# On Generalized Barycentric Coordinates and Their Applications in Geometric Modeling

**Torsten Langer**

**MPI Informatik**
**Saarbrücken, Germany**

Dissertation zur Erlangung des Grades
*Doktor der Naturwissenschaften*
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Eingereicht im Mai 2008 in Saarbrücken.

**Datum des Kolloquiums — Date of Defense**
18.12.2008

**Dekan — Dean**
Prof. Dr. Joachim Weickert
Universität des Saarlandes, Saarbrücken, Germany

**Mitglieder des Prüfungsausschusses — Board of Examiners**
Prof. Dr. Joachim Weickert
Universität des Saarlandes, Saarbrücken, Germany

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

Dr. Alexander Belyaev
Heriot-Watt University, Edinburgh, United Kingdom

Dr. Thorsten Thormählen
MPI Informatik, Saarbrücken, Germany

Torsten Langer
MPI Informatik
Campus E1 4
66123 Saarbrücken, Germany
langer@mpi-inf.mpg.de

# Abstract

Generalized barycentric coordinate systems allow us to express the position of a point in space with respect to a given polygon or higher dimensional polytope. In such a system, a coordinate exists for each vertex of the polytope such that its vertices are represented by unit vectors $\mathbf{e}_i$ (where the coordinate associated with the respective vertex is 1, and all other coordinates are 0). Coordinates thus have a geometric meaning, which allows for the simplification of a number of tasks in geometry processing.

Coordinate systems with respect to triangles have been around since the 19th century, and have since been generalized; however, all of them have certain drawbacks, and are often restricted to special types of polytopes. We eliminate most of these restrictions and introduce a definition for 3D mean value coordinates that is valid for arbitrary polyhedra in $\mathbb{R}^3$, with a straightforward generalization to higher dimensions.

Furthermore, we extend the notion of barycentric coordinates in such a way as to allow Hermite interpolation and investigate the capabilities of generalized barycentric coordinates for constructing generalized Bézier surfaces. Finally, we show that barycentric coordinates can be used to obtain a novel formula for curvature computation on surfaces.

# Zusammenfassung

Verallgemeinerte baryzentrische Koordinatensysteme ermöglichen es, die Position eines Punktes im Raum in Bezug auf ein gegebenes Polygon oder ein höherdimensionales Polytop auszudrücken. Ein solches Koordinatensystem enthält eine Koordinate für jeden Eckpunkt des Polytops. Die Eckpunkte selbst werden durch Einheitsvektoren $\mathbf{e}_i$ dargestellt: Die zu dem jeweiligen Eckpunkt gehörige Koordinate hat den Wert 1, alle anderen haben den Wert 0. In diesem Sinne haben die Koordinaten eine geometrische Bedeutung und können viele Aufgaben in der Geometrieverarbeitung vereinfachen.

Solche Koordinatensysteme sind, mit Bezug auf Dreiecke, bereits seit dem 19. Jahrhundert bekannt und wurden seither verallgemeinert. Dennoch haben alle von ihnen gewisse Nachteile; oft sind sie auf bestimmte Typen von Polytopen beschränkt. Wir schlagen eine Definition von 3D-Mittelwertkoordinaten vor, die die meisten dieser Einschränkungen behebt. Insbesondere gilt sie für beliebige Polyeder im $\mathbb{R}^3$ und kann ohne weiteres auf höhere Dimensionen verallgemeinert werden.

Weiterhin erweitern wir den Begriff der baryzentrischen Koordinaten in einer Weise, die Hermite-Interpolation ermöglicht, wir untersuchen, wie baryzentrische Koordinaten zur Definition von verallgemeinerten Bézier-Flächen verwendet werden können, und wir zeigen, dass baryzentrische Koordinaten benutzt werden können, um eine neue Formel zur Berechnung der Krümmung einer Fläche zu erhalten.

# Acknowledgements

This thesis would not have been possible without the help and support of many people. First of all, I thank Hans-Peter Seidel for providing me with the excellent working environment at the Max-Planck-Institut für Informatik, and for giving me the opportunity and support for pursuing my own research interests. I am grateful to Alexander Belyaev who always had time for me to discuss ongoing and future projects. I could rely on him for giving advice, ideas, and encouragement whenever I needed it.

I thank all my present and former colleagues. They make the MPI the special place it is, and I could always count on their support. Special thanks go to Christian Rössl and Hitoshi Yamauchi who created large parts of the geometric modeling library, of which I made extensive use when implementing my work.

The research in this thesis has been supported in part by the EU-Project “AIM@SHAPE” FP6 IST Network of Excellence 506766. The armadillo is courtesy of the Stanford University Computer Graphics Laboratory, Laurent’s hand is courtesy of the AIM@SHAPE Shape Repository, and the cow is courtesy of Kitware Inc.

Last but not least I am most grateful to my family and to Eva Maria Geuter for their unvarying and patient encouragement and support.

# Contents

1 Introduction 1

I Generalized Barycentric Coordinates 7

2 Introduction to Generalized Barycentric Coordinates 9
2.1 Definition of Generalized Barycentric Coordinates . . . . . . . . 10
2.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.3 A Geometric Interpretation for Planar Barycentric Coordinates . . 14
2.4 Coordinates in 3D . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.5 Planar Mean Value Coordinates . . . . . . . . . . . . . . . . . . . 16

3 Spherical Barycentric Coordinates 23
3.1 Definition of Spherical Barycentric Coordinates . . . . . . . . . . 23
3.2 Spherical Wachspress Coordinates . . . . . . . . . . . . . . . . . 25
3.3 Spherical Mean Value Coordinates . . . . . . . . . . . . . . . . . 27
3.4 Bézier Surfaces on Spherical Polygonal Domains . . . . . . . . . 29
3.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

4 Barycentric Coordinates for Arbitrary Polyhedra 31
4.1 Barycentric Coordinates for Arbitrary Polyhedra . . . . . . . . . . 31
4.1.1 Mean value coordinates for arbitrary polyhedra . . . . . . 32
4.1.2 Behavior of the mean value coordinates on the faces . . . 34
4.2 Higher Dimensions . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.3 Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.3.1 Interpolation and extrapolation . . . . . . . . . . . . . . . 35
4.3.2 Space deformations with 3D mean value coordinates . . . 36
4.3.3 Bernstein polynomials on polygons and polyhedra . . . . 37
4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

**5 Higher Order Barycentric Coordinates 39**
5.1 Introduction . . . 39
5.2 Construction of Higher Order Barycentric Coordinates . . . 40
5.2.1 Axioms for conventional barycentric coordinates . . . 40
5.2.2 Axioms for higher order barycentric coordinates . . . 41
5.2.3 Construction of higher order barycentric coordinates . . . 42
5.3 Higher Order Barycentric Coordinates for Space Deformations . . 43
5.3.1 Space deformations with barycentric coordinates . . . 43
5.3.2 The choice of coordinates . . . 44
5.3.3 Examples . . . 46
5.3.4 Running time . . . 51
5.4 Summary and Future Work . . . 51

**II Mean Value Bézier Maps 53**

**6 Mean Value Bézier Surfaces over Quadratic Domains 55**
6.1 Bézier Theory and Barycentric Coordinates . . . 55
6.2 Mean Value Bézier Surfaces . . . 58
6.3 Results . . . 65
6.4 Summary and Future Work . . . 65

**7 Mean Value Bézier Maps 69**
7.1 Introduction . . . 69
7.2 Theoretical Foundation . . . 71
7.3 Applications . . . 74
7.3.1 Bézier curves and surfaces . . . 74
7.3.2 Space deformations . . . 75
7.4 Summary and Future Work . . . 79

**III Curvature Estimation and Barycentric Coordinates 81**

**8 Discrete Normals and Curvatures of Polylines 83**
8.1 Introduction . . . 83
8.2 Approximation of Space Curves . . . 85
8.3 Summary and Future Work . . . 90

**9 Quadratures for Curvature Tensor Estimation 91**
9.1 Introduction . . . 91
9.1.1 Main contributions and discussion . . . 92

9.1.2 Related work . . . . . . . . . . . . . . . . . . . . . . . . 96
9.2 Exact Quadratures for Curvature Integrals . . . . . . . . . . . . . 98
9.2.1 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . 98
9.2.2 Mean curvature . . . . . . . . . . . . . . . . . . . . . . 99
9.2.3 Gaussian curvature . . . . . . . . . . . . . . . . . . . . . 101
9.2.4 The Taubin tensor . . . . . . . . . . . . . . . . . . . . . 102
9.2.5 Direct computation of principal curvatures and directions . 103
9.2.6 Practical curvature tensor estimation . . . . . . . . . . . . 103
9.3 Asymptotic Analysis . . . . . . . . . . . . . . . . . . . . . . . . 104
9.3.1 Averaging of face normals . . . . . . . . . . . . . . . . . 105
9.3.2 Averaging of edges . . . . . . . . . . . . . . . . . . . . . 106
9.3.3 Curvature estimation using quadratures . . . . . . . . . . 110
9.3.4 Necessary conditions and implementation details . . . . . 111
9.4 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . 112
9.4.1 Convergence properties . . . . . . . . . . . . . . . . . . . 112
9.4.2 Error distribution . . . . . . . . . . . . . . . . . . . . . . 113
9.4.3 Curvature tensor . . . . . . . . . . . . . . . . . . . . . . 113
9.5 Summary and Future Work . . . . . . . . . . . . . . . . . . . . . 116

**10 Conclusions 117**

**A Proof of Theorem 5.1 121**

**B Taylor Series Expansion of Space Curves 123**

**C Taylor Series Expansion of Geodesics 129**

**D Euler's Elastica for Space Curves 133**

**Bibliography 135**

# Chapter 1

# Introduction

One of the basic concepts in computer graphics and geometric modeling is location. Whether during acquisition, processing, or visualization of geometric objects, all of these steps are directed at obtaining, modifying, or displaying the location of object points. The mathematical tool to describe a location is an appropriate coordinate system. To describe an object as a whole, usually a global coordinate system is used. However, it is not desirable—or even possible—to work with a global coordinate system during the entire graphics pipeline. If the object is acquired by a laser range scan, it is usually necessary to perform a number of scans, each with a different viewpoint, and each defining a different coordinate system. In the animation stage, the decomposition of the object into parts, represented by a hierarchy of local coordinate systems, is often advantageous. When a triangular mesh is rendered, the shading of the triangles is usually computed by interpolating between its vertices. For this purpose, the points within a triangle are represented in the barycentric coordinate system that is determined by the triangle in question. The objective of this thesis is to extend the notion of barycentric coordinates from triangles to arbitrary polytopes and to explore possible applications.

Barycentric coordinate systems are a special kind of local coordinate system. While "standard" coordinate systems represent vectors as a linear combination of certain basis vectors, barycentric coordinate systems $(\lambda_i)_{i=1}^n$ represent points $\mathbf{x}$ as a linear combination of a set of vertices $\mathbf{v}_i$,

$$\sum_i \lambda_i \mathbf{v}_i = \mathbf{x}.$$

The strength of barycentric coordinate systems lies in their geometric meaning with respect to a given polytope. In particular, at a polytope vertex, the respective coordinate is 1 and all other coordinates are 0. In this way, objects having roughly the shape of a polytope can be described and manipulated in a direct and intuitive

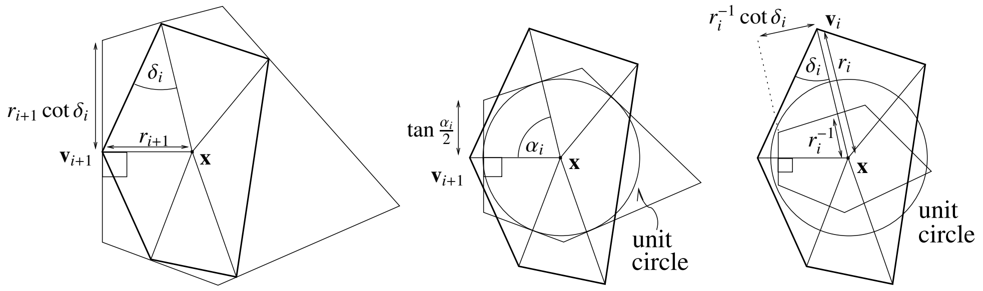


**Figure 1.1: A polygon (bold) and three of its duals corresponding to three different barycentric coordinate systems. The barycentric coordinate of a vertex is obtained by dividing the length of the respective dual edge by the distance to that edge.**

way. We demonstrate the close relation of barycentric coordinates to geometry by showing that many barycentric coordinates for polygons correspond to certain dual polygons, see Figure 1.1.

Another way to regard barycentric coordinates is as the natural means of achieving interpolation within polytopes. Linear interpolation is the simplest interpolation scheme. If only two values $f_1$ and $f_2$ at points $\mathbf{v}_1$ and $\mathbf{v}_2$ are given, it yields $pf_1 + qf_2$ for a point $p\mathbf{v}_1 + q\mathbf{v}_2$, $p + q = 1$. The iterative application of linear interpolation leads to the de Casteljau algorithm, which allows the construction of polynomials of arbitrarily high degree. Barycentric coordinates generalize the concept of linear interpolation to more general domains than line segments. For given values $f_i$ at points $\mathbf{v}_i \in \mathbb{R}^n$, the barycentric interpolation at a point $\mathbf{x}$ with barycentric coordinates $\lambda_i$, $\sum_i \lambda_i = 1$, is

$$\sum_i \lambda_i f_i.$$

Classical barycentric coordinates are defined with respect to (the vertices of) triangles or, more generally, simplices. They were developed by Möbius [Möb27] in the 19th century. Their generalization and application remain an active area of research in geometric modeling and computer graphics until today. The notion of barycentric coordinates has been extended from simplices to arbitrary polygons and polytopes [Wac75, Flo03, JLW07], to point clouds [CFL82, Sib80, BIK*97, Sug99, HS00], and to smooth curves and (hyper-)surfaces [WSHD07, SJW07, Bel06, Rus07, DF08]. Moreover, a general theory of planar barycentric coordinates has been developed [FHK06].

The main application areas of barycentric coordinates are interpolation and deformation. For example, the character articulation in the film “Ratatouille” [Rat07] is based on barycentric deformation using harmonic coordinates [JMD*07]. The

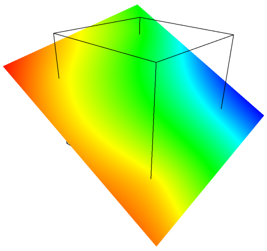


**Figure 1.2: Color values were specified for the vertices of the cube and interpolated with our 3D mean value coordinates in the whole $\mathbb{R}^3$ space. The interpolated values are displayed in a plane intersecting the cube.**

high number of vertices of the detailed character models is reduced to a much smaller number of vertices of a control net. Only their movement must be specified. The barycentric interpolation takes care of moving the character vertices along. This can be seen as reducing the number of degrees of freedom in a geometric model to a "geometrically relevant" subset. In its abstract form, this idea has been employed to minimize the dimension of a geometrically motivated system of linear equations [HSL*06].

However, the aforementioned types of barycentric coordinates are restricted to convex polytopes, polytopes with simplicial faces, or lack other desirable properties. In this thesis, we want to generalize barycentric coordinates such that these limitations are overcome. Let us now summarize our main contributions in greater detail.

**In Part I,** we develop several new types of barycentric coordinates. In particular, we introduce spherical barycentric coordinates which describe the location of points on a sphere with respect to a given spherical polygon, just like classical barycentric coordinates describe points in a plane with respect to a given planar polygon. We prove that spherical mean value coordinates are defined for (almost) arbitrary spherical polygons on the sphere. Furthermore, we extend 3D barycentric coordinates from polytopes with triangular faces to arbitrary polytopes, and we show that 3D mean value coordinates are well defined in the whole $\mathbb{R}^3$ space. Finally, we introduce higher order barycentric coordinates, a modification of these and other 3D barycentric coordinates, that allow the definition of interpolations with derivative control. This is especially useful in the context of deformations since transformations can be prescribed at the vertices. We present examples of barycentric interpolation (Figure 1.2) and deformation (Figure 1.3).

