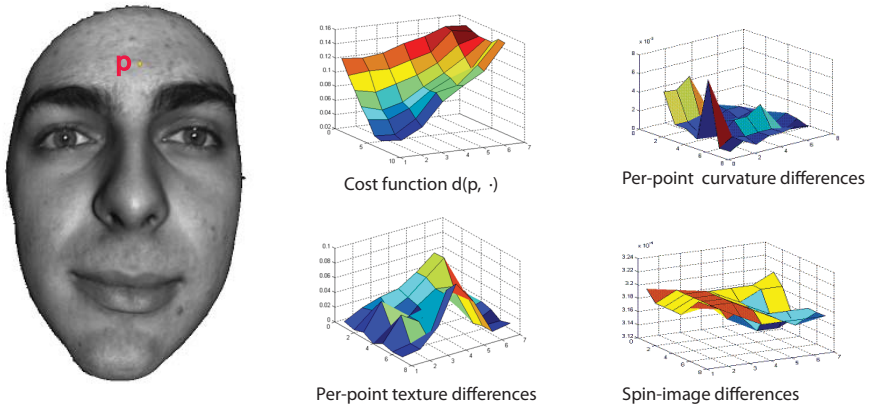


Figure 5.4: Evaluation of cost function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for the forehead part.

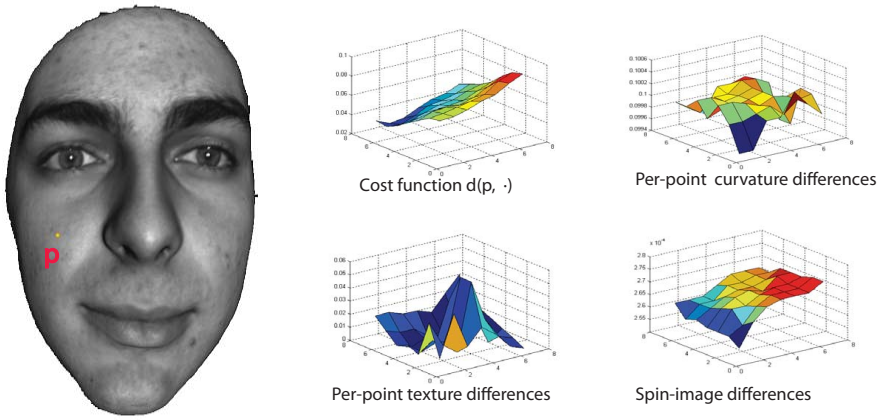


Figure 5.5: Evaluation of cost function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for the left cheek part.

geometric and textural similarities, spatio-temporal consistencies and handles error drift.

Let us denote by $\mathcal{M}^{1:t} \equiv \{\mathcal{M}^1, \ldots, \mathcal{M}^t\}$ the dynamic 3D data up to time t, and $\mathbf{x}^{1:t} \equiv \{\mathbf{x}^i \in \mathcal{M}^i | i = 1, \ldots, t\}$ as the trajectory of the initial given dense points $\mathbf{x}^0 = \{x_i^0 \in \mathbb{R}^3 | i = 1, \ldots, n\}$ where $\mathbf{x}^t = \{x_i^t \in \mathbb{R}^3 | i = 1, \ldots, n\}$. In order to simplify the problem and utilize the intrinsic measure defined in the previous section, we assume the initial points $\mathbf{x}^0$ are represented as a triangular mesh, *i.e.*, a planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The task of tracking the trajectory of $\mathbf{x}^0$ at time t given the dynamic data $\mathcal{M}^{1:t}$ and previous trajectory $\mathbf{x}^{1:t-1}$ therefore becomes the MAP problem

$$\arg\max_{\mathbf{x}^t} p(\mathbf{x}^t | \mathcal{M}^{1:t}, \mathbf{x}^{1:t-1}) \tag{5.9}$$

Here we assume $\mathcal{M}^{1:t'}$ is independent of $\mathbf{x}^t$ given $\mathbf{x}^{1:t'}$ whenever $t' < t$. From the Bayes' theorem, we have

$$\begin{aligned}
&p(\mathbf{x}^t|\mathcal{M}^{1:t}, \mathbf{x}^{1:t-1}) \\
&= \frac{p(\mathcal{M}^{1:t}|\mathbf{x}^{1:t})p(\mathbf{x}^t|\mathbf{x}^{1:t-1})}{p(\mathcal{M}^{1:t}|\mathbf{x}^{1:t-1})} \\
&\propto p(\mathcal{M}^{1:t}|\mathbf{x}^{1:t})p(\mathbf{x}^t|\mathbf{x}^{1:t-1}) \\
&= p(\mathcal{M}^t|\mathbf{x}^{1:t}, \mathcal{M}^{1:t-1})p(\mathcal{M}^{1:t-1}|\mathbf{x}^{1:t})p(\mathbf{x}^t|\mathbf{x}^{1:t-1}) \\
&\propto \underbrace{p(\mathcal{M}^t|\mathbf{x}^{1:t}, \mathcal{M}^{1:t-1})}_{\text{Data fidelity}} \underbrace{p(\mathbf{x}^t|\mathbf{x}^{1:t-1})}_{\text{Spatio-temporal prior}} .
\end{aligned} \tag{5.10}$$

Here $p(\mathcal{M}^t|\mathbf{x}^{1:t}, \mathcal{M}^{1:t-1})$ denotes the data likelihood defined by intrinsic similarities. $p(\mathbf{x}^t|\mathbf{x}^{1:t-1})$ denotes the spatio-temporal priors that takes into account the smoothness of the result. In the following we discuss each of the two terms in detail.

5.4.1 Data fidelity terms

The data fidelity terms considers the fidelity of the $3D$ data $\mathcal{M}^{1:t}$ given the tracking results $\mathbf{x}^{1:t}$. For dense tracking, we assume the tracking points $\mathbf{x}^0$ are dense enough to capture the detailed geometry of the surfaces. Hence the trajectory $\mathbf{x}^{1:t'}$ and the data $\mathcal{M}^{1:t'}$ are independent of the trajectory $\mathbf{x}^t$ given $\mathbf{x}^{t-1}$ and $\mathcal{M}^{t-1}$ when $t' < t-1$. We have

$$\begin{aligned}
&p(\mathcal{M}^t|\mathbf{x}^{1:t}, \mathcal{M}^{1:t-1}) \\
&= \frac{p(\mathcal{M}^t, \mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})}{p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})} \\
&\propto p(\mathcal{M}^t, \mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}) \\
&= p(\mathcal{M}^t|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})\times \\
&\quad p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, \mathcal{M}^t).
\end{aligned}$$

Here $p(M^t|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})$ denotes the registration between successive frames and

$$p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, \mathcal{M}^t)$$

denotes the consistency between current frame and the history, which avoids loss of tracking caused by accumulation of local registration errors.

Geometry and texture similarities

Intrinsic comparison takes into account both geometry and texture (if available) consistencies between frames. We define the inter-frame data similarity term as follows:

$$p(\mathcal{M}^t|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}) \propto \prod_{i=1}^{n} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t-1}}(x_i^t, x_i^{t-1})|0, \sigma_{data}). \tag{5.11}$$

Error drift term

The intrinsic distance $d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'})$, allows us to compare the similarity of a point in frames t and t'. We assume that $\mathbf{x}^t$ is consistent with the history if it "agrees" with the majority of the history. This is in a similar spirit with the median filter for video stabilization, and can be formulated as

$$p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, M^t) \tag{5.12}$$
$$\propto \prod_{i=1}^{n} \text{median}_{t' \in \{1,\ldots,t-2\}} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'})|0, \sigma_{drift})$$

However, computing the consistency between current frame to all the previous frames is very costly. An approximate sampling scheme is to consider a subset of $\{1, \ldots, t-2\}$, namely, $\mathcal{I}$. We can approximate Eq. 5.12 as follows:

$$\prod_{i=1}^{n} \text{median}_{t' \in \mathcal{I}} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'})|0, \sigma_d). \tag{5.13}$$

5.4.2 Spatio-temporal priors

The probability $p(\mathbf{x}^t|\mathbf{x}^{1:t-1})$ represents the prior knowledge of the trajectory $\mathbf{x}^{1:t}$, which also regularizes the tracking result. First of all, we decompose the probability into two terms:

$$p(\mathbf{x}^t|\mathbf{x}^{1:t-1}) = \frac{p(\mathbf{x}^t|\mathbf{x}^{t-1})p(\mathbf{x}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1})}{p(\mathbf{x}^{1:t-2}|\mathbf{x}^{t-1})} \propto p(\mathbf{x}^t|\mathbf{x}^{t-1})p(\mathbf{x}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1}). \tag{5.14}$$

Here $p(\mathbf{x}^t|\mathbf{x}^{t-1})$ denotes the spatial deformation consistency between consecutive frames and $p(\mathbf{x}^{1:t-2}|\mathbf{x}^t, \mathbf{x}^{t-1})$ denotes the dynamic inter-frame consistency.

Intrinsic spatial deformation prior

The spatial prior $p(\mathbf{x}^t|\mathbf{x}^{t-1})$ takes into account the plausible deformation of the surface between frames. For example, when the deformation of the surface is near-isometric, we can constrain that for every edge $(i,j) \in \mathcal{E}$, the length $|x_i^t - x_j^t|$ is close to $|x_i^{t-1} - x_j^{t-1}|$.

In the definition of the distance in Eq. 5.4, only matching cost between each correspondence $x_i^{t-1} \mapsto x_i^t$ is considered. There is no account for consistency between two neighboring correspondences $x_i^{t-1} \mapsto x_i^t$ and $x_j^{t-1} \mapsto x_j^t$ where $(i,j) \in \mathcal{E}$. Since each of the distance function Eq. 5.7 takes in to account the locally best Möbius transformation mapping a neighborhood of x_i^{t-1} to a neighborhood of x_i^t, it is reasonable to assume that such locally optimal transformation also map its neighbor x_j^{t-1} to a position nearby x_j^t. Let $\mathrm{Mo}_{p,q}^{opt}$ denote the optimal Möbius transformation that achieves the distance defined in Eq. 5.7. To constrain the deformation consistency between neighboring points $(i,j) \in \mathcal{E}$, we define the following distance in the uniformization domain:

$$d_{i\to j}^t = |\mathrm{Mo}_{x_i^{t-1},x_i^t}^{opt}(\phi^{t-1}(x_j^{t-1})) - \phi^t(x_j^t)|. \tag{5.15}$$

Here ϕ^t denotes the uniformization of the data $\mathcal{M}^t$. This distance measures how close the optimal transformation that map $\phi^{t-1}(x_i^{t-1})$ to $\phi^t(x_i^t)$ also transform $\phi^{t-1}(x_j^{t-1})$ to a nearby point of $\phi^t(x_j^t)$. When this distance is small, it means such optimal transformation also agrees with the neighbors. Formally, we define,

$$p(\mathbf{x}^t|\mathbf{x}^{t-1}) \propto \prod_{(i,j)\in\mathcal{E}} \mathcal{N}((d_{i\to j}^t + d_{j\to i}^t)/2|0,\sigma_{spa}).$$

Dynamic motion prior

Dynamic prior imposes temporal consistency of each vertex i by assuming the curve traced by each vertex i to be smooth, *i.e.*, we assume the acceleration to be small. If we define the angle between the vectors $x_i^t - x_i^{t-1}$ and $x_i^{t-1} - x_i^{t-2}$ as Ang_i^t, the dynamic prior can be defined as

$$p(\mathbf{x}^{1:t-2}|\mathbf{x}^t,\mathbf{x}^{t-1}) \propto \prod_{i=1}^{n} \mathcal{N}(\mathrm{Ang}_i^{t-1}|\mathrm{Ang}_i^t,\sigma_{dyn}). \tag{5.16}$$

In practice, the smoothness assumption is only applicable when the motion between two frames is not too large.

Algorithm 1: Intrinsic 3D surface tracking algorithm.

Input : Scanned 3D data.
Output : Registered template mesh for each frame t.

Initialization: Construct a template mesh from the 3D data in the first frame (Sec. 5.5.1).
Foreach new frame $t+1$

(i) Perform candidate selection (Sec. 5.5.2).

(ii) Compute cost functions for every point (Sec. 5.5.3).

(iii) Solve the composite MRFs (Sec. 5.5.5).

(iv) Repeat 2. until convergence or within a number of iterations.

5.5 Implementation details

The optimization of the above mentioned objective function (Eq. 5.10) is too complicated to be solved in the continuous space. In this paper, we employ the discrete MRF framework [83] to take into account the objectives discussed in the previous section. For each frame at time t, we select L matching candidates for each point $x_i^t, i = 1, \ldots, n$. As a result, our tracking problem becomes solving the best configurations $\mathbf{x}^t \in L^n$:

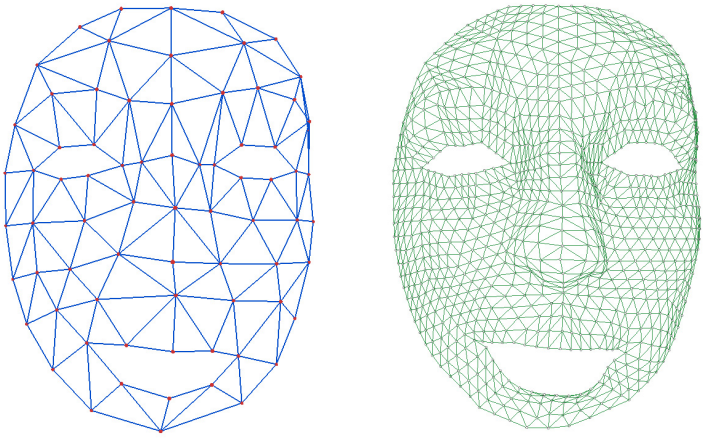
$$\arg\min_{\mathbf{x}^t} \sum_{i \in \mathcal{V}} \theta_i(x_i^t) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i^t, x_j^t), \tag{5.17}$$

where the energy functions $\theta_i(x_i^t)$, $\theta_{ij}(x_i^t, x_j^t)$ are defined according to probabilistic framework discussed in Sec. 5.4. In this section, we describe the definition of the MRF energy in details as described in Alg. 1.

5.5.1 Initialization

In the first frame $\mathcal{M}^0$, an initial mesh template $\mathbf{x}^0 = \{x_1^0, \ldots, x_n^0\}$ is constructed. The template can be constructed either automatically [277] or manually [34]. In this paper, we construct the template using the re-topology tool provided in MeshLab [3], which allows us to specify only a few base vertices and their connections then the dense mesh is automatically constructed. In our experiment, we normally specify less than 100 base vertices and obtain a dense mesh with around 1000 vertices (Fig. 5.6).

[3] http://meshlab.sourceforge.net/



(a) Base mesh (b) Dense mesh

Figure 5.6: Mesh template construction. A dense mesh (b) can be conveniently constructed from a base mesh (a) using the re-topology tool in MeshLab.

5.5.2 Candidate selection

To decide the matching candidates for each node x_i^t in the next frame $t+1$, we consider *embedding space* neighborhood, *view space* neighborhood and *intrinsic space* neighborhood. (1) For embedding space neighborhood, we uniformly sample L_1 points on $\mathcal{M}^{t+1}$ in the neighborhood of each x_i^t within radius R. (2) For view space neighborhood, we project the point x_i^t to an image plane where x_i^t is visible. Then we back-project L_2 neighboring points of the the projection of x_i^t on the image plane back to the surface $\mathcal{M}^{t+1}$. For 3D data obtained from the depth map of 2D images, the selection of the neighbors on the image plane is done in a hierarchical manner in order to take into account large deformations as in [86]. (3) For intrinsic space neighborhood, we randomly select L_3 triplets of initial correspondences (from closest point registration or results from previous iteration) among the vertices of base mesh and obtain a correspondence for each point from every triplets of correspondences [292]. Such intrinsic space sampling can achieve sub-sample accuracy [292]. In our experiments, we consider $L = 64$ candidates for each points, $L_1 = 24$ from embedding space sampling, $L_2 = 25$ from view space sampling and $L_3 = 15$ from intrinsic space sampling. R is set to be $\frac{1}{10}$ of the diameter of the mesh $\mathcal{M}^0$.

5.5.3 Computation of intrinsic distance

The computation of the intrinsic distance $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p,q)$ for any correspondence $p \mapsto q$ involves selection of the neighborhood $N(p)$, sampling of points $p_1, \ldots, p_r \in \partial N(p)$ and the numerical

approximation of the integration in Eq. 5.7 from the $N(p)$ to all its possible mappings on the target surface in the uniformization domain. We select the boundary points $p_1, \ldots, p_r$ among the vertices of the base mesh (Fig. 5.6(a)). If p is not on the boundary of the template and $p_a p_b p_c$ is the face of the base mesh that covers p, we choose $N(p)$ to be the largest triangle among the three triangles $\triangle p p_a p_b$, $\triangle p p_b p_c$ and $\triangle p p_c p_a$ and we evaluate the integration of Eq. 5.7 only in the region covered by the selected triangle. If p is on the boundary, we choose $N(p)$ to be the two largest triangles among its neighboring triangles mentioned above. In our implementation, we select 81 sampling points in each neighborhood. The distance function can be efficiently computed by the CUDA architecture [116] because each of the functions $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p, \cdot)$ is independent of each other.

Hardware acceleration: The CUDA architecture [116] allows us to efficiently compute the intrinsic distance $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p, q)$ in a highly parallel level. Because the computation of the integration in Eq. 5.7 for each mapping $pp_1p_2 \rightarrow qq_1q_2$ only involves the uniformization coordinates of the six points and the feature mapping from the surface to the uniformization domain, we store the mapping from the uniformization space to the feature space: $f_{\mathcal{M}} \circ \phi^{-1}_{\mathcal{M}} : \mathcal{U} \rightarrow f$, using a texture $\text{tex}_{\mathcal{M}}$ of size 1024×1024 for each input mesh.

