SIGGRAPH Asia 2008 Course Notes

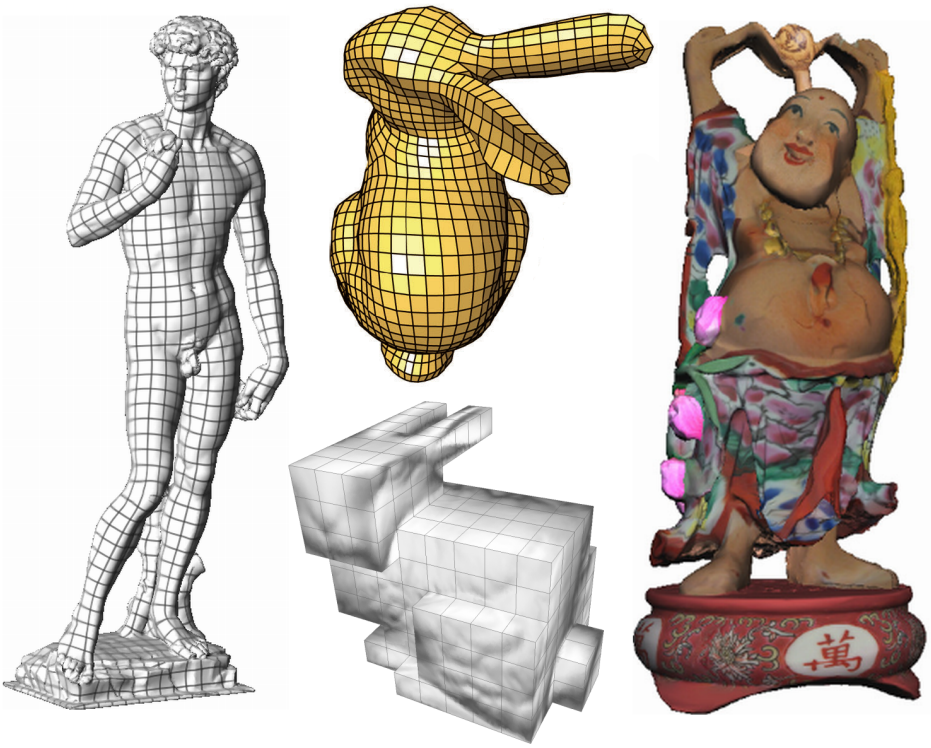
Mesh Parameterization: Theory and Practice

Kai Hormann
Clausthal University of Technology

Konrad Polthier
Freie Universität Berlin

Alla Sheffer
University of British Columbia

December 2008



Summary

Mesh parameterization is a powerful geometry processing tool with numerous computer graphics applications, from texture mapping to animation transfer. This course outlines its mathematical foundations, describes recent methods for parameterizing meshes over various domains, discusses emerging tools like global parameterization and inter-surface mapping, and demonstrates a variety of parameterization applications.

Prerequisites

The audience should have had some prior exposure to mesh representation of geometric models and a working knowledge of vector calculus, elementary linear algebra, and the fundamentals of computer graphics. Optional pre-requisites: some lectures may also assume some familiarity with differential geometry and graph theory.

Intended Audience

Graduate students, researchers, and application developers who seek to understand the concepts and technologies used in mesh parameterization and wish to utilize them. Listeners get an overview of the spectrum of processing applications that benefit from parameterization and learn how to evaluate different methods in terms of specific application requirements.

Sources

These course notes include material of two recent surveys on mesh parameterization co-authored by the course organizers:

- 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.

  This survey addresses the more theoretical aspects of mesh parameterization and focuses on planar parameterizations.

- A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, 2006.

  This survey focuses on applications as well as non-planar parameter domains.

The course notes from our ACM SIGGRAPH 2007 course as well as additional material can be found at the course website:

`http://www.in.tu-clausthal.de/~hormann/parameterization/index.html`

Speakers

Kai Hormann, Clausthal University of Technology, Germany

Kai Hormann is an assistant professor for computer graphics in the Department of Informatics at Clausthal University of Technology in Germany. His research interests are focussed on the mathematical foundations of geometry processing algorithms as well as their applications in computer graphics and related fields. Dr. Hormann has co-authored several papers on parameterization methods, surface reconstruction, and barycentric coordinates. Kai Hormann received his PhD from the University of Erlangen in 2002 and spent two years as a postdoctoral research fellow at the Multi-Res Modeling Group at Caltech, Pasadena and the CNR Institute of Information Science and Technologies in Pisa, Italy.

Konrad Polthier, Freie Universität Berlin, Germany

Konrad Polthier is full professor of mathematics at Freie Universität Berlin and the DFG research center MATHEON, and member of the Berlin Mathematical School. Konrad Polthier received his PhD from University of Bonn in 1994, and headed research groups at Technische Universität Berlin and Zuse-Institute Berlin. His current research focuses on discrete differential geometry and mathematical problems in geometry processing applications. Dr. Polthier co-edited several books on mathematical visualization, and co-produces mathematical video films. His recent video MESH (www.mesh-film.de, joint with Beau Janzen, Los Angeles) has received international awards including "Best Animation" at the New York International Independent Film Festival. Polthier served as paper co-chair on international conferences including ACM/Eurographics Symposium on Geometry Processing 2006.

Alla Sheffer, University of British Columbia, Canada

Alla Sheffer is an associate professor in the Computer Science department at the University of British Columbia. Dr. Sheffer investigates algorithmic aspects of digital geometry processing, focusing on several fundamental problems of mesh manipulation and editing. Her recent research addresses algorithms for mesh parameterization, processing of developable surfaces, mesh editing, reconstruction, and shape analysis. Her work on these topics had been published at top venues, including Siggraph, Eurographics, and the Symposium on Geometry Processing. She co-authored several parameterization methods, including ABF/ABF++, which are used in popular 3D modelers including Blender, Maya, and Catia. Alla Sheffer received her PhD from the Hebrew University of Jerusalem in 1999. Prior to moving to UBC in 2003, she was a postdoc at the University of Illinois at Urbana-Champaign and an assistant professor at Technion, Israel.

Syllabus

First session

- Introduction [10 min, Alla]
- Barycentric Mappings [25 min, Kai]
- Differential Geometry Primer [25 min, Kai]
- Non-Linear Methods [30 min, Alla]
- Comparison and Applications of Planar Methods [15 min, Kai]

Second session

- Non-Planar Domains [15 min, Kai]
- Cross-Parameterization and Constraints [40 min, Alla]
- Global Parameterizations and Cone Points [45 min, Konrad]
- Open Problems and Q/A [5 min, all]

Contents

Chapter 1

Introduction

For any two surfaces with similar topology, there exists a bijective mapping between them. If one of these surfaces is a triangular mesh, the problem of computing such a mapping is referred to as mesh parameterization. The surface that the mesh is mapped to is typically called the parameter domain. Parameterization was introduced to computer graphics for mapping textures onto surfaces. Over the last decade, it has gradually become a ubiquitous tool for many mesh processing applications, including detail-mapping, detail-transfer, morphing, mesh-editing, mesh-completion, remeshing, compression, surface-fitting, and shape-analysis. In parallel to the increased interest in applying parameterization, various methods were developed for different kinds of parameter domains and parameterization properties.

The goal of this course is to familiarize the audience with the theoretical and practical aspects of mesh parameterization. We aim to provide the skills needed to implement or improve existing methods, to investigate new approaches, and to critically evaluate the suitability of the techniques for a particular application.

The course starts with an introduction to the general concept of parameterization and an overview of its applications. The first half of the course then focuses on planar parameterizations while the second addresses more recent approaches for alternative domains. The course covers the mathematical background, including intuitive explanations of parameterization properties like bijectivity, conformality, stretch, and area-preservation. The state-of-the-art is reviewed by explaining the main ideas of several approaches, summarizing their properties, and illustrating them using live demos. We conclude by presenting a list of open research problems and potential applications that can benefit from parameterization.

1.1 Applications

Surface parameterization was introduced to computer graphics as a method for mapping textures onto surfaces [Bennis et al., 1991; Maillot et al., 1993]. Over the last decade, it has gradually become a ubiquitous tool, useful for many mesh processing applications, discussed below.

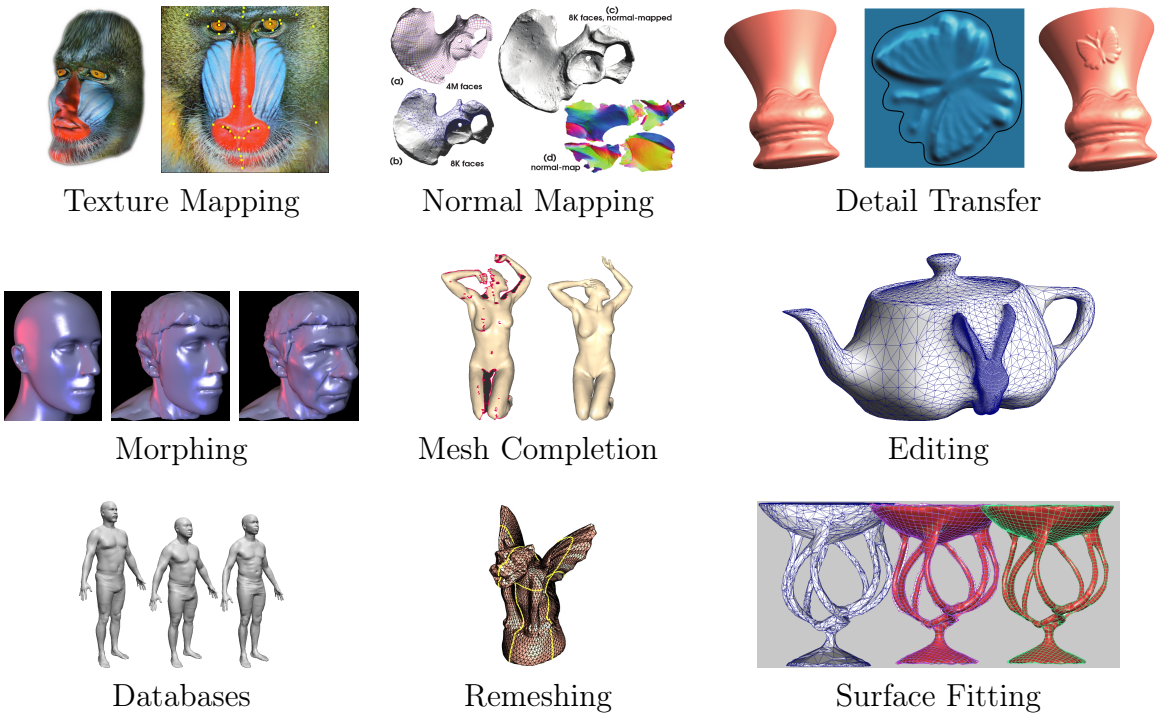


Figure 1.1: Parameterization Applications.

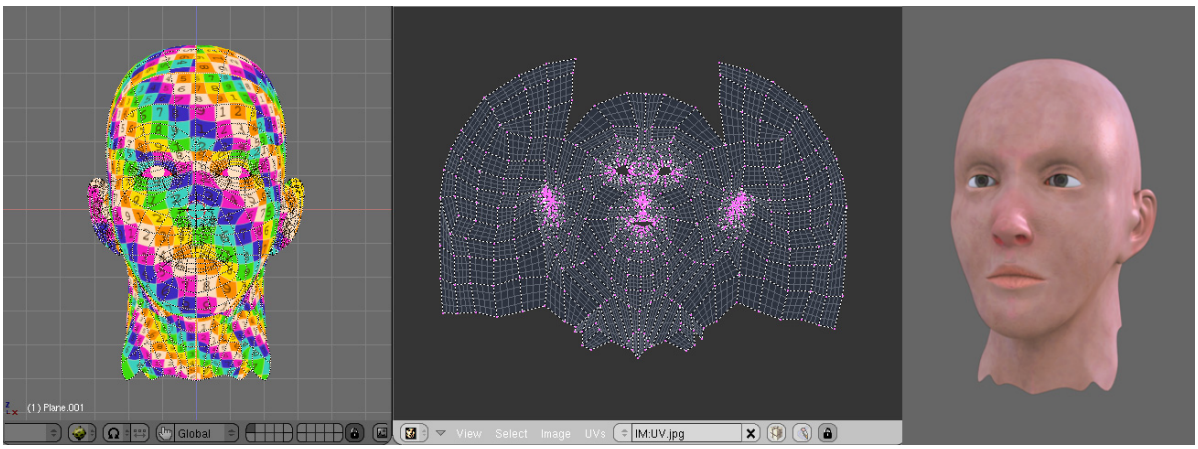


Figure 1.2: Application of parameterization: texture mapping (Least Squares Conformal Maps implemented in the Open-Source Blender modeler).

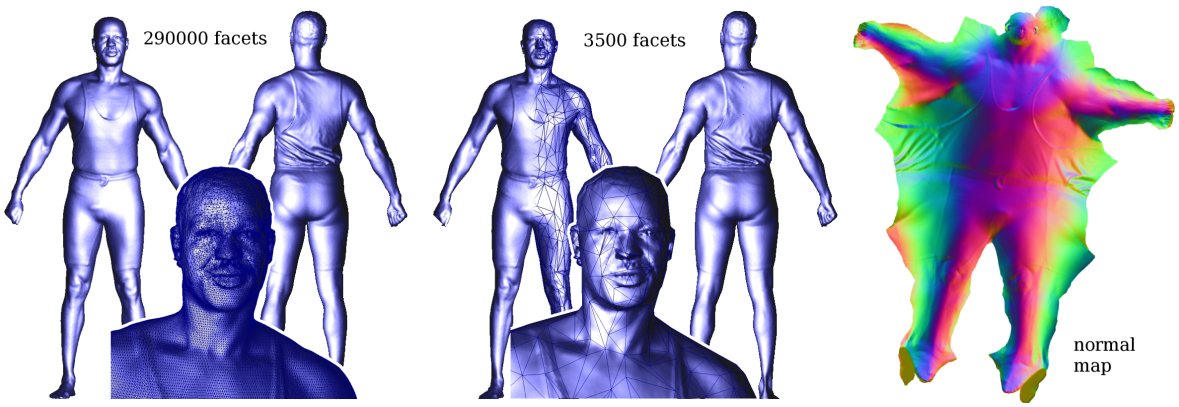


Figure 1.3: Application of parameterization: appearance-preserving simplification. All the details are encoded in a normal map, applied onto a dramatically simplified version of the model (1.5% of the original size).

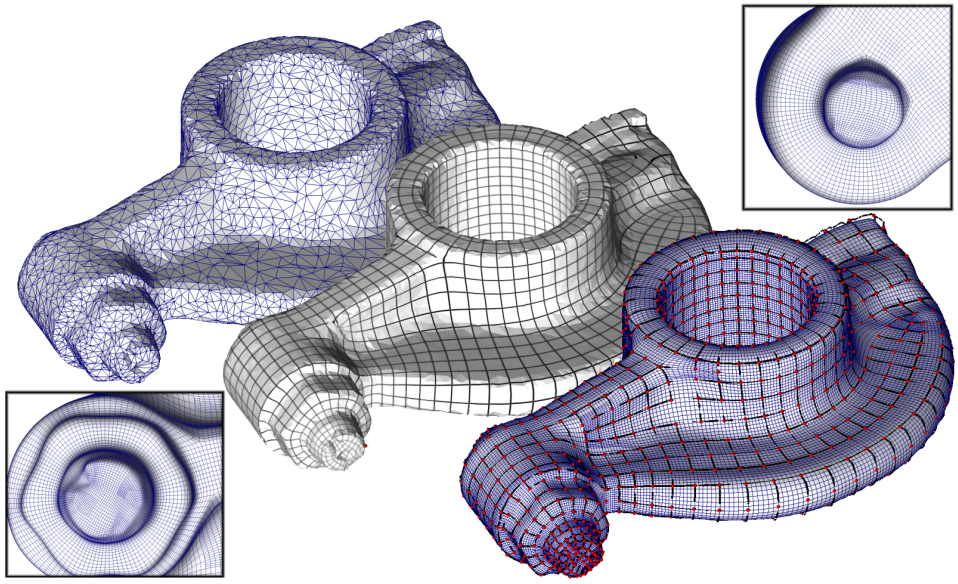


Figure 1.4: A global parameterization realizes an abstraction of the initial geometry. This abstraction can then be re-instanciated into alternative shape representations.

Detail Mapping

Detailed objects can be efficiently represented by a coarse geometric shape (polygonal mesh or subdivision surface) with the details corresponding to each triangle stored in a separate 2D array. In traditional texture mapping the details are the colors of the respective pixels. Models can be further enriched by storing bump, normal, or displacement maps. Recent techniques [Peng et al., 2004; Porumbescu et al., 2005] model a thick region of space in the neighborhood of the surface by using a volumetric texture, rather than a 2D one. Such techniques are needed in order to model detail with complicated topology or detail that cannot be easily approximated locally by a height field, such as sparsely interwoven structures or animal fur. The natural way to map details to surfaces is using planar parameterization.

Detail Synthesis

While the goal of texture mapping is to represent the complicated appearance of 3D objects, several methods make use of mesh parameterization to create the local detail necessary for a rich appearance. Such techniques can use as input flat patches with sample detail, e.g. [Soler et al., 2002]; parametric or procedural models; or direct user input and editing [Carr and Hart, 2004]. The type of detail can be quite varied and the intermediate representations used to create it parallel the final representations used to store it.

Morphing and Detail Transfer

A map between the surfaces of two objects allows the transfer of detail from one object to another (e.g. [Praun et al., 2001]), or the interpolation between the shape and appearance of several objects [Alexa, 2000; Kraevoy and Sheffer, 2004; Schreiner et al., 2004]. By varying the interpolation ratios over time, one can produce morphing animations. In spatially-varying and frequency-varying morphs, the rate of change can be different for different parts of the objects, or different frequency bands (coarseness of the features being transformed) [Allen et al., 2003; Kraevoy and Sheffer, 2004]. Such a map can either be computed directly or, as more commonly done, computed by mapping both object surfaces to a common domain. In addition to transferring the static appearance of surfaces, inter-surface parameterizations allow the transfer of animation data between shapes, either by transferring the local surface influence from bones of an animation rig, or by directly transferring the local affine transformation of each triangle in the mesh [Sumner and Popović, 2004].

Mesh Completion

Meshes from range scans often contain holes and multiple components. Lévy [2003] uses planar parameterization to obtain the natural shape for hole boundaries and to triangulate those. In many cases, prior knowledge about the overall shape of the scanned models exists. For instance, for human scans, templates of a generic human shape are

readily available. Allen et al. [2003] and Anguelov et al. [2005] use this prior knowledge to facilitate completion of scans by computing a mapping between the scan and a template human model. Kraevoy and Sheffer [2005] develop a more generic and robust template-based approach for completion of any type of scans. The techniques typically use an inter-surface parameterization between the template and the scan.

Mesh Editing

Editing operations often benefit from a local parameterization between pairs of models. Biermann et al. [2002] use local parameterization to facilitate cut-and-paste transfer of details between models. They locally parameterize the regions of interest on the two models in 2D and overlap the two parameterizations. They use the parameterization to transfer shape properties from one model to the other. Lévy [2003] uses local parameterization for mesh composition in a similar manner. They compute an overlapping planar parameterization of the regions near the composition boundary on the input models and use it to extract and smoothly blend shape information from the two models.

Creation of Object Databases

Once a large number of models are parameterized on a common domain one can perform an analysis determining the common factors between objects and their distinguishing traits. For example on a database of human shapes [Allen et al., 2003] the distinguishing traits may be gender, height, and weight. Objects can be compared against the database and scored against each of these dimensions, and the database can be used to create new plausible object instances by interpolation or extrapolation of existing ones.

Remeshing

There are many possible triangulations that represent the same shape with similar levels of accuracy. Some triangulation may be more desirable than others for different applications. For example, for numerical simulations on surfaces, triangles with a good aspect ratio (that are not too small or too skinny are important for convergence and numerical accuracy. One common way to remesh surfaces, or to replace one triangulation by another, is to parameterize the surface, then map a desirable, well-understood, and easy to create triangulation of the domain back to the original surface. For example, Gu et al. [2002] use a regular grid sampling of a planar square domain, while other methods, e.g. [Guskov et al., 2000] use regular subdivision (usually 1-to-4 triangle splits) on the faces of a simplicial domain. Such locally regular meshes can usually support the creation of smooth surfaces as the limit process of applying subdivision rules. To generate high quality triangulations, Desbrun et al. [2002] parameterize the input mesh in the plane and then use planar Delaunay triangulation to obtain a high quality remeshing of the surface. One problem these methods face is the appearance of visible discontinuities along the cuts created to facilitate the parameterization. Surazhsky and Gotsman [2003] avoid global parameterization, and instead use local parameterization to move vertices

along the mesh as part of an explicit remeshing scheme. Recent methods such as [Ray et al., 2006] use global parameterization to generate a predominantly quadrilateral mesh directly on the 3D surface.

Mesh Compression

Mesh compression is used to compactly store or transmit geometric models. As with other data, compression rates are inversely proportional to the data entropy. Thus higher compression rates can be obtained when models are represented by meshes that are as regular as possible, both topologically and geometrically. Topological regularity refers to meshes where almost all vertices have the same degree. Geometric regularity implies that triangles are similar to each other in terms of shape and size and vertices are close to the centroid of their neighbors. Such meshes can be obtained by parameterizing the original objects and then remeshing with regular sampling patterns [Gu et al., 2002]. The quality of the parameterization directly impacts the compression efficiency.

Surface Fitting

One of the earlier applications of mesh parameterization is surface fitting [Floater, 2000]. Many applications in geometry processing require a smooth analytical surface to be constructed from an input mesh. A parameterization of the mesh over a base domain significantly simplifies this task. Earlier methods either parameterized the entire mesh in the plane or segmented it and parameterized each patch independently. More recent methods, e.g. [Li et al., 2006] focus on constructing smooth global parameterizations and use those for fitting, achieving global continuity of the constructed surfaces.

Modeling from Material Sheets

While computer graphics focuses on virtual models, geometry processing has numerous real-world engineering applications. Particularly, planar mesh parameterization is an important tool when modeling 3D objects from sheets of material, ranging from garment modeling to metal forming or forging [Bennis et al., 1991; Julius et al., 2005]. All of these applications require the computation of planar patterns to form the desired 3D shapes. Typically, models are first segmented into nearly developable charts, and these charts are then parameterized in the plane.

Medical Visualization

Complex geometric structures are often better visualized and analyzed by mapping the surface normal-map, color, and other properties to a simpler, canonical domain. One of the structures for which such mapping is particularly useful is the human brain [Hurdal et al., 1999; Haker et al., 2000]. Most methods for brain mapping use the fact that the brain has genus zero, and visualize it through spherical [Haker et al., 2000] or planar [Hurdal et al., 1999] parameterization.

Chapter 2

Barycentric Mappings

In many applications, and in particular in computer graphics, it is nowadays common to work with piecewise linear surfaces in the form of triangle meshes, and we will mainly stick to this type of surface for most of these course notes.

2.1 Triangle Meshes

Let us denote points in $\mathbb{R}^3$ by $\boldsymbol{p} = (x, y, z)$ and points in $\mathbb{R}^2$ by $\boldsymbol{u} = (u, v)$. An *edge* is then defined as the convex hull of (or, equivalently, the line segment between) two distinct points and a *triangle* as the convex hull of three non-collinear points. We will denote edges and triangles in $\mathbb{R}^3$ with capital letters and those in $\mathbb{R}^2$ with small letters, for example, $e = [\boldsymbol{u}_1, \boldsymbol{u}_2]$ and $T = [\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3]$.

A *triangle mesh* $S_{\mathcal{T}}$ is the union of a set of *surface triangles* $\mathcal{T} = \{T_1, \ldots, T_m\}$ which intersect only at common edges $\mathcal{E} = \{E_1, \ldots, E_l\}$ and vertices $\mathcal{V} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n+b}\}$. More specifically, the set of vertices consists of n *interior* vertices $\mathcal{V}_I = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ and b *boundary* vertices $\mathcal{V}_B = \{\boldsymbol{p}_{n+1}, \ldots, \boldsymbol{p}_{n+b}\}$. Two distinct vertices $\boldsymbol{p}_i, \boldsymbol{p}_j \in \mathcal{V}$ are called *neighbours*, if they are the end points of some edge $E = [\boldsymbol{p}_i, \boldsymbol{p}_j] \in \mathcal{E}$, and for any $\boldsymbol{p}_i \in \mathcal{V}$ we let $N_i = \{j : [\boldsymbol{p}_i, \boldsymbol{p}_j] \in \mathcal{E}\}$ be the set of indices of all neighbours of $\boldsymbol{p}_i$.

A *parameterization* f of $S_{\mathcal{T}}$ is usually specified the other way around, that is, by defining the inverse parameterization $g = f^{-1}$. This mapping g is uniquely determined by specifying the *parameter points* $\boldsymbol{u}_i = g(\boldsymbol{p}_i)$ for each vertex $\boldsymbol{p}_i \in \mathcal{V}$ and demanding that g is continuous and linear for each triangle. In this setting, $g|_T$ is the linear map from a surface triangle $T = [\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}_k]$ to the corresponding *parameter triangle* $t = [\boldsymbol{u}_i, \boldsymbol{u}_j, \boldsymbol{u}_k]$ and $f|_t = (g|_T)^{-1}$ is the inverse linear map from t to T. The parameter domain Ω finally is the union of all parameter triangles (see Figure 2.1).

2.2 Parameterization by Affine Combinations

A rather simple idea for constructing a parameterization of a triangle mesh is based on the following physical model. Imagine that the edges of the triangle mesh are springs that are connected at the vertices. If we now fix the boundary of this spring network somewhere in the plane, then the interior of this network will relax in the energetically

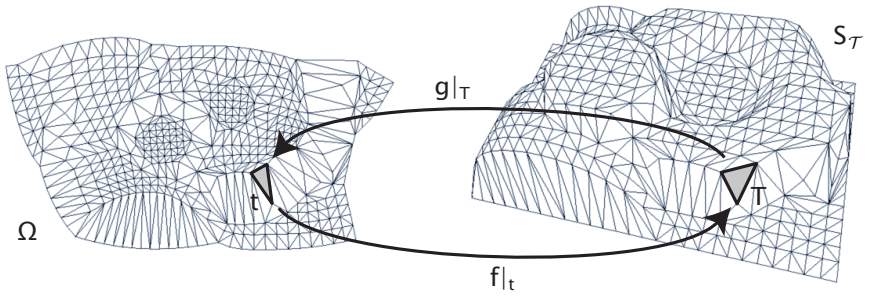


Figure 2.1: Parameterization of a triangle mesh.

most efficient configuration, and we can simply assign the positions where the joints of the network have come to rest as parameter points.