**In Part II,** we explore the relationship between barycentric coordinates and Bézier maps, which are a generalization of Bézier surfaces. We take ad-

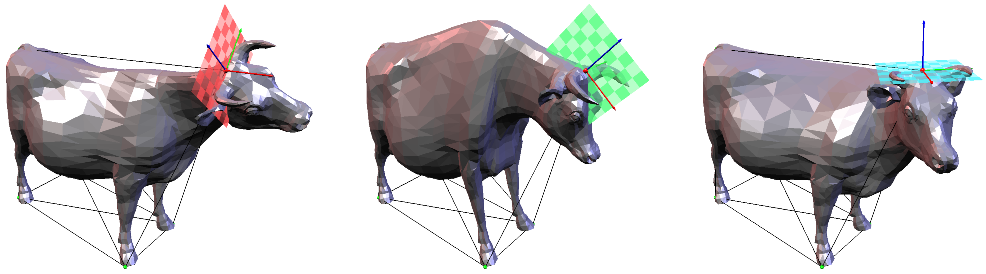


**Figure 1.3: Deformations of the cow model using our higher order 3D mean value coordinates. Here, different rotations around a control vertex at the head of the cow were specified.**

vantage of the fact that Bernstein polynomials, the main building blocks of Bézier maps, can be expressed as polynomials in barycentric coordinates. While Bernstein polynomials in three variables (the three barycentric coordinates with respect to a triangle) are a standard tool for constructing triangular Bézier surfaces, much less attention has been paid to the fact that tensor product Bézier surfaces may be represented as polynomials in (generalized) barycentric coordinates as well, namely in Wachspress coordinates. These Wachspress coordinates with respect to a square are polynomials of bidegree $(1,1)$ in $(x,y)$ such that every Bernstein polynomial of bidegree $(n,n)$ can be expressed as a polynomial in Wachspress coordinates of degree $n$. But Wachspress coordinates are not the only generalized barycentric coordinates for a square. A popular alternative are Floater's mean value coordinates. Mean value coordinates are defined for all convex and nonconvex polygons, and we are no longer restricted to surface patches over rectangular domains (Figure 1.4). We observe that they offer new, interesting possibilities for the construction of Bézier surfaces.

In particular, we prove smoothness conditions for mean value Bézier maps. However, most of our results are not only valid for these but also for the construction of all other differentiable barycentric coordinates. This leads to a whole family of different Bézier maps.

**Part III** reveals an unexpected connection between barycentric coordinates and Discrete Differential Geometry, which is the field of finding and defining discrete counterparts of curvature and other smooth notions of differential geometry.

We suggest a novel approach to calculating the curvature of a smooth surface with quadrature formulae, which are proven with the help of barycentric coordinates. In particular, the mean curvature can be computed quite

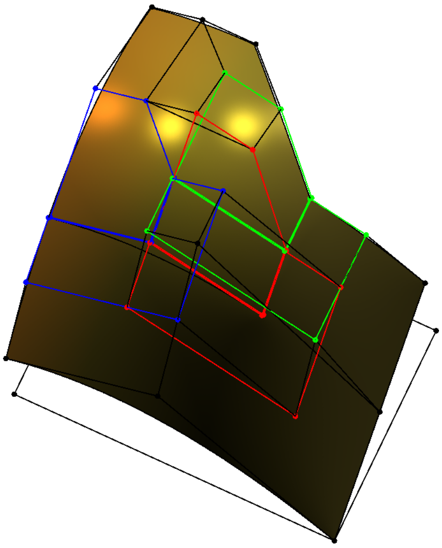


**Figure 1.4: A generalized Bézier surface defined with mean value coordinates.**

simply:

$$H = \frac{1}{2\pi}\int_0^{2\pi} \kappa(\phi)\,\mathrm{d}\phi = \sum_i \omega_i\kappa_i, \qquad \omega_i := \frac{\tan\beta_{i-1} + \tan\beta_i}{\sum_j(\tan\beta_{j-1} + \tan\beta_j)}.$$

Here, $\kappa(\phi)$ and $\kappa_i$ are normal curvatures in given directions, $\beta_j$ are the angles included by these directions, and $\omega_i$ are some kind of generalized barycentric coordinate which determine $H$ as a linear combination of $\kappa_i$ within the line segment $[\kappa_{\min}, \kappa_{\max}]$. We show that this quadrature and similar formulae for the Gaussian curvature and the curvature tensor can be discretized easily and give rise to a pointwise convergent curvature approximation for discrete surfaces.

We conclude this introduction by giving an overview of the remainder of this thesis. It consists of a brief summary of each of the chapters and the publications on which these chapters are based.

Chapter 2 gives an introduction to barycentric coordinates. Chapter 3 introduces spherical barycentric coordinates and explores their properties [LBS06, LBS07b]. Chapter 4 shows how spherical barycentric coordinates can be used to construct barycentric coordinates for arbitrary polytopes in three and more dimensions. We prove that 3D mean value coordinates are defined for the whole $\mathbb{R}^3$ space [LBS06, LBS07b]. In Chapter 5, higher order barycentric coordinates are defined and constructed [LS08]. Chapter 6 describes generalized Bézier surfaces and shows how they extend classical Bézier surfaces [LS07]. In Chapter 7, we consider Bézier maps, a generalization of Bézier surfaces. We investigate the possibilities of defining them on arbitrary polygons and polytopes as domains [LBS08]. In Chapter 8, we analyze convergence properties of curvature estimates for curves and suggest several new curvature formulae [LBS05b]. This is the first step towards the development and analysis of the novel, quadrature-based curvature estimation scheme for surfaces, which is presented in Chapter 9 [LBS07a, LBS05a]. We conclude the thesis in Chapter 10.

# Part I

# Generalized Barycentric Coordinates

# Chapter 2

# Introduction to Generalized Barycentric Coordinates

Barycentric coordinates were originally developed by Möbius [Möb27] in the nineteenth century as a special kind of local coordinate that express the location of a point with respect to a given triangle. Wachspress [Wac75] extended the notion of barycentric coordinates to arbitrary convex polygons to construct a finite element basis. In recent years, the research on barycentric coordinates has been intensified and led to a general theory of barycentric coordinates and extensions to higher dimensions [FKR05, FHK06, JSW05, JSWD05, JLW07]. Barycentric coordinates are natural coordinates for polyhedra. Their applications range from shading [Gou71, Pho75] over interpolation [JSW05, LBS06], finite element applications [AO06, SM06, WBG07], generalized Bézier surfaces [LS07, LBS08], and parameterization methods [DMA02, SAPH04, SCOL*04] to space deformations [SP86, JMD*07, LKCOL07] and dimensionality reduction [HSL*06].

In this chapter, we give an introduction to generalized barycentric coordinates and present related work. We introduce a geometric interpretation for them, and in Section 2.5, we recall the definition of a particularly interesting type of generalized barycentric coordinate, the mean value coordinates. That section contains a new proof that mean value coordinates are well defined in the whole plane and serves as a prelude to subsequent chapters where we extend the notion of barycentric coordinates in several directions, as follows: In Chapter 3, we introduce spherical barycentric coordinates. They are analogons of the classical barycentric coordinates, but they are defined for polygons on a sphere instead of a plane. We prove that spherical mean value coordinates are defined for arbitrary polygons on the sphere, and we show that the vector coordinates in [JSWD05] are a special case of our construction. One of the most interesting consequences is the possibility of constructing 3D coordinates for arbitrary polyhedra (Chapter 4). We prove that the 3D mean value coordinates, which are constructed using spherical mean value coordinates, are well defined in the whole $\mathbb{R}^3$ space. Furthermore, we show that the 3D mean value coordinates for triangular polyhedra in [FKR05, JSW05] are a special case of our construction. We conclude this part with an extension of the notion of barycentric coordinates that allows to interpolate derivatives (Chapter 5).

## 2.1 Definition of Generalized Barycentric Coordinates

In this section, we give a general (axiomatic) definition of generalized barycentric coordinates. Since they are the principle theme of this thesis, we will drop the adjective "generalized" in the future. If we want to consider the special case of barycentric coordinates with respect to a triangle (or simplex) as defined by Möbius [Möb27], we will refer to them as *classical barycentric coordinates*.

Let $P = \{\mathbf{v}_j\}_{j=1..n}$ be a point cloud or a polytope with vertices $\mathbf{v}_j \in \mathbb{R}^n$ as shown in Figure 2.1. *Barycentric coordinates* $\lambda_i(\mathbf{x}; P) = \lambda_i\left(\mathbf{x}; \{\mathbf{v}_j\}_{j=1..n}\right)$ of a point $\mathbf{x}$ with respect to $P$ are functions $\lambda_i(\mathbf{x}) : \Omega \to \mathbb{R},\ P \subset \Omega \subset \mathbb{R}^n$, which satisfy the two axioms below.

**Partition of unity.**

$$\forall \mathbf{x} \in \Omega \ \sum_i \lambda_i(\mathbf{x}; P) = 1. \tag{2.1}$$

**Linear precision.**

$$\forall \mathbf{x} \in \Omega \ \sum_i \lambda_i(\mathbf{x}; P)\mathbf{v}_i = \mathbf{x}. \tag{2.2}$$

This property ensures that a point $\mathbf{x}$ can be represented as a linear combination of the vertices $\mathbf{v}_i$ with coefficients $\lambda_i(\mathbf{x}; P)$, justifying the designation of these as coordinates. It is called "linear precision" since it ensures the correct interpolation of all linear functions $f$:

$$\sum_i \lambda_i(\mathbf{x}) f(\mathbf{v}_i) = f(\mathbf{x}). \tag{2.3}$$

Note that a polytope $P$ contains some kind of connectivity information, in addition to the positional information of a pure point cloud $P$. This information may be used in the definition of a specific set of barycentric coordinates to obtain coordinates with enhanced geometrical meaning with respect to the given polytope. We also use the notation $\lambda_i$ or $\lambda_i(\mathbf{x})$ if $P$ and the point $\mathbf{x}$ are obvious from the context. If $P$ is a triangle or simplex, its barycentric coordinates are uniquely determined by the above conditions. For more general $P$, the $\lambda_i$ are no longer unique,

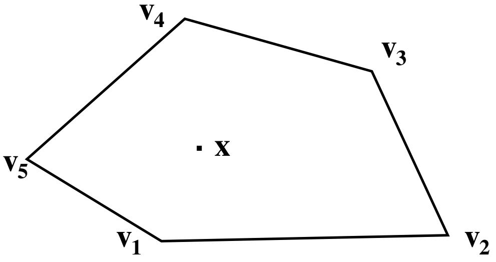


**Figure 2.1: We are looking for coordinates $\lambda_i$ such that $\sum_i \lambda_i \mathbf{v}_i = \mathbf{x}$.**

and we have to consider additional criteria to choose between different coordinate functions. A standard method to construct a set of barycentric coordinates is to construct *homogeneous coordinates* $w_i$ first, which satisfy

$$\sum_i w_i(\mathbf{v}_i - \mathbf{x}) = \mathbf{0}. \tag{2.2'}$$

They can be normalized to fulfill (2.1) as well if and only if $\sum_i w_i(\mathbf{x}) \neq 0$. After normalization, (2.2) and (2.2′) are equivalent, and the coordinates satisfy the linear precision property, too. We will use this method for the construction of barycentric coordinates in Sections 2.3, 2.5, and 4.1.

We conclude this section by listing some other desirable properties for barycentric coordinates.

**Domain.** The domain $\Omega \subset \mathbb{R}^n$ should be as large as possible (possibly depending on $P$), and the $\lambda_i$ should be defined for as general $P$ as possible.

**Non-negativity.**

$$\forall i \; \lambda_i(\mathbf{x}; P) \geq 0. \tag{2.4}$$

We would like to achieve (2.4) for a wide range of points $\mathbf{x}$. This often yields more intuitive behavior of the coordinates. If this property is fulfilled within all convex polytopes, we call the respective $\lambda_i(\mathbf{x})$ *positive barycentric coordinates*.

Note that at least one coordinate must be negative for points $\mathbf{x}$ outside the convex hull of $P$. This is implied by (2.2) since it represents a convex combination for positive $\lambda_i$.

**Smoothness.** The coordinate functions $\lambda_i$ should be as smooth as possible.

**Similarity invariance.** If $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ is a similarity transformation (translation, rotation, reflection, uniform scaling, or a combination of these), then $\forall \mathbf{x} \in \Omega \; \lambda_i\big(\varphi(\mathbf{x}); \varphi(P)\big) = \lambda_i(\mathbf{x}; P)$. This property is important for most of the applications of barycentric coordinates. It is satisfied by all coordinates considered in this thesis.

## 2.2 Related Work

The construction of barycentric coordinates with respect to more than three (or $n + 1$ in $\mathbb{R}^n$) vertices has been an active area of research for a long time. We list some of the suggested coordinates in Table 2.1. The discrete harmonic coordinates have been invented for polygons to create a discrete analog to harmonic functions [PP93]. They have been generalized to polytopes with simplicial faces in [JLW07], and together with the framework in [LBS06], it should be possible to generalize them to arbitrary, convex polytopes. However, it has so far been unclear as to how such a definition should look exactly. The coordinates introduced by Wachspress [Wac75] to construct a finite element basis are defined within arbitrary convex polygons $P \subset \mathbb{R}^2$ and on their boundary. They and their generalizations to convex polytopes [War96, JSWD05] are represented by rational polynomials.

The restriction to convex polytopes was overcome with the introduction of the mean value coordinates [Hui91, Flo03]. Since then, mean value coordinates have enjoyed ever increasing popularity in computer graphics and computational mathematics because they exhibit a variety of good properties. They are defined in the whole plane, for convex and non-convex polygons, and can even be extended to multiple polygons [HF06]. This allows interpolation and extrapolation in a wide range of situations. However, the mean value coordinates are only $C^0$-continuous at the vertices ($C^\infty$ everywhere else). The special case of mean value coordinates in a quadrilateral is explored in [HT04]. Generalizations to higher dimensions are considered in [FKR05, JSW05, JLW07]. But these constructions work only for polytopes with simplicial boundary, and the coordinates are in general not positive with respect to non-convex polytopes. While the positive mean value coordinates [LKCOL07] ensure positivity within arbitrary polytopes, they are no longer smooth. Furthermore, due to the need to compute them on the graphics card (to accelerate computation speed), they are only defined for polyhedra with triangular boundary. Finally, they are only defined within the polytopes and on their boundary. Although they could be defined outside the polytope, they lose the linear precision property at these points, and can no longer be called coordinates. Another solution was proposed by Joshi et al. [JMD*07] with the harmonic coordinates which are computed as a numerical solution of the Laplace equation. This guarantees positivity and $C^\infty$-smoothness within arbitrary polytopes ($C^0$ at the boundary) and allows the specification of additional vertices, but leads to relatively high computation times. An extension of these coordinates to points outside the polytope has been suggested in [Rus07]. A classification of all homogeneous coordinates for convex polytopes is given in [FHK06, JLW07].

An interesting alternative to the coordinates mentioned so far are the Green coordinates [LLCO08]. They are not barycentric coordinates in our sense as they

**Table 2.1: Properties of some well-known coordinates. We state the domain of the coordinates, we indicate whether the coordinates are positive within convex polytopes (that is, whether they are positive coordinates) and within arbitrary polytopes (coordinates that are not defined for non-convex polytopes are marked with "not applicable", n/a), and we specify the degree of smoothness for the coordinate functions within a polytope and at its vertices.[2] For a point cloud $P$, positivity is specified for points in the open convex hull of $P$ excluding the vertices, $C(P) \setminus P$, and smoothness for points in $C(P)$ and at the vertices, respectively. We compare the following coordinates: the discrete harmonic coordinates [PP93, JLW07], the Wachspress coordinates [MLBD02, JSWD05], the mean value coordinates [Flo03, LBS06], the positive mean value coordinates [LKCOL07], the harmonic coordinates [JMD*07, Rus07], and the Sibson coordinates [Sib80, HS00]. Here, we give references to the most general and easily implementable definitions. These are not the original sources in all cases. Further information can be found in the text.**

| | Domain | Positivity | Smoothness |
|---|---|---|---|
| Discrete Harmonic | $\mathbf{x} \in P,\ P$ convex polytope | $-$/n/a | $C^\infty/C^{\geq 0}$ |
| Wachspress | $\mathbf{x} \in P,\ P$ convex polytope | $+$/n/a | $C^\infty/C^\infty$ |
| Mean Value | $\mathbf{x} \in \mathbb{R}^n,\ P$ arbitrary polytope | $+/-$ | $C^\infty/C^0$ |
| Pos. Mean Value | $\mathbf{x} \in P,\ P$ arbitrary polytope | $+/+$ | $C^0/C^0$ |
| Harmonic | $\mathbf{x} \in \mathbb{R}^n,\ P$ arbitrary polytope | $+/+$ | $C^\infty/C^0$ |
| Sibson | $\mathbf{x} \in C(P),\ P$ point cloud | $+$ | $C^1/C^0$ |

not only require a coordinate for each vertex, but for each face as well. However, they allow the definition of an approximating function, which is, in a certain sense, shape-preserving.

Note that all of the above coordinates depend not only on the vertices $\mathbf{v}_i$, but also on a polytope whose vertices are the $\mathbf{v}_i$; that is, additional connectivity information is required. Coordinates that are based on natural neighbors in a Voronoi diagram instead are the Laplacian or Non-Sibsonian [BIK*97] and the Sibson coordinates [Sib80]. Sibson coordinates are $C^0$-continuous at the vertices, $C^1$-continuous on the Delaunay spheres, and $C^\infty$ elsewhere. They are at the beginning of the family of the Hiyoshi standard natural neighbor coordinates [HS00, BBU06], which exhibit $C^k$-smoothness at the Delaunay spheres. In [ST04], it is shown that Laplacian coordinates and Wachspress coordinates coincide on regular $n$-gons. An overview of barycentric coordinates from a finite

[2] The smoothness of discrete harmonic coordinates at the vertices is not known to the best of our knowledge. However, it is at least $C^0$.

element point of view is given in [SM06].