The computation of the L distance functions $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p, \cdot)$ is distributed into an $L \times L \times L$ grid where the each dimension denotes the L possible mappings of p, p_1 and p_2. In the CUDA program, the input to each kernel function includes the uniformization coordinates of the points p, p_1, p_2, q, q_1, q_2 and the feature texture mappings $T_{\mathcal{M}}$ and $T_{\mathcal{N}}$ of each surface. It evaluates the integration in Eq. 5.7 and keep track of the current minima in an array of size L as the output. To estimate the integration in Eq. 5.7 in each kernel, we first compute the Möbius transformation $Mo : pp_1p_2 \rightarrow qq_1q_2$ and sample 81 points inside $\triangle pp_1p_2$ on $T_{\mathcal{M}}$. For each of the sampled point, we compute its texture coordinate on $\text{tex}_{\mathcal{N}}$ from Mo and compute the texture differences. The improvement using CUDA computation is significant, a CPU implementation with only $L = 25$ labels takes $300ms$ to compute all the cost functions for one point while it takes only $3ms$ for GPU implementation with $L = 64$ labels. This makes our whole tracking algorithm practical.

5.5.4 Occlusion handling

Occlusion can be handled by introducing an extra label $\{Occ\}$ for each vertex x_i^t. We define the following cost function:

$$d^{\text{occ}}(x_i^t, x_i^{t'}) = \begin{cases} 0 & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) < \delta, x_i^t \neq Occ \\ E_1 & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) > \delta, x_i^t = Occ \\ E_2 & \text{otherwise} \end{cases}.$$

Here t' is the last frame before time t that the point i is visible and we set $E_1 = 1$, $E_2 = 10$ and $\delta = 0.05$. Intuitively, when the cost of matching a correspondence $x_i^{t'} \mapsto x_i^t$ is higher than a threshold δ, it is likely that i is occluded at frame t. In such case, we give it a a penalty E_1 to discourage too many occluded points. When occlusion occurs at point i, we set its default position as the position computed by the ICP algorithm [26] registering the un-occluded points.

5.5.5 Composite MRFs and optimization

With all the energy functions defined above, we solve for the MAP problem of Eq. 5.9 under the discrete MRF optimization framework. The singleton terms include the data-fidelity, dynamic motion consistency, error drift and occlusion handling *i.e.*, for all $i \in \mathcal{V}$, we define

$$\theta_i(x_i^t) = \frac{d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t-1}}(x_i^t, x_i^{t-1})^2}{\sigma_{data}} + \frac{(\text{Ang}_i^{t-1} - \text{Ang}_i^t)^2}{\sigma_{dyn}} + \frac{(\text{median}_{t' \in \mathcal{T}} d^{\text{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}))^2}{\sigma_{drift}} + d^{\text{occ}}(x_i^t, x_i^{t'}). \tag{5.18}$$

The pairwise terms include the spatial deformation prior defined in Sec. 5.4.2 and the smoothness of the occluded part, *i.e.*

$$\theta_{ij}(x_i^t, x_j^t) = \begin{cases} \frac{(d^t_{i\to j} + d^t_{j\to i})^2}{4\sigma_{spa}} & \text{if } x_i^t, x_j^t \neq Occ \\ 0 & \text{if } x_i^t = x_j^t = Occ \\ E_3 & \text{otherwise} \end{cases} \tag{5.19}$$

In our experiment, E_3 is set to be 1. Intuitively, such energy encourage the smoothness of the occluded part. We employ the TRW-S algorithm [126] for the optimization. The energy is solved iteratively until convergence or exceed an allowed number of iterations. For the drift handling term of Eq. 5.13, we randomly select 5 frames from previous tracking results $\{1, \ldots, t-1\}$. The weights of the energy is selected as $\sigma_{data} = 1$, $\sigma_{dyn} = 500$, $\sigma_{drift} = 2$, $\sigma_{spa} = 20$.

5.6 Experimental results

Data: We test our tracking system on a dynamic face data set captured by the 3D scanning system described in [271]. The data set consists of four actors with 24 different facial expressions, including coy flirtation, devious smirk, soft affection and fake smile, etc. Each of the expression is captured with a frame rate of $24fps$ for around $10 - 20$ seconds. The number of vertices for each frame is $79,000$ on average with only gray-scale textural information (the gray level is normalized in $[0, 1]$). Some of the captured raw 3D data suffer from sudden scale change. In such case, we remove the dynamic prior defined in Sec. 5.4.2. In our experiments, we use texture as the feature for computing the intrinsic distance (Eq. 5.7); Fig. 5.7 shows four different sequences from four different actors. Fig. 5.10 shows a tracking result in a very challenging situation (largely inconsistent boundaries, occlusions and anisometric deformations between frames). For this example, the average texture difference between every frame and the first frame is 0.0235. The maximal average area ratio change (to the first frame) is 1.26 and the maximal percentage of occlusion occurred in one frame is around 30%.

Analysis of intrinsic distance function: A key factor to achieve high accuracy of surface tracking is the distance function $d^{UNI}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ defined in Sec. 5.3. To see the capability of the distance to distinguish subtle differences in correspondences for a given point p, we sample 7×7 closest neighboring points in the next frame as matching candidates in the embedding space (Sec. 5.5.2). Fig. 5.13(a) shows the evaluation of the 49 values of the function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for different points on the surface and we compare the distance with simple per-point texture difference.

Furthermore, we compare our cost function with the result obtained by the optical flow algorithm in [158] based on [42]. We project the left part of the face to a 640×480 perspective view selected to maximize visibility and apply the optical flow algorithm to establish correspondences between two frames. For the template points that belong to the projected part, we compute the cost function and choose the correspondence with lowest cost as the matching result. We linearly interpolate the correspondence of other points within the template that are visible. We compare the average texture per-point differences based on the correspondences obtained by optical flow and by our method (Fig. 5.13 (b) shows the comparison for one sequence). It can be seen that when the deformation between two frames is large, the optical flow degrades much more significantly than our method.

Error and performance analysis: Fig. 5.14(a) shows the error evaluation based on the complete 24 tracking results. The error measures the average per-point texture difference between every frame and the first frame. Fig. 5.14(b) shows the amount of area ratio change (anisometry) between the first frame the the current frame for $23,000$ randomly selected triangles among the tracking results.

We compare the influence of the regularization terms in the optimization of Eq. 5.17. The error is evaluated based on the average texture difference between every frame and the first frame. Fig. 5.15 (a) shows the comparisons for sequence A_coyfirtion. Even by considering the data term of Eq. 5.18 only, our method is more accurate for most frames than a previous intrinsic surface tracking method based on Harmonic maps [270]. Fig 5.15 (b) is the comparison for 5 more sequences on the average per-point texture difference.

Computational time: Our algorithm is implemented on an Intel® Core(TM)2 Duo 3.16G PC with 4G RAM and an NVIDIA® Geforce 9800GTX+ graphics card with 128 CUDA cores. The preprocessing (mesh loading, nearest neighbor search data structure construction, candidate selection) takes 2–3s. The computation of the mid-edge uniformization [196] for each mesh takes less than 1s using GPU implementation. With the hardware acceleration described in Sec. 5.5.3, the computation of the $L = 64$ cost functions from Eq. 5.7 for one tracking point takes only 3ms on average. The MRF optimization using the TRW-S algorithm [126] takes around 1–3s for the 1000 template points described in Sec. 5.5.1 with 65 (64 for matching candidate and 1 for occlusion) labels per point. Therefore tracking one frame with 5 look-back frames takes only $18 - 25$s for each iteration. In our experiments, we observe that the algorithm often converges within 5 iterations.

5.7 Conclusion

In this chapter, we proposed a new cost function that compares two neighborhoods of a correspondences by searching among all the possible mappings in the uniformization domain. This local cost function is combined with regularization terms that takes into account spatio-temporal consistency, drift and occlusion problems, into a unified 3D tracking framework. By employing existing MRF optimization technique and hardware acceleration, our algorithm becomes practical for applications where high accuracy is essential, *e.g.*, subtle expression analysis. In the near future, we would like to apply our algorithm to track more dynamic 3D database and explore applications such as facial expression analysis/transfer, etc.

Figure 5.7: Tracking results selected from our data set.

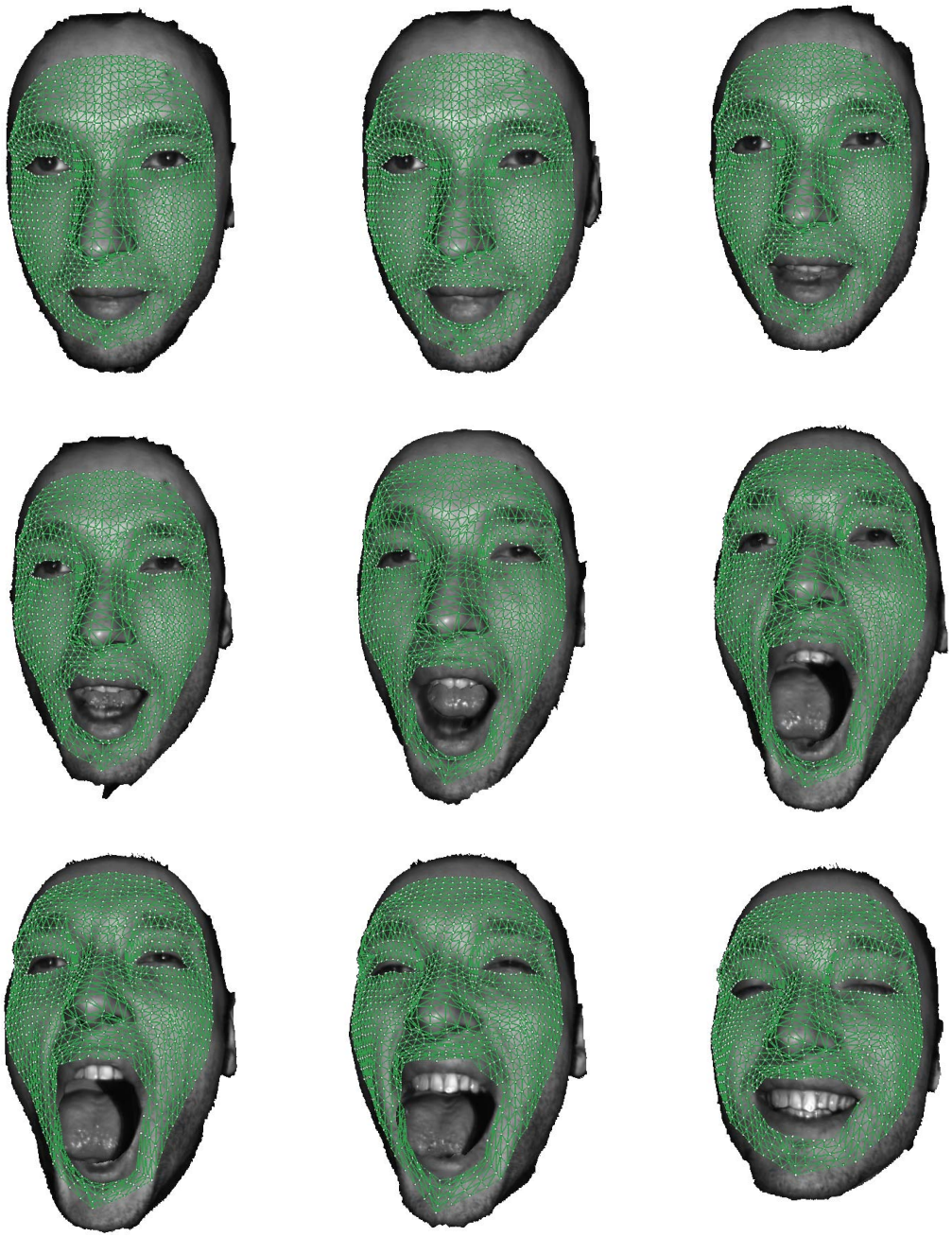


Figure 5.8: Challenging tracking results with occlusions.

Figure 5.9: Challenging tracking results with occlusions.

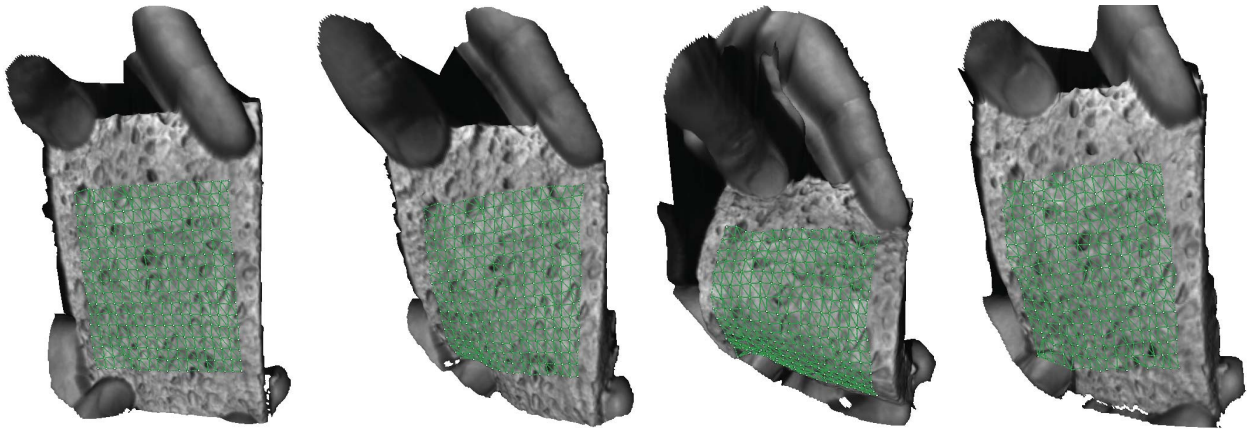


Figure 5.10: A challenging result with both anisometric deformation and inconsistent boundaries.

Figure 5.11: Sponge results with texture.

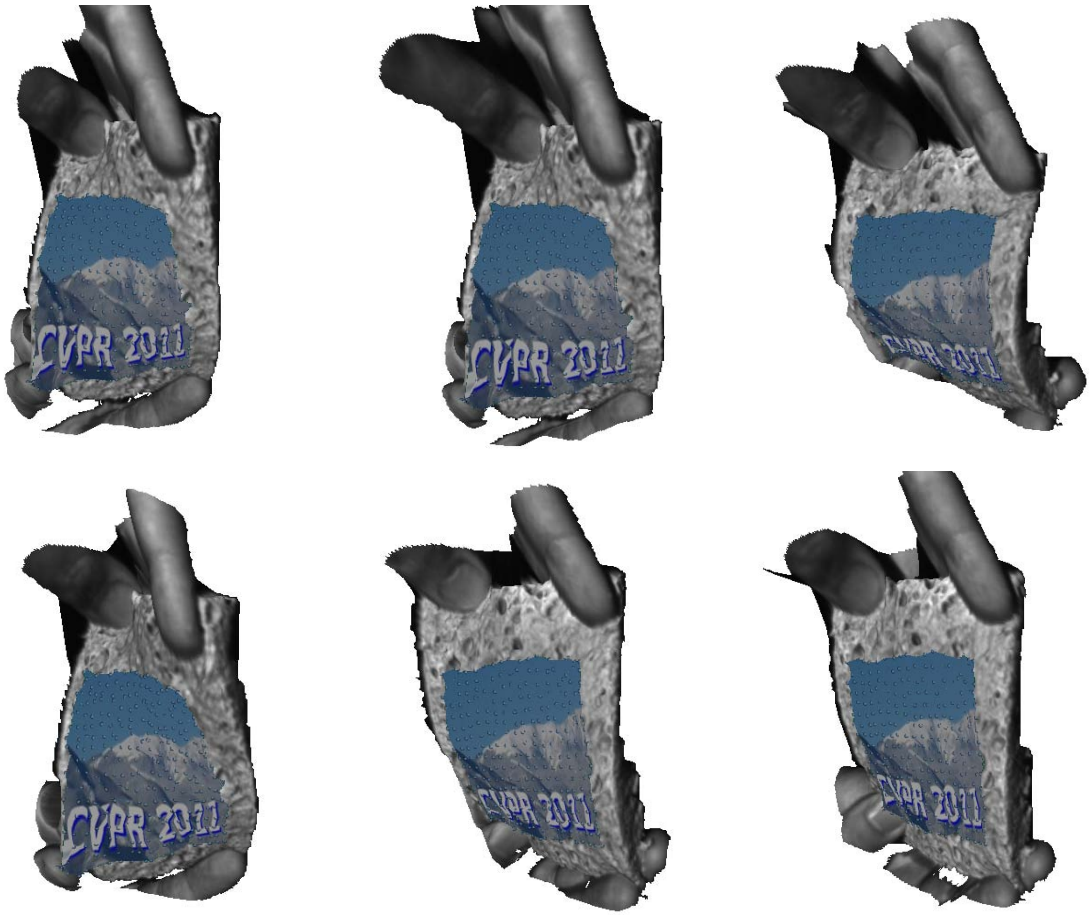
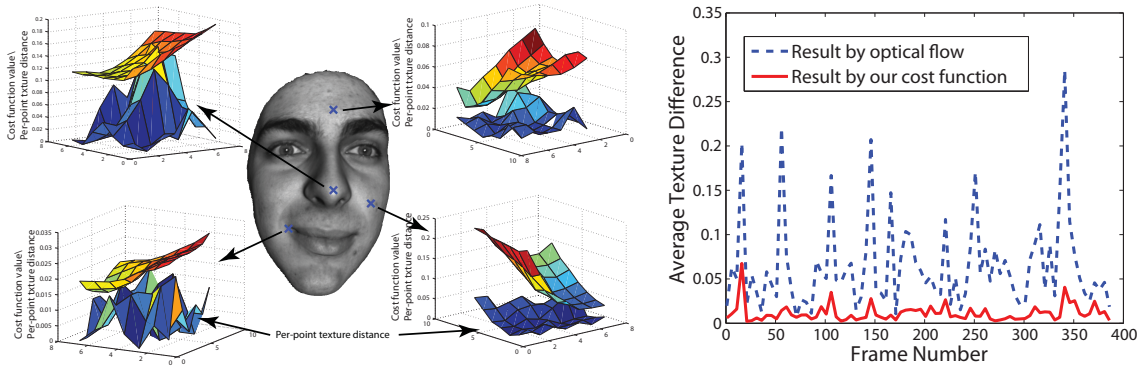
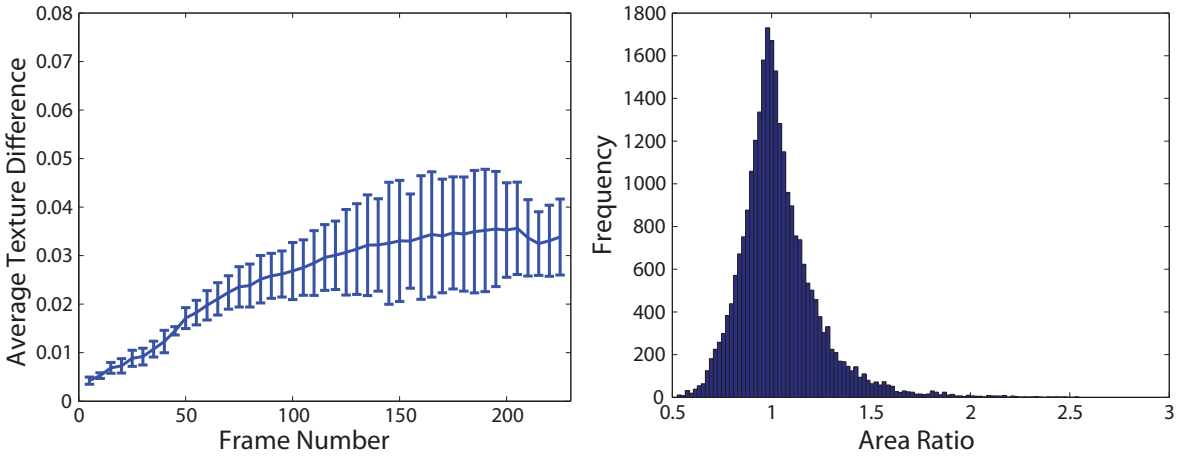


Figure 5.12: Sponge results with texture.