If we assume each spring to be ideal in the sense that the rest length is zero and the potential energy is just $\frac{1}{2}Ds^2$, where D is the spring constant and s the length of the spring, then we can formalize this approach as follows. We first specify the parameter points $\boldsymbol{u}_i = (u_i, v_i)$, $i = n+1, \ldots, n+b$ for the boundary vertices $\boldsymbol{p}_i \in \mathcal{V}_B$ of the mesh in some way (see Section 2.4). Then we minimize the overall spring energy

$$E = \tfrac{1}{2} \sum_{i=1}^{n+b} \sum_{j \in N_i} \tfrac{1}{2} D_{ij} \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2,$$

where $D_{ij} = D_{ji}$ is the spring constant of the spring between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, with respect to the unknown parameter positions $\boldsymbol{u}_i = (u_i, v_i)$ for the interior points[1]. As the partial derivative of E with respect to $\boldsymbol{u}_i$ is

$$\frac{\partial E}{\partial \boldsymbol{u}_i} = \sum_{j \in N_i} D_{ij}(\boldsymbol{u}_i - \boldsymbol{u}_j),$$

the minimum of E is obtained if

$$\sum_{j \in N_i} D_{ij} \boldsymbol{u}_i = \sum_{j \in N_i} D_{ij} \boldsymbol{u}_j$$

holds for all $i = 1, \ldots, n$. This is equivalent to saying that each interior parameter point $\boldsymbol{u}_i$ is an *affine combination* of its neighbours,

$$\boldsymbol{u}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{u}_j, \tag{2.1}$$

with normalized coefficients

$$\lambda_{ij} = D_{ij} \Big/ \sum_{k \in N_i} D_{ik}$$

that obviously sum to 1.

[1]The additional factor $\frac{1}{2}$ appears because summing up the edges in this way counts every edge twice.

By separating the parameter points for the interior and the boundary vertices in the sum on the right hand side of (2.1) we get

$$\boldsymbol{u}_i - \sum_{j \in N_i, j \le n} \lambda_{ij} \boldsymbol{u}_j = \sum_{j \in N_i, j > n} \lambda_{ij} \boldsymbol{u}_j,$$

and see that computing the coordinates u_i and v_i of the interior parameter points $\boldsymbol{u}_i$ requires to solve the linear systems

$$AU = \bar{U} \qquad \text{and} \qquad AV = \bar{V}, \tag{2.2}$$

where $U = (u_1, \ldots, u_n)$ and $V = (v_1, \ldots, v_n)$ are the column vectors of unknown coordinates, $\bar{U} = (\bar{u}_1, \ldots, \bar{u}_n)$ and $\bar{V} = (\bar{v}_1, \ldots, \bar{v}_n)$ are the column vectors with coefficients

$$\bar{u}_i = \sum_{j \in N_i, j > n} \lambda_{ij} u_j \qquad \text{and} \qquad \bar{v}_i = \sum_{j \in N_i, j > n} \lambda_{ij} v_j$$

and $A = (a_{ij})_{i,j=1,\ldots,n}$ is the $n \times n$ matrix with elements

$$a_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\lambda_{ij} & \text{if } j \in N_i, \\ 0 & \text{otherwise.} \end{cases}$$

The sparse linear systems (2.2) can be solved efficiently with state-of-the-art methods, like TAUCS [Toledo, 2003].

2.3 Barycentric Coordinates

The question remains how to choose the spring constants D_{ij} in the spring model, or more generally, the normalized coefficients λ_{ij} in (2.1). The simplest choice of constant spring constants $D_{ij} = 1$ goes back to the work of Tutte [1960, 1963] who used it in a more abstract graph-theoretic setting to compute straight line embeddings of planar graphs, and the idea of taking spring constants that are proportional to the lengths of the corresponding edges in the triangle mesh was used by Greiner and Hormann [1997]. A main drawback of both approaches is that they do not fulfill the following minimum requirement that we should expect from any parameterization method.

Linear reproduction: Suppose that $S_{\mathcal{T}}$ is contained in a plane so that its vertices have coordinates $\boldsymbol{p}_i = (x_i, y_i, 0)$ with respect to some appropriately chosen orthonormal coordinate frame. Then a globally isometric (and thus optimal) parameterization can be defined by just using the local coordinates $\boldsymbol{x}_i = (x_i, y_i)$ as parameter points themselves, that is, by setting $\boldsymbol{u}_i = \boldsymbol{x}_i$ for $i = 1, \ldots, n + b$. As the overall parameterization then is a linear function, we say that a parameterization method has *linear reproduction* if it produces such an isometric mapping in this setting.

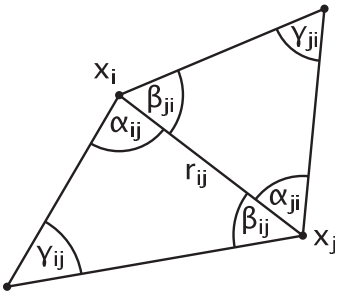


Figure 2.2: Notation for the construction of barycentric coordinates.

In the setting from the previous section, linear reproduction can be achieved if the parameter points for the boundary vertices are set correctly and the values λ_{ij} are chosen such that

$$\boldsymbol{x}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{x}_j \qquad \text{and} \qquad \sum_{j \in N_i} \lambda_{ij} = 1$$

for all interior vertices. Values λ_{ij} with both these properties are also called *barycentric coordinates* of $\boldsymbol{x}_i$ with respect to its neighbours $\boldsymbol{x}_j$, $j \in N_i$. If some $\boldsymbol{x}_i$ has exactly three neighbours, then the λ_{ij} are uniquely defined and these barycentric coordinates inside triangles actually have many useful applications in computer graphics (e.g., Gouraud and Phong shading, ray-triangle-intersection), geometric modelling (e.g., triangular Bézier patches, splines over triangulations), and many other fields (e.g., the finite element method, terrain modelling).

For polygons with more than three vertices, the barycentric coordinates of a point in the interior are, however, not unique anymore and there are several ways of defining them. The most popular of them can all be described in a common framework [Floater et al., 2006] that we shall briefly review. For any interior point $\boldsymbol{x}_i$ and one of its neighbours $\boldsymbol{x}_j$ let $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ be the length of the edge $e_{ij} = [\boldsymbol{x}_i, \boldsymbol{x}_j]$ between the two points and let the angles at the corners of the triangles adjacent to e_{ij} be denoted as shown in Figure 2.2. The barycentric coordinates λ_{ij} of $\boldsymbol{x}_i$ with respect its neighbours $\boldsymbol{x}_j$, $j \in N_i$ can then be computed by the normalization $\lambda_{ij} = w_{ij} / \sum_{k \in N_i} w_{ik}$ from any of the following *homogeneous coordinates* w_{ij}.

- *Wachspress coordinates*: The earliest generalization of barycentric coordinates goes back to Wachspress [1975] who suggested to set

  $$w_{ij} = \frac{\cot \alpha_{ji} + \cot \beta_{ij}}{{r_{ij}}^2}.$$

  While he was mainly interested in applying these coordinates in finite element methods, Desbrun et al. [2002] used them for parameterizing triangle meshes and Meyer et al. [2002] for interpolating e.g. colour values inside convex polygons. Moreover, a simple geometric construction of these coordinates was given by Ju et al. [2005b].

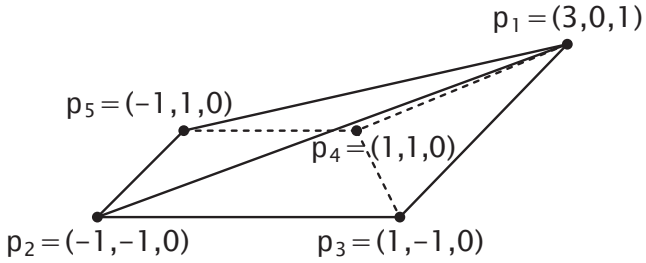


Figure 2.3: Example of a triangle mesh for which only the barycentric mapping with mean value coordinates is a bijection.

- *Discrete harmonic coordinates*: Another type of barycentric coordinates that stem from finite element methods and actually arise from the standard piecewise linear approximation to the Laplace equation are given by

$$w_{ij} = \cot \gamma_{ij} + \cot \gamma_{ji}.$$

  In the context of mesh parameterization, these coordinates were first used by Eck et al. [1995], but they have also been used to compute discrete minimal surfaces [Pinkall and Polthier, 1993].

- *Mean value coordinates*: By discretizing the mean value theorem, Floater [2003a] found yet another set of barycentric coordinates with

$$w_{ij} = \frac{\tan \frac{\alpha_{ij}}{2} + \tan \frac{\beta_{ji}}{2}}{r_{ij}}.$$

  While his main application was mesh parameterization, Hormann and Tarini [2004] and Hormann and Floater [2006] later showed that they have many other useful applications, in particular in computer graphics.

The beauty of all three choices is that the weights w_{ij} depend on angles and distances only, so that they can not only be computed if $\boldsymbol{x}_i$ and its neighbours are coplanar, but more generally for any interior vertex $\boldsymbol{p}_i \in \mathcal{V}_I$ of a triangle mesh if these angles and distances are just taken from the triangles around $\boldsymbol{p}_i$. Of course, an alternative approach that was introduced by Floater [1997] is to locally flatten the one-ring of triangles around $\boldsymbol{p}_i$ into the plane, e.g. with an exponential map, and then to compute the weights w_{ij} from this planar configuration.

A triangle mesh parameterization that is computed by solving the linear systems (2.2) with any set of barycentric coordinates λ_{ij} is called a *barycentric mapping* and obviously has the linear reproduction property, provided that an appropriate method for computing the parameter points for the boundary vertices, e.g. mapping them to the least squares plane (see Section 2.4), is used.

Despite this property, it may happen that a barycentric mapping, when constructed for a non-planar mesh, gives an unexpected result, as the simple example in Figure 2.3

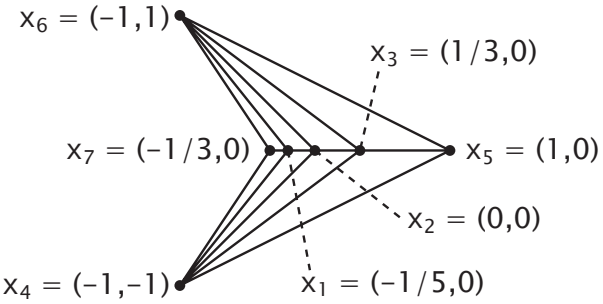


Figure 2.4: Example of a triangle mesh for which the linear system with Wachspress coordinates is singular.

illustrates. If we use $\boldsymbol{u}_2 = (-1,-1)$, $\boldsymbol{u}_3 = (1,-1)$, $\boldsymbol{u}_4 = (1,1)$, $\boldsymbol{u}_5 = (-1,1)$ as parameter points for the four boundary vertices and compute the barycentric weights λ_{12}, λ_{13}, λ_{14}, λ_{15} with the formulas described above, then we get the following positions for $\boldsymbol{u}_1$:

$$\begin{aligned} \text{Wachspress coordinates:} \quad & \boldsymbol{u}_1 = (-35.1369, 0), \\ \text{discrete harmonic coordinates:} \quad & \boldsymbol{u}_1 = (2.1138, 0), \\ \text{mean value coordinates:} \quad & \boldsymbol{u}_1 = (0.4538, 0). \end{aligned}$$

That is, only the mean value coordinates yield a position for $\boldsymbol{u}_1$ that is contained in the convex hull of the other four parameter points, and using the other coordinates will create parameter triangles that overlap, thus violating the bijectivity property that any parameterization should have.

The reason behind this behaviour is that the Wachspress and discrete harmonic coordinates can assume *negative* values in certain configurations like the one in Figure 2.3, whereas the mean values coordinates are always *positive*. And while overlapping triangles may occur for negative weights, this never happens if all weights are positive *and* the parameter points of the boundary vertices form a convex shape. The latter fact has first been proven by Tutte [1963] for the special case of $\lambda_{ij} = 1/\eta_i$ where $\eta_i = \#N_i$ is the number of $\boldsymbol{p}_i$'s neighbours, which are not true barycentric coordinates, but Floater [1997] observed that the proof carries over to arbitrary positive weights λ_{ij}. Recently, Gortler et al. [2006] could even show that the restriction to a convex boundary can be considerably relaxed, but this requires to solve a non-linear problem.

Another important aspect concerns the solvability of the linear systems (2.2) and it has been shown that the matrix A is always guaranteed to be non-singular for discrete harmonic [Pinkall and Polthier, 1993] and mean value coordinates [Floater, 1997]. For Wachspress coordinates, however, it may happen that the sum of homogeneous coordinates $W_i = \sum_{k \in N_i} w_{ik}$ is zero so that the normalized coordinates λ_{ij} and thus the matrix A are not even well-defined. In the example shown in Figure 2.4 this actually happens for all interior vertices $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$. But even if we skip the normalization and try to solve the equivalent and well-defined homogeneous systems $WAU = W\bar{U}$ and $WAV = W\bar{V}$ with $W = \operatorname{diag}(W_1, \ldots, W_n)$ instead, we find that the matrix WA is singular in this particular example, namely $WA = \begin{pmatrix} 0 & -50 & 0 \\ 40 & 0 & -24 \\ 0 & 18 & 0 \end{pmatrix}$.

2.4 The Boundary Mapping

The first step in constructing a barycentric mapping is to choose the parameter points for the boundary vertices and the simplest way of doing it is to just project the boundary vertices into the plane that fits the boundary vertices best in a least squares sense. However, for meshes with a complex boundary, this simple procedure may lead to undesirable fold-overs in the boundary polygon and cannot be used. In general, there are two issues to take into account here: (1) choosing the *shape* of the boundary of the parameter domain and (2) choosing the *distribution* of the parameter points around the boundary.

Choosing the shape

In many applications, it is sufficient (or even desirable) to take a rectangle or a circle as parameter domain, with the advantage that such a convex shape guarantees the bijectivity of the parameterization if positive barycentric coordinates like the mean value coordinates are used to compute the parameter points for the interior vertices. The convexity restriction may, however, generate big distortions near the boundary when the boundary of the triangle mesh $S_{\mathcal{T}}$ does not resemble a convex shape. One practical solution to avoid such distortions is to build a "virtual" boundary, i.e., to augment the given mesh with extra triangles around the boundary so as to construct an extended mesh with a "nice" boundary. This approach has been successfully used by Lee et al. [2002], and Kós and Várady [2003].

Choosing the distribution

The usual procedure mentioned in the literature is to use a simple univariate parameterization method such as *chord length* [Ahlberg et al., 1967] or *centripetal* parameterization [Lee, 1989] for placing the parameter points either around the whole boundary, or along each side of the boundary when working with a rectangular domain [Hormann, 2001, Section 1.2.5].

Despite these heuristics working pretty well in some cases, having to fix the boundary vertices may be a severe limitation in others and the next chapter studies parameterization methods that can include the position of the boundary parameter points in the optimization process and thus yield parameterizations with less distortion.

Chapter 3

Differential Geometry Primer

Barycentric mappings are efficient to compute and often give pretty good parameterizations. However, in order to quantify "good" and to derive more sophisticated parameterization methods, let us quickly review some of the basic properties from differential geometry that will be essential for understanding the motivation behind the methods described later. For more details and proofs of these properties, we refer the interested reader to the standard literature on differential geometry and in particular to the books by do Carmo [1976], Klingenberg [1978], Kreyszig [1991], and Morgan [1998].

3.1 Basic Definitions

Suppose that $\Omega \subset \mathbb{R}^2$ is some simply connected region (i.e., without any holes), for example,

$$\begin{aligned} \text{the } \textit{unit square}: &\quad \Omega = \{(u,v) \in \mathbb{R}^2 : u, v \in [0,1]\}, \quad \text{or} \\ \text{the } \textit{unit disk}: &\quad \Omega = \{(u,v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}, \end{aligned}$$

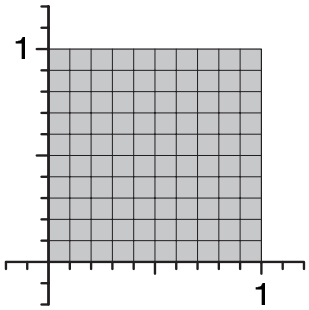
and that the function $f : \Omega \to \mathbb{R}^3$ is *continuous* and an *injection* (i.e., no two distinct points in Ω are mapped to the same point in $\mathbb{R}^3$). We then call the image S of Ω under f a *surface*,

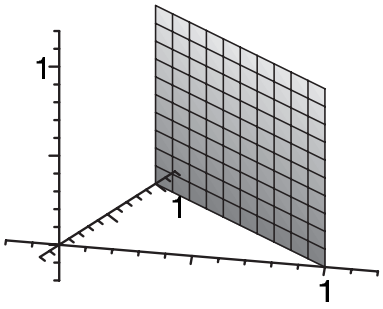
$$S = f(\Omega) = \{f(u,v) : (u,v) \in \Omega\},$$

and say that f is a *parameterization* of S over the *parameter domain* Ω. It follows from the definition of S that f is actually a *bijection* between Ω and S and thus admits to define its inverse $f^{-1} : S \to \Omega$. Here are some examples:

1. simple linear function:

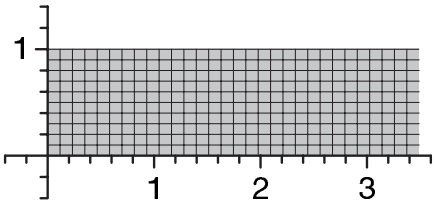
$$\begin{aligned} \textit{parameter domain}: &\quad \Omega = \{(u,v) \in \mathbb{R}^2 : u, v \in [0,1]\} \\ \textit{surface}: &\quad S = \{(x,y,z) \in \mathbb{R}^3 : x, y, z \in [0,1], x + y = 1\} \\ \textit{parameterization}: &\quad f(u,v) = (u, 1-u, v) \\ \textit{inverse}: &\quad f^{-1}(x,y,z) = (x,z) \end{aligned}$$

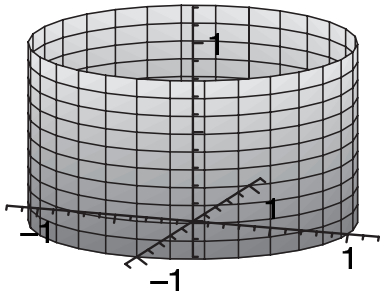




2. cylinder:

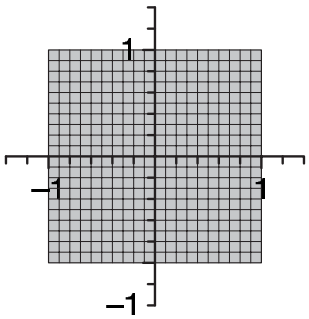
$$\begin{aligned} \textit{parameter domain:} \quad & \Omega = \{(u,v) \in \mathbb{R}^2 : u \in [0, 2\pi), v \in [0,1]\} \\ \textit{surface:} \quad & S = \{(x,y,z) \in \mathbb{R}^3 : x^2 + y^2 = 1, z \in [0,1]\} \\ \textit{parameterization:} \quad & f(u,v) = (\cos u, \sin u, v) \\ \textit{inverse:} \quad & f^{-1}(x,y,z) = (\arccos x, z) \end{aligned}$$

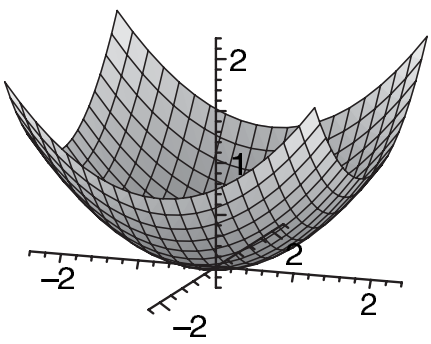




3. paraboloid:

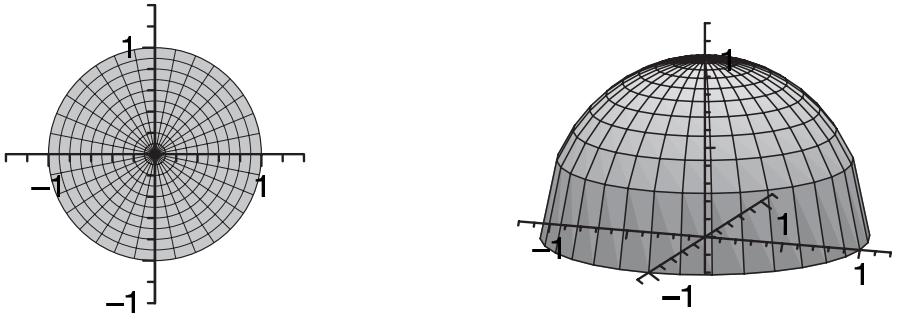
$$\begin{aligned} \textit{parameter domain:} \quad & \Omega = \{(u,v) \in \mathbb{R}^2 : u, v \in [-1,1]\} \\ \textit{surface:} \quad & S = \{(x,y,z) \in \mathbb{R}^3 : x, y \in [-2,2], z = \tfrac{1}{4}(x^2 + y^2)\} \\ \textit{parameterization:} \quad & f(u,v) = (2u, 2v, u^2 + v^2) \\ \textit{inverse:} \quad & f^{-1}(x,y,z) = (\tfrac{x}{2}, \tfrac{y}{2}) \end{aligned}$$





4. hemisphere (orthographic):

$$\begin{aligned} \textit{parameter domain:} &\quad \Omega = \{(u,v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\} \\ \textit{surface:} &\quad S = \{(x,y,z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1, z \geq 0\} \\ \textit{parameterization:} &\quad f(u,v) = (u, v, \sqrt{1-u^2-v^2}) \\ \textit{inverse:} &\quad f^{-1}(x,y,z) = (x,y) \end{aligned}$$



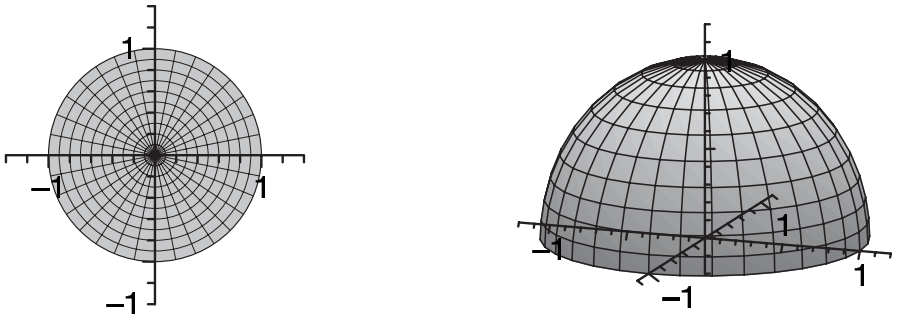
Having defined a surface S like that, we should note that the function f is by no means the only parameterization of S over Ω. In fact, given any bijection $\varphi : \Omega \to \Omega$, it is easy to verify that the *composition* of f and φ, i.e., the function $g = f \circ \varphi$, is a parameterization of S over Ω, too. For example, we can easily construct such a *reparameterization* φ from any bijection $\rho : [0,1] \to [0,1]$ by defining

$$\begin{aligned} \text{for the unit square:} &\quad \varphi(u,v) = (\rho(u), \rho(v)), \quad \text{or} \\ \text{for the unit disk:} &\quad \varphi(u,v) = (u\rho(u^2+v^2), v\rho(u^2+v^2)). \end{aligned}$$

In particular, taking the function $\rho(x) = \frac{2}{1+x}$ and applying this reparameterization of the unit disk to the parameterization of the hemisphere in the example above gives the following alternative parameterization:

5. hemisphere (stereographic):

$$\begin{aligned} \textit{parameter domain:} &\quad \Omega = \{(u,v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\} \\ \textit{surface:} &\quad S = \{(x,y,z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1, z \geq 0\} \\ \textit{parameterization:} &\quad f(u,v) = (\tfrac{2u}{1+u^2+v^2}, \tfrac{2v}{1+u^2+v^2}, \tfrac{1-u^2-v^2}{1+u^2+v^2}) \\ \textit{inverse:} &\quad f^{-1}(x,y,z) = (\tfrac{x}{1+z}, \tfrac{y}{1+z}) \end{aligned}$$



3.2 Intrinsic Surface Properties

Although the parameterization of a surface is not unique—and we will later discuss how to get the "best" parameterization with respect to certain criteria—it nevertheless is a very handy thing to have as it allows to compute a variety of properties of the surface. For example, if f is differentiable, then its *partial derivatives*

$$f_u = \frac{\partial f}{\partial u} \qquad \text{and} \qquad f_v = \frac{\partial f}{\partial v}$$

span the local *tangent plane* and by simply taking their cross product and normalizing the result we get the *surface normal*

$$n_f = \frac{f_u \times f_v}{\|f_u \times f_v\|}.$$

To simplify the notation, we will often speak of f_u and f_v as *the* derivatives and of n_f as *the* surface normal, but we should keep in mind that formally all three are functions from $\mathbb{R}^2$ to $\mathbb{R}^3$. In other words, for any point $(u, v) \in \Omega$ in the parameter domain, the tangent plane at the surface point $f(u, v) \in S$ is spanned by the two vectors $f_u(u, v)$ and $f_v(u, v)$, and $n_f(u, v)$ is the normal vector at this point[1]. Again, let us clarify this by considering two examples:

1. For the simple linear function $f(u, v) = (u, 1 - u, v)$ we get

   $$f_u(u, v) = (1, -1, 0) \qquad \text{and} \qquad f_v(u, v) = (0, 0, 1)$$

   and further

   $$n_f(u, v) = (\tfrac{-1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, 0),$$

   showing that the normal vector is constant for all points on S.

2. For the parameterization of the cylinder, $f(u, v) = (\cos u, \sin u, v)$, we get

   $$f_u(u, v) = (-\sin u, \cos u, 0) \qquad \text{and} \qquad f_v(u, v) = (0, 0, 1)$$

   and further

   $$n_f(u, v) = (\cos u, \sin u, 0),$$

   showing that the normal vector at any point $(x, y, z) \in S$ is just $(x, y, 0)$.