Spherical barycentric coordinates constitute another variant of barycentric coordinates. Analogous to planar barycentric coordinates that describe the positions of points in a plane with respect to the vertices of a given planar polygon, spherical barycentric coordinates describe the positions of points on a sphere with respect to the vertices of a given spherical polygon. They were first studied by Möbius [Möb46] with respect to spherical triangles, and were introduced to computer graphics by Alfeld et al. [ANS96]. Ju et al. [JSWD05] extended spherical barycentric coordinates (they called them "vector coordinates") from spherical triangles to arbitrary convex, spherical polygons. However, non-convex spherical polygons still posed a problem. The extension to arbitrary, spherical polygons is the topic of Chapter 3.

## 2.3 A Geometric Interpretation for Planar Barycentric Coordinates

We present a unified, geometric, and intuitive construction that explains the "linear precision" property of an especially interesting one-parameter family of barycentric coordinates that was introduced by Floater et al. [FHK06]. A different, but equivalent, approach was recently presented in [SJW07]. With this construction, we can derive analogons of the discrete harmonic, mean value, and Wachspress coordinates for arbitrary dimensions. In this thesis, we use it in Sections 3.2 and 3.3.

Our construction is indicated in Figure 2.2. Consider a polygon with vertices $\mathbf{v}_i$. It is always possible to construct a dual polygon (that may have self-intersections) with respect to a vertex $\mathbf{x}$ whose edges are orthogonal to the line segments $\mathbf{xv}_i$ and whose vertices are given by the intersection points of two consecutive edges. In fact, there are infinitely many dual polygons since we can choose the intersection point of the dual edges with the line given by $\mathbf{xv}_i$ freely. Since the normals of the dual edges are given by the edges $\mathbf{xv}_i$, we can employ a theorem of Minkowski which states that the sum over the unit edge normals of a polygon, weighted with the respective edge lengths, is zero. Hence, the lengths $a_i$ of the dual edges yield homogeneous coordinates $w_i$ for $\mathbf{x}$ that satisfy (2.2′). However, the edges $\mathbf{xv}_i$ do not have unit length in general. Therefore, the exact relationship between $a_i$ and $w_i$ is $a_i = r_i w_i$ with $r_i = ||\mathbf{xv}_i||$. Negative weights are assigned to inversely oriented dual edges. The barycentric coordinates $\overline{\lambda}_i$ can be obtained from the homogeneous coordinates $w_i$ by $\overline{\lambda}_i = \frac{w_i}{\sum_j w_j}$.

In Figure 2.2, three particular choices for the intersection point of the dual edges are depicted. On the left, the dual edges pass through the points $\mathbf{v}_i$, in the

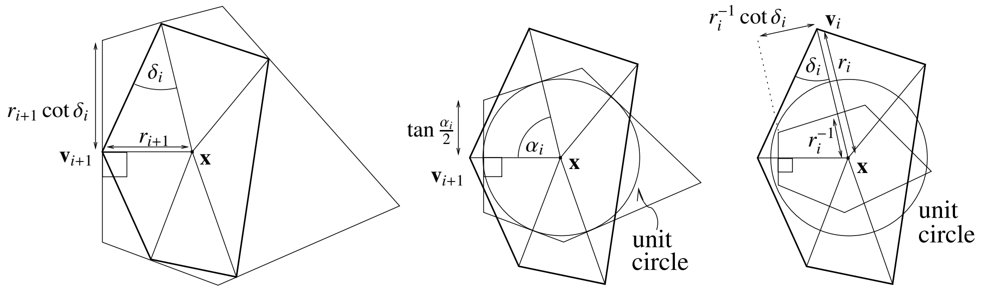


**Figure 2.2: A polygon (bold) and three of its duals. These constructions yield geometric definitions of the discrete harmonic, mean value, and Wachspress coordinates (from left to right).**

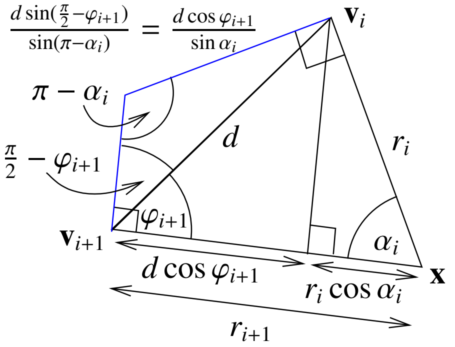


**Figure 2.3: We compute the edge length of a part of the dual polygon for the discrete harmonic coordinates (shown in blue), $\frac{d\sin(\frac{\pi}{2}-\varphi_{i+1})}{\sin(\pi-\alpha_i)} = \frac{r_{i+1}-r_i\cos\alpha_i}{\sin\alpha_i}$.**

middle, the dual edges have constant distance to $\mathbf{x}$, and on the right, the distance of the dual edges to $\mathbf{x}$ is $r_i^{-1}$. Using a little trigonometry, it is easy to show that these choices correspond to the standard formulae for discrete harmonic, mean value, and Wachspress coordinates [PP93, Flo03, MLBD02]. In fact, the respective instances of this construction have been used to derive the discrete harmonic coordinates (this is indicated in Section 9.3.2) and to extend the definition of Wachspress coordinates to higher dimensions [JSWD05].

Now, it is natural to ask what kind of coordinates are obtained if the distance of the dual edges to $\mathbf{x}$ is chosen as $r_i^p$, $p \in \mathbb{R}$. The answer, given as a formula, is

$$w_{i,p} = \frac{1}{r_i}\left(\frac{r_{i-1}^p - r_i^p\cos\alpha_{i-1}}{\sin\alpha_{i-1}} + \frac{r_{i+1}^p - r_i^p\cos\alpha_i}{\sin\alpha_i}\right), \quad \lambda_{i,p} = \frac{w_{i,p}}{\sum_i w_{i,p}},$$

compare Figure 2.3.

If we compare this to the one-parameter family of barycentric coordinates $\widehat{w}_{i,p}$ from [FHK06], we see that $w_{i,p} = \frac{1}{2}\widehat{w}_{i,p+1}$. Therefore, both families generate the same barycentric coordinates (after normalization), and the analysis by Floater et al. applies to our family as well:

**2.1 Corollary ([FHK06]).** *The only members of the one-parameter family $w_{i,p}$ which are positive for all convex polygons are the Wachspress and the mean value coordinates.*

## 2.4 Coordinates in 3D

An appealing property of the geometric construction in the previous section is that it can easily be generalized to higher dimensions, and barycentric coordinates for polytopes with simplicial boundary can be derived. In the three-dimensional case, the analogous weights $w_{i,-1}$ lead to three-dimensional Wachspress coordinates for arbitrary convex polyhedra, see [Grü67, JSWD05]. For general $p$, however, the dual polyhedron and the corresponding weights $w_{i,p}$ can only be constructed for polyhedra with triangular faces since four or more (dual) faces, corresponding to a polygonal face with four or more vertices, do not intersect in a single point in general. Furthermore, the dual polyhedron may have self-intersections, which can lead to negative coordinates even for convex polyhedra if $p \neq -1$. The corresponding weights $w_{i,p}$ for $p = 0$ and $p = 1$ do not yield the mean value and the discrete harmonic coordinates in 3D. For these reasons, we do not consider this construction as a useful generalization of mean value coordinates and discrete harmonic coordinates. We will develop a more suitable generalization in the subsequent chapters.

For the sake of completeness, we give a closed formula for the weights $w_{i,p}$ nevertheless.

$$w_{i,p}(\mathbf{x}) = \frac{1}{r_i} \sum_{j \in N(i)} w_{i,j,p}(\mathbf{x}),$$

$$w_{i,j,p}(\mathbf{x}) = \frac{d_{j,p} d_{j+1,p} - \frac{1}{2}(d_{j,p}^2 + d_{j+1,p}^2)\cos\alpha_j}{\sin\alpha_j}, \quad d_{j,p} = \frac{r_j^p - r_i^p \cos\theta_j}{\sin\theta_j}$$

where $r_i$ is the length of $\mathbf{v}_i - \mathbf{x}$; $\theta_j$ is the angle between $\mathbf{v}_i - \mathbf{x}$ and $\mathbf{v}_j - \mathbf{x}$; $\alpha_j$ is the dihedral angle between the planes given by $\mathbf{x}, \mathbf{v}_i, \mathbf{v}_j$ and $\mathbf{x}, \mathbf{v}_i, \mathbf{v}_{j+1}$; $w_{i,j,p}(\mathbf{x})$ is the area of the quad determined by $d_{j,p}$, $d_{j+1,p}$, and $\alpha_j$, see Figure 2.4 for the case $p = 1$.

## 2.5 Planar Mean Value Coordinates

In this section, we consider a particularly interesting instance of barycentric coordinates, the mean value coordinates [Flo03]. We recall their definition and a

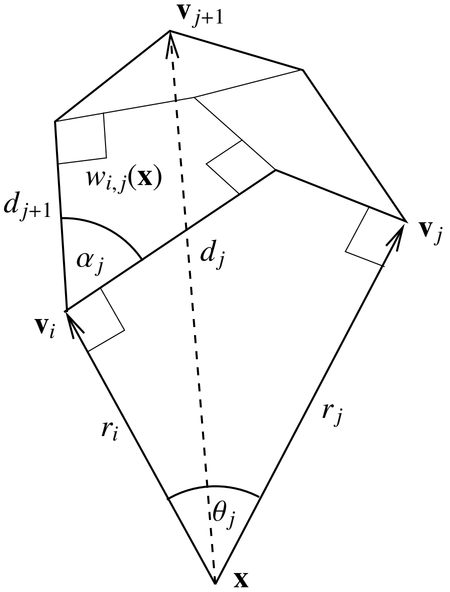


**Figure 2.4: Notation for the construction of 3D barycentric coordinates using dual polyhedra.**

theorem that they are well defined [HF06]. We use a different presentation of the definition than Floater and present an alternative proof compared with the one given by Hormann and Floater. Our representation has the advantage of being easily extendable to higher dimensions and serves as a building block for the results in the following chapters.

**2.2 Definition.** A (planar) *polygon* $P = (\mathbf{v}_1 \dots \mathbf{v}_n)$ is given by a finite sequence of distinct vertices $\mathbf{v}_i \in \mathbb{R}^2$ such that its edges $(\mathbf{v}_i,\ \mathbf{v}_{i+1}),\ i = 1 \dots n$ do not intersect. For an edge $\mathbf{e}$, $V(\mathbf{e})$ denotes the (two-element) set of indices $i$ such that $\mathbf{v}_i$ is incident to $\mathbf{e}$, and $V(P)$ denotes the set of all vertex indices.

Note that this definition is not purely geometrical, but also contains some combinatorial structure, represented by the edges.

**2.3 Algorithm (planar mean value coordinates).** Mean value coordinates $\overline{\lambda}_i$ for a point $\mathbf{x} \in \mathbb{R}^2$ with respect to a polygon $P$ can be defined in the following way [Flo03]:

- An edge vector $\mathbf{v}_\mathbf{e}$ is assigned to each edge $\mathbf{e}$ of the polygon such that $\sum_\mathbf{e} \mathbf{v}_\mathbf{e} = \mathbf{0}$. ($\mathbf{v}_\mathbf{e}$ can be considered as some kind of edge normal.)

  For an edge $\mathbf{e} = (\mathbf{v}_i,\ \mathbf{v}_{i+1})$, let $\mathbb{S}^1_\mathbf{e} \subset \mathbb{S}^1$ be the oriented circular arc with end points $\frac{\mathbf{v}_i - \mathbf{x}}{\|\mathbf{v}_i - \mathbf{x}\|}$ and $\frac{\mathbf{v}_{i+1} - \mathbf{x}}{\|\mathbf{v}_{i+1} - \mathbf{x}\|}$. Let $\mathbf{n} : \mathbb{S}^1 \to \mathbb{R}^2$ be the outward unit normal vector of the circle. Then $\mathbf{v}_\mathbf{e}$ is defined as the integral of $\mathbf{n}$ over $\mathbb{S}^1_\mathbf{e}$, see Figure 2.5 (left):

$$\mathbf{v}_\mathbf{e} := \int_{\mathbb{S}^1_\mathbf{e}} \mathbf{n} \, \mathrm{d}S.$$

- The edge vectors are distributed to their respective edge vertices by the unique weights $\mu_{\mathbf{e},j}$ such that $\mu_{\mathbf{e},i}(\mathbf{v}_i - \mathbf{x}) + \mu_{\mathbf{e},i+1}(\mathbf{v}_{i+1} - \mathbf{x}) = \mathbf{v}_\mathbf{e}$ for an edge $\mathbf{e} = (\mathbf{v}_i, \ \mathbf{v}_{i+1})$.

- The weights at each vertex $\mathbf{v}_i$ are cumulated as $w_i := \mu_{\mathbf{e}_{i-1},i} + \mu_{\mathbf{e}_i,i}$ where $\mathbf{e}_i$ denotes the edge $(\mathbf{v}_i, \ \mathbf{v}_{i+1})$.

- The weights are normalized to form a partition of unity:

$$\overline{\lambda}_i := \frac{w_i}{\sum_j w_j}. \tag{2.5}$$

It is straightforward to show that the definition of $\mathbf{v}_\mathbf{e}$ has the closed form solution [Flo03]

$$\mathbf{v}_\mathbf{e} = \tan\frac{\alpha_i}{2}\Big(\frac{\mathbf{v}_i - \mathbf{x}}{\|\mathbf{v}_i - \mathbf{x}\|} + \frac{\mathbf{v}_{i+1} - \mathbf{x}}{\|\mathbf{v}_{i+1} - \mathbf{x}\|}\Big) \tag{2.6}$$

where $\alpha_i$ denotes the angle between $\mathbf{v}_i - \mathbf{x}$ and $\mathbf{v}_{i+1} - \mathbf{x}$. Distributing these edge vectors to the vertices yields the formula proposed earlier [Flo03]:

$$w_i = \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{\|\mathbf{v}_i - \mathbf{x}\|}. \tag{2.7}$$

**2.4 Remark.** It is possible to choose different edge vectors $\mathbf{v}_\mathbf{e}$ such that $\sum_\mathbf{e} \mathbf{v}_\mathbf{e} = \mathbf{0}$. In all cases, the construction guarantees that the $w_i$ are homogeneous coordinates. Hence, the $\overline{\lambda}_i$ are barycentric coordinates. Possible choices for the edge vectors are discussed in [JLW07].

To show that the above defined mean value coordinates are well defined, it is necessary to ensure that the denominator in the normalization step (2.5) cannot become zero. As shown in [HF06], this can be done in two steps:

- A refinement lemma is proven that states that the denominator does not change if we refine our polygon by including additional vertices.

- For a particular refinement of the polygon, it is shown that the denominator does not vanish.

We will give an alternative proof for the first step. It has the advantage of being more general than the original proof in [HF06], and it can be easily generalized to higher dimensions. The second step proceeds as in [HF06].