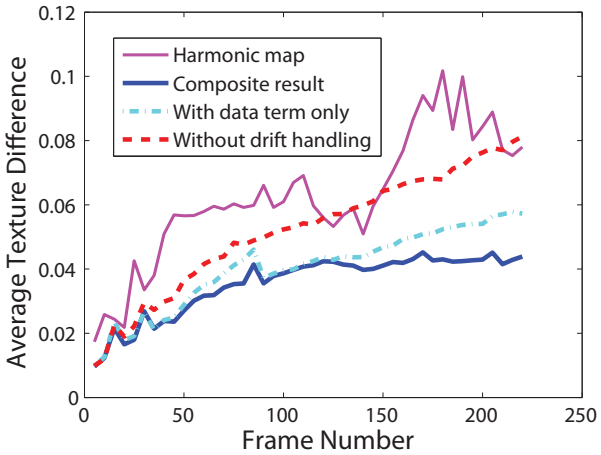
(a) $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ (b)

Figure 5.13: (a) Our cost function of Eq. 5.7 *v.s.* per-point texture distance in distinguishing subtle differences in correspondences and comparison with optical flow method for inter-frame registration (b) (details of comparison are described in the text).



(a) (b)

Figure 5.14: (a) shows the evaluation of the average per-point texture difference between every frame and the first frame on the whole 24 data set. (b) is the frequency of the area ratio of randomly selected triangles between current frame and the first frame, which shows that a significant number of them deforms anisometrically.



(a) Per-frame error for one sequence

Method \ Data	B_F	D_C	B_C	A_C	C_A
HM	0.0810	0.0987	0.0796	0.0983	0.0704
D.Term	0.0573	0.0423	0.0505	0.0452	0.0564
W.o. Drift	0.0364	0.0315	0.0431	0.0428	0.0396
Our Res.	0.0317	0.0262	0.0150	0.0299	0.0321

(b) Average error for 5 sequences

Figure 5.15: Results show influence of the regularization term used in the optimization of Eq. 5.17 and comparison with previous intrinsic tracking method used in Harmonic maps [270].

Chapter 6

A Generic Local Deformation Model for Surface Registration

> “To a man with a hammer, everything looks like a nail.”
>
> – “Mark Twain”

In our previous two approaches to the surface registration problem, we made an extensive use of the fact that a dense mapping between two surfaces can be achieved in a closed-form by knowing only a few correspondences. A limitation of relying on conformal mapping for finding correspondences is the assumption of isometric deformation of the underlying surface. Such assumption would certainly fail in situations where deformation details are required (*e.g.*, deformation transfer). In this chapter, we address the mathematical model of generic surface deformations.

6.1 Introduction

The problem of shape registration has become increasingly important in computer vision due to the wide availability of acquisition devices [3, 297]. Understanding the transformation between two shapes (*i.e.*, shape matching) is a fundamental task in shape analysis (*e.g.*, classification and recognition) and inference (*e.g.*, registration and tracking). Existing methods of matching shapes are often limited by the assumptions of the underlying deformations. In this

chapter, we present a generic shape deformation model that can represent arbitrary deformations between two shapes.

Our new deformation model is inspired by the basic ideas in Riemannian geometry [64] in which a shape is equipped with a metric tensor at each point in the parametrization domain. The problem of matching two shapes is hence equivalent to finding correspondences between their parameterized domains (Fig. 6.1). Different parametrizations represent the same surface (modulo the isometric deformation) if and only if for each correspondence, their metric tensors satisfy certain transformation rules. Hence, for surfaces undergoing isometric deformations, the task of matching becomes finding consistent parametrization such that the transformation rules are satisfied. To characterize more general deformations, at a given point, we consider a special *canonical parametrization* of the shape whose metric tensor *at that point* is Euclidean. Arbitrary deformations can be described by the local distortion of a circle between each pair of correspondence in the canonical parametrization domain [5, 233]. Such distortion can be consequently characterized by the eigenvalues of a canonical transformation relating to the Jacobian matrix between the two canonical parametrization domains (the *canonical distortion coefficient*). In the discrete setting, we consider the common case that a shape is represented as simplicial complex (*e.g.*, a planar or tetrahedral mesh). By assuming the deformation of the shape in the parametrization domain to be piecewise linear, the problem of computing the canonical distortion coefficient at a point becomes computing it at a face. The canonical parametrization at a point is simply equivalent to mapping each face to the 2D domain. Within this setting, the computation of the canonical distortion coefficient for each face becomes solving linear equations with a closed-form solution.

Given the above shape deformation model, the problem of finding the optimal shape registration result that best fits the deformation prior and the observed data can be formulated using the Markov Random Field (MRF) framework [83]. Recent combinatorial methods (*e.g.*, [252, 292]) have demonstrated the superiority of discrete optimization with respect to both solution quality and computational speed. Moreover, in our problem, an MRF formulation allows us to take multiple matching criteria into the same optimization framework. Because the deformation model is defined on each facet (*e.g.*, triangle for planar mesh), the MRF optimization involves high-order cliques. Compared to existing approaches, our method has the following advantages:

- The proposed deformation model is generic and flexible to handle arbitrary deformations. Most of the previous geometry-based deformation models (*e.g.*, isometry and conformality) fall into certain special classes of our model. As a result, our model can accurately describe the deviation from those existing models, which is important in characterizing real-world deformations.

- In the discrete setting, the computation of the canonical distortion coefficient for each basic element (*e.g.*, a triangle for a planar mesh) can be computed efficiently by solving linear equations with a closed-form solution.
- An efficient high-order MRF optimization algorithm is designed and implemented based on the dual-optimization technique ([279]) which requires minimal memory and achieves a good approximate solution.

In summary, we propose a generic deformation descriptor: the canonical distortion coefficient (CDC), and its use in a high-order MRF framework that allows us to impose the deformation of a template as a prior in a Bayesian setting. The novel formulation of the surface registration problem as an MRF problem with a cell complex structure seamlessly brings together Riemannian geometry and modern graphical model. Furthermore, our algorithm can be implemented in distributed hardware that achieves significant speedup. In our experiment, we demonstrate that the MRF optimization technique combined with the generalized deformation model leads to significant improvement for the problem of surface registration and tracking.

This chapter is organized as follows: In Sec. 6.2 we review related work on shape deformation models and techniques in high-order MRF optimization. The mathematical formulation of our general shape deformation model is presented in Sec. 6.3, where both continuous concepts and discrete counterparts are discussed. A high-order MRF formulation for shape registration is given in Sec. 6.4, with the design of an efficient optimization algorithm. In Sec. 6.5 we show the applications of our new deformation model and optimization technique to the problems of shape registration and tracking. Finally, we conclude our work in Sec. 6.6.

6.2 Related work

6.2.1 Surface deformation models

Quantitatively measuring the deformation of a surface is crucial in dynamic shape understanding. In the problem of surface registration, finding a surface representation that is invariant to a large set of deformations can help significantly simplify the registration process. In contrast, once two surfaces are registered, an effective surface deformation model would allow us to obtain meaningful information of the surface (*e.g.*, expression changes).

In the 90's, researchers usually studied surface deformation models based on physics-based models [114]. Among these models, where the surface are often represented as meshes or point set [240], different internal and external forces are considered.

There are many ongoing researches on the representations of the surface that allows us to manipulate the deformations of the surface. Examples include the mean-value coordinate [77], the Harmonic coordinates [21, 111], the Green-coordinates [157] and the Cauchy-coordinates [272, 273]. These deformation models are based on minimizing certain geometric features (*e.g.*, surface smoothness, gradient, conformality) given the deformations of a few key points on the surface. There are also heuristic methods such as region-based linear model [244].

Besides the above mentioned general surface deformation models, the deformations of a surface can be also represented according to specialized applications. In the case of facial expression analysis, the Facial Action Coding System (FACS) is a popular method for representing face expressions [71]. The idea of physics-based deformation has also been used for modeling facial expression changes [245], where a mechanics-based facial tissue model is studied. There are also facial expression models that are based on the muscles change [113], which can be also learnt from real data [223]. Another effective way of learning based facial deformation models is to use the idea of AAM [169] in 3D [28]. Such knowledge based models can help significantly simplify the search of deformations in new data.

Recently, data-based shape representation has become particularly popular in the application of shape animation or deformation transfer (*e.g.* [52, 183]). Many of these works are based on the idea of PCA. For example, the multilinear model [260], the bendshape model [20] and its various applications [154, 274, 276].

6.2.2 High-order MRF optimization

Although the idea of high-order MRF is not new [124, 263], its exact inference is often prohibited by the high complexity of the algorithm and/or the memory requirement. For example, the complexity of the belief propagation approach on a high-order graph [143] grows exponentially with the size of the maximal clique of the graph. Reducing a high-order graph to a pairwise one [106, 204] only sees its application in graphs with small clique size or label size due to memory limitations. An alternative approach to overcome such a limitation is to explore the tractable sub-structure of the problem [203], which often result in poor approximation to the original intractable problem. A better approximate of the original problem can be achieved by decomposing the problem into overlapping tractable sub-problems [133]. However, when the number of sub-problems become large, the memory that is required to coordinate the message becomes an obstacle for extending the algorithm to solve problems with large scale. A promising approach that requires the minimal memory (only the encoding of the energy is needed) is the message passing approach [279]. Nevertheless, the inference often takes long time to due to the slow convergence rate.

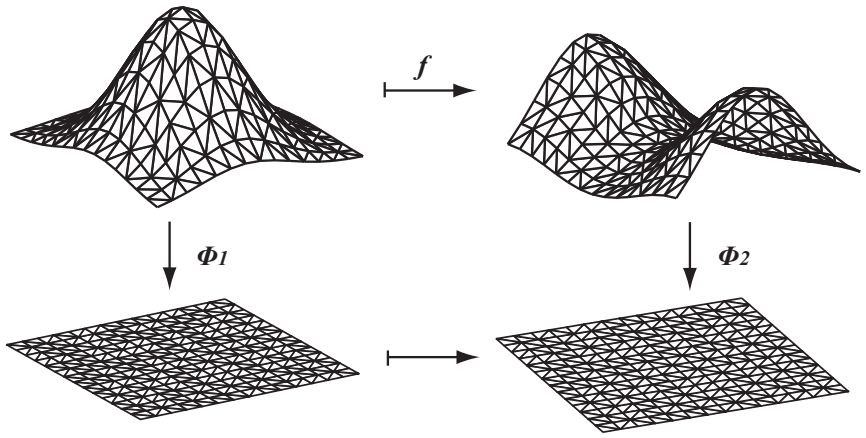


Figure 6.1: The problem of finding correspondences between two surfaces can be reduced to finding correspondences between their parametrization domain. The problem we solve in this paper is: given a predefined points on the first surface, find the correspondences of only those points on the second one. Efficient candidate selection schemes for surface registration and tracking are described in [292, 294].

6.3 Canonical distortion coefficient

In this section, we present the mathematical formulation of our deformation model in both continuous and discrete settings.

6.3.1 Continuous setting

Riemannian metric and parametrization

Let $(\mathcal{M}, g_{\mathcal{M}})$ denote a surface $\mathcal{M}$ equipped with a Riemannian metric $g_{\mathcal{M}}$. In Riemannian geometry ([64]), a surface is described by its local charts $\{(U_\alpha\}$, *i.e.*, $\mathcal{M} = U_\alpha \cup U_\beta \ldots$ and each open subset U_α is in $1-1$ correspondences $\phi_\alpha : U_\alpha \to \mathbb{R}^2$. Here ϕ_α is the local *parametrization*. For any $p \in U_\alpha \subset \mathcal{M}$, a metric tensor is associated to p as a symmetric positive definite matrix:

$$g^\alpha(p) = \begin{pmatrix} g^\alpha_{11}(p) & g^\alpha_{12}(p) \\ g^\alpha_{21}(p) & g^\alpha_{22}(p) \end{pmatrix}. \tag{6.1}$$

Within such setting, a point on $\mathcal{M}$ may be covered by multiple charts. In order for different local representations to describe the same surface, certain transformation rules must be satisfied, *i.e.*, if a point $p \in U_\alpha \cap U_\beta$, we must have

$$g^\alpha(p) = J_{\alpha\beta}(p)^T g^\beta(q) J_{\alpha\beta}(p). \tag{6.2}$$

Here $J_{\alpha\beta}$ is the Jacobian matrix of the transformation $x_\alpha(x_\beta)$ ($x_\alpha(x_\beta)$ is the local coordinate system of patch $U_\alpha(U_\beta)$) between charts U_α and U_β, *i.e.*,

$$J_{\alpha\beta} = \begin{pmatrix} \frac{\partial x_\alpha^1}{\partial x_\beta^1} & \frac{\partial x_\alpha^1}{\partial x_\beta^2} \\ \frac{\partial x_\alpha^2}{\partial x_\beta^1} & \frac{\partial x_\alpha^2}{\partial x_\beta^2} \end{pmatrix}. \tag{6.3}$$

Any local representation satisfying this transformation rule is a valid parametrization of the surface. Therefore, since the metric tensor at any point $p \in \mathcal{M}$ is positive definite, it is always possible to apply a proper linear transformation to its parametrization ϕ_α such that $g^\alpha(p)$ is the identity matrix. We call such a parametrization the *canonical parametrization* for point p:

Definition 24 ***(Canonical parametrization at a point)*** *For any $p \in \mathcal{M}$, a parametrization $\phi_\alpha : \mathcal{U}_\alpha \to \mathbb{R}^2$ is called* canonical parametrization for p *if the metric tensor at p is the identity matrix.*

Please note that although there exists infinite number of such a canonical parametrization for a surface, the parametrization at the particular point is unique. In the following we shall show that focusing on the parametrization only at one point at a time allows us to *characterize arbitrary deformations between two surfaces while regardless of both the intrinsic and extrinsic representations of the surface*, which is the main advantage of our new deformation model.

Diffeomorphisms between two shapes

Now we consider arbitrary diffeomorphisms between the parametrization domains of two surfaces (Fig. 6.1). For any correspondence $p \in \mathcal{U}_\alpha \subset \mathcal{M} \to q \in \mathcal{U}_\beta \subset \mathcal{N}$, the change of metric $g^\alpha(p) \to J_{\alpha\beta}(p)^T g^\beta(q) J_{\alpha\beta}(p)$ reflects how locally a circle is deformed into an eclipse (Fig. 6.2(a))[1]. In particular, under canonical parametrization for points p and q, the matrix $J_{pq}^T J_{pq}$ accurately characterizes such local deformation, where J_{pq} is the Jacobian at point p. If we only consider the change of shape, *i.e.*, how a circle is deformed into an eclipse regardless its orientation, the eigenvalues λ_1, λ_2 of $J_{pq}^T J_{pq}$ can best describe such change. Therefore, the local deformation between two surfaces can be characterized by the eigenvalues λ_1, λ_2 for each correspondence. Formally, we define:

[1] A rigorous formulation can be found in Appendix C.

Definition 25 ***(Canonical distortion coefficient)*** *The eigenvalues of the Jacobian transformation matrix $J_{pq}^T J_{pq}$ between any canonical parametrization at p and q are the* canonical distortion coefficients *between the two points.*

We call the Jacobian matrix J_{pq} between the two points p and q under the canonical parametrization the *canonical Jacobian*. The significance of using canonical Jacobian is that it allows us to *characterize arbitrary deformation by considering the parametrization of the surface at a single point*. Some special cases of deformations can be characterized by the canonical distortion coefficient as follows:

(i) In the case of the isometric deformation, a unit circle is mapped to a unit circle, *i.e.*, $\lambda_1 = \lambda_2 = 1$.

(ii) In the case of the conformal deformation, a unit circle can be mapped to a circle with arbitrary radius [233]. Thus, $\lambda_1 = \lambda_2$.