Note that in both examples the surface normal is *independent* of the parameterization. In fact, this holds for all surfaces and is therefore called an *intrinsic* property of the surface. Formally, we can also say that the surface normal is a function $n : S \to \mathbb{S}^2$, where $\mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}$ is the unit sphere in $\mathbb{R}^3$, so that

$$n(\boldsymbol{p}) = n_f(f^{-1}(\boldsymbol{p}))$$

[1] We tacitly assume that the parameterization is *regular*, i.e., f_u and f_v are always linearly independent and therefore n_f is non-zero.

for any $\boldsymbol{p} \in S$ and any parameterization f. As an exercise, you may want to verify this for the two alternative parameterizations of the hemisphere given above. Other intrinsic surface properties are the *Gaussian curvature* $K(\boldsymbol{p})$ and the *mean curvature* $H(\boldsymbol{p})$ as well as the total *area* of the surface $A(S)$. To compute the latter, we need the *first fundamental form*

$$\mathbf{I}_f = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_v \cdot f_u & f_v \cdot f_v \end{pmatrix} = \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$

where the product between the partial derivatives is the usual dot product in $\mathbb{R}^3$. It follows immediately from the Cauchy-Schwarz inequality that the determinant of this symmetric 2×2 matrix is always non-negative, so that its square root is always real. The area of the surface is then defined as

$$A(S) = \int_\Omega \sqrt{\det \mathbf{I}_f}\, du\, dv.$$

Take, for example, the orthographic parameterization $f(u, v) = (u, v, \sqrt{1 - u^2 - v^2})$ of the hemisphere over the unit disk. After some simplifications we find that

$$\det \mathbf{I}_f = \frac{1}{1 - u^2 - v^2}$$

and can compute the area of the hemisphere as follows:

$$\begin{aligned} A(S) &= \int_{-1}^{1} \int_{-\sqrt{1-v^2}}^{\sqrt{1-v^2}} \frac{1}{\sqrt{1 - u^2 - v^2}}\, du\, dv \\ &= \int_{-1}^{1} \left[\arcsin \frac{u}{\sqrt{1 - v^2}} \right]_{-\sqrt{1-v^2}}^{\sqrt{1-v^2}} dv \\ &= \int_{-1}^{1} \pi\, dv \\ &= 2\pi, \end{aligned}$$

as expected. Of course we get the same result if we use the stereographic parameterization, and you may want to try that as an exercise.

In order to compute the curvatures we must first assume the parameterization to be twice differentiable, so that its *second order* partial derivatives

$$f_{uu} = \frac{\partial^2 f}{\partial u^2}, \qquad f_{uv} = \frac{\partial^2 f}{\partial u \partial v}, \qquad \text{and} \qquad f_{vv} = \frac{\partial^2 f}{\partial v^2}$$

are well defined. Taking the dot products of these derivatives with the surface normal then gives the symmetric 2×2 matrix that is known as the *second fundamental form*

$$\mathbf{II}_f = \begin{pmatrix} f_{uu} \cdot n_f & f_{uv} \cdot n_f \\ f_{uv} \cdot n_f & f_{vv} \cdot n_f \end{pmatrix} = \begin{pmatrix} L & M \\ M & N \end{pmatrix}.$$

Gaussian and mean curvature are finally defined as the determinant and half the trace of the matrix $\mathbf{I}_f^{-1}\mathbf{II}_f$, respectively:

$$K = \det(\mathbf{I}_f^{-1}\mathbf{II}_f) = \frac{\det \mathbf{II}_f}{\det \mathbf{I}_f} = \frac{LN - M^2}{EG - F^2}$$

and

$$H = \tfrac{1}{2}\operatorname{trace}(\mathbf{I}_f^{-1}\mathbf{II}_f) = \frac{LG - 2MF + NE}{2(EG - F^2)}.$$

For example, carrying out these computations reveals that the curvatures are constant for most of the surfaces from above:

$$\begin{aligned} \text{simple linear function:} \quad & K = 0, \quad H = 0, \\ \text{cylinder:} \quad & K = 0, \quad H = \tfrac{1}{2}, \\ \text{hemisphere:} \quad & K = 1, \quad H = -1. \end{aligned}$$

As an exercise, show that the curvatures at any point $\boldsymbol{p} = (x, y, z)$ of the paraboloid from above are $K(\boldsymbol{p}) = \frac{1}{4(1+z)^2}$ and $H(\boldsymbol{p}) = \frac{2+z}{4(1+z)^{3/2}}$.

3.3 Metric Distortion

Apart from these intrinsic surface properties, there are others that depend on the parameterization, most importantly the *metric distortion*. Consider, for example, the two parameterizations of the hemisphere above. In both cases, the image of the surface on the right is overlaid by a regular grid, which actually is the image of the corresponding grid in the parameter domain shown on the left. You will notice that the surface grid looks more regular for the stereographic than for the orthographic projection and that the latter considerably stretches the grid in the radial direction near the boundary.

To better understand this kind of stretching, let us see what happens to the surface point $f(u, v)$ as we move a tiny little bit away from (u, v) in the parameter domain. If we denote this infinitesimal parameter displacement by $(\Delta u, \Delta v)$, then the new surface point $f(u + \Delta u, v + \Delta v)$ is approximately given by the first order Taylor expansion $\tilde{f}$ of f around (u, v),

$$\tilde{f}(u + \Delta u, v + \Delta v) = f(u, v) + f_u(u, v)\Delta u + f_v(u, v)\Delta v.$$

This linear function maps all points in the vicinity of $\boldsymbol{u} = (u, v)$ into the tangent plane $T_{\boldsymbol{p}}$ at $\boldsymbol{p} = f(u, v) \in S$ and transforms circles around $\boldsymbol{u}$ into ellipses around $\boldsymbol{p}$ (see Figure 3.1). The latter property becomes obvious if we write the Taylor expansion more compactly as

$$\tilde{f}(u + \Delta u, v + \Delta v) = \boldsymbol{p} + J_f(\boldsymbol{u})\tbinom{\Delta u}{\Delta v},$$

where $J_f = (f_u \; f_v)$ is the *Jacobian* of f, i.e. the 3×2 matrix with the partial derivatives of f as column vectors. Then using the *singular value decomposition* of the Jacobian,

$$J_f = U\Sigma V^T = U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix} V^T,$$

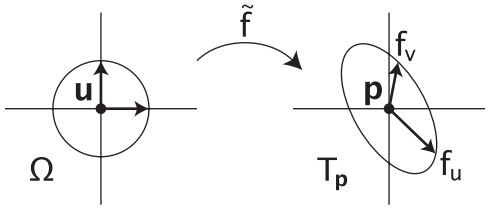


Figure 3.1: First order Taylor expansion $\tilde{f}$ of the parameterization f.

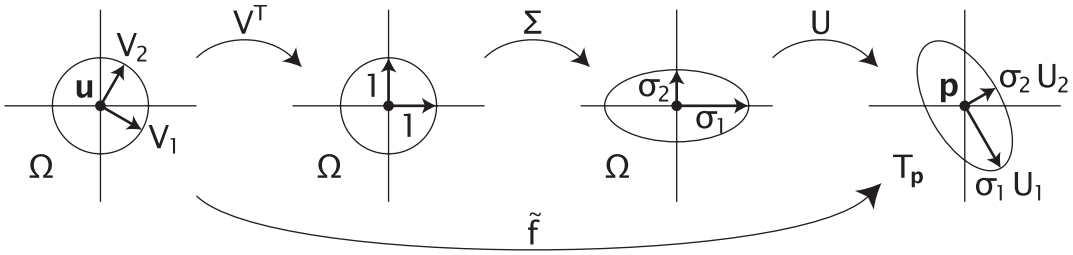


Figure 3.2: SVD decomposition of the mapping $\tilde{f}$.

with *singular values* $\sigma_1 \geq \sigma_2 > 0$ and *orthonormal* matrices $U \in \mathbb{R}^{3\times 3}$ and $V \in \mathbb{R}^{2\times 2}$ with column vectors U_1, U_2, U_3, and V_1, V_2, respectively, we can split up the linear transformation $\tilde{f}$ as shown in Figure 3.2:

1. The transformation V^T first rotates all points around $\boldsymbol{u}$ such that the vectors V_1 and V_2 are in alignment with the u- and the v-axes afterwards.

2. The transformation Σ then stretches everything by the factor σ_1 in the u- and by σ_2 in the v-direction.

3. The transformation U finally maps the unit vectors $(1, 0)$ and $(0, 1)$ to the vectors U_1 and U_2 in the tangent plane $T_{\boldsymbol{p}}$ at $\boldsymbol{p}$.

As a consequence, any circle of radius r around $\boldsymbol{u}$ will be mapped to an ellipse with semi-axes of length $r\sigma_1$ and $r\sigma_2$ around $\boldsymbol{p}$ and the orthonormal frame $[V_1, V_2]$ is mapped to the orthogonal frame $[\sigma_1 U_1, \sigma_2 U_2]$.

This transformation of circles into ellipses is called *local metric distortion* of the parameterization as it shows how f behaves locally around some parameter point $\boldsymbol{u} \in \Omega$ and the corresponding surface point $p = f(\boldsymbol{u}) \in S$. Moreover, all information about this local metric distortion is hidden in the singular values σ_1 and σ_2. For example, if both values are identical, then J_f is just a rotation plus uniform scaling and f does not distort angles around $\boldsymbol{u}$. Likewise, if the product of the singular values is 1, then the area of any circle in the parameter domain is identical to the area of the corresponding ellipse in the tangent plane and we say that f is locally area-preserving.

Computing the singular values directly is a bit tedious, so that we better resort to the fact that the singular values of any matrix A are the square roots of the *eigenvalues* of the matrix $A^T A$. In our case, the matrix ${J_f}^T J_f$ is an old acquaintance, namely the

first fundamental form,

$$J_f{}^T J_f = \begin{pmatrix} f_u{}^T \\ f_v{}^T \end{pmatrix} (f_u \; f_v) = \mathbf{I}_f = \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$

and we can easily compute the two eigenvalues λ_1 and λ_2 of this symmetric matrix by using the nifty little formula

$$\lambda_{1,2} = \tfrac{1}{2}\big((E+G) \pm \sqrt{4F^2 + (E-G)^2}\big).$$

We now summarize the main properties that a parameterization can have locally:

$$\begin{aligned} f \text{ is } \textit{isometric} \text{ or } \textit{length-preserving} &\iff \sigma_1 = \sigma_2 = 1 \iff \lambda_1 = \lambda_2 = 1, \\ f \text{ is } \textit{conformal} \text{ or } \textit{angle-preserving} &\iff \sigma_1 = \sigma_2 \iff \lambda_1 = \lambda_2, \\ f \text{ is } \textit{equiareal} \text{ or } \textit{area-preserving} &\iff \sigma_1\sigma_2 = 1 \iff \lambda_1\lambda_2 = 1. \end{aligned}$$

Obviously, any isometric mapping is conformal and equiareal, and every mapping that is conformal and equiareal is also isometric, in short,

$$\text{isometric} \iff \text{conformal} + \text{equiareal}.$$

Thus equipped, let us go back to the examples above and check their properties:

1. simple linear function:

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (u, 1-u, v) \\ \textit{Jacobian:} \quad & J_f = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \\ \textit{first fundamental form:} \quad & \mathbf{I}_f = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right) \\ \textit{eigenvalues:} \quad & \lambda_1 = 2, \quad \lambda_2 = 1 \end{aligned}$$

   This parameterization is neither conformal nor equiareal.

2. cylinder:

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (\cos u, \sin u, v) \\ \textit{Jacobian:} \quad & J_f = \left(\begin{smallmatrix} \cos u & 0 \\ -\sin u & 0 \\ 0 & 1 \end{smallmatrix}\right) \\ \textit{first fundamental form:} \quad & \mathbf{I}_f = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \\ \textit{eigenvalues:} \quad & \lambda_1 = 1, \quad \lambda_2 = 1 \end{aligned}$$

   This parameterization is isometric.

3. paraboloid:

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (2u, 2v, u^2+v^2) \\ \textit{Jacobian:} \quad & J_f = \begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 2u & 2v \end{pmatrix} \\ \textit{first fundamental form:} \quad & \mathbf{I}_f = \begin{pmatrix} 4+4u^2 & 4uv \\ 4uv & 4+4v^2 \end{pmatrix} \\ \textit{eigenvalues:} \quad & \lambda_1 = 4, \quad \lambda_2 = 4(1+u^2+v^2) \end{aligned}$$

This mapping is not equiareal and conformal only at $(u, v) = (0, 0)$.

4. hemisphere (orthographic):

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (u, v, \tfrac{1}{d}) \quad \text{with} \quad d = \tfrac{1}{\sqrt{1-u^2-v^2}} \\ \textit{Jacobian:} \quad & J_f = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -ud & -vd \end{pmatrix} \\ \textit{first fundamental form:} \quad & \mathbf{I}_f = \begin{pmatrix} 1+u^2d^2 & uvd^2 \\ uvd^2 & 1+v^2d^2 \end{pmatrix} \\ \textit{eigenvalues:} \quad & \lambda_1 = 1, \quad \lambda_2 = d^2 \end{aligned}$$

This mapping is isometric at $(u, v) = (0, 0)$, but neither conformal nor equiareal elsewhere.

5. hemisphere (stereographic):

$$\begin{aligned} \textit{parameterization:} \quad & f(u,v) = (2ud, 2vd, (1-u^2-v^2)d) \quad \text{with} \quad d = \tfrac{1}{1+u^2+v^2} \\ \textit{Jacobian:} \quad & J_f = \begin{pmatrix} 2d-4u^2d^2 & -4uvd^2 \\ -4uvd^2 & 2d-4v^2d^2 \\ -4ud^2 & -4vd^2 \end{pmatrix} \\ \textit{first fundamental form:} \quad & \mathbf{I}_f = \begin{pmatrix} 4d^2 & 0 \\ 0 & 4d^2 \end{pmatrix} \\ \textit{eigenvalues:} \quad & \lambda_1 = 4d^2, \quad \lambda_2 = 4d^2 \end{aligned}$$

This mapping is always conformal, but equiareal and thus isometric only at the boundary of Ω, i.e., for $u^2 + v^2 = 1$.

It turns out that the only parameterization that is optimal in the sense that it is isometric everywhere and thus does not introduce any distortion at all is the one for the cylinder. In fact, it was shown by Gauß [1827] that a globally isometric parameterization exists only for *developable* surfaces like planes, cones, and cylinders with vanishing Gaussian curvature $K(\boldsymbol{p}) = 0$ at all surface points $\boldsymbol{p} \in S$. As an exercise, you can try to find such a globally isometric parameterization for the planar surface patch from the first example. Other interesting parameterizations are those that are globally conformal like the stereographic projection for the hemisphere, and it was shown by Riemann [1851] that such a parameterization exists for any surface that is topologically equivalent to a disk and any simply connected parameter domain.

Chapter 4

Minimizing Metric Distortion

Using the machinery of the previous chapter, the "best" parameterization f of a surface S over a parameter domain Ω is generally found as follows. We first need a bivariate non-negative function $E : \mathbb{R}_+^2 \to \mathbb{R}_+$ that measures the local distortion of a parameterization with singular values σ_1 and σ_2. Usually, this function has a global minimum at $(1, 1)$ so as to favour isometry, but depending on the application, it may also be defined such that the minimal value is taken along the whole line (x, x) for $x \in \mathbb{R}_+$, for example, if conformal mappings shall be preferred. The overall distortion of a particular parameterization f is then measured by simply averaging the local distortion over the whole domain,

$$\bar{E}(f) = \int_\Omega E(\sigma_1(u, v), \sigma_2(u, v))\, du\, dv \Big/ A(\Omega), \tag{4.1}$$

and the best parameterization with respect to E is then found by minimizing $\bar{E}(f)$ over the space of all admissible parameterizations.

4.1 Distortion of Piecewise Linear Parameterizations

We are mainly interested in the case where the surface S is a triangle mesh $S_\mathcal{T}$ and f is a piecewise linear mapping from the parameter triangles $t \in \Omega$ to the corresponding surface triangles $T \in \mathcal{T}$ (cf. Section 2.1). In this setting, the distortion is constant per triangle with singular values σ_1^t and σ_2^t and the overall distortion (4.1) simplifies to

$$\bar{E}(f) = \sum_{t\in\Omega} E(\sigma_1^t, \sigma_2^t) A(t) \Big/ \sum_{t\in\Omega} A(t). \tag{4.2}$$

Alternatively, we can also consider the overall distortion of the inverse parameterization $g = f^{-1}$,

$$\bar{E}(g) = \sum_{T\in\mathcal{T}} E(\sigma_1^T, \sigma_2^T) A(T) \Big/ \sum_{T\in\mathcal{T}} A(T), \tag{4.3}$$

with the advantage that the sum of surface triangle areas in the denominator is constant and can thus be neglected upon minimization. Note that the singular values of the linear map $g|_T$ are just the inverse of the linear map $f|_t$, that is, $\sigma_1^T = 1/\sigma_2^t$ and $\sigma_2^T = 1/\sigma_1^t$.

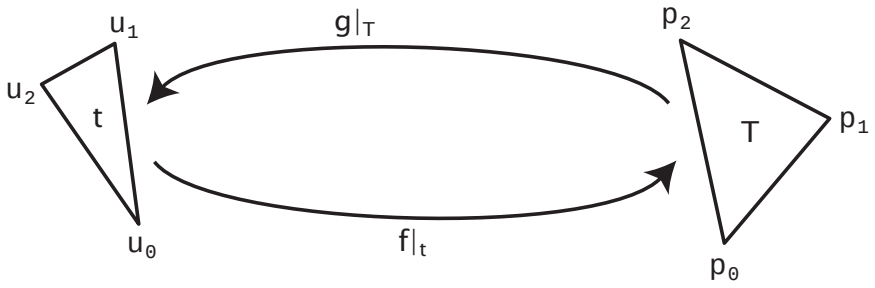


Figure 4.1: Local linear maps between a parameter triangle t and a surface triangle T.

In either case, the best parameterization with respect to the distortion measure E is then found by minimizing $\bar{E}$ with respect to the unknown parameter points. To this end, it helps to express its main constituents in terms of the unknowns $\boldsymbol{u}_i$ and the given vertices $\boldsymbol{p}_i$. The areas of the parameter triangle $t = [\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2]$ and its corresponding surface triangle $T = [\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2]$ are given by the formulas

$$A(t) = \tfrac{1}{2}\det(\boldsymbol{u}_1 - \boldsymbol{u}_0, \boldsymbol{u}_2 - \boldsymbol{u}_0) \qquad \text{and} \qquad A(T) = \tfrac{1}{2}\|(\boldsymbol{p}_1 - \boldsymbol{p}_0) \times (\boldsymbol{p}_2 - \boldsymbol{p}_0)\|.$$

And some basic calculations reveal the following identities for the singular values of the local linear maps $f|_t$ and $g|_T$ between both triangles (see Figure 4.1):

$$\begin{gathered}(\sigma_1^t)^2 + (\sigma_2^t)^2 = \frac{1}{A(t)^2}\sum_{i=0}^{2} \|\boldsymbol{u}_{i+2} - \boldsymbol{u}_{i+1}\|^2\big[(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)\cdot(\boldsymbol{p}_{i+2} - \boldsymbol{p}_i)\big], \\ \sigma_1^t\sigma_2^t = \frac{A(T)}{A(t)}\end{gathered} \tag{4.4}$$

and

$$\begin{gathered}(\sigma_1^T)^2 + (\sigma_2^T)^2 = \frac{1}{A(T)^2}\sum_{i=0}^{2} \|\boldsymbol{u}_{i+2} - \boldsymbol{u}_{i+1}\|^2\big[(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)\cdot(\boldsymbol{p}_{i+2} - \boldsymbol{p}_i)\big], \\ \sigma_1^T\sigma_2^T = \frac{A(t)}{A(T)},\end{gathered} \tag{4.5}$$

where all indices of parameter points and vertices must be taken modulo 3. Note that $A(t)$ as well as the two quantities in (4.5) are quadratic in the unknowns $\boldsymbol{u}_i$, whereas the two expressions in (4.4) are rational functions of $\boldsymbol{u}_i$.

4.2 Harmonic Maps

One of the first parameterization methods that were used in computer graphics [Pinkall and Polthier, 1993; Eck et al., 1995] considers the *Dirichlet energy* of the inverse parameterization g, which is given by $\bar{E}(g)$ in (4.3) with the local distortion measure

$$E_{\mathrm{D}}(\sigma_1, \sigma_2) = \tfrac{1}{2}\big({\sigma_1}^2 + {\sigma_2}^2\big).$$

As mentioned above, the energy $\bar{E}(g)$ is quadratic in the parameter points $\boldsymbol{u}_i$ in this case. It can thus be minimized by solving a linear system and it turns out that the inverse $f = g^{-1}$ of the resulting (discrete) harmonic map g is exactly the barycentric mapping with discrete harmonic coordinates that we already discussed in Section 2.3.

A potential disadvantage of harmonic maps is that they require to fix the boundary of the parameterization in advance. Otherwise, the parameterization degenerates, because E_{D} takes its minimum for mappings with $\sigma_1 = \sigma_2 = 0$, so that an optimal parameterization is one that maps all surface triangles T to a single point. And even if the boundary is set up correctly, it may happen that the parameterization violates the bijectivity property (cf. Section 2.3).

4.3 Conformal Maps

Another approach that was independently discovered by Desbrun et al. [2002] and Lévy et al. [2002] is to use the *conformal energy*

$$E_{\mathrm{C}}(\sigma_1, \sigma_2) = \tfrac{1}{2}(\sigma_1 - \sigma_2)^2$$

as a local distortion measure in (4.3). This still yields a linear problem to solve, but only two of the boundary vertices need to be fixed in order to give a unique solution. Unfortunately, the resulting parameterization depends on the choice of these two vertices and can vary significantly, but Mullen et al. [2008] recently showed how to get the best of all choices. However, the problem of potential non-bijectivity remains.

Conformal and harmonic maps are closely related. Indeed, we first observe for the local distortion measures that

$$E_{\mathrm{D}}(\sigma_1, \sigma_2) - E_{\mathrm{C}}(\sigma_1, \sigma_2) = \sigma_1\sigma_2$$

and it is then straightforward to conclude that the overall distortions defer by

$$\bar{E}_{\mathrm{D}}(g) - \bar{E}_{\mathrm{C}}(g) = \sum_{t\in\Omega} A(t) \Big/ \sum_{T\in\mathcal{T}} A(T) = \frac{A(\Omega)}{A(S_{\mathcal{T}})}.$$

Therefore, if we take a conformal map, fix its boundary and thus the area of the parameter domain Ω, and then compute the harmonic map with this boundary, then we get the same mapping, which illustrates the well-known fact that any conformal mapping is harmonic, too.

The conformal energy E_{C} is clearly minimal for locally conformal mappings with $\sigma_1 = \sigma_2$. However, it is not the only energy that favours conformality. Hormann and Greiner [2000a] introduced the so-called *MIPS energy*

$$E_{\mathrm{M}}(\sigma_1, \sigma_2) = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} = \frac{{\sigma_1}^2 + {\sigma_2}^2}{\sigma_1\sigma_2},$$

which is also minimal if and only if $\sigma_1 = \sigma_2$. An advantage of this distortion measure is the symmetry with respect to inversion,

$$E_{\mathrm{M}}(\sigma_1^T, \sigma_2^T) = E_{\mathrm{M}}(\sigma_1^t, \sigma_2^t),$$

so that it measures the distortion of both mappings $f|_t$ and $g|_T$ at the same time. The disadvantage is that minimizing either of the overall distortion energies in (4.2) and (4.3) is a non-linear problem. However, $\bar{E}_{\mathrm{M}}(f)$ is a quadratic rational function in the $\boldsymbol{u}_i$ and $\bar{E}_{\mathrm{M}}(g)$ is a sum of quadratic rational functions, and both can be minimized with standard gradient descent methods. Moreover, it is possible to determine the first and second derivatives analytically and to guarantee the bijectivity of the resulting mapping [Hormann, 2001].

4.4 Other Distortion Measures

A variety of other local distortion measures have been proposed in the literature, which all yield different non-linear optimization problems. The *Green-Lagrange deformation tensor* [Maillot et al., 1993]

$$E_{\mathrm{G}}(\sigma_1, \sigma_2) = \left({\sigma_1}^2 - 1\right)^2 + \left({\sigma_2}^2 - 1\right)^2$$

as well as the energy by Degener et al. [2003],

$$E_\theta(\sigma_1, \sigma_2) = \left(\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}\right)\left(\sigma_1\sigma_2 + \frac{1}{\sigma_2\sigma_1}\right)^\theta \qquad \text{with} \qquad \theta > 0,$$

are both minimal for locally isometric mappings with $\sigma_1 = \sigma_2 = 1$. But while the first one tends to favour shrinking, e.g., $E_{\mathrm{G}}(1, 1/2) = E_{\mathrm{G}}(2, 1)/4$, the second one is symmetric with respect to shrinking and stretching, e.g., $E_\theta(1, 1/2) = E_\theta(2, 1)$, which is similar to the difference between the conformal and the MIPS energy. Moreover, the parameter θ allows to mediate between angle and area distortion, and a good heuristic is $\theta \approx 3$ [Tarini et al., 2004].

Sander et al. [2001] introduced the L_∞ *stretch energy*

$$E_\infty(\sigma_1, \sigma_2) = \sigma_1$$

and the L^2 *stretch energy*

$$E_2(\sigma_1, \sigma_2) = \sqrt{\tfrac{1}{2}\left({\sigma_1}^2 + {\sigma_2}^2\right)},$$

where the latter is just the square root of the local Dirichlet energy E_{D}. Still, the results of Sander et al. [2001] are different from harmonic maps, because they consider the singular values of the piecewise linear maps $f|_t$ instead of $g|_T$.

Finally, Sorkine et al. [2002] suggested to symmetrize the L_∞ stretch and use

$$E_{\mathrm{S}}(\sigma_1, \sigma_2) = \max(\sigma_1, 1/\sigma_2)$$

as a local deformation measure.

4.5 Angle-Space Methods

Instead of defining a planar parameterization in terms of vertex coordinates, both the ABF/ABF++ method [Sheffer and de Sturler, 2000, 2001; Sheffer et al., 2005] and the circle-patterns algorithm [Kharevych et al., 2006] define it in terms of the angles of the planar triangles. Figure 5.3 provides a comparison between angle-space and direct conformal methods. As demonstrated, angle-space methods introduce significantly less stretch into the parameterization on models that have regions of high Gaussian curvature.