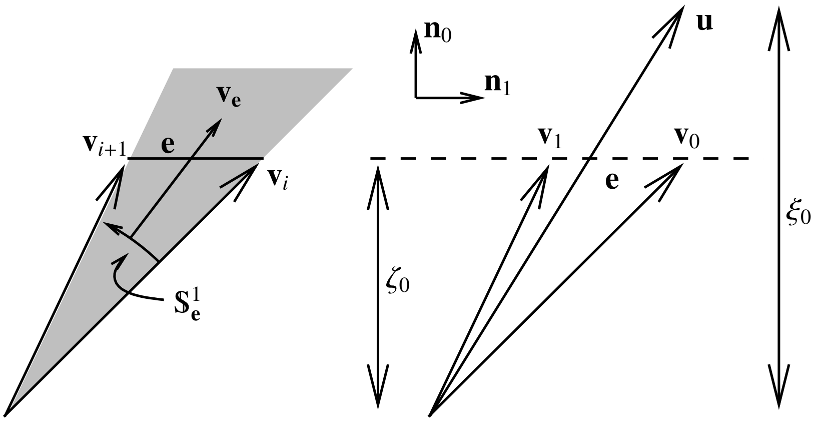


**Figure 2.5: Left: Definition of the edge vector $\mathbf{v}_{\mathbf{e}}$. Right: Notation for Lemmas 2.5 and 2.6.**

**2.5 Lemma.** *Let $\mathbf{u} \in \mathbb{R}^2$ be a point, let $\mathbf{v}_j \in \mathbb{R}^2$ be a set of points that lie on a common line $\mathbf{e}$ that does not contain the origin, and let $\mathbf{u} = \sum_j \mu_j \mathbf{v}_j$ for some coefficients $\mu_j$. Denote by $(\mathbf{n}_k)_{k=0,1}$ an orthonormal basis of $\mathbb{R}^2$ such that $\mathbf{n}_0$ is the normal vector of $\mathbf{e}$, and let $\xi_k$ and $\zeta_{jk}$ be the coefficients of $\mathbf{u}$ and $\mathbf{v}_j$, respectively: $\mathbf{u} = \sum_k \xi_k \mathbf{n}_k$, $\mathbf{v}_j = \sum_k \zeta_{jk} \mathbf{n}_k$. We define $\zeta_0 := \zeta_{j0} > 0$. (All $\zeta_{j0}$ are equal since the $\mathbf{v}_j$ lie on a common line with normal $\mathbf{n}_0$; the last inequality can be achieved by choosing the orientation of $\mathbf{n}_0$ appropriately.) This setting is sketched in Figure 2.5 (right).*

*Then $\sum_j \mu_j = \frac{\xi_0}{\zeta_0}$.*

*Proof.* We have

$$\mathbf{u} = \sum_j \mu_j \mathbf{v}_j = \sum_j \mu_j \zeta_0 \mathbf{n}_0 + \sum_j \mu_j \zeta_{j1} \mathbf{n}_1 .$$

Since $\mathbf{u}$ has a unique representation in the basis $(\mathbf{n}_k)$, the claim follows. □

**2.6 Lemma.** *In the situation of Lemma 2.5, assume that $\mathbf{u}$ is given as the sum of points, $\mathbf{u} = \sum_i \mathbf{u}_i$, $\mathbf{u}_i = \sum_j \mu_{ij} \mathbf{v}_j \in \mathbb{R}^2$.*

*Then $\sum_j \mu_j = \sum_{ij} \mu_{ij}$.*

*Proof.* Let $\mathbf{u}_i = \sum_k \xi_{ik} \mathbf{n}_k$ be the representations of $\mathbf{u}_i$ in the basis $(\mathbf{n}_k)$. From the unique representation of $\mathbf{u} = \sum_k \xi_k \mathbf{n}_k$ in this basis and the fact that $\mathbf{u} = \sum_i \mathbf{u}_i$, we can conclude $\xi_0 = \sum_i \xi_{i0}$. Using Lemma 2.5 for $\mathbf{u}$ and the $\mathbf{u}_i$, we obtain

$$\sum_j \mu_j = \frac{\xi_0}{\zeta_0} = \sum_i \frac{\xi_{i0}}{\zeta_0} = \sum_{ij} \mu_{ij} .$$

□

**2.7 Definition.** A *refinement* $\widehat{P} = (\widehat{\mathbf{v}}_i)_{i=1..m}$ of a polygon $P = (\mathbf{v}_i)_{i=1..n}$ is a polygon such that $(\mathbf{v}_i)_{i=1..n}$ is a subsequence of $(\widehat{\mathbf{v}}_i)_{i=1..m}$, all additional vertices $\widehat{\mathbf{v}}_i$ of $\widehat{P}$ lie on edges of $P$ ( $\widehat{\mathbf{v}}_i \in (\mathbf{v}_j, \mathbf{v}_{j+1})$ for some $j$ ) and $P$ and $\widehat{P}$ bound the same area in $\mathbb{R}^2$.

This means that a refinement of a polygon leaves the geometry of the polygon unchanged, only the combinatorial structure is refined. As desired, this does not change the mean value coordinates essentially:

**2.8 Lemma (refinement of planar polygons).** *Let $P$ be a polygon, and let $\widehat{P}$ be a refinement of $P$. Let $w_i$ and $\widehat{w}_i$ be the weights in step 3 of Algorithm 2.3 for $P$ and $\widehat{P}$.*

*Then $\sum_{i \in V(P)} w_i = \sum_{i \in V(\widehat{P})} \widehat{w}_i$.*

*Proof.* Let $\mathbf{e}$ be an edge of $P$, and let $\widehat{E}$ be the set of edges in $\widehat{P}$ that compose the refinement of $\mathbf{e}$. From the definition of $\mathbf{v}_\mathbf{e}$, it is obvious that $\mathbf{v}_\mathbf{e} = \sum_{\hat{\mathbf{e}} \in \widehat{E}} \mathbf{v}_{\hat{\mathbf{e}}}$. Therefore, Lemma 2.6 implies that $\sum_{i \in V(\mathbf{e})} \mu_{\mathbf{e},i} = \sum_{\hat{\mathbf{e}} \in \widehat{E},\ i \in V(\hat{\mathbf{e}})} \widehat{\mu}_{\hat{\mathbf{e}},i}$. By taking the sum over all edges $\mathbf{e}$, we obtain the claim. □

Note, that this lemma is not restricted to mean value coordinates but applies to all kinds of coordinates as defined in [FHK06]. The reason is that the edge vector for an edge $\mathbf{e}$ can always be expressed as the sum of the edge vectors of its refinement [JLW07]. The following lemma, in contrast, is in general not true if applied to other coordinates than mean value coordinates.

**2.9 Lemma.** *Let $\mathbf{e} = (\mathbf{v}_i, \mathbf{v}_{i+1})$ be an edge of a polygon $P$. Let $\mu_{\mathbf{e},i}$ and $\mu_{\mathbf{e},i+1}$ be the coefficients of its edge vector $\mathbf{v}_\mathbf{e}$.*

*Then $\mu_{\mathbf{e},i} + \mu_{\mathbf{e},i+1}$ is greater than zero if and only if $\mathbf{S}^1_\mathbf{e}$ is positively oriented.*

*Proof.* Without loss of generality let $\mathbf{S}^1_\mathbf{e}$ be positively oriented. Then $\mathbf{v}_\mathbf{e}$ is contained in the cone defined by the convex hull of the $\mathbf{v}_j - \mathbf{x},\ j \in V(\mathbf{e}) = \{i, i+1\}$. (The cone is indicated by the shaded area in the left part of Figure 2.5.) Since all the $\zeta_{j0} = \zeta_0$ are greater than zero by definition of $\mathbf{n}_0$, $\xi_0$ must be greater than zero as well. Therefore, the claim follows from Lemma 2.5. □

**2.10 Theorem ([HF06]).** *Let $P$ be a polygon. Then the planar mean value coordinates with respect to $P$ are well defined in $\mathbb{R}^2$.*

*Proof.* Consider the denominator $W := \sum_i w_i$ in Equation (2.5). We have to show that $W$ does not vanish if computed for an arbitrary $\mathbf{x} \in \mathbb{R}^2$. We refine $P$ by adding all intersection points of rays from $\mathbf{x}$ through the vertices $\mathbf{v}_i$ with the edges of $P$, see Figure 2.6. According to Lemma 2.8, this does not change $W$. If we now split the denominator of the refined polygon into partial sums $\mu_\mathbf{e} := \sum_{i \in V(\mathbf{e})} \mu_{\mathbf{e},i}$ associated to the edges $\mathbf{e}$, we know from Lemma 2.9 that the sign of these partial

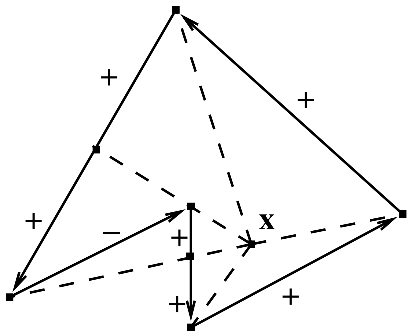


**Figure 2.6: Refinement of a polygon such that all edges with negative weight (-) are counterbalanced by an edge with positive weight (+)**

sums depends only on the orientation of $\mathbb{S}^1_{\mathbf{e}}$. From (2.6), we can see that the absolute value of the partial sums decreases with increasing distance of **e** to **x**. By grouping together edges that lie behind each other as seen from **x**, $W = \sum_{\mathbf{e}} \mu_{\mathbf{e}}$ splits into sums of alternating sign and decreasing absolute value starting with a positive number if **x** is inside and a negative number if **x** is outside the polygon. Hence, $W$ is positive inside and negative outside the polygon. In particular, $W$ is not equal to zero, and the theorem is proven. (For points **x** on an edge, a continuous extension of the definition in Algorithm 2.3 exists.) For further details see [HF06]. □

# Chapter 3

# Spherical Barycentric Coordinates

In this chapter, we introduce spherical barycentric coordinates. Analogous to classical, planar barycentric coordinates that describe the positions of points in a plane with respect to the vertices of a given planar polygon, spherical barycentric coordinates describe the positions of points on a sphere with respect to the vertices of a given spherical polygon. We show how arbitrary barycentric coordinates from the family of planar barycentric coordinates [FHK06] can be defined on a sphere (Section 3.1). In particular, we consider the special cases of spherical Wachspress coordinates (Section 3.2) and spherical mean value coordinates (Section 3.3), which inherit many good properties from their planar counterparts. We show that the former coincide with the vector coordinates in [JSWD05], which are defined for convex spherical polygons, and that the latter are defined for arbitrary polygons on the whole sphere.

One of the most interesting consequences is the possibility of constructing 3D barycentric coordinates for arbitrary polyhedra. This is the topic of Chapter 4. Furthermore, spherical barycentric coordinates can be used for all applications, such as Bézier surfaces and parameterization, where up until now only planar barycentric coordinates were available.

## 3.1 Definition of Spherical Barycentric Coordinates

In this section, we deal with the problem of finding (positive) coefficients $\lambda_i$ for vectors $\mathbf{v}_i \in \mathbb{R}^3$ such that their linear combination yields a given vector $\mathbf{v} \in \mathbb{R}^3$. It is sufficient to restrict ourselves to vectors of unit length. In this case, the $\lambda_i$ represent barycentric coordinates for $\mathbf{v}$ with respect to the $\mathbf{v}_i$ on the unit sphere.

First, we give a general introduction.

While Equations (2.1) and (2.2) are well-chosen to characterize planar barycentric coordinates, it is obviously not possible to fulfill both conditions if the vertices $\mathbf{v}_i$ and the point $\mathbf{x} = \mathbf{v}$ are located on a sphere instead of a plane. This is especially easy to see if $P$ is a triangle with three vertices. (2.1) requires all points described by $\sum_i \lambda_i \mathbf{v}_i$ to lie in the triangle plane while (2.2) demands that this sum yields a point $\mathbf{v}$ that does not lie in this plane but on the sphere. A similar observation was made in [BW92].

Consequently, we have to relax the above conditions. We follow the suggestion by Alfeld et al. [ANS96] and replace (2.1) by

$$\sum_i \lambda_i(\mathbf{x}) \geq 1 \tag{2.1'}$$

for $\mathbf{x}$ within convex polygons. Of course, by dividing by $\sum_j \lambda_j$, we could also obtain coordinates that constitute a partition of unity (2.1) instead of satisfying the linear precision property (2.2) if desired. However, for our applications, in particular for constructing barycentric coordinates with respect to polyhedra, linear precision is more important. Note that this property still implies that linear functions *defined on* $\mathbb{R}^3$ are correctly interpolated (2.3). Constant functions, however, cannot be correctly interpolated if a partition of unity (2.1) is not given. Nevertheless, these coordinates can be used for exact interpolation of spherical harmonics of degree one. A different approach that preserves the partition of unity was proposed in [BF01]. We call a set of coordinates $\lambda_i$ satisfying conditions (2.1′) and (2.2) *spherical barycentric coordinates*. If $P$ is a triangle, a unique solution exists obviously: the unique linear combination of the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ such that $\sum_{i=1}^{3} \lambda_i \mathbf{v}_i = \mathbf{v}$. A geometric interpretation of these spherical barycentric coordinates was given in [ANS96].

To proceed, we need to define a spherical polygon.

**3.1 Definition.** A *spherical polygon* $P = (\mathbf{v}_1 \dots \mathbf{v}_n)$ consists of a finite sequence of distinct vertices $\mathbf{v}_i \in \mathbb{S}^2$ located on a sphere such that its edges $(\mathbf{v}_i, \mathbf{v}_{i+1})$, $i = 1 \dots n$ do not intersect. The edges are geodesic lines (these are the arcs of great circles on the sphere) that connect the vertices $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$.

It is *admissible* if it contains no antipodal points on vertices or edges.

We consider a spherical polygon on the unit sphere centered at the origin. Let $\mathbf{v}$ be a point on the sphere. Let $\overline{\mathbf{v}}_i$ be the intersection points of the tangent plane $T_{\mathbf{v}}\mathbb{S}^2$ at $\mathbf{v}$ to the sphere and the line passing through the origin and $\mathbf{v}_i$ (see Figure 3.1). The points $\overline{\mathbf{v}}_i$ determine a polygon $\overline{P}$ (shown in boldface) in the plane $T_{\mathbf{v}}\mathbb{S}^2$. (The map $\mathbf{v}_i \mapsto \overline{\mathbf{v}}_i$ is a gnomonic projection. It is especially useful for our purpose since it projects geodesics to straight lines.) Now, we can compute the

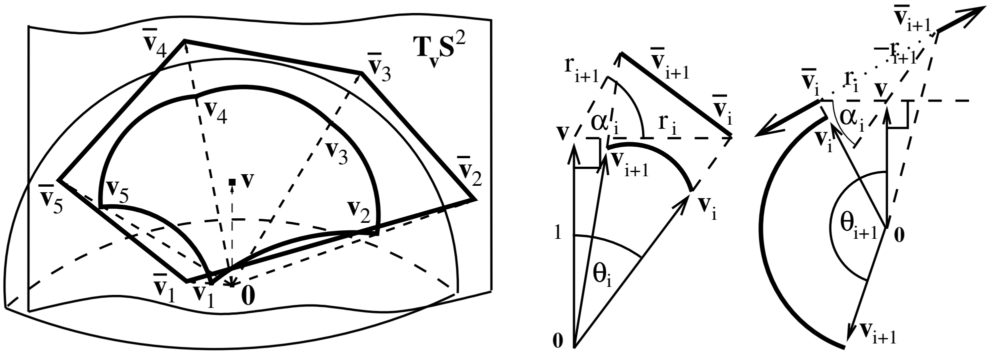


**Figure 3.1: Projection of a spherical polygon to the tangent plane at v.**

planar barycentric coordinates $\overline{\lambda}_i$ of $\mathbf{v}$ with respect to $\overline{P}$. The 3D position of $\mathbf{v}$ is an affine linear function on $T_\mathbf{v}\mathbb{S}^2$. Consequently, any set $\overline{\lambda}_i$ of planar barycentric coordinates yields, by Equations (2.1) and (2.2),

$$\sum_i \overline{\lambda}_i \overline{\mathbf{v}}_i = \mathbf{v}.$$

To obtain spherical barycentric coordinates $\lambda_i(\mathbf{v})$ that satisfy the linear precision property (2.2), we define them by

$$\lambda_i := \langle \mathbf{v}_i, \overline{\mathbf{v}}_i \rangle \overline{\lambda}_i, \qquad \sum_i \lambda_i \mathbf{v}_i = \mathbf{v} \tag{3.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the usual scalar product in $\mathbb{R}^3$. Note that $\langle \mathbf{v}_i, \overline{\mathbf{v}}_i \rangle$ is just $\pm\|\overline{\mathbf{v}}_i\|$. Although this value becomes very large and finally undefined if the angle $\theta_i$ between $\mathbf{v}$ and $\mathbf{v}_i$ approaches $\frac{\pi}{2}$, this is usually compensated by a shrinkage of $\overline{\lambda}_i$ such that the definition of $\lambda_i$ can be extended continuously to the case $\theta_i = \frac{\pi}{2}$. We demonstrate this in Sections 3.2 and 3.3 for spherical Wachspress and spherical mean value coordinates. Note that basically the same construction can be used to obtain planar barycentric coordinates from spherical barycentric coordinates. It follows that there is a bijective correspondence between planar barycentric coordinates and spherical barycentric coordinates.

Finally, we remark that these coordinates can also be extended to vectors $\mathbf{v}_i$ and $\mathbf{v}$ of arbitrary length by defining

$$\lambda_i \Big( \mathbf{v}; (\mathbf{v}_j)_{j=1\ldots n} \Big) := \frac{\|\mathbf{v}\|}{\|\mathbf{v}_i\|} \cdot \lambda_i \Big( \tfrac{\mathbf{v}}{\|\mathbf{v}\|}; \big( \tfrac{\mathbf{v}_j}{\|\mathbf{v}_j\|} \big)_{j=1\ldots n} \Big). \tag{3.2}$$

## 3.2 Spherical Wachspress Coordinates

In this section, we show that the vector coordinates, introduced by Ju et al. [JSWD05] to express a vector $\mathbf{v}$ as a linear combination of other vectors $\mathbf{v}_i$, coin-

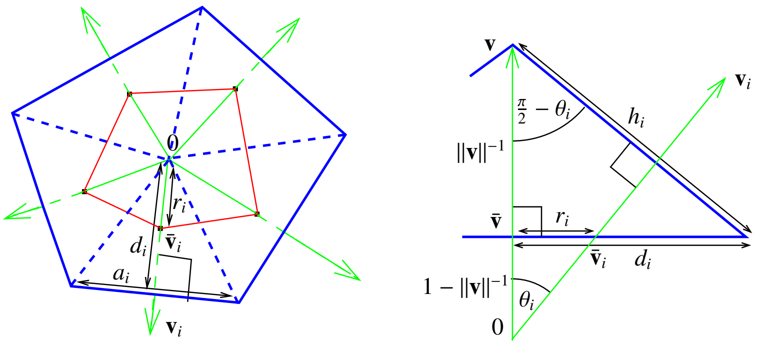


**Figure 3.2: Bottom and side view of the polar dual.**

cide with our spherical Wachspress coordinates. It is sufficient to consider only vectors $\mathbf{v}_i$ and $\mathbf{v}$ of unit length and to show that the coordinates coincide up to a constant factor $c$. Then, it follows from (3.2) and (2.2), which hold for both sets of coordinates, that they coincide for arbitrary vectors and that $c = 1$.

