Numerical and Variational Aspects of Mesh Parameterization and Editing

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Rhaleb Zayer

Saarbrücken
2007

delivered by
EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

Datum des Kolloqiums: 17.09.2007

Dekan der Naturwissenschaftlich-Technischen Fakultät I:
Prof. Dr. Thorsten Herfet

Mitglieder des Prüfungsausschusses:
1. Gutachter: Prof. Dr. Hans-Peter Seidel
2. Gutachter: Prof. Dr. Bruno Lévy
3. Gutachter: Prof. Dr. Denis Zorin
Akademischer Mitarbeiter: Dr. Hendrik Lensch

Abstract

A surface parameterization is a smooth one-to-one mapping between the surface and a parametric domain. Typically, surfaces with disk topology are mapped onto the plane and genus-zero surfaces onto the sphere. As any attempt to flatten a non-trivial surface onto the plane will inevitably induce a certain amount of distortion, the main concern of research on this topic is to minimize parametric distortion. This thesis aims at presenting a balanced blend of mathematical rigor and engineering intuition to address the challenges raised by the mesh parameterization problem. We study the numerical aspects of mesh parameterization in the light of parallel developments in both mathematics and engineering. Furthermore, we introduce the concept of quasi-harmonic maps for reducing distortion in the fixed boundary case and extend it to both the free boundary and the spherical case. Thinking of parameterization in a more general sense as the construction of one or several scalar fields on a surface, we explore the potential of this construction for mesh deformation and surface matching. We propose an "on-surface parameterization" for guiding the deformation process and performing surface matching. A direct harmonic interpolation in the quaternion domain is also shown to give promising results for deformation transfer.

The increasing need for processing large amounts of geometric data delivered by modern acquisition technology poses new challenges concerning robustness, efficiency, and solution accuracy. This work makes the following contributions to the research effort towards addressing these challenges.

In mesh parameterization, we start with one of the most intuitive and promising approaches, namely the *angle based flattening*, which attempts to flatten surfaces based on angular constraints in the parametric plane. The nonlinearity of the constraints gives rise to a challenging but well posed optimization problem. We present and discuss several numerical approaches for improving the convergence rate of this method. We take advantage of the structure of the system matrix in order to decouple the problem making the matrix equations easier to solve. Also, the use of a modified version of the Uzawa algorithm typically used in saddle point problems allows for a better performance. Furthermore, close inspection of the problem reveals that it is possible to approximate the nonlinear constraints by linear ones thus yielding for the first time a linear version of the angle based parametrization. The latter can be solved in a straightforward manner using the normal equation for underdetermined systems of linear equations.

An alternative way for preserving angles is based on the notion of conformal maps. We explore the potential of differential geometry by analyzing several methods for constructing a discrete version of such maps. In order to contribute to the understanding of the discretization schemes of the continuous characterization, an overview of several existing methods in mesh processing is given in the light of parallel developments in other fields. The influence of the discretization on the parameterization is substantiated by a simple but conclusive benchmark.

We further rely on differential geometry for quantifying parametric distortion and propose appropriate methods for reducing it. In this context, we introduce the concept of quasi-harmonic maps which elegantly allows for handling distortion directly as a tensor thus allowing for directional as well as areal distortion to be taken into account under a variational framework. The potential of quasi-harmonic maps is shown for the case of fixed boundary parameterization and further extended to the free boundary case as well as spherical parameterization in curvilinear coordinates.

Thinking of planar parameterization as the construction of two scalar fields on a given surface, it seems natural to think of more general forms of parameterization arising from the construction

of one or several scalar fields on the surface. This leads to a more flexible notion which will be called in this work ”on-surface parameterization”. The usefulness of this notion is substantiated by applications such as mesh deformation and surface matching. We present an interactive method for applying deformations to a surface mesh while preserving its global shape and local properties. Two surface editing scenarios which conceptually differ in the specification of deformations are discussed: Either interpolation constraints are imposed explicitly, e.g., by dragging a subset of vertices, or, implicitly by mimicking the deformation of a reference surface. The contribution of this work is a novel approach for interpolation of local deformations over the manifold and for efficiently establishing correspondence to a reference surface from only few pairs of markers. As a general tool for both cases, a harmonic field is constructed to guide the interpolation of constraints and to find correspondence required for deformation transfer. We show that our approach fits nicely in a unified mathematical framework, where the same type of linear operator is applied in all phases.

Zusammenfassung

Eine Flächenparameterisierung ist eine globale bijektive Abbildung zwischen der Fläche und einem zugehörigen parametrischen Gebiet. Gewöhnlich werden Flächen mit scheibenförmiger Topologie auf eine Kreisscheibe und Flächen mit Genus Null auf eine Sphäre abgebildet. Das Hauptinteresse der Forschung an diesem Thema ist die Minimierung der parametrischen Verzerrung, die unweigerlich bei jedem Versuch, eine nicht triviale Fläche über einer Ebene zu parameterisieren, erzeugt wird. Diese Arbeit strebt zur Behandlung des Parametrisierungsproblems eine ausgeglichene Mischung zwischen mathematischer Präzision und ingenieurwissenschaftlicher Intuition an. Wir behandeln dabei die numerischen Aspekte des Parameterisierungsproblems im Hinblick auf die aktuellen parallelen Entwicklungen in der Mathematik und den Ingenieurwissenschaften. Weiterhin führen wir das Konzept der quasi-harmonischen Abbildungen ein, um die Verzerrung bei gegebenen Randbedingungen zu verringern. Anschließend verallgemeinern wir dieses Konzept auf den sphärischen Fall und auf den Fall mit freien Randbedingungen. Durch allgemeinere Betrachtung der Parameterisierung als Konstruktion eines oder mehrerer skalarer Felder auf einer Fläche ergibt sich ein neuer Ansatz zur Netzdeformation und der Erzeugung von Flächenkorrespondenzen. Wir stellen eine "on-surface parameterization" vor, welche den Deformationsprozess leitet und Flächenkorrespondenzen erstellt. Darüber hinaus zeigt eine direkte harmonische Interpolation in der Domäne der Quaternionen auch vielversprechende Resultate für die Übertragung von Deformationen.

Die zunehmende Notwendigkeit große Mengen geometrischer Daten zu verarbeiten stellt eine Herausforderung dar, um Faktoren wie Robustheit, Effizienz und Lösungsgenauigkeit zu erzielen. Diese Arbeit liefert folgende Beiträge zu den Forschungsbemühungen, die sich diesen Herausforderungen widmen:
Bei der Behandlung des Flächenparameterisierungsproblems beginnen wir mit einem der intuitivsten und vielversprechendsten Ansätze, nämlich dem *angle based flattening*, welches eine Fläche parameterisiert, indem Winkelbedingugen in der parametrischen Ebene festgelegt werden. Die Nichtlinearität der Bedingungen erzeugt ein schwieriges aber gut gestelltes Optimierungsproblem. Wir stellen einige numerische Ansätze für die Verbesserung der Konvergenzrate dieser Methode vor. Wir nutzen dabei die Struktur der Matrix, um das Problem zu entkoppeln, welches die Lösung der Matrixgleichungen vereinfacht. Auch der Gebrauch einer abgewandelten Version des Uzawa Algorithmus, der gewöhnlich in den Sattelpunktproblemen verwendet wird, erlaubt eine Verbesserung der Leistung. Außerdem zeigte eine sorgfältige Kontrolle des Problems die Möglichkeit auf, die nichtlinearen durch lineare Bedingungen zu approximieren, so dass zum ersten Mal eine lineare Version der "angle-based flattening" Parameterisierung formuliert wurde. Die Letztere kann in einer direkten Weise, basierend auf der Normalengleichung des unterbestimmten Gleichungssystems, gelöst werden.
Eine andere Methode zur Winkelerhaltung ist durch die Verwendung konformer Diagramme gegeben. Wir erforschen das Potential der differentialen Geometrie, indem wir einige Methoden für das Konstruieren einer diskreten Variante der konformen Diagramme analysieren. Um zum Verständnis der Methoden zur Diskretisierung der kontinuierlichen Varianten beizutragen, geben wir einen Überblick über einige vorhandenen Methoden in der Netzbearbeitung unter Berücksichtigung der Entwicklungen in anderen Gebieten. Der Einfluss der Diskretisierung auf die Parametrisierung wird durch einen einfachen aber aufschlussreichen Test dargelegt. Des weiteren nutzen wir die differentiale Geometrie um die parametrische Verzerrung zu quan-

tifizieren und schlagen geeignete Methoden vor diese zu reduzieren. In diesem Kontext stellen wir das Konzept der Quasi- Harmonischen Abbildung vor, welches erlaubt, die Verzerrung elegant als Tensor zu betrachten. Dadurch ist es möglich, in einem System sowohl die gerichtete als auch die Flächenverzerrung mit einzubeziehen. Das Potential der Quasi-Harmonischen Abbildungen wird für Parametrisierungen mit festen Rändern dargelegt und erweitert für Parametrisierungen mit freien Rändern und Kugelparametrisierungen mit krummlinigen Koordinaten.
Betrachtet man die planare Parametrisierung als Kombination aus zwei Skalarfeldern, so erscheint es natürlich, über allgemeinere Formen der Parameterisierung durch eine beliebige Anzahl skalarer Felder nachzudenken. Dies führt zu einem flexibleren Begriff des Problems, welches im weiteren Rahmen dieser Arbeit ”on-surface” Parametrisierung genannt wird. Die Nützlichkeit dieser Definition wird des weiteren mit Anwendungen zur Netz-Verformung und Flächenanpassung untermauert. Wir präsentieren eine interaktive Methode, um Verformungen auf ein Flächennetz anzuwenden, wobei zugleich dessen globalen Form und lokale Eigenschaften erhalten werden. Zwei Flächenbearbeitungs-Szenarien werden genauer dargelegt, die sich konzeptuell in der Spezifikation der Verformung unterscheiden: Entweder werden Interpolationsbedingungen explizit definiert, z.B. durch das Verschieben einer Untermenge der Netz-Knoten, oder Verformungen eines Referenz-Netzes werden nachgeahmt. Der Beitrag dieser Arbeit ist ein neuer Ansatz zur Interpolation von lokalen Verformungen über eine Fläche und zur effizienten Bildung einer Korrespondenz zu der Referenz-Fläche mit Hilfe einer kleinen Anzahl von Markern. Als allgemeines Werkzeug für beide Fälle wird ein harmonisches Feld berechnet, welches zur Interpolation der Randbedingungen und zur Bestimmung der Korrespondenz für den Verformungstransfer dient. Wir zeigen, dass unser Ansatz in einen einheitlichen mathematischen Rahmen passt, in dem dieselbe Klasse eines linearen Operators für alle Phasen verwendet wird.

To my parents

Acknowledgements. This work could not have got off the ground without the support of many persons. The first and foremost debt of gratitude is owed to Hans-Peter Seidel. His trust, vision, and commitment are the ground on which this inquiry could gradually develop.
I also thank the members of my thesis committee, Bruno Lévy and Denis Zorin, for many valuable suggestions and for their interest in reviewing the work in this thesis.
During the time I have spent at Max Planck Institute, I was fortunate to work closely with several talented people who left their fingerprints on the development of the ideas in this work. Unfortunately I can only mention few of their names. Special thanks to Christian Rössl whose experience and knowledge helped much in the shaping of this work. Common projects together with Ioannis Ivrissimtziss, Holger Theisel, Hitoshi Yamauchi, Stefan Gumhold, Zachi Karni, and Christian Theobalt opened wider possibilities for me and gave me the chance to learn from their broad expertise. The special environment at MPI had a great impact on the progress of this work. Credit and a big round of thanks go to all the AG4 members thanks to whom the academic and social life could not have been more pleasant.
Keeping to the tradition, the people who keep you from becoming insane are mentioned last. I am most grateful to my family, and to Birgitta, Adel, and Trish for their tremendous understanding and ongoing inspiration and support.

Contents

1

Introduction

> The notion of a mathematical and engineering approach as it is meant here, is roughly as follows: Mathematical approach is fully deductive; the statements are precise, using exactly formulated assumptions. Engineering approach is inductive; relies on intuition and is heuristic. Neither the claims nor the assumptions are precise; nevertheless they are supported by extensive numerical experimentations on benchmark problems of engineering importance.
>
> I. Babuška in [Bab94]

In the field of computer graphics in general and in geometry processing in particular the boundaries between mathematics and engineering are blurry. Sound mathematical theory is well appreciated but has to compete or cooperate with heuristics for breeding efficient algorithms capable of processing large data sets and producing results of high visual quality, if possible at interactive rates. Surface parametrization is a peculiar problem where mathematical theory rules out the existence of an ideal solution for general surfaces, more precisely, except for the case of developable surfaces there is no bijective, distance preserving mapping, from the plane onto a given surface. This lack of optimal solution brought forward several quantifications and definitions of what is a good parameterization. Subsequently, except for a few methods which have deeply rooted theoretical background, the majority of existing techniques are of heuristic nature. Nonetheless, in most instances, visual inspection remains de facto the only reliable and tacitly approved evaluation for parameterization quality.
The aim of this thesis is to present a balanced blend of mathematical rigor and engineering intuition to address the challenges raised by the mesh parameterization problem as well as some aspects of mesh editing. Our investigation begins with the fundamental problem in mesh parametrization which addresses the flattening of surfaces with disc-like topology onto the plane, as illustrated in figure (1.1). For the parameterization to be valid it is imperative that the planar representation does not fold onto itself. Surface parameterization has been known to map makers for a long time and the significant number of map projections they developed underscores the difficulties raised by the problem, especially when attempting to achieve a flattening with low distortion. In mesh processing the task is relatively more involved as the surfaces under study exhibit generally a far more complex geometry than the spherical earth representation. Furthermore, the discrete nature of the geometric data makes it not so straightforward to take full advantage of available differential geometry tools. Ideally it would be preferable to establish isometric parameterizations which preserve distances. However, the existence of such maps is limited to the rarely encountered case of developable surfaces. Thus a parameterization cannot

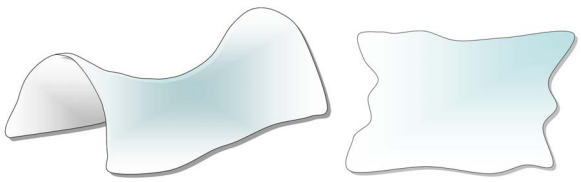


Figure 1.1: *A 3D surface and its 2D flattening.*

be achieved without inducing a certain amount of area and angles distortion. The impact of angle distortion on visual quality has steered the development of angle preserving parameterization schemes. A direct approach along these lines consists of setting up the problem as a constrained nonlinear optimization in terms of angles. We discuss the numerical aspects of this problem and propose several schemes for improving its convergence properties. We further reformulate the problem allowing for the first time for a linear version of angle based parameterization.
An alternative approach to preserving angles is based on the notion of conformal maps, which in contrast to the previous approach stems from the continuous setting and is well studied in differential geometry and complex analysis. The key element in this formulation is that the problem can be stated as a variational problem and full advantage can be taken of existing discretization schemes of differential operators. In the current work we study a fair amount of these schemes and discuss their suitability for mesh parameterization. Furthermore, as the field of mesh processing has become sufficiently standardized, we touch upon relevant historic aspects of the discretization problem to relate current research to parallel advances in mathematics and engineering.
One of the major shortcomings of angle preserving maps is the lack of control over area distortion, which in certain instances might be crucial to applications such as remeshing and texture mapping. We build upon the variational framework for developing the concept of quasi-harmonic parameterization in order to account for area distortion without compromising visual quality. The potential of quasi-harmonic maps is shown for the case of fixed boundary parameterization and is further extended to the free boundary case as well as spherical parameterization in curvilinear coordinates. Thinking of planar parameterization as the construction of two scalar fields on a given surface, it seems natural to think of more general forms of parameterization arising from the construction of one or several scalar fields on the surface. This leads to a more flexible notion which will be called in this work ”on-surface parameterization”. The usefulness of such notion is substantiated by applications to mesh deformation and surface matching. As it is generally desired to edit and deform surface meshes directly, we propose an ”on-surface parameterization” for guiding the deformation process and also for surface matching throughout the use of a generalized form of harmonic weights. Additionally, a direct harmonic interpolation in the quaternion domain is also shown to give promising results for deformation transfer.

1.1 Related work

The recent technical advances in data acquisition make large data sets describing complex and highly detailed geometries easily accessible to the research community and the general public. The strive for efficient and reliable methods for processing this bulk of geometric information fosters research on topics such as texturing, editing, and motion transfer. An exhaustive account of the large body of literature on these subjects is well beyond the scope of this brief review. In the following particular emphasis will therefore be placed on topics directly related to the

present work.

Planar parameterization

Mesh parameterization has emerged within the last decade as an important tool in geometry processing, a fact which is reflected by the significant amount of research on the topic. We briefly overview the related work in the field and we refer to [FH05] for a more general survey. Parameterization methods generally map a surface (patch) onto a planar parameter domain. A common approach is the explicit minimization of a certain local distortion measure to guide the parameterization process. The resulting numerical problems are often of a nonlinear type, e.g., [DMK03, HG00, SSGH01, SdS00, SCGL02, ZMT05, YBS04], and considerable research effort is dedicated to increasing the efficiency of the related optimization procedures. On the other hand, [EDD$^+$95, HAT$^+$00, PP93, DMA02, LPRM02, GY03] offer an elegant alternative which preserves angles and proceeds by solving the Laplace equation. Thus, the resulting parameterization could be considered as a discrete version of conformal maps. Variants of this linear setting propose alternative discretizations of the Laplace-Beltrami operator [Flo97, Gus02, SdS02, Flo03]. In general, the shape of the boundary in the parametric domain has a significant influence on the overall distortion. Therefore it is crucial to take it into account in the parameterization setup in order to reflect the geometry of the original surface patch.
The most prominent approach to surface parameterization which features a freely evolving boundary is the angle based flattening [SdS00]. It is primarily an angle preserving parameterization, which formulates the problem as a constrained optimization fully specified in terms of the inner angles of the planar representation. As the method is of nonlinear nature, a fair amount of work has been devoted to the design and speeding up of appropriate numerical solvers, e.g., the most recent [SLMB05] and the references therein.
The simplicity of linear methods based on the Laplace equation makes them generally more attractive and several schemes for reducing their boundary distortion have been developed. [LKL02] embed the surface patch in a larger one by heuristically growing a “virtual boundary”, which to some extent absorbs distortion induced by the convex boundary setting. Similarly, the nonlinear approach in [ZMT05] applies virtual boundaries in combination with “scaffolding triangles”. Another interesting approach was proposed in [PB00] which attaches long springs to the boundaries of the mesh and literally stretches out the mesh on the plane by appropriate control of the stiffness of the strings. Alternatively, [DMA02, LPRM02] propose two powerful methods which turn out to be mathematically equivalent although they start from different setups. The former applies Neumann boundary conditions to the Laplace equation and the later works directly on the Cauchy-Riemann equations. Both methods require fixing (at least) two boundary vertices for the system to admit a solution. [SLMB05] note that in practice, unsatisfactory results are often obtained for geometrically complex boundaries. [KGG05] discuss the design of such boundary conditions in the presence of constraints.

Spherical parameterization

A direct extension of conformal planar parameterization to the spherical domain can be formulated as a minimization problem of the Dirichlet energy over the surface subject to the constraint

$$||\mathbf{x}||^2 = 1$$

for all vertices. The nonlinearity of the constraints makes the spherical setting more involved

than its planar counterpart.
A straightforward way to solve this optimization problem is the Gauss-Seidel method, in which each iteration uses a local relaxation step, e.g., tangential Laplacian relaxation, followed by back projection onto the sphere. This type of solution was carried out in [Ale00, KVLS99]. The result is used as a starting point for computing a minimal Möbius transform in [GWC+04, GY02]. A principal problem is that a minimum can be reached for degenerate configurations, e.g., with vertices slipping over the sphere until triangles collapse. Heuristic solutions to this issue such as imposing stopping criteria or introducing additional boundary conditions fall short from providing satisfactory results in general. A more promising alternative is the careful analysis of the discrete objective function which tries to account for the particularities of the spherical setting and blends several measures for controlling distortion [FSD05].
A more theoretical study of the problem was carried out in [GGS03] and devises sufficient requirements for generating provably bijective maps. An alternative constrained minimization problem proposed in [SGD03] adapts the planar angle based flattening method to the spherical setting. While theoretically interesting, both approaches are computationally too expensive to be of general use in practice.
Methods based on the multi-resolution paradigm, which follow the original algorithmic frameworks outlined in [EDD+95, HGC99, LSS+98], can also be applied to the spherical setting. In particular the choice of the base mesh and the objective function can be tuned for spherical parameterization as proposed in [Kan05, PH03].
Another breed of methods resorts to the existence of simple maps from the plane onto the sphere. In [HAT+00], a triangle is cut from the mesh and then the whole mesh is mapped into a triangular boundary. The resulting planar parameterizing is lifted to the sphere through an inverse stereographic projection. Besides the high distortion, this method generally suffers from foldovers. This is due to the fact that the boundary of the mesh in the planar domain is considered to extend implicitly to infinity. In order to overcome such limitations, the method of [SYGS05] cuts the mesh into two halves and maps each half to a circle. The resulting planar embeddings are mapped onto the sphere and serve as starting point for subsequent nonlinear optimization on the sphere with an appropriate distortion measure. The latter work is motivated by [GGS03] and yields relatively faster computation time. Most related to the present work is the method proposed in [BGK92] which first establishes an initial parameterization based on curvilinear coordinates. This solution is further improved using a nonlinear optimization in the spherical domain. However, the highly nonlinear nature of this optimization problem makes it unstable for practical use. An improvement was proposed in [QBH+00] by using a hierarchical implementation for speeding up convergence. However, the reported performance still penalizes the approach as a whole.

Mesh deformation

The design and deformation of smooth surfaces is a well-studied problem, and a multitude of approaches exist in the literature. In the following, we focus on deformation preserving the local detail of a shape which is modified on a global scale.
Multi-resolution techniques are well-known and probably exhibit the most prominent approach to this problem, e.g., for surface modeling [ZSS97, KCVS98, GSS99, BK04]. Shape information is decomposed encoding geometric detail relative to local frames [FB88] which are defined with respect to a smooth base surface. After deformation of the smooth base, the local frames are implicitly adapted (i.e., rotated) and detail is reintegrated into the reconstruction.
Alternative methods for various kinds of surface editing rely on a relative representation [Ale03]

which captures local shape detail, namely Laplacian coordinates. However, the inherent lack of rotation and scale invariance of the linear operator requires local transformation of the differential coordinates. To account for this, local rotations are estimated either from normal averaging over an initial reconstruction [LSC+04] or from embedding a simultaneous least-squares optimization for linear approximants to rotations and isotropic scales in the linear reconstruction operator [SLC+04].
Here, we note that both of the latter approaches, multi-resolution methods and differential coordinates, transform (rotate) their respective shape detail information in such a way that the reconstruction consistently preserves local detail while the overall shape follows the global deformation. We can interpret this transformation as updating the mappings of local regions from the source to the target shape – hence, as finding meaningful local deformations.
In fact, the recent work of [YZX+04] directly applies this notion: Generalizing the concept of Poisson image editing [PGB03], local deformations [Bar84] are specified and then propagated over the surface mesh. The resulting deformation field and additional conditions capturing smoothness (including continuity) and local shape detail yield a linear system whose solution defines the desired target shape.

Deformation transfer

So far, we discussed deformation for direct, interactive surface editing. Deformation transfer [SP04] is a different metaphor applied to mimic a reference deformation of one triangulated shape on another one. Here, local deformations of a target are sampled from the deformation of a reference surface; given a corresponding deformation field, a linear operator is used for the reconstruction similarly to the mesh editing approach discussed above. This process is steered by pairs of corresponding markers which are used to establish meaningful correspondence between the overall shapes using an iterated closest point algorithm to deform the target into the initial reference geometry.
Sumner and Popović [SP04] point out that no bijective mapping is necessary for establishing meaningful correspondence, in fact, a many-to-many mapping is generally obtained. While this seems sufficient for deformation transfer, we remark that there exist techniques to construct constrained bijective maps between surfaces [KS04, SAPH04], which provide a computationally more expensive alternative.
In addition to explicit shape manipulation, we mention free form deformation methods [SP86, Coq90] which achieve surface deformation by warping the surrounding 3-space. This space-warp can possibly be defined using (local) rotational constraints [LKG+03] or weight functions parameterized with respect to geodesic distances on the surface [BK03] for improving the propagation of constraints. In the context of deformation transfer, as-rigid-as possible shape interpolation [ACL00] defines a piecewise linear space-warp over a triangulated domain, corresponding to deformation of the boundary.
Finally, we note that the advantageous properties of harmonic fields have been used most recently for quadrilateral remeshing [DKG05].

1.2 Objectives of current work

The general objective of this work is to develop a complete solution to fundamental aspects of the surface parameterization problem. This solution should allow for robust and efficient handling of large data sets. In our investigation we start from a purely discrete method which is the angle

based parameterization. In fact this method does not have any continuous counterpart and does not rely on any advanced differential geometry concepts. We then explore the potential of differential geometry by analyzing several methods for constructing discrete conformal maps. We further rely on differential geometry for quantifying distortion and propose appropriate methods for reducing it in the planar and the spherical case. Using the same differential framework we then develop practical tools for mesh editing and deformation transfer.
The key aspects addressed are

- the robust and efficient solution of the angle based parameterization problem
- the analysis of existing discrete conformal maps
- the quantification and reduction of parameterization distortion
- the efficient and robust solution of the spherical parameterization problem, and
- the investigation of ”on-surface parameterization” potential for mesh editing, shape matching, and deformation transfer.

Research over the last two decades has touched upon these topics to a varying degree. However robustness, efficiency and, solution accuracy remain generally hard to achieve. In this work special attention is given to the reformulation of some existing problems and the introduction of new concepts. In the light of a new representation of parameters, many involved problems turn out to be more tractable and can be tackled in a straightforward manner. This aspect is illustrated in the proposed solution to the angle based parameterization and the spherical parameterization problems. In the effort of distortion reduction we introduce the concept of quasi-harmonic maps which elegantly allows for handling distortion directly as a tensor. In this way, directional as well as areal distortion are taken into account under a variational framework. Intimately related problems to mesh parameterization are the interpolation of local deformations over manifolds and the correspondence between meshes from only few pairs of markers. As a general tool for both scenarios, a harmonic field is constructed to guide the interpolation of constraints and to establish the shape matching for deformation transfer.

1.3 Overview

The remainder of this thesis is organized as follows:
Chapter (2) discusses several numerical procedures for solving the nonlinear angle based flattening problem. In chapter (3), the whole problem is reformulated as a system of linear equations and appropriate solution methods are presented. In the context of conformal maps, a panoramic view of existing discretizations of the Laplace operator is given in chapter (4) with emphasis on historic aspects of their development. Quasi-harmonic maps are introduced in chapter (5) and an extension to the free boundary parameterization is presented in chapter (6). Spherical parameterization is discussed in chapter (7). Applications of harmonic fields on surfaces for mesh editing and deformation transfer are presented in chapter (8). A brief description of implementation strategies is given in chapter (9), and the conclusions drawn from the present work are summarized in chapter (10).
Parts of this thesis have been previously published in these papers [ZRS03, ZRS04b, ZRS04a, ZRS05a, ZRS05b, ZCKS05, ZRS06].

2

Numerical Aspects of Nonlinear Angle Based Parameterization

Research on surface parameterization is handicapped by the fact that differential geometry rules out the existence of isometric maps from the plane onto a given surface except for the special case of developable surfaces, see e.g., [Kre91]. Considering the inexistence of an optimal solution to the parametrization problem, an alternative goal which is not less attractive nor less challenging is the construction of maps which preserve angles, better known as conformal maps.

In the effort of developing angle preserving maps, it seems natural to formulate the problem directly in terms of the interior angles of the planar mesh. The first step into this direction in modern mesh parametrization research was undertaken by Sheffer and de Sturler [SdS00] and is known as *angle based flattening* (ABF). The ABF algorithm constructs such a parameterization by minimizing an objective function which penalizes the angular distortion of the planar mesh with respect to the angles of the original mesh. A set of linear and nonlinear equality constraints on the planar angles guarantees the validity of the parameterization. These constraints have been extensively studied in graph theory in [BV96] and have been also used for the construction of geodetic triangulation since the 18th century. However, as these constraints affect only internal angles, they do not prevent the boundary from having global self-intersection. The authors of [SdS00] propose an additional post-processing approach to handle edge crossings at the boundary. Each post-processing step first identifies the nodes causing intersections in the flat mesh, then imposes additional constraints on the local configurations in order to avoid intersections. The flat mesh is recomputed each time as a solution of the updated nonlinear system. The post-processing algorithm is repeated until no more intersections are found.

The complexity of the constrained optimization arising from the ABF setup makes finding a solution in reasonable time a challenging task. This has prompted the development of several numerical schemes to speed up the convergence of the original algorithm [SdS00] by using preconditioning [LdSS$^+$01] and smoothing [SdS02] and most recently by combining a hierarchical approach along with an intelligent processing of the sub-matrices associated with problem [SLMB05]. A common trait to these methods is that they target improving convergence rates mainly at the level of the numerical solvers and do not alter the original nonlinear setup.

We approach this problem from a slightly different angle by identifying the main causes for poor convergence within the setup of the constrained problem itself. First, the post-processing used for handling global boundary intersections might be costly. Any detected global intersections require incorporating new constraints and solving the whole nonlinear system as many times as needed. We take advantage of a characterization of convex planar drawings of tri-connected graphs to eliminate boundary intersections in the first place. This way we can steer or even avoid post-processing. Furthermore, we note that the nonlinear equations in the ABF method lead to a

dense sparsity pattern of the Hessian matrix of the system which in turn weighs heavily on the computational cost. We show how the convergence can be improved alternatively by a simple yet effective transformation of the problem that relaxes the nonlinear equality constraints. In fact, the Hessian becomes diagonal and its sparsity pattern becomes independent of the valences of the vertices of the input mesh.
Along with these modifications to the original setup we discuss several numerical approaches for efficiently handling the arising matrix equations. Previously, general iterative solvers which do not take full advantage of the structure of the system matrix have been proposed in [SdS00, SdS02, LdSS$^+$01]. As the system of equations is symmetric we test the performance of several iterative solvers specifically tailored for this type of systems. In order to further take advantage of the matrix structure we propose two iterative approaches. The first approach is based on the fact that decoupling the system equations yields easier matrix equations. The second approach is based on the analysis of saddle point problems and relies on a modified Uzawa algorithm for efficiently carrying out the numerical optimization. Lastly, we propose an experimental approach which treats the optimization problem as an undetermined nonlinear system of equations.

2.1 Conventions

Throughout this work, we try to restrict ourselves to the essential amount of formalism. In this chapter, the following notations are used:

- N is the total number of interior mesh angles.
- α_i^* ($i = 1, \ldots, N$) denote the angles of the *original* mesh.
- α_i are the corresponding angles of the ***flat*** mesh. As these are the variables of the optimization problem, then in this context, the more usual notation x_i is used as an alternative.
- v denotes the central vertex in a centered drawing of a *wheel*, i.e. of its 1-neighborhood. d is the number of direct neighbors of v or its *valence*. α_j ($j = 1, \ldots, d$) refer to the angles at v, while β_j and γ_j denote the opposite left and right angles of a face with central angle a_j, respectively. All faces are oriented counter-clockwise.
- Variables and functions without subscripts may refer to multivariate vectors as explained by the context.