The ABF method (Angle Based Flattening) [Sheffer and de Sturler, 2000, 2001] is based on the following observation: a planar triangulation is uniquely defined by the corner angles of its triangles (modulo a similarity transformation). Based on this observation the authors reformulate the parameterization problem – finding (u_i, v_i) coordinates – in terms of angles, that is to say finding the angles α_k^t, where α_i^t denotes the angle at the corner of triangle t incident to vertex k.

To ensure that the 2D angles define a valid triangulation, a set of constraints needs to be satisfied.

- Triangle validity (for each triangle t):

$$\forall t \in T, \quad \alpha_1^t + \alpha_2^t + \alpha_3^t - \pi = 0.$$

- Planarity (for each internal vertex v):

$$\forall v \in V_{int}, \quad \sum_{(t,k)\in v^*} \alpha_k^t - 2\pi = 0,$$

  where V_{int} denotes the set of internal vertices, and where v^* denotes the set of angles incident to vertex v.

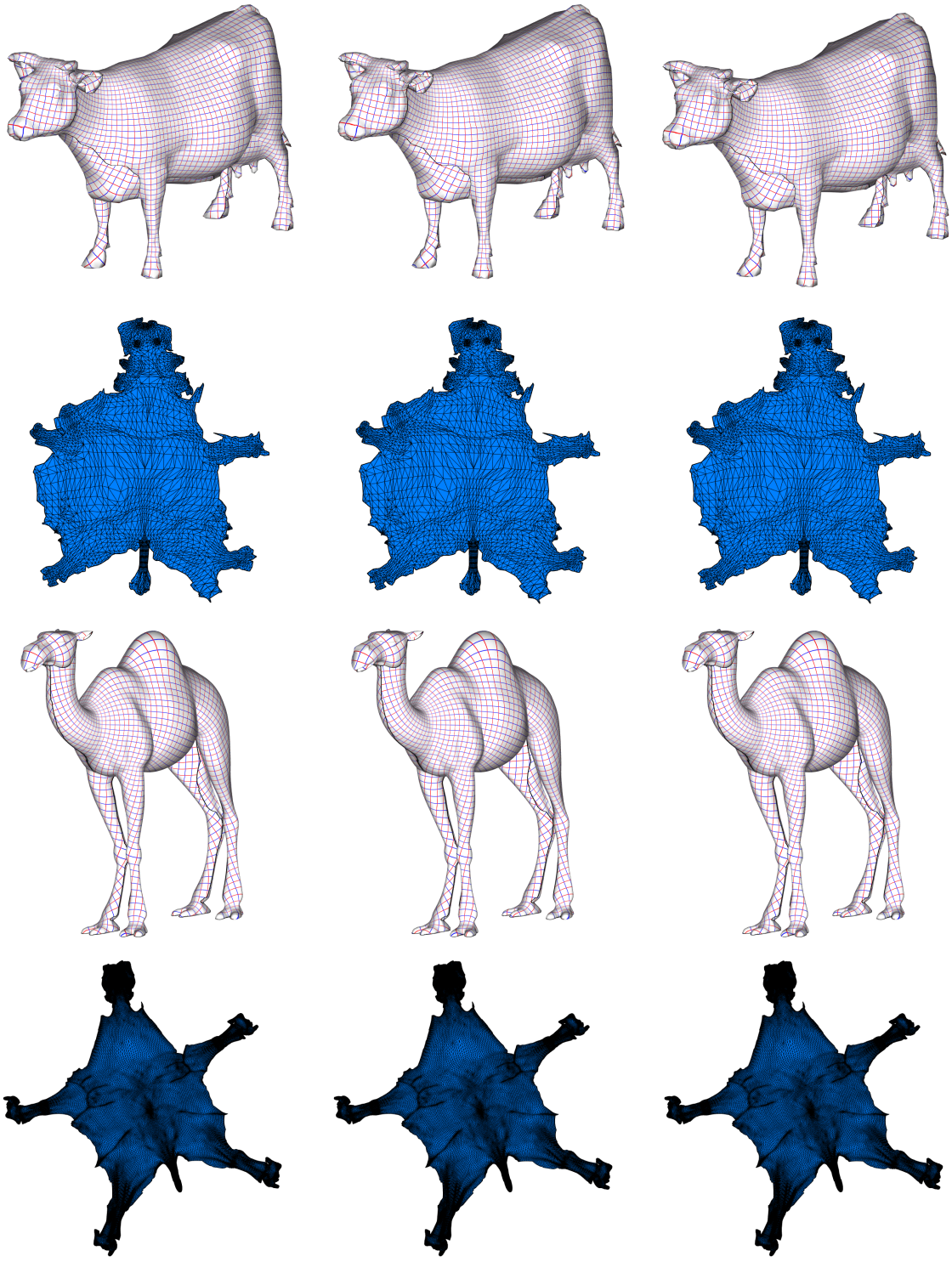
- Positivity: $\alpha_k^t > 0$ for all angles. We note that this constraint can be ignored in most practical setups [Sheffer et al., 2005], simplifying the solution process.

- Reconstruction (for each internal vertex): this constraints ensures that edges shared by pairs of triangles has the same length:

$$\forall v \in V_{int}, \quad \prod_{(t,k)\in v^*} \sin \alpha_{k\oplus 1}^t - \prod_{(t,k)\in v^*} \sin \alpha_{k\ominus 1}^t = 0.$$

  The indices $k \oplus 1$ and $k \ominus 1$ denote the next and previous angle in the triangle. Intuitively, note that the product $\sin \alpha_{k\oplus 1}^t \sin \alpha_{k\ominus 1}^t$ corresponds to the product of the ratio between the lengths of two consecutive edges around vertex k. If they do not match, it is the possible to "turn around" vertex k without "landing" on the starting point.

They search for angles that are as close as possible to the original 3D mesh angles β_k^t and which satisfy those constraints.



(a) (b) (c)

Figure 4.2: Convergence of ABF++ (cf. Table 4.1). Parameterization result after one (a), two (b), and ten iteration (c).

	iterations	L_2^{stretch}	L_2^{shear}	$AngD$
cow	1	1.10634	$4.69 \cdot 10^{-4}$	$4.19 \cdot 10^{-4}$
	2	1.10641	$4.64 \cdot 10^{-4}$	$3.81 \cdot 10^{-4}$
	10	1.10638	$4.64 \cdot 10^{-4}$	$3.81 \cdot 10^{-4}$
camel	1	1.09616	$5.21 \cdot 10^{-3}$	$5.09 \cdot 10^{-3}$
	2	1.09645	$5.15 \cdot 10^{-3}$	$5.04 \cdot 10^{-3}$
	10	1.09639	$5.15 \cdot 10^{-3}$	$5.04 \cdot 10^{-3}$

Table 4.1: Convergence of ABF++ (cf. Figure 4.2). The formulas for the stretch, shear and angular distortion are taken from [Sheffer et al., 2005].

The constrained numerical optimization problem is solved using the Lagrange multipliers method. The stationary point of the Lagrangian is computed using Newton's method. Each step requires to solve a linear system of size $2n_V + 4n_F$, where n_V denotes the number of interior mesh vertices and n_F the number of facets. They then convert the solution angles into actual (u, v) vertex coordinates using a propagation procedure. The resulting parameterizations are guaranteed to have no flipped triangles, i.e. be *locally* bijective, but can contain global overlaps. The authors provided a mechanism for resolving such overlaps, but it has no guarantees of convergence. The original ABF method is relatively slow and suffers from stability problems in the angle-to-uv conversion stage for large meshes.

ABF was augmented to yield ABF++ [Sheffer et al., 2005], a technique addressing both problems. ABF++ introduces a stable angle-to-uv conversion using the LSCM method to obtain (u, v) coordinates from the set of angles. Sheffer et al. [2005] also drastically speeds up the solution by introducing both direct and hierarchical solution approaches. For the direct solver they switch from Newton to Gauss-Newton solution setup, simplifying the structure of the linear system solved at each step. They then use the structure of the Hessian matrix to solve the linear system without explicitly inverting the entire Hessian. Thus they manage to reduce the size of the explicitly inverted matrix at each step by a factor of five to about $2n_V$ (note that on a manifold mesh $2n_V \approx n_F$). This is what one may expect for a parameterization problem since this corresponds to the number of degrees of freedom. We make an interesting observation about the convergence behaviour of ABF+ throughout the Gauss-Newton process. While it takes five to ten iterations to reduce the minimization error (the gradient of the Lagrangian) below 10^{-6}, practically all the changes from iteration two and on occur in the Lagrange multipliers. Even after one iteration (Figure 4.2(a)) the computed angles are very close to the final ones and they completely converge within a couple of iterations (Figure 4.2 and Table 4.1).

Several modifications of the formulation were proposed over the last few years. For instance, Zayer et al. [2003] introduced additional constraints on the angles enforcing the parameter domain to have convex boundaries, thus guaranteeing global bijectivity.

Kharevych et al. [2006] use a circle patterns approach where each circle corresponds to a mesh face. In contrast to classical circle packing, they use intersecting circles, with

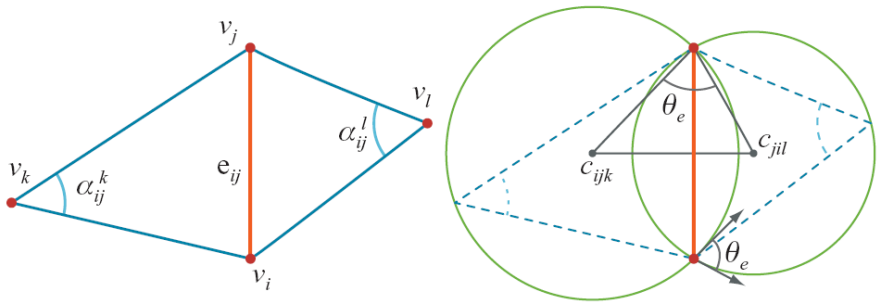


Figure 4.3: Circle patterns notations.

prescribed intersection angles θ_e (Figure 4.3). Given these angles, the circle radii follow as the unique minimizer of a convex energy. The method first computes the intersection angles using non-linear constrained optimization and then finds the circle radii using unconstrained minimization. To find the intersection angles it first computes a set of feasible triangle angles α_{ij}^k which are close to the 3D angles β_{ij}^k and satisfy the following constraints:

- Triangle validity (for each triangle t):

$$\forall t_{ijk} \in T, \quad \alpha_{ij}^t + \alpha_{ik}^t + \alpha_{kj}^t - \pi = 0.$$

- Planarity (for each internal vertex v):

$$\forall v \in V_{int}, \quad \sum_{(t,ij) \in v^*} \alpha_{ij}^t - 2\pi = 0,$$

  where V_{int} denotes the set of internal vertices, and where v^* denotes the set of angles incident to vertex v.

- Positivity: $\alpha_{ij}^t > 0$ for all angles.

- Local Delaunay property (for each edge e_{ij}):

$$\forall e_{ij} \in E, \quad \alpha_{ij}^k + \alpha_{ij}^l < \pi$$

We note that three of these constraints are similar to those imposed by the ABF setup, with the non-linear reconstruction constraint on interior vertices replaced by the inequality local Delaunay constraint per edge. The intersection angles are computed from the feasible triangle angles using a simple formula. Since the solution for the intersection angles is conformal only for a Delaunay triangulation, the authors employ a pre-processing stage that involves "intrinsic" Delaunay triangulation. At the final stage of the method, the computed radii are converted to actual uv-coordinates. The method supports equality and inequality constraints on the angles along the boundary of the

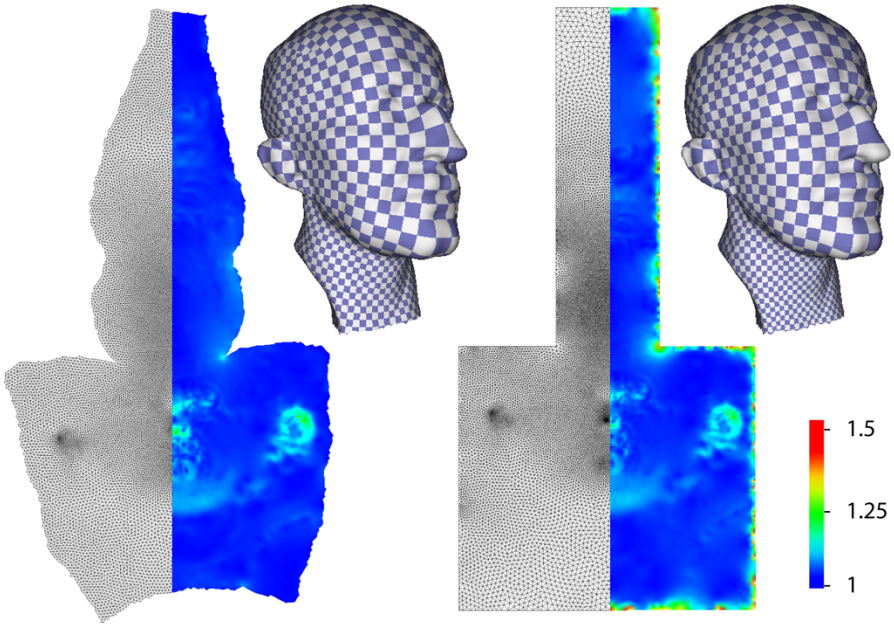


Figure 4.4: Fixed and free-boundary parameterizations computed using circle patterns.

planar parameter domain. Similar to ABF, the parameterization is locally bijective, but can contain global overlaps. The amount of distortion introduced by the method is comparable with that of ABF/ABF++ techniques.

Kharevych et al. [2006] propose an extension of the method to global parameterization of meshes by introducing cone singularities. They observe that in angle space formulation the only difference between parameterizing the mesh boundary and its interior mesh is the constraints imposed on the interior vertices which are not imposed on boundary ones, and that it is possible to define a global parameterization by specifying an unconnected subset of mesh vertices as boundary vertices. Thus they first compute a solution in angle space with a set of cone singularity vertices specified by the user as boundary. For planar parameterization, to perform the angle-to-uv conversion they later compute edge paths between these vertices. The obtained parameterization is globally continuous up to translation and rotation, everywhere except at the cone singularities. The proposed approach can be directly applied to other angle-space methods such as ABF/ABF++.

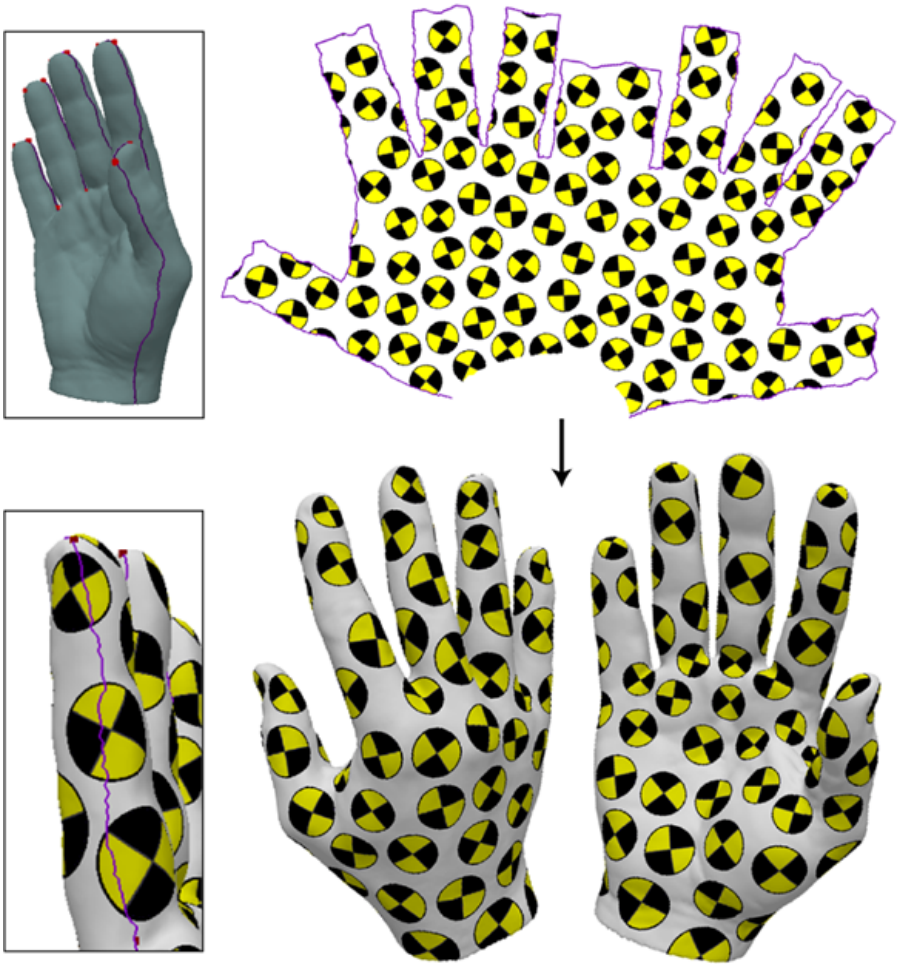


Figure 4.5: Global parameterization with cone singularities.

Chapter 5

Comparison of Planar Methods

The different methods reviewed minimize different types of distortion metrics. Ideally, most parameterization applications work best on zero distortion parameterizations, though most are tolerant to some amount of distortion, some being more tolerant to shear and others to stretch. In general, applications that depend on regular grids for sampling, such as different types of detail mapping and synthesis, as well as compression and regular resampling schemes (e.g. geometry images [Gu et al., 2002]), tend to perform better on stretch minimizing parameterizations, since stretch is directly related to under-sampling. In contrast, applications based on irregular sampling, such as remeshing [Desbrun et al., 2002], are very sensitive to shearing, but can handle quite significant stretch. When acceptable levels of shear or stretch are not attainable because a surface is too complex, the surface needs to be cut prior to parameterization in order to achieve acceptable distortion.

In addition to distortion, several other factors should be considered when choosing a parameterization method for an application at hand:

- *Free versus fixed boundary.* Many methods assume the boundary of the planar domain is pre-defined and convex. Fixed boundary methods typically use very simple formulations and are very fast. Such methods are well suited for some applications, for instance those that utilize a base mesh parameterization, see Section 5.1. Free-boundary techniques, which determine the boundary as part of the solution, are often slower, but typically introduce significantly less distortion.

- *Robustness.* Most applications of parameterization require it to be bijective. For some applications local bijectivity (no triangle flips) is sufficient while others require global bijectivity conditions (the boundary does not self-intersect). Only a subset of the parameterization methods can guarantee local or global bijectivity. Some of the others can guarantee bijectivity if the input meshes satisfy specific conditions.

- *Numerical Complexity.* The existing methods can be roughly classified according to the optimization mechanism they use into linear and non-linear methods. Linear methods are typically significantly faster and simpler to implement. However, as expected the simplicity usually comes at the cost of increased distortion.

Method	Distortion minimized	Boundary	Bijectivity	Complexity
Uniform [Tutte 1963]	None	Fixed, convex	Yes	Linear
Harmonic [Eck 1995]	Angles	Fixed, convex	No	Linear
Shape preserving [Floater 1997]	Angles	Fixed, convex	Yes	Linear
Mean-value [Floater 2003]	Angles	Fixed, convex	Yes	Linear
LSCM/DCP [Lévy et al. 2002, Desbrun et al. 2002]	Angles (&Area)	Free	No	Linear
ABF/ABF++ [Sheffer and de Sturler 2000, Sheffer et al. 2005]	Angles	Free	Local (no flips)	Non-linear
MIPS [Hormann and Greiner, 2000]	Angles	Free	Global	Non-linear
Circle Patterns [Kharevych et al. 2006]	Angles	Free	Local (no flips)	Non-linear (unique minimum)
Stretch minimizing [Sander et al. 2001]	Distances	Free	Global	Non-linear
MDS [Zigelman et al. 2002]	Distances	Free	No	Non-linear
[Degener et al. 2003]	Areas	Free	Yes	Non-linear

Table 1: Planar method summary.

Method	Cat Head (257 Δ)	Nefertiti (8071 Δ)	Cow (5804 Δ)	Camel (78K Δ)
Uniform [Tutte 1963]*	0.02 s.	0.23 s.	0.14 s.	2.91 s.
Harmonic [Eck et al. 1995]*	0.02 s.	0.26 s.	0.17 s.	3.21 s.
Mean-value [Floater 2003]*	0.02 s.	0.25 s.	0.16 s.	3.19 s.
LSCM [Lévy et al. 2002]*	0.03 s.	0.38 s.	0.20 s.	5.28 s.
ABF++ [Sheffer et al. 2005]*	0.06 s.	1.87 s.	0.77 s.	36.31 s.
MIPS [Hormann and Greiner, 2000]‡	1 s.	8 s.	5 s.	83 s.
Circle Patterns [Kharevych et al. 2006]*	0.1 s.	3.6 s.	2.1 s.	76.7 s.
Stretch minimizing [Sander et al. 2001]	5.17 s.†	12 s.	8.5 s.	127 s.

†1.2 GHz Pentium M *3.0 GHz Pentium 4 ‡3.2 GHz Pentium 4

Table 2. Timings of various parameterization techniques.

Table 1 summarizes the more commonly used methods in terms of these properties. Table 2 lists the runtimes for some of the more commonly used methods. We used the Ray et al. [2003] 3D modeling system to time the fixed boundary methods, LSCM and ABF++. For the other methods the timings were provided by the authors. As expected, linear techniques are about one order of magnitude faster than the non-linear ones. Nevertheless, even the non-linear methods are fairly fast taking less than two minutes to process average size models. Figures 5.1 to 5.3 show some typical parameterization results.

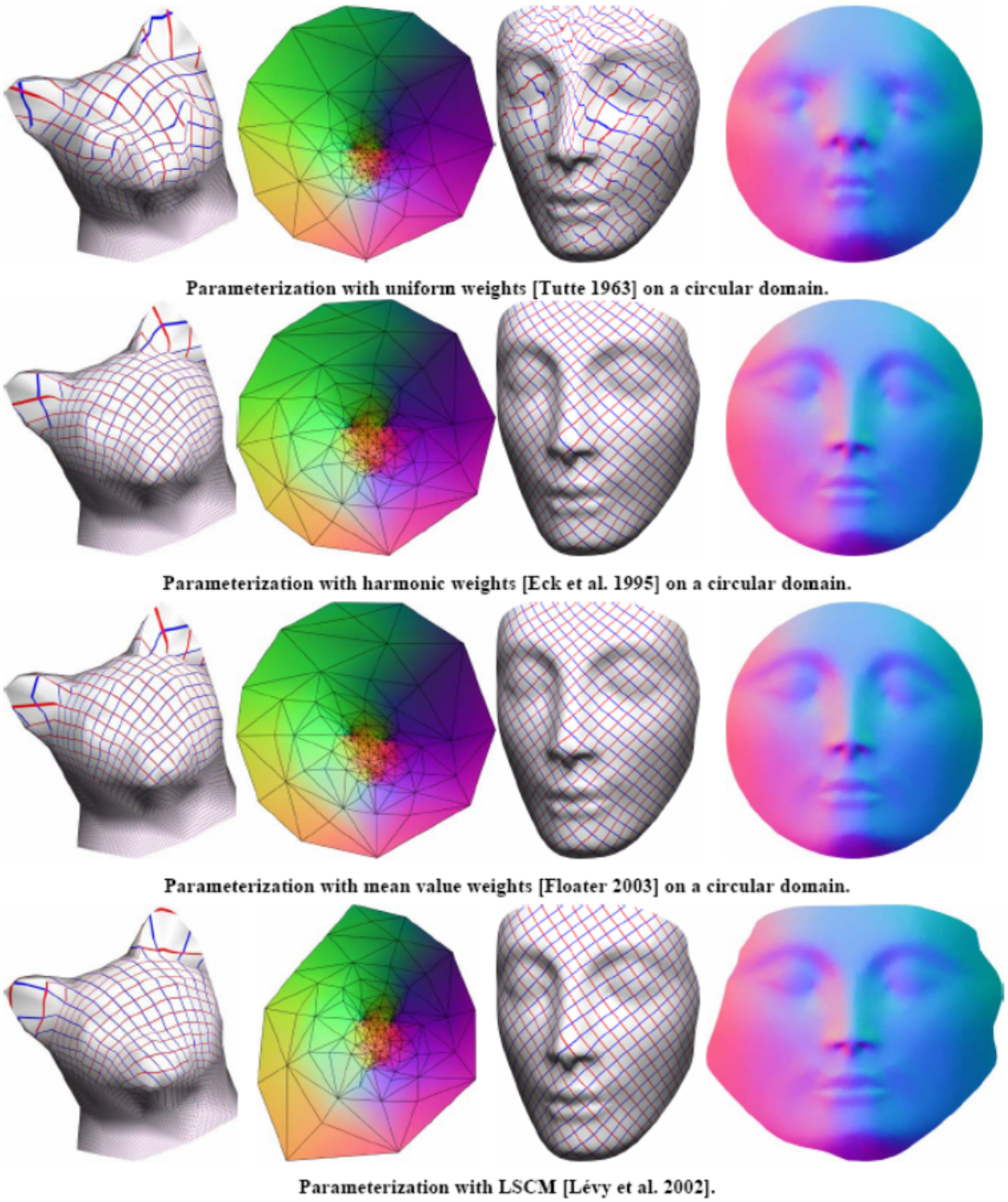
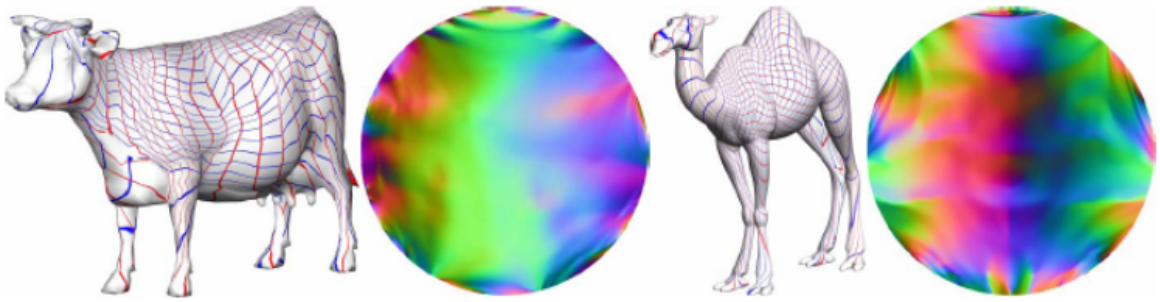
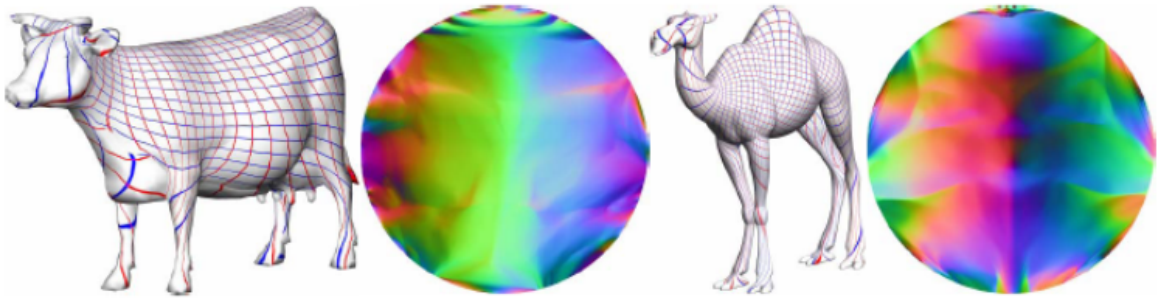


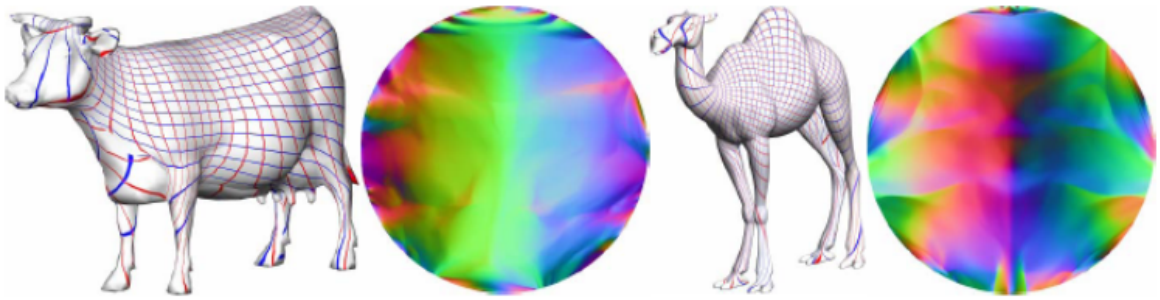
Figure 5.1: Surfaces with nearly convex boundaries parameterized with linear methods (images made with Graphite, http://alice.loria.fr/software).