By (3.1) and Figure 3.1, spherical Wachspress coordinates are given by

$$\lambda_i = \frac{1}{\cos\theta_i}\overline{\lambda}_i \tag{3.3}$$

where $\overline{\lambda}_i$ are the planar Wachspress coordinates for the planar polygon with vertices $\overline{\mathbf{v}}_i$ constructed in Section 3.1. We show now that the same formula holds for the vector coordinates constructed in [JSWD05].

Ju et al.'s vector coordinates are defined for vectors $\mathbf{v}_i$ that are the vertices of a convex spherical polygon. The coordinates are proportional to the area $\beta_i$ of the triangular faces of the polar dual. The polar dual is the convex polyhedron that is bounded by the planes that have $\mathbf{v}_i$ as normal and pass through the point $\mathbf{v}$ and by the plane perpendicular to $\mathbf{v}$ with distance $\frac{1}{\|\mathbf{v}\|}$ to the point $\mathbf{v}$ (see Figure 3.2; the polar dual is shown in blue). Let $\tilde{\mathbf{v}}$ be the intersection point of $\mathbf{v}$ and the latter plane. Let $\tilde{\mathbf{v}}_i$ be the intersection points of the same plane and the rays (green) determined by the $\mathbf{v}_i$. Let $\tilde{P}$ be the polygon (shown in red) formed by the $\tilde{\mathbf{v}}_i$. Then $\tilde{\mathbf{v}}$ and $\tilde{P}$ are, by construction, similar to $\mathbf{v}$ and $\overline{P}$, the polygon defined in Section 3.1. Therefore, the respective Wachspress coordinates coincide: $\overline{\lambda}_i(\tilde{\mathbf{v}}; \tilde{P}) = \overline{\lambda}_i(\mathbf{v}; \overline{P})$. Note that the boundary polygon $Q$ (solid blue in Figure 3.2, left) of the bottom face of the polar dual is dual to $\tilde{P}$ (red) with respect to $\tilde{\mathbf{v}}$. That is, its edges $\mathbf{a}_i$ are orthogonal to $\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}$ (see Chapter 2.3 for dual polygons).

The triangle areas $\beta_i$ can be computed (up to a factor $\frac{1}{2}$) as the product of the length $a_i$ of the edge $\mathbf{a}_i$ and the respective height $h_i$. The latter can be computed as $h_i = \frac{\|\mathbf{v}\|^{-1}}{\cos(\frac{\pi}{2}-\theta_i)} = \frac{\|\mathbf{v}\|^{-1}}{\sin\theta_i}$. In the bottom face, the distance $d_i$ of the edge $\mathbf{a}_i$ to the center $\tilde{\mathbf{v}}$ is given by $\|\mathbf{v}\|^{-1}\tan(\frac{\pi}{2} - \theta_i) = \|\mathbf{v}\|^{-1}\cot\theta_i$, and the distance of the intersection points $\tilde{\mathbf{v}}_i$ to $\tilde{\mathbf{v}}$ is given by $r_i = (1 - ||\mathbf{v}||^{-1})\tan\theta_i$. The product $d_i r_i = \frac{||\mathbf{v}||-1}{||\mathbf{v}||^2}$ is independent of $i$. Therefore, $Q$ is (up to scaling) that dual polygon of $\tilde{P}$ such that the distance $d_i$ of the edges $\mathbf{a}_i$ to the center $\tilde{\mathbf{v}}$ is inverse to the distance $r_i$ between $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}$ (up to a constant factor). The edges of such a polygon have lengths $a_i$ proportional to $r_i\overline{\lambda}_i$ (see Chapter 2.3). If we put everything together, we obtain the following formula for the vector coordinates

$$\beta_i = \frac{1}{2}a_i h_i = c_1 r_i h_i \overline{\lambda}_i = c\frac{\overline{\lambda}_i}{\cos\theta_i}$$

with some constants $c$ and $c_1$. A comparison with Equation (3.3) concludes our proof.

## 3.3 Spherical Mean Value Coordinates

Now we develop spherical mean value coordinates. They inherit positivity from their planar counterparts: the $\lambda_i(\mathbf{v})$ are positive if $\mathbf{v}$ is contained in the kernel of the polygon given by the $\mathbf{v}_i$. If $\mathbf{v}$ is contained in the convex hull of the $\mathbf{v}_i$, this can always be arranged by reordering the vertices $\mathbf{v}_i$ with respect to their polar angle around $\mathbf{v}$. (The kernel is the region inside the polygon from which the whole polygon is "visible".) Furthermore, these coordinates are well defined for arbitrary $\mathbf{v}$ and arbitrary admissible polygons.

Planar mean value coordinates are given by Floater's formula [Flo03]

$$\overline{\lambda}_i = \frac{w_i}{\sum_j w_j}, \qquad w_i = \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{r_i} \tag{3.4}$$

where $\alpha_i$ is the signed angle between $\overline{\mathbf{v}}_i - \mathbf{v}$ and $\overline{\mathbf{v}}_{i+1} - \mathbf{v}$ and $r_i$ is the distance $||\overline{\mathbf{v}}_i - \mathbf{v}||$. As shown in Figure 3.1 (middle), $\alpha_i$ is given as the dihedral, signed angle between the planes determined by $\mathbf{v}$, $\mathbf{v}_i$, and the origin, and $\mathbf{v}$, $\mathbf{v}_{i+1}$, and the origin, respectively. That is, $\alpha_i$ is the signed angle between $\mathbf{v} \times \mathbf{v}_i$ and $\mathbf{v} \times \mathbf{v}_{i+1}$. The distance $r_i$ is given by $r_i = \tan\theta_i$ where $\theta_i$ is the angle between $\mathbf{v}$ and $\mathbf{v}_i$. By inserting these terms into (3.4) and (3.1) and using $\langle\mathbf{v}_i, \overline{\mathbf{v}}_i\rangle = \frac{1}{\cos\theta_i}$, we obtain

$$\lambda_i(\mathbf{v}) = \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{\sin\theta_i} \Bigg/ \sum_j \cot\theta_j \left(\tan\frac{\alpha_{j-1}}{2} + \tan\frac{\alpha_j}{2}\right). \tag{3.5}$$

Note that this formula gives us a continuous definition of $\lambda_i$ for arbitrary $\mathbf{v}$. In particular, it is valid for $\theta_i \geq \frac{\pi}{2}$ although our geometric motivation in Figure 3.1 (left) assumed angles $\theta_i < \frac{\pi}{2}$. However, we can now give a geometric interpretation for $\theta_i > \frac{\pi}{2}$ as well as indicated on the right of Figure 3.1. In this case, the intersection point of the tangent plane $T_{\mathbf{v}}\mathbb{S}^2$ and the line through $\mathbf{v}_{i+1}$ and the origin lies on the "negative part" of that line, and we consider the vector $\overline{\mathbf{v}}_{i+1} - \mathbf{v}$ in the projected polygon $\overline{P}$ as having negative length without changing its orientation; this means that $\alpha_i$ is effectively measured as the angle between $\overline{\mathbf{v}}_i - \mathbf{v}$ and $-(\overline{\mathbf{v}}_{i+1} - \mathbf{v})$. It can be seen that the planar mean value coordinates still satisfy linear precision (2.2) for this case (for example, this is implied by the construction in Section 2.3). Therefore, the derived spherical barycentric coordinates fulfill linear precision as well.

From (3.5), it is easy to see that the spherical mean value coordinates are well defined and positive if $\mathbf{v}$ is inside a convex spherical polygon and $\theta_i < \frac{\pi}{2}$ for all $\theta_i$. For such polygons, inequality (2.1′) is implied by the triangle inequality.

Next, we show that the spherical mean value coordinates are well defined on the whole sphere $\mathbb{S}^2$ for arbitrary admissible spherical polygons. Since the planar mean value coordinates $\overline{\lambda}_i$ are well defined and the spherical mean value coordinates are a scaled version of them, we can basically conclude that spherical mean value coordinates are well defined, too. When doing so, one new difficulty occurs: when a spherical polygon is projected to the tangent plane, more general polygons may occur than considered so far, see Figures 3.1 (right) and 3.3. We need to show that the results from Section 2.5 still hold in this case.

**3.2 Definition.** A *projective polygon* consists of a finite sequence of distinct vertices $\mathbf{v}_i \in \mathbb{R}^2$ and a sequence of non-intersecting edges, given either by $\mathbf{e}_i = \{\mathbf{x} = p\mathbf{v}_i + q\mathbf{v}_{i+1} \in \mathbb{R}^2 | p + q = 1,\ p, q \in [0, 1]\}$ or by $\mathbf{e}_i = \{\mathbf{x} = p\mathbf{v}_i + q\mathbf{v}_{i+1} \in \mathbb{R}^2 | p + q = 1,\ p, q \in \mathbb{R} \setminus (0, 1)\}$.

Such a polygon can be represented in polar coordinates as a sequence of distinct vertices $\mathbf{v}_j = r_j \mathrm{e}^{\mathrm{i}\phi_j}$ with $\phi_j \in [0, 2\pi)$ and $r_j \in \mathbb{R}$ (including the negative numbers). Using this notation, the complement of the line segment $(\mathbf{v}_j, \mathbf{v}_{j+1})$ is chosen as edge $\mathbf{e}_j$ if and only if the signs of $r_j$ and $r_{j+1}$ differ, see Figure 3.3. Without loss of generality let $\mathbf{x}$ be at the origin. Then we can define mean value coordinates for projective polygons just as in Algorithm 2.3, and Formula (2.7) becomes

$$w_i = \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{r_i}$$

with $\alpha_i = \phi_{i+1} - \phi_i$.

Since projective polygons should actually be defined for the real projective plane $\mathbb{RP}^2 \supset \mathbb{R}^2$, we can use the same proofs as in Section 2.5 if we take care of the infinite points. The idea is that infinite points have infinite norm $r_i$ and therefore weight zero. More accurately, we had to consider the limit of points that approach infinity on a given edge. Doing this, we find out that Lemmas 2.5–2.8 still hold. Lemma 2.9 has to be stated more precisely as

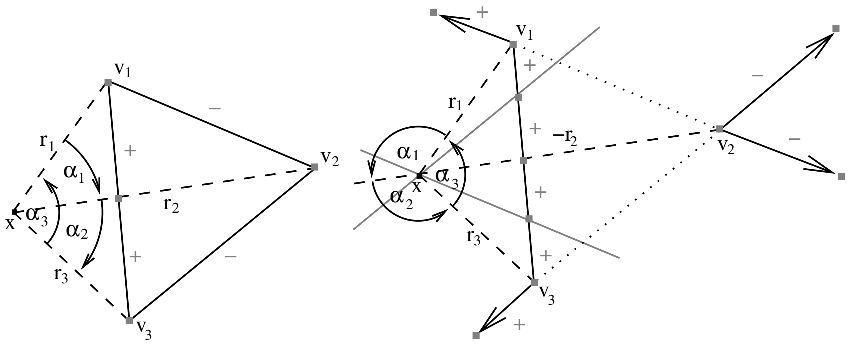


**Figure 3.3: Two projective triangles with the same vertices but different edges. The refinement in Theorem 3.4 and the sign of the edge weights are indicated in grey.**

**3.3 Lemma.** *Let* $\mathbf{e} = (\mathbf{v}_i, \mathbf{v}_{i+1})$ *be an edge that does not contain an infinite point in its interior. Let* $\mu_{\mathbf{e},i}$ *and* $\mu_{\mathbf{e},i+1}$ *be the coefficients of the respective edge vector* $\mathbf{v}_\mathbf{e}$*.*

*Then* $\mu_{\mathbf{e},i} + \mu_{\mathbf{e},i+1}$ *is greater than zero if and only if the respective angle* $\alpha_i$ *and the distances* $r_i$ *and* $r_{i+1}$ *have the same sign.*

We can now state

**3.4 Theorem.** *Planar mean value coordinates for projective polygons are well defined.*

*Proof.* In a first step, the projective polygon is refined at all infinite points. Then, no edge contains an infinite point in its interior any more, and we can proceed as in the proof of Theorem 2.10 using Lemmas 2.8 and 3.3. □

**3.5 Corollary.** *Let* $P$ *be an admissible spherical polygon. Then the spherical mean value coordinates with respect to* $P$ *are well defined on* $\mathbb{S}^2$*.*

*Proof.* Since $P$ is admissible, its projection $\overline{P}$ is a projective polygon (in particular, the projection is still non-intersecting). Hence, the mean value coordinates with respect to $P$ are well defined by Theorem 3.4. □

## 3.4 Bézier Surfaces on Spherical Polygonal Domains

Spherical barycentric coordinates can be used to construct Bézier surfaces on spherical domains. For a triangulation of the sphere, this has been done in [ANS96]. Our extension to arbitrary spherical polygons makes it possible to handle arbitrary tessellations of a sphere. As in the planar case, the de Casteljau algorithm can be used to compute the spherical Bézier patches. The challenging part is to ensure smoothness across the polygon boundaries of the tessellation. The respective theory for planar tessellations is developed in part II. If it is combined with the methods from [ANS96], it should be possible to define smooth Bézier surfaces on polygonal tessellations of the sphere. However, the details of such a construction remain future work.

## 3.5 Summary

We have introduced spherical barycentric coordinates. The special case of spherical Wachspress coordinates generalize Ju et al.'s vector coordinates. The spherical mean value coordinates are defined not only for convex spherical polygons but for arbitrary spherical polygons without antipodal points. We have shown that the spherical mean value coordinates are well defined for such polygons.

In the next chapter, we show how spherical barycentric coordinates can be used to construct 3D barycentric coordinates for polyhedra with arbitrary polygonal faces. Before, this was not possible in this generality.

# Chapter 4

# Barycentric Coordinates for Arbitrary Polyhedra

In this chapter, we extend 3D barycentric coordinates from polyhedra with triangular faces to arbitrary polyhedra. This makes it possible to use barycentric coordinates in conjunction with subdivision surfaces and conical meshes [LPW*06]. In particular, we introduce 3D mean value coordinates and prove that they are defined in the whole $\mathbb{R}^3$ space. We show that the 3D mean value coordinates for triangular polyhedra in [FKR05, JSW05] are a special case of our construction. Finally, we discuss several applications including interpolation, extrapolation, and space deformations.

## 4.1 Barycentric Coordinates for Arbitrary Polyhedra

First, we present the approach to computing barycentric coordinates for polyhedra with triangular faces that was introduced in [JLW07]. Then, we show how barycentric coordinates for arbitrary polyhedra can be obtained by using the spherical barycentric coordinates proposed in Section 3.1. 3D mean value coordinates are computed by Equations (4.1)–(4.3).

**4.1 Definition.** A *polyhedron $P$* consists of a finite set of distinct vertices $\mathbf{v}_i \in \mathbb{R}^3$ and a set of non-intersecting faces that are bounded by planar polygons. $F(\mathbf{v}_i)$ denotes the set of faces incident to $\mathbf{v}_i$. For a face $\mathbf{f}$, $V(\mathbf{f})$ denotes the set of indices $i$ such that $\mathbf{v}_i$ is incident to $\mathbf{f}$, and $V(P)$ denotes the set of all vertex indices.

Let $\mathbf{x} \in \mathbb{R}^3$ be a point. Its barycentric coordinates with respect to a polyhedron with vertices $\mathbf{v}_i$ consist of coordinate functions $\lambda_i^{3D}(\mathbf{x})$ that fulfill (2.1) and (2.2). Once more, we begin by constructing homogeneous coordinates $w_i$ that satisfy (2.2′) and normalize them afterwards. This is done in four steps [JLW07]:

**4.2 Algorithm (3D barycentric coordinates).** Barycentric coordinates $\lambda_i^{3D}$ for a point $\mathbf{x} \in \mathbb{R}^3$ with respect to a polyhedron $P$ can be defined in the following way:

- A face vector $\mathbf{v_f}$ is assigned to each face $\mathbf{f}$ of the polyhedron such that $\sum_{\mathbf{f}} \mathbf{v_f} = \mathbf{0}$. ($\mathbf{v_f}$ can be considered as some kind of face normal.)
- For each face $\mathbf{f}$, its face vector $\mathbf{v_f}$ is distributed to its respective face vertices by choosing coefficients $\mu_{\mathbf{f},i}$ such that $\sum_{i \in V(\mathbf{f})} \mu_{\mathbf{f},i}(\mathbf{v}_i - \mathbf{x}) = \mathbf{v_f}$.
- The weights at each vertex $\mathbf{v}_i$ are cumulated as $w_i := \sum_{\mathbf{f} \in F(\mathbf{v}_i)} \mu_{\mathbf{f},i}$.
- The weights are normalized to form a partition of unity:

$$\lambda_i^{3D} := \frac{w_i}{\sum_j w_j}.$$

Note that two choices need to be made in this procedure: the face vectors $\mathbf{v_f}$ in Step 1 and the coefficients $\mu_{\mathbf{f},i}$ in Step 2 are in general not uniquely defined. Ju et al. [JLW07] discuss the choice of $\mathbf{v_f}$ and suggest several definitions to obtain Wachspress coordinates, mean value coordinates, discrete harmonic coordinates, Voronoi coordinates, or any other barycentric coordinates. However, they consider only the case of polyhedra with triangular faces, and in this case, the coefficients $\mu_{\mathbf{f},i}$ are uniquely determined.