2.2 Characterization of drawings of planar angle graphs

Sheffer and de Sturler [SdS00] addressed the problem of the validity of the planar embedding by requiring the following consistency condition on the set of positive angles of the planar mesh:

- *Vertex consistency*

  For each internal vertex v, with central angles $\alpha_1, \ldots, \alpha_d$:

$$\sum_{i=1}^{d} \alpha_i - 2\pi = 0 \tag{2.1}$$

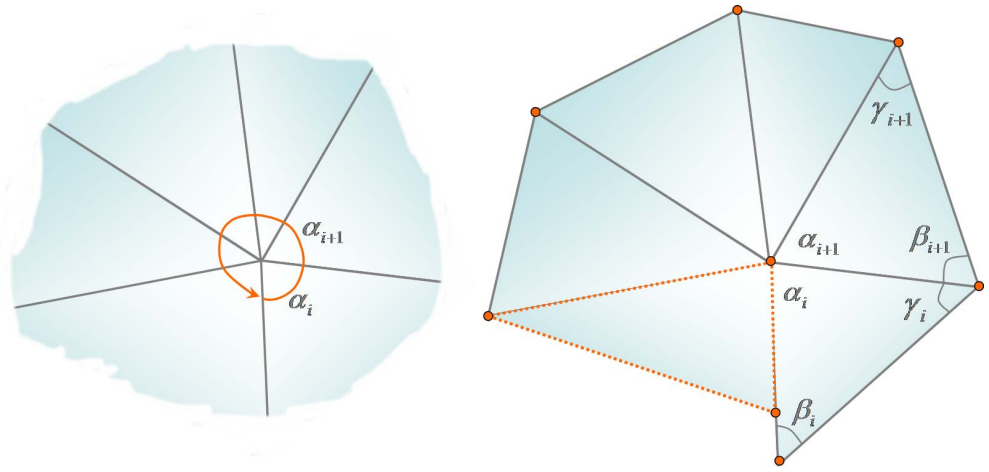


Figure 2.1: *Illustration of angle validity conditions. Vertex consistency (left) guarantees planarity, and wheel consistency guarantees closed vertex rings.*

– *Triangle consistency*

For each triangular face with angles α, β, γ the face consistency:

$$\alpha + \beta + \gamma - \pi = 0 \tag{2.2}$$

– *Wheel consistency*

For each internal vertex v with left angles β_1,..,β_d and right angles $\gamma_1, \ldots, \gamma_d$:

$$\prod_{i=1}^{d} \frac{\sin(\beta_i)}{\sin(\gamma_i)} = 1 \tag{2.3}$$

Condition (2.1) enforces the planarity of vertex rings whereas condition (2.3) enforces the triangle sine rule over a vertex ring and guarantees the closedness of the ring. Failure to satisfy this condition yields the situation illustrated in figure (2.1). These conditions guarantee the centered embedding of internal vertices without any overlapping of the interior edges. However, they do not prevent the overlapping of boundary edges. This issue is a well-studied in graph theory [BV96, Gar98]. Di Battista and Vismara provide a characterization of the convex planar straight line drawing of a tri-connected graph for a given set of positive angles [BV96]. Their minimal constraints for the planarity of the graph impose in addition to (2.1), (2.2), and (2.3) the following condition:

– *Convex external face condition*

For each external vertex v, with internal angles $\alpha_1, \ldots, \alpha_d$:

$$\sum_{i=1}^{d} \alpha_i \leq \pi. \tag{2.4}$$

Condition (2.4) guarantees the *convexity of the boundary* and hence prevents boundary overlapping. Note that the inequality (2.4) prevents local and global self-intersection simultaneously. So it does not only prevent adjacent boundary edges from overlapping, but

(a) (b) (c) (d)

Figure 2.2: *Flattening an α-shaped mode: (a) Original mesh. (b) The flattened mesh with boundary control coefficient $t = 2$. (c) $t = 1.1$. (d) $t = 1.03$. (The views are scaled differently.)*

it also guarantees that the boundary loop as a whole does not cross itself. For the local configuration it would in fact be sufficient to require the following weakened condition to hold:

– *Adjacent boundary edges consistency*
For each external vertex v, with internal angles $\alpha_1, \ldots, \alpha_d$:

$$\sum_{i=1}^{d} \alpha_i \leq 2\pi. \tag{2.5}$$

This prevents adjacent boundary triangles from crossing each other. But, condition (2.5) is not strong enough to globally enforce a valid mesh with no boundary intersections as shown in figure (2.2.b).
For a better understanding and better control of the boundary behavior, we propose to multiply the left hand side in (2.4) by a positive scalar t, formally

$$\sum_{i=1}^{d} \alpha_i \leq t\pi. \tag{2.6}$$

The scalar t can be interpreted as *boundary control coefficient* that steers the convexity of the boundary. A lower bound for this factor can be derived using discrete curvature measure. Consider the angular defect of the flat mesh can be expressed as:

$$\sum_{v=1}^{n} (\pi - A_v) = 2\pi, \tag{2.7}$$

where A_v is the sum of angles at vertex v and n is the number of boundary vertices. By a simple calculation, we establish the lower bound

$$t_0 = 1 - 2/n.$$

The trivial case is a single triangle, its angles cannot be all smaller than $\pi/3$.
We experimented with different values for t, and summarize the following interpretations that can be used as reference for choosing appropriate values for t:

– $t > 2$ results in the classic ABF method without. No adjacent edge overlapping or boundary self-crossing is taken into consideration.

– $1 < t \leq 2$ prevents adjacent edges from overlapping, but does not necessarily prevent global self-intersections of the boundary loop. We experienced such cases only for "boundary-heavy" (with respect to ratio of boundary to inner vertices, e.g. figure (2.2)) surfaces with non-trivial geometry.

– $t = 1$ globally prevents the boundary loop from self-intersection for any valid input mesh, note that this suffices to induce a convex boundary .

– $t_0 < t < 1$ forces the boundary to become concave.

Figures (2.2) and (2.3) illustrate the behavior of the boundary for different values of t. We can take advantage of these facts in order to avoid an iterative post-processing and thus have better control over the convergence of the constrained optimization problem. In the next sections we show how this problem with the additional inequalities included can be solved efficiently.

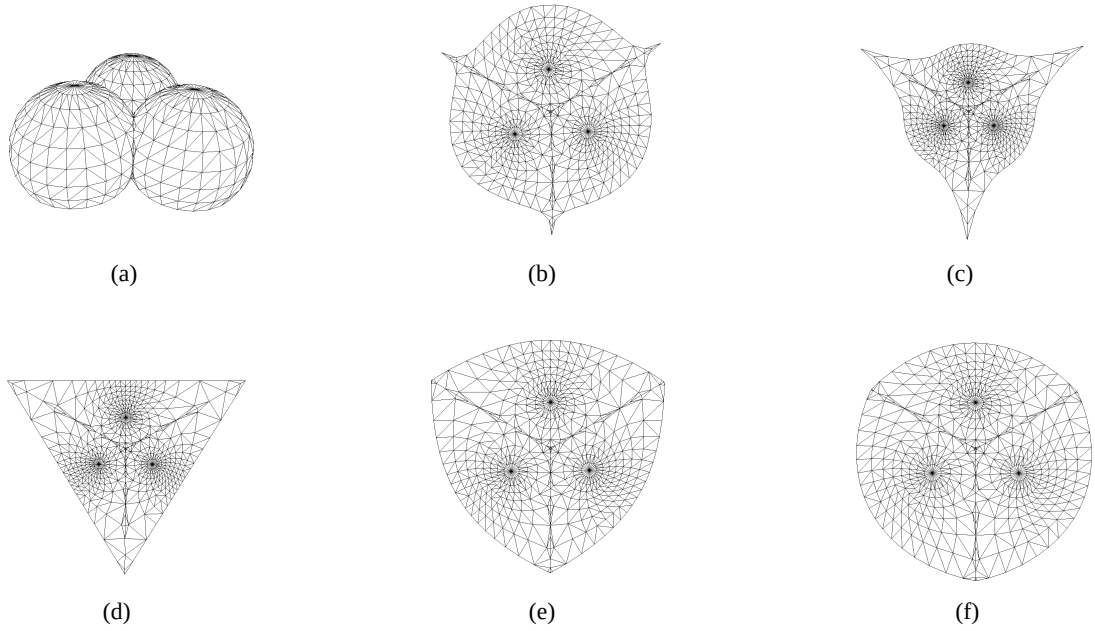


Figure 2.3: *Effect of the boundary control coefficient* t *on the* 3-balls *model. (a) Original mesh. (b) Flat mesh for* $t \geq 2$ *(ABF). (c)* $t = 1.05$*. (d)* $t = 1$ *(convex boundary ABF). (e)* $t = 0.98$*. (f)* $t = 0.968$*.*

2.3 Constrained optimization problem

A general approach for establishing a surface parameterization consists of minimizing an objective function $f(x)$ that quantifies distortion with respect to certain quality criteria. As the validity of the flat mesh is guaranteed by the angle constraints of section (2.2). A typical choice of such functions is the weighted angle distortion [SdS00]

$$f(x) = \sum_{i=1}^{N} w_i (x_i - a_i)^2 \tag{2.8}$$

with the weights $w_i = \frac{1}{a_i^2}$. The variables a_i represent the *optimal angles* of the flat mesh, which are

$$a_i = \begin{cases} \alpha_i^* \frac{2\pi}{\sum_{i=1}^{d} \alpha_i^*} & \text{around an interior vertex} \\ \alpha_i^* & \text{around a boundary vertex.} \end{cases}$$

We can now formulate the optimization problem as

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h(x) & = 0 \\ & g(x) & \leq 0, \end{array} \tag{2.9}$$

where g and h are multivariate functions of the equality (2.1), (2.2), (2.11) and the inequality constraints (2.6) respectively.
Large constrained optimization systems of the form (2.9) are still open problems in the field of nonlinear optimization [BN98]. The adequacy of a minimization method depends on the properties of the objective function as well as on the constraints.
In order to solve the optimization problem we use the method of Lagrange multipliers as it guarantees the exact satisfaction of constraints. We handle the inequality constraints by means of the so-called *active set* approach, a variant of Newton-like methods. It transforms inequalities to equalities, which are generally easier to handle.
The active set is defined as the set of indices for which the inequality constraint (2.4) is active. Formally

$$A(x, \mu) = \{i | g_i \geq -\frac{\mu_i}{c}, i = 1, ..., r\},$$

where μ_i is the Lagrange multiplier associated with g_i, and c is a fixed positive scalar.
The active set approach converts inequality constraints to equality constraints by altering the Lagrange multipliers associated with them. If a constraint does not figure in the active set, its associated multipliers are set to zero. Otherwise it is treated as an equality constraint. The numerical advantage of this method is that as the iterates get closer to the solution, the active set becomes more and more stable. A detailed description of the active set method can be found in [Ber82].
In every Newton iteration the following system is solved

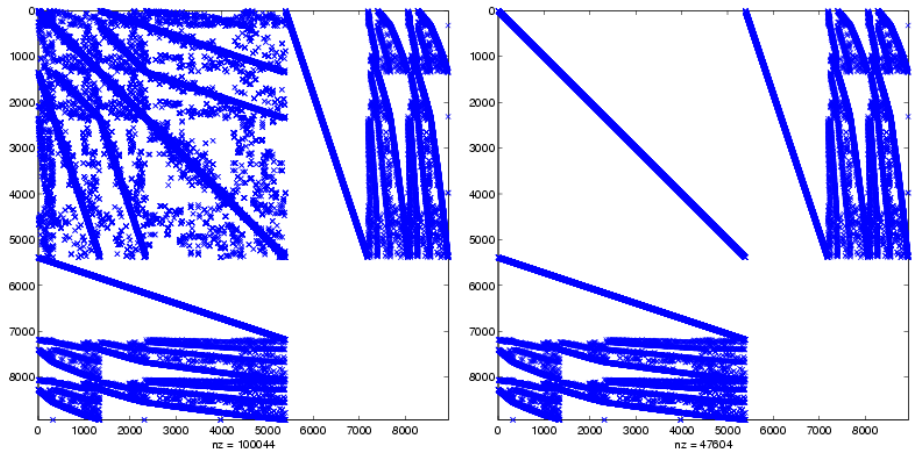
$$\begin{bmatrix} \nabla_{xx}^2 L & J_h^T & J_g^T \\ J_h & 0 & 0 \\ J_g & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \mu_h \\ \Delta \mu_g \end{bmatrix} = - \begin{bmatrix} \nabla_x L \\ h \\ g \end{bmatrix} \tag{2.10}$$

where the Lagrangian L is given by

$$L = f(x) + \mu_h^T h(x) + \mu_g^T g(x).$$

In the classic ABF algorithm, the computation of the Hessian matrix $\nabla_{xx}^2 L$ involves finding the second derivatives of the products involved in condition (2.3). The resulting matrix is sparse, but it still contains a considerable number of non-zero elements (cf. figure (2.4.a)). This number depends largely on the valences of the input mesh vertices.
Instead, we propose to use a *modified wheel condition* (2.11). Since the angles are strictly positive we can safely rewrite condition (2.3) as

$$\sum_{i=1}^{d} \log(\sin \beta_i) - \log(\sin \gamma_i) = 0. \tag{2.11}$$



(a) (b)

Figure 2.4: *System matrices of equation (2.10) generated from the* ear *model using the (a) original wheel condition. (b) simplified wheel condition. The Diagonal Hessian brought the number of nonzero elements from 100044 down to 47607.*

The virtue of this modification resides in the fact that it yields a diagonal Hessian matrix.

$$\nabla^2_{xx} L = \text{diag}(f''(x_i) + m_i \frac{-1}{\sin^2(x_i)}),$$

where m_i is the linear combination of the Lagrange multipliers involved with x_i in condition (2.11). The amount of computation and effort by the iterative solvers is hence reduced considerably. The Hessian can be computed efficiently as this reduction also avoids the estimation of complex derivatives with all the floating error they may induce. Figure (2.4) illustrates the structure of a typical system matrix and the improvement induced by the *modified wheel condition.*

2.4 Solution of the matrix equation

2.4.1 Iterative solvers

Although the system matrix established in the previous section is not not necessarily positive-definite, it still enjoys symmetry and a diagonal Hessian. We can exploit this structure by using adequate iterative solvers such as MINRES or SYMMLQ both developed by Paige and Saunders [PS75] for symmetric matrices, instead of the non-symmetric GMRES and BiCGStab that were used in [SdS00, LdSS$^+$01]. The latter solvers have higher cost per iteration and may suffer from breakdowns or simply stagnate while MINRES and SYMMLQ have a relatively cheap cost per iteration which is just 4 axpys [1] higher than the iteration cost of the conjugate gradient method. Another alternative, which is relatively inexpensive, is the CGNR algorithm introduced by Hestens and Steifel [HS52]. A key benefit in our case is that there is no need to transpose the system matrix as it is symmetric. The cost per iteration is just one matrix-vector multiplication higher than the cost of the conjugate gradient method. A comparison of the convergence of these iterative solvers for typical meshes is given in section (2.5).

[1] The term axpy denotes the addition of a scalar multiple of a vector to a vector i.e. vector1+=scalar*vector2

In order to guarantee that the resulting angles remain within the positive domain during the Newton iterations, we can reject negative iterates and append increased weights to the corresponding angles in a similar fashion to [SdS00]. However, in our experiments we hardly ever run into this situation.

2.4.2 Matrix decoupling based solutions

The current approach relies on the analysis of the structure of the system matrix in (2.10). Keeping the modified wheel condition in mind, we take advantage of the underlying system structure in order to decouple the system into two simpler matrix equations. The optimization problem at hand can be restated as the following general matrix system

$$\begin{aligned} Ax + B^t\lambda &= f, \\ Bx &= g. \end{aligned} \tag{2.12}$$

In our ABF setting (2.10) the matrix A is the diagonal Hessian and B is the sparse Jacobian matrix associated with the equality constraints. Then the system matrix S can be expanded as

$$S = \begin{bmatrix} A & 0 \\ B & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} A & B^t \\ 0 & I \end{bmatrix}, \tag{2.13}$$

where $M = BA^{-1}B^t$. According to Sylvester's law of inertia the eigenvalues of S are the same as the eigenvalues of A and M. This suggests that the system is highly indefinite as both A and M are not positive definite. The alternative approach to directly dealing with S is to decouple the system equations in order to reduce the computational effort. In order to address this numerical problem we propose two approaches:

MINRES

Since the matrix A is diagonal, its inverse can be easily obtained. The problem (2.12) now reduces to solving the following system of equations

$$\begin{aligned} x &= A^{-1}(f - B^t\lambda), \\ BA^{-1}B^t\lambda &= BA^{-1}f - g. \end{aligned}$$

The matrix $M = BA^{-1}B^t$ is symmetric, but necessarily positive definite. This rules out the existence of a unique solution as the matrix might be singular. The system can be solved using the MINRES algorithm, which is the equivalent of the conjugate gradient method for general symmetric matrices. The advantage of using MINRES over other existing iterative algorithms resides in the fact that it converges for definite, indefinite or singular cases and avoids break-ups or stagnation [PS75]. Since the matrix size might be large when dealing with large meshes, it is not desirable to directly perform matrix-matrix multiplication unless an efficient matrix package is accessible. The algorithm can be implemented without explicit computation of the matrix M, by only performing matrix-vector multiplication within the MINRES method especially that the matrix A is diagonal. This method reduces the size of the matrix problem to a smaller matrix problem of only approximately twice the number of triangles in the mesh.

Modified Uzawa algorithm

In the following approach we rely on the similarity of the system equation (2.12) with general saddle point problems. In saddle point problems, the matrix A is in general positive definite, which is not the case in our setting. However, we have the advantage that the inverse of A is immediate. A widely used approach for solving saddle point problems relies on the Uzawa algorithm [AHU58]. We cannot apply the Uzawa algorithm directly as it depends on the condition number of the matrix M. As an alternative we take advantage of direct inversion of matrix A for using the conjugate directions [Bra01]. This yields the modified Uzawa algorithm given below. It should be noted that the matrix multiplications involving the diagonal matrix A should be treated as element by element multiplication, the matrix A can be stored as a vector.

Modified Uzawa algorithm

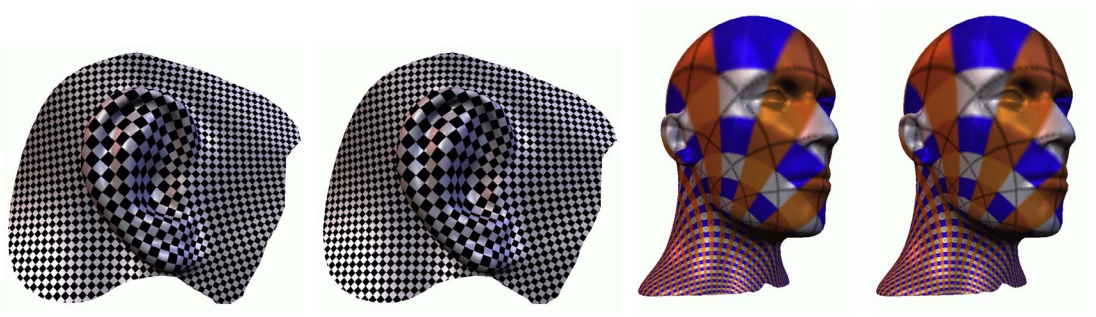
1. Initialize
   $x_0 = A^{-1}(f - B^t\lambda)$,
   $d_0 = Bu_0 - g$,$q_0 = -d_0$.

2. Repeat until convergence

   - $p_k = B^t d_k, h_k = A^{-1}p_k$,
   - $\alpha_k = \frac{<q_k,q_k>}{<p_k,h_k>}$,
   - $\lambda_{k+1} = \lambda_k + \alpha_k d_k$,
   - $x_{k+1} = x_k - \alpha_k h_k, q_{k+1} = g - Bx_{k+1}$
   - $\beta_k = \frac{<q_{k+1},q_{k+1}>}{<q_k,q_k>}$,
   - $d_{k+1} = -q_{k+1} - \beta d_k$

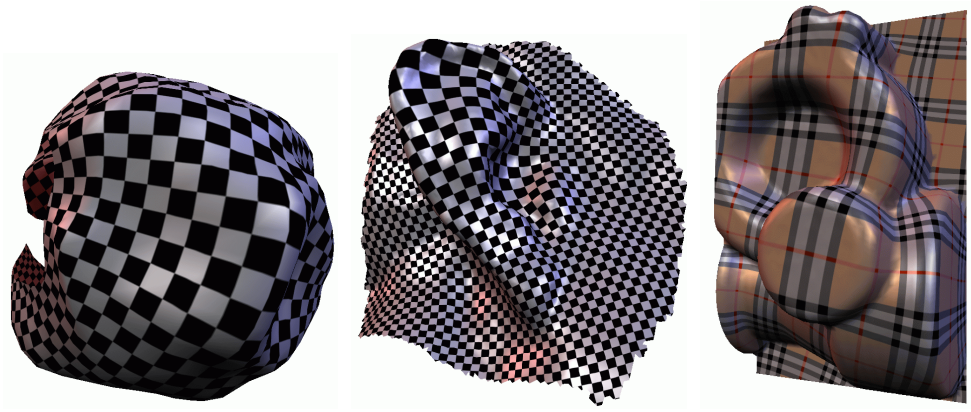
2.4.3 Underdetermined nonlinear system based approach

As the method of Lagrange multipliers is a local optimization method, the solution it provides is a local minimum that is dependent on the initial guess. In other words it is the closest minimum to the initial guess. Since the initial guess we provide is very close to the solution as only few Newton iterations are needed for convergence. We can assume that any feasible point that is close to the initial guess mentioned above gives a good estimate for the solution and would yield a low angular distortion.
With this consideration in mind, the problem can be restated as how to get a feasible point. The idea is to have a null objective function, i.e set $f = 0$. This means that we reduce the problem to solving the underdetermined system of equality and inequality constraints. This change leads to considerable speed up of convergence as there is no extra load from the objective function. In the following, we call this solution the underdetermined solution and the one using the angular distortion functional the minimization solution (literally speaking both solutions are



(a) (b) (c) (d)

Figure 2.5: *Comparison of underdetermined (a, c) and minimization (b, d) solutions.*



(a) (b) (c)

Figure 2.6: *Textured models: Clumpy (a) Large ear (b) and Mechanical part (c). Notice the quasi-conformality of the parameterization.*

just approximate solutions).
In general, the difference between the minimization and the underdetermined solution is hardly noticeable. Figure (2.5) shows a comparison between such solutions. Table (2.1) shows the numerical performance of the two methods.
Our experiments with several different initial starting guesses suggest that, if we are looking only for a topological mapping of the mesh to the plane, we can get a very fast feasible solution by setting the initial guess to zero or to $\frac{\pi}{3}$ and the objective function to zero. This solution does not reflect the geometry of the mesh and might not be suited for texture mapping.
Another alternative method for finding a feasible point would be to use least squares methods for solving underdetermined nonlinear problems. However, as these methods do not guarantee the exact satisfaction of the constraints for large problems, they failed in general to produce valid parameterizations.

2.5 Results and discussion

We applied the proposed algorithms to a set of different triangular meshes. Tables (2.1) and (2.2) summarize the numerical results of the proposed methods, all timings were mea-

Model	CGNR		SYMMLQ		MINRES	
	minim.	*under.*	*minim.*	*under.*	*minim.*	*under.*
3 Balls (1032△)	66	1	21	2	7	2
Ear (1796△)	159	3	44	9	11	7
Man head (5420△)	> 900	26	292	69	126	56
Mech. part (7938△)	> 999	86	> 999	245	> 999	192
Large ear (24914△)	> 999	237	> 999	654	> 999	522

Table 2.1: *Comparison of runtime (in seconds) of the full matrix based minimization and the underdetermined system based method using different iterative solvers.*

sured on a 1.7 GHz Intel Xeon CPU. The resulting parameterizations are shown in figures (2.5), (2.6), and (2.7). The parameterization time depends on the number of triangles, on the geometry as well as on the connectivity of the input mesh. Our measurements rely on a simple self-implemented matrix library, and we expect our algorithms to perform much faster using highly optimized matrix packages.

We observe that the Uzawa algorithm outperforms the MINRES in our experiments. In general, the decoupling of the system equations provides an efficient means for improving the convergence.

Furthermore, the numerical data suggests that the underdetermined method in association with the CGNR algorithm delivers high quality quasi-conformal parameterizations in very competitive time. However, this method does not come with any guarantees and the approximation used is completely heuristic. In the next chapter we show how to develop a well justified approximation and construct a method which overcomes most of the numerical difficulties encountered in this chapter.

	MINRES	Uzawa
Ear (1796△)	3	1
Man head (5420△)	26	18
twists (6K△)	2	1
Goldfeather (10K△)	16	7
Goldfeather (24K△)	74	31
foot (20K△)	346	115

Table 2.2: *Runtime (in seconds) using different iterative solvers in the decoupled matrix case. For most models, 3 to 4 Newton iterations were needed.*

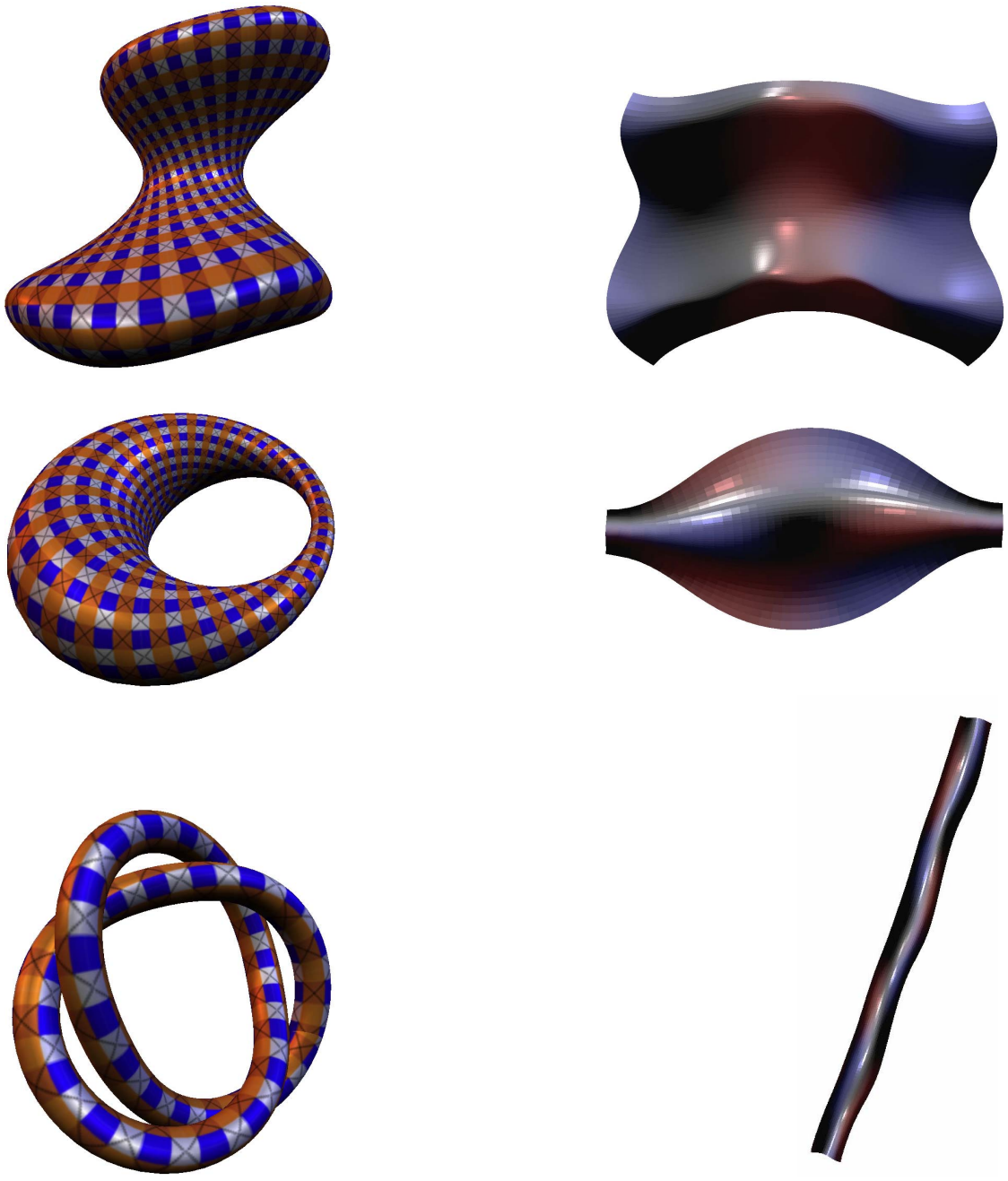


Figure 2.7: *Textured implicit surfaces and their corresponding planar embedding obtained using the modified Uzawa algorithm: From top to bottom: Goldfeather surface [GI04], ring cyclide, twists.*

3

Linear Angle Based Parameterization

> Once the mathematical equations governing a problem have been established, then excepting very simple cases, they must be solved using an approximate method. The type of approximation used is important as it affects the accuracy of the results and the economy of the solution.
>
> C. A. Brebbia and S. Walker in [BW80]

With the ever increasing computational power delivered by modern processors, it is possible to address a wide range of nonlinear problems in a reasonable time. This sheer power still has to deal with the increased size of data dictated by the strive for more detailed problem representations. This brings forward the need for efficient and reliable numerical tools allowing the redesign or reformulation of these problems in more tractable way. Angle based flattening belongs to this category of nonlinear problem and the numerical experiments conducted in chapter (2) suggest that the problem remains challenging. Recently, a combination of hierarchical structure with an intelligent matrix decoupling approach was proposed in [SLMB05] allowing for increased performance. Nevertheless most of the proposed approaches in the literature address mainly numerical issues at the level of the arising matrix equation and do not touch upon the setup of the original problem.
In this chapter, a new setup of the problem is laid out. The standard procedure adopted for establishing the parametric representation is to set up the nonlinear optimization problem in terms of the unknown planar angles and approximate the minimum of the Lagrangian using a Newton based approach (see chapter (2)). Instead, the new proposed procedure addresses the problem in terms of the angle error, more specifically the angle difference between the optimal solution and the initial guess. In this setup the nonlinear constraints are approximated by linear ones. This is well justified as the error induced by this linearization is quadratic in terms of the error in angles. The windfall of this new representation is that the problem need not be addressed as a constrained optimization but as an underdetermined system of linear equations. The latter is equivalent to a weighted least norm problem and can be solved using the normal equation. Alternatively a quadratic programming approach might be used as well for enforcing error bounds.

3.1 Ancient technique, new problem

In this section, we propose an alternative formulation of the problem, that leads to a linearization of the constraints. Linearization was already used in previous methods (e.g. ABF++). However,

in our case, before linearizing the constraints, we carefully reformulate the problem in terms of alternative variables, that will make this linearization so accurate that solving single linear system will converge to the solution directly without requiring multiple Newton steps used in previous work. In more details, our approach is based on the notion of *error adjustement*, i.e. it uses the relative error of estimation of the angles rather than their absolute values.
The idea of working with error in estimations is not new and goes back to the era of Gauss who used the technique for predicting the position of the comet Ceres[1]. The problem of error adjustment is known for fueling many quarrels between mathematicians such as Gauss and Legendre. More importantly it is at the origin of the principle of least squares which marks a turning point in the history of computing and statistical analysis. In the following the notion of estimation error is used to reformulate the angle based parameterization problem.
Let us denote the ideal angles which solve the parametrization problem by α^* and the initial guess as α , the estimation error is then given by

$$\alpha^* \ = \ \alpha \ + \ e_\alpha \tag{3.1}$$

The variables α_i represent an initial estimation of the angles of the flat mesh and will be discussed later in this chapter.
In this setup the constraints on the planar angles read :

– *Vertex consistency*

For each internal vertex v, with central angles $\alpha_1, \ldots, \alpha_d$:

$$\sum_{i=1}^{d} e_i = 2\pi \ - \ \sum_{i=1}^{d} \alpha_i \tag{3.2}$$

– *Triangle consistency*

For each triangular face with angles α, β, γ the face consistency:

$$e_\alpha + e_\beta + e_\gamma = \pi \ - \ (\alpha \ + \ \beta \ + \ \gamma) \tag{3.3}$$

– *Wheel consistency*

Based on the logarithmic modification introduced in [ZRS04b], we have for each internal vertex v with left angles β_1,..,β_d and right angles $\gamma_1, \ldots, \gamma_d$:

$$\sum_{i=1}^{d} \log\left(\sin\beta_i \ + \ e_{\beta_i}\right) - \log\left(\sin\gamma_i \ + \ e_{\gamma_i}\right) = 0. \tag{3.4}$$

The changes introduced so far do not affect the nature of the linear conditions. On the other hand, the nonlinear expression in (3.4) looks as if it just got more complicated. However, the Taylor expansion of $\log(\sin(\alpha \ + \ e))$ can be written as

$$\begin{aligned} \log(\sin(\alpha \ + \ e)) \ &= \log(\sin(\alpha)) + \cot(\alpha)\, e \\ &\quad - \ \frac{1}{2}(1 \ + \ \cot^2(\alpha))\, e^2 \\ &\quad + \cdots \end{aligned} \tag{3.5}$$

[1] As of August 24, 2006, Ceres has been promoted to the status of a 'dwarf planet' by the International Astronomical Union.

Inspection of this series reveals that we can safely use the approximation

$$\log(\sin(\alpha + e)) \backsimeq \log(\sin(\alpha)) + \cot(\alpha)\, e. \tag{3.6}$$

The error induced by this approximation depends quadratically on the error in angle e. In other words, for a small error e in the angle estimation, the error in the consistency constraint is even smaller (e.g. an angle error of the order of 10^{-3} induces an error of order 10^{-6}). This point is of utter importance to the method and it is in fact similar to the linear approximation generally used in finite elements for approximating potential energy as discussed in e.g. [Bra01]. Considering that even in the most general case enforcing equality constraints amount to a minimization up to a certain reasonable accuracy, our approximation is then well justified. We will also backup this claim with numerical experiments in the results section.
In the light of this new approximation, the nonlinear equation (3.4) can be then replaced by following linear variant

$$\sum_{i=1}^{d} \cot(\beta_i)\, e_{\beta_i} - \cot(\gamma_i)\, e_{\gamma_i} = \sum_{i=1}^{d} \log(\sin\gamma_i) - \log(\sin\beta_i) \tag{3.7}$$

The term on the right hand side measures the error in the wheel consistency condition induced by the initial estimation. Similarly the right hand sides of equations (3.2), (3.3) measure triangle consistency and the angular deficit respectively. In this way, given an angle estimation α, we describe the error induced on the constraints as linear function of the estimation error e.

3.2 Numerical solution

At this stage a least norm solution to the resulting underdetermined system of linear equations can be readily obtained through the normal equation. This setup however is evenhanded as it treat all angles in the same way and at times, this may cause instabilities for very small and very large angles. One straightforward approach consists of using additional bounds on the error e and solving the system using standard techniques e.g. Matlab^{TM}Optimization Toolbox. However, as this work is geared towards simple implementation, it is more interesting to maintain the new gains from the linearization of the constraints and associate an objective function with the constraints which allows for introducing additional weights to control the errors in similar fashion to the original ABF [SdS00]. The weighted objective function described in the following subsection allows for a balanced treatment of angles by penalizing large angles and enforcing smaller ones.