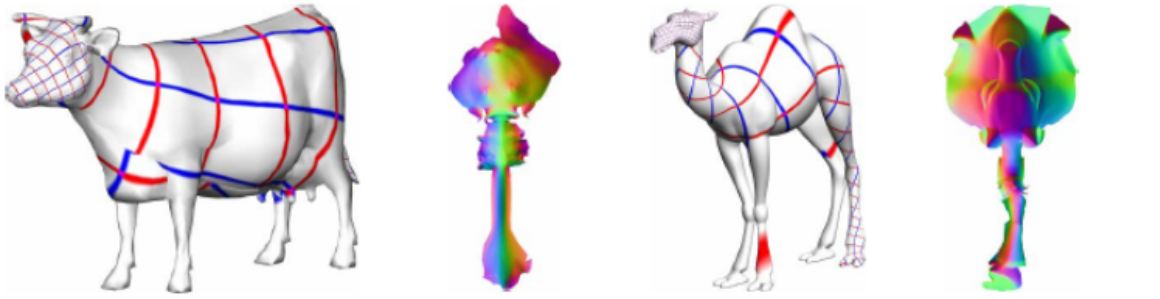
Parameterization with uniform weights [Tutte 1963] on a circular domain.



Parameterization with harmonic weights [Eck et al. 1995] on a circular domain.



Parameterization with mean value weights [Floater 2003] on a circular domain.



Parameterization with LSCM [Lévy et al. 2002].

Figure 5.2: Surfaces with non-convex boundaries parameterized with linear methods (images made with Graphite, http://alice.loria.fr/software).

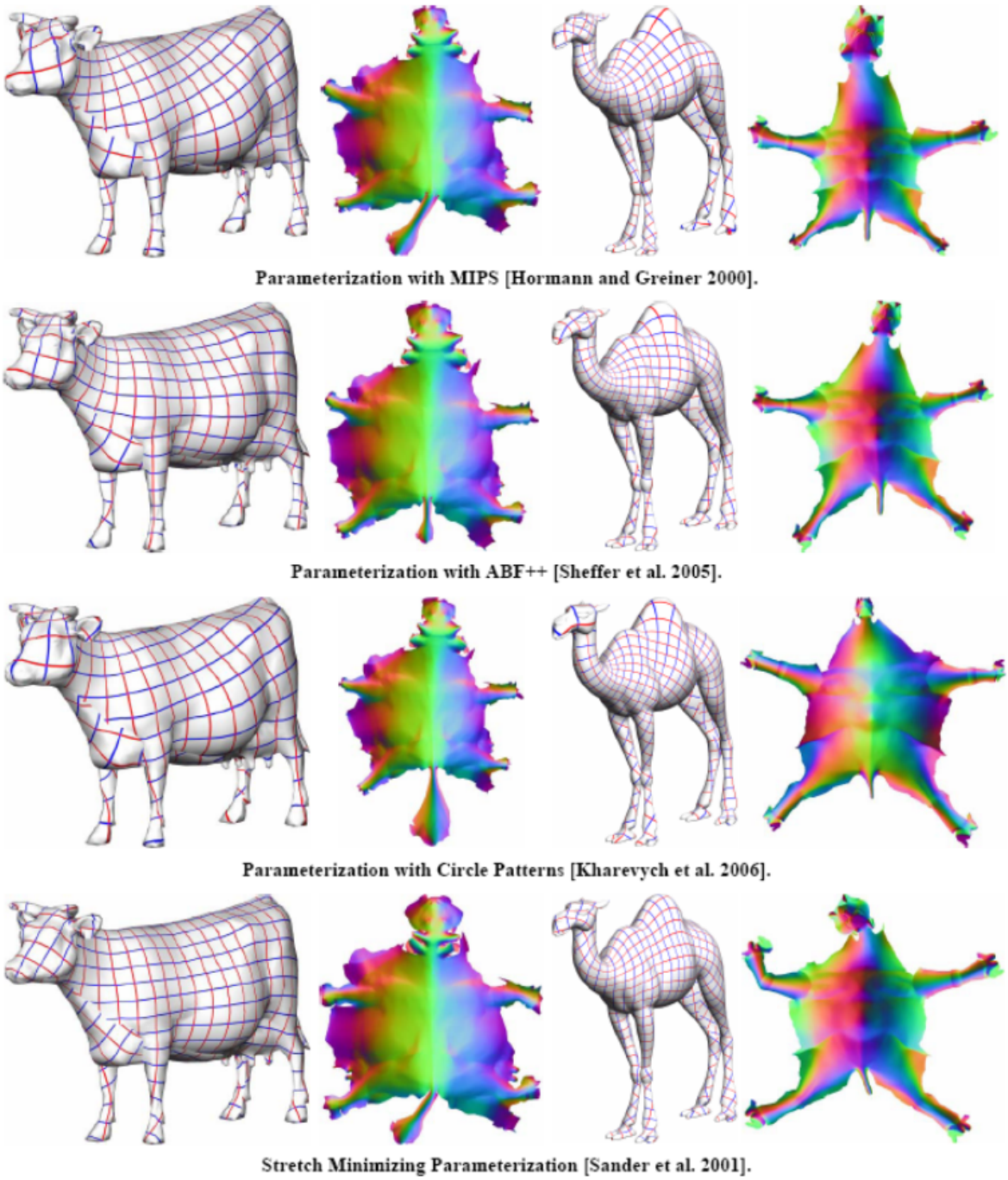


Figure 5.3: Surfaces with non-convex boundaries parameterized with non-linear methods (images made with Graphite, http://alice.loria.fr/software).

Chapter 6

Non-Planar Domains

Some applications are quite sensitive to discontinuities in the parameterization, or cannot tolerate them at all. In such cases, when the object to be parameterized is not a topological disc, it is worthwhile to use a different base domain for the parameterization. Examples of such domains that have been investigated include spheres, simplicial complexes, and periodic planar regions with transition curves.

In addition, numerous applications of parameterization require cross-parameterization or intersurface mapping between multiple models; see Chapter 7. Pairwise mapping between models can be used for the transfer of different properties between the models, including straightforward ones, such as texture, and less obvious ones such as deformation and animation. It can also be used for blending and morphing, as well as mesh completion and repair. The most common approach for pair-wise mapping is to parameterize both models on a common base domain. Free-boundary planar parameterization is clearly unsuitable for this purpose. Instead alternate domains such as a sphere or a simplicial complex are commonly used.

6.1 The Unit Sphere

The big advantage of the spherical domain over the planar one is that it allows for seamless, continuous parameterization of genus-0 models, and there are a large number of such models in use. Thus, the spherical domain has received much attention in the last few years, with several papers published about this topic. Some rigorous theory is being developed, getting close to the level of understanding we have of planar parameterizations. These notes cover four main types of spherical parameterization approaches: Gauss-Seidel iterative extension of planar barycentric methods; stereographic projection; spherical generalization of barycentric coordinates; and multi-resolution embedding.

One attractive approach for spherical parameterization is to extend the barycentric, convex boundary planar methods to the sphere. Several methods [Alexa, 2000; Gu and Yau, 2002; Kobbelt et al., 1999] used Gauss-Seidel iterations to obtain such parameterization. They start by computing an initial guess and then moving the vertices one at a time, first computing a 3D position for the vertex using a barycentric formulation [Eck et al., 1995], and then projecting the vertex to the unit sphere. Isenburg et al. [2001]

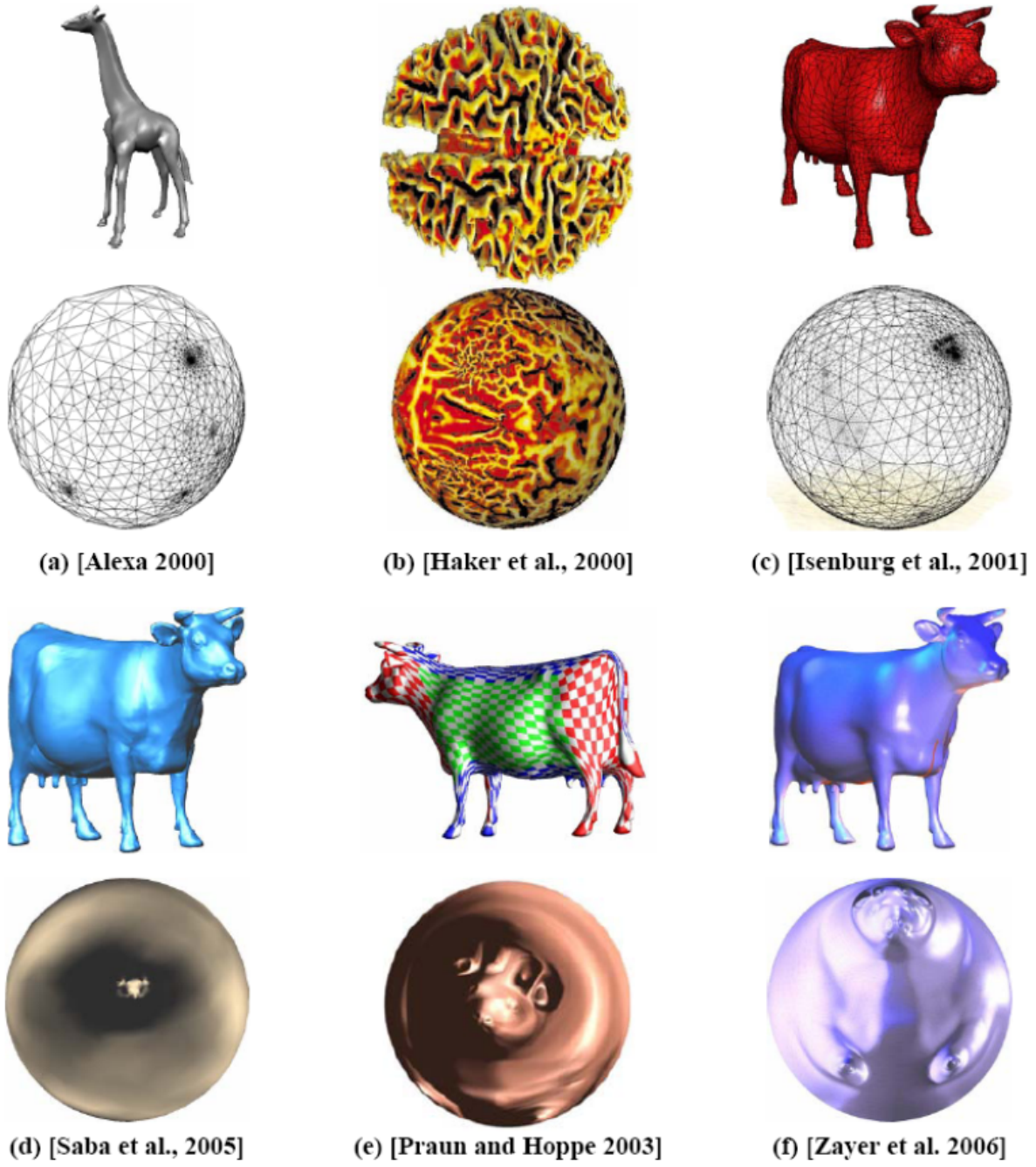


Figure 6.1: Spherical Parameterization.

split the mesh in two, map the cut onto a great circle and embed each half-mesh onto a hemisphere using a modified Tutte procedure. Regrettably, as proven by Saba et al. [2005] projected Gauss-Seidel iterations decrease the residual for only a finite number of iterations. As the result approaches a bijective solution, the scheme ultimately becomes unstable, the residual increases, and the system collapses to a degenerate solution. Saba et al. [2005] note that this behaviour is independent of step size.

Haker et al. [2000] compute a planar parameterization of the mesh first, using one of the triangles as a boundary. They then use the stereographic projection to obtain the spherical mapping. The result depends quite heavily on the choice of the boundary triangle. This approach works quite well in practice, however it doesn't offer any theoretical guarantees since the stereographic projection is bijective only for the continuous case, and can produce triangle flips in the discrete case. A simple proof by example of this statement can be obtained by imagining the great circle supporting the edge AB of a mapped spherical triangle ABC. The (continuous) stereographic projection maps this great circle to a circle in the original plane. The third vertex C can be perturbed in the plane to cross from the interior to the exterior of the circle, without changing the triangle orientation. The spherical triangle ABC will flip however as a result of this perturbation, as the image of C on the sphere will cross from one side to the other of the spherical edge AB.

Gotsman et al. [2003] showed how to correctly generalize the method of barycentric coordinates, with all its advantages, to the sphere. The generalization is based on results from spectral graph theory due to de Verdière [1993] and extensions due to Lovász and Schrijver [1999]. They provide a quadratic system of equations which is a spherical equivalent of the barycentric formulation. The authors do not provide an efficient way to solve the resulting system, and thus their method is limited to very small meshes. Saba et al. [2005] introduce a method for efficiently solving the system, by providing a good initial guess and using a robust solver. First, similar to [Isenburg et al., 2001], they partition the mesh in two, and embed each half on a hemisphere using a planar parameterization followed by a stereographic projection. They then use a numerical solution mechanism which combines Gauss-Seidel iteration with nonlinear minimization to obtain the final solution.

An efficient and bijective alternative is suggested by multi-resolution techniques. These methods obtain an initial guess by simplifying the model until it becomes a tetrahedron (or at least, convex), trivially embed it on the sphere, and then progressively add back the vertices [Shapiro and Tal, 1998; Praun and Hoppe, 2003]. Shapiro and Tal [1998] compute the embedding using purely topological operations and do not attempt to minimize any type of distortion. Praun and Hoppe [2003] obtain a spherical parameterization by alternating refining steps that add vertices from a multi-resolution decomposition of the object with relaxation of single vertex locations inside their neighbourhoods. The relaxation is aimed to minimize the stretch metric of the parameterization and is guaranteed to maintain a valid embedding.

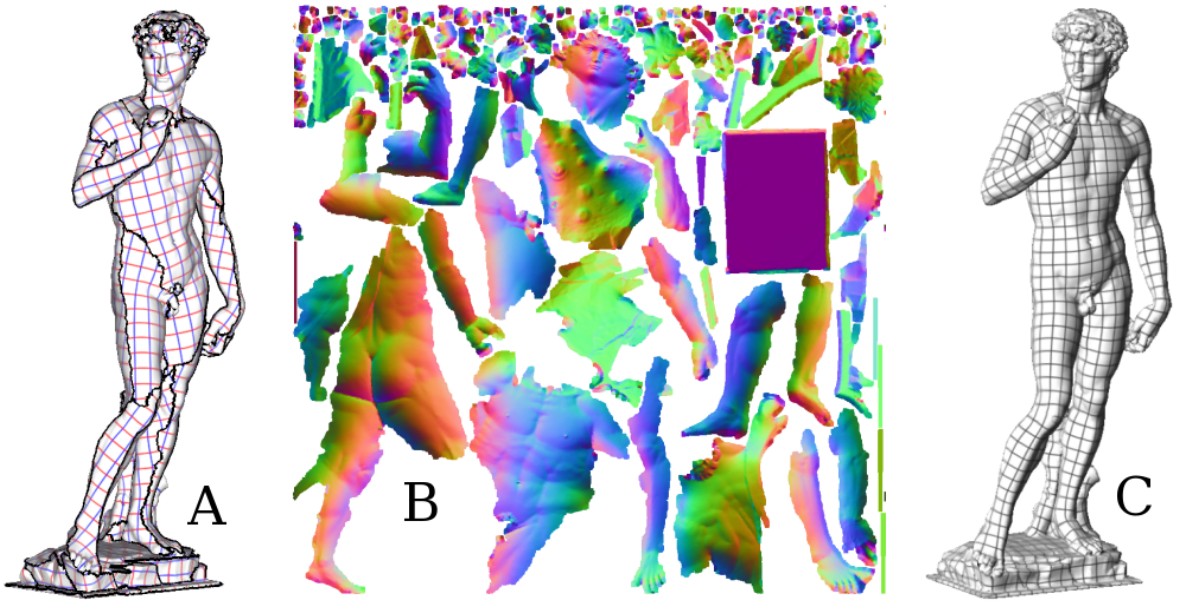


Figure 6.2: A,B: Parameterization methods for disk-topology combined with segmentation algorithms can create a texture atlas from a shape of arbitrary topology. However, the large number of discontinuities can be problematic for the applications. C: Global parameterization algorithms do not suffer from this problem. (Data courtesy of the Digital Michelangelo Project, Stanford).

6.2 Simplicial and Quadrilateral Complexes

As seen in Chapters 2 and 4, parameterization methods can put a 3D shape with disk topology in one-to-one correspondence with a 2D domain. For a shape with arbitrary topology, it is possible to decompose the shape into a set of charts, using a segmentation algorithm (e.g. VSA [Cohen-Steiner et al., 2004]). Each chart is then parameterized (see Figure 6.2-A,B). Even if this solution works, it is not completely satisfactory: why one should "damage" the surface just to define a coordinate system on it? From the application point of view, chart boundaries are difficult to handle in remeshing algorithms, and introduce artefacts in texture mapping applications. For this reason, we focus in this section on *global* parameterization algorithms, that do not require segmenting the surface (Figure 6.2-C).

To compute such a global parameterization, the geometry processing community first developed methods that operate by segmenting, parameterizing, and resampling the object. To our knowledge, this idea was first developed in the MAPS method [Lee et al., 1998] (Multiresolution Adaptive Parameterization of Surfaces). As shown in Figure 6.3, this method starts by partitioning the initial object (Figure 6.3-A) into a set of triangular charts, called the *base complex* (Figure 6.3-C). Then, a parameterization of each chart is computed, and the object is regularly resampled in parametric space (Figure 6.3-C). Further refinements of the method improved the inter-chart continuity [Kho-

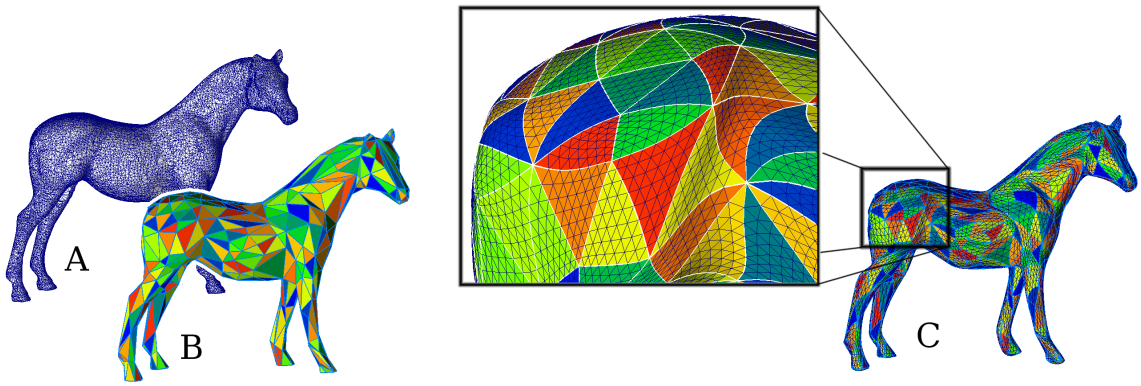


Figure 6.3: The MAPS method and its derivatives compute a global parameterization by decomposing the initial surface (A) into a set of triangular charts (B) and regularly re-samples the geometry in the parameter space of these charts (C).

dakovsky et al., 2003], formalized by the notion of *transition function*, explained further in this section. This representation facilitates defining hierarchical representations and implementing multiresolution processing tools on top of it [Guskov et al., 1999].

Historically, the most popular non-planar base domain has been a simplicial complex [Lee et al., 1998, 1999; Guskov et al., 2000, 2002; Lee et al., 2000; Praun et al., 2001; Khodakovsky et al., 2003; Purnomo et al., 2004; Schreiner et al., 2004; Kraevoy and Sheffer, 2004]. A simplicial complex can be considered as just the connectivity part of a traditional triangle mesh: the sets of vertices, edges, and faces. Most applications typically use simplicial complexes representing 2-manifolds with a boundary (an edge can only be adjacent to 1 or 2 faces) with a small number of elements. One method for obtaining such complexes is to simplify an original mesh. Once a suitable base mesh has been chosen, the original mesh is parameterized by assigning each of its vertices to a simplex of the base domain (vertex, edge, or face), along with barycentric coordinates inside it.

Early methods took a two-step approach to computing a parameterization; in the first step, elements of the fine mesh were assigned to faces of the base simplicial complex, while the second step would compute barycentric coordinates for these elements, usually using one of the fixed boundary parameterization methods discussed earlier. These steps could be repeated, but typically not mixed. More recent methods, such as [Khodakovsky et al., 2003], try to perform both steps at the same time.

6.2.1 Computing base complexes

To obtain the simplicial complex, [Eck et al., 1995] grow Voronoi regions of faces from seed points and then use the dual triangulation. The seed points are initially linked using shortest paths across mesh edges that provide the initial boundaries of the patches corresponding to base domain faces. To straighten each of these paths, the two adjacent patches are parameterized to a square. The path in question is then replaced with the diagonal of the square mapped onto the mesh surface.

Lee et al. [1998] simplify the original mesh, keeping track of correspondences between the original vertices and the faces of the simplified mesh. Others, like [Guskov et al., 2000] and [Khodakovsky et al., 2003] use clustering techniques to generate the patch connectivity and derive the base-mesh from it.

The construction becomes more challenging when multiple models need to be parameterized on the same complex [Praun et al., 2001; Schreiner et al., 2004; Kraevoy and Sheffer, 2004]. Praun et al. [2001] partition a mesh into triangular patches, which correspond to the faces of a user given simplicial complex, by drawing a network of paths between user-supplied feature vertices that correspond to the vertices of the base mesh.

Schreiner et al. [2004] and Kraevoy and Sheffer [2004] extend the methods of Praun et al. [2001] and Kraevoy et al. [2003] to construct the simplicial complex automatically, in parallel to the patch formation. The input to both methods includes a set of correspondences between feature vertices on the two input models. The methods use those as the vertices of the base complex. They simultaneously trace paths on the input meshes between corresponding pairs of vertices, splitting existing mesh edges if necessary. Tarini et al. [2004] were the first, to our knowledge, to use a quadrilateral base domain. Such a domain is much more suitable for quadrilateral remeshing of the input surface and for spline fitting. Tarini et al. [2004] generate the base domain manually.

6.2.2 Mapping to the base mesh

Once the discrete assignment to base domain faces has been done, the barycentric coordinates can be computed using fixed-boundary planar parameterization. Earlier methods computed the barycentric coordinates once, based on the initial assignment of the vertices to the base triangles. More recent methods [Khodakovsky et al., 2003; Kraevoy and Sheffer, 2004, 2005; Tarini et al., 2004] use an iterative process where vertices can be reassigned between base faces.

Khodakovsky et al. [2003] perform the vertex-to-patch assignment and coordinate relaxation in a single procedure, by letting vertices cross patch boundaries using transition functions. A transition function expresses the barycentric coordinates of a vertex with respect to a base domain face as barycentric coordinates for a neighboring base domain face. For this procedure only the images of the base domain vertices needs to be fixed, rather than the edges as well as in the previous methods. The authors relax the base domain vertices separately, prompting a new run of the main relaxation. In practice, this cycle is repeated only very few times. The implementation sometimes needs to discard some relaxation results when mesh vertices moved around base domain vertices end up with barycentric coordinates that are invalid for all the base domain faces around that vertex.

Tarini et al. [2004] and Kraevoy and Sheffer [2004, 2005] fix the boundary of a group of base mesh faces, update the barycentric coordinates in the interior, and then possibly re-assign some vertices to different faces inside the group. The methods differ in the grouping they use and the choice of parameterization technique used for the barycentric coordinates computation.