For the case of arbitrary polyhedra, we propose choosing spherical barycentric coordinates $\lambda_i(\mathbf{v_f}; P_\mathbf{f})$ as coefficients $\mu_{\mathbf{f},i}$ in Step 2 where $P_\mathbf{f} := (\mathbf{v}_i - \mathbf{x})_{i \in V(\mathbf{f})}$ is the boundary polygon of $\mathbf{f}$, relative to $\mathbf{x}$. Note that the vertices $\mathbf{v}_i - \mathbf{x}$ do not lie on the unit sphere in general. Therefore, the $\lambda_i$ refer to the generalized spherical barycentric coordinates (3.2). It is clear from the construction that the $w_i$ satisfy Equation (2.2′). It follows that the $\lambda_i^{3D}$ satisfy (2.1) and (2.2). In the next section, we describe the necessary choices to obtain 3D mean value coordinates.

### 4.1.1 Mean value coordinates for arbitrary polyhedra

Mean value coordinates are the most flexible since they can be computed for non-convex polyhedra with non-convex (planar) faces. The following construction of the associated face vector is due to [FKR05, JSW05]. Let $\mathbf{x} \in \mathbb{R}^3$ be a point. For a face $\mathbf{f}$, let $P_\mathbf{f}$ be the boundary polygon with respect to $\mathbf{x}$ as above. Let $\mathbb{S}^2$ be the unit sphere centered at $\mathbf{x}$. We know from Stokes' theorem that the integral over the outward unit normal $\mathbf{n} : \mathbb{S}^2 \to \mathbb{R}^3$ of this sphere is zero:

$$\int_{\mathbb{S}^2} \mathbf{n} \, \mathrm{d}S = \mathbf{0}.$$

For a face $\mathbf{f}$, let $\mathbb{S}^2_{\mathbf{f}} \subset \mathbb{S}^2$ be the spherical patch obtained by projecting $\mathbf{f}$ to $\mathbb{S}^2$. Choose the orientation of $\mathbb{S}^2_{\mathbf{f}}$ consistently with the orientation of the boundary of $\mathbf{f}$. The patches $\mathbb{S}^2_{\mathbf{f}}$ tessellate the sphere $\mathbb{S}^2$, and we can define $\mathbf{v_f}$ as the integral of $\mathbf{n}$ over $\mathbb{S}^2_{\mathbf{f}}$:

$$\mathbf{v_f} := \int_{\mathbb{S}^2_{\mathbf{f}}} \mathbf{n}\, \mathrm{d}S, \qquad \sum_{\mathbf{f}} \mathbf{v_f} = \mathbf{0}.$$

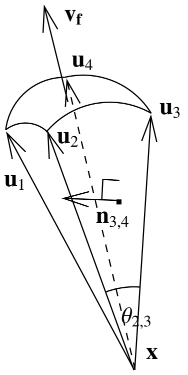


**Figure 4.1: The face vector $\mathbf{v_f}$.**

Let $Q_\mathbf{f}$ be the spherical polygon that is obtained by projecting the vertices of $P_\mathbf{f}$ to $\mathbb{S}^2$. Let the vertices of $Q_\mathbf{f}$ be $\mathbf{u}_1, \ldots \mathbf{u}_n$. Another application of Stokes' theorem yields the formula

$$\mathbf{v_f} = \sum_{i=1}^{n} \frac{1}{2} \theta_{i,i+1} \mathbf{n}_{i,i+1} \tag{4.1}$$

where $\theta_{i,i+1}$ is the angle between $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$, and $\mathbf{n}_{i,i+1} := \frac{\mathbf{u}_i \times \mathbf{u}_{i+1}}{\|\mathbf{u}_i \times \mathbf{u}_{i+1}\|}$ is the oriented unit normal to the plane determined by these vectors (Figure 4.1).

The construction of this face vector $\mathbf{v_f}$ is essentially the same as in the case of polyhedra with triangular faces which was derived from the continuous case [FKR05, JSW05]. The choice of the $\mu_{\mathbf{f},i}$, however, can not be guided by the continuous case since it reflects different possibilities to extend a function, which is defined on the vertices of a polyhedron, to its faces. Therefore, just looking at the continuous case, we have several choices for the $\mu_{\mathbf{f},i}$ which can be called 3D mean value coordinates with equal right. Nevertheless, it seems reasonable to require additionally that the restriction of the coordinates to the faces of the polyhedron yields the planar mean value coordinates of the respective faces. This can be achieved by choosing spherical mean value coordinates $\lambda_i$ for the coefficients $\mu_{\mathbf{f},i}$ (Section 4.1.2).

For each face $\mathbf{f}$, the $\lambda_i(\mathbf{v_f}; P_\mathbf{f})$ can easily be computed with (3.5) and (3.2). (In these formulae, $\mathbf{v} = \mathbf{v_f}$ and $\theta_i$ is the angle between $\mathbf{v_f}$ and $\mathbf{u}_i$.) We obtain

$$\lambda_i(\mathbf{v_f}; P_\mathbf{f}) = \frac{\|\mathbf{v_f}\|}{\|\mathbf{v}_i - \mathbf{x}\|} \cdot \frac{\tan\frac{\alpha_{i-1}}{2} + \tan\frac{\alpha_i}{2}}{\sin\theta_i} \Bigg/ \sum_j \cot\theta_j \left(\tan\frac{\alpha_{j-1}}{2} + \tan\frac{\alpha_j}{2}\right). \tag{4.2}$$

Since the vertices $\mathbf{v}_i$, $i \in V(\mathbf{f})$ of $P_\mathbf{f}$ are the boundary vertices of the planar face $\mathbf{f}$, the projected polygon $Q_\mathbf{f}$ (the boundary of $\mathbb{S}^2_{\mathbf{f}}$) is contained in one hemisphere of $\mathbb{S}^2$ and hence admissible. This ensures, by Corollary 3.5, that the $\lambda_i$ are well defined. This is true even if $\frac{\mathbf{v_f}}{\|\mathbf{v_f}\|}$ is not contained in the interior of $Q_\mathbf{f}$ which may happen for non-convex faces. Consequently, we can assign barycentric coordinates $\lambda_i^{3D}$ to each vertex $\mathbf{v}_i$:

$$\lambda_i^{3D}(\mathbf{x}) := \frac{w_i(\mathbf{x})}{\sum_j w_j(\mathbf{x})}, \qquad w_i(\mathbf{x}) := \sum_{\mathbf{f} \in F(\mathbf{v}_i)} \lambda_i(\mathbf{v_f}; P_\mathbf{f}). \tag{4.3}$$

By construction, they coincide with the mean value coordinates from [FKR05, JSW05] if they are computed for polyhedra with triangular faces.

Finally, we show that the denominator in (4.3) does not vanish. It turns out that the proof can be given along the same lines as in Section 2.5 for planar mean value coordinates. Therefore, we indicate only the necessary changes.

It is easy to see that analogons of Lemmas 2.5–2.8 can be proven for polyhedra by replacing lines and edges by planes and faces and so on. We obtain

**4.3 Definition.** A *refinement* $\widehat{P}$ of a polyhedron $P$ is a polyhedron that contains all the vertices of $P$ and additional vertices and edges that lie on faces of $P$ such that $P$ and $\widehat{P}$ bound the same volume in $\mathbb{R}^3$.

**4.4 Lemma (refinement of polyhedra).** *Let $P$ be a polyhedron, and let $\widehat{P}$ be a refinement of $P$. Let $w_i$ and $\widehat{w}_i$ be the weights (4.3) of Algorithm 4.2 for $P$ and $\widehat{P}$. Then $\sum_{i \in V(P)} w_i = \sum_{i \in V(\widehat{P})} \widehat{w}_i$.*

The proofs for Lemma 2.9 and Theorem 2.10 carry over to the 3D case as well, and we arrive at

**4.5 Theorem.** *3D mean value coordinates are well defined in* $\mathbb{R}^3$.

Note that this theorem holds also for non-convex polyhedra with multiple components (if these are oriented alternatingly, compare [HF06]). Also, we do not require the faces to be simply connected.

### 4.1.2 Behavior of the mean value coordinates on the faces

The denominator of Equation (4.2) becomes zero if $\mathbf{x}$ is contained in a face $\mathbf{f}$ of the polyhedron. In this case, the face vector $\mathbf{v_f}$ is orthogonal to $\mathbf{f}$ and $Q_\mathbf{f}$ lies on a great circle since its vertices $\frac{\mathbf{v}_i - \mathbf{x}}{\|\mathbf{v}_i - \mathbf{x}\|}$ lie in the plane determined by $\mathbf{f}$. We show now that the 3D mean value coordinates have nevertheless a continuous extension to the faces and that this extension coincides with the 2D mean value coordinates with respect to these faces. Assume that $\mathbf{x}$ approaches a point located on the face $\mathbf{f}$. Then $\mathbf{v_f}$ approaches the face normal. For $j \in V(\mathbf{f})$, we can conclude that the denominator of the $\lambda_j(\mathbf{v_f}; P_\mathbf{f})$ defined in Equation (4.2) approaches zero and $\lambda_j$ approaches infinity. Therefore, in the limit (due to the normalization)

$$\lambda_i^{3D}(\mathbf{x}) = \begin{cases} \dfrac{w_i}{\sum_{j \in V(\mathbf{f})} w_j}, \; w_i = \lambda_i(\mathbf{v_f}), & i \in V(\mathbf{f}) \\ 0, & \text{otherwise.} \end{cases}$$

This approaches the usual 2D mean value coordinates of $\mathbf{x}$ with respect to $\mathbf{f}$.

## 4.2 Higher Dimensions

Using barycentric coordinates for arbitrary polyhedra in $\mathbb{R}^3$, we can construct spherical barycentric coordinates for arbitrary spherical polyhedra on the three-dimensional sphere. These can then be used to obtain barycentric coordinates for arbitrary polytopes in $\mathbb{R}^4$ and successively in higher dimensions. At each stage, we must make a choice for the (hyper-)face vector and for the coefficients $\mu_{\mathbf{f},i}$. However, if we define the (hyper-)face vector as the integral of the outward unit normal over a suitable part of the unit (hyper-)sphere and choose the spherical mean value coordinates of the respective dimension as coefficients $\mu_{\mathbf{f},i}$, we obtain a generalization of the mean value coordinates to $n$ dimensions. The proofs given in this chapter carry over to $n$ dimensions for this generalization so that we obtain all the characteristic properties of mean value coordinates: They are positive within convex polytopes, they have the whole $\mathbb{R}^n$ space as domain, and they coincide with the mean value coordinates of the respective lower dimension if restricted to the (hyper-)faces of the polytope. To make this construction more precise, we give a recursive definition for a polytope.

**4.6 Definition.** A *solid one-dimensional polytope* is a line segment. Its end points are also called hyperfaces.

A *$(n + 1)$-dimensional polytope* $P \subset \mathbb{R}^{n+1}$ consists of a set $F$ of solid $n$-dimensional polytopes (rigidly embedded in $\mathbb{R}^{n+1}$), which are called the hyperfaces of $P$, such that the interiors of the hyperfaces are disjoint and each $(n - 1)$-dimensional hyperface of a hyperface in $F$ coincides with exactly one of the other $(n - 1)$-dimensional hyperfaces in $F$.

$P$ divides $\mathbb{R}^{n+1}$ into an interior and an exterior part. The union of $P$ and the associated interior part of $\mathbb{R}^{n+1}$ is called a *solid $(n + 1)$-dimensional polytope*.

Note that we do not require that polytopes are connected. Therefore, the associated interior and exterior parts of $\mathbb{R}^{n+1}$ may have multiple connected components as well. (Nevertheless, a hyperface should always separate a region labeled as “interior” from a region labeled as “exterior”.) A 2-dimensional polytope is a polygon, and a 3-dimensional polytope is a polyhedron.

## 4.3 Applications

### 4.3.1 Interpolation and extrapolation

The most direct application of mean value coordinates is their use for interpolation and extrapolation using (2.3). In Figure 4.2, color values are specified at the eight vertices of the cube. In (a) and (c), the values are interpolated on the faces.

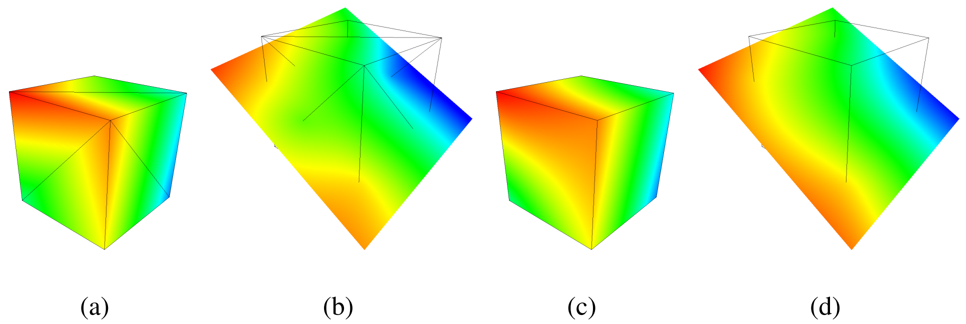


**Figure 4.2: An example of interpolation of color values using 3D mean value coordinates. The color values are specified at the vertices of the cube. They are interpolated on the faces ((a) and (c)) and on a plane passing through the cube ((b) and (d)). If the cube is triangulated beforehand ((a) and (b)), the interpolation is less smooth than with our method ((c) and (d)).**

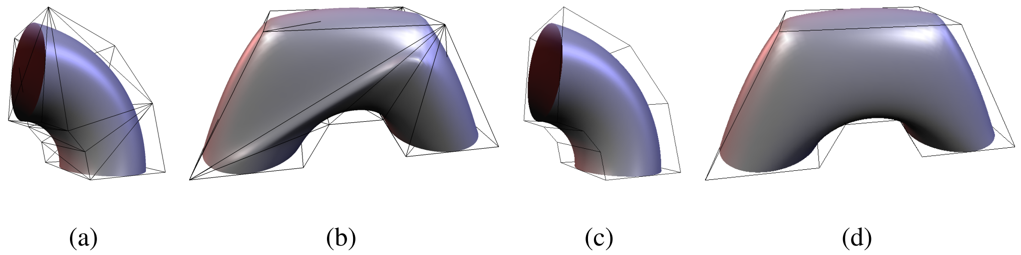


**Figure 4.3: An example of a space deformation using 3D mean value coordinates with respect to the polygonal control mesh. If the control mesh is triangulated beforehand, strong artifacts may be introduced ((a) and (b)). No triangulation is necessary with our method ((c) and (d)).**

In (b) and (d), the color values are interpolated and extrapolated on a plane that passes through the cube. In (a) and (b), the cube was triangulated before the interpolation. The piecewise linear structure of the interpolation on the triangles is clearly visible. With our 3D mean value coordinates, a triangulation is no longer necessary, and the resulting interpolation is much smoother.

### 4.3.2 Space deformations with 3D mean value coordinates

Figure 4.3 shows an example how mean value coordinates can be used for space deformations. We determine the mean value coordinates of the vertices of the

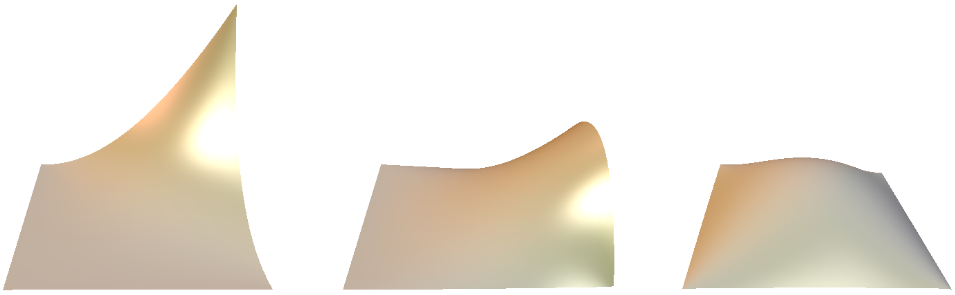


**Figure 4.4: The quadratic mean value Bernstein polynomials $B^2_{2000}$, $B^2_{1100}$, and $B^2_{1010} + B^2_{0101}$.**

tube with respect to the black control mesh with vertices $\mathbf{v}_i$. Then we deform the control mesh by moving the vertices to points $\mathbf{w}_i$ and calculate the new location of the tube by $\mathbf{x} = \sum_i \lambda_i^{3D}\left(\mathbf{x}; (\mathbf{v}_j)_j\right) \mathbf{w}_i$. Note that we can compute these coordinates for non-convex (control) meshes with non-convex faces ((c) and (d)). If all faces are triangulated, the number of faces is nearly tripled, the result depends on the chosen triangulation, and large artifacts may be introduced ((a) and (b)).