3.2.1 Normal equation setup

In this subsection, We aim at minimizing a weighted error objective function while enforcing the equality constraint. In the light of the new representation in terms of the error, the objective function

$$F(\alpha) = \sum_{i=1}^{N} \frac{1}{\alpha_i^2} (\alpha_i^* - \alpha_i)^2, \tag{3.8}$$

can be stated as:

$$\text{minimize} \sum_{i=1}^{N} \frac{1}{\alpha_i^2} e_i^2 \quad \text{subject to } Ae = b \tag{3.9}$$

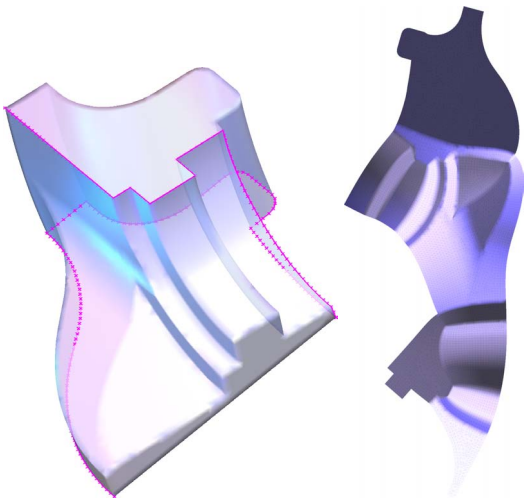


Figure 3.1: Parameterization of the fan disk model ($13K\Delta$). Solution runtime ($0.15s$).

using a simple change of variables

$$r_i = \frac{e_i}{\alpha_i} \tag{3.10}$$

In matrix notation this change of variables gives $e = D_\alpha\ r$, where $D_\alpha = \text{diag}(\alpha_i)$ is the diagonal matrix with the angles as entries. Thus the angle parameterization problem reads as simple as

$$\text{minimize } ||r||^2 \quad \text{subject to } Cr = b \tag{3.11}$$

This is now clearly a least-norm problem. The size of the matrix $C = A\ D_\alpha$ is $(n_t + 2 \cdot n_i) \times (3 \cdot n_t)$, where n_t is the number of triangles and n_i is the number of internal vertices. As the equality constraints are independent, the matrix C has full rank, and it ensures that the least-norm problem has a unique solution (see e.g. [Lue69]) :

$$r = C^T\ (C\ C^T)^{-1} b \tag{3.12}$$

Thus, the problem can be solved by finding a solution to the normal equation

$$(C\ C^T)x = b \tag{3.13}$$

After solving this equation, r can be obtained as $r = C^T\ x$. The angles of the mesh in the parametric domain can be obtained by substituting back in equations (3.10) and (3.1).

3.2.2 Choice of initial estimation

In order to reduce error in the above presented method, the choice of the initial estimation is very important as it directly affect the global error. For this purpose it is imperative that very large and very small angles do not force the solution out of the $(0, \pi)$ domain. Setting α equal to the original angles of the mesh yields valid parameterizations in most cases. However, it is not difficult to tailor cases which yield invalid angles. In order to enforce a valid solution for general cases, we set a threshold on the error of fan angles of vertices and also on angles at the vicinity of 0 and π. When the error associated with a fan is large (e.g. spikes), we replace the original

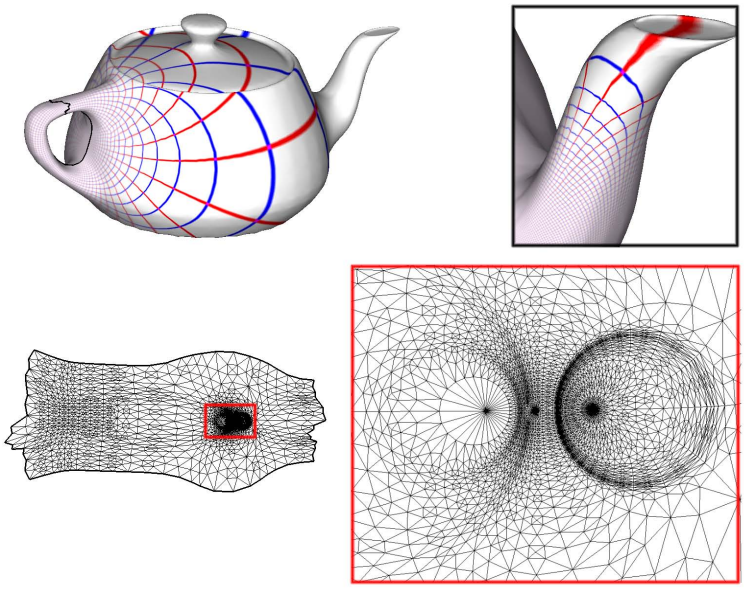


Figure 3.2: This teapot, with high Gauss curvature, is a numerical challenge for parameterization methods. As can be seen, our linearization satisfies the constraints and produces a valid parameterization, even in this difficult configuration.

angles by the angles obtained from an exponential map of the vertex one-ring. They are given in terms of the original angles original angles α_i^o as

$$\alpha_i = \begin{cases} \alpha_i^o \frac{2\pi}{\sum_{i=1}^{d} \alpha_i^0} & \text{around an interior vertex} \\ \alpha_i^o & \text{around a boundary vertex} \end{cases}$$

In practice for vertex rings with an angular deficit larger than 1 it is recommendable to switch to the angles obtained from the exponential map for that specific ring. For example, in the obtuse case illustrated in figure (3.3), the angular deficit (4.56) is very large and triggers the threshold switch.

3.3 Algorithmic outline

The algorithmic approach outlined in this chapter is simple and easy to implement. The whole algorithm for setting up the normal equation system and solving for the angles spans around 30 lines of vectorized Matlab^{TM}code. The algorithmic flow can be summarized in the following steps:

a. Establish the initial angle estimation α as explained in subsection (3.2.2)

b. Setup the constraints, i.e. vertex consistency (equations 3.2), triangle consistency (equation 3.3), and linearized wheel consistency (equation 3.7), as a linear system, $A\, e_\alpha = b$

c. Compute $C = AD_\alpha$, where $D_\alpha = \text{diag}(\alpha_i)$ is the diagonal matrix described in subsection(3.2.1)

d. Solve for x in $(CC^T)x = b$ (equation 3.13)

e. Compute the estimation error $e_\alpha = D_\alpha C^T x$ (equations 3.10 and 3.1)

f. Get the angle solution as $\alpha^* = \alpha + e_\alpha$

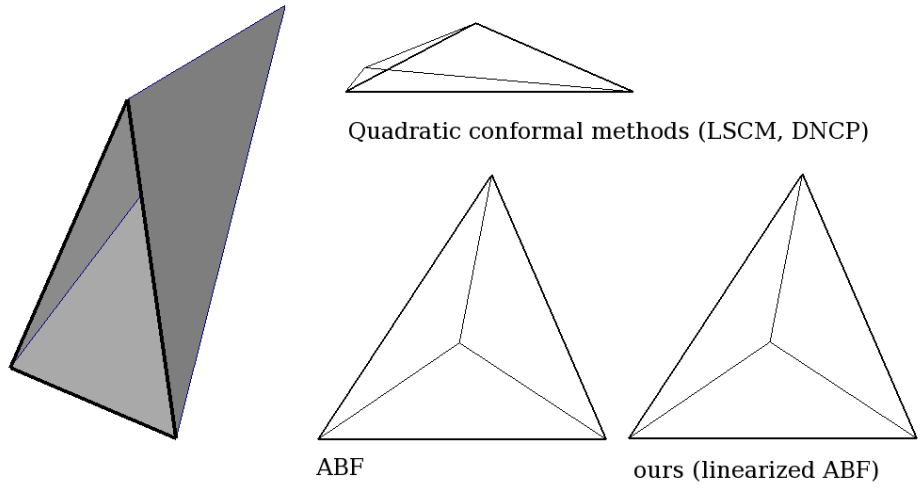
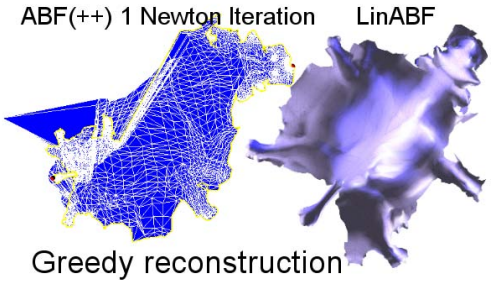


Figure 3.3: The simple example shown here is known to make linear conformal parameterization methods (LSCM [LPRM02], DNCP[DMA02]) generate an invalid parameterization. As shown here, our linearized method generates the same (valid) result as ABF.

One can use two approaches for obtaining the uv coordinates from the angles. The first one is a greedy reconstruction, which constructs the triangles one by one using a depth-first traversal. The second one is an angle based least squares formulation which solves a set of linear equations relating angles to coordinates [SLMB05].
In the greedy approach, the last triangle of a fan is never explicitly constructed as two of its edges must have already been constructed. When reconstructing with non-optimal angles, error accumulation may lead to degenerate meshes (see figure). Although our method behaves generally well with the greedy approach (see figure), we recommend using the least squares reconstruction, since it better balances the cumulative error, especially for large meshes. We used that approach for all our experiments.



3.4 Results

The current method was tested on a benchmark of nontrivial meshes. Table (3.1) shows typical values of the error in angles induced by the original ABF++ and the current linearized version for the models depicted in this chapter. The angular distortion measure we use is $\|e_\alpha\|^2 / 3nt$, where nt denotes the number of triangles. Some of these meshes were made homeomorphic to a disk by using Seamster [SH02]. To make sure that convergence comparison is accurate, we only used the algebraic transform of ABF++ (and did not use the hierarchical+algebraic HABF++). No difference is visually noticeable in the results. Timings are up to 27 $\times$ faster than the algebraic ABF++ (or up to 4 $\times$ faster than HABF++).
The numerical examples confirm the validity of the approximation used in equation (3.6). This is further illustrated in figures (3.1), (3.5), (3.6), and (3.7). Besides speed, the main advantage of

model	♯Δ	timing ABF++	timing linABF	$F/3nt$ ABF++	$F/3nt$ linABF
obtuse	3	0.1s	0.05s	1.267	1.267
cow	5.8K	0.45 s	0.1 s	5.041e-3	5.256e-3
fandisk	13 K	0.85 s	0.15 s	5.041e-3	5.256e-3
teapot	14K	3 s	0.3 s	2.154e-3	2.282e-3
foot	20K	2 s	0.4 s	1.867e-4	1.921e-4
gargo	20K	2.5 s	0.4 s	1.603e-3	1.604e-3
bull	34K	4 s	0.8 s	5.323e-4	5.331e-4
bunny1	40K	5.5	0.9 s	2.597e-4	2.593e-4
dino	48K	15 s	1 s	1.363e-3	1.184e-3
kiss	48K	8 s	1 s	7.092e-4	7.109e-4
tweety	54K	8.6s	1.5 s	1.5671e-4	1.5672e-4
bunny2	70K	13s	2 s	2.232e-4	2.243e-4
hand	73K	13 s	2 s	1.191e-4	1.213e-4
camel	78K	23 s	2.5 s	5.896e-4	6.202e-4
horse	97K	34 s	3 s	2.746e-4	2.906e-4
man	120K	36 s	2.7 s	5.293e-4	5.602e-4
head	128K	87 s	3.5 s	1.243e-4	1.240e-4
male	293K	272 s	9.5 s	3.577e-4	3.841e-4
isis	374K	250 s	11.5 s	4.834e-5	4.7941e-5
david	505K	355 s	12 s	5.776e-4	5.844e-4

Table 3.1: Timings and angular deviation.

our approach is that the setup of the problem is simplified to a great extent in comparison to previous work, that require both complex sparse matrix manipulation and a hierarchical mesh data structure. In contrast, reproduction of our results is straightforward. This way the performance of the angle based parameterization becomes comparable to the well established discrete versions of conformal maps (see [FH05]), while keeping the much better balance of deformations achieved by ABF.

We experimented the method with a large number of complicated test cases, including surfaces with high curvature (see Figure 3.2), and the example known to make linear conformal parameterization methods fail ('obtuse' entry in Table 3.1 and Figure 3.3). Although this is not guaranteed, for all these test cases, a valid parameterization was obtained. A failure case is shown in Figure 3.4. In such a (very unlikely) configuration, one can use multiple iterations. The angles computed at one iteration are constrained in $[0, \pi]$ and used to define the α's for the next iteration. In other words, we use a constrained Newton method with an active set approach. In terms of memory consumption, all the tests of our linearized method were conducted on a computer with 1Gb of system RAM, whereas ABF++ required more than 2Gb for some meshes.

Discussion

As in previous constrained optimization based methods, in this work, the constraints are satisfied up to a certain precision. In Newton-based approaches, each step improves the approximation

by linearizing the gradient of the (non-linear) Lagrangian. In our work, we perform a Taylor expansion at the level of the non-linear constraints, thus avoiding non-linear optimizations in the first place.

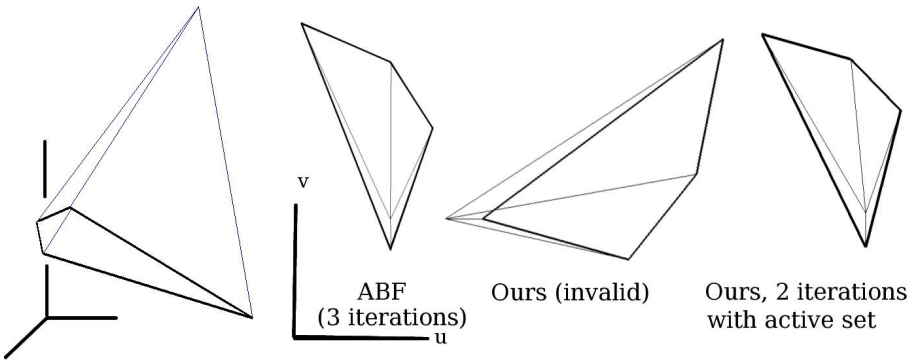


Figure 3.4: An example that makes our 1-iteration method fail. An additional iteration fixes the problem.

We tested our method on a representative benchmark of meshes. No triangle flips were detected on the results. As shown in Figure 3.4, it is possible however to engineer specific situation where a single iteration may fail. This would happen for example when a single one ring is brought to have an obtuse solid angle and sheared triangles. However, such a situation is seldom encountered in practice, and can be fixed by using our method with an active set approach.

3.5 Conclusion

We presented a complete reformulation of the angle based parameterization problem. Working directly with the approximation error instead of angles we developed a linearized version of the challenging nonlinear constraints associated with this type of parameterization. In the light of this new representation, the planar angles can be obtained as a solution to a least norm problem. The approximations used in our framework are well justified and lead to easier implementation and faster solution in comparison to previous nonlinear formulations.
Furthermore, an even faster method may be obtained by combining our linearization with the hierarchical acceleration technique used by ABF++. We plan also to investigate incorporating global non-intersection boundary constraints in our framework as well as properly handling meshes with holes.

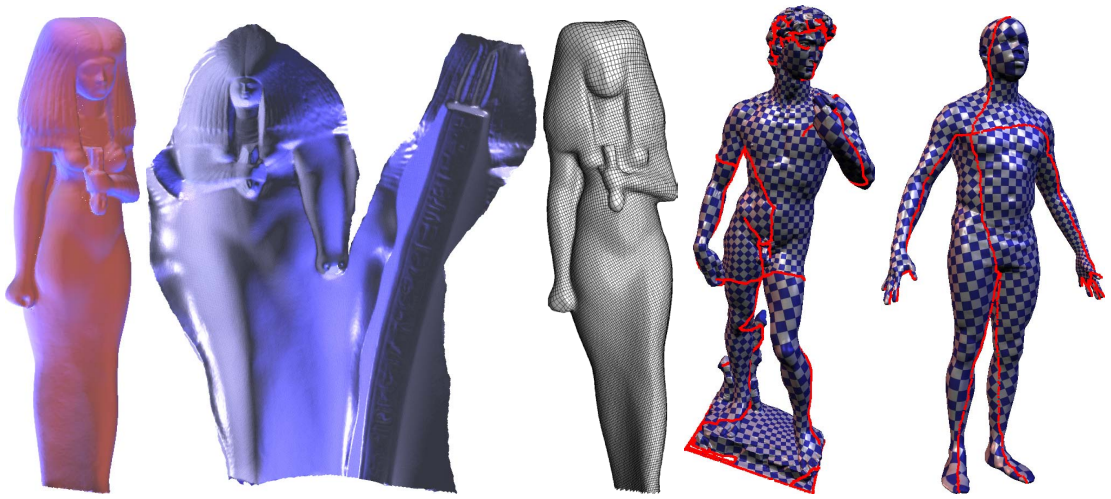


Figure 3.5: Parameterization of the Isis ($374K\Delta$), David ($505K\Delta$), and man ($120K\Delta$) models. Quadrangular remeshing and texture mapping reflect the quality of the parameterization. Respective runtimes are (11.5s), (12s), and (2.7s).

Figure 3.6: Flattening of several animal models. Model sizes and runtime are given in table(3.1).

Figure 3.7: Parameterization results of the head, tweety, foot, and hand models.

4

Discrete Conformal Parameterization

The parameterization approach discussed so far works directly on the mesh angles for generating angle preserving maps. Thus it could be considered as a purely discrete method. Alternatively, another way for constructing angle preserving maps would be to establish a characterization of these maps in the continuous setting and then derive an equivalent characterization for the discrete case. In the continuous setting, angle preserving functions are well studied in complex analysis and in differential geometry and are known as conformal maps, see e.g. [Arf85]. Thus a large amount of surface parameterization research has been dedicated to generating a discrete version of conformal maps over the last two decades. As the field has got more and more standardized, there is increasing need to relate to parallel developments in much older disciplines such as mathematics and engineering. Describing this relation is generally a difficult task and remains beyond the scope of this work. On the other hand, viewing recent advances in the light of similar ones in other fields may help us understand the historic development of certain ideas not for the sake of history itself but for building new prospects upon them and not dispersing valuable research time and effort.

The development of a discrete version of conformal maps is in essence the discretization of the functions which characterize their continuous counterpart. While the central aspects of this process are differential equations (calculus) and their associated matrix equations (linear algebra), it is worthwhile to consider some related physical problems for fixing the ideas and for gaining the physical intuition behind certain principles. In the pursuit of a solution of those continuous problems many approaches have been proposed which rely on surface discretization as an approximation tool for building solutions which in the limit should coincide with the solution of the underlying continuous problems.

A closely related problem to surface parameterization is the construction of minimal surfaces spanning a given contour. Radó, in his famous treatment of this problem back in 1930, used a discretization of the unit circle into a finite set of curvilinear triangles which are brought into conformal correspondence with a given polyhedron spanning the boundary to establish one of the first successful proofs of existence for minimal surfaces. Long before that, Schellbach in his treatment of the same problem used a discretization using linear functions on triangles which he curiously called ”triangle elements”. The striking resemblance to the setup of modern surface parameterization incites a closer look at some aspects of the minimal surface problem, especially the notion of Dirichlet energy, also called potential energy. We capitalize on the relation of this energy and the Laplace operator to introduce several possible discretization approaches for establishing a discrete form of harmonic maps. Furthermore, we discuss and evaluate the influence of these discretizations on the quality of the results.

In the rest of this chapter, we introduce the notion of Dirichlet or potential energy through the membrane equilibrium problem and then address the problem of minimal surfaces. As the

Laplace operator is of importance to both problems, we give an account of several possible discretizations of this operator, some of which have been already used in the parameterization literature. We start with the simplest discretization on the regular grid and gradually cover more general discretizations. This coverage is far from being comprehensive and more aspects of discretization history can be found in e.g. [Tho01]. Finally we compare the results of several discretization on a simple mesh parameterization benchmark.

4.1 Related topics

4.1.1 Membrane equilibrium

We consider the case of a thin elastic membrane stretched over a planar frame enclosing a region D. The potential energy of the membrane and the change of membrane area are proportionally related by the membrane tension [Cou50]. The equilibrium of the membrane can be modeled as the deformation of the membrane form its rest state. Under the assumption that the transverse membrane deflection $u(x, y)$ is small, the membrane area

$$\iint_D (1 + u_x^2 + u_y^2)^{\frac{1}{2}} \, dx \, dy \tag{4.1}$$

can be approximated by

$$\iint_D (1 + \frac{1}{2}(u_x^2 + u_y^2)) \, dx \, dy. \tag{4.2}$$

Thus the potential energy up to a constant is then given by

$$\frac{1}{2} \iint_D (u_x^2 + u_y^2) \, dx \, dy. \tag{4.3}$$

The equilibrium of the membrane can be characterized as a variational problem: the displacement $u(x, y)$ in equilibrium position is the function which minimizes the potential energy (4.3), which is equivalent to the solution of the Laplace equation with prescribed boundary conditions.

4.1.2 Minimal surfaces

The relation of minimal surfaces problem to modern parameterization was pointed out in many papers on surface parameterization. For example, the authors of [DMA02], emphasize the importance of Dirichlet energy minimization to the computation of conformal maps as well as conformal invariants (the later will not be addressed here). As the problem of minimal surfaces is a vast topic and stretches far beyond the scope of the current work, the rest of this section will address only the aspects that relate to the modern surface parameterization problem and the reader is referred to the lucid presentation of [Cou50] or the more recent [PP93].
The problem of constructing minimal surfaces has fascinated generations of mathematicians since its initial formulation by the Belgian physicist Plateau more than two centuries ago. In the original statement of the problem the aim is to find the surface $\mathcal{S}$ of least area which spans a given boundary contour Γ in $\mathbb{R}^3$. The first successful general existence proofs of minimal surfaces were first established independently by Radó [Rad30] and Douglas [Dou31]. Courant analyzed the relation of both solutions to the Dirichlet principle and gave a general overview over the underlying theory in [Cou37].

Assuming the candidate surface is represented as $X(u,\ v)$ with the parameters spanning a given domain $\mathcal{B}$ (e.g. unit disc) bounded by a curve $\mathcal{C}$ (e.g. unit circle). The problem consists of constructing a surface of least area bounded by given contour Γ in $\mathbb{R}^3$. Analytically this translates to the minimization of

$$A(X) \ = \ \iint (E\ G \ - \ F^2)^{\frac{1}{2}}\ du\ dv, \tag{4.4}$$

where

$$E \ = \ ||X_u||^2; \ \ G \ = \ ||X_v||^2; \ \ F \ = \ X_u^t\ X_v. \tag{4.5}$$

This requires the mapping X to take $\mathcal{C}$ onto Γ, to be continuous on $\mathcal{B} + \mathcal{C}$, and to have piecewise continuous second derivatives in $\mathcal{B}$.
The integral (4.4) is independent of the choice of the parameters u and v. In the case when u and v correspond to a conformal mapping of the surface on the parametric plane i.e.

$$E \ - \ G \ = \ 0; \ \ F \ = \ 0, \tag{4.6}$$

then

$$A_c(X) \ = \ \frac{1}{2} \iint (E \ + \ G)\ du\ dv. \tag{4.7}$$

A_c is in fact the Dirichlet energy and of course it attains a minimum, given prescribed boundary values. When the conformal condition on the parameters u and v breaks, then we still have

$$A(X) \ = \iint (E\ \ G)^{\frac{1}{2}}\ du\ dv \ \leq A_c(X). \tag{4.8}$$

Equality is attained if and only if condition (4.6) is satisfied. Thus a minimum of the Dirichlet energy is attained only for conformal mappings. Again the minimization of A_c is equivalent to solving the Laplace equation

$$\Delta\ X \ = \ 0. \tag{4.9}$$

4.1.3 Surface parameterization

In surface parameterization, the aim is to establish mappings to a parameter domain given prescribed boundary conditions. One way to take upon this task is to minimize the Dirichlet energy of the mapping which then yields a conformal parameterization. In the continuous setting the theory is well grounded, however, in the discrete setting of modern surface parameterization certain objections should be addressed. The transition from continuous to discrete is not always a straightforward task. In fact, conformal maps in the strict Riemann sense preserve angle measures continuously. This is clearly impossible in a discrete setting as the simple case of a non-planar 1-ring can easily reveal.
We note that many existing parameterization methods rely on the notion of weighted averages which satisfy certain qualitative properties. As these weighting schemes are generally derived in completely different ways the comparison of their quality for surface parameterization is in general not systematic. This leads to the following question, what properties should any arbitrary set of weights satisfy in order to generate good parameterizations at least visually? To address this question we recall the well known property that conformal maps preserve Laplace's equation. So instead of looking for practically non-existent conformal maps in the discrete sense, the idea is to restrict the search to mappings which preserve the Laplace equation locally. This is in fact the idea behind a whole line of weight-based parameterization methods aspiring to preserve angles. Thus the quality of any weighting scheme should be measured by how accurately

it approximates the Laplace operator on an arbitrary grid. Finally, we note that given the lack or consensus on valid definition for conformal maps in the discrete sense many authors define such maps as *discrete conformal*, *discrete harmonic*, or *shape preserving* maps to mention just a few. These methods in fact differ only in the way they construct the discrete form of the Laplace equation and specify the boundary conditions.

4.2 The Laplace equation

The Laplace equation belongs to the class of elliptic partial differential equations. The problems governed by this kind of PDEs are said to be equilibrium problems. Thus it is not surprising that numerous problems in engineering, physics and mathematics involve this equation either fully or partially. In this class of equations the value of the solution at any interior point of the domain is determined completely by the specified boundary conditions. The boundary conditions generally depend on the nature of the problem at hand. In practice, two types of boundary conditions can be singled out, namely the Dirichlet boundary conditions and the Neumann boundary conditions: Dirichlet conditions are simply specified values for the solution on the boundary; Neumann conditions are specified values for the normal derivative of the solution on the boundary. By simple physical similarity the velocity is the gradient of the solution. The normal component of velocity is the normal derivative of the solution. Thus, Neumann conditions are equivalent to specifying the normal velocity at the boundary.
For a set of special cases solutions to this equation can be obtained analytically, e.g., using separation of variables. Unfortunately, this privilege does not extend to cover the wealth of problems encountered in practical applications. In the majority of cases a solution can only be sought through numerical approximation.

4.3 Finite difference Laplacian

Finite difference is one of the oldest and simplest approaches for discretization of differential equations. On a planar regular grid, the Laplacian operator can be discretized by replacing the second partial derivatives by finite differences. The Taylor series expansion in two variables reads

$$u(x+h,y) = u(x,y) + u_x(x,y)h + u_{xx}(x,y)\frac{h^2}{2!} + \dots \tag{4.10}$$

$$u(x-h,y) = u(x,y) - u_x(x,y)h + u_{xx}(x,y)\frac{h^2}{2!} + \dots \tag{4.11}$$

This yields

$$u_{xx}(x,y) = \frac{1}{h^2}\left[u(x+h,y) + u(x-h,y) - 2u(x,y))\right]. \tag{4.12}$$

Applying the same procedure to the y variables we get

$$\nabla^2 u = \frac{1}{h^2}\left[u(x+h,y) + u(x-h,y) - 2u(x,y))\right] + \frac{1}{k^2}\left[u(x,y+k) + u(x,y-k) - 2u(x,y))\right] \tag{4.13}$$

when the grid is uniform i.e. $h = k = r$

$$\nabla^2 u \mid_i = \frac{1}{r^2}\sum_{i=1}^{4}(u_j - 4u_i), \tag{4.14}$$

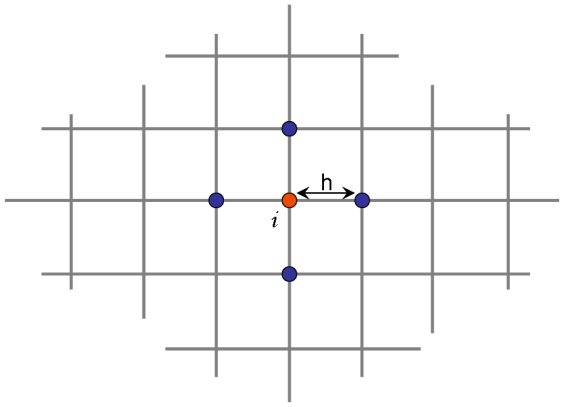


Figure 4.1: *Regular grid.*

where j spans the four neighbors of central vertex i. This expression is well in accordance with the intuitive meaning of the Laplace equation, which says that u is always equal to the average of its neighbors.

4.4 Chord length Laplacian

Bearing in mind the above mentioned averaging property of the Laplace operator it seems natural to work with local coordinates of the neighboring vertices with respect to the central vertex. The multivariate Taylor series expansion in two coordinates is given by

$$u(x,y) - u_0 = x\ u_x|_0\ +\ y\ u_y|_0\ +\ x\ y\ u_{xy}|_0\ + \frac{1}{2!}(x^2\ u_{xx}|_0\ +\ y^2\ u_{yy}|_0) + \dots \tag{4.15}$$

Setting $x = r\ \cos(\theta)$ and $y = r\ \sin(\theta)$ and integrating over the circle we get

$$\int_0^{2\pi} (u(r,\theta) - u_0)\ d\theta\ \backsimeq\ 0 + 0 + 0 + \frac{1}{2!}\ r^2 \left[u_{xx}|_0 \int_0^{2\pi} (cos^2(\theta) +\ u_{yy}|_0 \int_0^{2\pi} sin^2(\theta)) d\theta \right], \tag{4.16}$$

which gives us

$$(u_{xx}|_0\ +\ u_{yy}|_0)\ \backsimeq\ \frac{4}{r^2}\ \frac{1}{2\pi} \int_0^{2\pi} (u(r,\theta) - u_0)\ d\theta \tag{4.17}$$

$$\backsimeq\ \frac{4}{r^2}\ \overline{(u(r,\theta) - u_0)}, \tag{4.18}$$

where the upper bar denotes the mean. The above approximation is similar to the finite difference derived earlier. The only advantage is that it can be applied directly to regular or n-gonal grids, e.g. hexagonal. In this case the edge lengths and angles are all equal and the Laplacian can be written as

$$\nabla^2 u\ |_0\ \backsimeq\ \frac{4}{r^2} \frac{1}{n} \sum_{j=1}^{n} (u(r,\theta) - u_0). \tag{4.19}$$

An extension of this expression for rings with regular angles and irregular distance could be achieved by adding virtual u_i^vpoints at the intersection of the circle of radius r' with the edges [Hui91]. Simple linear interpolation over the line yields

$$u_i^v\ =\ u_0\ +\ \frac{r'}{r_i}\ (u_i\ - u_0), \tag{4.20}$$

which can be directly plugged into 4.19. The resulting approximation is then given as

$$\nabla^2 u \mid_0 \; \backsimeq \; \frac{4}{r'} \frac{1}{n} \sum_{j=1}^{n} \frac{1}{r_j} (u(r,\theta) - u_0). \tag{4.21}$$

For consistency with the finite difference approximation r' should be taken as the mean of edge lengths. While this extension reveals to be instructive, it is however still restrictive as we are looking for an approximation that would apply to unstructured meshes with irregular angles and distances.

4.5 Mean value weights

For the general case when the angles and distances are irregular, see figure (4.2), we borrow from the derivation proposed in [Hui91]. An alternative derivation was carried out in [Flo03]. Given is a point $q(r',\ \alpha)$ inside a triangle $(i, j, j+1)$. By simple projection over the adjacent edges the vector $p_i q$ can be represented as a linear combination of $p_i p_j$ and $p_i p_{j+1}$

$$p_i q = \lambda \, p_i p_j + \mu \, p_i p_{j+1}. \tag{4.22}$$

The interpolation coefficients λ and μ are given by the sine law

$$\lambda = \frac{r'}{r_j} \frac{sin(\theta_i - \alpha)}{sin(\theta_j)}, \qquad \mu = \frac{r'}{r_{j+1}} \frac{sin(\alpha)}{sin(\theta_j)}. \tag{4.23}$$

This interpolation applies also to the values of the function u on the mesh.
Going back to the Taylor expansion established earlier in equation (4.17), and assuming that the the point $q(r', \alpha)$ describes a circle of radius r' we get

$$\begin{aligned} \Delta u|_0 &= \frac{4}{r^2} \frac{1}{2\pi} \sum_{j=1}^{n} \int_0^{\theta_j} (u(r',\alpha) - u_0) \, d\alpha \\ &= \frac{4}{r'^2} \frac{1}{2\pi} \sum_{j=1}^{n} \int_0^{\theta_j} \left[\frac{r'}{r_j} \frac{sin(\theta_j - \alpha)}{sin(\theta_j)} (u_j - u_i) + \frac{r'}{r_{j+1}} \frac{sin(\alpha)}{sin(\theta_j)} (u_{j+1} - u_i) \right] d\alpha. \end{aligned} \tag{4.24}$$

A simple integration and rearrangement of terms yield the mean value coordinates.

$$\Delta u|_0 \; = \frac{4}{r'} \frac{1}{2\pi} \sum_j \left[\frac{1 - \cos\theta_j}{\sin\theta_j} + \frac{1 - \cos\theta_{j+1}}{\sin\theta_{j+1}} \right] \frac{(u_{j+1} - u_i)}{r_{j+1}} \; = \; 0 \, . \tag{4.25}$$

The exact value for r' can be obtained by requiring consistency with the uniform grid case. When the Laplacian is set equal to zero we get the well known mean value coordinates.

4.6 Generalized finite differences

One of the earliest and successful derivations of the generalized finite differences on unstructured grids goes back to the late 1940's. MacNeal [Mac49, Mac53] proposed the use of a dual

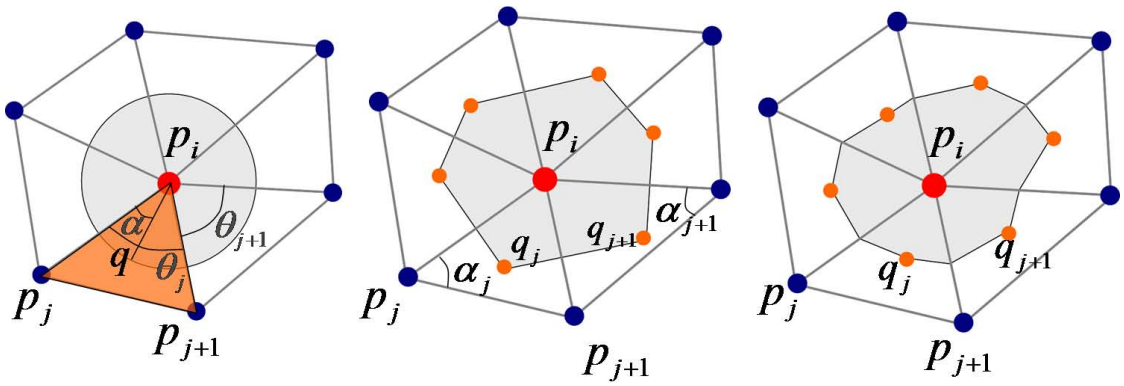


Figure 4.2: *Integration domains for the mean value, median dual and barycenteric dual.*

mesh for the derivation of generalized finite differences by using analogy to electrical networks. The approach proceeds by applying Gauss's theorem to the weak form of the Laplacian, see section (4.7)

$$\frac{1}{2}\int |\nabla u|^2 ds = \int \nabla u \cdot n dr. \tag{4.26}$$

The discretization amounts then to approximating the boundary integral above.
Taking the integration domain as the one defined by the bisector dual of the polygon see figure (4.2), the integration is then done over the boundary segments and for a single segment we have

$$\int_{q_j}^{q_{j+1}} \nabla u \cdot n dr = |q_j q_{j+1}| \, \nabla u \cdot n \tag{4.27}$$

$$= |q_j q_{j+1}| \frac{u_{j+1} - u_i}{|p_{j+1} p_i|} \tag{4.28}$$

By simple geometric argument it is possible to show that

$$\frac{|q_j q_{j+1}|}{|p_{j+1} p_i|} = \cot \alpha_j + \cot \alpha_{j+1} \tag{4.29}$$

using a different integration domain, namely the barycenteric dual, see figure (4.2), Winslow [Win66] derived a similar expression. While both, MacNeal and Winslow were aware that the approximation fails when the mesh does not satisfy the Delaunay condition, it was only Winslow who proposed a mesh generator that takes care of the problem. An extensive analysis of both schemes which falls in the category of what is called box integration methods can be found in [Hei87]. Interestingly enough, the expression (4.29) can also be obtained through a finite element derivation.