Chapter 7

Cross-Parameterizations and Constraints

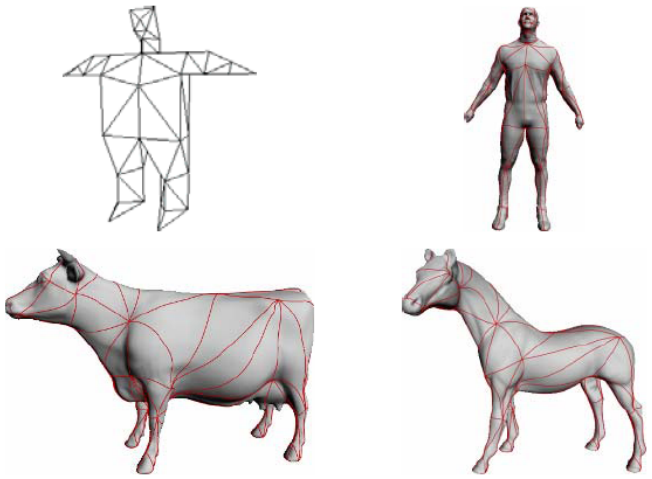
In a cross-parameterization or inter-surface-mapping setup the parameter domain for one mesh is another surface mesh. Cross-parameterization is used to morph or blend between meshes and to transfer properties between them. For morphing in addition to obtaining a mapping it is necessary to obtain a common compatible connectivity for the two meshes. The most common approach for pair-wise mapping is to parameterize both models on a common base domain. Free-boundary planar parameterization is clearly unsuitable for this purpose. Instead alternate domains such as a simplicial complex or a sphere are commonly used.

7.1 Base Complex Methods

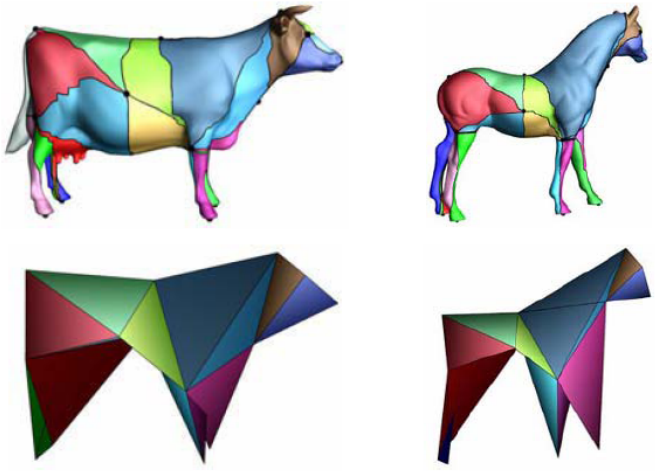
Many methods use mapping to a common base-complex to obtain a cross-parameterization. Since the base must be shared the construction is significantly more challenging than when parameterizing a single mesh. Praun et al. [2001] partition a mesh into triangular patches, which correspond to the faces of a user given simplicial complex, by drawing a network of paths between user-supplied feature vertices that correspond to the vertices of the base mesh.

Schreiner et al. [2004] and Kraevoy and Sheffer [2004] extend the methods of Praun et al. [2001] and Kraevoy et al. [2003] to construct the simplicial complex automatically, in parallel to the patch formation. The input to both methods includes a set of correspondences between feature vertices on the two input models. The methods use those as the vertices of the base complex. They simultaneously trace paths on the input meshes between corresponding pairs of vertices, splitting existing mesh edges if necessary.

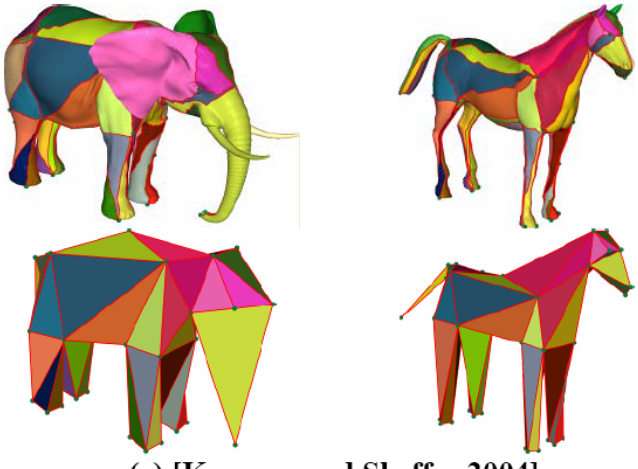
Once the base is created, the meshes can be mapped to the base using the techniques reviewed in the chapter on alternative domains. Schreiner et al. [2004] use an alternative approach. They never compute an explicit map between the full-resolution objects and the base domain. Instead, they alternate the role of base domain between the two meshes, at various complexity levels in a multi-resolution representation. They progressively refine each mesh by adding new vertices and relaxing their location using a stretch-based metric measured on a temporary planar unfolding of their neighborhoods.



(a) [Praun et al. 2001]



(b) [Schreiner et al. 2004];



(c) [Kraevoy and Sheffer 2004]

Figure 7.1: Base complex construction.

7.2 Energy Driven Methods

Several methods use an energy-driven approach for cross-parameterization, where one mesh is directly attracted towards the other [Allen et al., 2003; Sumner and Popović, 2004]. The attraction energy function consists of components that pull the vertices of one mesh towards nearest locations on the other while trying to preserve the shape of the mesh as much as possible. To facilitate correspondence they require the user to specify several dozen point-to-point correspondences between the input models. The methods work well when the meshes are very similar, e.g. humans in the same pose [Allen et al., 2003] but tend to converge to a poor local minimum with increase in shape difference. These methods are quite sensitive to the weights used inside the energy functional to account for the different components. One advantage of these approaches is that they can find mappings between models of different topology (genus, etc.), though these maps are no longer bijective.

7.3 Compatible Remeshing

For applications such as morphing it is not enough to obtain a cross-parameterization between the two models. For these applications, at the end of the process the two models are typically required to have the same connectivity. There are three main approaches for generating such common connectivity.

- *Base mesh subdivision:* Several methods including [Praun et al., 2001] use the base mesh connectivity and refine it using the one-to-four subdivision pattern, introducing as many levels of subdivision as necessary to capture the geometry of both models. The advantage of the method is simplicity. It's drawback is the dependence on the shape of base mesh triangles. The method also tends to require large triangle count to achieve acceptable accuracy (roughly factor 10 compared to input mesh sizes).

- *Overlay:* Another approach for generating common connectivity [Alexa, 2000; Schreiner et al., 2004] is to intersect the two input meshes in the parameter domain, combining all their vertices and generating new vertices at edge-edge intersections. The method preserves exactly the input geometries but is not-trivial to implement and like subdivision increases the triangle count by roughly a factor of 10.

- *Remeshing:* In [Kraevoy and Sheffer, 2004] the authors propose an alternative where they use the connectivity of one of the input meshes as a basis and then refine it as necessary based on an approximation error with respect to the second mesh. The resulting meshes have significantly lower triangle count than using the other two approaches. The result heavily depends on which of the inputs is selected as the source for common connectivity. Unlike in the overlay approach, some fine features of the second mesh are only approximated and it is difficult to faithfully reproduce sharp features.

Figure 7.2: Two examples of constrained texture mapping [Lévy, 2001]

(a) (b) (c)

Figure 7.3: Constraint enforcement with Matchmaker [Kraevoy et al., 2003]: (a) Texture (two photos) with feature points specified; (b) input model with corresponding vertices highlighted; (c) resulting texture.

7.4 Constraints

Sometimes a parameterization needs to accommodate user constraints, specifying correspondences between vertices of the mesh. The most important application of constrained parameterization is texture mapping 3D models from photographs [Lévy, 2001; Kraevoy et al., 2003] (see Figures 7.2 and 7.3). Zhou et al. [2005] use more complex constraints to allow the user to combine several images to produce a complete texture for a mesh. Constrained parameterization can also be used to hide cross-seam discontinuities [Kraevoy et al., 2003; Zhou et al., 2005].

The methods for enforcing constraints can be split into two types, those that enforce soft or approximate constraints and that that enforce hard constraints. Methods based on energy minimization can accommodate soft constraints by adding a quadratic term to the energy function, measuring the distance between the constraint features in the

current configuration and their desired location [Lévy, 2001]. Such constraints work reasonably well in practice, and can be solved efficiently, since they only add linear terms to the energy, but sometimes break theoretical guarantees about the original parameterization method, such as bijectivity. The degree of constraint approximation typically decreases with the distance between the constrained and unconstrained vertex locations. Some applications, such as hiding texture discontinuities along seams in the parameterization [Kraevoy et al., 2003; Zhou et al., 2005] require hard constraints to achieve perfect alignment of the texture along the seams.

One way to enforce hard constraints is by adding them into a regular parameterization formulation using Lagrange multipliers [Desbrun et al., 2002]. This approach allows constraints to be defined on points inside triangles and on arbitrary line segments (the vertices of the triangulation can be constrained more easily by taking the corresponding variables out of the system). However, it is easy to show that for a given mesh connectivity not every set of constraints can be satisfied. Thus methods like this that preserve the mesh connectivity will fail to generate bijective parameterizations for many inputs.

Methods that enforce hard-constraints robustly, introduce additional vertices into the mesh as they go along to ensure that a constrained solution exists. Eckstein et al. [2001] enforce hard constraints by deforming an existing embedding while adding new vertices when necessary. Theoretically, this method can handle large sets of constraints but is extremely complicated.

The Matchmaker algorithm [Kraevoy et al., 2003] compute the parameterization by establishing coarse patch correspondences between the input and the parameter domain. The provided feature points on the input model and the parameter domain are connected using a network of curves that partition the surface into patches that are then parameterized while trying to maintain continuity and smoothness between them. The curve tracing process is guided by a set of topological rules that ensure that the resulting patches will be consistent between the objects being parameterized and the domain. They compute the triangulations of the input and the parameter domain simultaneously. Continuity and smoothness between patches can be obtained by relaxing the parameterization.

Zhou et al. [2005] allow the user to combine several images to produce texture for a mesh, by assigning some surface patches to different images, as well as using in-painting techniques to create texture for any unassigned transition patches between them. In addition to geometric smoothness of the map, they take into account the continuity of the texture signal being applied since it may come from different sources for two neighboring patches.

Chapter 8

Global Parameterizations and Cone Points

The parameterization algorithms presented in previous sections mainly focused on mapping the surface continuously onto the parameter domain. On simply connected surfaces, the parameterization can then be visualized by showing the pre-image of a quadrilateral grid on the surface. On higher genus surfaces, one first has to cut the surface open resulting in discontinuous gaps of the grid lines across the cuttings.

We will now focus on parameterizations, whose grid lines form a global closed quadrilateral grid. The QuadCover algorithm [Kälberer et al., 2007] produces parameterizations of this kind. Furthermore, the alignment of the parameter lines can be driven by user given user input (e.g. align to principle curvature directions).

8.1 Overview

This chapter focuses on the automatic construction of a global surface parameterization. The parameterization is guided by a so-called frame field, which can be seen as a collection of four vectors in each point (for example, the field of principal curvature directions).

Starting from a given frame field, the algorithm first constructs a locally integrable frame field. Second, this field is integrated yielding a parameterization. Third, the

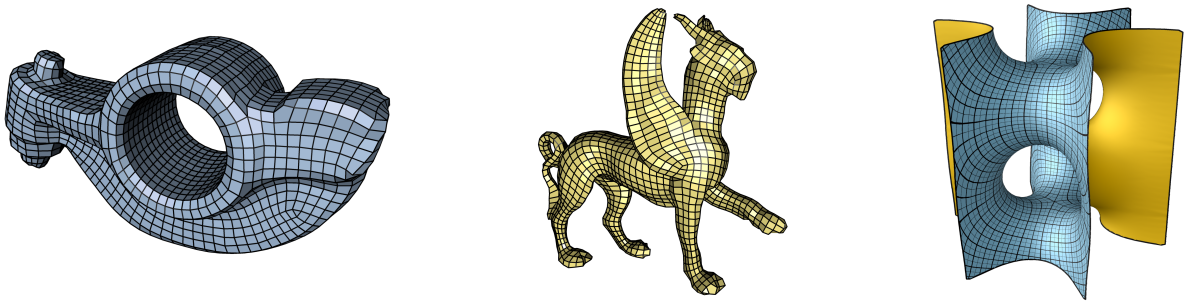


Figure 8.1: The QuadCover algorithm generates high quality parameter lines on simplicial surfaces. The automatic parameterization is guided by a user-given frame field, such as principal curvature directions, and is well suited for regular quadrilateral remeshing.

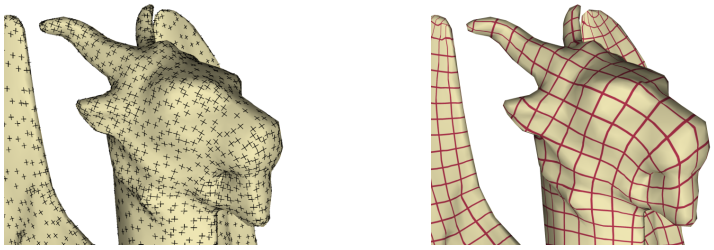


Figure 8.2: Left: Frame field. Right: Corresponding parameterization.

parameterization gets adapted such that the grid lines are connected across the cuttings.

Special issues arise, when the frame field has singularities. In QuadCover, they are resolved by using branched covering spaces. The frame field naturally simplifies to a single vector field on the covering, and then standard Hodge decomposition techniques are used to assure global integrability. The details are explained in Section 8.2.2.

Given a global parameterization on any simplicial surface there is a wealth of future applications. From texture mapping to extension of image processing algorithms, from remeshing to the automatic construction of hierarchical subdivision surfaces, all applications using natural coordinates will benefit from the added structure of a global parameterization.

8.1.1 Algorithms for global parameterization

The research area of surface parameterization has a long and fruitful tradition. There already exists a wealth of different approaches to surface parameterization and, more general, the generation of quad and quad dominant meshes from given triangle meshes.

The method of Boier-Martin et al. [2004] clusters the surface into macropatches and parameterizes each surface patch. Kharevych et al. [2006] find a conformal parameterization via circle patterns. In contrast to Gu and Yau, they use cone-singularities to increase the flexibility of purely conformal mappings. Dong et al. [2006] compute the Morse-Smale complex of eigenfunctions of the mesh Laplacian to compute a patch layout. The nodes of the complex are then utilized similarly to the cone singularities in [Kharevych et al., 2006].

Early approaches for quadrangular remeshing guided by principal curvature directions are from Alliez et al. [2003]. They were extended by Marinov and Kobbelt [2004], and base on the integration of curvature lines on the surface. Dong et al. [2005b] presented an algorithm which traces isolines in two conjugate harmonic vector fields. Marinov and Kobbelt [2006] focus on creating coarse quad-dominant meshes by approximating the surface with very few patches, which are then individually subdivided into quads.

Tong et al. [2006] use harmonic one-forms for surface parameterization. They enlarge the space of harmonic one-forms by allowing additional singular points on the surface. The extended cut graph increases the homology group and thus the space of harmonic one-forms on the surface. As a consequence, the user-defined choice of placing the

singular points and the cut graph allows a controlled modeling of the harmonic one-forms. Still, the approach is constrained by the global nature of harmonic one-forms, in some sense, similar to the algorithm of Gu and Yau [2003].

Ray et al. [2006] parameterize surfaces of arbitrary genus with periodic potential functions guided by two orthogonal input vector fields. This leads to a continuous parameterization except in the vicinity of singular points on the surface. These singular regions are detected and reparameterized afterwards. Our approach was strongly inspired by their work.

8.1.2 QuadCover algorithm

QuadCover is an algorithm to compute a global continuous parameterization for an arbitrary given simplicial 2-manifold. The algorithm runs automatically and the parameter lines align optimally with a user-defined frame field, for example, the principal curvature directions.

In a first stage of the algorithm, the frame field is slightly changed, such that it turns into a locally integrable field. After cutting the surface open, the frame field is integrated yielding a parameterization, which is discontinuous at the cuttings.

In the second part of QuadCover, the frame field is once more altered, such that the parameter lines close globally, even at the cuttings. The details are explained in the following sections.

8.2 Setting

In this section the underlying concepts of the QuadCover algorithm are introduced.

8.2.1 Matchings

Smooth setting. Given a smooth 2-manifold M with charts $U_i \subset M$. A parameterization maps all charts into the parameter domains $f_i \colon U_i \to \Omega_i \subset \mathbb{R}^2$. One can now visualize the parameter grid on M as the pre-image under f_i of the unit grid lines $\mathbb{N} \times \mathbb{R}$ and $\mathbb{R} \times \mathbb{N}$. A **globally continuous parameterization** consists of a set of charts U_i with parameter functions f_i for which the parameter lines coincide in all regions where two charts U_i, U_j overlap.

The transition functions between adjacent charts of a global parameterization satisfy two conditions, see also [Ray et al., 2006]:

First, the gradients of the parameterization functions have to agree up to a cyclic permutation of the vectors $(\nabla_u f_i, \nabla_v f_i, -\nabla_u f_i, -\nabla_v f_i)$, because u- and v-lines should not be distinguished on the parameterized surface. Thus, the Jacobians of the charts are related by

$$Df_i(p) = J^{r_{ij}} Df_j(p), \qquad J := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad p \in U_i \cap U_j \tag{8.1}$$

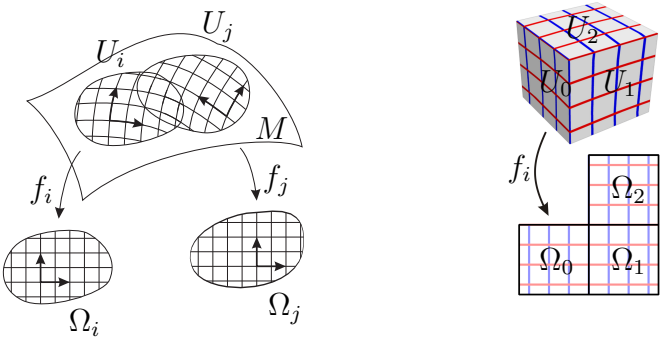


Figure 8.3: Parameterization of two overlapping charts with a matching of $r_{ij} = 3$ (left), and of a cube with matchings $r_{01} = r_{12} = 0$, $r_{20} = 1$ (right).

with a constant integer $r_{ij} \in \{0, 1, 2, 3\}$ on the intersection $\Omega_i \cap \Omega_j$. We call the values r_{ij} **matchings** between charts U_i and U_j. See Figure 8.3 for nontrivial matchings.

Second, the parameter values may differ only by integer values in the u and v coordinate, since the unit grid is invariant under translations by integer values.

The transition functions $f_j \circ f_i^{-1}$ of a parameterization fulfilling the two conditions above are automorphisms of the unit grid. We call a linear function $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ which meets

$$f(z) = J^r z + t, \qquad r \in \mathbb{N}, \quad t \in \mathbb{N}^2, \quad z \in \mathbb{R}^2 \tag{8.2}$$

a **grid automorphism**.

Discretization. Each triangle of the mesh is considered as a chart. The transition function between two adjacent triangles is fully determined by the matching and the translation vector associated to their common edge (see Equation 8.2).

8.2.2 Frame fields

A parameterization f can be characterized by its derivatives. In each chart, the gradient fields $\nabla_u f$ and $\nabla_v f$ are continuous vector fields. At the transition between two charts U_i, U_j, the vectors $\nabla_u f$, $\nabla_v f$, $-\nabla_u f$ and $-\nabla_v f$ interchange cyclically depending on the matching number.

Thus, we cannot describe the derivatives of f as globally defined vector fields, but have to introduce **frame fields** which are invariant under cyclic permutations (see Figure 8.4 for examples).

Definition 1. *Given a manifold M with charts U_i and matchings r. A **frame field** on M is a collection of four vector fields $X_{i,0}$, $X_{i,1}$, $X_{i,2}$, $X_{i,3}$, in each chart U_i which satisfy in all overlapping charts $U_i \cap U_j$:*

$$X_{j,k} = X_{i,(k-r_{ij}) \bmod 4}, \qquad k \in \{0, 1, 2, 3\}.$$

*This means that the vectors $X_{j,k}$ are cyclically permuted to $X_{i,k}$ by a shift of $-r_{ij}$. If $X_{i,2} = -X_{i,0}$, $X_{i,3} = -X_{i,1}$ in all domains Ω_i, the frame field is called **symmetric**.*

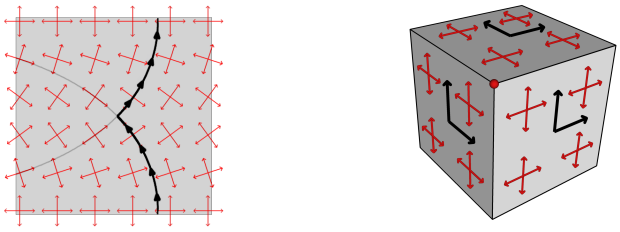


Figure 8.4: A set of four vectors given in each point which cannot be described with global vector fields. Left: On a flat torus an integral line meets itself perpendicularly. Right: Around the vertex of a cube exists no match of vectors.

In the example of Figure 8.3 (left), $X_{i,1}$ coincides with $X_{j,0}$, corresponding to $r_{ij} = 3$. Figure 8.4 shows examples of a frame field on a flat torus and a cube, which cannot be expressed in terms of 4 global vector fields.

A parameterization f can be represented in differential form by a global frame field on M using the following two vector fields in each chart U_i:

$$X_{i,0} := (Df_i)^{-1}(e_1), \qquad X_{i,1} := (Df_i)^{-1}(e_2) \tag{8.3}$$

with the unit vectors e_1, e_2 in $\mathbb{R}^2$. With $e_3 := -e_1$, $e_4 := -e_2$ and $X_{i,2} := (Df_i)^{-1}(e_3)$, $X_{i,3} := (Df_i)^{-1}(e_4)$, we obtain indeed a frame field: for $k \in \{0, 1, 2, 3\}$

$$\begin{aligned} X_{j,k} &= (Df_j)^{-1}(e_k) = (D(J^{r_{ij}} f_i))^{-1}(e_k) \\ &= (Df_i)^{-1}(J^{-r_{ij}} e_k) = X_{i,(k-r_{ij}) \bmod 4} \end{aligned} \tag{8.4}$$

Discretization. The frame fields are discretized to be constant on each triangle. For symmetric frame fields, only two of the four vectors are stored. Together with the matchings, this defines discrete symmetric frame fields uniquely.

Definition 2. *A **matching** on a discrete manifold M is a map*

$$r \colon \{\text{edges } e_{ij} \mid T_i \cap T_j = e_{ij}\} \to \{0, 1, 2, 3\},$$

which determines the matching r_{ij} for two adjacent triangles T_i and T_j. We denote the space of all those maps by R_M.

8.2.3 Branched covering spaces

The QuadCover algorithm uses the notion of branched covering surfaces for an equivalent description of frame fields. A frame field on the input surface can be seen as a vector field on a covering surface. The advantage of this notion is, that standard vector field calculus can now be applied to frame fields.

First, recall some definitions about Riemann surfaces, see [Farkas and Kra, 1992], [Fulton, 1995], [Jost, 2002].

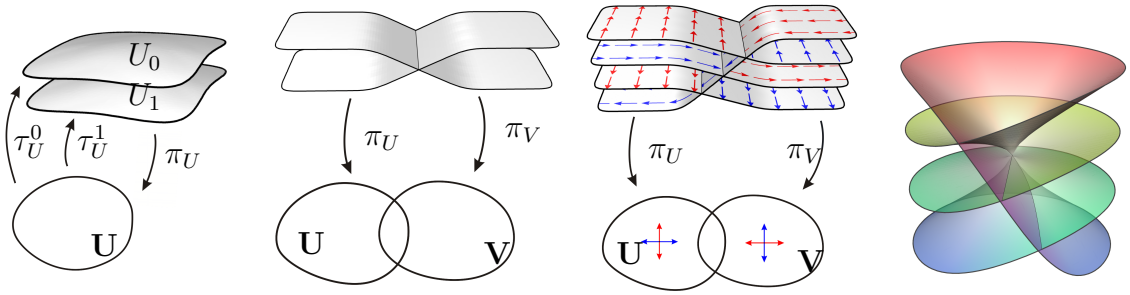


Figure 8.5: From left to right: trivial covering; patching two coverings together; a frame field lifted to a vector field on the covering; branch point.

Definition 3. *Let M be a Riemann surface. An n-sheeted covering M' of M is a Riemann surface with a local homeomorphism $\pi\colon M' \to M$, such that there exists a disk U_p of each point $p \in M$, whose pre-image $\pi^{-1}(U_p)$ is the union of exactly n pairwise disjoint open sets (Figure 8.5, left, shows a 2-sheeted covering).*

In our setting, we allow some exceptional points p (branch points), where the pre-image of a neighbourhood of p is the union of less than n pairwise disjoint open sets (Figure 8.5, right).

Definition 4. *A **trivial n-sheeted covering** of a map $U_i \subset M$ is an n-sheeted covering U_i' which consists of n disjoint components which are mapped by π homeomorphically onto U_i.*

*The components $\pi^{-1}(U)$ of a trivial covering are called **layers**. They will be denoted by $U^l \subset U'$, $l \in \{0, \dots, \#\text{layers} - 1\}$. Let $\tau_U^l = (\pi_{|U^l})^{-1}\colon U \to U^l$ be the inverse of the projection operator in the given layer l.*

Here we consider coverings whose metric is induced from the surface M by π^{-1}. Thus, an n-sheeted trivial covering of M can be seen as just n copies of M, cf. Figure 8.5, left.

One way of constructing a covering on M is to take trivial coverings of all charts and glue them together at their intersection: For each two charts U_i, U_j of M with $U_i \cap U_j \neq \emptyset$, let $\varrho\colon \pi_i^{-1}(U_i \cap U_j) \to \pi_j^{-1}(U_i \cap U_j)$ be an isomorphism. The patches can then be merged together by identifying the corresponding points of the two coverings, cf. Figure 8.5, second image.

The following construction shows, how the matchings r_{ij} of a manifold M canonically induce a covering of M. We restrict to 4-sheeted coverings as they naturally appear in the study of frame fields.

Definition 5. *Let (U_i', π_i) be 4-sheeted trivial coverings of the charts U_i. For two overlapping charts U_i, U_j, the map $\varrho\colon \pi_i^{-1}(U_i \cap U_j) \to \pi_j^{-1}(U_i \cap U_j)$, which maps a point $p \in \tau_{U_i}^l(U_i \cap U_j)$ from layer l to $\tau_{U_j}^{(r_{ij}+l) \bmod 4} \circ \pi_U(p)$ into layer $(r_{ij} + l) \bmod 4$, is an isomorphism. Thus, by identifying the two layers with ϱ, the trivial coverings of the charts can be glued together (Figure 8.5, second image). We call the resulting covering M' of the **covering induced by** r.*

The covering M' induced by the matchings has no branch points. To enlarge the space of possible frame fields on the manifold, the covering surface may possess single points $p \in M$, which have less than 4 pre-images. One can get these points by excluding them from the surface (see Figure 8.5, right). See below for the construction of branch points in the discrete setting.