Although this approach is very simple, it is possible to obtain pleasing results. These can be considerably improved by modifying the barycentric coordinates such that derivatives are taken into account. This is discussed in detail in Chapter 5.

### 4.3.3 Bernstein polynomials on polygons and polyhedra

In this section, we introduce Bernstein polynomials in mean value coordinates on polygons. They can be used to define generalized Bézier surfaces. The full theory, including examples of Bernstein polynomials on polyhedra, is developed in Part II.

Bézier surfaces are defined by a linear combination of Bernstein polynomials which are polynomials in barycentric coordinates. Using classical barycentric coordinates, this was only possible for triangles. Using tensor product polynomials, Bernstein polynomials can be defined on quadrangular domains as well, but this leads to a higher degree of the polynomial. The only approaches for general polygons that we are aware of are restricted to convex polygons [LD89, Gol02].

We can define mean value Bernstein polynomials for arbitrary polygons and polyhedra. For a polygon or polyhedron with $k$ vertices, the general form for the Bernstein polynomials in the coordinates $\lambda = (\lambda_1, \ldots \lambda_k)$ is

$$B^n_\alpha(\mathbf{x}) = \frac{n!}{\alpha!}\lambda^\alpha(\mathbf{x})$$

where we use multi-indices $\alpha = (\alpha_1, \dots \alpha_k) \in \mathbb{N}^k$ with the notation $\alpha! := \alpha_1! \cdots \alpha_k!$ and $\lambda^\alpha := \lambda_1^{\alpha_1} \cdots \lambda_k^{\alpha_k}$. In Figure 4.4, we show some quadratic Bernstein polynomials on a square using mean value coordinates [Flo03].

Important properties of classical Bézier surfaces like the convex hull property and the de Casteljau algorithm still hold in this extended setup.

## 4.4 Summary

We have shown that spherical mean value coordinates can be used to construct 3D mean value coordinates for polyhedra with arbitrary polygonal faces while before only 3D mean value coordinates for triangular polyhedra were known. The same method can be used to construct mean value coordinates for arbitrary polytopes of successively higher dimensions. We showed that the $n$-D mean value coordinates are well defined. This concludes the generalization of mean value coordinates from two to $n$ dimensions.

In the future, it would be interesting to find a general theory for barycentric coordinates for arbitrary polytopes similar to the one given in [FHK06, JLW07]. It should shed light on the relationship between “Euclidean” and spherical coordinates. To construct the general 3D mean value coordinates, we had to make two choices. First, we chose to use the “mean value” face vector as in [FKR05, JSW05], then we chose spherical mean value coordinates as coefficients $\mu_{\mathbf{f},i}$. However, we do not yet know which choices have to be made to obtain other types of coordinates like the Warren-Wachspress coordinates [War96]. While it seems obvious that the face vector of the respective type should be chosen by integrating over the respective generating surface (see [JLW07] for details) to ensure consistency with existing definitions, it is less obvious which choice is the “right” one for the coefficients $\mu_{\mathbf{f},i}$ and which effects would result from different choices.

# Chapter 5

# Higher Order Barycentric Coordinates

In recent years, a wide range of generalized barycentric coordinates has been suggested. However, they usually lack control over derivatives. We show how the notion of barycentric coordinates can be extended to specify derivatives at control points. This is also known as Hermite interpolation. We introduce a method to modify existing barycentric coordinates to higher order barycentric coordinates and demonstrate, using higher order mean value coordinates, that our method, although conceptually simple and easy to implement, can be used to give easy and intuitive control at interactive frame rates over local space deformations such as rotations.

## 5.1 Introduction

Barycentric coordinates have a wide range of applications as shown before. For some of them, the currently available barycentric coordinates are sufficient. However, many others, like space deformations, are based on barycentric interpolation. For these, it is often desirable to have a *Hermite interpolation*. That is, we want to specify not only certain *values* at interpolation points but also the *derivatives*. One way to do this for transfinite interpolation in the case that values and derivatives are given for a boundary curve, is suggested by Dyken and Floater [DF08]. They propose using a weighted combination of two transfinite barycentric interpolation functions to achieve Hermite interpolation. We pursue a different approach designed for Hermite interpolation at discrete points. We attain interpolation of derivatives by modifying existing barycentric interpolation schemes. For space deformations, this allows the specification of rotations and other linear transformations directly at a single point. Before, this had to be done by moving a whole group of control points. A survey of further interpolation techniques is given in [Alf89].

In this chapter, we introduce *higher order barycentric coordinates* which can be used for Hermite interpolation. To achieve this, the definition of barycentric coordinates has to be adapted. We refer to the barycentric coordinates considered so far as *conventional barycentric coordinates* to distinguish them from higher order barycentric coordinates. Using the higher order coordinates, we gain better local control over interpolations, and less control points are necessary compared to conventional barycentric coordinates. Their name is due to their ability to interpolate first order derivatives at the vertices while conventional coordinates only interpolate zero order derivatives (values). However, it is not implied that an interpolation with precision of higher order is achieved with higher order barycentric coordinates. They retain the linear precision property but are still not able to reproduce quadratic or cubic polynomials exactly.

The main contributions of this chapter are:

- We give an axiomatic definition of higher order barycentric coordinates (Section 5.2.2),
- we describe a method to modify conventional barycentric coordinates to obtain higher order barycentric coordinates (Section 5.2.3),
- and we demonstrate the capabilities of our coordinates in the context of space deformations (Section 5.3).

## 5.2 Construction of Higher Order Barycentric Coordinates

In this section, we introduce higher order barycentric coordinates. We begin by recapitulating the relevant properties of conventional barycentric coordinates from an interpolation point of view to allow a better understanding of the way in which our extension of the notion of barycentric coordinates fits into this framework.

### 5.2.1 Axioms for conventional barycentric coordinates

Barycentric coordinates with respect to a polytope or point cloud $P = \{\mathbf{v}_j\}_{j=1\ldots n}$, $\mathbf{v}_j \in \mathbb{R}^n$, are a set of functions $\lambda_i(\mathbf{x}) = \lambda_i(\mathbf{x}; P) : \Omega \to \mathbb{R}$, $P \subset \Omega \subset \mathbb{R}^n$, which satisfy the three axioms below.

**Lagrange property** $\lambda_i(\mathbf{v}_j) = \delta_{ij}$**.** This property is the foundation of using barycentric coordinates for interpolation purposes. Given a set of barycentric coordinates $\lambda_i$ and a set of function values $f_i$, we obtain an *interpolation function*

$$f_{\lambda,\mathbf{f}}(\mathbf{x}) := \sum_i \lambda_i(\mathbf{x}) f_i.$$

The Lagrange property is equivalent to **vertex interpolation**: $f_{\lambda,\mathbf{f}}(\mathbf{v}_i) = f_i$. Note that the Lagrange property was not required by our original definition in Chapter 2. However, all barycentric coordinates we know of satisfy the Lagrange property as well. In particular, it is fulfilled by all positive barycentric coordinates with respect to convex polygons [FHK06].

**Partition of unity $\sum_i \lambda_i = 1$.** This property yields **constant precision** for the interpolation function. That is, constant functions can be reproduced exactly from the interpolation values: Let $f_i = g(\mathbf{v}_i)$ for any constant function $g$. Then $f_{\lambda,\mathbf{f}} = g$.

**Linear precision.** Linear functions can be reproduced exactly from the interpolation values: Let $f_i = g(\mathbf{v}_i)$ for any linear function $g$. Then $f_{\lambda,\mathbf{f}} = g$. Together with constant precision, we obtain **affine precision** and can reproduce affine functions exactly.

Often the following additional properties are required.

**Domain $\Omega = \mathbb{R}^n$.** The domain should be as large as possible. Wachspress coordinates, for example, are only defined within convex polygons while mean value coordinates are defined everywhere in $\mathbb{R}^n$.

**Non-negativity $\forall \mathbf{x} \in \Omega \forall i \ \lambda_i(\mathbf{x}) \geq 0$.** Unfortunately, all known coordinates that fulfill this property in full generality have deficits either with the domain or the smoothness property.

**Smoothness.** The coordinate functions should be as smooth as possible.

A more detailed description of the interrelations of these properties can be found in [FHK06].

### 5.2.2 Axioms for higher order barycentric coordinates

The interpolation function $f_{\lambda,\mathbf{f}}$ does not allow the specification of any derivatives since it blends only the constant terms $f_i$. We suggest blending the first two terms of the general Taylor series at the points $\mathbf{v}_i$ instead by using the interpolation function

$$f_{\lambda,\mathbf{f},\mathbf{D}}(\mathbf{x}) := \sum_i \lambda_i(\mathbf{x})(f_i + D_i(\mathbf{x} - \mathbf{v}_i)) \qquad (5.1)$$

where the $D_i$ are the linear functions (usually represented as matrices) which specify the derivatives at the $\mathbf{v}_i$.

To make sure that $f_{\lambda,\mathbf{f},\mathbf{D}}$ interpolates the derivatives $D_i$, the axioms for barycentric coordinates have to be modified as well.

**Lagrange property** $\lambda_i(\mathbf{v}_j) = \delta_{ij}$**.** This property remains unmodified and implies **vertex interpolation**: $f_{\lambda,\mathbf{f},\mathbf{D}}(\mathbf{v}_i) = f_i$.

**Partition of unity** $\sum_i \lambda_i = 1$**.** This property remains unmodified as well. Furthermore, it directly implies **affine precision** in conjunction with our new interpolation function $f_{\lambda,\mathbf{f},\mathbf{D}}$. However, the notion of affine precision has to be adapted slightly to our new interpolation function: Let $f_i = g(\mathbf{v}_i)$ and $D_i = \nabla g(\mathbf{v}_i)$ for any affine function $g$. Then $f_{\lambda,\mathbf{f},\mathbf{D}} = g$.

*Proof.* We note that $D := D_i$ is independent of $i$ for an affine function $g$ and $f_i + D(\mathbf{x} - \mathbf{v}_i) = g(\mathbf{x})$. Using the partition of unity, we obtain $f_{\lambda,\mathbf{f},\mathbf{D}}(\mathbf{x}) = \sum_i \lambda_i(\mathbf{x}) g(\mathbf{x}) = g(\mathbf{x})$. □

Consequently, linear precision is obsolete as an independent axiom.

**Derivative property** $\nabla\lambda_i(\mathbf{v}_j) = \mathbf{0}$**.** This property is equivalent to **derivative interpolation** $\nabla f_{\lambda,\mathbf{f},\mathbf{D}}(\mathbf{v}_i) = D_i$: Using the Lagrange property, it simplifies to $D_i + \sum_j \nabla\lambda_j(\mathbf{v}_i)(f_j + D_j(\mathbf{v}_i - \mathbf{v}_j)) = D_i$. Thus, $\forall i \forall j\ \nabla\lambda_i(\mathbf{v}_j) = \mathbf{0}$ is a necessary and sufficient condition.

The properties domain, non-negativity, and smoothness remain unchanged.

### 5.2.3 Construction of higher order barycentric coordinates

The key observation for constructing higher order barycentric coordinates is that the axioms of conventional and higher order barycentric coordinates differ mainly by substituting derivative interpolation for linear precision (with respect to $f_{\lambda,\mathbf{f}}$). Therefore, it is possible to obtain higher order barycentric coordinates by modifying conventional barycentric coordinates $\lambda_i$. This can be done by first concatenating them with a function $m : \mathbb{R} \to \mathbb{R}$ such that $m \circ \lambda_i$ satisfies the Lagrange and the derivative property. Afterwards, a normalization step is needed to reestablish the partition of unity. Detailed conditions are given by the following theorem.

**5.1 Theorem.** *Let $\lambda_i$ be a set of functions that satisfy the Lagrange property, partition of unity, and a Hölder condition with exponent $\alpha > \frac{1}{2}$ at the vertices ($|\lambda_i(\mathbf{v}_j) - \lambda_i(\mathbf{w})| < C\|\mathbf{v}_j - \mathbf{w}\|^\alpha$). Let $m : \mathbb{R} \to [0,\infty)$ be a non-negative, $C^1$-continuous function with $m(0) = 0$, $m(1) = 1$, and $m'(0) = m'(1) = 0$ such that*

*$m(x) > 0$ for $x > 0$. Let furthermore be an $\varepsilon > 0$ given such that $m$ is twice differentiable in $(-\varepsilon, \varepsilon) \setminus \{0\}$ and in $(1-\varepsilon, 1+\varepsilon) \setminus \{1\}$.*

*Then $\lambda_i^{ho} := \dfrac{m \circ \lambda_i}{\sum_k m \circ \lambda_k}$ is a set of higher order barycentric coordinate functions. Furthermore, domain and positivity are preserved, and if the $\lambda_i$ are $C^1$-continuous apart from the vertices, then the $\lambda_i^{ho}$ are $C^1$-continuous everywhere.*

The proof is contained in Appendix A.

**5.2 Corollary.** *Planar higher order mean value coordinates exist and are $C^1$-continuous everywhere.*

*Proof.* Given a function $m$ with the properties that are required in Theorem 5.1 (see Section 5.3.2 for a possible choice), it remains to show that planar mean value coordinates $\lambda_i$ satisfy a Hölder condition with exponent $\alpha > \frac{1}{2}$ at the vertices. While Hormann and Floater only state that the $\lambda_i$ are $C^0$, a closer examination of their proof [HF06, Theorem 4.6] reveals that they are in fact Lipschitz continuous. Therefore, they satisfy a Hölder condition with exponent $\alpha = 1$. □

So far, we have not proven this corollary for mean value coordinates in higher dimensions. Nevertheless, our experiments show that it holds for mean value coordinates in $\mathbb{R}^3$ as well.

## 5.3 Higher Order Barycentric Coordinates for Space Deformations

Our higher order barycentric coordinates allow a Hermite interpolation of arbitrary functions $f : \mathbb{R}^n \to \mathbb{R}^m$ at specified vertices $\mathbf{v}_i$ (if values $f_i$ and derivatives $D_i$ of the respective dimensions are given for each vertex). In particular, we can create interpolating curves $f : \mathbb{R} \to \mathbb{R}^m$, image deformations $f : \mathbb{R}^2 \to \mathbb{R}^2$, and space deformations $f : \mathbb{R}^3 \to \mathbb{R}^3$. In this section, we demonstrate the capabilities of our Hermite interpolation method for space deformations as a representative example.

### 5.3.1 Space deformations with barycentric coordinates

As pointed out by Joshi et al. [JMD*07], barycentric coordinates are particularly well-suited for space deformations using the function $f_{\lambda,\mathbf{f}} : \mathbb{R}^3 \to \mathbb{R}^3$ of Section 5.2.1. In practice, only the deformation of objects in the space $\mathbb{R}^3$ needs to be computed. This is done in two steps:

- The *binding step*: A control net (cage in the terminology of [JMD*07]) is bound to the object (for example a triangular mesh): For a given control net with control points $\mathbf{v}_i$, the barycentric coordinates $\lambda_i(\mathbf{p})$ are precomputed for each point $\mathbf{p}$ of the object (for example at the mesh vertices).

- The *deformation step*: The deformation is specified by assigning new positions $f_i$ to each control point. Then, the points $f_{\lambda,\mathbf{f}}(\mathbf{p})$ are calculated as the new positions of the object points.

Both steps can be done in $O(V \cdot P)$ time where $V$ is the number of control points and $P$ is the number of object points. The deformation has the following properties:

- The object is not deformed by the binding step: The linear precision property ensures that $f_{\lambda,\mathbf{f}} = \mathrm{id}_{\mathbb{R}^3}$ if $f_i = \mathbf{v}_i$.

- Direct and intuitive control over local translations. The vertex interpolation property allows the direct specification of new (control) point positions.

Higher order barycentric coordinates retain these properties (when the interpolation function $f_{\lambda,\mathbf{f},\mathbf{D}}$ is used instead of $f_{\lambda,\mathbf{f}}$). Additionally they allow:

- Direct and intuitive control over local rotations and shears. The derivative property allows us to specify the transformation directly at the vertices. For example, a rotation around a vertex can be specified or undesired shears can be prevented.

### 5.3.2 The choice of coordinates

Note that our axiomatic definition of higher order barycentric coordinates in Section 5.2.2 defines not only a single set of these coordinates but a whole family of them. Here, we discuss which of them are suitable choices for practical applications.

Of course, it is possible to construct the desired coordinates from scratch such that they satisfy the axioms of Section 5.2.2. However, the ongoing research for good conventional barycentric coordinates (which satisfy the axioms of Section 5.2.1) shows that this is a difficult research topic on its own. We are aware of only one known set of such coordinates, the Shepard weights $\lambda_i^{\mathrm{Sh}}(\mathbf{x}) = \frac{1/\|\mathbf{x}-\mathbf{v}_i\|^2}{\sum_j 1/\|\mathbf{x}-\mathbf{v}_j\|^2}$ [She68], the interpolation properties of which were studied in [Far86]. Unfortunately, they are not very well-suited for space deformations as it turned out in our experiments. Therefore, we suggest taking the short-cut indicated in Section 5.2.3 and constructing higher order coordinates by modifying conventional barycentric coordinates. Furthermore, this allows us to take advantage of existing implementations of these conventional coordinates.