4.7 Finite element linear triangle

It is hard to clearly date the origins of the finite element method. According to [Bab94, Bra01, Ode87], the earliest appearances of the world "triangular element" is in the work of Schellbach [Sch52] who used piecewise linear functions on a triangular mesh for the minimization of the deformation energy for solving a special case of the plateau problem. In mathematics, Courant's paper [Cou43] is generally considered to be at the origin of the method while the term

"finite element" itself was first coined in engineering by Clough in 1960 and most of the early successful codes were developed by engineers.
The principle of minimum potential energy requires that the potential distribution on a surface must be such as to minimize the stored filed energy per unit surface. The energy can be expressed as

$$E(u) = \frac{1}{2} \int |\nabla u|^2 ds. \tag{4.30}$$

The minimum energy principle is mathematically equivalent to Laplace's equation in the sense that a potential energy which satisfies the Laplace equation will also minimize the energy, and vice-versa. Therefore an alternative approach to direct discretization of the Laplace operator as performed above could be replaced by a suitable approximation of the stored energy $E(u)$ by assuming the potential u to be given by a simple combination of suitably chosen simple functions with as yet underdetermined coefficients. The minimization of the energy then determines the coefficients, and thereby implicitly determines an approximation to the potential energy. Virtually all finite element methods follow this approach or adaptations of it.
Let us suppose that $u(x, y)$ minimizes the true solution to the Dirichlet problem for the Laplace equation and consider some sufficiently differentiable function $h(x, y)$ taking zero values at boundary points where the value of u is prescribed. This way for a scalar ϵ the prescribed values of $(u + \epsilon h)$ are the same as the prescribed values of u. The energy associated with $E(u + \epsilon h)$ is

$$E(u + \epsilon h) = E(u) + \epsilon \int \nabla u \cdot \nabla h ds + \frac{1}{2} \int |\nabla h|^2 ds. \tag{4.31}$$

Green's theorem reveals that

$$\int \nabla u \cdot \nabla h \, ds = \oint_c h \, \nabla u \cdot n \, ds - \int h \, \nabla^2 u \, ds. \tag{4.32}$$

The last term on the right hand side vanishes because u satisfies the Laplace equation. The first term also vanishes since at each boundary point, either h or the normal derivative of u vanish. We have then

$$E(u + \epsilon h) = E(u) + \epsilon^2 \int E(h). \tag{4.33}$$

This equality reveals two important facts. $E(u)$ is the minimum energy as the last term in the right hand side is always positive. This minimum is reached for $\epsilon = 0$ for any admissible function h. Admissibility here is understood as the fulfilment of two properties. First, h must vanish at the boundary points where u is prescribed and second, h must be at least once differentiable.
Inspection of the last equation reveals that the energy error depends quadratically on ϵ. In other words, for a small error $epsilon$ in the estimation of u, the error in the estimation of the energy is even smaller. This point is of utter importance to most applications where the energy computation is much more important than the potential.
The essence of the finite element discretization lies first in approximating the potential u within each triangle in a standard fashion and then interrelating the potential distributions in the various elements so as to constrain the potential to be continuous across the triangle boundaries.
Within a typical triangular element, it will be assumed that the potential can be adequately approximated by the expression

$$u \; = \; a \; + \; bx \; + \; cy \; . \tag{4.34}$$

The true solution is thus replaced by a piecewise linear function. From this setup it becomes clear that the potential along any triangle edge is the linear interpolate between its two vertex

values, so that if two triangles are adjacent, the potential will be continuous along the shared edge. The approximate solution is piecewise linear but continuous everywhere.
The coefficients a, b and c can be readily obtained by solving the linear system

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}. \tag{4.35}$$

Substitution of the solution in equation (4.34) yields

$$u \;=\; \begin{pmatrix} 1 & x & y \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 & x & y \end{pmatrix} \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}^{-1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}. \tag{4.36}$$

Expanding and rearranging the above expression we can rewrite

$$u \;=\; \sum_{1}^{3} \lambda_i \, u_i, \tag{4.37}$$

where

$$\lambda_i = \frac{1}{2A} \left[(x_j y_k - x_k y_j + (y_j - y_k)x + (x_k - x_j)y) \right]. \tag{4.38}$$

In this equation, A represents the area of the triangle and λ is clearly linear in position. An interesting property of these bilinear functions is that they vanish at all vertices but one where they have unit value.
The potential gradient within an element triangle can be found from equation (4.34)

$$\nabla u \;=\; \sum_{i=1}^{3} u_i \, \nabla \lambda_i. \tag{4.39}$$

Now we have all the ingredients necessary to determine the energy contribution associated with a single triangular element. The integration area here is the triangle itself.

$$E(u) = \frac{1}{2} \int |\nabla u|^2 \, ds \;=\; \frac{1}{2} U^t \, S \, U \tag{4.40}$$

where u is given by $U \;=\; [u1 \; u2 \; u3 \,]^t$, and the 3x3 matrix S is given in terms of the triangle angles $\theta_1,\ \theta_2$, and θ_3 as

$$S = \frac{1}{2} \begin{pmatrix} \cot\theta_3 \;+\; \cot\theta_2 & -\cot\theta_3 & -\cot\theta_2 \\ -\cot\theta_3 & \cot\theta_3 \;+\; \cot\theta_1 & -\cot\theta_1 \\ -\cot\theta_2 & -\cot\theta_1 & \cot\theta_2 \;+\; \cot\theta_1 \end{pmatrix}. \tag{4.41}$$

4.8 Linearity-preserving weights

The linearity-preserving weighting scheme was first proposed by Holmes and Connell in [HC89]. This approach takes advantage of the local averaging property of the Laplacian to generate the weighting scheme. The idea consists of perturbing the weights from unit weights

and minimize the sum of squares of the perturbation subject to the null Laplacian condition for linear data. Interestingly enough the constrained system can be solved explicitly. Assuming the Laplacian can be written as

$$\nabla^2 u = \sum_{i=1}^{n} w_i (u_i - u_0) \tag{4.42}$$

and the perturbed weights as

$$w_i \; = \; 1 \; + \; \epsilon_i. \tag{4.43}$$

Which reflects that weights should be as close to unity as possible similarly to the finite difference case where the weights are exactly one as shown in section (4.3).
The aim is then to minimize the cost function

$$\sum_{i}^{n} \epsilon_i^2 \tag{4.44}$$

under the linearity constraints conditions

$$\sum_{i=1}^{n} w_i (x_i - x_0) \;\; = \;\; 0, \tag{4.45}$$

$$\sum_{i=1}^{n} w_i (y_i - y_0) \;\; = \;\; 0. \tag{4.46}$$

Using Lagrange multipliers, the weights can be expressed as

$$w_i \; = \; 1 \; + \; \frac{I_{xy}\, R_y \; - \; I_{yy}\, R_x}{I_{xx}\, I_{yy} - I_{xy}^2} \, (x_i \; - \; x_0) \; + \; \frac{I_{xy}\, R_x \; - \; I_{xx}\, R_y}{I_{xx}\, I_{yy} \; - \; I_{xy}^2} \, (y_i \; - \; y_0), \tag{4.47}$$

where the moments are given by

$$I_{xx} \;\; = \;\; \sum_{i} \, (x_i - x_0)^2 \tag{4.48}$$

$$I_{yy} \;\; = \;\; \sum_{i} \, (y_i - y_0)^2 \tag{4.49}$$

$$I_{xy} \;\; = \;\; \sum_{i} \, (x_i - x_0) \, (y_i - y_0) \tag{4.50}$$

$$R_x \;\; = \;\; \sum_{i} \, (x_i - x_0) \tag{4.51}$$

$$R_y \;\; = \;\; \sum_{i} \, (y_i - y_0). \tag{4.52}$$

4.9 Least squares weights

In general, the finite difference Laplacian of a function u on an unstructured grid at a certain node P_0 of valence m can be expressed as the linear combination of a weighted sum of the values of the function over the 1-ring of P, i.e.

$$\nabla^2 u = \sum_{i=1}^{m} w_i (u_i - u_0). \tag{4.53}$$

We assume here that the vertex P is the center of coordinates $(0,0)$ for simplicity of the notation. For numerical stability the weights are required to be strictly positive although convergence can be achieved without guarantees for negative weights, as positivity is only a sufficient but not necessary condition.
Using the two-dimensional Taylor expansion of our function at each vertex in the 1-ring around P we have

$$u_i - u_0 = x_i\partial_x u + y_i\partial_y u + x_i y_i \partial_{xy} u + \frac{1}{2}x_i^2\partial_x^2 u + \frac{1}{2}y_i^2\partial_y^2 u + ... \tag{4.54}$$

A straightforward substitution of (4.54) in (4.53) yields the following expression

$$\nabla^2 u \;=\; \sum_{i=1}^{m} w_i \left[x_i\partial_x u + y_i\partial_y u\right] \;+\; \sum_{i=1}^{m} w_i \left[x_i y_i \partial_{xy} u + \frac{1}{2}x_i^2\partial_x^2 u \;+\; \frac{1}{2}y_i^2\partial_y^2 u\right] + ... \tag{4.55}$$

From the above equation, it comes clear that to have an accurate approximation of the Laplacian operator, the following set conditions should hold

$$\sum_{i=1}^{n} w_i x_i = 0, \tag{4.56}$$

$$\sum_{i=1}^{n} w_i y_i = 0, \tag{4.57}$$

$$\sum_{i=1}^{n} w_i x_i^2 - 2 = 0, \tag{4.58}$$

$$\sum_{i=1}^{n} w_i y_i^2 - 2 = 0, \tag{4.59}$$

$$\sum_{i=1}^{n} w_i x_i y_i = 0. \tag{4.60}$$

It is easy to check that these conditions yield the well known uniform Laplacian on regular grids and also the edge weighted Laplacian on grids with regular angles and irregular distances. It should be also noted that for valences larger than five the least squares solution is unique although not necessarily positive. When valence is less then five a least norm solution is used.
While this approximation is mathematically sound, it may yields negative weights in certain instances. In order to guarantee the positivity of the weights it seems natural to treat the problem as an optimization problem. If we seek a zero degree accuracy Laplacian it is sufficient to satisfy conditions (4.56) and (4.57), however, it is also desirable to have a consistency of these solutions over the whole mesh. This can be achieved by considering the weights as extrema of a certain penalty function $f(w)$ such that

$$\begin{aligned} w_i &> 0. \\ \sum_{i=1}^{m} w_i x_i &= 0. \\ \sum_{i=1}^{m} w_i y_i &= 0. \end{aligned} \tag{4.61}$$

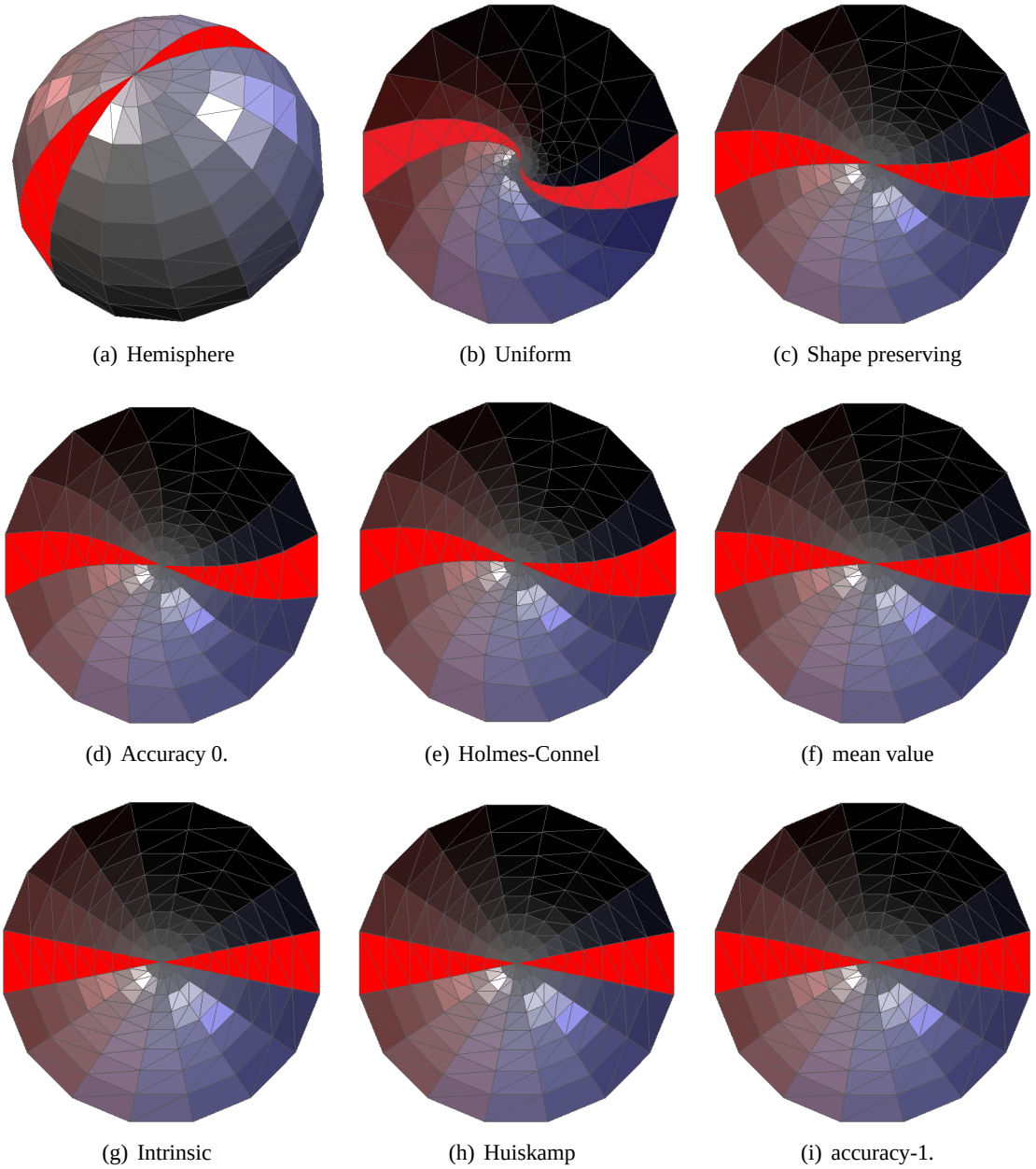
One possible choice for $f(w)$ would be the sum of logarithms of weights, in this way the Laplacian operator would be similar to the one established by Shontz et al. [SV03]. The results achieved using this 0-accuracy Laplacian in surface parameterization are surprisingly similar to the shape preserving weights [Flo97] and to the linearity preserving weights by Holmes and Connell [HC89]
In order to achieve a 1-degree accuracy approximation of the Laplacian the conditions (4.58), (4.59), and (4.60) should be incorporated. A possible approach for forcing the positivity of the weights consists of solving the following optimization problem

$$\begin{array}{ll} \text{minimize} & (\sum_{i=1}^{m} w_i x_i^2 - 2)^2 + (\sum_{i=1}^{m} w_i y_i^2 - 2)^2 + (\sum_{i=1}^{m} w_i x_i y_i)^2. \\ \text{such that} & w_i > 0. \\ & \sum_{i=1}^{m} w_i x_i = 0. \\ & \sum_{i=1}^{m} w_i y_i = 0. \end{array} \tag{4.62}$$

The local optimization problem can be solved using the Lagrange multipliers approach, see [Ber82].
In order to generates higher accuracies it is clear that similar equations can be derived by using higher order Taylor expansion, however, the satisfaction of all constraints becomes highly difficult and costly.

4.10 A simple benchmark

In this very restrictive overview of discretization schemes we are far from covering the wealth of the literature on this broad topic, we briefly mention some of the additional weighting schemes that have been used in the mesh parameterization literature. The shape preserving weights were proposed by Floater in [Flo97], The rational basis function were introduced by Wachspress in [Wac71], and the natural neighbor [Sib81] by Sibson.
In order to evaluate the performance of the different approximation, a simple benchmark consisting of a half sphere, see figure (4.3), is used and simple visual inspection reveals how great circle arcs (geodesics) on the sphere are transformed onto the plane. A comparison of some of the mentioned Laplacian approximations can also be found in [Hui91].
From the benchmark it seems that the linear triangle and the least squares based weights perform better as their approximation of the Laplacian operator is more accurate. Furthermore the simplicity of and ease of implementation of the linear triangle makes very attractive for surface parameterization. In addition the mean value weights merit attention in that, they remain always positive and are comparable in their simplicity to the linear triangle.



(a) Hemisphere (b) Uniform (c) Shape preserving

(d) Accuracy 0. (e) Holmes-Connel (f) mean value

(g) Intrinsic (h) Huiskamp (i) accuracy-1.

Figure 4.3: *Half a sphere parameterized using different methods.*

5

Quasi-Harmonic Maps

The mappings addressed thus far do not preserve measures other than angles. As a typical observation, the area distortion tends to grow as one moves away from the boundary – the more complicated the geometry of the mesh, the more significant the distortion. Figure (5.3) illustrates this situation. Given the scarcity of isometric maps which preserve angles and areas at the same time, a good direction of investigation is to establish smooth blends of both kinds – authalic (area preserving) and conformal parameterizations.
Our approach to the problem stems from the following question: Given a map g from the plane to a surface mesh, can we find a mapping f from the plane onto itself that has the same properties as the initial map (see figure (5.2)). Finding such a map would yield an optimal parameterization that mimics the original mapping, but this time from the plane into itself. In the following we define the required properties of such maps, and we propose a method for establishing them. In brief, the basic approach can be overviewed as follows:

1. Provide an initial map g, e.g., a discrete conformal parameterization. (Any non-degenerate setting will work, e.g., a reasonable projection.)

2. For every triangle estimate the Jacobian matrix C of the map f from the first fundamental form of g. (Section (5.2))

3. Building upon our discretization of the quasi-harmonic equation, compute weights w_{ij} and set up a linear system. (Sections (5.3) and 5.4)

4. Solve the arising linear system. The same boundary condition as in step (1) are used.

5. If required, iterate restarting from step 2.
   (Note that our method stabilizes after few iterations and would not degenerate.)

5.1 Motivation

The inspiring idea behind the approach outlined in this chapter stems from simple physical considerations. Let us consider the steady state heat equation on a quadrangular plate. In a first stage we treat the surface as a homogenous domain in the sense that the heat conductance is constant over the whole surface. The heat distribution can be obtained by solving a laplace equation with prescribed conditions. In this example, we apply Dirichlet conditions to the top and bottom sides and Neumann boundary conditions to the right and left sides as illustrated in figure (5.1-left).
On the other hand if we impose specific conductance values for the circular and rectangular subdomains inside the plate, the standard laplace equations is not suitable anymore for modeling the

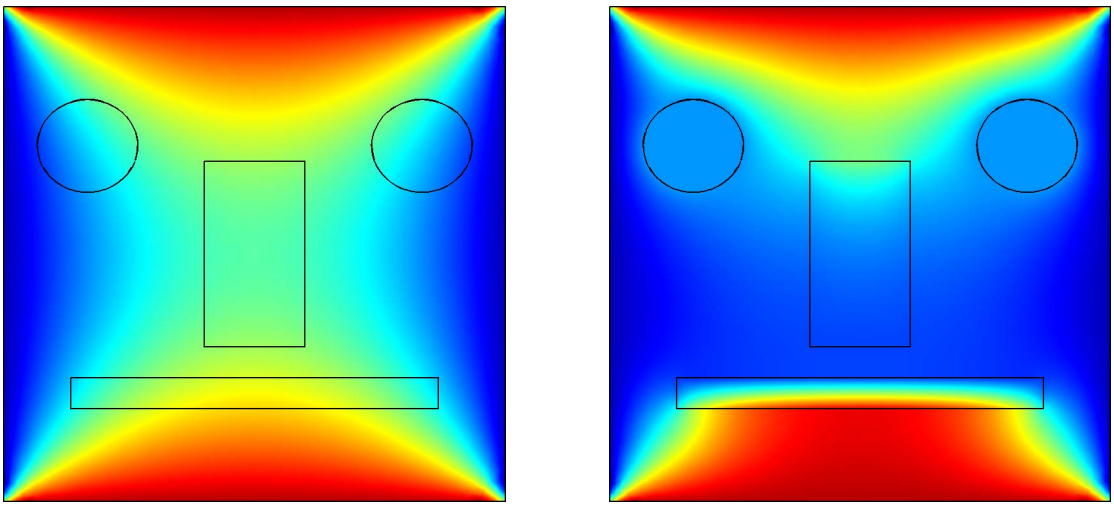


Figure 5.1: *Heat distribution on a homogenous (left) and inhomogeneous (right) plate, modeled using the Laplacian operator and the quasi-harmonic operator respectively.*

heat distribution and we have to rely on the so called quasi-harmonic equation which incorporates the conductance terms and is therefore sensitive to the inhomogeneous nature of the plate. The heat distribution in this setup is depicted in figure (5.1-right).
This example illustrates how simple scalar conductance values can alter the heat distribution on a simple domain. We capitalize on this observation for controlling the distortion in surface parameterization not only using scalar distortion values but we present the more general setup where the distortion is quantified in tensorial form.

5.2 Quasi-harmonic maps

In order to motivate the ensuing discussion and fix the ideas let us consider the following problem: Given a single triangle of a surface mesh and a map g that carries the triangle into a corresponding triangle in the planar domain. Solely from the properties of the map g, we can establish a second map f that reproduces exactly the initial triangle on the plane from the second triangle. Formally, the problem can be coined as follows: Recalling basic theory of differential geometry (see e.g. [Kre91]), the deformation of the initial mapping is given by

$$dX \;=\; JdU\;, \tag{5.1}$$

where the Jacobian J is a 3x2 tensor. Taking the norm we get

$$\|dX\|^2 \;=\; dU^\top J^\top JdU\;. \tag{5.2}$$

In this expression, we already recognize the first fundamental form $\mathbf{I} = J^\top J$. In general, a mapping is locally characterized by the first fundamental form, which tells how distances (and hence angles and area) measured in the parametric domain are transformed to distances on the surface. Hence, all information on the distortion is captured by this tensor. An ideal mapping in the plane, which exactly mimics the behavior of the initial mapping, would induce the same distortion of lengths as this initial mapping, i.e.

$$\|dx\|^2 \;=\; dU^\top \mathbf{I} dU\;, \tag{5.3}$$

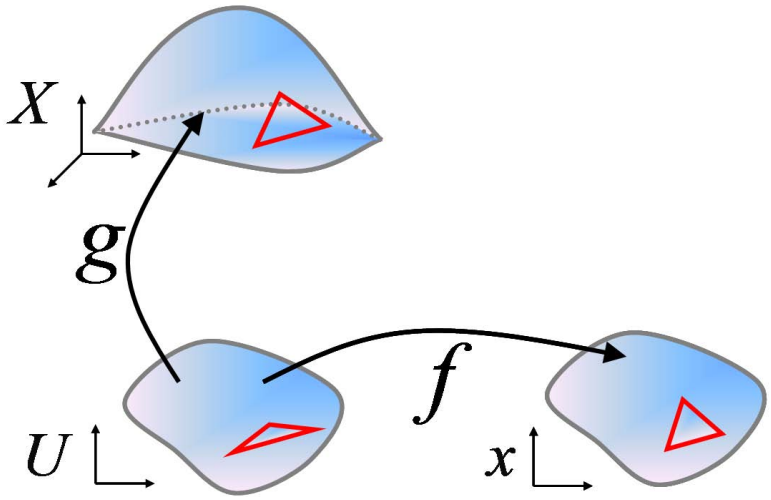


Figure 5.2: *g maps the planar domain to the surface, and f is a mapping of the plane onto itself.*

where x denotes a displacement in the plane. Now, we would like to derive an expression similar to (5.1) – this time in the plane – i.e.

$$dx \;=\; K dU \;. \tag{5.4}$$

Here, the Jacobian K of the new mapping is a 2×2 matrix which should satisfy

$$\mathbf{I} = K^{\top} K \;. \tag{5.5}$$

Since the first fundamental form is symmetric positive definite, the matrix K can be found as the square root of $\mathbf{I}$. For a concise formulation of the square root of a 2×2 matrix we refer to [Lev80]. This derivation of the Jacobian K is original compared to existing parameterization methods where emphasis is solely on the eigenvalues and the related norms.
So far we illustrated the basic idea and considered only a single triangle. In the case of a surface mesh, the question arises, whether we can compute the vertex positions of the new mesh from the Jacobian matrices of the transformations defined over each triangle. It turns out that the problem is generally overdetermined and may not have a solution at all. This does not come as a surprise due to the fact that isometric mappings exists only for the special case of developable surfaces. Excluding a direct approach to the problem, we propose an alternative to account for the Jacobian matrix. Recalling that harmonic maps minimize the Dirichlet energy

$$\int_{\Omega} \|\nabla g\|^2 \;, \tag{5.6}$$

our approach attempts to recover the geometry of the original mesh by minimizing the following energy functional

$$\int_{\Omega} (C \nabla f) \cdot (\nabla f) \;, \tag{5.7}$$

where $C = K^{-1}$, and can be computed explicitly as K is only 2x2. The partial differential equation associated with this energy is the following quasi-harmonic equation

$$\mathbf{div}(C \mathbf{grad} f) \;=\; 0 \;. \tag{5.8}$$

This equation and its many variants are typically used for modeling many steady state problems in mechanics and electromagnetism (see e.g. [Zie71]). Say, the distribution of vertices is given by f, the flux of the distribution f is given by C **grad**f, and due to conservation law this type of problem conforms to solving the two-dimensional elliptic steady state equation (7.1). Consequently this guarantees that our solution will show the same qualitative behavior as general elliptic problems. In particular the solution will not show jumps but will vary smoothly.
Under mild conditions the solution to the problem exists and is unique, see [Glo84], especially their Appendix I for a detailed analysis of the convergence properties. In our case the symmetric nature of the tensor C and given that it does not stem from a degenerate initial parameterization(triangle collapsing to one point) guarantees that a solution to the quasi-harmonic equation exists and is unique.
Moreover, we note that our setup is a special case of the Leray-Lions equation [IS01]. A geometric interpretation of such maps can be found in [AN02].

5.3 Discretization

In the following we solve the variational problem associated with the quasi-harmonic equation (7.1). We refer to vertices and triangles as explained in figure (5.4.a).
In dealing with equations of this type, it is much more advantageous to tackle the integral form rather then the partial differential equation itself. This is not fortuitous as the Gauss divergence theorem yields a much easier expression to handle

$$\int_A \mathbf{div}(C\mathbf{grad}f)dA = \oint_{\partial A} (C\mathbf{grad}f)\cdot n \; dl \; = \; 0 \; . \tag{5.9}$$

Hence, the integral form of the quasi-harmonic equation obtained by integrating over some non-overlapping regions reduces to a simple integration of the gradient over their boundaries. We define the non-overlapping areas for the integration as the dual mesh defined by the centroids of

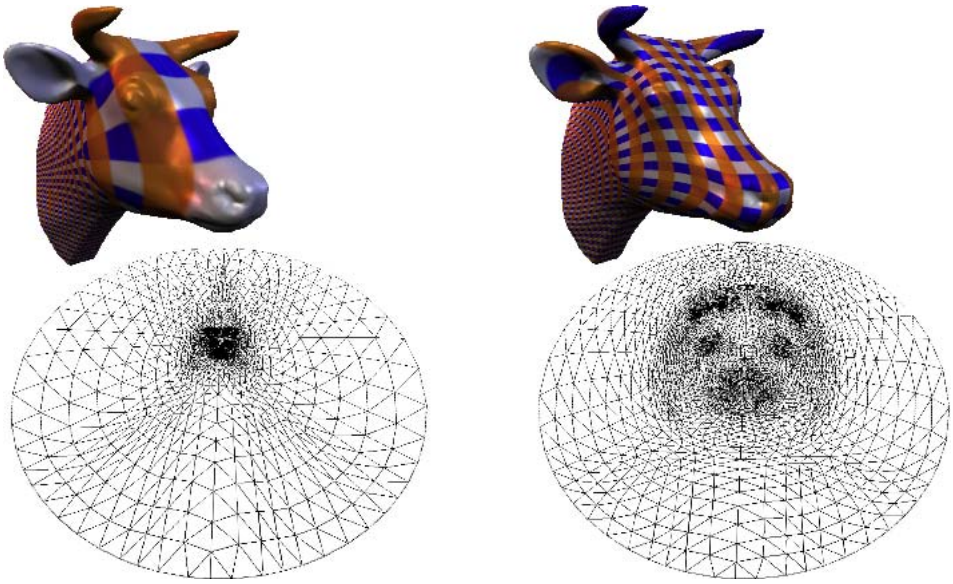
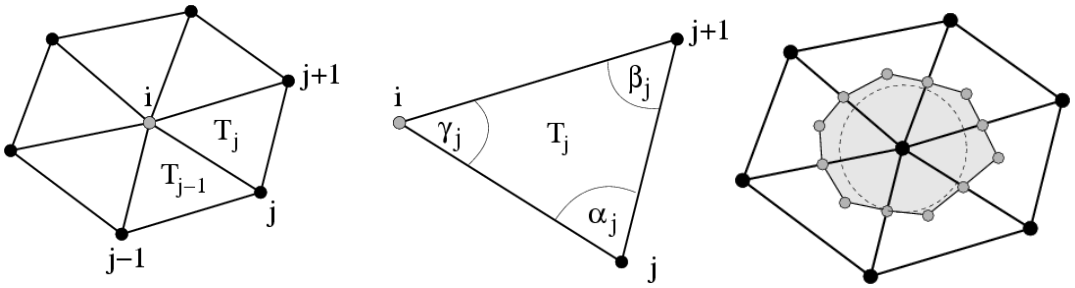


Figure 5.3: *Texturing a cow head model ($6K\Delta$) using a discrete conformal map ([LPRM02, DMA02]; left) and a quasi-harmonic map (right).*

the triangles and the midpoints of the edges over every 1-ring neighborhood (see figure (5.4.b)). We remark that this dual has exactly the same area as the mesh.



(a) (b)

Figure 5.4: *(a) Indexing of direct neighbors j and associated triangles in the 1-ring of the center vertex i, angles are indexed per triangle j. As all derivations are local to the 1-ring, we avoid double indices, so implicitly $T_j = T_{i,j,j+1}$. (b) The integration area for the discretization of (5.9) is given by the dual mesh. For the mean value coordinates the integration area is a circle.*

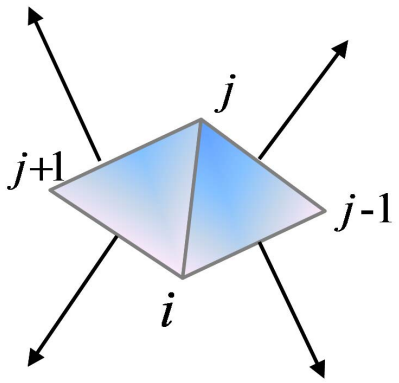


Figure 5.5: *The two triangles adjacent to edge (i, j). The outward vectors $x^{\perp}$ correspond to the respective edges rotated by $\frac{pi}{2}$.*

The gradient on a triangle $\{i, j, j+1\}$ associated with the edge $\{i, j+1\}$ is given by

$$\mathbf{grad}_T(f_{j,j+1}) \;=\; \frac{\mathbf{x}^{\perp}_{i,j+1}}{2A_T}\,,$$

where $x^{\perp}_{i,j}$ denotes the edge vector rotated by $\frac{\pi}{2}$ (see figure (5.5)).
Equation (5.9) involves only the normal to the boundary of the ring, which is just the normalized boundary edge rotated by $\frac{\pi}{2}$. So given an edge and its two adjacent triangles $T_1 = \{i, j-1, j\}$ and $T_2 = \{i, j, j+1\}$, the weight associated the edge can be written in the following form

$$w_{ij} \;=\; \frac{\mathbf{x}^{\perp}_{j-1,i} C_{j-1} \cdot \mathbf{x}^{\perp}_{j,j-1}}{4A_{j-1}} \;+\; \frac{\mathbf{x}^{\perp}_{j+1,j} C_j \cdot \mathbf{x}^{\perp}_{i,j+1}}{4A_j}\,. \tag{5.10}$$

In the discrete setting, the tensor field C is defined piecewise per triangle. When the pieces C_{j-1} and C_j are both equal to the identity, this expression simplifies to the well-known cotangent weights, which are widely used for the computation of the discrete Laplacian operator. We note that in this special case the initial mapping is isometric, and hence the solution will be identical to the initial map.