To further connect the canonical distortion coefficient to a general class of diffeomorphisms defined in the complex plane $f : \mathcal{U}_\alpha \to \mathcal{U}_\beta$, between any canonical parametrization x_α and x_β for p and q respectively, we define

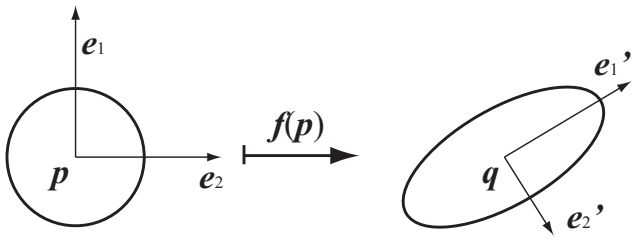
$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial x_\alpha^1}{\partial x_\beta^1} + \frac{\partial x_\alpha^2}{\partial x_\beta^2}\right) + \frac{i}{2}\left(\frac{\partial x_\alpha^2}{\partial x_\beta^1} - \frac{\partial x_\alpha^1}{\partial x_\beta^2}\right)$$
$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial x_\alpha^1}{\partial x_\beta^1} - \frac{\partial x_\alpha^2}{\partial x_\beta^2}\right) + \frac{i}{2}\left(\frac{\partial x_\alpha^2}{\partial x_\beta^1} + \frac{\partial x_\alpha^1}{\partial x_\beta^2}\right),$$

The notion of quasi-conformality [5] can be characterized by the following *Beltrami-coefficient*:

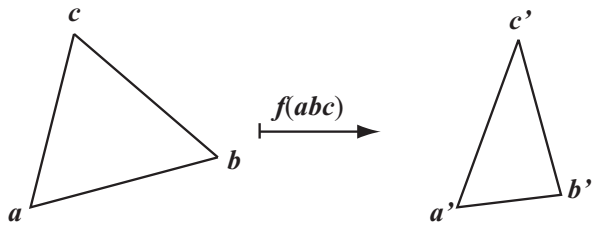
$$\mu(z) \equiv \frac{\partial f}{\partial \bar{z}} \Big/ \frac{\partial f}{\partial z},$$

which gives us all the information about the conformality of f. Suppose $\lambda_1 \geq \lambda_2$, it can be shown that $|\mu(z)| = (\sqrt{\lambda_1} - \sqrt{\lambda_2})/(\sqrt{\lambda_1} + \sqrt{\lambda_2})$. In particular, f is called *holomorphic* if $\mu(z) = 0$ ([78]), *i.e.*, $\lambda_1 = \lambda_2$, coinciding with the fact that holomorphic function is another description of conformal mapping[2]. Hence the Beltrami-coefficient generalizes conformal mapping and can be partially determined using the canonical distortion coefficient. However, the Beltrami-coefficient is for surface parametrization, where the scaling factor is lost. The proposed canonical distortion coefficient preserves the scale information which is important for shape matching. Besides, the canonical distortion coefficient is directly extendable to nD. In this paper, we propose an efficient approach to compute the local deformation based on the canonical distortion coefficient in the context of shape registration.

[2]Strictly speaking, conformal mapping is a holomorphic function whose derivative is everywhere non-zero.



(a) Continuous setting



(b) Discrete setting

Figure 6.2: The finite element method assumes the transformation between facets to be piecewise linear and $f(\vec{ab}) = \vec{a'b'}$, $f(\vec{ac}) = \vec{a'c'}$. Under the linearity assumption, the Jacobian can be computed in a closed form for each triangle pair $\triangle abc \mapsto \triangle a'b'c'$.

6.3.2 Finite element discretization

The basic assumption in finite element analysis [35] is that the continuous space can be approximated using a set of *basic elements* (*e.g.*, polynomial functions defined on each face). Meanwhile, consistencies must be preserved at the boundaries among the basic elements. In this paper, we consider the most common representation of a continuous surface – a triangular mesh, whose basic finite element is a triangular face. In this discrete setting, the canonical distortion coefficient (CDC) is assumed to be constant for each basic element (*i.e.*, $\triangle abc$ shown in Fig. 6.2).

Thus, the concept of canonical parametrization (Sec. 6.3.1) can be expressed in the following manner: a parametrization of a point p is locally Euclidean at p if the images of any two tangent vectors have the same angle and length. In the discrete setting, this means a triangle $\triangle abc$ keeps all its angles and lengths in its parametrization, which can be achieved by simply mapping the face $\triangle abc$ onto a 2D domain by keeping all its edge lengths.

Finally, we consider the computation of the canonical Jacobian (Sec. 6.3.1). In the continuous setting, the Jacobian matrix at a point p is a linear operation that transforms tangent vectors at p to tangent vectors at q. Given a basic element $\triangle abc$ in the discrete setting, the tangent space at p is equivalent to the linear space spanned by $\triangle abc$. Hence the linear mapping $J(\cdot)$ between two canonical domains should satisfy $J(\vec{ab}) = \vec{a'b'}$ and $J(\vec{ac}) = \vec{a'c'}$. The Jaco-

bian of a linear transformation between two triangles is a 2×2 matrix and can be computed in closed-form. Since $J(\cdot)$ is linear, $J(\vec{bc}) = \vec{b'c'}$ must be satisfied as well, i.e.,

$$\text{Jacobian for mapping } p \rightarrow q \Leftrightarrow \text{Linear transformation matrix for mapping } \vec{ab} \rightarrow \vec{a'b'}, \vec{ac} \rightarrow \vec{a'c'}$$

For clarity, Alg. 2 summarizes the algorithm for computing the canonical distortion coefficient. Note that the computation is in analogy with previous work for surface parametrization (*e.g.*, [196, 201, 212]) since both are based on the same piecewise linear assumption. However, here we derive it from a different continuous setting in the context of shape deformation estimation. Also note that when the shape is n-manifold, the computation of CDC becomes solving n linear equations and eigenvalues.

Algorithm 2: Algorithm for computing the canonical distortion coefficient (CDC) for each triangular facet.

Input : $\triangle abc$ and its mapping $\triangle a'b'c'$
Output : CDC for mapping from $\triangle abc$ to $\triangle a'b'c'$.

Step One: Map the triangles $\triangle abc$ and $\triangle a'b'c'$ to 2D and keep their orientation.
Step Two: Compute the 2×2 linear transformation J mapping $\vec{ab}$ to $\vec{a'b'}$ and $\vec{ac}$ to $\vec{a'c'}$.
Step Three: Compute the eigenvalues, λ_1 and λ_2 of $J^T J$.
Step Four: Output λ_1 and λ_2

6.4 High-order MRF-based shape registration

6.4.1 MRF formulation for shape registration

Given the canonical distortion coefficient (CDC) defined for each basic element, *i.e.*, the triangular face, one can either deform the original shape (*e.g.*, [229]), or find the correspondences between two shapes combined with other matching cues (*e.g.*, [292]). Here we consider the problem of finding the mapping f between two shapes $\mathcal{M}$ and $\mathcal{N}$. Similar to [292], we assume a set of n points $\mathcal{V} = \{p_u | p_u \in \mathcal{M}, u = 1, \ldots, n\}$ are sampled on the surface $\mathcal{M}$ and a triangulation of these points are constructed (Fig. 6.1). Hence the task of shape registration becomes finding the correspondences for the set $\mathcal{V} \subset \mathcal{M}$ on shape $\mathcal{N}$.

To formulate the shape matching problem using graphical models, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ is a set of faces. For each $u \in \mathcal{V}$, let the random variable $x_u \in \mathcal{L} = \{1, \ldots, L\}$ denote the discrete labeling of all the possible

matching candidates of vertex $u \in \mathcal{M}$ on shape $\mathcal{N}$. In the rest of this paper, we use x_u to denote the labeling of vertex u or the point on shape $\mathcal{N}$ when it is clear from the context.

Firstly, let $\text{fea}(\cdot)$ be the feature vector (*e.g.*, texture or shape context) at each point on the shape. We define the cost function $\theta_u(x_u)$ of matching u to x_u:

$$\theta_u(x_u) = |\text{fea}_{\mathcal{M}}(u) - \text{fea}_{\mathcal{N}}(x_u)|^2.$$

Next, we denote by $\lambda_{\mathcal{M}}(u,v,w) = \begin{pmatrix}\lambda_1(u,v,w)\\ \lambda_2(u,v,w)\end{pmatrix}$ as the prior knowledge of CDC that characterizes the deformation for the face (u,v,w) and $\lambda_{\mathcal{N}}(x_u,x_v,x_w) = \begin{pmatrix}\lambda_1(x_u,x_v,x_w)\\ \lambda_2(x_u,x_v,x_w)\end{pmatrix}$ as the CDC computed from deforming $\triangle uvw$ to $\triangle x_u x_v x_w$ (Alg. 2). Hence we define:

$$\theta_{uvw}(x_u,x_v,x_w) = d(\lambda_{\mathcal{M}}(u,v,w), \lambda_{\mathcal{N}}(x_u,x_v,x_w))$$

as the deformation energy. Here $d(\cdot,\cdot)$ is the distance function that is defined according to the application.

Finally, given the feature function $\text{fea}(\cdot)$ and the deformation prior $\lambda_{\mathcal{M}}(\cdot,\cdot,\cdot)$ for each vertex and face, the problem of shape registration becomes solving the optimal configuration $\mathbf{x}$ that minimizes the following energy:

$$\min_{\mathbf{x}} E(\mathbf{x}) = \sum_{u\in\mathcal{V}} \theta_u(x_u) + \sum_{(u,v,w)\in\mathcal{F}} \theta_{uvw}(x_u,x_v,x_w). \tag{6.4}$$

To reduce the search space $\mathcal{L}$ and avoid local minima, we adopt a hierarchical optimization scheme in solving the energy function (6.4). Inspired by [292], sparse feature points are first selected to compute the initial matching with the global constraint $\lambda_1 = \lambda_2 = 1$ for each triangle $(u,v,w) \in \mathcal{F}$. Then, a small set of candidates $\mathcal{L}$ are selected to find the optimal local match by minimizing the energy function (6.4).

6.4.2 An efficient high-order MRF optimization

The high-order potential of Eq. 6.4 presents the difficulty for solving our surface registration problem. Existing algorithms either transform high-order cliques into pairwise ones (*e.g.*, [106]) or decompose the original problem into a union of sub-problems (*e.g.*, [133]). However, these algorithm are designed for general problems and requires extra memory to store the transformed problem or the dual variables that relates the sub-problems. For the problem of high-order MRF inference, memory efficiency is an important issue since a poorly designed optimization algorithm can easily reach the limit of current hardware. In this paper, we follow

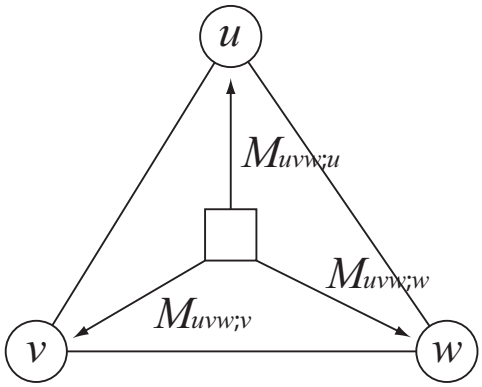


Figure 6.3: An example of the messages defined for the dual problem of Eq. 6.5. For each face $\triangle uvw$, the messages are defined from the high-order clique (u, v, w) to each of the nodes u, v and w.

the framework by Thomas [279] to design a fast and memory efficient MRF optimization algorithm using the linear programming (LP) relaxation technique.

The first step in the LP relaxation of Eq. 6.4 is to introduce the indicator variables. For any $u \in \mathcal{V}$ and $i \in \mathcal{L}$, we define

$$\tau_{u;i} = \begin{cases} 1 & \text{if } x_u = i \\ 0 & \text{otherwise} \end{cases}.$$

Also for any $(u, v, w) \in \mathcal{F}$ and $(i, j, k) \in \mathcal{L} \times \mathcal{L} \times \mathcal{L}$, we define

$$\tau_{uvw;ijk} = \begin{cases} 1 & \text{if } x_u = i, x_v = j, x_w = k \\ 0 & \text{otherwise} \end{cases}.$$

Similarly, we define $\theta_{u;i} = \theta_u(i)$ and $\theta_{uvw;ijk} = \theta_{uvw}(i, j, k)$. Hence we have the integer LP formulation of the problem of Eq. 6.4:

$$
\begin{aligned}
&\min_{\tau} \sum_{u\in\mathcal{V}} \sum_{i\in\mathcal{L}} \theta_{u;i}\tau_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} \sum_{(i,j,k)\in\mathcal{L}^3} \theta_{uvw;ijk}\tau_{uvw;ijk} \\
&\text{s.t.} \sum_{i} \tau_{u;i} = 1, && \forall u \in \mathcal{V} \\
&\sum_{i,j,k} \tau_{uvw;ijk} = 1, && \forall (u,v,w) \in \mathcal{F} \\
&\sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}, && \forall (u,v,w) \in \mathcal{F} \text{ and } i \in \mathcal{L} \\
&\tau_{u;i}, \tau_{uvw;ijk} \in \{0,1\}.
\end{aligned}
$$

By relaxing the variables to the domain $[0,1]$, we obtain the dual form of the above LP problem as

$$
\begin{aligned}
&\max_{M} \sum_{u} \min_{i} \overline{\theta}_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} \min_{i,j,k} \overline{\theta}_{uvw;ijk} \\
&\text{s.t. } \overline{\theta}_{u;i} = \theta_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} M_{uvw;u:i}, \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L} \\
&\quad \overline{\theta}_{uvw;ijk} = \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \\
&\quad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L}.
\end{aligned}
\tag{6.5}
$$

Here $M_{uvw;u:i}$ is the dual variable (message) corresponding to the constraint $\sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}$. (Fig. 6.3).

The dual problem of Eq. 6.5 can be solved by the simple min-sum diffusion algorithm [279] as shown in Alg. 3 (Note that in [279] their optimization problem is a maximization so the algorithm is called max-sum). It has been shown that at convergence, the solution satis-

Algorithm 3: Min-sum diffusion algorithm.

repeat
  for each $M_{uvw;u:i}$ **do**
    $M_{uvw;u:i} -= \frac{1}{2}[\theta_{u;i} - \min_{j,k} \theta_{uvw;ijk}]$ and reparameterize $\theta_{u;i}$ and $\theta_{uvw;ijk}$ according to the constraints in Eq. 6.5.
  end for
until convergence

fies J-consistency condition as introduced in [279]. Since after each update of the message,

reparameterization is performed, no extra memory is needed for storing all the dual variables $M_{uvw;u:i}$. Hence the memory requirement for the Alg. 4 is the storage for the primal variables, *i.e.*, $O(|V||\mathcal{L}| + |\mathcal{F}||\mathcal{L}|^3)$, which can not be avoided by any algorithm.

Each update of the message in Alg. 3 only involves the parameters in a triangle. Also within each face $\triangle uvw$, the update of each label $M_{uvw;u:i}, i = \{1, \ldots, L\}$ is independent. Hence the algorithm can be significantly accelerated using distributed hardware.

In order to explore the parallelism of the min-sum Algorithm 3, we define the concept of independent face set:

Definition 26 ***(Independent face set)*** *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$, a subset $\mathcal{F}_k \subset \mathcal{F}$ is called* independent face set *if for any $f_i, f_j \in \mathcal{F}_k$, $f_i \cap f_j = \varnothing$.*

The decomposition of a set $\mathcal{F}$ into subsets of independent face sets $\mathcal{F} = \cup_i \mathcal{F}_i$ can be efficiently computed in polynomial time by a simple greedy algorithm. Hence we can implement Alg. 3 in parallel as shown in Alg. 4. The maximal speedup achieved in Alg. 4 is $\max_i(|\mathcal{F}_i||\mathcal{L}|)$.

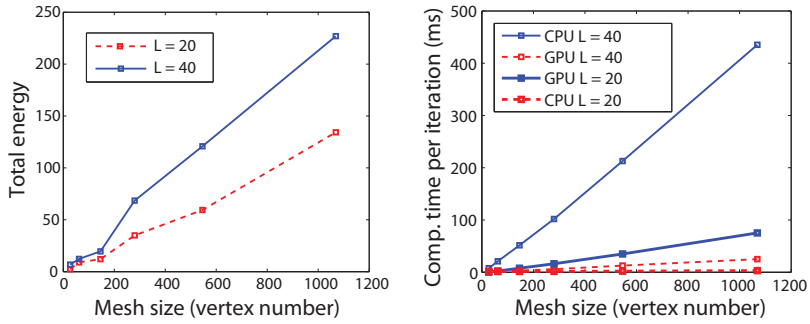
Algorithm 4: Parallel min-sum diffusion algorithm.

Decompose $\mathcal{F}$ into independent face sets $\cup_i \mathcal{F}_i$
repeat
  for each Independent face set $\mathcal{F}_i$, in parallel for all $(u, v, w) \in \mathcal{F}_i$ and $k \in \mathcal{L}$ **do**
    Update the message $M_{uvw;u:k}$, $M_{uvw;v:k}$ and $M_{uvw;w:k}$ and do reparameterization (Alg. 3).
  end for
until convergence

6.4.3 Performance evaluation

We implement Alg. 4 using the NVIDIA® CUDA architecture [116][3]. In approximation algorithms [258], the approximation error (AE) is defined as the gap between the optimal integral solution and the solution obtained by the algorithm. The gap between the result by the LP relaxation and the optimal integral solution is often used to upper bound the approximation error. Here we aim to test the true approximation error, as well as the speedup, by designing the test inputs as follows: Given any input mesh, we randomly assign a default labeling l_u for each node $u \in \mathcal{V}$. We define the singleton potentials of Eq. 6.4 as

[3]The source code for CUDA implementation of the MRF optimization algorithm and the executables used in our comparison between CPU and GPU implmentations can be downloaded from http://www.cs.sunysb.edu/~yzeng/software_HighorderMRF.html.