Discretization. In the discrete setting, branch points are located at vertices. On a 4-sheeted covering they occur when the (oriented) sum of all matchings of outgoing edges differs from 0 (modulo 4). This means starting somewhere in the neighborhood of v and walking around the vertex, ends on a different layer in the covering than the start point. The following notion is used for describing different types of branch points.

Definition 6. *Let v be a vertex of M, $T_0, \dots, T_{n-1}$ the triangles incident to v in counter-clockwise order, $T_n = T_0$ and $r_{i,i+1}$ the matching at the edge between adjacent triangles. The **layer shift** around v is then given by:*

$$ls(v) := \Big(\sum_{i=0}^{n} r_{i,i+1} \Big) \bmod 4 \tag{8.5}$$

8.2.4 Vector fields on covering spaces

In this section, we show how frame fields can be described with vector fields on a covering surface. This result allows us to apply the classical theory for vector fields to frame fields.

A frame field $X_{i,k}$ on M with matchings r canonically lifts to a vector field X on the covering: in each chart U_i, lift $X_{i,k}$ to a vector field X on a trivial 4-sheeted covering U_i' of U_i as follows: For $p \in \tau_U^l(U)$, set $X(p) = X_{i,l}(\pi_i(p))$ (Figure 8.5, third image). The result is a globally defined vector field X on the covering M' induced by the matchings r.

When the coverings of the charts are patched together as described in Definition 5, X becomes a well defined vector field on M', since the layers of the covering are connected in the same way as the vector fields permute when another chart is chosen.

Definition 7. *Let M be a manifold with matchings r and M' the induced covering. A frame field lifted to a vector field X on M' is called a **covering field** of M.*

8.3 QuadCover Algorithm

Computing the potential function. Given a surface M together with a frame field $X_{i,j}$ on M. Equivalently, given a covering surface M' with a vector field X on M'. The parameterization is a scalar function $f' \colon M' \to \mathbb{R}$. It can be projected back to a parameter function $f \colon M \to \mathbb{R}^2$ by taking the values of f' on the first two layers.

The parameterization should align with the given input field as well as possible, i.e., f' should minimize the energy

$$E(f) = \int_{M'} \|\nabla f - K\|^2 dA. \tag{8.6}$$

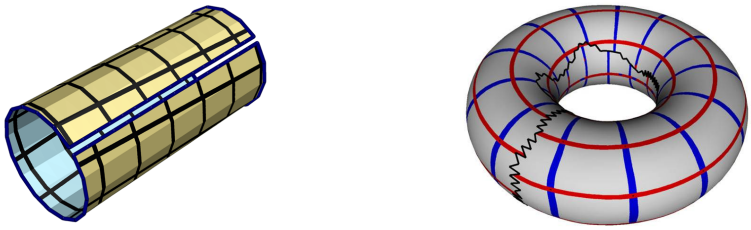


Figure 8.6: Parameterization after first stage of the QuadCover algorithm. The grid lines are discontinuous across the cuttings.

Recall, that K is a symmetric covering field and the covering surface is symmetric due to a cyclic permutation of the layers by 2. Since the energy has a unique minimum, the solution f' is also symmetric.

This optimization problem can also be formulated using covering fields instead of the scalar function f'. The Hodge-Helmholtz decomposition of a vector field K on a manifold M' is a unique description of K as

$$K = P_K + C_K + H_K \tag{8.7}$$

with a gradient field P_K, a cogradient field C_K and a harmonic field H_K. Discarding the second term leads to a curl free field $\tilde{X} := P_K + H_K$ whose integral is the minimizer of Energy (8.6). For details on the Hodge decomposition of discrete vector fields see [Polthier and Preuss, 2003].

So far, the parameterization algorithm outlines as follows:

1. Perform Hodge-Helmholtz decomposition of input field K.
2. Discard second term and obtain a locally integrable field.
3. Cut the surface open to be simply connected and lift the cut graph to the covering, such that the covering gets cut into 4 connected pieces.
4. Obtain the parameterization f' by integration.

8.3.1 Global continuity

If we take the solution f' from the previous paragraph as parameterization map, the parameter lines would not be continuous everywhere on the surface. They may have a mismatch at the cut graph G (see Figure 8.6).

Definition 8. *Given a cut graph G on M. Let $\{\gamma_i\}$ be a set of **cutting paths** with $\cup_i \gamma_i = G$. Each cutting path is closed or connect a boundary or branchpoint with another boundary or branch point.*

For all paths γ_i and each point $p \in \gamma_i$, one can measure the **gap** (discontinuous jump) as the difference of function values on the right and left side of the path.

The parameterization can now be "repaired" such that the grid lines match up. This is the case iff all gaps are integer values. The repairing algorithm bases on the following observation: Along each path γ_i, the gap is always a constant d_i, since the derivative of the function is locally integrable. Note, that there is an exception if two paths γ_i, γ_j merge and run on top of each other. In this case, the gap turns into $d_i + d_j$. For further details, see [Kälberer et al., 2007].

Thus, the grid lines are globally continuous if and only if all $d_i \in \mathbb{Z}$. In order to adapt f to fulfill the global continuity condition, we add another scalar covering function ψ to f' such that $\tilde{f}' := f' + \psi$ satisfies $\tilde{d}'_j \in \mathbb{Z}$ (where $\tilde{d}'_j$ are the gaps of $\tilde{f}'$).

The remaining problem is to find this scalar function, with given gaps. In order to distort the initial parameterization as few as possible, ψ is taken to be a harmonic function, because they are the smallest functions with given gaps in L^2 norm. ψ is found via minimizing the Dirichlet energy. For the exact algorithm, refer to [Kälberer et al., 2007].

The second stage of QuadCover has the following outline:

1. Compute cutting paths γ_i.
2. Measure gaps d_i.
3. Find harmonic map ψ with gaps $\text{round}(d_i) - d_i$.
4. Add ψ to f'.

8.4 Resulting Parameterizations of QuadCover

The algorithm delivers parameterizations on a wide range of models, as shown in Figures 8.1, 8.7, 8.8, 8.9, and 8.10. Comparisons to state-of-the-art show that QuadCover produces very competitive results, see Figure 8.10. As the figure illustrates, it roughly shares the curvature alignment of [Ray et al., 2006], but managed to drastically reduce the occurrence of irregular points. In contrast to the methods of [Tong et al., 2006] and [Dong et al., 2006], QuadCover is suited to handle arbitrary locations of branch points, as it does not restrict the branch points to be the corners of some coarse meta mesh.

Table 8.1 shows that QuadCover exhibits the smallest edge length variation, at the cost of higher angular deviation. Discarding the curvature alignment term during smoothing significantly reduces angular deviation, but generally, more wrinkles in the final quad mesh are introduced where parameter lines do not follow high curvature.

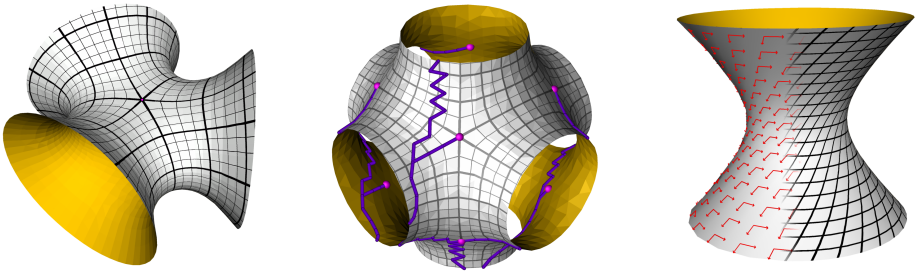


Figure 8.7: Parameterized minimal surfaces: Trinoid (left), Schwarz P-surface with its cut graph, and the Hyperboloid parameterized using non-orthogonal frames.

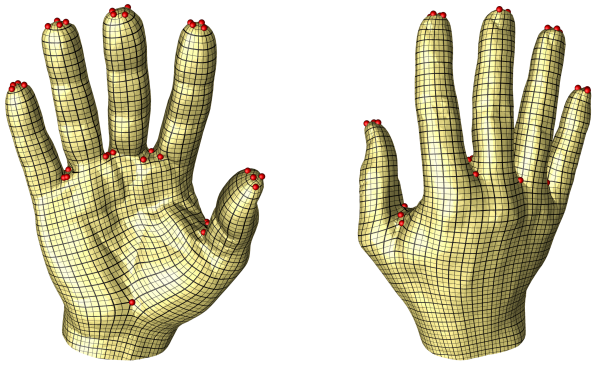


Figure 8.8: Parameterization of the hand model. Branch points are marked red.

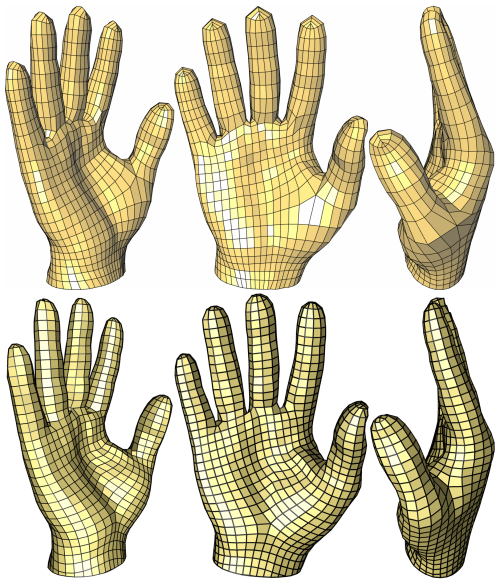


Figure 8.9: Comparison of the remeshed hand model with [Tong et al., 2006] (top). Bottom: constructed with QuadCover. The hand model is courtesy of Pierre Alliez. Except from setting preprocessing parameters, no interaction was involved.

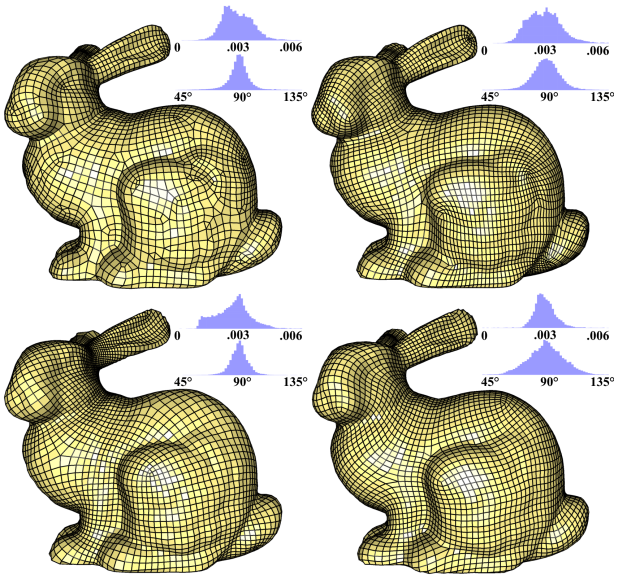


Figure 8.10: Comparison of remeshing results of the Stanford bunny. Models were produced by [Ray et al., 2006] and [Tong et al., 2006] (top), [Dong et al., 2006] and QuadCover (bottom). The upper histogram next to each model shows the distribution of edge lengths, the lower histogram represents angle distribution.

	[Ray et al., 2006]	[Tong et al., 2006]	[Dong et al., 2006]	QuadCover
vertices	6355	6576	7202	6535
irreg. vert.	314	34	26	37
RSD edge	25.0%	28.3%	30.8%	18.2%
RSD angle	10.7%	12.6%	7.8%	14.8%

Table 8.1: The number of total and irregular vertices of the models shown in Figure 8.10, as well as the relative standard deviation of their edge lengths and vertex angles.

Bibliography

J. H. Ahlberg, E. N. Nilson, and J. L. Walsh. *The theory of splines and their applications*. Academic Press, New York, 1967.

B. Aksoylu, A. Khodakovsky, and P. Schröder. Multilevel solvers for unstructured surface meshes. *SIAM Journal on Scientific Computing*, 26(4):1146–1165, 2005.

M. Alexa. Merging polyhedral shapes with scattered features. *The Visual Computer*, 16 (1):26–37, 2000.

M. Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–196, 2002.

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. Proceedings of SIGGRAPH 2003.

P. Alliez, M. Attene, C. Gotsman, and G. Ucelli. Recent advances in remeshing of surfaces. In L. De Floriani and M. Spagnuolo, editors, *Shape Analysis and Structuring*, Mathematics and Visualization, pages 53–82. Springer, Berlin, Heidelberg, 2008.

P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. Anisotropic polygonal remeshing. *ACM Transactions on Graphics*, 22(3):485–493, 2003. Proceedings of SIGGRAPH 2003.

P. Alliez and C. Gotsman. Recent advances in compression of 3D meshes. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 3–26. Springer, Berlin, Heidelberg, 2005.

D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics*, 24(3): 408–416, 2005. Proceedings of SIGGRAPH 2005.

N. Arad and G. Elber. Isometric texture mapping for free-form surfaces. *Computer Graphics Forum*, 16(5):247–256, 1997.

P. N. Azariadis and N. A. Aspragathos. On using planar developments to perform texture mapping on arbitrarily curved surfaces. *Computers & Graphics*, 24(4):539–554, 2000.

L. Balmelli, G. Taubin, and F. Bernardini. Space-optimized texture maps. *Computer Graphics Forum*, 21(3):411–420, 2002. Proceedings of Eurographics 2002.

A. Belyaev. On transfinite barycentric coordinates. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 89–99. Eurographics Association, 2006.

M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum*, 27(2):449–458, 2008. Proceedings of Eurographics 2008.

C. Bennis, J.-M. Vézien, and G. Iglésias. Piecewise surface flattening for non-distorted texture mapping. *ACM SIGGRAPH Computer Graphics*, 25(4):237–246, 1991. Proceedings of SIGGRAPH '91.

E. Bier and K. Sloan. Two-part texture mappings. *IEEE Computer Graphics and Applications*, 6(9):40–53, 1986.

H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. *ACM Transactions on Graphics*, 21(3):312–321, 2002. Proceedings of SIGGRAPH 2002.

H. Birkholz. Shape-preserving parametrization of genus 0 surfaces. In *Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2004)*, pages 57–64, 2004.

V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, 2003. Proceedings of Eurographics 2003.

V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. Exchanging faces in images. *Computer Graphics Forum*, 23(3):669–676, 2004. Proceedings of Eurographics 2004.

V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH '99*, pages 187–194. ACM Press, 1999.

J. F. Blinn. Simulation of wrinkled surfaces. *ACM SIGGRAPH Computer Graphics*, 12 (3):286–292, 1978. Proceedings of SIGGRAPH '78.

T. Bobach, G. Farin, D. Hansford, and G. Umlauf. Discrete harmonic functions from local coordinates. In R. Martin, M. Sabin, and J. Winkler, editors, *Mathematics of Surfaces XII*, volume 4647 of *Lecture Notes in Computer Science*, pages 93–103. Springer, 2007. Proceedings of the 12th IMA International Conference.

A. I. Bobenko and B. A. Springborn. Variational principles for circle patterns and koebe's theorem. *Transactions of the American Mathematical Society*, 356(2):659–689, 2004.

I. Boier-Martin, H. Rushmeier, and J. Jin. Parameterization of triangle meshes over quadrilateral domains. In *Proceedings of the Second Eurographics Symposium on Geometry Processing (SGP 2004)*, pages 193–203. Eurographics Association, 2004.

M. Botsch, D. Bommes, and L. Kobbelt. Efficient linear system solvers for mesh processing. In *Mathematics of Surfaces XI*, volume 3604 of *Lecture Notes in Computer Science*, pages 62–83. Springer, Berlin, Heidelberg, 2005. Proceedings of the 11th IMA International Conference.

P. L. Bowers and M. K. Hurdal. Planar conformal mappings of piecewise flat surfaces. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 3–34. Springer, Heidelberg, 2003.

S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, second edition, 2002.

S. Campagna and H.-P. Seidel. Parameterizing meshes with arbitrary topology. In *Proceedings of Image and Multidimensional Digital Signal Processing '98*, pages 287–290, 1998.

N. A. Carr and J. C. Hart. Meshed atlases for real-time procedural solid texturing. *ACM Transactions on Graphics*, 21(2):106–131, 2002.

N. A. Carr and J. C. Hart. Painting detail. *ACM Transactions on Graphics*, 23(3): 587–594, 2004. Proceedings of SIGGRAPH 2004.

N. A. Carr, J. Hoberock, K. Crane, and J. C. Hart. Rectangular multi-chart geometry images. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 181–190. Eurographics Association, 2006.

Z. Chen, L. Liu, Z. Zhang, and G. Wang. Surface parameterization via aligning optimal local flattening. In *Proceedings of the 2007 Symposium on Solid and Physical Modeling (SPM '07)*, pages 291–296. ACM Press, 2007.

G. Choquet. Sur un type de transformation analytique généralisant la représentation conforme et défiué au moyen de fonctions harmoniques. *Bulletin des Sciences Mathématiques*, 69:156–165, 1945.

B. Chow and F. Luo. Combinatorial ricci flows on surfaces. *Journal of Differential Geometry*, 63(1):97–129, 2003.

F. R. K. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

B. Cipra. You can't always hear the shape of a drum. In *What's Happening in the Mathematical Sciences*, volume 1. American Mathematical Society, Providence, RI, 1993.

U. Clarenz, N. Litke, and M. Rumpf. Axioms and variational problems in surface parameterization. *Computer Aided Geometric Design*, 21(8):727–749, 2004.

D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, 2004. Proceedings of SIGGRAPH 2004.

C. R. Collins and K. Stephenson. A circle packing algorithm. *Computational Geometry: Theory and Applications*, 25(3):233–256, 2003.

R. Courant. *Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces*. Interscience, New York, 1950.

Y. C. de Verdière. On a new graph invariant and a criterion for planarity. In *Graph Structure Theory*, volume 147 of *Contemporary Mathematics*, pages 137–148. American Mathematical Society, 1993.

P. Degener, J. Meseth, and R. Klein. An adaptable surface parameterization method. In *Proceedings of the 12th International Meshing Roundtable (IMR 2003)*, pages 201–213, 2003.

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

M. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.

S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Quadrangulating a mesh using Laplacian eigenvectors. Technical Report UIUCDCS-R-2005-2583, University of Illinois, June 2005a.

S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *ACM Transactions on Graphics*, 25(3):1057–1066, 2006. Proceedings of SIGGRAPH 2006.

S. Dong and M. Garland. Iterative methods for improving mesh parameterizations. In *Proceedings of Shape Modeling International (SMI '07)*, pages 199–210. IEEE Computer Society, 2007.

S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design*, 22(5):392–423, 2005b.

J. Douglas. Solution of the problem of Plateau. *Transactions of the American Mathematical Society*, 33(1):263–321, 1931.

T. Duchamp, A. Certain, T. DeRose, and W. Stuetzle. Hierarchical computation of PL harmonic embeddings. Technical report, University of Washington, July 1997.

M. Eck, T. D. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH '95*, pages 173–182. ACM Press, 1995.

I. Eckstein, V. Surazhsky, and C. Gotsman. Texture mapping with hard constraints. *Computer Graphics Forum*, 20(3):95–104, 2001. Proceedings of Eurographics 2001.

J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.

J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the 16th Symposium on Discrete Algorithms (SODA '05)*, pages 1038–1046. SIAM, 2005.

G. Farin. Surfaces over Dirichlet tessellations. *Computer Aided Geometric Design*, 7 (1-4):281–292, 1990.

H. M. Farkas and I. Kra. *Riemann Surfaces*, volume 71 of *Graduate Texts in Mathematics*. Springer, second edition, 1992.

H. Ferguson and A. Rockwood. Multiperiodic functions for surface design. *Computer Aided Geometric Design*, 10(3–4):315–328, 1993.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23: 298–305, 1973.

M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–633, 1975.

E. Fiume, A. Fournier, and V. Canale. Conformal texture mapping. In *Proceedings of Eurographics '87*, pages 53–64, 1987.

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

M. S. Floater. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling*, 4(3–4):165–182, 1998.

M. S. Floater. Meshless parameterization and B-spline surface approximation. In R. Cipolla and R. Martin, editors, *The Mathematics of Surfaces IX*, pages 1–18, London, 2000. Springer.

M. S. Floater. Convex combination maps. In J. Levesley, I. J. Anderson, and J. C. Mason, editors, *Algorithms for Approximation IV*, pages 18–23, 2002.

M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003a.

M. S. Floater. One-to-one piecewise linear mappings over triangulations. *Mathematics of Computation*, 72(242):685–696, 2003b.

M. S. Floater and K. Hormann. Parameterization of triangulations and unorganized points. In A. Iske, E. Quak, and M. S. Floater, editors, *Tutorials on Multiresolution in Geometric Modelling*, Mathematics and Visualization, pages 287–316. Springer, Berlin, Heidelberg, 2002.

M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. Springer, Berlin, Heidelberg, 2005.

M. S. Floater, K. Hormann, and G. Kós. A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*, 24(1–4): 311–331, 2006.

M. S. Floater, K. Hormann, and M. Reimers. Parameterization of manifold triangulations. In C. K. Chui, L. L. Schumaker, and J. Stöckler, editors, *Approximation Theory X: Abstract and Classical Analysis*, Innovations in Applied Mathematics, pages 197–209. Vanderbilt University Press, Nashville, TN, 2002.

M. S. Floater, G. Kós, and M. Reimers. Mean value coordinates in 3D. *Computer Aided Geometric Design*, 22(7):623–631, 2005.

M. S. Floater and M. Reimers. Meshless parameterization and surface reconstruction. *Computer Aided Geometric Design*, 18(2):77–92, 2001.

J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Boston, second edition, 1995.

I. Friedel, P. Schröder, and M. Desbrun. Unconstrained spherical parameterization. *Journal of Graphics Tools*, 12(1):17–26, 2007.

W. Fulton. *Algebraic Topology: A first course*, volume 153 of *Graduate Texts in Mathematics*. Springer, 1995.

V. A. Garanzha. Maximum norm optimization of quasi-isometric mappings. *Numerical Linear Algebra with Applications*, 9(6,7):493–510, 2002.

M. Garland, A. Willmott, and P. S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (SI3D '01)*, pages 49–58. ACM Press, 2001.

C. F. Gauß. Disquisitiones generales circa superficies curvas. *Commentationes Societatis Regiæ Scientiarum Gottingensis Recentiores*, 6:99–146, 1827.

W. J. Gordon and J. A. Wixom. Pseudo-harmonic interpolation on convex domains. *SIAM Journal on Numerical Analysis*, 11(5):909–933, 1974.

S. J. Gortler, C. Gotsman, and D. Thurston. Discrete one-forms on meshes and applications to 3D mesh parameterization. *Computer Aided Geometric Design*, 23(2):83–112, 2006.

C. Gotsman. On graph partitioning, spectral analysis, and digital mesh processing. In *Proceedings of Shape Modeling International (SMI '03)*, pages 165–171. IEEE Computer Society, 2003.

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

G. Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13(3):143–154, 1994. Proceedings of Eurographics '94.

G. Greiner and K. Hormann. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In A. L. Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, Innovations in Applied Mathematics, pages 163–172. Vanderbilt University Press, Nashville, 1997.

C. M. Grimm and J. F. Hughes. Modeling surfaces of arbitrary topology using manifolds. In *Proceedings of SIGGRAPH '95*, pages 359–368. ACM Press, 1995.

C. M. Grimm and J. F. Hughes. Parameterizing N-holed tori. In *Mathematics of Surfaces*, volume 2768 of *Lecture Notes in Computer Science*, pages 14–29. Springer, Berlin, Heidelberg, 2003. Proceedings of the 10th IMA International Conference.

L. Gross and G. Farin. A transfinite form of Sibson's interpolant. *Discrete Applied Mathematics*, 93(1):33–50, 1999.

X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Transactions on Graphics*, 21(3):355–361, 2002. Proceedings of SIGGRAPH 2002.

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 Transaction on Medical Imaging*, 23(8):949–958, 2004a.

X. Gu, Y. Wang, and S.-T. Yau. Computing conformal invariants: Period matrices. *Communications in Information and Systems*, 3(3):153–170, 2003a.

X. Gu, Y. Wang, and S.-T. Yau. Geometric compression using Riemann surface structure. *Communications in Information and Systems*, 3(3):171–182, 2003b.

X. Gu, Y. Wang, and S.-T. Yau. Multiresolution computation of conformal structures of surfaces. *Journal of Systemics, Cybernetics and Informatics*, 1(5):45–50, 2004b.

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

X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Proceedings of the First Eurographics Symposium on Geometry Processing (SGP 2003)*, pages 127–137. Eurographics Association, 2003.

I. Guskov. An anisotropic mesh parameterization scheme. *Engineering with Computers*, 20(2):129–135, 2004.

I. Guskov, A. Khodakovsky, P. Schröder, and W. Sweldens. Hybrid meshes: Multiresolution using regular and irregular refinement. In *Proceedings of the 18th Annual Symposium on Computational Geometry (SCG '02)*, pages 264–272. ACM Press, 2002.

I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH '99*, pages 325–334. ACM Press, 1999.

I. Guskov, K. Vidimče, W. Sweldens, and P. Schröder. Normal meshes. In *Proceedings of SIGGRAPH 2000*, pages 95–102. ACM Press, 2000.

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.

Y. He, X. Gu, and H. Qin. Rational spherical splines for genus zero shape modeling. In *Proceedings of the 2005 International Conference on Shape Modeling and Applications (SMI '05)*, pages 82–91. IEEE Computer Society, 2005.

A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *Proceedings of SIGGRAPH 2000*, pages 517–526. ACM Press, 2000.

A. Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, Pasadena, CA, 2003.

H. Hiyoshi and K. Sugihara. Voronoi-based interpolation with higher continuity. In *Proceedings of the 16th Annual Symposium on Computational Geometry (SCG '00)*, pages 242–250. ACM Press, 2000.

H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH '96*, pages 99–108. ACM Press, 1996.

H. Hoppe and E. Praun. Shape compression using spherical geometry images. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 27–46. Springer, Berlin, Heidelberg, 2005.

K. Hormann. Fitting free form surfaces. In B. Girod, G. Greiner, and H. Niemann, editors, *Principles of 3D Image Analysis and Synthesis*, volume 556 of *The International Series in Engineering and Computer Science*, chapter 4.7, pages 192–202. Kluwer Academic Publishers, Boston, MA, 2000.