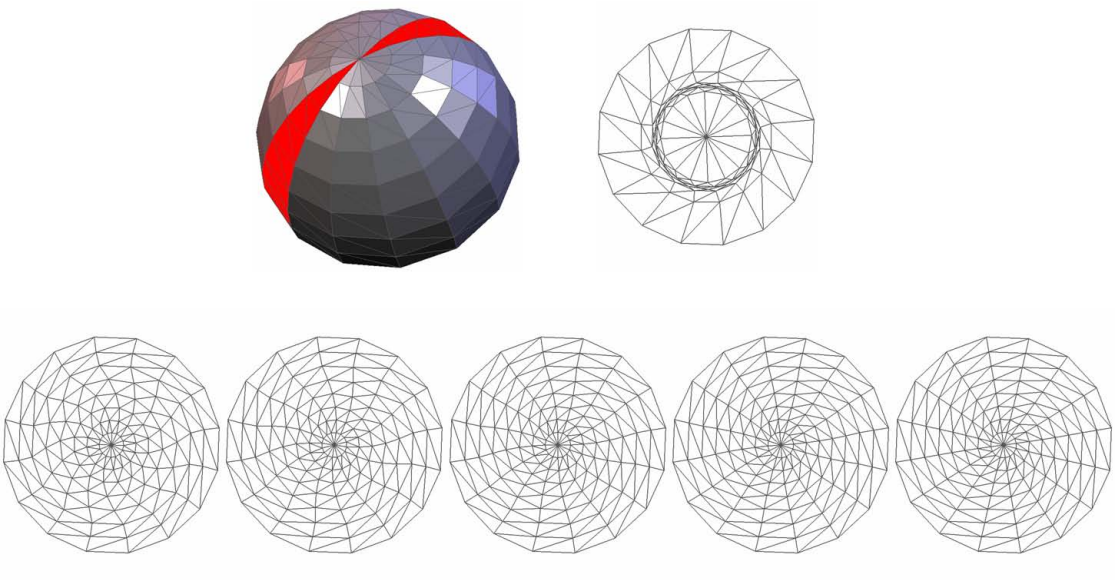


Figure 5.6: *Self-correction and convergence for a regular half-sphere model. An ill-shaped initial map (top right) is chosen. The series on the bottom show the quasi-harmonic maps obtained from the 1st to 5th iteration.*

5.4 Mean value coordinates extension

A sufficient condition for guaranteeing the validity of the approach is the positivity of the weights in equation (5.10), formally expressed as the discrete maximum principle. For solving the Laplace equation based on the cotangent weights, the Delaunay criterion should hold for all internal vertices while all angles facing boundary edges should be acute. Given appropriate boundary conditions this condition is sufficient to guarantee the construction of a valid mapping. However, it is not necessary in general and in many instances the arising system may have a valid solution even in the presence of negative weights. For the quasi-harmonic setting described above, the Delaunay condition is no longer sufficient. In fact, the validity is guaranteed only if *all* angles are acute.

One way to satisfy this condition (or at least the Delaunay criterion) is, for instance, to apply Rivara's edge bisection algorithm [Riv84], which is guaranteed to keep angles away from 0 and π.

Another way is to use alternative weights. A promising class of weights are the *mean value coordinates* [Flo03, Hui91], as they guarantee positivity regardless of the interior angles of the triangle mesh.

In the following, we will show how to derive a new class of discrete quasi-harmonic maps that are based on the mean value coordinates. Here, we note that for a general tensor C, rather complicated trigonometric expressions have to be integrated over circular regions resulting in a comparatively complicated formula for the weights. To fix the ideas with minimal formalism, we restrict ourselves at a first stage to the case where C is a scaled unit tensor, formally $C = \kappa I$. (In fact, this is the special case of a scalar version of the approach, for example, κ can be taken as the per triangle ratio of the initial map and the original mesh.) Note that all our numerical examples are computed with general tensors, for which we state at then of this section for the general formula.

We proceed similarly to [Hui91, Flo03], however, we discretize the quasi-harmonic equation again starting from the divergence theorem. For this purpose, we first compute the gradient: Given a triangle $T_j = (i, j, j+1)$ with vertex positions $\mathbf{u}_i$, $\mathbf{u}_j$, $\mathbf{u}_{j+1}$ and a function f defined over this triangle, the following relations hold for a point $\mathbf{u} = (u, v)$ in the triangle. Let γ_j denote the angle at u_i. In local polar coordinates

$$\mathbf{u} - \mathbf{u}_i = r\,(\cos\gamma, \sin\gamma)^\top\,, \tag{5.11}$$

and for f

$$f(\mathbf{u}) - f(\mathbf{u}_i) \;=\; \mathbf{grad} f \cdot (\mathbf{u} - \mathbf{u}_i)\,,$$

where the gradient is given by

$$\mathbf{grad} f \;=\; H_T(\mathbf{u} - \mathbf{u}_i)\,.$$

Here, the value of H_T can be found by applying a simple linear interpolation over the triangle T, we get

$$\begin{aligned} H_T \;= & \frac{\sin(\gamma_j - \gamma)}{r\, r_j\, \sin\gamma_j}(f(\mathbf{u}_j) - f(\mathbf{u}_i)) \;+ \\ & \frac{\sin(\gamma)}{r\, r_{j+1}\, \sin\gamma_j}(f(\mathbf{u}_{j+1}) - f(\mathbf{u}_i))\,. \end{aligned}$$

Where r_j and r_{j+1} denote the length of $u_i - u_j$ and $u_i - u_{j+1}$ respectively. Now substituting this expression in (5.9), and keeping in mind that the gradient and the outward normal to the circle are collinear, we obtain the following expression

$$\int_0^{2\pi} \kappa\, \mathbf{grad} f \cdot n d\gamma \;=\; \sum_j \int_0^{\gamma_j} \kappa_j H_{T_j}\, r\, d\gamma\,, \tag{5.12}$$

and expanding the right hand side we obtain

$$\begin{aligned} \sum \kappa_j \int_0^{\gamma_j} & \frac{\sin(\gamma_j - \gamma)}{r_j\, \sin\gamma_j}(f(\mathbf{u}_j) - f(\mathbf{u}_i)) \;+ \\ & \frac{\sin(\gamma)}{r_j\, \sin\gamma_j}(f(\mathbf{u}_{j+1}) - f(\mathbf{u}_i))\,. \end{aligned}$$

(Notice that the r term was simplified.) After integration and reassembly of the term, we get the following

$$\sum_j \left[\frac{1 - \cos\gamma_j}{\sin\gamma_j}\kappa_j + \frac{1 - \cos\gamma_{j+1}}{\sin\gamma_{j+1}}\kappa_{j+1}\right] \frac{(f(\mathbf{u}_j) - f(\mathbf{u}))}{r_j} \;=\; 0\,. \tag{5.13}$$

If κ is constant over the mesh, we can recognize the mean value coordinates in the above expression. We note that for this derivation, the dual mesh can be used as circles around the vertices (see equation (5.11)), and since the arising equations are independent of the radii of the circles, we conjecture a possible connection to circle packing. Consequently, the integration area is smaller than the total area of the mesh. This explains the loss of conformality accumulated by this discretization method, which is here traded for robustness.

In the above, we simplified the discretization of the quasi-harmonic equation (7.1) effectively to the scalar case. Now given a general (symmetric) tensor $C = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$, the integral (5.12) can be

written in the following form again using local polar coordinates

$$\int_0^{2\pi} C\,\mathbf{grad}f \cdot n d\gamma \;=\; \sum_j \int_0^{\gamma_j} r H_{T_j} C \begin{pmatrix} \cos(\gamma) \\ \sin(\gamma) \end{pmatrix} \cdot \begin{pmatrix} \cos(\gamma) \\ \sin(\gamma) \end{pmatrix} d\gamma \;,$$

and we obtain the weights associated with each edge j as

$$\begin{aligned} w_j \;=\; & \frac{1}{3r_j} \left[(a_j + 2b_j + 2c_j \sin\gamma_j) \tan\tfrac{\gamma_j}{2} \right. \\ & \qquad \left. + (a_j - b_j) \sin\gamma_j \right] \\ + \; & \frac{1}{3r_{j+1}} \left[(a_{j+1} + 2b_{j+1}) \tan\tfrac{\gamma_j}{2} \right. \\ & \qquad + 2c_{j+1} \sin\gamma_{j+1}^2 \\ & \qquad \left. + (a_{j+1} - b_{j+1}) \sin\gamma_{j+1} \cos\gamma_{j+1} \right] , \end{aligned}$$

where again indices denote edges and associated triangles.
It is easy to see that the above expression simplifies to (5.13) if $a = b$ and $c = 0$, and consequently if in this case a is constant over the mesh, we obtain the standard mean value coordinates.

5.5 Discussion

This approach accounts for directional as well as areal distortion. This is due to the newly introduced planar Jacobian that mimics the reference Jacobian. In fact, this means that we can start from any initial parameterization, possibly suffering from high distortion, as the use of the geometric tensor defined over the mesh implicitly yields a self-correcting scheme. This is illustrated by a small example in figure (5.6). For real models, we can even start from a projection (eventually forcing a convex boundary) onto a least-squares plane – ignoring foldovers. All the examples were computed in this manner, showing the robustness to high-distortion for complex data.
Successive solutions of weighted linear systems have been used for surface parameterization in [YBS04]. According to the authors, their method degenerates after few iterations. We observe that our method does not degenerate while iterating, rather it stabilizes after few iterations (3-5 in the examples). We believe that this difference arises from the fact that although the spring analogy gives a physical interpretation of differential equations, it is not always intuitive to derive accurate discretizations, like for the discrete conformal parameterization, solely based on this analogy.
It is worthwhile to note that taking powers of the tensor C extends the range of possible mappings.

5.6 Results

We applied our parameterization method to a variety of reasonably complex, non-trivial geometric models, for the planar setting. The figures (5.7), (5.8), (5.9), and (5.12) show examples, where the map is visualized either by mapping a regular checker texture, or the planar image is rendered with the original shading. The solution of the linear system within one iteration is in the order of seconds as it is typical for solving this class of problems on current hardware. Three to five iterations have been used as described above.

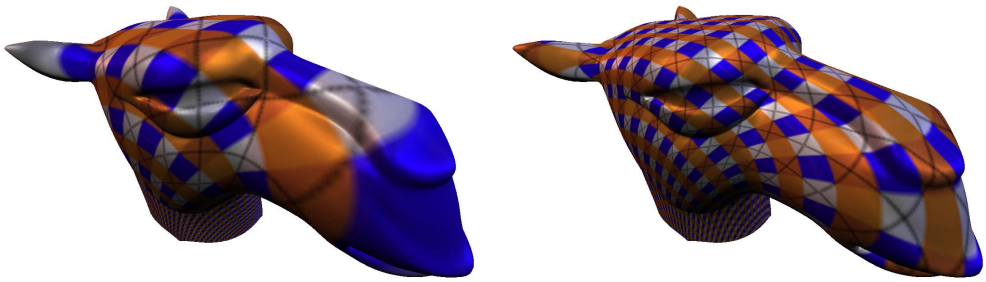


Figure 5.7: *Texturing a camel head model (*$11K\Delta$*) using a discrete conformal map ([LPRM02, DMA02]; left) and a quasi-harmonic map (right).*

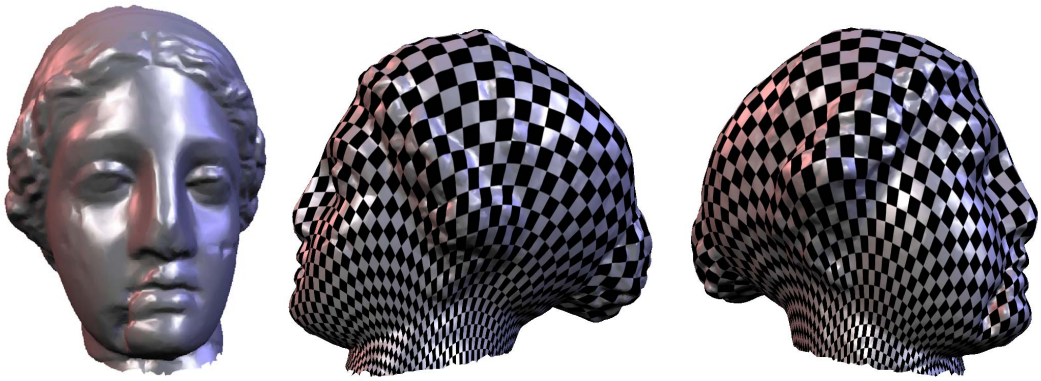


Figure 5.8: *The tensorial discrete quasi-harmonic map of the* venus *model is visualized by applying a checker board texture.*

Figure (5.11) compares the distortion of angles and triangle areas of discrete conformal maps([LPRM02, DMA02]), stretch minimization using [YBS04], and quasi-harmonic map for the *mannequin head* model depicted in (5.10). In the graphs, the distortion on the horizontal axis is plotted over the sequence of mesh elements, showing a peak for every triangle. The optimal ratio is 1. The figure shows that the quasi-harmonic map preserves conformality to a great extent while the area distortion is effectively reduced.

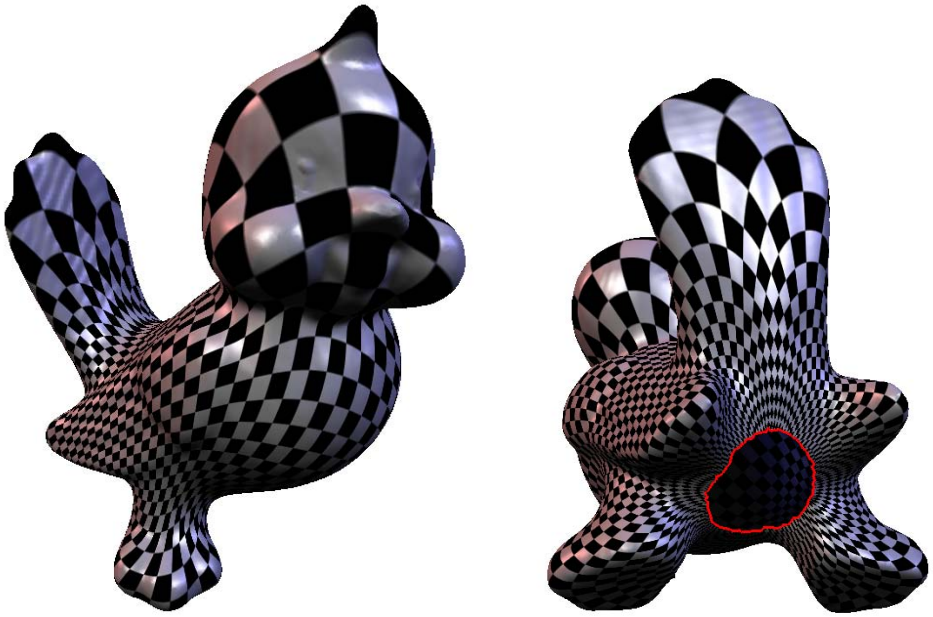


Figure 5.9: *Quasi-harmonic map of the* tweety *model ($97K\Delta$) with a hole (red boundary) cut in its bottom.*

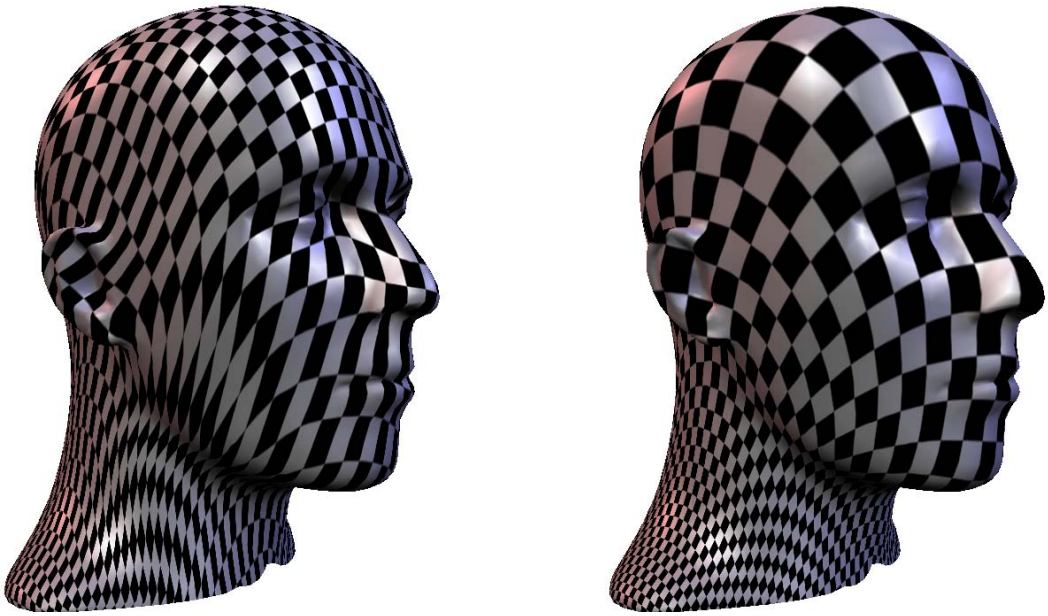


Figure 5.10: *The figure compares results obtained for the* mannequin head *using [YBS04] (left) with the tensorial quasi-harmonic map (right).*

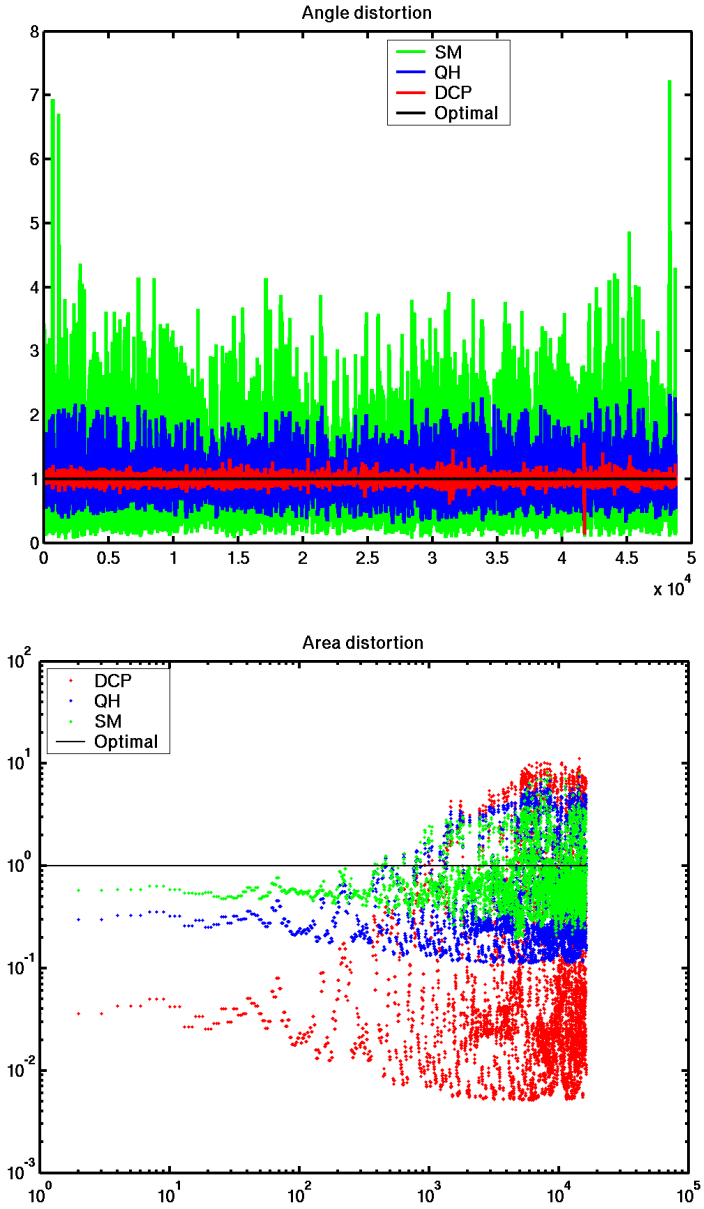


Figure 5.11: *Distortion of angles and triangle areas for the* mannequin head *model. Red: discrete conformal map ([LPRM02, DMA02]). Green: stretch minimization using [YBS04], blue: quasi-harmonic map.*

Figure 5.12: *Maps of a sculpture (*$89K\Delta$*). Left: Mean value coordinates. Right: Quasi-harmonic map. The mesh was obtained from a range scan of the wooden sculpture* Freezing Old Woman *(*Frierende Alte*, 1937) by the German expressionist sculptor Ernst Barlach (1870-1938).*

6

Compounded of Many Simples: A Differential Approach to Free Boundary Parameterization

In the previous chapter the notion of quasi-harmonic maps was introduced in order to address one of the major shortcomings of conformal mapping methods. The characterization of deformation as a tensor allowed us to recover much of the initial area distortion. However, the initial shape of the boundary still has considerable impact on the quality of the parameterization. In this chapter we take upon the task of allowing the parametric representation to evolve freely on the planar domain without *any* fixed boundary vertices. So far conformal mapping based methods need to pin at least two boundary vertices, where the choice of vertices is not obvious in practice and may affect the result to a large extent especially for surfaces with geometrically complex boundaries. Based on geometric distortion tensors and general elliptic partial differential equations, we lay out the theoretical foundation of our approach and derive the ensuing linear optimization problems. The results reflect a fair balance between high-quality parameterization and computational efficiency.

6.1 Motivation

Our approach has been motivated by the following two questions:

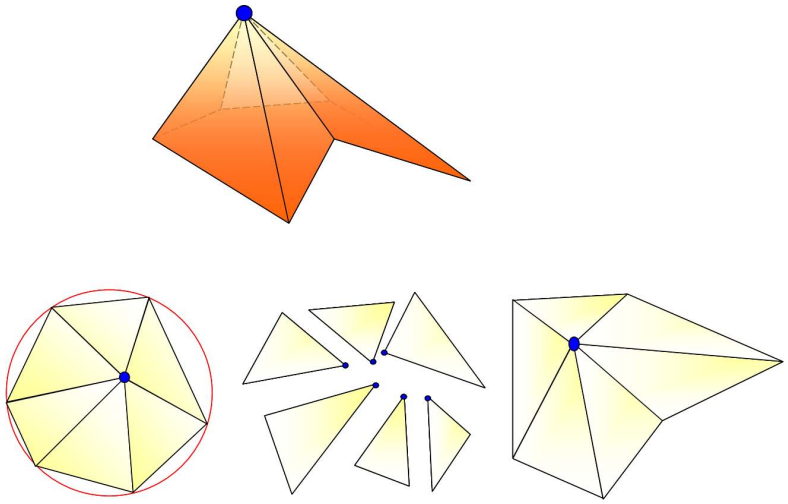
- How can we quantify the geometric distortion induced by a convex boundary discrete conformal map?
- How can we use this information for establishing a boundary-free parameterization?

We show that a fixed boundary conformal map has all the information necessary for allowing a flat mesh to evolve freely in the plane towards a configuration exhibiting less boundary distortion. Our method proceeds in a series of simple steps, each involving the solving of a linear problem. We can briefly summarize these steps as follows:

1. For a surface $\mathcal{S}$ compute an initial parameterization $\mathcal{P}_0$, preferably a conformal map with fixed boundary (figure (6.2), center left).
2. Establish a new parameterization $\mathcal{P}_1$ such that $\mathcal{P}_0 \rightarrow \mathcal{P}_1$ is a conformal map, where the image boundary tends to roughly mimic the original boundary of $\mathcal{S}$ (figure (6.2), center right).
3. To further improve the shape of the boundary such that geometric details of the original are captured and to reduce distortion, construct a new map $\mathcal{P}_2$, based on a variant of quasi-harmonic maps (figure (6.2), right).

4. In an optional step, $\mathcal{P}_2$ can be further improved by fixing its current boundary and applying discrete tensorial quasi-harmonic maps.

The core of this framework consists of steps 2 and 3, which allow the boundary to evolve freely without any constraints. This is achieved by combining distortion tensors with Poisson equations in a manner that treats all – inner and boundary – vertices the same way.



(a)

Figure 6.1: *Poisson setting on a 1-ring: Starting from the 3D surface, a prescribed convex boundary discrete conformal map is established. The tensor field* $\mathbf{C}$ *is computed from this map and then applied to the linear pieces of the flat configuration. The triangles of the discontinuous setting show less distortion and are used to setup the right hand side of the Poisson equation. The solution is a (continuous) boundary-free conformal map.*

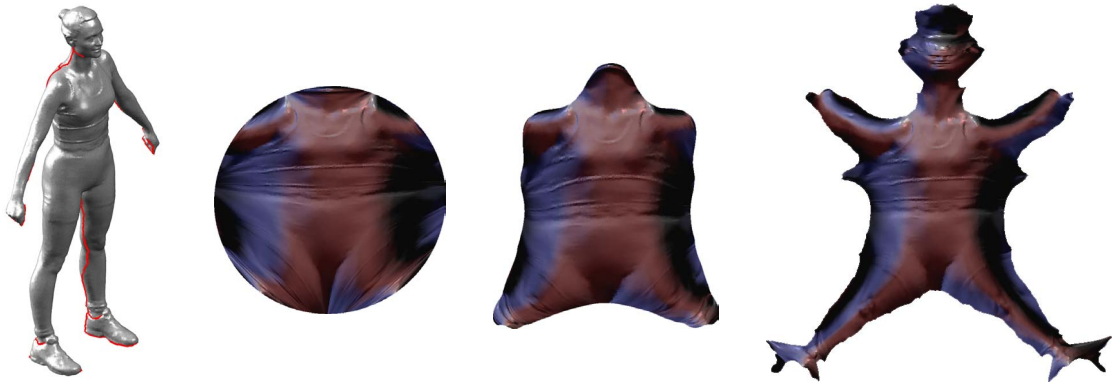


Figure 6.2: *Boundary-free parameterization of a human model (66K*$\triangle$*). From left: surface mesh with cut highlighted, initial discrete conformal map, boundary-free conformal map, and boundary-free quasi-harmonic map.*

6.2 Conformal maps

In this section we concisely characterize the different types of maps used as basic tools in our approach. We start with conformal maps and continue with quasi-harmonic maps in section (6.3).

6.2.1 Prescribed-boundary conformal maps

For the mapping function from the surface to the plane we have

$$\nabla^2 u \;=\; \mathbf{div}(\mathbf{Id}\;\mathbf{grad} u) \;=\; 0\,. \tag{6.1}$$

Given appropriate Dirichlet conditions on the boundary, the resulting parameterization is guaranteed to be a one-to-one mapping. The parameterization depends closely on the choice of weights used for the discrete operator in equation (6.1) as shown in chapter (5). In practice, the cotangent weights or the mean value coordinates are used with a fixed convex boundary, e.g., the boundary is fixed to a circle.

6.2.2 Boundary-free conformal maps

Solving the Laplace equation with Dirichlet boundary conditions is an extremely efficient way to establish a conformal map from the plane to the surface. The fixed boundary however does not necessarily respect the geometry of the original mesh as illustrated, e.g., in figure (6.2), center left. In order to improve the boundary shape, [DMA02, LPRM02] fix a small (≥ 2) number of boundary points and impose certain local equations on the rest of the boundary. [LPRM02] derive an alternative formulation of these equations from the Cauchy-Riemann equations. Although most boundary vertices are "free", the solution is sensitive to the choice of the fixed vertices. For geometrically simple boundaries, satisfactory results are obtained. For more complex boundary shapes, the results may suffer from considerable distortion or the solution might even be invalid as the flat mesh does not unfold correctly (see e.g. [SLMB05], who partially motivate their non-linear ABF++ method by this fact). Virtual boundary methods [LKL02] allow for additional degrees of freedom and provide an interesting alternative but do not seem to always provide a satisfactory solution due to the limited effect of padding with virtual vertices (see also [KGG05]).
Our approach is motivated by the Poisson equation. So instead of letting boundary conditions dictate the behavior of the whole solution, we aim at specifying the same type of conditions over the whole mesh. Hence, all vertices are treated in a similar manner, and boundary distortion is reduced simultaneously with the overall distortion.
For modeling this problem, we have to correctly specify the right hand side b of the Poisson equation (8.2). More formally, we are looking for a suitable vector field F, which can be used as a guidance field for the partial differential equation

$$\nabla^2 u \;=\; \mathbf{div} F\,. \tag{6.2}$$

We note that unlike the Laplace equation, here, the right hand side is specified for *all* vertices including boundary points as well. We refer to [PGB03, YZX$^+$04] for a discussion of this type of equation in the context of computer graphics.
The solution of equation (6.2) is very sensitive to the guidance field F and thus a meticulous setup of this field is de rigueur. Since our aim is to reduce the distortion of the planar configuration, it seems natural to characterize the right hand side in terms of an initial prescribed

boundary conformal map. In the previous chapter, we introduced a piecewise constant tensor field that locally mimics the Jacobian of the map from a given planar configuration to the surface in 3D. This tensor comes in handy as it is a key ingredient for our Poisson equation setup.
Given an initial conformal map, we define such 2×2 tensors $\mathbf{C}_j$ per triangle $j \in \mathcal{K}$. We refer to chapter (5) for details on the derivation of the tensor field $\mathbf{C}$. In contrast to the latter we *apply* the tensor field here *directly* to the planar configuration.
So for every triangle we get $T'_j = \mathbf{C}_j\, T_j$, i.e., this intermediate step can be imagined as splitting mesh triangles apart resulting in a geometrically fragmented mesh whose vertices are three times the number of triangles of the original. Figure (6.1) illustrates the local situation for a 1-ring. Considering the new coordinates as scalar fields over the initial parameterization, we obtain the right hand side $b := \mathbf{div}F$ of (6.2) as follows: We compute the gradient per triangle on the discontinuous mesh using the discretization (5.3). The divergence of this gradient field is summed at vertices shared between triangles according to the original mesh connectivity, yielding the right hand side vector b. In other words, the Laplacian of the fragmented mesh is first computed and then applied to the coordinates of the fragmented mesh. The resulting coordinates are summed according to the original connectivity, which provides the per-vertex components of b.

6.3 Quasi-harmonic maps

Quasi-harmonic maps are the second tool used in our approach. This type of maps has been introduced for prescribed boundary parameterization in the previous chapter, and here we take advantage of their properties for our boundary-free setting.

6.3.1 Prescribed-boundary quasi-harmonic maps

Discrete tensorial quasi-harmonic maps can account for distortion away from the boundary. The approach attempts to reduce the distortion by minimizing the quasi-harmonic energy functional

$$\int_\Omega (\mathbf{C}\nabla f) \cdot (\nabla f)\ ,$$

where the tensor field $\mathbf{C}$ is defined as in section (6.2.2). The partial differential equation associated with this energy is the quasi-harmonic equation

$$\mathbf{div}(\mathbf{C}\ \mathbf{grad} f)\ =\ 0\ . \tag{6.3}$$

Quasi-harmonic maps establish planar maps which mimic the original three dimensional shape not only in angles but also in area as $\mathbf{C}$ captures the properties of the Jacobian of the initial map.

6.3.2 Boundary-free quasi-harmonic maps

So far, quasi-harmonic maps have only been applied to fixed boundaries. Building upon the Poisson setting of section (6.2.2), we can easily extend the method to completely boundary-free parameterization.
Starting from boundary-free conformal maps, we aim to further improve it. This can be achieved by applying a variant of the quasi-harmonic maps but in the more general Poisson setting. Here, we solve the following differential equation

$$\mathbf{div}(\mathbf{C}\ \mathbf{grad} u)\ =\ \mathbf{div}F\ . \tag{6.4}$$

This formula differs from (7.1) only in the right hand side, which is computed similarly to the Poisson setting (6.2). Hence, we have now all tools necessary to set up the linear systems for the boundary-free maps.

6.4 Overall method

In the previous sections, we described all ingredients required for our approach. Now putting it all together, the overall method reads simply as sketched in section (6.1). We proceed in several steps, each step corresponds to one of the previously described differential settings.
In the first step, an initial *prescribed-boundary conformal map* (figure (6.2, center left)) is constructed in the standard way by solving the Laplace equation with fixed convex boundary, e.g., using a circle.
To let the boundary evolve freely, a *boundary-free conformal map* (figure (6.2, center right)) is established as a second step. Here, the guidance tensor field for the setup of the Poisson equation (6.2) is obtained from the Jacobian of the initial mapping.
The resulting planar configuration is generally non-convex and tends to roughly mimic the shape of the 3D boundary, however, it fails to capture completely the full detail of the original surface boundary. In order to further improve it, a *boundary-free quasi-harmonic map* is constructed from equation (6.4), see figure (6.2, right), this time the tensor field is computed based on the previous boundary-free conformal parameterization. The boundary of the consequent parameterization resembles its 3D counterpart much more closely.
Optionally, the result can be further improved by fixing the boundary and solving equation (7.1) for a *prescribed-boundary quasi-harmonic map*, which is exactly the formulation of discrete tensorial quasi-harmonic maps. We note that in this last step the role of the tensor field is restricted compared to the previous steps.
Finally, we note that there is no loop or iteration in this process, each step is carried out once, solving only one single sparse symmetric linear system. We refer to [BBK05] for a recent overview on efficient solving strategies for these type of linear problems. In particular, the following properties can be exploited: The two initial (discrete conformal) steps apply the same linear operator ∇^2, i.e., one can easily reuse for instance a Cholesky factorization of the system matrix and hence obtain the second map simply by back-substitution. Note that this is not possible for the subsequent (quasi-harmonic) steps, as different tensor fields are applied. However, one still can take advantage of a symbolic factorization, which depends only on the mesh connectivity, over all the four steps.