(a) (b)

Figure 6.4: Performance analysis of our MRF optimization algorithm. (a) shows the optimality using the test cases described in Sec. 6.4.3. (b) shows the speedup using the parallel implementation of Alg. 4. We show the runtime per iteration since different inputs would result in different iteration counts.

$$\theta_u(x_u) = \begin{cases} 0 & \text{if } x_u = l_u \\ rnd(1) & \text{otherwise} \end{cases},$$

where $rnd(1)$ is a random number between $[0, 1]$. Also we define the high-order potentials as

$$\theta_{uvw}(x_u, x_v, x_w) = \begin{cases} 0 & \text{if } (x_u, x_v, x_w) = (l_u, l_v, l_w) \\ rnd(1) & \text{otherwise} \end{cases}.$$

In such case, the optimal solution of Eq. 6.4 should be $\{l_u | u \in \mathcal{V}\}$. Fig. 6.4 (a) shows the result of our algorithm using the above designed test cases for different mesh and label size. Note that although the total energy increases with mesh size, the average energy per term (vertex and face) remains significantly low (< 0.01 for all cases). Fig. 6.4 (b) shows the comparison on average time taken per iteration, between the implementations with and without GPU accelerations. The total number of iterations depends on the energy. In our experiment the algorithm converges within 3000 iterations. Hence our algorithm is both memory and computational efficient, which is important for shape tracking.

6.5 Experimental results

In this section, we apply our deformation model and optimization technique to the problems of shape registration and tracking. The input to our algorithm is two 3D shapes $\mathcal{M}$ and $\mathcal{N}$, and a triangulated sampling point set $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ on $\mathcal{M}$ (Fig. 6.6 (a)). The output is the registered result on $\mathcal{N}$ for each point in $\mathcal{V}$ (Fig. 6.6 (c)). We implement our algorithm on an Intel®

Core(TM)2 Duo 3.16G PC with 4G RAM and an NVIDIA® Geforce 9800GTX+ graphics card with 128 CUDA cores. The number of labels L (matching candidates) is set to be 2^n for best hardware performance. In most of our experiments, we set $L = 64$. The computation of all the L^3 possible CDCs for one face takes only $2.0ms$ on average on GPU. Hence the computation of the energy term $\theta_{uvw;ijk}$ for a mesh with 165 vertices and 272 faces takes only 0.5s.

Estimation of deformation prior: In our experiment, we assume that *the deformation priors are similar across different shapes of the same type* (*e.g.*, human faces [260]). The ground truth deformation prior can be obtained by 3D scanning systems with reliable texture information (*e.g.*, markers). As shown in Fig. 6.5, the two 3D data in (a) and (c) are captured with markers using the system introduced in [270]. Here we select two frames with the largest expression difference to measure the maximal possible change of CDC. Fig. 6.5 (b), (d), (f) and (g) show the visualization of the distribution of CDC. From the above data set we obtain the allowed bound for human face expression changing from neutral to large deformation as $I_1 = [0.7, 5.66], I_2 = [0.1, 4]$ for λ_1 and λ_2, respectively. For the problem of surface registration, we define a Potts-like energy for the high-order terms in Eq. 6.4 as follows:

$$\theta_{uvw}(x_u, x_v, x_w) = \begin{cases} 0 & \text{if } \lambda_1 \in I_1 \text{ and } \lambda_2 \in I_2 \\ 10 & \text{otherwise} \end{cases}, \tag{6.6}$$

where λ_1 and λ_2 are the CDCs obtained by matching $\triangle uvw$ to $\triangle x_u x_v x_w$.

6.5.1 Shape registration

Fig. 6.6 shows one of our shape registration results and its comparison with the results of Zeng et.al [292]. The singleton terms are defined similarly as in [292]. To obtain enough matching candidates, re-sampling is made near its original candidates. The high-order graph matching formulation in [292] assumes the two surfaces are isometrically deformed so they have consistent conformal mapping if three correspondences are found. Hence when the deformation is not isometric, the registered points can be significantly distorted (Fig. 6.6 (b) and (c)). Besides, the optimization technique in [292] requires reducing high-order terms into pairwise ones so the memory required is huge and it only handle label size < 5. In contrast, our method poses deformation constraint onto each triangle and guarantees the consistency condition for the final solution. As a result, unlikely matching is avoided and the results are visually plausible (Fig. 6.6 (e) and (f)). More quantitative comparisons are given in Table 6.1. It can been seen that the unnatural deformations presented in [292] are significantly reduced.

To test the accuracy achieved by using a anisometric deformation prior, defined as the canonical distortion coefficient (CDC) in our paper, we design the following experiment. The

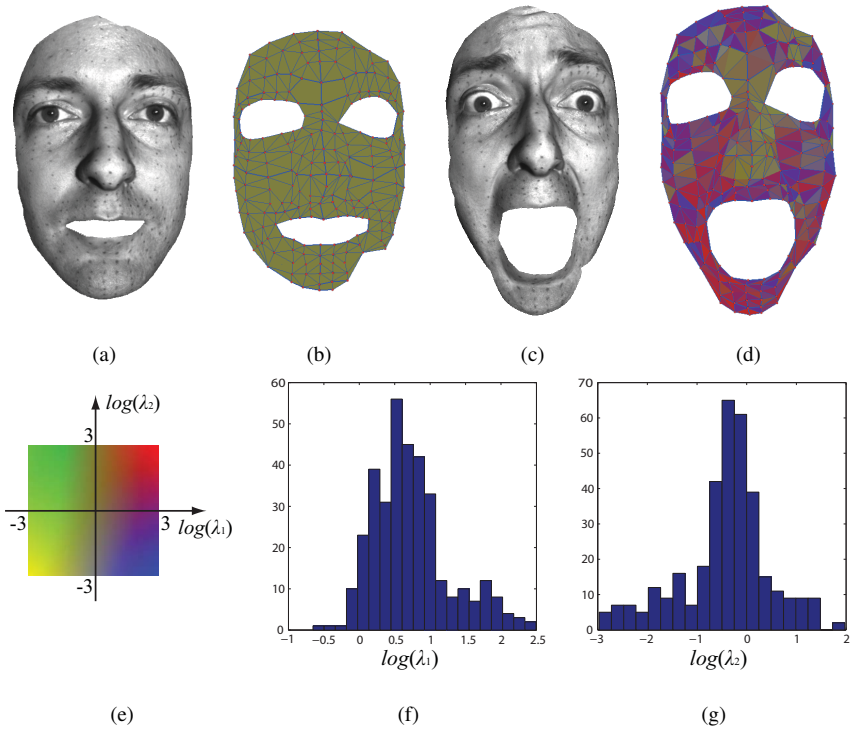
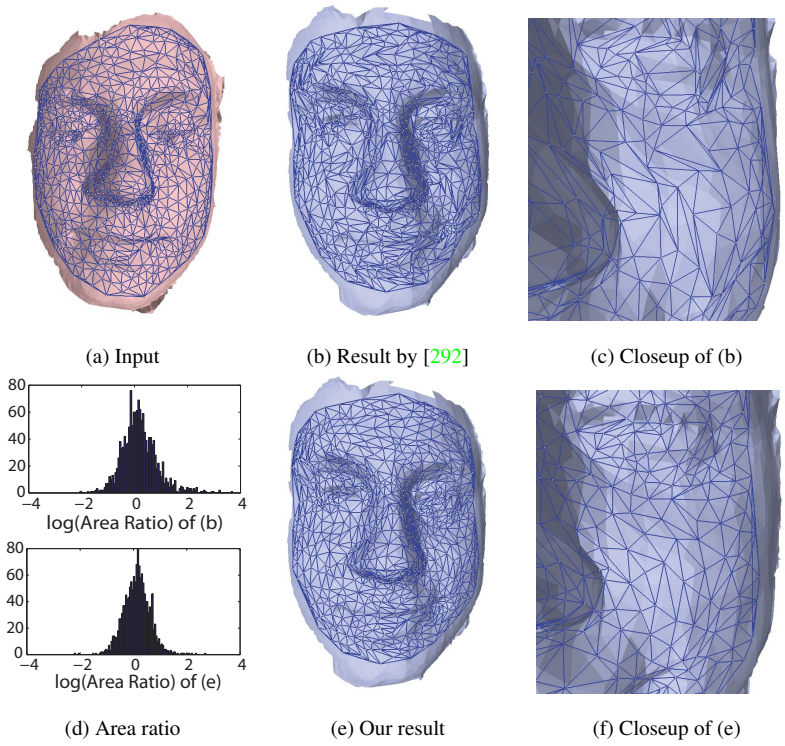


Figure 6.5: Expression deformation prior is obtained by 3D scanned data with markers. (a) and (b) show the 3D scan of the onset and peak of a facial expression with large shape deformations respectively. (c) and (d) are the corresponding triangular meshes constructed from the 3D scan data. The color coding in (d) shows the deformation intensity as illustrated in (e). The histogram of the canonical distortion coefficient values are shown in (f) and (g).

3D scan of a highly deformable toy is captured by the system introduced in [270] before and after a large deformation, as shown in Fig. 6.7 (a) and (b), respectively. Outliers are pruned off using the selection tool provided by the MeshLab software [4]. An original set of points and their candidates are selected using the method described in [292]. To establish the ground truth and estimate the deformation prior, we manually select 20 facets and their matches based on the texture information. The two shapes are then matched *without using the texture information*, *i.e.*, in Eq. 6.4 of in the paper, we use curvature as the singleton term (data likelihood) and the learnt deformation prior as the high-order term (the deformation prior). Fig. 6.8(b) shows the result using isometric assumption $(\lambda_1, \lambda_2) = (1, 1)$ and Fig. 6.8(c) shows the result using the learnt CDC prior $(\lambda_1, \lambda_2) = (1.028, 0.993)$. To compare the accuracy achieved by the two assumptions, we compare the average texture difference between the original area covered by the matching points (the blue mesh in Fig. 6.8) and the matched area.

[4] http://meshlab.sourceforge.net/



(a) Input (b) Result by [292] (c) Closeup of (b)

(d) Area ratio (e) Our result (f) Closeup of (e)

Figure 6.6: Shape registration result. (a) shows the input mesh with triangulated discrete samples obtained by the method described in [292]. The matching result and its closeup is shown in (b) and (c).

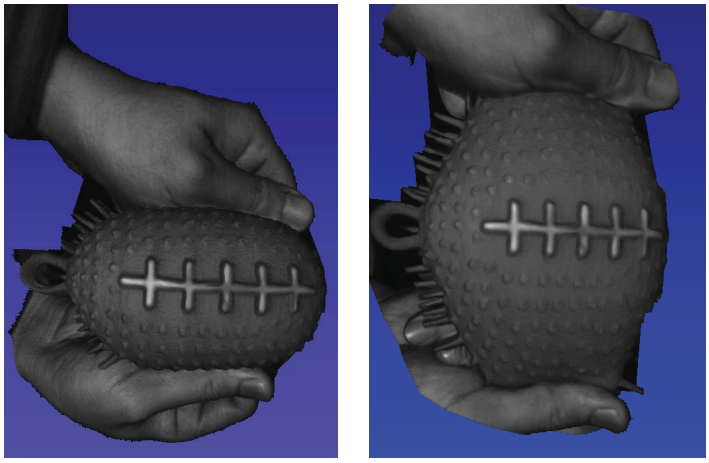
6.5.2 Shape tracking

To apply our method to tracking dynamic 3D scanned data, we consider both inter-frame consistency and the consistency between the current frame and the first frame. For the singleton term in Eq. 6.4, we use the robust metric defined in the paper of [294]. To impose inter-frame consistency, we use the same data set (here we select two consecutive frames with largest deformation change) and obtain the allowed change of CDC between frames to be $I_1 = [0.874, 1.143]$ and $I_2 = [0.846, 1.182]$ for λ_1 and λ_2 respectively. Also we handle drift error by imposing the consistency between the first frame and the current frame, using the same face deformation prior learnt in Fig. 6.5.

Fig. 6.9 shows tracking results on the BU-4DFE database [284]. A template is constructed in the first frame and tracked in the subsequent frames (same as in [294]. Because of the temporal continuity in consecutive frames, sufficient matching candidates can be obtained by only looking at the neighborhood of each point. In this data set, the texture information is noisy so relying on texture information only can lead to erroneous results. By imposing a

Data	Zeng *et.al.* [292]	Our method
Face_Smile	$(2.26, 0.19, 67.83)$	$(1.24, 0.86, 4.2)$
Face_Laugh	$(1.75, 0.12, 111.11)$	$(1.36, 0.82, 11.0)$
Face_Sad	$(1.87, 0.19, 78.62)$	$(1.48, 0.87, 7.52)$

Table 6.1: Comparison with [292]. Here $(\cdot, \cdot, \cdot)$ denote the average, minimal and maximal area ratios between the original facets and the matched facets.



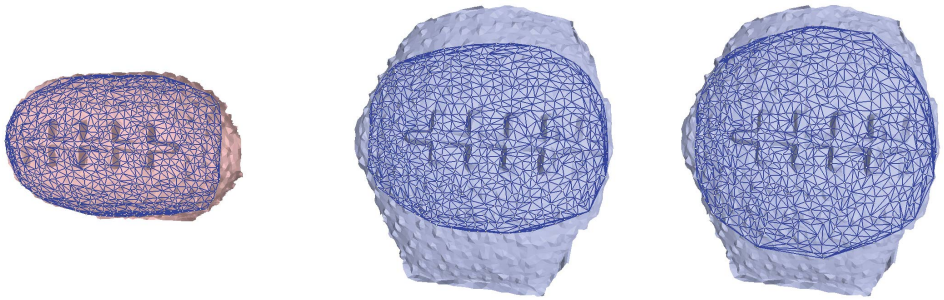
(a) Original shape (b) Deformed shape

Figure 6.7: The 3D scan of a highly deformable toy.

simple prior on the bound of the deformation, we have achieved plausible tracking results as shown in Fig. 6.9.

6.6 Conclusion

We have presented a generic, geometry-inspired deformation model for characterizing arbitrary diffeomorphisms between shapes. An efficient and accurate algorithm to compute the canonical distortion coefficients (CDCs) is proposed based on finite element analysis in the discrete setting. Searching for the optimal shape registration result given such model is made possible with a high-order MRF framework. An efficient optimization algorithm is designed for such problem. We demonstrated the speed and accuracy performance on 2-manifold shape registration. However, both the deformation model and the optimization algorithm can be easily extended to high-dimensional space.



(a) Original shape with discrete samples (b) Result with isometric assumption (c) Result with learnt CDC prior

Figure 6.8: The comparison between the results of shape registration with isometric assumption (b) and with learnt CDC prior (c). Only curvature and high-order deformation prior are used in the registration.

With isometry assumption	With learnt CDC
0.073 ($(\lambda_1, \lambda_2) = (1, 1)$)	0.005 ($(\lambda_1, \lambda_2) = (1.028, 0.993)$)

Table 6.2: Comparison between results with and without isometric assumption. The matching is done without using the texture information. The comparison is based on the average texture difference (the gray level is normalized in $[0, 1]$), by warping the source mesh to the matched target mesh.

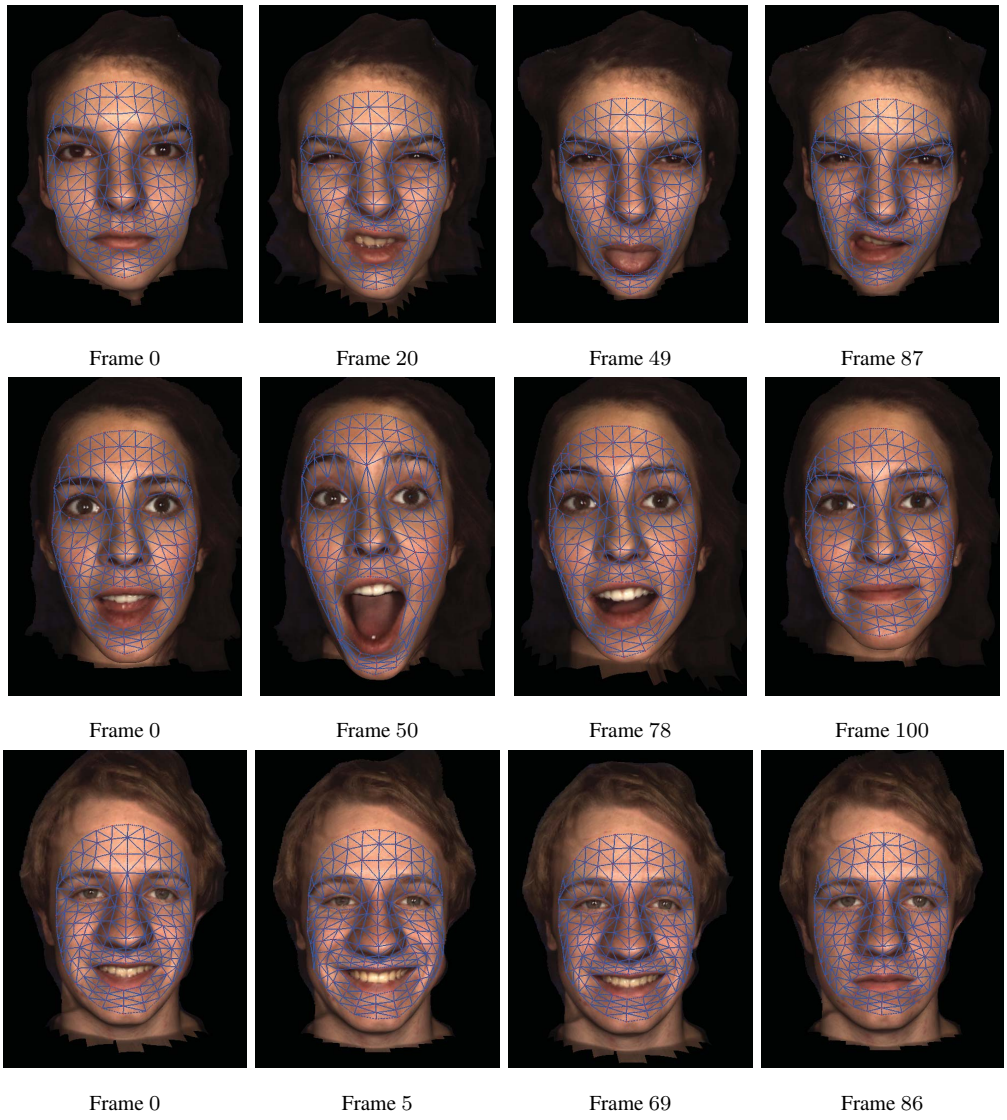


Figure 6.9: Shape tracking results.