K. Hormann. *Theory and Applications of Parameterizing Triangulations*. PhD thesis, Department of Computer Science, University of Erlangen, Nov. 2001.

K. Hormann and M. S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics*, 25(4):1424–1441, 2006.

K. Hormann and G. Greiner. MIPS: An efficient global parametrization method. In P.-J. Laurent, P. Sablonnière, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 1999*, Innovations in Applied Mathematics, pages 153–162. Vanderbilt University Press, Nashville, TN, 2000a.

K. Hormann and G. Greiner. Quadrilateral remeshing. In B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Proceedings of Vision, Modeling, and Visualization 2000*, pages 153–162, Saarbrücken, Germany, 2000b. infix.

K. Hormann, G. Greiner, and S. Campagna. Hierarchical parametrization of triangulated surfaces. In B. Girod, H. Niemann, and H.-P. Seidel, editors, *Proceedings of Vision, Modeling, and Visualization 1999*, pages 219–226, Erlangen, Germany, 1999. infix.

K. Hormann, U. Labsik, and G. Greiner. Remeshing triangulated surfaces with optimal parameterizations. *Computer-Aided Design*, 33(11):779–788, 2001.

K. Hormann and M. Reimers. Triangulating point clouds with spherical topology. In T. Lyche, M.-L. Mazure, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 2002*, Modern Methods in Applied Mathematics, pages 215–224. Nashboro Press, Brentwood, TN, 2003.

K. Hormann and N. Sukumar. Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum*, 27(5):1513–1520, 2008. Proceedings of SGP 2008.

K. Hormann and M. Tarini. A quadrilateral rendering primitive. In *Proceedings of the 2004 Workshop on Graphics Hardware (GH 2004)*, Eurographics Symposium Proceedings, pages 7–14. Eurographics Association, 2004.

M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper, and D. A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI '99)*, volume 1679 of *Lecture Notes in Computer Science*, pages 279–286. Springer, Berlin, Heidelberg, 1999.

T. Igarashi and D. Cosgrove. Adaptive unwrapping for interactive texture painting. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (SI3D '01)*, pages 209–216. ACM Press, 2001.

M. Isenburg, S. Gumhold, and C. Gotsman. Connectivity shapes. In *Proceedings of IEEE Visualization 2001*, pages 135–142. IEEE Computer Society, 2001.

M. Isenburg and P. Lindstrom. Streaming meshes. In *Proceedings of IEEE Visualization 2005*, pages 231–238. IEEE Computer Society, 2005.

M. Jin, J. Kim, and X. Gu. Discrete surface ricci flow: Theory and applications. In R. Martin, M. Sabin, and J. Winkler, editors, *Mathematics of Surfaces XII*, volume 4647 of *Lecture Notes in Computer Science*, pages 209–232. Springer, 2007. Proceedings of the 12th IMA International Conference.

M. Jin, F. Luo, and X. Gu. Computing surface hyperbolic structure and real projective structure. In *Proceedings of the 2006 Symposium on Solid and Physical Modeling (SPM '06)*, pages 105–116. ACM Press, 2006.

M. Jin, Y. Wang, S.-T. Yau, and X. Gu. Optimal global conformal surface parameterization. In *Proceedings of IEEE Visualization 2004*, pages 267–274. IEEE Computer Society, 2004.

P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Gr.*, 26(3):Article 71, 2007. Proceedings of ACM SIGGRAPH 2007.

J. Jost. *Compact Riemann Surfaces: An Introduction to Contemporary Mathematics*. Springer, second edition, 2002.

T. Ju, P. Liepa, and J. Warren. A general geometric construction of coordinates in a convex simplicial polytope. *Computer Aided Geometric Design*, 24(3):161–178, 2007.

T. Ju, S. Schaefer, and J. Warren. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics*, 24(3):561–566, 2005a. Proceedings of ACM SIGGRAPH 2005.

T. Ju, S. Schaefer, J. Warren, and M. Desbrun. A geometric construction of coordinates for convex polyhedra using polar duals. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 181–186. Eurographics Association, 2005b.

D. Julius, V. Kraevoy, and A. Sheffer. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum*, 24(3):581–590, 2005. Proceedings of Eurographics 2005.

M. Kac. Can one hear the shape of a drum? *American Mathematical Monthly*, 73(4), 1966.

F. Kälberer, M. Nieser, and K. Polthier. Quadcover—surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2007. Proceedings of Eurographics 2007.

Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of SIGGRAPH 2000*, pages 279–286. ACM Press, 2000.

Z. Karni, C. Gotsman, and S. J. Gortler. Free-boundary linear parameterization of 3D meshes in the presence of constraints. In *Proceedings of the 2005 International Conference on Shape Modeling and Applications (SMI '05)*, pages 266–275. IEEE Computer Society, 2005.

L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics*, 25(2):412–438, 2006.

A. Khodakovsky, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics*, 22(3):350–357, 2003. Proceedings of SIGGRAPH 2003.

W. Klingenberg. *A Course in Differential Geometry*. Springer, Berlin, Heidelberg, 1978.

H. Kneser. Lösung der Aufgabe 41. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 35:123–124, 1926.

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

S. Kolmanič and N. Guid. The flattening of arbitrary surfaces by approximation with developable stripes. In U. Cugini and M. J. Wozny, editors, *From geometric modeling to shape modeling*, volume 80 of *International Federation for Information Processing*, pages 35–46. Kluwer Academic Publishers, Boston, 2001.

Y. Koren. On spectral graph drawing. In *Computing and Combinatorics*, volume 2697 of *Lecture Notes in Computer Science*, pages 496–508. Springer, Berlin, Heidelberg, 2003. Proceedings of the 9th Annual International Conference (COCOON 2003).

G. Kós and T. Várady. Parameterizing complex triangular meshes. In T. Lyche, M.-L. Mazure, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 2002*, Modern Methods in Mathematics, pages 265–274. Nashboro Press, Brentwood, TN, 2003.

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

V. Kraevoy and A. Sheffer. Template based mesh completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 13–22. Eurographics Association, 2005.

V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: constructing constrained texture maps. *ACM Transactions on Graphics*, 22(3):326–333, 2003. Proceedings of SIGGRAPH 2003.

E. Kreyszig. *Differential Geometry*. Dover, New York, 1991.

U. Labsik, K. Hormann, and G. Greiner. Using most isometric parametrizations for remeshing polygonal surfaces. In R. Martin and W. Wang, editors, *Proceedings of Geometric Modeling and Processing 2000*, pages 220–228, Hong Kong, China, 2000. IEEE Computer Society.

J. H. Lambert. *Beyträge zum Gebrauche der Mathematik und deren Anwendung*, volume 3. Buchhandlung der Realschule, Berlin, 1772.

T. Langer, A. Belyaev, and H.-P. Seidel. Spherical barycentric coordinates. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 81–88. Eurographics Association, 2006.

T. Langer, A. Belyaev, and H.-P. Seidel. Mean value coordinates for arbitrary spherical polygons and polyhedra in $\mathbb{R}^3$. In P. Chenin, T. Lyche, and L. L. Schumaker, editors, *Curve and Surface Design: Avignon 2006*, Modern Methods in Applied Mathematics, pages 193–202. Nashboro Press, Brentwood, TN, 2007.

T. Langer and H.-P. Seidel. Higher order barycentric coordinates. *Computer Graphics Forum*, 27(2):459–466, 2008. Proceedings of Eurographics 2008.

F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SCG '01)*, pages 80–89. ACM Press, 2001.

A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proceedings of SIGGRAPH 2000*, pages 85–94. ACM Press, 2000.

A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proceedings of SIGGRAPH '99*, pages 343–350. ACM Press, 1999.

A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. In *Proceedings of SIGGRAPH '98*, pages 95–104. ACM Press, 1998.

E. T. Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989.

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

J. Lengyel, E. Praun, A. Finkelstein, and H. Hoppe. Real-time fur over arbitrary surfaces. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (SI3D '01)*, pages 227–232. ACM Press, 2001.

B. Lévy. Constrained texture mapping for polygonal meshes. In *Proceedings of SIGGRAPH 2001*, pages 417–424. ACM Press, 2001.

B. Lévy. Dual domain extrapolation. *ACM Transactions on Graphics*, 22(3):364–369, 2003. Proceedings of SIGGRAPH 2003.

B. Lévy. Laplace-Beltrami eigenfunctions: Towards an algorithm that "understands" geometry. In *Proceedings of the 2006 International Conference on Shape Modeling and Applications (SMI '06)*, pages 65–72. IEEE Computer Society, 2006.

B. Lévy and J.-L. Mallet. Non-distorted texture mapping for sheared triangulated meshes. In *Proceedings of SIGGRAPH '98*, pages 343–352. ACM Press, 1998.

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

W.-C. Li, N. Ray, and B. Lévy. Automatic and interactive mesh to T-spline conversion. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 191–200. Eurographics Association, 2006.

J. Liesen, E. de Sturler, A. Sheffer, Y. Aydin, and C. Siefert. Preconditioners for indefinite linear systems arising in surface parameterization. In *Proceedings of the 9th International Meshing Roundtable*, pages 71–82, 2001.

Y. Lipman, J. Kopf, D. Cohen-Or, and D. Levin. GPU-assisted positive mean value coordinates for mesh deformations. In A. Belyaev and M. Garland, editors, *Geometry Processing*, Eurographics Symposium Proceedings, pages 117–123, Barcelona, Spain, 2007. Eurographics Association.

Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM Transactions on Graphics*, 27(3):Article 78, 2008. Proceedings of SIGGRAPH 2008.

N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 207–216. Eurographics Association, 2005.

L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. *Computer Graphics Forum*, 27(5):1495–1504, 2008. Proceedings of SGP 2008.

F. Losasso, H. Hoppe, S. Schaefer, and J. Warren. Smooth geometry images. In *Proceedings of the First Eurographics Symposium on Geometry Processing (SGP 2003)*, pages 138–145. Eurographics Association, 2003.

L. Lovász and A. Schrijver. On the null space of a Colin de Verdière matrix. *Annales de l'institut Fourier*, 49(3):1017–1026, 1999.

S. D. Ma and H. Lin. Optimal texture mapping. In *Proceedings of Eurographics '88*, pages 421–428, 1988.

W. Ma and J. P. Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.

J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *Proceedings of SIGGRAPH '93*, pages 27–34. ACM Press, 1993.

E. A. Malsch and G. Dasgupta. Interpolations for temperature distributions: A method for all non-concave polygons. *International Journal of Solids and Structures*, 41(8): 2165–2188, 2004.

E. A. Malsch and G. Dasgupta. Algebraic construction of smooth interpolants on polygonal domains. *The Mathematica Journal*, 9(3):641–658, 2005.

E. A. Malsch, J. J. Lin, and G. Dasgupta. Smooth two dimensional interpolants: a recipe for all polygons. *Journal of Graphics Tools*, 10(2):27–39, 2005.

M. Marinov and L. Kobbelt. Direct anisotropic quad-dominant remeshing. In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications*, pages 207–216, 2004.

M. Marinov and L. Kobbelt. A robust two-step procedure for quad-dominant remeshing. *Computer Graphics Forum*, 25(3):537–546, 2006. Proceedings of Eurographics 2006.

S. R. Marschner, B. K. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In B. Péroche and H. Rushmeier, editors, *Rendering Techniques 2000*, pages 231–242. Springer, Wien, New York, 2000. Proceedings of the 11th Eurographics Workshop on Rendering.

J. McCartney, B. K. Hinds, and B. L. Seow. The flattening of triangulated surfaces incorporating darts and gussets. *Computer-Aided Design*, 31(4):249–260, 1999.

W. H. Meeks. A survey of the geometric results in the classical theory of minimal surfaces. *Bulletin of the Brazilian Mathematical Society*, 12(1):29–86, 1981.

G. Mercator. Nova et aucta orbis terrae descriptio ad usum navigantium emendate accommodata. Duisburg, 1569.

O. Meshar, D. Irony, and S. Toledo. An out-of-core sparse symmetric indefinite factorization method. *ACM Transactions on Mathematical Software*, 32(3):445–471, 2006.

M. Meyer, H. Lee, A. H. Barr, and M. Desbrun. Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, 7(1):13–22, 2002.

V. J. Milenkovic. Rotational polygon containment and minimum enclosure. In *Proceedings of the 14th Annual Symposium on Computational Geometry (SCG '98)*, pages 1–8. ACM Press, 1998.

J. Mitani and H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics*, 23(3):259–263, 2004. Proceedings of SIGGRAPH 2004.

F. Morgan. *Riemannian Geometry: A Beginner's Guide*. A K Peters, Wellesley, MA, second edition, 1998.

P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. *Computer Graphics Forum*, 27(5):1487–1494, 2008. Proceedings of SGP 2008.

T. Needham. *Visual Complex Analysis*. Oxford University Press, Oxford, New York, 1997.

X. Ni, M. Garland, and J. C. Hart. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics*, 23(3):613–622, 2004. Proceedings of SIGGRAPH 2004.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, second edition, 2006.

J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. In *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 263–274. Springer, Berlin, Heidelberg, 1998. Proceedings of the 6th International Symposium (GD '98).

L. Parida and S. P. Mudur. Constraint-satisfying planar development of complex surfaces. *Computer-Aided Design*, 25(4):225–232, 1993.

H. K. Pedersen. Decorating implicit surfaces. In *Proceedings of SIGGRAPH '95*, pages 291–300. ACM Press, 1995.

J. Peng, D. Kristjansson, and D. Zorin. Interactive modeling of topologically complex geometric detail. *ACM Transactions on Graphics*, 23(3):635–643, 2004. Proceedings of SIGGRAPH 2004.

U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *Proceedings of SIGGRAPH 2000*, pages 471–478. ACM Press, 2000.

J. A. F. Plateau. *Statistique Expérimentale et Théorique des Liquides Soumis aux Seules Forces Moléculaires*. Gauthier-Villars, Paris, 1873.

K. Polthier and E. Preuss. Identifying vector field singularities using a discrete Hodge decomposition. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, Mathematics and Visualization, pages 113–134. Springer, 2003.

S. D. Porumbescu, B. Budge, L. Feng, and K. I. Joy. Shell maps. *ACM Transactions on Graphics*, 24(3):626–633, 2005. Proceedings of SIGGRAPH 2005.

E. Praun, A. Finkelstein, and H. Hoppe. Lapped textures. In *Proceedings of SIGGRAPH 2000*, pages 465–470. ACM Press, 2000.

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

E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *Proceedings of SIGGRAPH 2001*, pages 179–184. ACM Press, 2001.

C. Ptolemy. *The Geography*. Dover, 1991. Translated by E. L. Stevenson.

B. Purnomo, J. D. Cohen, and S. Kumar. Seamless texture atlases. In *Proceedings of the Second Eurographics Symposium on Geometry Processing (SGP 2004)*, pages 67–76. Eurographics Association, 2004.

T. Radó. Aufgabe 41. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 35:49, 1926.

T. Radó. The problem of least area and the problem of Plateau. *Mathematische Zeitschrift*, 32:763–796, 1930.

N. Ray and B. Lévy. Hierarchical least squares conformal maps. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG 2003)*, pages 263–270. IEEE Computer Society, 2003.

N. Ray, W.-C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Transactions on Graphics*, 25(4):1460–1485, 2006.

N. Ray, W. C. Li, B. Vallet, F. Boutantin, B. Lévy, and C. Borgese. Graphite. `http://alice.loria.fr/software/graphite/`, 2003.

N. Ray, B. Vallet, W. C. Li, and B. Lévy. N-symmetry direction field design. *ACM Transactions on Graphics*, 27(2):Article 10, 2008.

M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

B. Riemann. *Grundlagen für eine allgemeine Theorie der Functionen einer veränderlichen complexen Größe*. PhD thesis, Universität Göttingen, 1851.

A. Rockwood and H. Park. Interactive design of smooth genus N objects using multi-periodic functions and applications. *International Journal of Shape Modeling*, 5(2): 135–157, 1999.

B. Rodin and D. Sullivan. The convergence of circle packings to the Riemann mapping. *Journal of Differential Geometry*, 26(2):349–360, 1987.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. Practical spherical embedding of manifold triangle meshes. In *Proceedings of the 2005 International Conference on Shape Modeling and Applications (SMI '05)*, pages 256–265. IEEE Computer Society, 2005.

P. V. Sander, S. J. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. In *Proceedings of the 13th Eurographics Workshop on Rendering (EGWR '02)*, pages 87–98. Eurographics Association, 2002.

P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Proceedings of SIGGRAPH 2001*, pages 409–416. ACM Press, 2001.

P. V. Sander, Z. J. Wood, S. J. Gortler, J. Snyder, and H. Hoppe. Multi-chart geometry images. In *Proceedings of the First Eurographics Symposium on Geometry Processing (SGP 2003)*, pages 138–145. Eurographics Association, 2003.

S. Schaefer, T. Ju, and J. Warren. A unified, integral construction for coordinates over closed curves. *Computer Aided Geometric Design*, 24(8–9):481–493, 2007.

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

E. L. Schwartz, B. Merker, E. Wolfson, and A. Shaw. Applications of computer graphics and image processing to 2D and 3D modeling of the functional architecture of visual cortex. *IEEE Computer Graphics and Applications*, 8(4):13–23, 1988.

E. L. Schwartz, A. Shaw, and E. Wolfson. A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):1005–1008, 1989.

A. Shamir. A formulation of boundary mesh segmentation. In *Proceedings of the Second International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004)*, pages 82–89. IEEE Computer Society, 2004.

L. Shapira and A. Shamir. Local geodesic parametrization: An ant's perspective. In T. Möller, B. Hamann, and R. Russel, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization. Springer, 2008.

A. Shapiro and A. Tal. Polyhedron realization for shape transformation. *The Visual Computer*, 14(8–9):429–444, 1998.

A. Sheffer. Spanning tree seams for reducing parameterization distortion of triangulated surfaces. In *Proceedings of Shape Modeling International (SMI '02)*, pages 61–66. IEEE Computer Society, 2002.

A. Sheffer. Non-optimal parameterization and user control. In T. Lyche, M.-L. Mazure, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 2002*, Modern Methods in Applied Mathematics, pages 355–364. Nashboro Press, Brentwood, TN, 2003a.

A. Sheffer. Skinning 3D meshes. *Graphical Models*, 65(5):274–285, 2003b.

A. Sheffer and E. de Sturler. Surface parameterization for meshing by triangulation flattening. In *Proceedings of the 9th International Meshing Roundtable (IMR 2000)*, pages 161–172, 2000.

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

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.

A. Sheffer, C. Gotsman, and N. Dyn. Robust spherical parametrization of triangular meshes. In D. Cohen-Or, N. Dyn, G. Elber, and A. Shamir, editors, *Proceedings of the 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, pages 94–99, Tel Aviv, Israel, 2003.

A. Sheffer and J. C. Hart. Seamster: inconspicuous low-distortion texture seam layout. In *Proceedings of IEEE Visualization 2002*, pages 291–298. IEEE Computer Society, 2002.

A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. ABF++: fast and robust angle based flattening. *ACM Transactions on Graphics*, 24(2):311–330, 2005.

A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171, 2006.

J. R. Shewchuk. An introduction to the conjugate method without the agonizing pain. Technical report, School of Computer Science, Carnegie Mellon University, 1994. `http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf`.

T. Shimada and Y. Tada. Approximate transformation of an arbitrary curved surface into a plane using dynamic programming. *Computer-Aided Design*, 23(2):153–159, 1991.

R. Sibson. A vector identity for the Dirichlet tesselation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151–155, 1980.

R. Sibson. A brief description of natural neighbour interpolation. In V. Barnett, editor, *Interpolating Multivariate Data*, pages 21–36. Wiley, New York, 1981.

C. Soler, M.-P. Cani, and A. Angelidis. Hierarchical pattern mapping. *ACM Transactions on Graphics*, 21(3):673–680, 2002. Proceedings of SIGGRAPH 2002.

O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parametrization. In *Proceedings of IEEE Visualization 2002*, pages 355–362. IEEE Computer Society, 2002.

O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the Second Eurographics Symposium on Geometry Processing (SGP 2004)*, pages 179–188. Eurographics Association, 2004.

B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. *ACM Transactions on Graphics*, 27(3):Article 77, 2008. Proceedings of SIGGRAPH 2008.

D. Steiner and A. Fischer. Planar parameterization for closed manifold genus-g meshes using any type of positive weights. *Journal of Computing and Information Science in Engineering*, 5(2):118–125, 2005.

K. Stephenson. *Introduction to Circle Packing: The theory of discrete analytic functions*. Cambridge University Press, New York, 2005.

N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61(12):2159–2181, Nov. 2004.

N. Sukumar and E. A. Malsch. Recent advances in the construction of polygonal finite element interpolants. *Archives of Computational Methods in Engineering*, 13(1):129–163, 2006.

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

V. Surazhsky and C. Gotsman. Explicit surface remeshing. In *Proceedings of the First Eurographics Symposium on Geometry Processing (SGP 2003)*, pages 20–30. Eurographics Association, 2003.

M. Tarini, K. Hormann, P. Cignoni, and C. Montani. PolyCube-Maps. *ACM Transactions on Graphics*, 23(3):853–860, 2004. Proceedings of ACM SIGGRAPH 2004.

G. Taubin. A signal processing approach to fair surface design. In *Proceedings of SIGGRAPH '95*, pages 351–358. ACM Press, 1995.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

G. Tewari, J. Snyder, P. V. Sander, S. J. Gortler, and H. Hoppe. Signal-specialized parameterization for piecewise linear reconstruction. In *Proceedings of the Second Eurographics Symposium on Geometry Processing (SGP 2004)*, pages 57–66. Eurographics Association, 2004.

S. Toledo. TAUCS: A library of sparse linear solvers. `http://www.tau.ac.il/~stoledo/taucs/`, 2003.

Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP 2006)*, pages 201–210. Eurographics Association, 2006.

G. Turk. Texture synthesis on surfaces. In *Proceedings of SIGGRAPH 2001*, pages 347–354. ACM Press, 2001.

W. T. Tutte. Convex representations of graphs. *Proceedings of the London Mathematical Society*, 10:304–320, 1960.

W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 13:743–767, 1963.

B. Vallet and B. Lévy. Manifold harmonics. *Computer Graphics Forum*, 27(2):251–260, 2008. Proceedings of Eurographics 2008.

E. L. Wachspress. *A Rational Finite Element Basis*. Academic Press, New York, 1975.

C. C. L. Wang, S. S.-F. Smith, and M. M. F. Yuen. Surface flattening based on energy model. *Computer-Aided Design*, 34(11):823–833, 2002.

J. Warren. Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics*, 6(2):97–108, 1996.

J. Warren. On the uniqueness of barycentric coordinates. In R. Goldman and R. Krasauskas, editors, *Topics in Algebraic Geometry and Geometric Modeling*, volume 334 of *Contemporary Mathematics*, pages 93–99. American Mathematical Society, 2003.

J. Warren, S. Schaefer, A. N. Hirani, and M. Desbrun. Barycentric coordinates for convex sets. *Advances in Computational Mathematics*, 27(3):319–338, 2007.

L.-Y. Wei and M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of SIGGRAPH 2001*, pages 355–360. ACM Press, 2001.

Y.-L. Yang, J. Kim, F. Luo, S.-M. Hu, and X. Gu. Optimal surface parameterization using inverse curvature map. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1054–1066, 2008.

L. Ying, A. Hertzmann, H. Biermann, and D. Zorin. Texture and shape synthesis on surfaces. In S. J. Gortler and K. Myszkowski, editors, *Rendering Techniques 2001*, pages 301–312. Springer, Wien, New York, 2001. Proceedings of the 12th Eurographics Workshop on Rendering.

L. Ying and D. Zorin. A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM Transactions on Graphics*, 23(3):271–275, 2004. Proceedings of SIGGRAPH 2004.

S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Proceedings of the 2004 International Conference on Shape Modeling and Applications (SMI '04)*, pages 200–208. IEEE Computer Society, 2004.

F. W. Young. Multidimensional scaling. In S. Kotz, N. L. Johnson, and C. B. Read, editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 649–659. Wiley, New York, 1985.

R. Zayer, B. Lévy, and H.-P. Seidel. Linear angle based parameterization. In *Proceedings of the 5th Eurographics Symposium on Geometry Processing (SGP 2007)*, pages 135–141. Eurographics Association, 2007.

R. Zayer, C. Rössl, and H.-P. Seidel. Convex boundary angle based flattening. In T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, and R. Westermann, editors, *Proceedings of Vision, Modeling, and Visualization 2003*, pages 281–288, München, Germany, 2003. infix.

R. Zayer, C. Rössl, and H.-P. Seidel. Discrete tensorial quasi-harmonic maps. In *Proceedings of the 2005 International Conference on Shape Modeling and Applications (SMI '05)*, pages 276–285. IEEE Computer Society, 2005a.

R. Zayer, C. Rössl, and H.-P. Seidel. Setting the boundary free: A composite approach to surface parameterization. In *Proceedings of the Third Eurographics Symposium on Geometry Processing (SGP 2005)*, pages 91–100. Eurographics Association, 2005b.

R. Zayer, C. Rössl, and H.-P. Seidel. Variations on angle based flattening. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 187–199. Springer, Berlin, Heidelberg, 2005c.

R. Zayer, C. Rössl, and H.-P. Seidel. Curvilinear spherical parameterization. In *Proceedings of the 2006 International Conference on Shape Modeling and Applications (SMI '06)*, pages 57–64. IEEE Computer Society, 2006.

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

L. Zhang, L. Liu, Z. Ji, and G. Wang. Manifold parameterization. In T. Nishita, Q. Peng, and H.-P. Seidel, editors, *Advances in Computer Graphics*, volume 4035 of *Lecture Notes in Computer Science*, pages 160–171. Springer, 2006. Proceedings of the 24th Computer Graphics International Conference.

K. Zhou, J. Synder, B. Guo, and H.-Y. Shum. Iso-charts: Stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the Second Eurographics Symposium on Geometry Processing (SGP 2004)*, pages 47–56. Eurographics Association, 2004.

K. Zhou, X. Wang, Y. Tong, M. Desbrun, B. Guo, and H.-Y. Shum. TextureMontage: Seamless texturing of arbitrary surfaces from multiple images. *ACM Transactions on Graphics*, 24(3):1148–1155, 2005. Proceedings of SIGGRAPH 2005.

G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.