6.5 Results and discussion

We applied our approach to a variety of surface meshes with non-trivial boundaries. In fact, most models were cut explicitly to make any parameterization method feasible. Results are shown in Figures (6.3), (6.5), (6.6), (6.8), (6.10), some with the cuts highlighted for the 3D surfaces. The maps are visualized as flat meshes rendered with the original shading and/or as texture maps. In all our tests, the resulting maps are valid, i.e., bijective with no triangle foldovers. Some of the models contain holes, they were processed as is with no special treatment necessary. The size of the models is indicated for every example, and it is generally in the order of tens of thousands of triangles. Note that the mapping is inherently independent of the particular tessellation and resolution. The runtime for solving the linear systems is in the order of a few seconds for all examples, we refer to the previous section for remarks on efficient solution

strategies. Furthermore, for these kind of problems multi-grid methods are well-understood and readily available as standard tools to process even very large input. The current results neither require nor apply a supporting hierarchy. In addition, we remark that the implementation of our method is straightforward. Any existing implementation of linear parameterization can be extended with only little effort to confirm our results.
As mentioned before, linear methods with Neumann-type boundary conditions do not always lead to valid solutions [SLMB05], in particular this applies to many of the surfaces shown here. However, we note that with careful choice of (eventually more than two) fixed vertices the method often produces good results solving a single linear system. It is hard to quantify this aspect as the approach is sensitive to this choice as illustrated for a simple model in figure (6.9). (A non-linear alternative based on Green's functions remedies this problem for small boundaries [DMA02].)
We therefore compare our approach to non-linear angle based flattening (ABF) [SdS00, SLMB05]. Figure (6.4) compares the distortion of angles and triangle areas (vertical axis) plotted over the sequence of faces as they appear in the data set (horizontal axis), i.e., distortion on every triangle is visualized. The optimal ratio indicating no distortion is 1.
The diagram shows that ABF produces high quality maps as expected, and that we come close. Note that for both methods the results depend largely on boundaries or choice of cuts, respectively. For the horse model (figure (6.3)), we use the original cut of [SLMB05], and our result is still competitive. Of course, there must be tradeoffs between the efficiency and simplicity of our linear setting against the high quality of the non-linear ABF. We think, that our approach provides a fair balance between the computational efficiency of establishing discrete conformal maps and the free boundary ABF.
As mentioned before, the boundary path resulting from cutting surfaces to disk-like patches affects parameterization methods. For all our cuts we used simple Dijkstra shortest paths joining a set of marked vertices. Where indicated in the examples, we use the cuts from from [SLMB05]. Our method depends largely on the initial prescribed boundary conformal map. If there are nearly degenerate triangles (very small area) in the flat mesh, then there might be an overreaction in the subsequent steps, i.e., these triangles may end up having large areas in the boundary-free maps. This may happen for long cuts traversing a relatively small surface area (or more figuratively, volume). In fact, for the initial arc-length parameterization of the fixed boundary over the circle, such triangles will cover a large perimeter and will be extremely distorted. Figure (7.7) shows an example, the tail of the camel ends up having an area comparable to the legs. This situation could be greatly improved by optimizing the cut or using a suitable re-parameterization of the initial convex boundary curve.

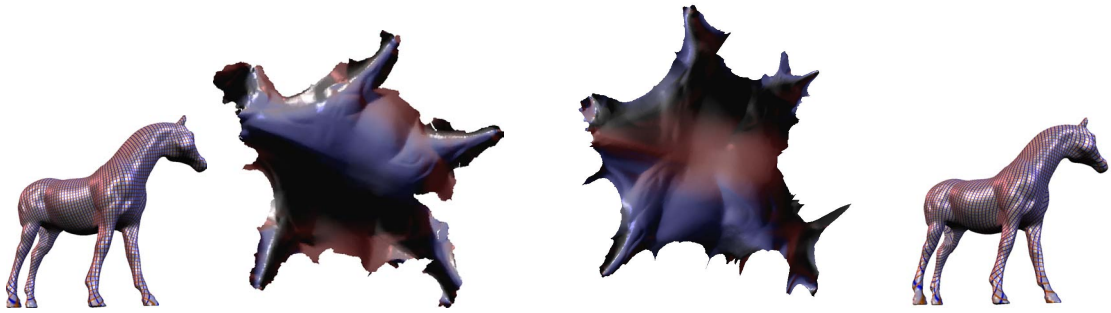


Figure 6.3: *Comparing maps of the horse model (40K△, original cut of [SLMB05]). Left: ABF. Right: Our method.*

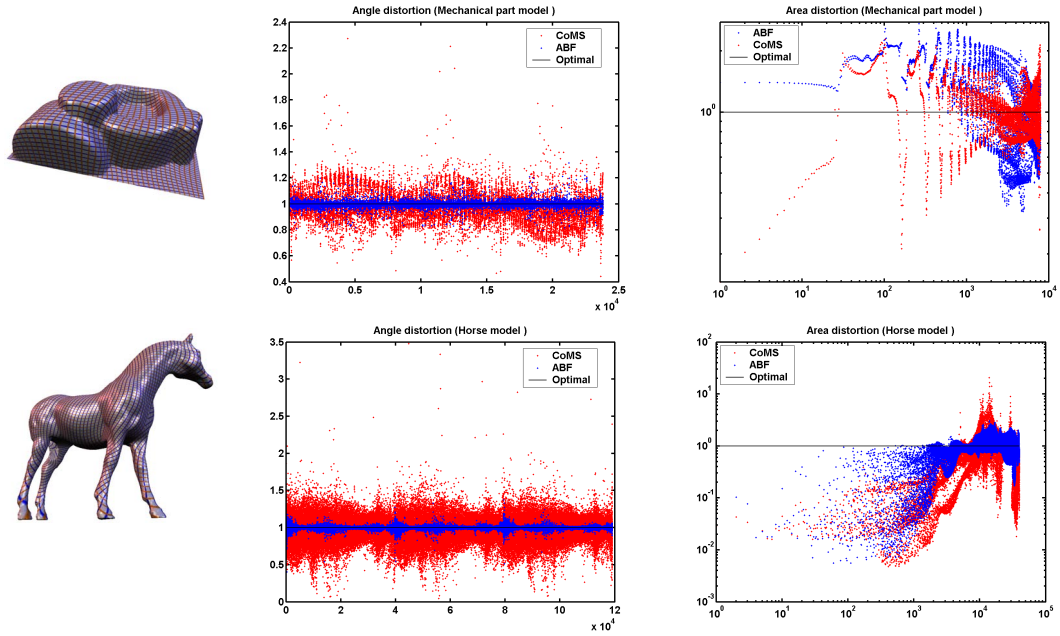


Figure 6.4: *Distortion of angles (center column) and triangle areas (right column, log-scale) for the mech-part and the horse model using the original cut of [SLMB05]. Blue: ABF. Red: our method.*

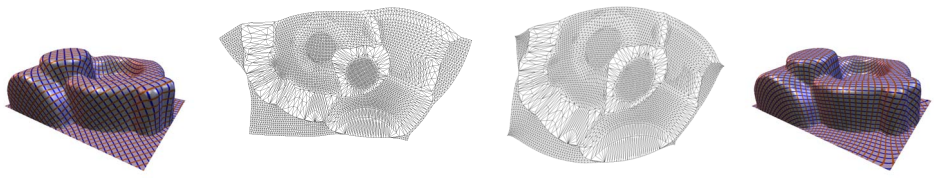


Figure 6.5: *Maps of the mech-part (8K△). Left: ABF. Right: Our method.*

Figure 6.6: *Subsequent maps of the gargoyle model (20K△). From left: prescribed boundary conformal map, boundary-free conformal-map, boundary-free quasi-harmonic, improvement by prescribed boundary quasi-harmonic maps.*

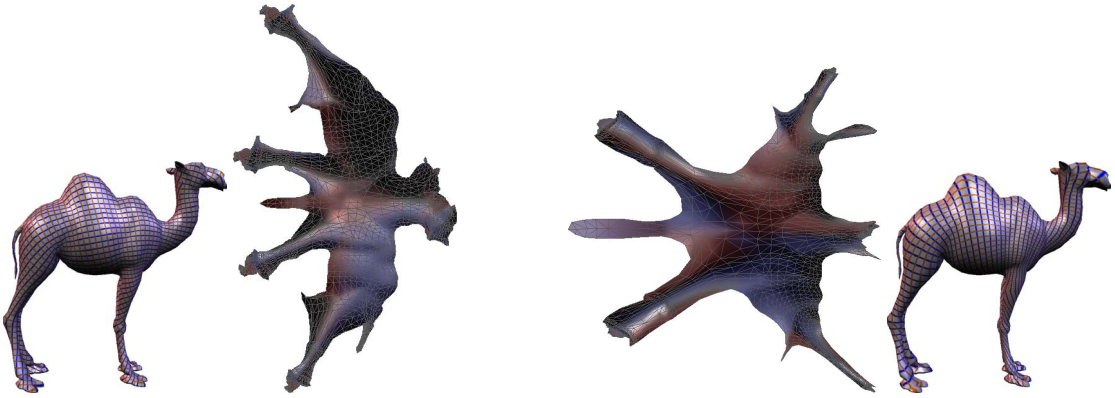


Figure 6.7: *Mapping the camel model (4k△, original cut of [SLMB05]). Left: ABF. Right: Our method. For the tail, the system overreacts due to extreme distortion for the initial prescribed boundary conformal map.*

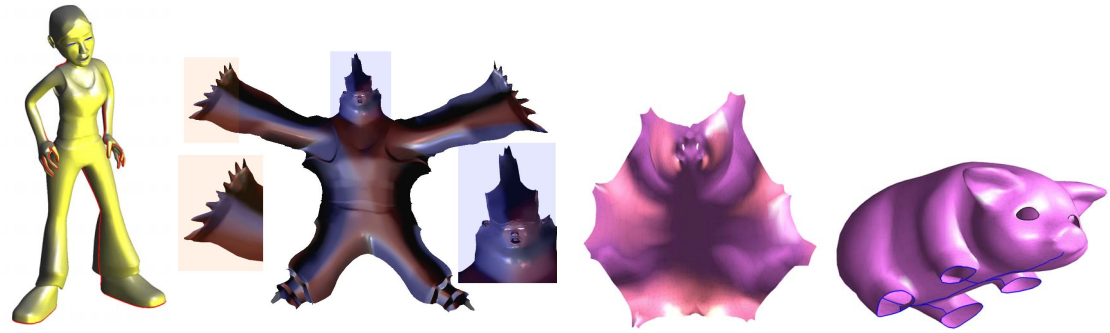


Figure 6.8: *Mapping another (four-fingered) human model (20K△) and a pig (4K△) model to the plane. For both, the eyes are holes, which do not require any special treatment.*

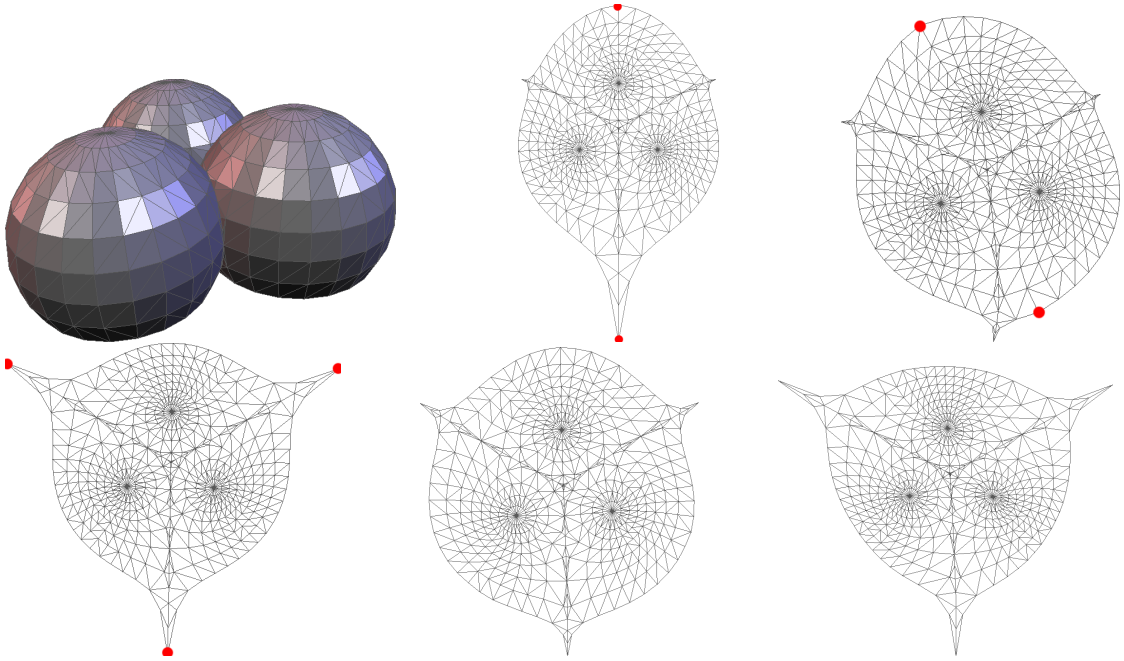


Figure 6.9: *Mapping the balls model (top left, 1300△), using [DMA02] fixing different boundary vertices depicted in red. We compare to the results of ABF (bottom center) and our method (bottom right).*

Figure 6.10: *More maps computed with our method: The head model (16K△) self-intersects in 3D, the flat mesh shows no foldovers. Dinosaur (48K△). Double torus (2K△). Holes like the windows of the beetle model (2K△) do not require any special treatment. Santa (114K△).*

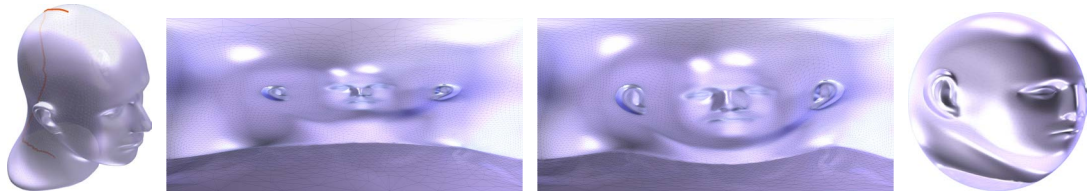
7

Spherical Parameterization

In theory any higher genus model can be described as a union of patches through an atlas. The discontinuities induced by this segmentation render it unattractive, especially in the particular case of genus-zero surfaces for which the sphere is the most natural domain. The fact that many of the available geometric models are indeed homeomorphic to a sphere makes spherical parameterization an appealing geometry processing tool. The past and recent research interest in this topic echoes its relevance to applications and reflects the demand for efficient algorithms for establishing low-distortion spherical maps. Despite the recent advances in the field, the construction of spherical maps still raises theoretical and numerical challenges. The spherical setting is much more complex than the planar one, and any robust method to solve this problem cannot solely rely on simple modification or extension of traditional planar methods as pointed out in e.g., [FSD05]. Furthermore, the additional spherical constraints and the ever increasing need for processing large input data raise challenging theoretical issues such as convergence guarantees and validity, along with practical ones, associated with finding robust numerical schemes and efficient custom solvers. Currently, methods with the most sound theoretical foundation are not in measure of addressing even moderately sized problems numerically [GGS03].
In this work, we present a novel approach to spherical parameterization, where computation time is dominated by solving only linear systems. Our method relies on setting the problem in a curvilinear coordinates system, hence reducing it to a two-dimensional problem. The singularities of this coordinate system are effectively addressed by removing the poles and introducing a date line connecting the poles. This way an initial harmonic map can be established following the outline in [BGK92]. In general, this initial map suffers from unacceptable distortion. As a novel contribution we undertake further steps to improve distortion. The merit of our new method is that we perform the crucial distortion improvement of the initial map in curvilinear coordinates as well, based on a variant of quasi-harmonic maps. This way we benefit from guarantees on validity as well as the availability of highly efficient and robust solving strategies, while at the same time the overall algorithm and its implementation are conceptually simple. In a final step, we apply a local distortion improvement on a small sub-patch along the date line in order to account for the unavoidable distortion induced from the Dirichlet boundary conditions.
Our results on non-trivial and considerably large input meshes show high-quality maps with a fair balance between angle and area distortion. Computation times are significantly lower than those of preexisting methods.

7.1 Initial parameterization

In order to solve the spherical parameterization problem we need to pose the problem in a computationally tractable way. This involves adopting one of the many possible characterizations of



(a) (b) (c) (d)

Figure 7.1: *Curvilinear spherical parameterization in brief: (a) The poles are identified on the shape by the user (or automatically). The path from pole to pole constitutes the date line along which the mesh is cut open. (b) The poles are removed and an initial solution in curvilinear coordinates is obtained on the remaining mesh. This initial map suffers from unacceptably high distortion. (c) Therefore it is improved in a second step taking into account spherical distortion. Both, the initial and second step operate in the curvilinear domain with fixed boundaries. (d) The secondary solution is then lifted back onto the sphere. The poles are restored, and a local spherical relaxation is performed in a region along the date line. The runtime for the mannequin head model with 23KΔ is 3 seconds (3.2GHz P4 laptop).*

the sphere as a mathematical object. In Euclidean space, a natural description of point locations on the sphere is achieved through the curvilinear or polar coordinates represented by the angles (θ, ϕ). In this orthogonal coordinate system only two parameters are needed to characterize point positions on the sphere. Certainly, this representation is more compact than its Cartesian counterpart where three parameters are required.

In curvilinear coordinates, the spherical parameterization problem reduces to defining two appropriate scalar fields over the surface, the azimuthal angle $\theta \in (0, 2\pi)$ also known as longitude, and the polar angle or latitude $\phi \in [0, \pi]$. Although this setup simplifies the problem to a great extent, it exhibits clear limitations which are, in fact, inherent to the coordinate system itself: The first one is the *pole singularity* — the longitude spans the whole range at the poles. The second one is the *periodicity* of the longitude range. The pole singularity can be addressed by first excluding the north and south poles from the problem setup and then reinserting them at the end of the optimization process. On the other hand, the periodicity of θ requires the construction of a date line which connects the poles and marks the beginning and the end of the range. The date line can be efficiently setup, e.g., as shortest path between poles (see also section (7.4)).

The mesh resulting from cutting along the date line and removing the poles is topologically equivalent to a disk, and we can readily profit from existing techniques developed for planar parameterization. A first attempt along these lines was proposed by [BGK92]. Their method proceeds by solving the Laplace equation for the pair (θ, ϕ) over the domain $[0, 2\pi] \times (0, \pi)$. The north and south pole are assigned the ϕ-values 0 and π, respectively. Technically the domain of ϕ can be represented as $[\epsilon_1, \pi - \epsilon_2]$ where ϵ_1 and ϵ_2 are very small. In our implementation we used a value of 0.02 for both, however, the method as a whole is insensitive to the chosen value as we will see in section (7.3). The solution to the Laplace equation can be efficiently carried out using either the cotangent weights or the mean value coordinates discretization of the Laplacian operator. Vertex positions on the sphere are then given by the usual polar-to-cartesian mapping

$$x = \cos\theta \ \sin\phi, \quad y = \sin\theta \ \sin\phi, \quad z = \cos\phi \ .$$

A simple computation of the first fundamental form of this mapping from polar to cartesian coordinates reveals that it is neither conformal nor equal-area. Consequently, the resulting composite map from the surface onto the sphere enjoys neither properties. This is not a limitation in itself as

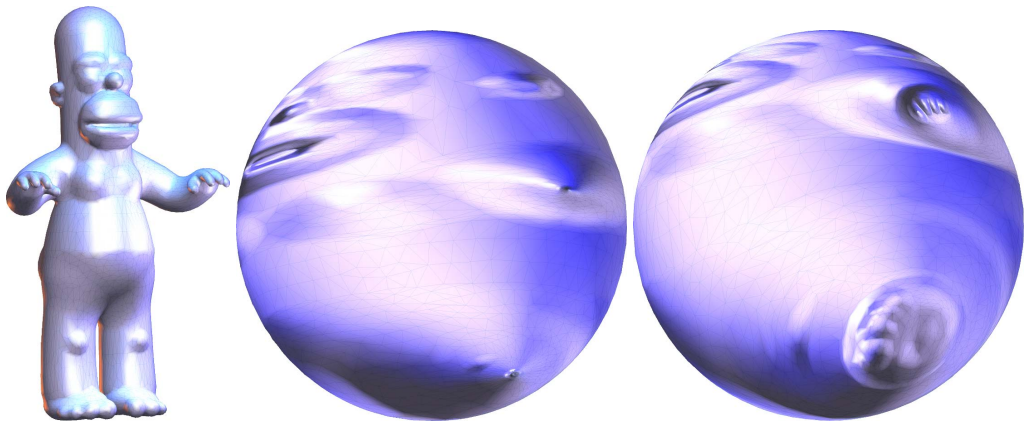


Figure 7.2: *Comparison of the initial (center) and the secondary map (right) for the Homer model. The view puts emphasis on the extremities, only after improvement the left arm and leg develop adequately.*

the results from this initial parameterization reflect well behaved maps, see figure (7.2), although they may suffer from unacceptable high area distortion as noted also in [BGK92]. Furthermore, there exist no single conformal map from a finite planar domain onto the whole sphere. On the other hand aiming only for a conformal mapping may yield in general results which exhibit high area distortions, see figures (7.9) and (7.11). Our aim is to establish a mapping which fairly balances angle and area distortion, a desirable property for the spherical case as also discussed in [FSD05]. Furthermore, we wish to be able to process large meshes efficiently. So in order to avoid a costly non-linear optimization over the sphere we wish to perform the optimization in the plane and reduce it to a linear problem.

7.2 Secondary parameterization

In chapter (5) we introduced tensorial quasi-harmonic maps for improving the distortion of planar parameterization. In the same spirit we develop a method for improving the initial spherical parameterization by incorporating a measure for spherical distortion. Arguably the most natural choice to quantify such distortion is the Jacobian of the mapping. However, it is not obvious how to incorporate this 3×3 tensor into the current two-dimensional curvilinear coordinate setting. Here, we restrict ourselves to the determinant of the Jacobian of the spherical mapping which quantifies area distortion. On each triangle T of the input mesh $\mathcal{M}$ we define

$$\kappa_T = \left(\frac{\sum_{T_i \in \mathcal{M}} A(T_i)}{\sum_{T_i \in \mathcal{M}} A'(T_i)} \right) \frac{A'_T}{A_T} \,,$$

where A'_T measures the area of T on the sphere, and A_T corresponds to the area on the initial surface.
The secondary mapping can be then obtained as the solution of the the scalar quasi-harmonic equation

$$\mathbf{div}(\kappa \, \mathbf{grad} \, U) \;=\; 0$$

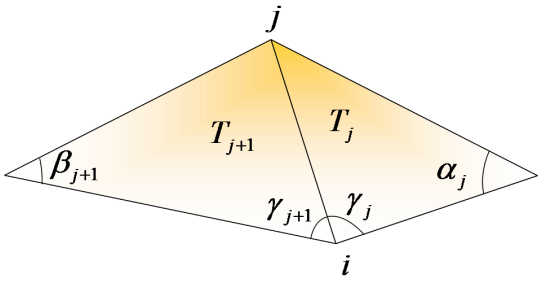


Figure 7.3: *Local configuration for an edge* (i, j)*. The coefficients* κ_j *and* κ_{j+1} *are associated with the triangles* T_j *and* T_{j+1}*.*

in terms of the pair (θ, ϕ) with similar boundary conditions as for the initial mapping. In our discrete setting, where the support of the Laplacian operator is restricted to the 1-ring of a vertex, the parameterization problem reduces to solving the following equation for all internal vertices

$$\sum_{j \in \mathcal{N}_i} w_{ij}(\mathbf{U}_j - \mathbf{U}_i) \quad = \quad 0 \, .$$

A direct discretization based on defining linear basis function over the triangles, see chapter (5), yields

$$\frac{1}{2} \sum_j (\kappa_j \cot \alpha_j \; + \; \kappa_{j+1} \cot \beta_{j+1})(\mathbf{U}_j - \mathbf{U}_i) \; = \; 0 \, .$$

Where scalars κ are defined per triangle, i.e., κ_j corresponds to triangle T_j. In the above expression we can recognize the cotangent weights generally associated with the discretization of the Laplacian when κ is constant over the mesh as discussed in chapter (4).

Alternatively, for the mean value coordinates, we have from chapters (4) and (5)

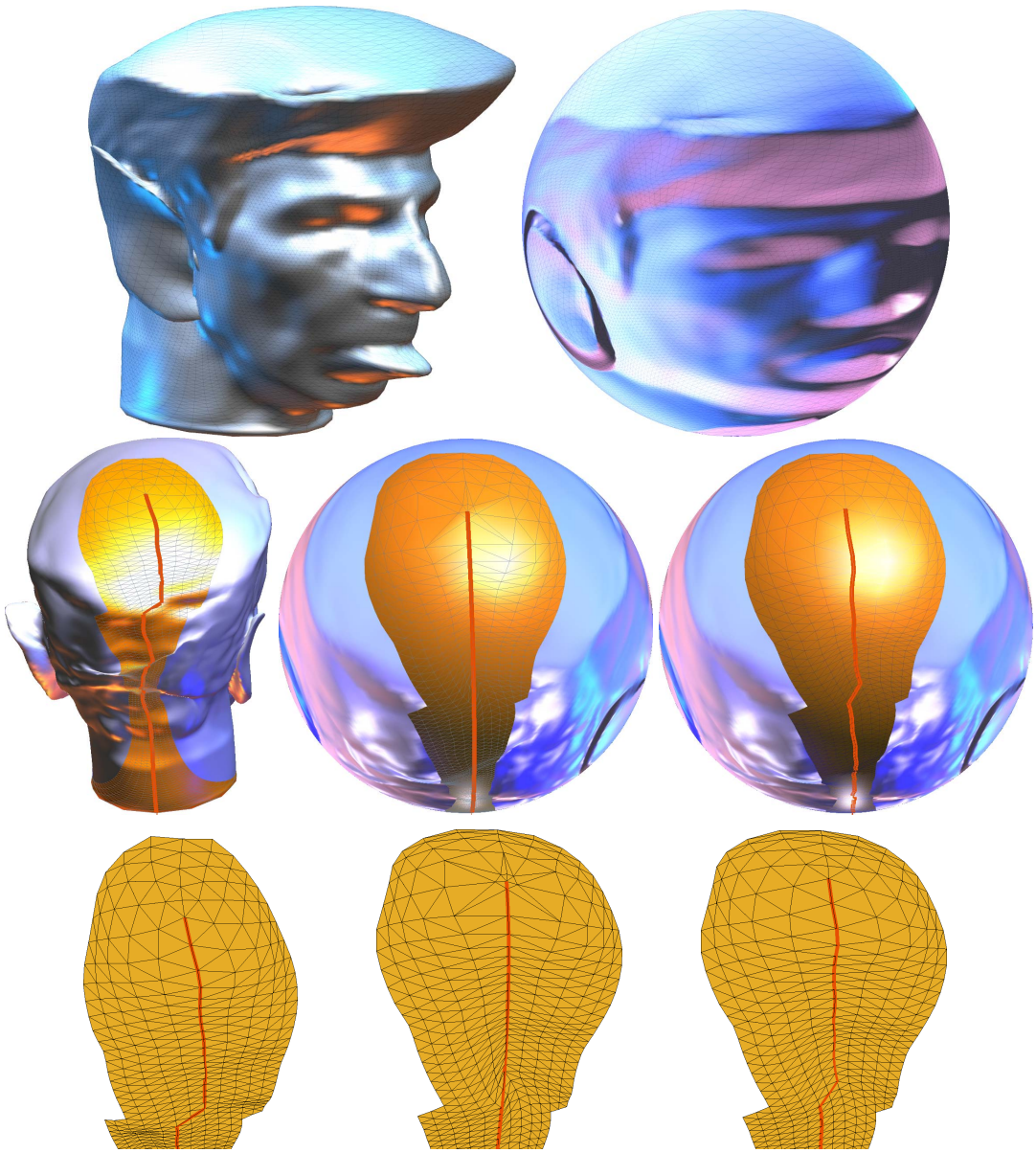
$$\sum_j \left[\kappa_j \; \tan \frac{\gamma_j}{2} + \kappa_{j+1} \tan \frac{\gamma_{j+1}}{2} \right] \frac{(\mathbf{U}_j - \mathbf{U}_i)}{r_j} \; = \; 0 \, .$$

If the function κ is constant over the mesh, the above expression reduces to the mean value coordinates. The later coordinates are insensitive to the quality of the triangulation in the sense that they are guaranteed to be positive.
In this framework, the resulting solution can be further improved by reiterating the same process until convergence. For all our results two to three iterations revealed to be largely sufficient.

7.3 Local domain distortion reduction

As our spherical mapping stems from lifting a quadrilateral patch onto the sphere, the result is expected to have a higher distortion around the poles and along the date line (see figure (7.4)). In order to overcome such artifacts, we define a sub-patch around the date line by choosing, e.g., third to fifth order neighborhood (five rings are used in our implementation), and we perform tangential Laplacian relaxation only on the sub-patch. Then the algorithm reads as simple as

1. For each vertex i of the sub-patch



(a) (b) (c)

Figure 7.4: *Effect of local distortion reduction along the date line. The highlighted region shows a five-neighborhood of the date line for the original model (a), before (b) and after (c) local improvement (close-ups in bottom row). Only this region with fixed boundary is used for relaxation.*

- $\mathbf{x}_i := \mathbf{x}_i + [\mathcal{L}(\mathbf{x}_i) - \langle \mathcal{L}(\mathbf{x}_i), \mathbf{x}_i \rangle \mathbf{x}_i]$ (update)
- $\mathbf{x}_i := \mathbf{x}_i / ||\mathbf{x}_i||$ (back projection)

2. Repeat step (1) until convergence.

In all our experiments three iterations were sufficient. A simple choice of $\mathcal{L}(\cdot)$ would be to use the uniform tangential Laplacian operator. However, such operator ignores the geometry of the mesh and may cause features to fade out along the sub-patch. Since the positivity of weights is crucial for the tangential Laplacian, the mean value based operator seems to be more appropriate in this step.
Furthermore, since the boundary of the sub-patch is fixed, there is *no risk* of slippage or folding and collapsing of the mesh onto one region of the sphere. Such limitations are in fact common to spherical parameterization methods based solely on the tangential Laplacian operator, and they do not apply here.

7.4 Results and discussion

We conducted experiments on a variety of meshes. The implementation of our method needs only a simple modification of existing planar mesh parameterization methods. All results are bijective maps and reflect a good balance between area and angle distortion as illustrated in figures (7.1), (7.5), (7.6), and (7.7). Since our method is based on a composition of several maps, the validity of each of them guarantees the validity of the whole map. In theory, our extended mean value based weights are all positive and thus ensure a bijective map. Furthermore, our tangential relaxation on the sphere is only local and thus there is no risk of mesh slippage. This way, all our intermediate maps are valid and thus the resulting map is guaranteed to be bijective. Typical timings of our method are in the order of a few seconds for meshes with tens of thousand of triangles. This confirms that our approach is significantly more efficient than preexisting methods.
We compare our results to the results of tangential relaxation methods (which lack convergence guarantees in practice) in figure (7.9), and to the practical approach of [SYGS05] in figure (7.8). We second the visual inspection of the figures with numerical charts comparing their respective area and angle distortion, see figures (7.10) and (7.11). We compute the angle distortion as the ratio of angles of the result to the input. The area distortion is computed similarly and scaled accordingly by the ratio of total areas. In the polar charts, the values are placed using the distortion value as the radius and the triangle or angle index as the polar angle.
At the current stage, the choice of the poles is not automatic. We do not see this as limitation as it can be useful for aligning spherical maps. In the following, we provide simple guidelines for defining the date line. The poles and the date line should in general reflect the symmetry of the models when they exhibit symmetry traits. The date line should be as straight as possible and the poles should have sufficient distance to allow the surface to evolve correctly but not too far away to cause additional stretch. For all our examples, we used the Dijkstra algorithm for the computation of shortest paths.