Chapter 7

Conclusion and Future Work

> “End is only the beginning.”
> – Anonymous

7.1 Conclusion

In this dissertation, we approached the dense surface registration problem by exploring the connections / discrepancies between geometric based methods and graph-based approaches. Our work is largely motivated by recent trends in computer vision and computer theory – the rapid development of 3D scanning techniques, the popularity of graphical models and the search of boundaries in computation.

In Chap. 2 and 3, we reviewed and investigated the connections between the continuous geometry and the combinatorial structure. Specifically, the following important theoretical aspects are explored:

- Geometry studies the properties of the surface under different invariant groups. In Riemannian geometry, a metric tensor is defined for each point on the surface. Properties such as curvature, geodesics can all be defined using the metric tensor. The deformation of the surface can be also denoted by the change of the metric tensor. In contrast, conformal geometry disregards the scale information, which introduces certain flexibilities.

- When we disregards both the scale and angle information, we are then left with topology information of the surface. Topology is the gate that connects continuous geometry to discrete (algebraic) graph structure. Nevertheless, we have shown that in fact the graph structure contains much richer information than the topology of surfaces.

- Graphical models are based on the concept of probability functions. It has been partially shown that the discrete structure (graphical models) has its physical meaning, and is closely related to the complexity of computation.

- In optimization, we show that convexity (or submodularity for set functions) plays an important role in characterizing tractable problems. It is also important to explore the duality in these problems to achieve efficient solutions.

- We show that a wide class of graphical models can be represented by the complexity of its description. It eliminates the gap between the order of the graph representation but focuses on the description complexity. However, the boundary between a concise description (polynomial) and a complex description is still yet to be explored for answering the P=NP problem.

Practically, we explored the applications of the ideas in three distinct scenarios:

- In Chap. 4, we made the first attempt to encode the geometric information into the energy of a graph and formulated the surface registration as a high-order graph matching problem. An important innovation is the candidate selection scheme based on the closed-form solution of conformal geometry, which allows us to convert a continuous problem into a combinatorial problem. We also extended the existing graph matching algorithm, namely the dual-decomposition framework, to solve high-order cases.

- In Chap. 5, we addressed the challenging problem of measuring the quality of match for each possible correspondence between two surfaces. Again we take advantage of the property of conformal mapping that a closed form solution for surface matching can be efficiently computed. Our new distance function is derived under a general framework for surface / image matching, which involves the idea of min-marginal in statistics and considers global / local tradeoffs. Finally, a complete probabilistic framework for surface tracking is used that encodes the new distance in its terms, allowing us to do robust inference.

- In Chap. 6, we proposed a new surface deformation model that is able to characterize arbitrary deformations between two surfaces. Based on Riemannian geometry and

finite element analysis, we connect the linear transformation at each point to the linear transformation on each facet. As a result, the proposed deformation model is both independent of parametrization and 3D embedding. To apply the proposed model for surface registration, we formulated the surface registration problem into a high-order MRF optimization. By employing the LP relaxation and min-sum diffusion optimization technique, an efficient parallel algorithm is derived for solving the surface registration problem.

Despite the effort made in this dissertation for fully understanding the connections in things, many problems remain to be solved or clarified. In the following, we highlight the possible future works.

7.2 Future work

The ideas explored in this dissertation open many possibilities for future investigation.

Theories

- In Chap. 2, we discussed two ways of comparing the continuous structure and the discrete structure, namely structure-based and function-based. This beautiful duality indicates a possible unified theory for studying the continuous and discrete structure. A unified theory can be beneficial in multiple ways. At this point, it is still not clear to us whether such a unified theory exists.

- The optimization of non-submodular energy functions remains a challenging task. It is also important since many applications of graphical models turn out to require solving non-submodular energies. Since submodularity is the discrete counterpart of convexity, it would be helpful to investigate into the literature in non-convex optimization [81].

- In the MRF energy functions proposed in this dissertation, we set the parameters of the energy to be fixed values. To achieve better performance, learning the parameters from ground truth data can surely help improve the matching accuracy. Learning the parameters for the energy function remains a challenging problem [89, 124, 131, 241] since it usually involves optimization with both continuous and discrete variables. Hence it would be interesting to investigate algorithms that solve the mixed optimization problem that can outperform the EM style algorithms.

- The unified framework proposed in Chap. 3 leaves us several open problems to explore. It first allows us to design general algorithms for MRF optimization. Its connection to other approaches such as cutting plane method ([228]) is yet to be investigated.

Applications

- The surface tracking algorithm developed in Chap. 5 still suffers from sensitivity to noises. To achieve more robust surface tracking results that is applicable to low end 3D scanning data such as Microsoft Kinect, it is important to incorporate global shape prior into the existing framework. To this end, learning based approaches, such as AAM [169] based methods, can be incorporated into the existing framework. It is a challenge to balance global and local metrics to achieve the optimal tracking results.

- The analysis of dynamic expression change remains an open field due to the lack of accurate registered database. In the near future, we plan to establish a registered facial expression database, probably based on existing 3D expression data. With such registered data, we would like to explore possible dynamic features for expression analysis / recognition.

- The generic surface deformation model and the high-order MRF optimization framework proposed in Chap. 6 allows us to not only do surface registration, but also animation. One immediate application is to use the surface deformation model, namely the CDC, to do expression cloning and recognition.

- The high-order MRF optimization framework in Chap. 6 only explored the parallelism of the min-sum diffusion algorithm. It does not guarantee good convergence. Hence it would be very useful if we can find an algorithm that achieves better convergence. A possible idea is to explore non-local range in each optimization step. Another possible solution is to explore recent advances in optimization methods (e.g., [24, 68]) to achieve possibly better convergence rate and optimality.

Things that are not covered in this thesis

A system is not complete if it does not includes the complements ($\neg$). Therefore, in the very end, we throw a light on what is not covered in this dissertation that might be investigated in the future.

In the theoretical part, there remain many more theories to be connected into the bigger picture, *e.g.*, ideas in statistical physics [164], quantum physics / mechanics [165], dynamic

system [115] or economics. One very interesting direction is the study of the accumulated behaviors of multiple agent systems [221]. In computer vision, it is a de facto approach to define an energy function first to solve a problem, as mostly motivated by the MAP framework. However, as we have shown in Chap. 3, due to the boundary in computation, the energy functions are often intractable to solve, preventing us from getting better results. One way to deal with this is to employ other frameworks beyond the MAP, *e.g.*, compressive/sparse sensing [47], wavelets [163], or spectral methods [53]. Another way to possibly avoid defining a unique energy is the fields that study the behaviors of multiple agents. For example, in game theory [178, 182] and social networks [69, 180], what is of interest is how the behaviors of individuals influence each other and achieve global effects. Thus, if one considers the search for correspondences as the cumulative behavior of individual optimizers, one can certainly apply these principle to the surface registration problem (*e.g.*, application of game theory). Finally, we are also interested in designing randomized algorithms [174, 255] for optimizing high-order MRF problems.

Bibliography

[1] Graphical models definition: http://en.wikipedia.org/wiki/graphical_model.

[2] Hermann kober – application of conformal mapping: http://www.gap-system.org/ history/biographies/kober.html.

[3] *Microsoft© Kinect, 2010*.

[4] The millennium prize problems: http://www.claymath.org/millennium/.

[5] L. V. Ahlfors. *Lectures on Quasiconformal Mappings*. American Mathematical Society, 2 edition, 2006.

[6] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[7] G. Alberti, G. Bouchitté, and G. D. Maso. The calibration method for the Mumford-Shah functional and free-discontinuity problems. *Calc. Var. Partial Differential Equations*, 16(3):299–333, 2003.

[8] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, 2004.

[9] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):106–118, 2006.

[10] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[11] S. Arora, C. Daskalakis, and D. Steurer. Message passing algorithms and improved lp decoding. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 3–12, New York, NY, USA, 2009. ACM.

[12] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 339–348, Washington, DC, USA, 2005. IEEE Computer Society.

[13] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 227–236, New York, NY, USA, 2007. ACM.

[14] S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 3. *International Journal of Computer Vision*, 56:221–255, 2002.

[15] E. Balas and M. Saltzman. An algorithm for the three-index assignment problem. *Operations Research*, 39:150–161, 1991.

[16] D. Bao, S. Chern, and Z. Shen. *An Introduction to Riemann-Finsler Geometry*. Spring-Verlag, 2000.

[17] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28:24:1–24:11, July 2009.

[18] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29:40:1–40:9, July 2010.

[19] M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum*, 27:449–458, 2008.

[20] M. Ben-Chen, O. Weber, and C. Gotsman. Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 67–74, New York, NY, USA, 2009. ACM.

[21] M. Ben-Chen, O. Weber, and C. Gotsman. Variational harmonic maps for space deformation. *ACM Trans. Graph.*, 28:34:1–34:11, July 2009.

[22] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, pages 152–171, 1981.

[23] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[24] D. P. Bertsekas. Extended monotropic programming and duality, 2008.

[25] D. P. Bertsekas and H. Yu. A unifying polyhedral approximation framework for convex optimization. *SIAM Journal on Optimization*, 21(1):333–360, 2011.

[26] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992.

[27] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[28] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[29] A. I. Bobenko, P. Schröder, J. M. Sullivan, and G. M. Ziegler. *Discrete Differential Geometry*. American Mathematical Society, 2008.

[30] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[31] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[32] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pages 26–33, 2003.

[33] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, November 2001.

[34] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Trans. Graph.*, 29(4):1–10, 2010.

[35] S. C. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2007.

[36] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences*, 103:1168–1172, 2006.

[37] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1):188–197, 2007.

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

[39] M. M. Bronstein and A. M. Bronstein. Shape recognition with spectral distances. *TPAMI*.

[40] B. J. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Trans. Graph.*, 26(3), 2007.

[41] J. W. Brown and R. V. Churchill. *Complex variables and applications*. McGraw-Hill, 2009.

[42] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61:211–231, 2005.

[43] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society.

[44] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, 2009.

[45] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1048–1058, 2009.

[46] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2), 2001.

[47] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb. 2006.

[48] N. C.E., T.E.Vollmann, and J.Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[49] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[50] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2002.

[51] J. Chen and Y. Han. Shortest paths on a polyhedron. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 360–369, New York, NY, USA, 1990. ACM.

[52] E. S. Chuang. *Analysis, synthesis, and retargeting of facial expressions*. PhD thesis, Stanford, CA, USA, 2004. AAI3128633.

[53] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[54] M. M. Cohen. *A Course in Simple-Homotopy Theory*. Springer, 1973.

[55] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[56] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[57] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

[58] K. Crane, M. Desbrun, and P. SchrÃűder. Trivial connections on discrete surfaces. *Computer Graphics Forum*, 29(5):1525 – 1533, 2010.

[59] J. Dattorro. *Convex Optimization and Euclidean Distance Geometry*. Lulu.com, 2006.

[60] M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2 edition, 2000.

[61] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer, 1997.

[62] E. A. DINIC and M. A. KRONROD. An algorithm for solution of the assignment problem. *Soviet Math. Doklady*, 1:23–24, 1969.

[63] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[64] M. P. do Carmo. *Riemannian Geometry*. Birkhäuser, 1992.

[65] R. L. Dobrushin. The description of a random field by means of conditional probabilities and conditions of its regularity. *Theory of Probability and its Applications*, 13(2):197–224, 1968.

[66] D.-Z. Du, R. L. Graham, P. M. Pardalos, P.-J. Wan, W. Wu, and W. Zhao. Analysis of greedy approximations with nonsubmodular potential functions. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 167–175, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[67] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[68] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Automat. Contr.*, pages 592–606, 2012.

[69] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[70] T. Easterfield. A combinatorial algorithm. *The Journal of the London Mathematical Society*, 21:219–226, 1946.

[71] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.

[72] L. C. Evans. *Lectures on Quasiconformal Mappings*. American Mathematical Society, 2 edition, 1998.

[73] H. M. Farkas and I. Kra. *Riemann Surfaces*. Springer, 2004.

[74] M. Feldman, J. S. Naor, and R. Schwartz. A unified continuous greedy algorithm for submodular maximization. *Foundations of Computer Science, IEEE Annual Symposium on*, 0:570–579, 2011.

[75] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. Comput. Vision*, 18:99–119, May 1996.

[76] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe. Design of tangent vector fields. *ACM Trans. Graph.*, 26, July 2007.

[77] M. S. Floater. Mean value coordinates. *Comput. Aided Geom. Des.*, 20:19–27, March 2003.

[78] T. Frankel. *The Geometry of Physics–An Introduction*. Cambridge, 2 edition, 2004.

[79] M. Fromer and A. Globerson. An LP view of the m-best map problem. In *In Advances in Neural Information Processing Systems (NIPS)*, pages 567–575, 2009.

[80] S. Fujishige. *Submodular Functions and Optimization*. Elsevier, 2 edition, 2005.

[81] D. Y. Gao. *Duality Principles in Nonconvex Systems - Theory, Methods and Applications*. Springer, 1999.

[82] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[83] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *TPAMI*, 6(6):721–741, 1984.

[84] B. Glocker, H. Heibel, N. Navab, P. Kohli, and C. Rother. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *11th European Conference on Computer Vision (ECCV)*, Crete, Greece, September 2010.

[85] B. Glocker, N. Komodakis, N. Paragios, G. Tziritas, and N. Navab. Inter and intra-modal deformable registration: Continuous deformations meet efficient optimal linear programming. In *Information Processing in Medical Imaging*, Kerkrade, Netherlands, July 2007.

[86] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through mrfs and efficient linear programming. *Medical Image Analysis*, 12(6):731–741, 2008.

[87] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab. Optical flow estimation with uncertainties through dynamic mrfs. *CVPR*, 2008.

[88] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2 edition, 2002.

[89] S. Gould. Max-margin learning for lower linear envelope potentials in binary markov random fields. In *International Conference on Machine Learning*, pages 193–200, 2011.

[90] L. Grady and C. Alvino. The piecewise smooth mumford-shah functional on an arbitrary graph. *IEEE Transactions on Image Processing*, 2009.

[91] X. Gu, Y. He, and H. Qin. Manifold splines. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, SPM '05, pages 27–38, New York, NY, USA, 2005. ACM.

[92] X. Gu, S. Wang, J. Kim, Y. Zeng, Y. Wang, H. Qin, and D. Samaras. Ricci flow for 3d shape analysis. In *IEEE International Conference on Computer Vision*, pages 1–8, Los Alamitos, CA, USA, 2007.

[93] X. Gu and S.-T. Yau. Computing conformal structures of surafces. In *Communication of Informtion and Systems*, 2002.

[94] X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '03, pages 127–137, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[95] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(10):1025–1039, 1998.

[96] D. Hahnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI'03*, pages 915–920, 2003.

[97] P. Hall. On representatives of subsets. *Journal of London Math. Soc*, 10:26–30, 1935.

[98] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1971.

[99] C. Hernńdez, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *ICCV*, 2007.

[100] A. N. Hirani. *Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003.

[101] R. V. Hogg, A. Craig, and J. W. McKean. *Introduction to Mathematical Statistics*. Prentice Hall, 2004.

[102] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[103] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[104] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. Symposium on Geometry Processing, pages 1449–1457, 2008.

[105] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, Oct. 2003.

[106] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[107] M. Jin, J. Kim, and X. D. Gu. Discrete surface ricci flow: theory and applications. In *Proceedings of the 12th IMA international conference on Mathematics of surfaces XII*, pages 209–232, Berlin, Heidelberg, 2007. Springer-Verlag.

[108] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[109] A. E. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, NRC '97, pages 121–, Washington, DC, USA, 1997. IEEE Computer Society.

[110] J. Jonsson. *Simplicial Complexes of Graphs*. Springer, 2007.

[111] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26, July 2007.

[112] H. Y. Jung, K. M. Lee, and S. U. Lee. Toward global minimum through combined local minima. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, pages 298–311, Berlin, Heidelberg, 2008. Springer-Verlag.

[113] K. Kähler, J. Haber, and H.-P. Seidel. Geometry-based muscle modeling for facial animation. In *No description on Graphics interface 2001*, GRIN'01, pages 37–46, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.

[114] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[115] A. Katok and B. Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, 1996.

[116] A. Keane. CUDA (compute unified device architecture). 2006.

[117] M. J. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[118] L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.*, 25:412–438, April 2006.

[119] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1033–, Washington, DC, USA, 2003. IEEE Computer Society.

[120] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *ACM Trans. Graph. (Proc. of SIGGRAPH 2011)*, 30:79:1–79:12, August 2011.

[121] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 95(15):8431–8435, July 1998.

[122] K.-I. Ko. *Computational Complexity of Real Functions*. Birkhauser Boston, 1991.

[123] P. Kohli and M. P. Kumar. Energy minimization for linear envelope MRFs. In *CVPR*, pages 1863–1870. IEEE, 2010.

[124] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[125] V. Kolmogorov. A note on the primal-dual method for the semi-metric labeling problem. In *Technical report, June 2007*.

[126] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2006.

[127] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, pages 564–571, Washington, DC, USA, 2005. IEEE Computer Society.

[128] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE TPAMI*, 29(7):1274–1279, 2007.

[129] V. Kolmogorov and A. Shioura. New algorithms for convex cost tension problem with application to computer vision. *Discrete Optimization*, 6:378–393, November 2009.

[130] N. Komodakis. Image completion using global optimization. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 442–452, Washington, DC, USA, 2006. IEEE Computer Society.

[131] N. Komodakis. Efficient training for pairwise or higher order CRFs via dual decomposition. In *CVPR*, pages 1841–1848. IEEE, 2011.

[132] N. Komodakis and N. Paragios. Beyond loose LP-relaxations: Optimizing mrfs by repairing cycles. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 806–820, Berlin, Heidelberg, 2008. Springer-Verlag.

[133] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009.

[134] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *International Conference on Computer Vision*, 2007.

[135] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. *Comput. Vis. Image Underst.*, 112(1):14–29, 2008.

[136] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[137] J. Krarup. Quadratic assignment. *Saertryk of data*, pages 3–72, 1972.