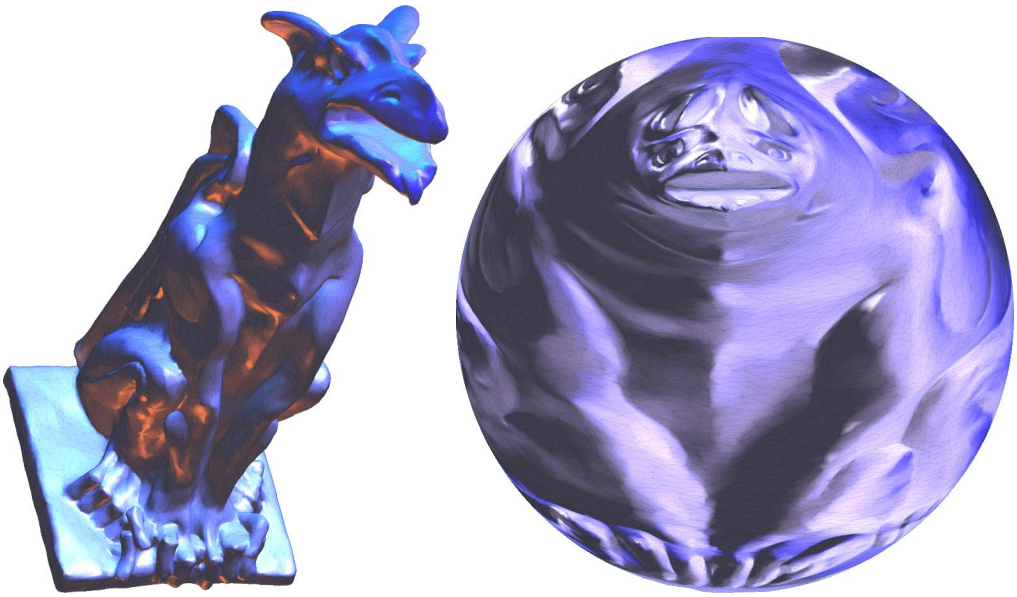


Figure 7.5: *Spherical parameterization of the gargoyle2 model (50KΔ), runtime 7s.*

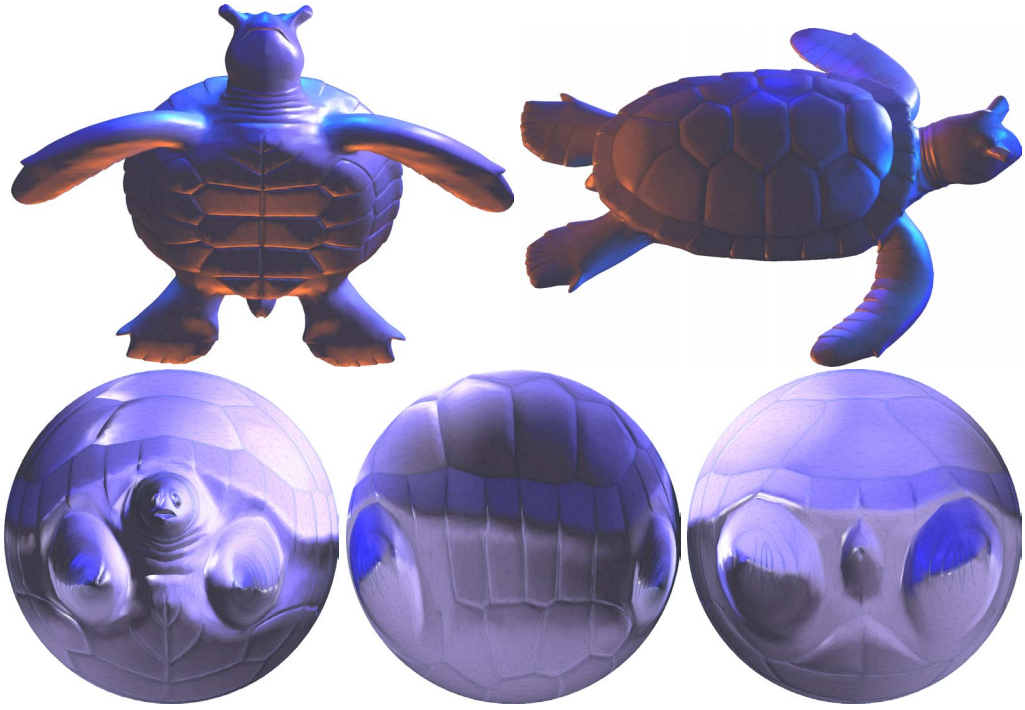


Figure 7.6: *Different views of the turtle model (38KΔ), and its spherical parameterization viewed from the front, the side, and the back.*

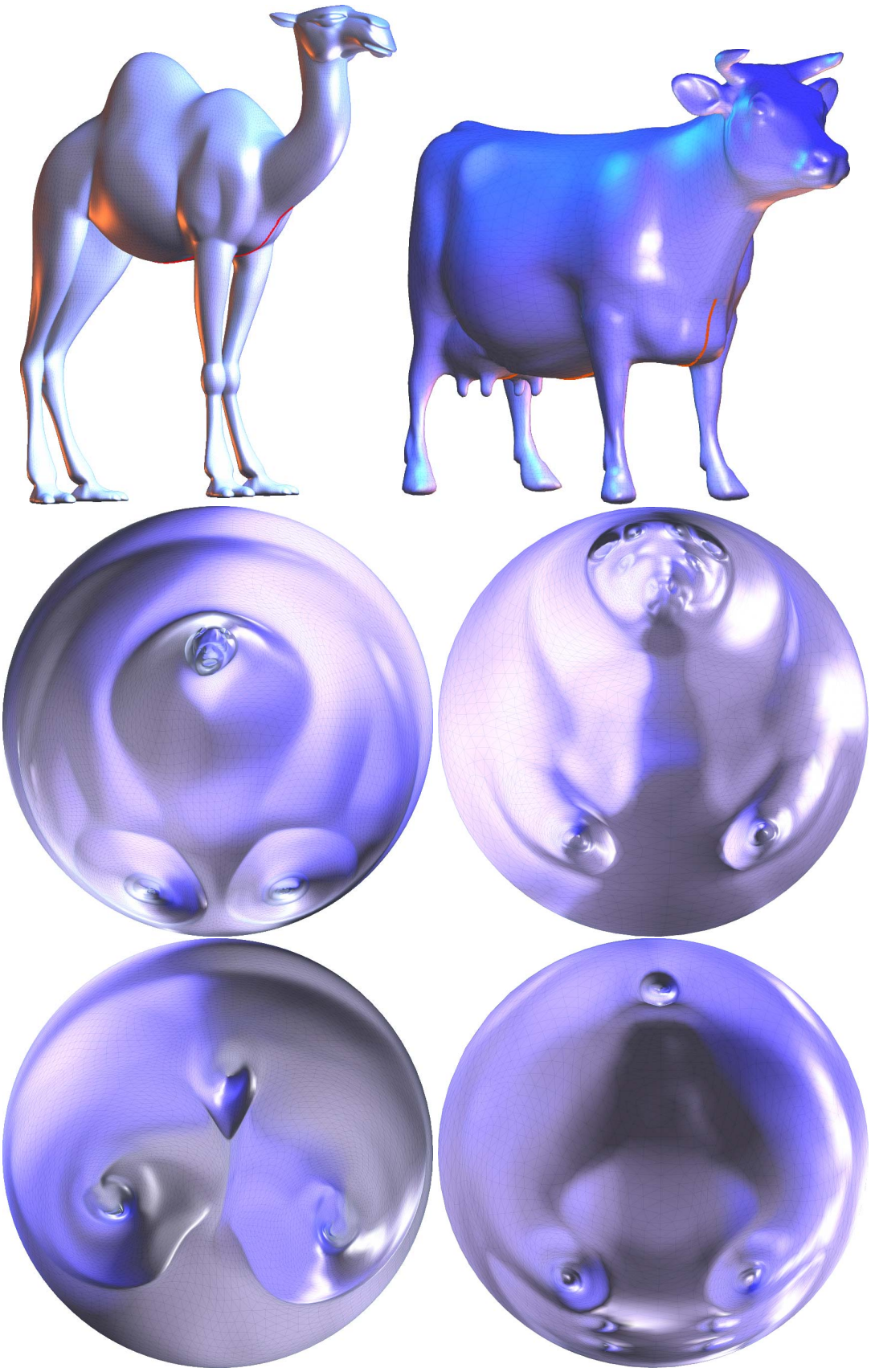


Figure 7.7: *Features such as the long legs of the camel model (78KΔ) and the legs and horns of the cow model (23KΔ) are extreme challenges for most spherical parameterization methods. Our method efficiently embeds the models on the sphere.*

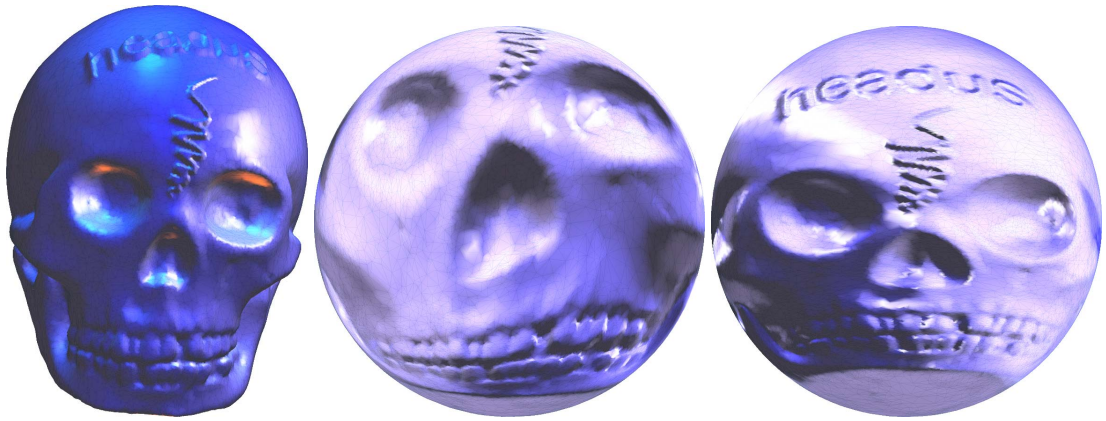


Figure 7.8: *Comparison of spherical parameterization results for the skull model 40KΔ. Center: using [SYGS05]. Right: our method. Overall runtimes are 132 and 5s, respectively.*

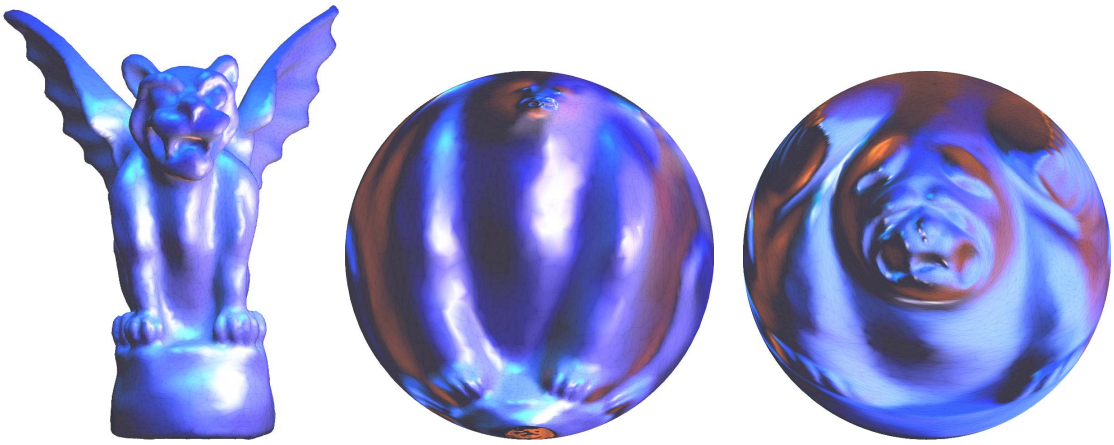


Figure 7.9: *Comparison of spherical parameterization results for the gargoyle model (20KΔ). Center: [KVLS99, Ale00] using mean value discretization. Right: our method.*

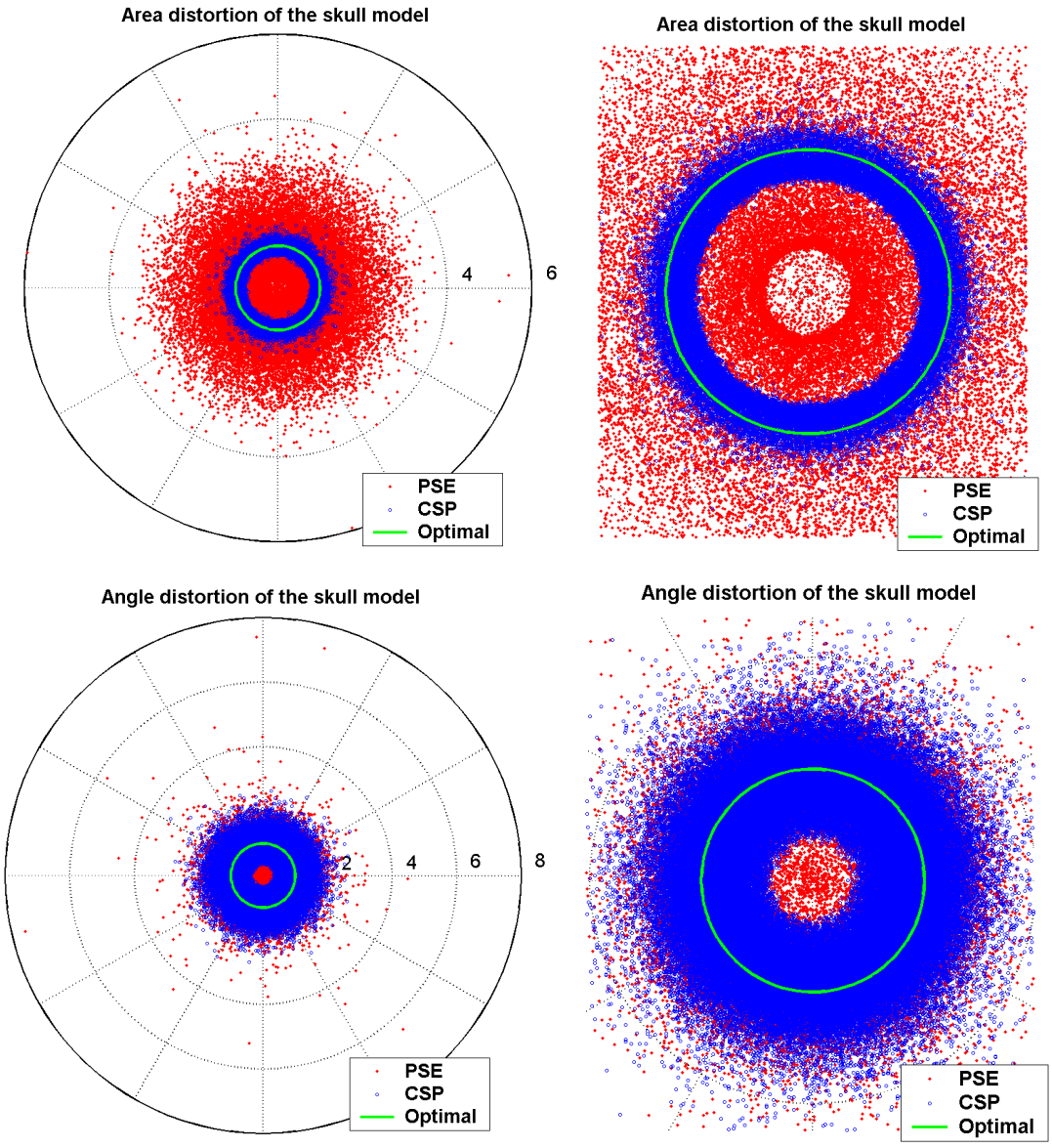


Figure 7.10: *Comparison of the distortion induced by the method of [SYGS05] in red and our method in blue for the skull model. The quality of a map is based on how well distortion values localize around the optimal solution (green). Left: Area distortion and a zoom-in on the optimal region. Right: Angle distortion and a zoom-in.*

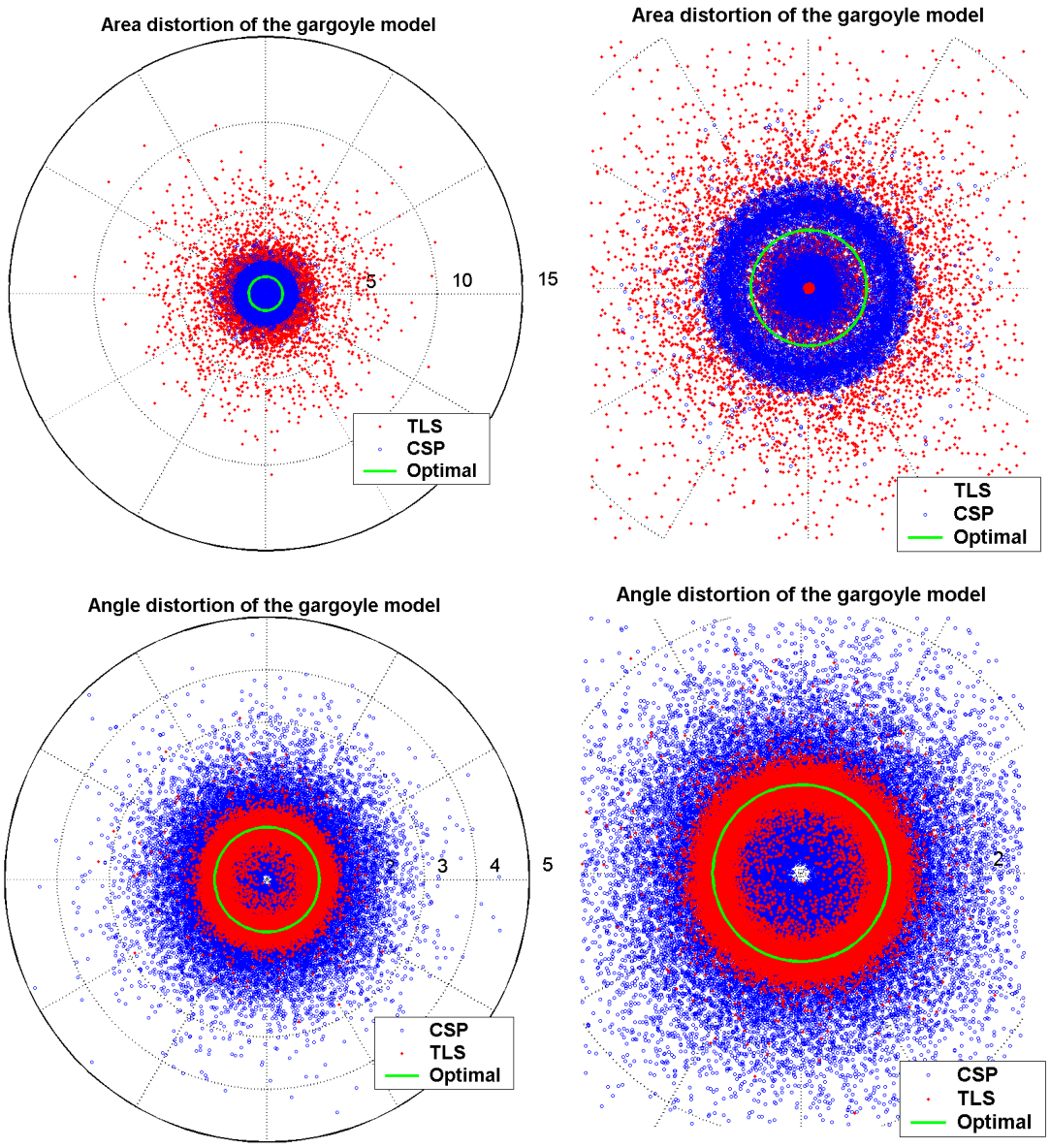


Figure 7.11: *Comparison of the distortion induced by tangential relaxation [KVLS99, Ale00] in red and our method in blue for the gargoyle model. The quality of a map is based on how well distortion values localize around the optimal solution (green). Left: Area distortion and a zoom-in on the optimal region. Right: Angle distortion and a zoom-in.*

8

Mesh Deformation

One of the widely adopted representations of 3D objects by the computer graphics community is the 2-manifold mesh. In recent years, many digital geometry processing algorithms operating on such mesh structures were developed. These methods generally deform the surface geometry and/or alter the mesh connectivity.
In many applications it is important to preserve the local shape properties under global changes of the surface mesh. The best-known example is probably surface editing: A user selects and drags a subset of vertices to a new location in space, and within a region of influence the shape should follow this manipulation in a natural way. This means that the overall shape follows the specified global deformation while geometric details like, e.g., folds and wrinkles, are locally deformed in a way that their characteristic appearance persists. To be useful in practice, such an editing tool must be intuitive and provide interactive response.
In this chapter, we describe interactive surface deformation tools based on a linear framework. We identify the propagation of local transformations over the surface as a key technique: This interpolation plays a vital role because it ensures that the overall deformation looks natural and that the required geometric detail is preserved. We establish smooth harmonic scalar fields over the mesh to guide this process.
In addition, we consider a different surface editing scenario: deformation transfer. In this scenario, manipulation of a reference model is transferred and applied on a target model, e.g., copying an animation sequence of an object to another similar object. Instead of the explicit specification of complex deformation, the user provides only a small number of corresponding markers on the two models. These are used for establishing correspondence which essentially guides the transfer of local deformations. As a result, the target object mimics the global deformations of the reference object.
As basic technique, we apply harmonic fields, this time for efficiently finding correspondence. We show that this second contribution fits nicely in the overall framework, despite the conceptual difference of this second surface deformation scenario. Harmonic guidance provides a natural and efficient tool for interactive and intuitive surface deformation: the required scalar fields provide inherent smoothness, and they are constructed at virtually no extra cost. This is because all required computational ingredients are available in form of the linear reconstruction operator necessary for applying the deformations.

8.1 Overview

We construct harmonic fields over a manifold and apply them to guide local deformations for surface editing and to establish correspondence for deformation transfer. We show that both problems can be addressed by a unified mathematical framework, where the same linear operator

is applied in all stages: decomposition, propagation of local deformations, correspondence, and detail preserving shape reconstruction.
The aim here is not to present a completely new shape deformation technique. Instead, we propose our novel interpolation as a natural ingredient to [YZX+04], which provides a basic mesh editing framework. Here, the propagation of local deformations is an essential stage for surface manipulation, as the linear reconstruction operator alone cannot account for local rotations and scaling. To achieve fair and efficient propagation, we propose harmonic fields, which are smooth scalar functions in the manifold (section (8.3)). We believe that this construction is natural, we show that it is efficient. Also, we remark that it avoids an approximation of – generally non-smooth – geodesic distance fields for (uniform, linear, or Gaussian) blending.
Deformation transfer requires an additional phase for establishing correspondence between reference and target shape. We propose to use harmonic fields in a novel approach: Inspired by the notion of barycentric coordinates, we establish a family of – not necessarily independent – coordinate functions over the manifold. The partition of unity property of these coordinates provides a convex combination setting, which is used to match corresponding pairs of triangles (section (8.4)).
Our method is very different from the iterative deformation process in [SP04]. Working entirely in the 2D manifold, it does not depend on absolute coordinates, which makes it less sensitive to the initial alignment of the shapes, in general a prerequisite for such iterated closest point algorithms. Also, we are not applying a non-linear optimization process, and in particular no user parameters are required for appropriate blending of an energy functional.
Our results confirm that we find good correspondence efficiently using relatively few markers, hence few user interaction. Again, we apply only additional back-substitution, using the same linear operator as within all phases of the whole deformation process.
Finally, we show as an additional contribution, that reasonable transfer can be achieved differently and even more efficiently by harmonic guided interpolation of deformations at the markers (section (8.5)).

8.2 General differential setting

In the following sections, we will consider the steady-state elliptic equation

$$\nabla^2 u \;=\; f\;, \tag{8.1}$$

with appropriate boundary conditions. For a non-null function f, this setting is called the *Poisson equation*.
We consider piecewise linear functions over triangulated manifolds, which can be scalar fields or multi-dimensional vector fields. The discretization of this setting is well-known and leads to a sparse linear system

$$Lu \;=\; b, \tag{8.2}$$

where the matrix L represents the discrete Laplace-Beltrami operator matrix.
Equation (8.1) and the associated linear system (8.2) are the basic tools applied throughout this work. The same system will be solved for different right hand sides, in the sense of either the Laplace equation or the Poisson equation. This means effectively that the matrix L is decomposed only once and subsequent solutions are obtained from back-substitution alone.

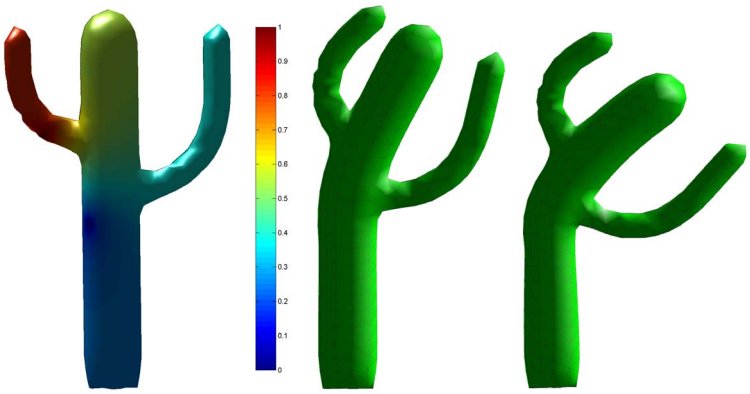


Figure 8.1: *A simple edit: The visualized harmonic field is used as guidance for bending the cactus (left). Here, the field is defined by one source (red) at the tip of the left arm and one sink (blue) below the middle of the trunk. The result is shown in the center image. Notice the different propagation of the rotation compared to the edit on the right, where three sources on all arms were chosen (without picture).*

8.3 Surface editing along harmonic fields

For intuitive surface editing, the user interacts only with a small number of vertices, while the system automatically places all other vertices within the selected region of interest in a natural, detail preserving way. Hence, the editing tool acts on a submesh of interest, whose vertices are classified as fixed, edited, or free. The last class includes the majority of vertices, whose new positions are to be determined. Fixed vertices just stay in place and impose boundary constraints. Edited vertices are positioned by the user, and the relocation of a region of such vertices defines a deformation. Alternatively, a deformation can be directly prescribed for the vertex, e.g., for twisting.

In this section, assume that the same deformation is defined for all edited vertices, i.e., a single rigid mesh region is repositioned, as is the case for the majority of manipulations. In section (8.5), we show how multiple local deformations can be interpolated simultaneously.

Given this local deformation for the edited vertices, it should be interpolated in a natural way over the whole (ROI) manifold in order to compute the global shape edit. We achieve this by using harmonic fields for deformation guidance.

A harmonic function h satisfies the Laplace equation $\nabla^2 h = 0$. For discrete scalar harmonic fields, we prescribe the value 1 as boundary conditions for the edited vertices acting as sources, and 0 for fixed vertices, which we denote sinks. Solving for $Lh = 0$, with respect to these boundary conditions, the resulting harmonic functions smoothly blend between 0 and 1 with no local extrema other than sources and sinks. As before, L denotes the discrete Laplace-Beltrami operator. Due to the chosen discretization the gradient flow of the fields respects the intrinsic surface geometry, and it is independent of the particular tessellation of the surface. Figure (8.1, left) visualizes a harmonic field over a simple surface mesh using a color table.

We decompose the local deformation into scaling and rotation, the last represented by unit quaternions. At each vertex, the value of the harmonic field is used to blend between the source deformation with the identity fixed at sinks. The result is a smooth deformation field over the manifold.

For the reconstruction of the globally edited surface from this deformation field we follow the approach in [YZX$^+$04] based on solving the discrete Poisson equation (see also [PGB03]). By

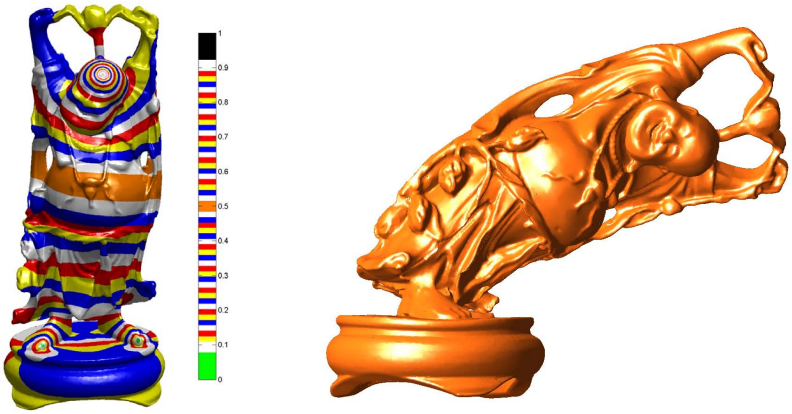


Figure 8.2: *The harmonic field on the Happy Buddha model, visualized left, guides the bending and twisting edit. This is the largest mesh used with $220K$ triangles, two sinks at the feet and one source on the head were applied.*

sampling the deformation field at the barycenters of the triangles, we obtain a piecewise constant field, i.e., deformations per triangle. Then each triangle is deformed separately, which yields a deformed but fragmented, discontinuous mesh. For integration of the right hand side of the Poisson equation, we compute the gradients for the deformed triangles. The divergence of this gradient field is summed at vertices shared between triangles according to the original mesh connectivity, yielding the right hand side vector b. The solution of the associated Poisson equation $Lx = b$ provides the new vertex positions x.

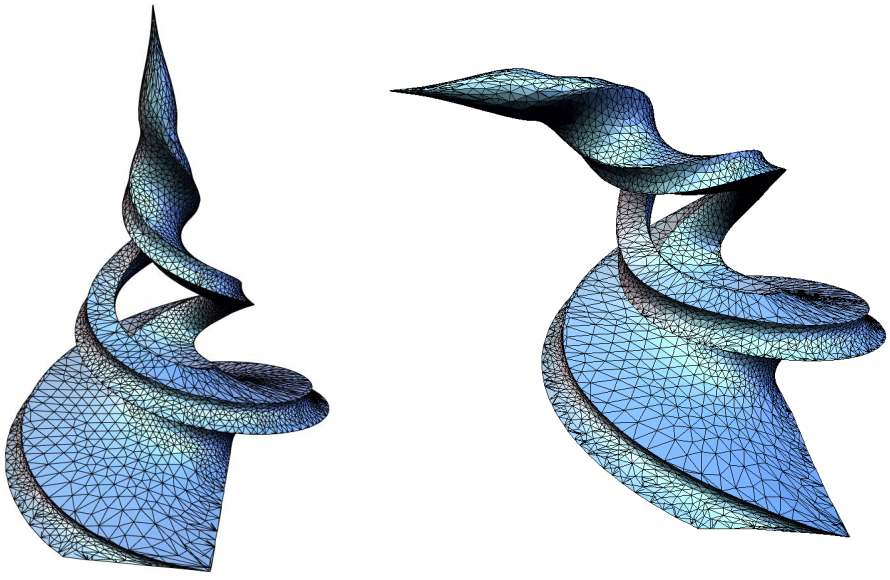


Figure 8.3: *Edit of the twirl object with sharp features.*

We note that in contrast to [LSC$^+$04, SLC$^+$04] L represents the Laplacian matrix without any changes, in particular all constraints are incorporated into the right hand side.

Figure 8.4: *Left: The head of the cow is rotated and scaled. Right: Head and fin of the turtle (top) are bent.*

8.4 Harmonic fields for shape matching and deformation transfer

The main goal of this section is to present an approach for finding correspondence between two surfaces. This correspondence is vital for deformation transfer, a tool for mimicking deformation of the reference on the target shape [ACP03, SP04]. Certainly, meaningful correspondence can only be achieved if there exists a semantic correlation between shapes, like between several models of four-legged animals.

To establish correspondence, the user marks a few pairs of corresponding points on the reference and target surfaces. This interaction establishes the semantic correlation between models and will guide the whole process. Hence, it is essential to carefully select markers so that good coverage of the reference deformation is obtained. At the same time, the number of markers should be kept low to minimize interaction. The usual procedure is a corrective loop, where the user iteratively specifies new markers or updates old ones, and recomputes the deformation transfer until satisfying results are achieved. For this reason, efficiency for finding correspondence and its quality directly account for user-friendly interaction.

The correspondence maps each triangle of the target mesh to one face of the reference. Note that we do not understand correspondence as a bijective mapping and do not solve a cross-parameterization problem [KS04, SAPH04], which in general applies constraint non-linear optimization.

In fact, for deformation transfer a non-bijective mapping is sufficient as stated by Sumner and Popović [SP04] and verified by our experiments (see also figure (8.5)). However, it is vital for the mapping that it respects semantic correspondence, and that it is independent of the tessellation or resolution of the model. Furthermore, it should be established efficiently to allow quick response for fine tuning. Based on such mapping, we decide on match or mismatch between target and reference triangles, for matches local deformations are transferred.

We use harmonic fields to guide the correspondence. Motivated by the notion of shape functions for linear triangles[1] presented in chapter (4), we propose a generalization to manifold meshes. Given a triangle, we can associate with it three functions with values 1 at one vertex and 0 at the two opposite vertices. These shape functions define a set of basis functions over the triangle, well-known as the barycentric coordinates, which can be used for linear interpolation over the triangle. We aspire to establish a generalization of such characterization to manifold meshes, usable for deformation transfer.

Given a set of markers $\{m_1, \ldots, m_k\}$, we associate with it a family of "shape functions" defined over a mesh. For each marker m_i, we define one harmonic field h_i with Dirichlet boundary

[1] Formally, shape functions for triangular Lagrangian finite elements of degree 1.

conditions by setting its value to 1 at m_i and to 0 at all other markers m_j, $i \neq j$. This family of functions defines an k-dimensional vector field over the entire manifold, assigning each vertex a vector $(h_1, \ldots, h_k)$.
We remark that this family is in general not independent and does not form a basis. However, they satisfy the partition of unity property $\sum_{1 \leq i \leq k} h_i \equiv 1$. This stems from the principle of superposition and the fact that there is no dissipation of the harmonic field. This property is of crucial interest for our approach, and we exploit it for finding correspondence.
For two surface meshes with clear semantic correspondence and a set of correspondence markers, we expect that the harmonic fields will be similar and hence correspond. Our experiments verify this observation. Correspondence between triangles is then achieved by matching the vector field of each target triangle (given by the barycentric average) to a triangle of the reference mesh which yields the closest field value. Here, we define closeness in terms of the 2-norm of h. We remark that due to our experiments this can be reduced to taking into account only the n maximal vector components for each value of h, where $3 \leq n \leq k$. This way we transfer local deformations to the target from matching reference triangles. Analog to editing, the solution of the Poisson equation provides the new target mesh, which mimics the deformed reference shape.

8.5 Harmonic interpolation in quaternion domain for editing and deformation transfer

In this section, we show how to interpolate multiple deformations simultaneously, which can be regarded as blending between several sources, each of which propagates a different deformation. We remark that this can be directly applied to the editing setting of section (8.3). Here, we discuss the interpolation in another context: we explore the possibility of direct deformation transfer without establishing any correspondence other than the markers. In other words, local deformations of the reference surface are sampled at the markers. On the target surface, they are used as constraints for harmonic interpolation.
Given correspondence between marked triangles on the reference and target surface, we can compute the deformation on each of these triangles, using a virtual tetrahedron as in [SP04]. The deformations are converted into unit quaternion form. The four components of the quaternion (plus a scaling component if required, see figure (8.4.b)) are considered as separate scalar fields in the manifold. We consider the propagation of the deformations over the surface corresponds as establishing harmonic functions, i.e., we find a solution to the Laplace equation, which satisfies all given deformations independently in every component. Once the solution is obtained, the Poisson scheme is used to apply the deformation.
Despite the simplicity of this approach, it works surprisingly well, it is extremely efficient, and it provides a natural propagation at no extra cost. However, the approach is limited by its nature: First, all local deformations which are obtained from the interpolation are within the convex hull of the given deformations at the markers. The scheme cannot synthesize deformations in between, which have not been captured. Second, the interpolation assumes a smooth variation of deformations relative to the surface domain. It cannot predict and react on strong variations of the target surface which are not reflected in the reference. Hence, if the two surfaces locally differ too much, the interpolated deformations may not be meaningful. Of course, both issues can be addressed by placing additional markers, in particular near geometric detail.

8.6 Results

Figures (8.1), (8.3), (8.4), (8.2), and (8.8) show several edits of models with varying complexity and shape detail. The linear framework enables edits at interactive rates. The Laplacian matrices are decomposed only once after a region of influence is selected, the factors can be reused for efficient back-substitution for reconstruction after every single change of the deformation. The time for decomposition and initial solution when editing the entire Happy Buddha model at a resolution $220K$ triangles is in the order of few seconds on current hardware.
Surface deformation is a very broad topic, and we believe there is no single solution to the variety of editing problems anticipating the needs and desires of modelers and designers. So obviously, the proposed methods within this work are limited and target only some specific problems. While the method is in general robust to triangle foldovers, there are no guarantees on mesh self-contact and penetration.
Figure (8.6) shows the transfer of the lion poses to the cat, correspondence was established using 68 markers for the upper left pair of models. (The correspondence map is illustrated in Figure (8.5).) The models are taken from [SP04], but we reverse their transfer: the reference

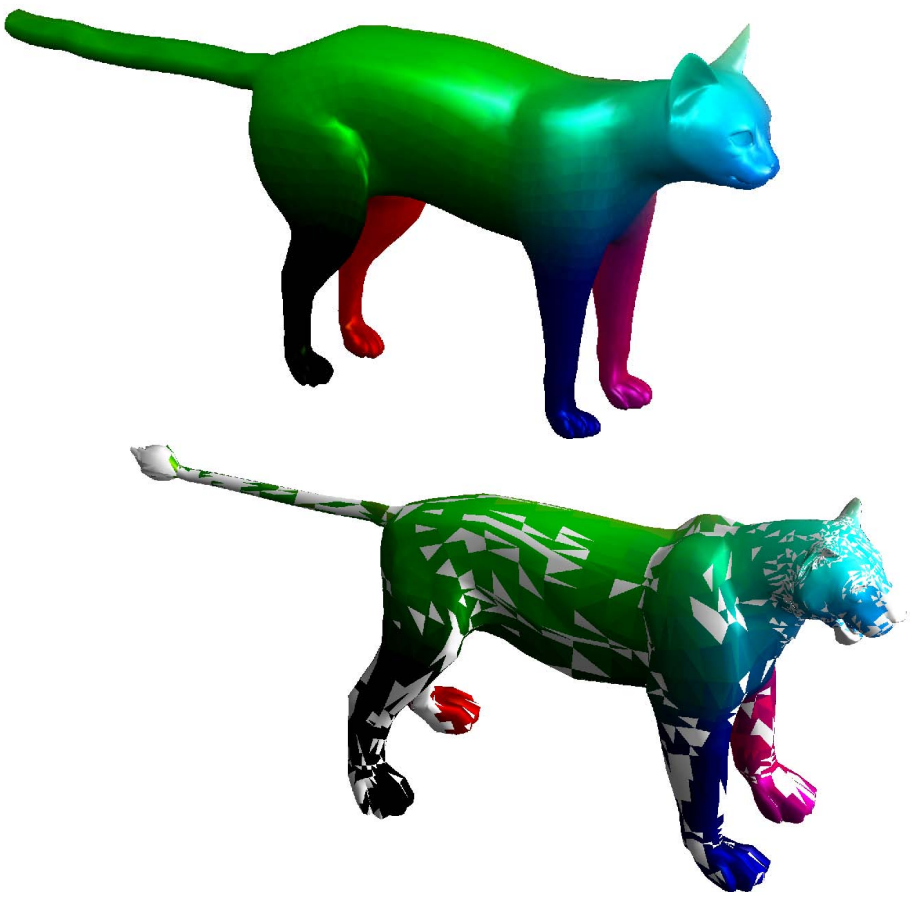


Figure 8.5: *Illustration of a non-bijective correspondence map: The cat as target surface is color coded. For deformation transfer from the lion, the cat triangles are mapped onto the lion, shown in the bottom visualization using the same color code. Indeed the lion is not fully covered, however, the good match in colors indicates a meaningful transfer of deformations, i.e., the left front legs correspond.*

Figure 8.6: *Harmonic guided deformation transfer from the lion onto the cat (top left pair). The models are taken from [SP04], we reverse their transfer (see also section (8.6)).*

deformation sequence was given for the lion, in fact, we are using their output of their algorithm as input. We consider this as a hard but fair test for our correspondence computation, and the ideal result would be to recover the original manually animated cat sequence. The results confirm that our mapping is good. Certainly, artifacts in the input reproduced but rarely amplified. The deformed cat gets close to the original sequence, we observe most problems with the deformation of the tail and inevitable surface self-intersections. We emphasize that the correspondence was established only from the solution of a linear system, in particular, no non-linear optimization was applied.

Figure (8.7) shows a similar test: using 120 markers, the deformation sequence is transferred back onto the original horse model only by harmonic interpolation. For each pair of images, the left one shows the manually animated model (from [SP04]), while the right one shows the original horse deformed according to the result of the interpolation. In an ideal setting, the second one should reproduce the first one. The results show, that this is nearly the case, and it confirms the harmonic interpolation of local deformations in general. Certainly, for the plain harmonic interpolation method more markers are needed for achieving good results.

Figure 8.7: *Testing the harmonic interpolation by transferring the horse animation (from [SP04]) onto the horse (see also section (8.6)).*

Comparing Figure (8.6) and (8.7) and the approaches of sections (8.4) and (8.5), respectively, we observe that for the first approach approximately the same number of markers as for [SP04] is sufficient while we require no additional user parameters and only applications of the linear operator and simple searches over the harmonic coordinates in contrast to non-linear minimization. However, this process is still not quite interactive. Our second approach using harmonic interpolation requires more markers in general. It eliminates matching and hence greatly reduces computational cost, which are independent of the number of markers. In brief, our two approach balance number of markers versus more efficiency and interactiveness.
Discussing limitations, we would like to emphasize that the matching relies heavily on the semantic similarity of the models at hand. In fact the method would not yield convincing results for models with different semantics (e.g., very short legs versus very long legs). The number of markers used in our method is of the same order of magnitude as the ones used in [SP04]. We note that it is possible to establish a matching with less markers especially when dealing with very simple poses, however, the current number of markers is reasonable given that poses of the cat and lion are rather extreme.

Furthermore, a judicious placement of markers in order to better capture deformations can always reduce the number of needed markers. On the other hand for the harmonic interpolation usually more markers are needed as the solution tends to have a slight “rubbery” behavior especially around the joints (see figure (8.7)). This is in part due to the elliptic nature of the interpolation, so more markers are needed around the joints in order to maintain a good approximation of the the deformations. However, in practical applications one might prefer this interactive approach using more markers which can be added and readjusted on the fly yielding immediate response.

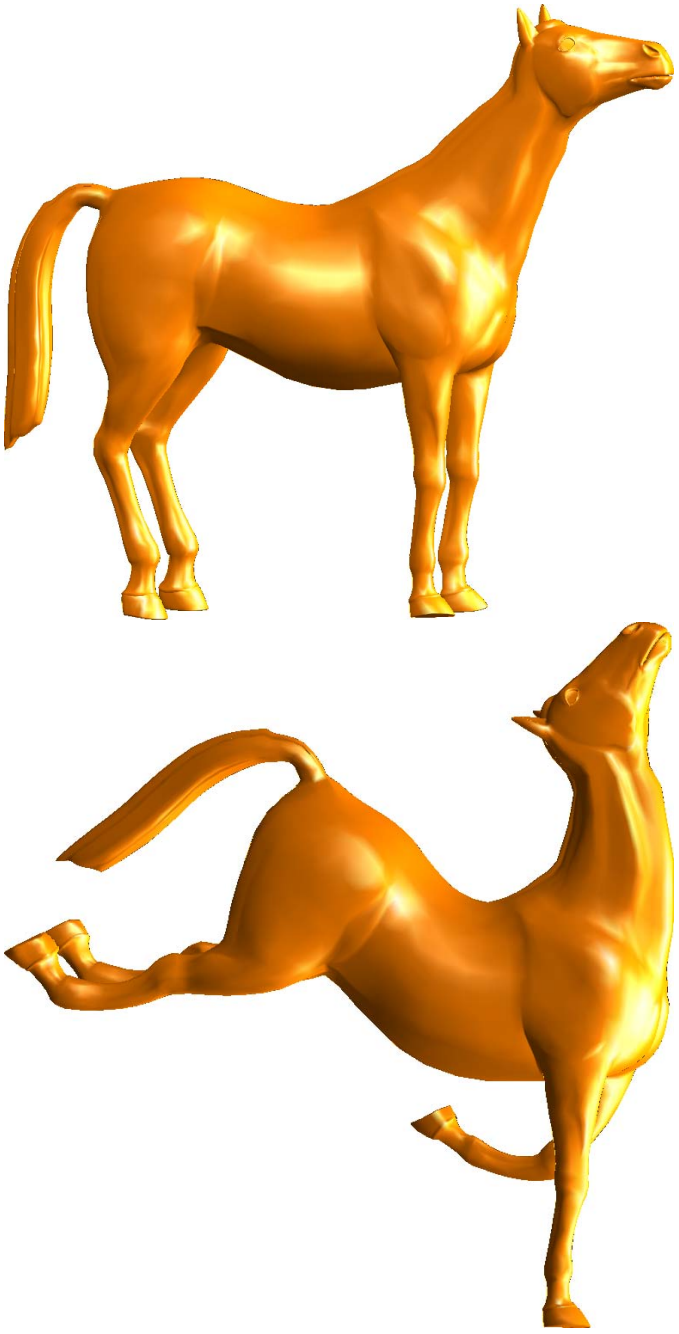


Figure 8.8: *A complex edit of the reference horse model.*

9

Some Remarks on Data Structure and Implementation

In digital geometry processing there exist several data structures for storing mesh connectivity information, see e.g.[CKS98, Ket99]. These data structures, e.g., half-edge, enhance the original mesh and offer possibilities such as circulating around vertices and faces for getting their one-ring neighbors or fans. While these structure are generally suitable for problems which exhibit a combinatorial nature, it turns out that for problems of variational nature this leads to redundant computations especially that in many instances, the ultimate goal is to setup system matrices describing the governing equations. Generally a matrix data structure and a combinatorial geometric structure are used together for handling this kind of problems. Our approach consists of working directly on the original vertex and face lists without constructing any intermediate representations. The key ingredient is to use the sparse matrix structure, e.g. [GMS92] for extracting geometric information as well as performing numerical operations. This way the implementation is generally concise and lends itself to vectorization.
For example, the construction of the connectivity matrix f a mesh (V, T) where V is a nVx3 array of vertex positions and T is nTx3 array describing the triangulation can be as simple as the following matlab code

$$A = sparse(T([1\ 2\ 3],\ :), T([2\ 3\ 1],\ :),\ 1,\ np,\ np). \tag{9.1}$$

The construction of a uniform Laplacian from this matrix is straightforward. Additionally the same idea can be used for constructing different discretizations of the Laplacian by changing the third field of the sparse function to the desired vector valued function, e.g., cotangent weights.
Most of the methods presented in this work have been developed as vectorized Matlab^{TM}code except for a few cases where the highly combinatorial nature of some problems did not allow for an intuitive vectorization and processing with matrices. Directly operating on sparse matrices allows for a transparent transition between geometric processing operations and the setup of matrix equations which usually arise in practical problems.

10

Conclusion

> All solvable problems are trivial, all nontrivial problems are unsolvable.
>
> G. Santayana in [BLM00]

The objective of this work was to develop efficient methods for surface parameterization and editing. In order to simplify implementation and guarantee efficiency, special emphasis was placed on working with sparse matrix data structures. Building upon this platform, several contributions to mesh parameterization and mesh editing have been proposed. This includes linearization of the angle based flattening, introduction of quasi-harmonic maps, curvilinear spherical parameterization and several applications of harmonic fields for mesh editing, shape matching, and deformation transfer. In following, a summary of the key points of these contributions is given.

Angle based mesh parameterization

Conformal mesh parameterization has received extensive attention from the graphics community. The research effort in this direction has been shaped by the strive for increased efficiency for nonlinear methods and improved discretization schemes of differential operators. This work contributes to these efforts first by addressing numerical aspects of one of the most promising but challenging nonlinear problems in mesh parameterization namely, the angle based flattening, and second by surveying some of the recent developments of weighting schemes in the light of development in other disciplines.

For the angle based flattening problem, we presented and discussed several numerical approaches for addressing the arising nonlinear constrained optimization problem. Advantage was taken of the structure of the system matrix in order to decouple the problem making the matrix equations much easier to solve. The use of a modified version of the Uzawa algorithm typically used in saddle point problems allowed for a better performance. Furthermore, close inspection of the problem revealed that it was possible to approximate the nonlinear constraints by linear ones thus yielding for the first time a linear version of the angle based parametrization. The latter can be solved in a straightforward manner based on the normal equation of underdetermined linear systems of equations.

In order to contribute to the understanding of the discretization schemes of the governing equations in discrete conformal maps, an overview of some of the existing methods is given along with some of the parallel developments in other fields. The influence of the discretization on the parameterization is substantiated by a simple but conclusive benchmark.

Quasi-harmonic maps

The main concern of conformal parameterization is the reduction of angular distortion regardless of how distorted other measures such as area or length might be affected. Many applications, however, require a parameterization to have low area distortion. This requirement opens the door to a more difficult problem and also rises the question of how a ”good parameterization” should look like. The contribution of this work is a new low-distortion surface mesh parameterization method based on solving a discrete quasi-harmonic equation. For developing the method, we introduce new linear operators, which capture the metric properties of the surface. The variational nature of the approach ensures the smoothness of the resulting map. The parameterization method is robust and insensitive to possible ill configurations of the initial map. In fact, for most cases the method converges from a simple projection on the least squares plane even for complex models.
As an extension for these fixed boundary harmonic maps, a boundary-free parameterization method consisting of few simple steps was presented. The overall process is controlled using suitable guidance tensor fields reflecting the intrinsic surface geometry. The arising linear systems are symmetric and of similar structure, i.e., the maps can be computed robustly and efficiently.
Our results confirm that despite their simplicity, methods based purely on linear settings have the potential to produce high-quality boundary-free parameterizations, a goal that was only partially met by prior linear methods. We cannot claim the optimality of our results, however, we observe a fair compromise between quality and efficiency and hope that this work will stimulate more research towards pushing the limits of linear parameterization.

Spherical parameterization

For surfaces with a zero genus the natural parameterization domain is the sphere. This change of domain from plane to sphere brings with it more challenges as the parametric coordinates are constrained to lay on the unit sphere. In order to avoid this constraint we presented an efficient approach for solving the spherical parameterization problem based on a curvilinear coordinates representation. The essence of the approach is to look for a solution in the curvilinear coordinate system without requiring the additional spherical constraints usually needed in Cartesian formulations. This setup allows us to take full advantage of some existing techniques originally developed for planar parameterization. Our results substantiate the efficiency of the method and confirm its robustness. Meshes of non-trivial geometry with tens of thousands of triangles are processed in a few seconds, always yielding bijective maps. This computational achievement bridges a so far wide gap in performance between spherical and planar parameterization.

Mesh editing

The planar parameterization of a surface mesh can be regarded as the construction of two scalar fields on the surface which satisfy some quality requirement. Based on the same principle it is possible to construct more general parameterizations by constructing one or many scalar fields on the mesh. These fields can be used as guidance fields for performing different mesh processing tasks on the surface. As proof of concept we presented a surface deformation framework for shape editing and deformation transfer. As a central tool we apply harmonic fields to guide the deformations, i.e., to interpolate of local deformations over the manifold and to establish correspondence for deformation transfer. Our unified approach is elegant and natural: the same

mathematical building blocks, solving the discrete Laplace and Poisson equations, are applied in all stages. Inherent smoothness is provided, and nonlinear minimization of energy functionals is avoided. In fact, the use of linear operators render the approach efficient, the same sparse linear system is solved multiple times per problem instance. Hence, only a single matrix decomposition is required. Our results confirm effectiveness and efficiency of the approach. Harmonic fields offer a practical framework for surface editing and they proved to yield promising results in surface resampling [DKG05]. We see potential use for them in other areas of computer graphics such as mesh segmentation and motion planning.

Future research prospects

As a follow up to the presented work, the following research venues are considered:
Most of the parameterization machinery developed in this thesis is geared towards solving general, but fundamental, aspects of the problem. An interesting direction for future work is to tune the proposed methods for specific application purposes. For instance, texture mapping remains one of the most relevant applications of mesh parameterization, a natural extension would be to couple image processing techniques with geometric methods for addressing problems such as seams along patch boundaries and hard positional constraints in the texture domain. Another direction is the automatic update of parameterization for dynamic models. As there is increasing interest in dynamic scenes, it is desirable to tailor efficient algorithms for handling the additional temporal dimension. The challenge here is to cope with the changes in the model geometry over time and update the parameterization coordinate without solving the whole parameterization problem for every frame.
In depth investigation of the potential of sparse matrices data structure in geometry processing operations is an other interesting research direction. While many operations such as smoothing, deformation, and parameterization can readily take advantage of this framework, devising matrix operations capable of describing more combinatorial applications such as, for instance, mesh simplification remains an interesting and challenging topic.
In the realm of mesh deformation, most of the existing work relies on simple discretizations which generally fall short from accurately describing deformation behavior or incorporating mechanical material properties in the model description. An interesting venue of future research is to investigate the potential of more elaborate discretizations for editing surface meshes. For instance building upon shell theory for describing the deformation of surface meshes and developing intuitive tools for fulfilling designers needs received so far only little interest from the community and exploring this direction would fill the gap between engineering simulation and mesh editing.

Bibliography

[ACL00] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH*, pages 157–164, 2000.

[ACP03] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics*, 22(3):587–594, 2003.

[AHU58] K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, Stanford, Calif., 1958.

[Ale00] M. Alexa. Merging polyhedral shapes with scattered features. In *The Visual Computer*, volume 16(1), pages 26–37. Springer, 2000.

[Ale03] M. Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2):105–114, 2003.

[AN02] G. Alessandrini and V. Nesi. Area formulas for σ-harmonic mappings. In *Nonlinear problems in mathematical physics and related topics, I*, volume 1 of *Int. Math. Ser. (N. Y.)*, pages 1–21. Kluwer/Plenum, New York, 2002.

[Arf85] G. Arfken. *Mathematical methods for physicists*. Academic Press, 3rd edition, 1985.

[Bab94] I. Babuška. Courant element: before and after. In *Finite element methods (Jyväskylä, 1993)*, volume 164 of *Lecture Notes in Pure and Appl. Math.*, pages 37–51. Dekker, New York, 1994.

[Bar84] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics (Proc. SIGGRAPH)*, 18(3):21–30, 1984.

[BBK05] M. Botsch, D. Bommes, and L. Kobbelt. Efficient linear system solvers for mesh processing. In *11th IMA Conference on the Mathematics of Surfaces (preprint)*, 2005.

[Ber82] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press, New York, 1982.

[BGK92] Ch. Brechbühler, G. Gerig, and O. Kübler. Towards representation of 3D shape: global surface parametrization. In *IAPR: Proceedings of the international workshop on Visual form: analysis and recognition*, pages 79–88, 1992.

[BK03] G. H. Bendels and R. Klein. Mesh forging: editing of 3D-meshes using implicitly defined occluders. In *Proc. Symposium on Geometry Processing*, pages 207–217, 2003.

[BK04] M. Botsch and L. Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634, 2004.

[BLM00] Ted Belytschko, Wing Kam Liu, and Brian Moran. *Nonlinear finite elements for continua and structures*. John Wiley & Sons Ltd., Chichester, 2000.

[BN98] R. H. Byrd and J. Nocedal. Active set and interior methods for nonlinear optimization. In *Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998)*, pages 667–676 (electronic), 1998.

[Bra01] D. Braess. *Finite elements*. Cambridge University Press, Cambridge, second edition, 2001. Theory, fast solvers, and applications in solid mechanics, Translated from the 1992 German edition by Larry L. Schumaker.

[BV96] G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM Journal on Discrete Mathematics*, 9(3):349–359, 1996.

[BW80] C.A. Brebbia and S. Walker. *Boundary element techniques in engineering*. Newnes-Butterworths, London, 1980.

[CKS98] Swen Campagna, Leif Kobbelt, and Hans-Peter Seidel. Directed edges–a scalable representation for triangle meshes. *J. Graph. Tools*, 3(4):1–11, 1998.

[Coq90] S. Coquillart. Extended free-form deformation: a sculpturing tool for 3D geometric modeling. In *Proc. SIGGRAPH*, pages 187–196, 1990.

[Cou37] R. Courant. Plateau's problem and Dirichlet's principle. *Ann. of Math. (2)*, 38(3):679–724, 1937.

[Cou43] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49:1–23, 1943.

[Cou50] R. Courant. *Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces*. Interscience Publishers, Inc., New York, N.Y., 1950.

[DKG05] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.*, 22(5):392–423, 2005.

[DMA02] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum (Proc. Eurographics)*, 21(3):209–218, 2002.

[DMK03] P. Degener, J. Meseth, and R. Klein. An adaptable surface parameterization method. *Proc. 9th International Meshing Roundtable*, pages 201–213, 2003.

[Dou31] J. Douglas. Solution of the problem of Plateau. *Trans. Amer. Math. Soc.*, 33(1):263–321, 1931.

[EDD+95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics (Proc. SIGGRAPH)*, 29:173–182, 1995.

[FB88] D. Forsey and R. Bartels. Hierarchical B-spline refinement. In *Proc. SIGGRAPH*, pages 205–212, 1988.

[FH05] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. Springer, 2005.

[Flo97] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[Flo03] M. S. Floater. Mean value coordinates. *Computer Aided Geom. Design*, 20(1):19–27, 2003.

[FSD05] I. Friedel, P. Schröder, and M. Desbrun. Unconstrained spherical parameterization. In *SIGGRAPH technical sketch*, 2005.

[Gar98] A. Garg. New results on drawing angle graphs. *Comput. Geom.*, 9(1-2):43–82, 1998.

[GGS03] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3D meshes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):358–363, 2003.

[GI04] J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004.

[Glo84] R. Glowinski. *Numerical methods for nonlinear variational problems*. Springer Series in Computational Physics. Springer-Verlag, New York, 1984.

[GMS92] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.

[GSS99] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proc. SIGGRAPH*, pages 325–334, 1999.

[Gus02] I. Guskov. An anisotropic mesh parameterization scheme. In *IMR*, pages 325–332, 2002.

[GWC+04] X. Gu, Y. Wang, T.F. Chan, P.M. Thompson, and S-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging*, 23(8):949–958, 2004.

[GY02] X. Gu and S.-T. Yau. Computing conformal structures of surfaces. *Communications in Information and Systems*, 2:121–146, 2002.

[GY03] X. Gu and S.-T. Yau. Global conformal parameterization. In *Proc. Symposium on Geometry Processing*, pages 127–137, 2003.

[HAT+00] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):181–189, 2000.

[HC89] D. G. Holmes and S. D. Connell. Solution of the 2D navierstokes equations on unstructured adaptive grids. In *Proc. AIAA 9th CFD Conference*, 1989.

[Hei87] B. Heinrich. *Finite difference methods on irregular networks*, volume 82 of *Internationale Schriftenreihe zur Numerischen Mathematik [International Series of Numerical Mathematics]*. Birkhäuser Verlag, Basel, 1987.

[HG00] K. Hormann and G. Greiner. MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, pages 153–162. Vanderbilt University Press, 2000.

[HGC99] K. Hormann, G. Greiner, and S. Campagna. Hierarchical parametrization of triangulated surfaces. In *Proceedings of Vision, Modeling, and Visualization*, pages 219–226, 1999.

[HS52] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436, 1952.

[Hui91] G. Huiskamp. Difference formulas for the surface Laplacian on a triangulated surface. *J. Comput. Phys.*, 95(2):477–496, 1991.

[IS01] T. Iwaniec and C. Sbordone. Quasiharmonic fields. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 18(5):519–572, 2001.

[Kan05] T. Kanai. Parametric curves on meshes. In *GRAPHITE '05: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pages 413–416, 2005.

[KCVS98] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Proc. SIGGRAPH*, pages 105–114, 1998.

[Ket99] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Comput. Geom. Theory Appl.*, 13(1):65–90, 1999.

[KGG05] Z. Karni, C. Gotsman, and S. J. Gortler. Free-boundary linear parameterization of 3d meshes in the presence of constraints. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 268–277, 2005.

[Kre91] E. Kreyszig. *Differential geometry*. Dover Publications, New York, 1991. Reprint of the 1963 edition.

[KS04] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3D models. *ACM Transactions on Graphics*, 23(3):861–869, 2004.

[KVLS99] L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. *Computer Graphics Forum (Proc. Eurographics)*, 18(3):119–130, 1999.

[LdSS+01] J. Liesen, E. de Sturler, A. Sheffer, Y. Aydin, and C. M. Siefert. Preconditioners for indefinite linear systems arising in surface parameterization. In *Proc. 10th Int. Meshing Roundtable*, pages 71–81. Sandia Nat. Lab., Oct 2001.

[Lev80] B.W. Levinger. The square root of a 2×2 matrix. *Mathematics Magazine*, 53(4):222–224, 1980.

[LKG+03] I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw. Twister: A space-warp operator for the two-handed editing of 3D shapes. *ACM Transactions on Graphics*, 22(3):663–668, 2003.

[LKL02] Y. Lee, H. Seok Kim, and S. Lee. Mesh parameterization with a virtual boundary. *Computers & Graphics*, 26(5):677–686, 2002.

[LPRM02] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21, 3:362–371, 2002.

[LSC+04] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *Proc. Shape Modeling International*, pages 181–190, 2004.

[LSS+98] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. *Computer Graphics (Proc. SIGGRAPH)*, 32:95–104, 1998.

[Lue69] David G. Luenberger. *Optimization by vector-space methods*. Wiley-Interscience, 1969.

[Mac49] R. H. MacNeal. *The solution of partial differential equations by means of electrical networks*. PhD thesis, California Institute of Technology, January 1949.

[Mac53] R. H. MacNeal. An asymmetrical finite difference network. *Quart. Math. Appl.*, 11:295–310, 1953.

[Ode87] T. Oden. Some historic comments on finite elements. In *Proceedings of the ACM Conference on History of Scientific and Numeric Computation*, pages 125–130, New York, NY, USA, 1987. ACM Press.

[PB00] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *Proc. SIGGRAPH*, pages 471–478, 2000.

[PGB03] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[PH03] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):340–349, 2003.

[PP93] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experiment. Math.*, 2(1):15–36, 1993.

[PS75] C. C. Paige and M. A. Saunders. Solutions of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.

[QBH+00] M. Quicken, Ch. Brechbühler, J. Hug, H. Blattmann, and G. Székely. Parameterization of closed surfaces for parametric surface description. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2000*, volume 1, pages 354–360, June 2000.

[Rad30] T. Radó. On Plateau's problem. *Ann. of Math. (2)*, 31(3):457–469, 1930.

[Riv84] M.-C. Rivara. Design and data structure of fully adaptive, multigrid, finite-element software. *ACM Trans. Math. Softw.*, 10(3):242–264, 1984.

[SAPH04] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *ACM Transactions on Graphics*, 23(3):870–877, 2004.

[SCGL02] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. In *Proc. IEEE Visualization*, pages 355–362, 2002.

[Sch52] K. Schellbach. Probleme der Variationsrechnung. *J. Reine Angew. Math.*, 41:293–363, 1852.

[SdS00] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers*, 17(3):326–337, 2000.

[SdS02] A. Sheffer and E. de Sturler. Smoothing an overlay grid to minimize linear distortion in texture mapping. *ACM Transactions on Graphics*, 21(4):874–890, 2002.

[SGD03] A. Sheffer, C. Gotsman, and N. Dyn. Robust spherical parameterization of triangular meshes. In *Proc. of* 4^{th} *Israel-Korea Binational Workshop on Geometric Modeling and Computer Graphics*, pages 94–99, 2003.

[SH02] A. Sheffer and J. Hart. Seamster: Inconspicuous low-distortion texture seam layout. In *Proceedings of IEEE Visualization*, 2002.

[Sib81] R. Sibson. A brief description of natural neighbour interpolation. In V. Barnet, editor, *Interpreting Multivariate Data*, pages 21–36. Wiley, Chichester, 1981.

[SLC+04] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proc. Symposium on Geometry Processing*, pages 179–188, 2004.

[SLMB05] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++: fast and robust angle based flattening. *ACM Trans. Graph.*, 24(2):311–330, 2005.

[SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proc. SIGGRAPH*, pages 151–160, 1986.

[SP04] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, 2004.

[SSGH01] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Computer Graphics (Proc. SIGGRAPH)*, pages 409–416, 2001.

[SV03] S. M. Shontz and S. A. Vavasis. A mesh warping algorithm based on weighted laplacian smoothing. In *IMR*, pages 147–158, 2003.

[SYGS05] S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. Practical spherical embedding of manifold triangle meshes. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 258–267, 2005.

[Tho01] V. Thomée. From finite differences to finite elements a short history of numerical analysis of partial differential equations. *J. Comput. Appl. Math.*, 128(1-2):1–54, 2001.

[Wac71] E. L. Wachspress. A Rational Basis for Function Approximation. *IMA J. Appl. Math.*, 8(1):57–68, 1971.

[Win66] A. M. Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

[YBS04] S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Proc. Shape Modeling International*, pages 200–208, 2004.

[YZX+04] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 23(3):644–651, 2004.

[ZCKS05] R. Zayer, Rössl C., Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. In Marc Alexa and Joe Marks, editors, *The European Association for Computer Graphics 26th Annual Conference : EUROGRAPHICS 2005*, volume 24 of *Computer Graphics Forum*, pages 601–609, Dublin, Ireland, 2005. Eurographics, Blackwell.

[Zie71] O. C. Zienkiewicz. *The finite element method in engineering science*. McGraw-Hill, London, 1971.

[ZMT05] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics*, 24(1):1–27, 2005.

[ZRS03] R. Zayer, C. Rössl, and H.-P. Seidel. Convex boundary angle based flattening. In Thomas Ertl, Bernd Girod, Günther Greiner, Heinrich Niemann, Hans-Peter Seidel, Eckehard Steinbach, and Rüdiger Westermann, editors, *Vision, Modeling and Visualization 2003 (VMV-03) : proceedings*, Vision, Modeling, and Visualization 2003., pages 281–288, Munich, Germany, November 2003. Akademische Verlagsgesellschaft Aka.

[ZRS04a] R. Zayer, C. Rössl, and H.-P. Seidel. Efficient iterative solvers for angle based flattening. In Bernd Girod, Marcus Magnor, and Hans-Peter Seidel, editors, *Vision, modeling, and visualization 2004 (VMV-04)*, pages 347–354, Stanford, USA, 2004. Akademische Verlagsgesellschaft Aka.

[ZRS04b] R. Zayer, C. Rössl, and H.-P. Seidel. Variations of angle based flattening. In Neil A. Dodgson, Michael S. Floater, and Malcom A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 187–199. Springer, Berlin, Germany, 2004.

[ZRS05a] R. Zayer, C. Rössl, and H-P. Seidel. Discrete tensorial quasi-harmonic maps. In *Shape Modeling International 2005 (SMI 2005) : proceedings*, pages 276–285, Cambridge, MA, U.S.A, 2005.

[ZRS05b] R. Zayer, C. Rössl, and H-P. Seidel. Setting the boundary free: A composite approach to surface parameterization. In *Symposium on Geometry Processing*, pages 101–110, Vienna, Austria, 2005. Eurographics/ACM.

[ZRS06] R. Zayer, C. Rössl, and H.-P. Seidel. Curvilinear spherical parameterization. In *Shape Modeling International (SMI)*, pages 57–64, Matsushima, Japan, 2006. IEEE.

[ZSS97] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Proc. SIGGRAPH*, pages 259–268, 1997.

A

Publications

[1] Rhaleb Zayer, Bruno Lévy, Hans-Peter Seidel. *Linear angle based parameterization.* Symposium on Geometry Processing 2007, 135-141

[2] Edilson de Aguiar, Rhaleb Zayer, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. *Video-driven animation of human body scans.* IEEE 3DTV Conference, Kos Island, Greece, 2007

[3] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Curvilinear Spherical Parameterization.* In: Shape Modeling International (SMI), Matsushima, Japan, IEEE, Los Alamitos, 2006, 57-64.

[4] Rhaleb Zayer, Christian Rössl, Zachi Karni, and Hans-Peter Seidel. *Harmonic Guidance for Surface Deformation.* In: The European Association for Computer Graphics 26th Annual Conference : EUROGRAPHICS 2005, Dublin, Ireland, Blackwell, Oxford, 2005, 601-609.

[5] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Setting the Boundary Free: A Composite Approach to Surface Parameterization.* In: Symposium on Geometry Processing, Vienna, Austria, Eurographics/ACM, Aire-la-Ville, 2005, 91-100.

[6] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Discrete Tensorial Quasi-Harmonic Maps.* In: Shape Modeling International 2005 (SMI 2005), Cambridge, MA, U.S.A, IEEE, Los Alamitos, 2005, 276-285.

[7] Ioannis Ivrissimtzis, Rhaleb Zayer, and Hans-Peter Seidel. *Polygonal decompositions of quadrilateral subdivision meshes.* Computer Graphics & Geometry 7 (1): 16-30, 2005.

[8] Hitoshi Yamauchi, Stefan Gumhold, Rhaleb Zayer and, Hans-Peter Seidel. *Mesh Segmentation Driven by Gaussian Curvature.* The Visual Computer 21 (8-10): 649-658, 2005.

[9] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Efficient Iterative Solvers for Angle Based Flattening.* In: Vision, modeling, and visualization 2004 (VMV-04), Stanford, USA, Akademische Verlagsgesellschaft Aka, Berlin, 2004, 347-354.

[10] Holger Theisel, Christian Rössl, Rhaleb Zayer and Hans-Peter Seidel. *Normal Based Estimation of the Curvature Tensor for Triangular Meshes.* In: 12th Pacific Conference on Computer Graphics and Applications, PG 2004, Seoul, South Korea, IEEE, Los Alamitos, 2004, 288-297.

[11] Ioannis Ivrissimtzis, Rhaleb Zayer and Hans-Peter Seidel. *Polygonal decomposition of the 1-ring neighborhood of the Catmull-Clark scheme.* In: Shape Modeling International 2004 (SMI 2004), Genoa, Italy, IEEE, Los Alamitos, 2004, 101-109.

[12] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Variations of angle based flattening*. In: Advances in Multiresolution for Geometric Modelling, Springer, Berlin, 2004, 187-199.

[13] Rhaleb Zayer, Christian Rössl, and Hans-Peter Seidel. *Convex Boundary Angle Based Flattening*. In: Vision, Modeling and Visualization 2003 (VMV-03) : proceedings, Munich, Germany, November, 19 - November, 21, Akademische Verlagsgesellschaft Aka, Berlin, 2003, 281-288