[138] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[139] M. P. Kumar, V. Kolmogorov, and P. H. S. Torr. An analysis of convex relaxations for map estimation of discrete MRFs. *J. Mach. Learn. Res.*, 10:71–106, June 2009.

[140] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving Markov random fields using second order cone programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1045–1052, 2006.

[141] D. Kwon, K. J. Lee, I. D. Yun, and S. U. Lee. Nonrigid image registration using dynamic higher-order mrf model. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 373–386, Berlin, Heidelberg, 2008. Springer-Verlag.

[142] Y.-K. Lai, S.-M. Hu, and R. R. Martin. Automatic and topology-preserving gradient mesh generation for image vectorization. *ACM Trans. Graph.*, 28:85:1–85:8, July 2009.

[143] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *Proceeding of the European Conference on Computer Vision*, pages 269–282, 2006.

[144] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[145] S. L. Lauritzen and D. J. Spiegelhalter. *Local computations with probabilities on graphical structures and their application to expert systems*, pages 415–448. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[146] J. R. Lee and A. Sidiropoulos. Genus and the geometry of the cut graph. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 193–201, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[147] M. D. Lemonick. The geometry of music. *Time Magazine*, 2007.

[148] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008.

[149] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, 2002.

[150] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph.*, 28:175:1–175:10, December 2009.

[151] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz. Temporally coherent completion of dynamic shapes.

[152] H. Li, R. Straub, and H. Prautzsch. Fast subpixel accurate reconstruction using color structured light. In *Proceedings of the Fourth IASTED International Conference on Visualization, Imaging, and Image Processing*, pages 396–401. The International Association of Science and Technology for Development (IASTED), ACTA Press, September 2004.

[153] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1421–1430, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[154] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *ACM Trans. Graph.*, 29:32:1–32:6, July 2010.

[155] Y. Lipman and I. Daubechies. Surface comparison with mass transportation. Technical report, Princeton University, 2010.

[156] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Trans. Graph.*, 28(3):1–12, 2009.

[157] Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM Trans. Graph.*, 27:78:1–78:10, August 2008.

[158] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[159] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 28–42, Berlin, Heidelberg, 2008. Springer-Verlag.

[160] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[161] F. Luo. Combinatorial Yamabe flow on surfaces. *Commun. Contemp. Math.*, 6:765 – 780, 2004.

[162] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2007.

[163] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2 edition, 1999.

[164] F. Mandl. *Statistical Physics*. Wiley, 2 edition, 1988.

[165] F. Mandl. *Quantum Mechanics*. Wiley, 1 edition, 1992.

[166] S. Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman and Hall, 2009.

[167] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Comput. Vis. Image Underst.*, 61:295–307, May 1995.

[168] J. Matousek. *Geometric Discrepancy: An Illustrated Guide (Algorithms and Combinatorics)*. Springer, 2 edition, 2009.

[169] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60:135–164, 2003.

[170] T. Mcinerney and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. In *Computer Vision*, pages 518–523, 1993.

[171] D. N. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Springer, 1996.

[172] C. W. Misner, K. S. Thorne, J. A. Wheeler, J. Wheeler, and K. Thorne. *Gravitation*. W. H. Freeman, 1973.

[173] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, August 1987.

[174] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[175] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral Conformal Parameterization. *Computer Graphics Forum*, 27(5):1487–1494, 2008.

[176] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, 42(5):577–685, 1989.

[177] K. Murota. *Discrete Convex Analysis*. SIAM, 2003.

[178] R. B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1997.

[179] T. Needham. *Visual Complex Analysis*. Oxford University Press, 1999.

[180] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[181] M. Nikolova, S. Esedoglu, and T. F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.

[182] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 1 edition, 2007.

[183] J.-y. Noh and U. Neumann. Expression cloning. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 277–288, New York, NY, USA, 2001. ACM.

[184] M. Novotni and R. Klein. Computing Geodesic Distances on Triangular Meshes. In *The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2002.

[185] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging,Vision,and Graphics*. Springer-Verlag, 2003.

[186] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[187] S. J. Osher and S. Esedoglu. Decomposition of images by the anisotropic rudin-osher-fatemi model. *Comm. Pure Appl. Math*, 57:1609–1626, 2003.

[188] M. Ovsjanikov, Q. Merigot, F. Memoli, and L. J. Guibas. One point isometric matching with the heat kernel. *Comput. Graph. Forum*, 29(5):1555–1564, 2010.

[189] J. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, 2 edition, 2011.

[190] N. Paragios, Y. Chen, and O. D. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer, 2005.

[191] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA, 1982.

[192] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2 edition, 1988.

[193] R. Penrose. *The Road to Reality*. Vintage books, 2007.

[194] R. Penrose and W. Rindler. *Spinors and Space-Time: Volume 1, Two-Spinor Calculus and Relativistic Fields*. Cambridge University Press, 1987.

[195] W. P. Pierskalla. The multidimensional assignment problem. *Operations Research*, 16:422–431, 1968.

[196] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[197] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8):986–1004, July 2003.

[198] T. Pock, A. Chambolle, H. Bischof, and D. Cremers. A convex relaxation approach for computing minimal partitions. In *CVPR*, 2009.

[199] T. Pock, D. Cremers, H. Bishof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *ICCV*, 2009.

[200] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *ECCV*, pages 792–805, 2008.

[201] K. Polthier. Computational aspects of discrete minimal surfaces. *Global Theory of Minimal Surfaces*, 2005.

[202] T. Popa, I. South-Dickinson, D. Bradley, A. Sheffer, and W. Heidrich. Globally consistent space-time reconstruction. *Computer Graphics Forum (Proc. SGP)*, 29:1633–1642, 2010.

[203] B. Potetz and T. S. Lee. Efficient belief propagation for higher-order cliques using linear constraint nodes. *Comput. Vis. Image Underst.*, 112:39–54, October 2008.

[204] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Exact inference in multi-label crfs with higher order cliques. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2008.

[205] Y. Reshetnyak. Two-dimensional manifolds of bounded curvature. *Geometry IV, volume 70 of the Encyclopedia of Mathematical Sciences*, 70, 1993.

[206] S. Roth and M. J. Black. On the spatial statistics of optical flow. *Int. J. Comput. Vision*, 74:33–50, August 2007.

[207] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[208] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, Aug. 1999.

[209] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[210] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of ACM*, 23:555–565, 1976.

[211] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[212] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *SIGGRAPH '01*, pages 409–416, 2001.

[213] J. A. Schnabel, D. Rueckert, M. Quist, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G.

Hill, and D. J. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '01, pages 573–581, London, UK, UK, 2001. Springer-Verlag.

[214] A. Scientific. *Nonlinear Programming*. Elsevier, 2 edition, 1999.

[215] A. Shaji, A. Varol, L. Torresani, and P. Fua. Simultaneous point matching and 3D deformable surface reconstruction. In *CVPR*, 2010.

[216] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5:3–55, January 2001.

[217] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph.*, 27:110:1–110:10, December 2008.

[218] A. Sharma, R. P. Horaud, J. Cech, and E. Boyer. 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, CO, 2011.

[219] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.

[220] A. Shekhovtsov, I. Kovtun, and V. Hlaváč. Efficient mrf deformation model for non-rigid image matching. *Comput. Vis. Image Underst.*, 112:91–99, October 2008.

[221] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 1 edition, 2008.

[222] H.-Y. Shum and R. Szeliski. Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment. *International Journal of Computer Vision*, 36(2):101–130, Feb. 2000.

[223] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, 24:417–425, July 2005.

[224] R. M. Smullyan. *Diagonalization and Self-Reference*. Oxford University Press, 1994.

[225] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

[226] D. Sontag and T. Jaakkola. New Outer Bounds on the Marginal Polytope. In *Neural Information Processing Systems*, 2007.

[227] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP Relaxations for MAP using Message Passing. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

[228] D. A. Sontag. *Cutting plane algorithms for variational inference in graphical models*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2007.

[229] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *SGP*, pages 109–116, 2007.

[230] B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. *ACM Trans. Graph.*, 27:77:1–77:11, August 2008.

[231] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. The MIT Press, 2011.

[232] L. Steinberg. The backboard wiring problem: A placement algorithm. *SIAM Reiview*, 3:37–50, 1961.

[233] K. Stephenson. *Introduction to Circle Packing: The Theory of Discrete Analytic Functions*. Cambridge University Press, 2005.

[234] P. Strandmark, F. Kahl, and N. C. Overgaard. Optimizing parametric total variation models. In *CVPR*, 2009.

[235] G. Strang. Maximal flow through a domain. *Math. Prog.*, (26):123–143, 1983.

[236] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPRW '04*, volume 12, page 189, 2004.

[237] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *SIGGRAPH*, pages 399–405, 2004.

[238] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[239] R. Szeliski and S. Lavallée. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *Int. J. Comput. Vision*, 18:171–186, May 1996.

[240] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 82–87, June 1993.

[241] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 582–595, Berlin, Heidelberg, 2008. Springer-Verlag.

[242] D. Tarlow, I. Givoni, R. Zemel, and B. Frey. Graph cuts is a max-product algorithm. In *The 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, 2011.

[243] C. Teleman. *Riemann Surfaces*. Online Lecture Notes: http://math.berkeley.edu/ teleman/math/Riemann.pdf, 2003.

[244] J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. *ACM Trans. Graph.*, 30:76:1–76:10, August 2011.

[245] D. Terzopoulos and K. Waters. Physically-based facial modelling, analysis, and animation. *Journal of Visualization and Computer Animation*, pages 73–80, 1990.

[246] A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel. Intrinsic shape matching by planned landmark sampling. In *Eurographics*, 2011.

[247] A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. Isometric registration of ambiguous and partial data. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1185 – 1192, 2009.

[248] D. W. Thompson and J. T. Bonner. *On Growth and Form*. Dover Publications, 1917.

[249] N. Thorstensen and R. Keriven. Non-rigid shape matching using geometry and photometry. In *Asian Conference on Computer Vision*, 2009.

[250] Y. Tian and S. G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *ICCV'09*, pages 2303–2310, 2009.

[251] Y. Tian and S. G. Narasimhan. Globally optimal estimation of nonrigid image distortion. *International Journal of Computer Vision*, 2011.

[252] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV08*.

[253] W. Trobin, T. Pock, D. Cremers, and H. Bischof. Continuous energy minimization via repeated binary fusion. In *ECCV*, 2008.

[254] W. Trobin, T. Pock, D. Cremers, and H. Bischof. Continuous energy minimization via repeated binary fusion. *ECCV*, 5305(813396):677–690, 2008.

[255] M. M. E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[256] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27:1134–1142, November 1984.

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

[258] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[259] L. Vietoris. über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Math. Ann.*, 97:454–472, 1927.

[260] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24:426–433, July 2005.

[261] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, Dec. 1928.

[262] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[263] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[264] C. Walder, M. Breidt, H. Bülthoff, B. Schölkopf, and C. Curio. Markerless 3d face tracking. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 41–50, Berlin, Heidelberg, 2009. Springer-Verlag.

[265] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.*, 28:15:1–15:15, May 2009.

[266] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[267] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *International Conference on Computer Vision*, 2009.

[268] C. Wang, Y. Zeng, L. Simon, I. Kakadiaris, D. Samaras, and N. Paragios. Viewpoint invariant 3d landmark model inference from monocular 2d images using higher-order priors. In *ICCV*, 2011.

[269] S. Wang, Y. Wang, M. Jin, X. D. Gu, and D. Samaras. Conformal geometry and its applications on 3D shape matching, recognition, and stitching. *TPAMI*, 29(7):1209–1220, 2007.

[270] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps. In *ICCV*, 2005.

[271] Y. Wang, X. Huang, C. su Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum*, pages 677–686, 2004.

[272] O. Weber, M. Ben-Chen, and C. Gotsman. Complex barycentric coordinates with applications to planar shape deformation. *Comput. Graph. Forum*.

[273] O. Weber and C. Gotsman. Controllable conformal maps for shape deformation and interpolation. *ACM Trans. Graph.*, 29:78:1–78:11, July 2010.

[274] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. Graph.*, 30:77:1–77:10, August 2011.

[275] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[276] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 7–16, New York, NY, USA, 2009. ACM.

[277] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 7–16, 2009.

[278] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:1165–1179, July 2007.

[279] T. Werner. Revisiting the linear programming relaxation approach to gibbs energy minimization and weighted constraint satisfaction. *TPAMI*, 32:1474–1488, 2010.

[280] T. Werner. How to compute primal solution from dual one in MAP inference in MRF? *Internal Journal on Control Systems and Computers*, (2), 2011.

[281] H. Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, pages 509 – 533, 1935.

[282] L. Wietzke, G. Sommer, and O. Fleischmann. The geometry of 2D image signals. In *CVPR*, 2009.

[283] S.-Q. Xin and G.-J. Wang. Improving chen and han's algorithm on the discrete geodesic problem. *ACM Trans. Graph.*, 28:104:1–104:8, September 2009.

[284] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3D dynamic facial expression database. In *FGR*, 2008.

[285] X. Yin, J. Dai, S.-T. Yau, and X. Gu. Slit map: conformal parameterization for multiply connected surfaces. In *Proceedings of the 5th international conference on Advances in geometric modeling and processing*, GMP'08, pages 410–422, Berlin, Heidelberg, 2008. Springer-Verlag.

[286] C. Zach, M. Niethammer, and J.-M. Frahm. Continuous maximal flows and Wulff shapes: Application to MRFs. In *CVPR*, 2009.

[287] W. Zeng, D. Samaras, and X. D. Gu. Ricci flow for 3D shape analysis. *TPAMI*, 32:662–677, 2010.

[288] W. Zeng, X. Yin, Y. Zeng, Y. Lai, X. Gu, and D. Samaras. 3D face matching and registration based on hyperbolic ricci flow. In *CVPR Workshop on 3D Face Processing*, pages 1–8, 2008.

[289] W. Zeng, Y. Zeng, Y. Wang, X. Yin, X. Gu, and D. Samaras. 3D non-rigid surface matching and registration based on holomorphic differentials. In *The 10th European Conference on Computer Vision (ECCV)*, 2008.

[290] X. Zeng, W. Chen, and Q. Peng. Efficiently solving the piecewise-constant mumford shah model using graph cuts. Technical report, 2006.

[291] Y. Zeng, D. Samaras, W. Chen, and Q. Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Comput. Vis. Image Underst.*, 112:81–90, October 2008.

[292] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, 2010.

[293] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. A Generic Local Deformation Model for Shape Registration. Research Report RR-7676, INRIA, July 2011.

[294] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Intrinsic dense 3D surface tracking. In *CVPR*, 2011.

[295] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *CVPR*, 1999.

[296] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-driven shape correspondence. *Computer Graphics Forum (SGP)*, 27(5):1431–1439, 2008.

[297] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph.*, 23(3):548–558, 2004.

[298] D. zhu Du and K.-I. Ko. *Problem Solving in Automata, Languages and Complexity*. Wiley-Interscience, 2001.

Appendix

A. Standard convex relaxations for discrete MRFs

In this section, we review the standard LP convex relaxation for solving the MRF problems. First of all, let us consider the exponential family

$$p(x;\theta) \propto \exp\{-\langle\theta, \phi(x_a)\rangle\} = \exp\{-\sum_a \theta_a \phi_a(x_a)\}, \tag{7.1}$$

where x_a is the subset of variables that belong to the clique a. In the MAP-MRF problem, finding optimal configuration of the above density function is equivalent to minimizing the following energy

$$E(x) = \sum_a \theta_a \phi_a(x_a). \tag{7.2}$$

Here we only consider the situation when the variables $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ are discrete, namely $\mathbf{x} \in \mathcal{L} \times \ldots, \mathcal{L} \times \mathcal{L}, \mathcal{L} = \{1, \ldots, L\}$.

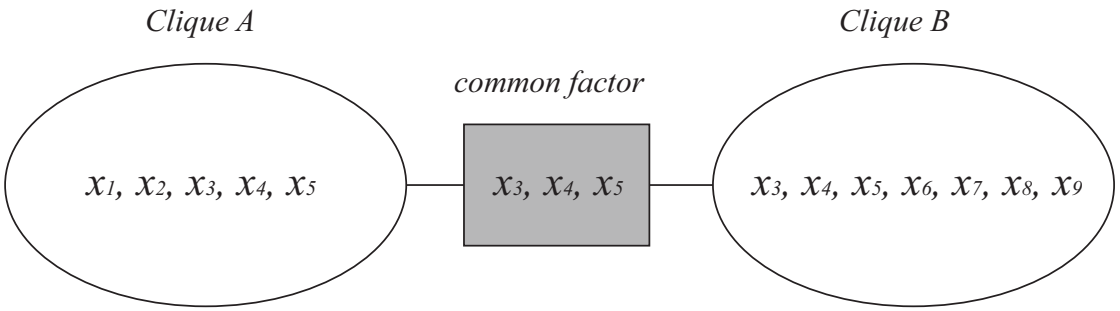


Figure 7.1: The dependency between overlapping cliques.

The first step in relaxing the energy 7.2 is to introduce the indicator variable for each clique a and its each value $\tilde{a}_a$:

$$\tau_{a;\tilde{x}_a} = \begin{cases} 1 & \text{if } x_a = \tilde{x}_a \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

Hence the energy of Equ. 7.2 can be denoted by

$$E(x) = \sum_a \sum_{\tilde{x}_a \in \text{dom}(x_a)} \theta_a \phi_a(\tilde{x}_a) \tau_{a;\tilde{x}_a}. \tag{7.4}$$

Meanwhile, additional constraints must be satisfied. For example, one can verify that the following local constraints must be satisfied:

$$\sum_{\tilde{x}_a} \tau_{a;\tilde{x}_a} = 1, \qquad \forall a \tag{7.5}$$

$$\sum_{\tilde{x}_{a\setminus b}} \tau_{a;\tilde{x}_a} = \tau_{b;\tilde{x}_b}, \qquad \forall a, b, a \cap b \neq \emptyset, \tilde{x}_a \setminus \tilde{x}_{a\setminus b} = \tilde{x}_b \setminus \tilde{x}_{b\setminus a}. \tag{7.6}$$

Hence, it can be seen that the complexity of the constraints depends both on the level of dependency among cliques and the size of each clique (Fig. 7.1).

The linear programming relaxation of the above integer programming problem can be formulated as

$$\min E(\mathbf{x}) = \sum_a \sum_{\tilde{x}_a} \theta_a \phi_a(\tilde{x}_a) \tau_{a;\tilde{x}_a}. \tag{7.7}$$

$$\text{s.t.} \sum_{\tilde{x}_a} \tau_{a;\tilde{x}_a} = 1, \qquad \forall a \in \mathcal{C} \forall a \in \mathcal{C} \tag{7.8}$$

$$\sum_{\tilde{x}_{a\setminus b}} \tau_{a;\tilde{x}_a} = \tau_{b;\tilde{x}_b}, \qquad \forall a, b \in \mathcal{C} \text{ and } a \cap b \neq \emptyset \text{ and } \tilde{x}_a \setminus \tilde{x}_{a\setminus b} = \tilde{x}_b \setminus \tilde{x}_{b\setminus a} \tag{7.9}$$

$$\tau_{a;\tilde{x}_a} \geq 0, \qquad \forall a \in \mathcal{C} \text{ and } \tilde{x}_a \tag{7.10}$$

Note that when the hyper-graph have loops, *i.e.*, $\exists a_1 \to a_2, \ldots, a_n \to a_1, a_i \cap a_{i+1} \neq \emptyset$, the solution to the relaxed problem is not equivalent to the optimal solution of the original integral programming problem. In such cases, additional constraints must be included to tighten the relaxation [132, 226, 227].

B. Primal-dual principle for pairwise MRF optimization

Now let us connect the graph based formulation to the continuous models discussed above. We shall derive a formulation for the pairwise MRF-MAP problem that is similar to the continuous formulation (Equ. 3.75). An MRF is defined on a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. And for each node $p \in \mathcal{V}$ we assign a discrete variable x_p whose value is defined in a discrete set $\{1, 2, \ldots, L\}$. The pairwise MRF is given by

$$E(\mathbf{x}|\theta) = \sum_{p \in \mathcal{V}} \theta_p(x_p) + \sum_{(p,q) \in \mathcal{E}} \theta_{pq}(x_p, x_q). \tag{7.11}$$

To relate the MRF energy 7.11 to the continuous energy 3.74, similar to the continuous case, we discretize Ω into $\mathcal{V}^\Omega$. Each variable x_p now stands for the discretized gray level Γ at $p \in \mathcal{V}^\Omega$. We introduce the binary variables $x^{\mathcal{G}}_{p,a} = [x_p = a], a \in \mathcal{K}^\Gamma$ [1], which lifts the original multi-label MRF problem into a high-dimensional binary variable in the discretized space Σ. The constraint for the new variable is

$$\mathcal{D}^q = \{\mathbf{x} | \sum_{a \in \mathcal{K}^\Gamma} x^{\mathcal{G}}_{p,a} = 1, x^{\mathcal{G}}_{p,a} \in \{0, 1\}, \forall p \in \mathcal{V}\},$$

which is consistent with the continuous constraint $\mathcal{C}$. Now the MRF energy 7.11 becomes an integer quadratic programming problem:

$$\min_{\mathbf{x}^{\mathcal{G}} \in \mathcal{D}^q} E^q(\mathbf{x}^{\mathcal{G}}) = \sum_{p \in \mathcal{V}^\otimes} \theta_{p,a} x^{\mathcal{G}}_{p,a} + \sum_{(p,q) \in \mathcal{E}^\Omega} \theta_{pq,ab} x^{\mathcal{G}}_{p,a} x^{\mathcal{G}}_{q,b}. \tag{7.12}$$

We can establish a one-to-one correspondence between the discretized upper level set function $\mathbf{x}^C$ and the MRF formulation. If we discretize the grayscale into L levels, let $x^C_{p,L+1} = 0$ for $\forall p \in \mathcal{V}^\Omega$. We have the following relation

$$x^{\mathcal{G}}_{v,j} = x^C_{v,j} - x^C_{v,j+1}, \forall v \in \mathcal{V}^\Omega, j = 1, 2, \ldots, L$$

which corresponds to the discrete version of $\partial\phi/\partial t$. The discrete version of the layer cake formula 3.73 now becomes

$$x^{\mathcal{C}}_{i,j} = \sum_{j}^{j_{max}} x^{\mathcal{G}}_{i,j}.$$

For the spatial derivative $\partial\phi/\partial x$ or $\partial\phi/\partial y$, consider the edges (p, q) in the grid Ω, we have the relation

$$|x_p - x_q| = \sum_{a \in \mathcal{K}} |(x^{\mathcal{C}}_{p,a} - x^{\mathcal{C}}_{q,a})| = \sum_{a,b \in \mathcal{K}^{\mathcal{G}}} |a - b| x^{\mathcal{G}}_{p,a} x^{\mathcal{G}}_{q,b}$$

[1] $[\cdot]$ is the iverson symbol

However, the quadratic energy 7.12 is not always convex. Therefore, a popular approach is to linearize the MRF energy by introducing the binary variable $x^{\mathcal{G}}_{pq,ab} = x^{\mathcal{G}}_{p,a} x^{\mathcal{G}}_{q,b}$. The MRF energy now becomes a integer linear programming with linear constraint:

$$E^l(\mathbf{x}|\theta) = \langle \mathbf{x}, \theta \rangle \text{ s.t. } A\mathbf{x} = b, \mathbf{x} \geq 0. \tag{7.13}$$

Similar to the continuous case, we obtain the feasible domain for the dual variable $W(\mathbf{x}) = \{\theta' | \langle \theta', \mathbf{x} \rangle \leq E^l(\mathbf{x}|\theta)\}$, which defines a vector field on the high dimensional space. It is immediate that the following equation holds $E(\mathbf{x}|\theta) = \max_{\theta' \in W(\mathbf{x})} \langle \theta', \mathbf{x} \rangle$. The MRF problem is now converted into a minmax problem

$$\min_{\mathbf{x} \in \mathcal{D}} E(\mathbf{x}|\theta) = \min_{\mathbf{x} \in \mathcal{D}} \max_{\theta' \in W(\mathbf{x})} \langle \theta', \mathbf{x} \rangle \tag{7.14}$$

If we denote the null space $N(A^T) = \{\nu_0 | A^T \nu_0 = 0\}$, it is easily verified that $\theta + \nu_0^T A \in W(\mathbf{x}), \forall \mathbf{x}$. So the following minmax problem solves a lower bound for the original MRF,

$$\begin{aligned} \min_{x \in D} \max_{\nu_0 \in N(A^T)} \langle \theta + \nu_0^T A, \mathbf{x} \rangle \leq & \min_{x \in D} \max_{\theta' \in W(x)} \langle \theta', \mathbf{x} \rangle \\ = & \min_{\mathbf{x} \in \mathcal{D}} E(\mathbf{x}|\theta) \end{aligned} \tag{7.15}$$

Here $N(A^T)$ is independent of the value of $\mathbf{x}$ so in practice it is easy to handle.

C. Theoretical analysis of the canonical distortion coefficient

In this section, we give the rigorous definitions for characterizing arbitrary diffeomorphisms between two surfaces using Riemannian geometry. Specifically, we give the precise meaning of "deformation of a unit circle into an eclipse" mentioned above and explain why it is able to represent arbitrary deformations. We also connect our canonical distortion coefficient to the Beltrami-coefficient defined in quasi-conformal mapping theory. Most of our notations and definitions are adapted from the classic textbooks [63], [78] and [5].

Diffeomorphisms, isometries and conformal maps

Given a diffeomorphism $f : \mathcal{M} \to \mathcal{N}$, the mapping between two tangent spaces $df : T_p(\mathcal{M}) \to T_q(\mathcal{N})$, where $q = f(p)$, is a linear mapping [63]. The mapping f can be fully characterized by its differential df at each point on the surface given an initial correspondence

[72]. The advantage of studying the mapping f from its first-order derivative is its simplicity since df is defined on a linear space. Within this setting, the change of the tensor metric, *i.e.*,

$$\langle \mathbf{v}, \mathbf{w} \rangle \rightarrow \langle df(\mathbf{v}), df(\mathbf{w}) \rangle_q \tag{7.16}$$

is determined by the linear transformation df. Hence, under this the linear transformation, a unit circle in $T_p(\mathcal{M})$ is mapped into an eclipse in $T_q(\mathcal{N})$. Under canonical parametrization (Def. 1), such change of metric tensor can be represented as $I \rightarrow J_{pq}^T J_{pq}$. Since $J_{pq}^T J_{pq} = O^T \text{diag}(\lambda_1, \lambda_2) O$, this deformation represents both scale change information and angle change information. Here λ_1 and λ_2 represents the scale change along the two axes and the (rotational) orthogonal matrix O represents the angle change. Equivalently, such changes can be fully characterized by the distortion of a unit circle in the tangent space at p into an eclipse in the tangent space at q. In the following, we show that the canonical distortion coefficient can fully characterize two deformation models that is prevalent in computer vision, *i.e.*, isometry and conformality.

Definition 27 *A diffeomorphism $f : \mathcal{M} \rightarrow \mathcal{N}$ is an **isometry** if for all $p \in \mathcal{M}$ and all pairs of tangent vectors $\mathbf{v}, \mathbf{w} \in T_p(\mathcal{M})$, we have*

$$\langle \mathbf{v}, \mathbf{w} \rangle_p = \langle df(\mathbf{v}), df(\mathbf{w}) \rangle_q \tag{7.17}$$

Hence, under canonical parametrization, the mapping must satisfy

$$J_{pq}^T J_{pq} = I, \tag{7.18}$$

i.e., $\lambda_1 = \lambda_2 = 1$.

Definition 28 *A diffeomorphism $f : \mathcal{M} \rightarrow \mathcal{N}$ is called a **conformal map** if for all $p \in \mathcal{M}$ and all pairs of tangent vectors $\mathbf{v}, \mathbf{w} \in T_P(\mathcal{M})$, we have*

$$\langle df(\mathbf{v}), df(\mathbf{w}) \rangle_q = \lambda^2(p) \langle \mathbf{v}, \mathbf{w} \rangle_p, \tag{7.19}$$

where λ^2 is a nonwhere-zero differentiable function on $\mathcal{M}$.

Again, under canonical parametrization, we have $J_{pq}^T J_{pq} = \lambda^2(p)$, implying $\lambda_1 = \lambda_2$.

Canonical distortion coefficient and quasi-conformal mapping

The above discussion established the link between the popular deformation models (*i.e.*, isometry and conformality) and the canonical distortion coefficient. In fact, the canonical distortion

coefficient is also closely related to the more general quasi-conformal mapping [5]. The quasi-conformal mapping studies the deformation between two planes $w = f(z)$, where $z = x + iy$ and $w = u + iv$. At a given point z_0, f induces a linear mapping of the differentials

$$du = u_x dx + u_y dy \tag{7.20}$$

$$dv = v_x dx + v_y dy \tag{7.21}$$

which we can also write in the complex form:

$$f_z = \frac{1}{2}(f_x - if_y) \text{ and } f_{\overline{z}} = \frac{1}{2}(f_x + if_y) \tag{7.22}$$

Note that this transformation is defined on two charts U_α and U_β which is assumed to have Euclidean metric. Hence one can write in classical notation

$$du^2 + dv^2 = Edx^2 + 2Fdxdy + Gdy^2 \tag{7.23}$$

with

$$E = u_x^2 + v_x^2, \quad F = u_x u_y + v_x v_y, \quad G = u_y^2 + v_y^2 \tag{7.24}$$

Under canonical parametrizations at q ($f(z_0)$), since it is Euclidean at q, the metric tensor can be represented as $du^2 + dv^2$. Hence we can establish

$$J_{pq}^T J_{pq} = \begin{pmatrix} E & F \\ F & G \end{pmatrix}. \tag{7.25}$$

In the theory of quasi-conformal mapping, the beltrami-coefficient $\mu(z) = \frac{f_{\overline{z}}}{f_z}$ can be partially determined by the eigenvalues of the matrix $\begin{pmatrix} E & F \\ F & G \end{pmatrix}$, which is equivalent to the canonical distortion coefficient. To see this, we first define

$$D_f = \frac{|f_z| + |f_{\overline{z}}|}{|f_z| - |f_{\overline{z}}|}. \tag{7.26}$$

From

$$dw = f_z dz + f_{\overline{z}} d\overline{z}, \tag{7.27}$$

we have

$$(|f_z| - |f_{\overline{z}}|)|dz| \leq |dw| \leq (|f_z| + |f_{\overline{z}}|)|dz|. \tag{7.28}$$

Hence D_f denotes the ratio of the major to the minor axis of the mapping f at z_0, *i.e.*,

$$D_f = (\frac{\lambda_1}{\lambda_2})^{\frac{1}{2}} \tag{7.29}$$

Therefore we can easily obtain

$$|\mu(z)| = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1} + \sqrt{\lambda_2}}. \tag{7.30}$$

D. Details of high-order MRF optimization algorithm

Derivation of dual optimization

In linear programming (LP), a general form of primal-dual relation is the following

$$\begin{aligned} &\min c^T x && && \max b^T y \\ &\text{s.t. } Ax = b && \Leftrightarrow && \text{s.t. } A^T y \leq c \\ &\quad x \geq 0 \end{aligned} \tag{7.31}$$

Here c and b are two vectors and A is a matrix.

Now let us consider the LP relaxation of the problem of Eq. 6.4 as derived in Sec. 6.4:

$$\begin{aligned} &\min_{\tau} \sum_{u\in\mathcal{V}} \sum_{i\in\mathcal{L}} \theta_{u;i}\tau_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} \sum_{(i,j,k)\in\mathcal{L}^3} \theta_{uvw;ijk}\tau_{uvw;ijk} && && (7.32)\\ &\text{s.t.} \sum_{i} \tau_{u;i} = 1, && \forall u \in \mathcal{V} \\ &\quad \sum_{i,j,k} \tau_{uvw;ijk} = 1, && \forall (u,v,w) \in \mathcal{F} \\ &\quad \sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}, && \forall (u,v,w) \in \mathcal{F} \text{ and } i \in \mathcal{L} \\ &\quad \tau_{u;i} \geq 0, \tau_{uvw;ijk} \geq 0, \end{aligned}$$

Introducing the dual variables

$$\sum_{i} \tau_{u;i} = 1 \qquad \rightarrow y_u \tag{7.33}$$

$$\sum_{i,j,k} \tau_{uvw;ijk} = 1 \qquad \rightarrow y_{uvw} \tag{7.34}$$

$$\sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i} \qquad \rightarrow M_{uvw;u:i}, \tag{7.35}$$

by following the primal-dual formulate of Eq. 7.31, we have the dual problem of the LP problem (Eq. 7.32):

$$\begin{aligned} &\max \sum_{u\in\mathcal{V}} y_u + \sum_{(u,v,w)\in\mathcal{F}} y_{uvw} && && (7.36)\\ &\quad y_u - \sum_{(v,w),(u,v,w)\in\mathcal{F}} M_{uvw;u:i} \leq \theta_{u;i}, && \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L} \\ &\quad y_{uvw} + M_{uvw;u:i} + M_{uvw;v:j} + M_{uvw;w:k} \leq \theta_{\theta_{uvw;ijk}}, \\ &\qquad\qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L} \end{aligned}$$

If we define the reparametrization

$$\overline{\theta}_{u;i} := \theta_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} M_{uvw;u:i}, \qquad \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L}$$

$$\overline{\theta}_{uvw;ijk} := \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L},$$

we have for any $u \in \mathcal{V}$

$$y_u \leq \overline{\theta}_{u;i} \qquad \forall i \in \mathcal{L} \tag{7.37}$$

and for all $(u,v,w) \in \mathcal{F}$

$$y_{uvw} \leq \overline{\theta}_{uvw;iijk} \qquad \forall (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L} \tag{7.38}$$

Hence the dual optimization problem of Eq. 7.36 can be equivalently formulated as

$$\max_{M} \sum_{u} \min_{i} \overline{\theta}_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} \min_{i,j,k} \overline{\theta}_{uvw;ijk} \tag{7.39}$$

$$\text{s.t. } \overline{\theta}_{u;i} = \theta_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} M_{uvw;u:i}, \qquad \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L}$$

$$\overline{\theta}_{uvw;ijk} = \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L}.$$

An efficient algorithm for finding independent face sets

Alg. 5 shows the algorithm that decomposes a third-order graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ into subsets of independent face sets defined in Sec. 4.2 in the paper. To see to complexity of this algorithm, for each iteration of step two, it take $O(|\max(\mathcal{V}, \mathcal{F})|)$ to traverse all the faces to construct a new independent face set. The total number of iterations depends on the total number of decomposed sets ($< O(|\mathcal{F}|)$). Hence the worst case complexity of this algorithm is $O(|\mathcal{V}||\mathcal{F}|)$. However in practice the number of iterations is expected to be very small.

Algorithm 5: Greedy algorithm for independent face sets.

Input : A third-order graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$.

Output : Decomposition of $\mathcal{F}$ into independent face sets $\cup_i \mathcal{F}_i$.

Step One: Initialization.

for each $f \in \mathcal{F}$ **do**
visited$[f] \leftarrow$ false
end for

Step Two: Find maximal independent face set among the un-visited faces.

$count = 0$
$i = 0$
while count $\neq |\mathcal{F}|$ **do**
$\mathcal{F}_i \leftarrow \varnothing$
for each $v \in \mathcal{V}$ **do**
$visited[v] \leftarrow$ false
end for
for each $f = (v_1, v_2, v_3) \in \mathcal{F}$ **do**
if $visited[f]$, $visited[v_1]$, $visited[v_2]$ and $visited[v_3]$ are all false **then**
$\mathcal{F}_i = \mathcal{F}_i \cup f$
$visited[v_1] \leftarrow$ true, $visited[v_2] \leftarrow$ true, $visited[v_3] \leftarrow$ true
$visited[f] \leftarrow$ true
$count = count + 1$
end if
end for
$i = i + 1$
end while