Before choosing one set of conventional coordinate functions as the basis for our higher order coordinates, we recapitulate the strengths and weaknesses of some prospective candidates.

- 3D mean value coordinates [FKR05, JSW05, LBS06] are defined everywhere in $\mathbb{R}^3$ for arbitrary polyhedra. They are only $C^0$ at the vertices and coordinate functions may become negative for non-convex polyhedra.

- Positive mean value coordinates [LKCOL07] are modified mean value coordinates such that the coordinate functions are always non-negative. However, they introduce additional singularities and are only defined with respect to triangular polyhedra due to the need to compute them on the graphics card.

- Harmonic coordinates [JMD*07] are obtained by solving Laplace equations within the polyhedron. They are always non-negative and allow the specification of additional vertices within the polyhedron but are not defined outside the polyhedron. Furthermore, the solving of the Laplace equations needs a comparably high preprocessing time. However, Rustamov addressed the issue of computational time and proposed an extension to the exterior of the polyhedron [Rus07].

- Natural neighbor based coordinates [Sib80, BIK*97, HS00] depend only on the positions of the vertices $\mathbf{v}_i$ and not on the additional specification of a polyhedron like the other methods. However, they are usually only defined in the convex hull of the vertices and require the computation of a Voronoi tessellation.

Taking all this into account, we decided to use 3D mean value coordinates as the basis for higher order barycentric coordinates in our examples. They are defined everywhere in $\mathbb{R}^3$, which turned out to be beneficial for our applications, they are more flexible than the positive mean value coordinates, and their main shortcomings vanish in the context of higher order coordinates while their benefits are retained. Note, however, that Section 5.2.3 also provides a construction for other types of higher order barycentric coordinates if other properties are considered more important. If, for example, coordinates are desired, which have control vertices within the polyhedron or do not need a polyhedron at all, higher order harmonic coordinates or higher order natural neighbor coordinates can be constructed.

Finally, we have to choose a modifying function $m$. We use the piecewise polynomial

$$m(x) = \begin{cases} 0 & \text{if } x < 0, \\ -2x^2(x - \frac{3}{2}) & \text{if } 0 \leq x < 1, \\ 1 + (x - 1)^2 & \text{if } 1 \leq x < \frac{3}{2}, \\ x - \frac{1}{4} & \text{if } x \geq \frac{3}{2}. \end{cases}$$

In our first implementation, we used the polynomial $-2x^2(x - \frac{3}{2})$ on the whole domain and obtained good results. However, to satisfy the conditions of Theorem 5.1 and make sure that the denominator in the definition of the $\lambda_i^{\mathrm{ho}}$ cannot become zero, we changed $m$ for $x \geq 1$. To avoid $m(x)$ changing $x$ too drastically, we introduced additional modifications for $x < 0$ and $x \geq \frac{3}{2}$. This is possible with minimal additional computational effort. However, different choices for $m$ are possible.

Therefore, we use $m$ as above to derive *higher order 3D mean value coordinates* from 3D mean value coordinates as described in Theorem 5.1. They have the following properties:

- They are defined everywhere in $\mathbb{R}^3$. Therefore, they define a deformation $f_{\lambda,\mathbf{f},\mathbf{D}} : \mathbb{R}^3 \to \mathbb{R}^3$ on the whole space.
- They are expected to be smooth everywhere, analogous to Corollary 5.2.
- They are non-negative.
- When computing the coordinates of a point, it is automatically classified as inside or outside the control mesh (by the sign of the denominator, see [HF06, LBS07b]). This makes it easy to specify local deformations on partial control nets.

### 5.3.3 Examples

We demonstrate the extended interpolation capabilities of higher order mean value coordinates in Figure 5.1. In this figure, we did not perform any translations in order to focus on the deformations that are possible by altering the derivative at a single control point. Figure 5.2 shows additional examples of shears and rotations. It can be clearly seen how angles are preserved under rotations.

Figure 5.3 shows how higher order mean value coordinates can be used for partial deformations. It is possible to construct a control net around the whole model where only those parts of it are "switched on" at a certain stage that are actually to be deformed while the remainder of the model remains unchanged. Figure 5.3 shows the basic building block for such a deformation system. Although this can be done with other types of higher order barycentric coordinates

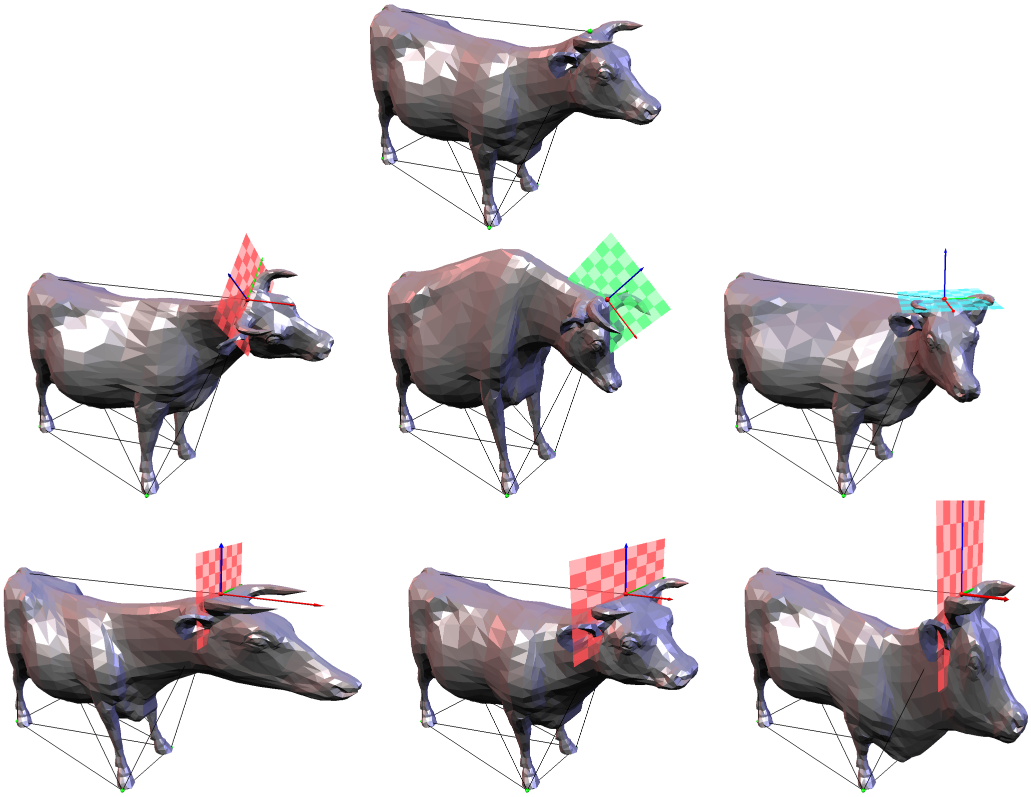


Figure 5.1: We demonstrate the effects that can be achieved by changing a single derivate in our control net. The top row shows the undeformed cow model and our control net. Note that it consists of only six points and that the control net penetrates the model. No care needs to be taken to enclose the model within the control net. The middle row shows rotations at the control point at the head around the red, green, and blue axis, respectively. This is done by changing the derivative for that control point from the identity matrix to the respective rotation matrix. The third row shows the effect of scaling along the red, green, and blue axis by a factor of two.

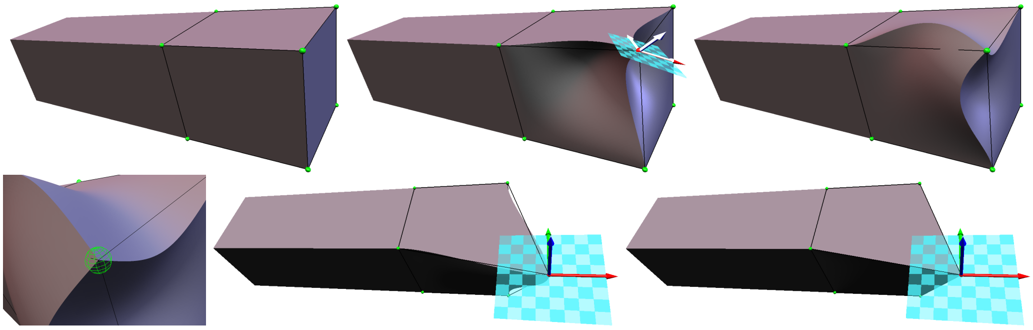


**Figure 5.2: The cuboid shows how features can be preserved or manipulated. The undeformed cuboid is at the top left. To the right of it, it is shown how shears can be specified with higher order barycentric coordinates. However, shears are often undesired. The image at the top right (with a close-up at the bottom left) shows that the vertices of the cuboid (in particular their right angles) are preserved under rotations. The bottom row (middle and right) contains a translation. With conventional mean value coordinates (right), a shear is introduced at the vertex while angles are preserved with higher order coordinates.**

as well, higher order mean value coordinates are especially appropriate for this task since no additional effort is needed to find out if a particular point is inside or outside of a (partial) control net. Note, however, that smoothness across the faces can only be guaranteed here because the armadillo remains unchanged outside of the control net. In more complex situations, an occasional rebinding might be necessary between the deformation stages.

Since higher order barycentric coordinates are rather an extension than a direct competitor of existing barycentric coordinate schemes which are not able to interpolate rotations and other linear transformations directly, we only briefly compare the two. Figure 5.2 shows how undesired shears during a translation are prevented by higher order coordinates. In Figure 5.4, it can be seen that higher order coordinates can achieve much smoother deformations with the same number of control points by specifying rotations as derivatives. However, it can also be seen that these rotations must be specified to achieve best results if the desired deformation involves such rotations. Figure 5.5 shows that higher order mean value coordinates overcome a problem of conventional mean value coordinates that motivated the development of the positive mean value coordinates and the harmonic coordinates. Like these, higher order mean value coordinate can handle highly non-convex control nets without “repelling” artifacts. This is due to the choice of *m*, which is always non-negative.

The only other higher order barycentric coordinates, we know of, are Shep-

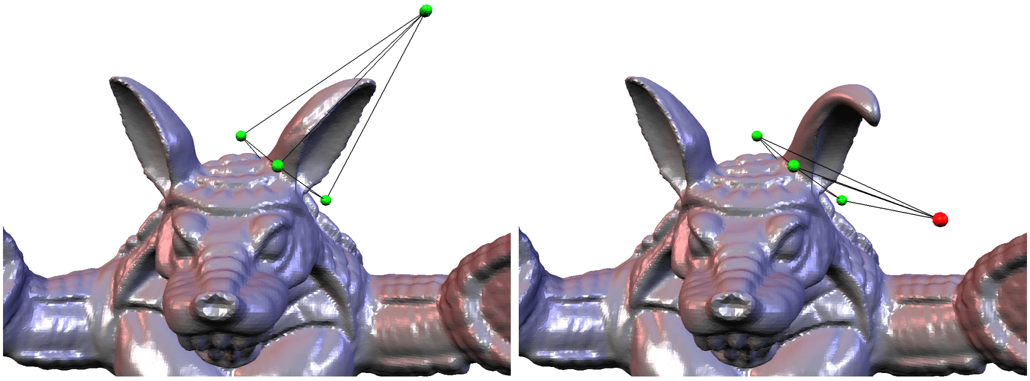


**Figure 5.3: Higher order barycentric coordinates are well-suited for partial deformations. Maximum local control over the deformation is gained by deforming only the part of the model within the control net. Furthermore, restricting the deformation enables large models like the armadillo to be deformed interactively. Using (higher order) mean value coordinates, mesh vertices can be automatically classified as either inside or outside the control net without additional cost, even in the case of complex control nets. The above deformation was specified by moving and rotating the red control vertex. The derivative constraints at the base of the pyramid ensure a smooth transition between the deformed ear and the undeformed part of the armadillo.**

ard's coordinates. However, although they achieve similar results to higher order mean value coordinates for simple control nets, they cannot handle complex, non-convex situations as demonstrated in Figures 5.4 and 5.5. This is probably due to their more global nature. Although higher order mean value coordinate are in principle global as well, they take the structure of the control net into account such that space points are mainly influenced by control points that are nearby—measured within the control net.

To be more specific, let's look at the examples again. In Figure 5.4, points on the faces of the control net are only influenced by mean value coordinates of the vertices of the respective face. This keeps these points fixed to the face. Using Shepard's coordinates, all coordinates contribute to the deformed position and points tend to move towards an average location. A similar effect can be seen in Figure 5.5: Only in the immediate vicinity of a control point, the influence of the respective Shepard coordinate is dominating and pulls part of the finger to the new position. For points farther away from a specific control point, all coordinates are roughly equal. Therefore, the respective parts of the middle finger remain in place during the deformation since most of the control points are not moved. Using mean value coordinates, however, the (interior of the) middle finger is mainly influenced by the control points of the middle finger. Consequently, the whole

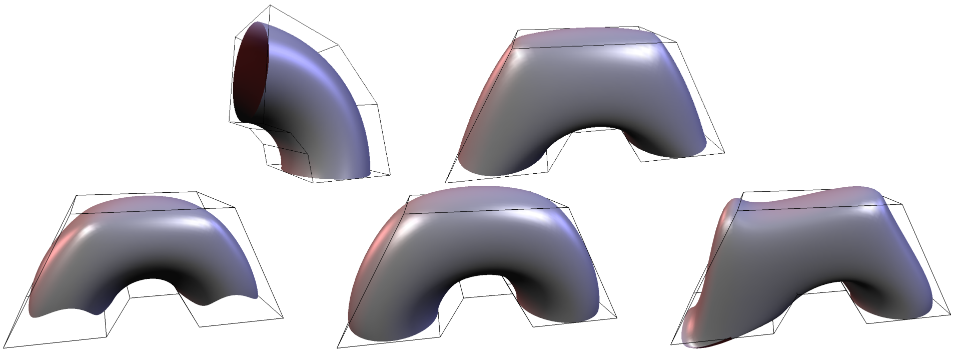


Figure 5.4: We compare different types of barycentric coordinates using the torus example from Figure 4.3. At the top left is the original model, the quarter torus, and its control net. To its right is a deformation with conventional 3D mean value coordinates. The bottom row shows the same deformation with higher order barycentric coordinates where additional rotations can be specified at the control points (by 0, 30, 60, and 90 degrees, respectively). If this is done (bottom left and middle), the result looks much smoother. Nevertheless, the deformation at the bottom left, which was done with Shepard interpolation, shows an undesired “shrinking” effect that does not occur with higher order mean value coordinates shown to its right. However, if such a rotational deformation is performed with higher order coordinates while leaving the derivatives unchanged (as the identity), the deformation at the bottom right is obtained.

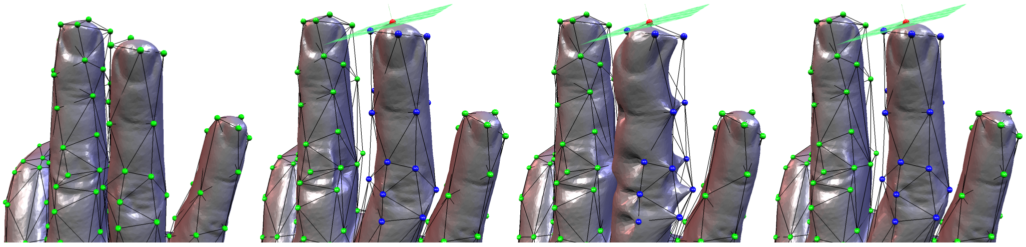


Figure 5.5: Highly non-convex models like the hand (left) are difficult for many conventional barycentric coordinate techniques. If mean value coordinates are used (second to the left) a “repelling” effect occurs in the interior of the control net when the ring finger is moved to the right. This is especially visible at the joint of the middle finger. Shepard interpolation (second to the right) also fails to give a pleasing deformation. Higher order mean value coordinates (right) create an interpolation without strong artefacts.

finger follows the movement of the control points.

### 5.3.4 Running time

As noted in Section 5.3.1, both the binding and the interpolation step can be done in $O(V \cdot P)$ time. Since the interpolation function $f_{\lambda,\mathbf{f},\mathbf{D}}$ for higher order barycentric is slightly more complex than $f_{\lambda,\mathbf{f}}$, its evaluation also takes longer (by a constant factor). Nevertheless, in our experience, this was still fast enough to allow interactive manipulation for most of our models by dragging and rotating our control points on the screen. Only the hand model, with 50085 vertices and 334 control points was considerably slower. Here, the binding step took (once) 1:17 min (compared to 1:14 min for the conventional mean value coordinates) on a AMD Athlon$^{\mathrm{TM}}$ 64 X2 Dual Core Processor 3800+ with 2 GHz. The deformation step took slightly above 1 second for this model (compared to 0.8 seconds). Further speed-ups are undoubtedly achievable. It would be possible, for example, to recompute only the contribution of those control points that were actually modified when evaluating $f_{\lambda,\mathbf{f},\mathbf{D}}$. This would yield a running time of $O(P)$.

## 5.4 Summary and Future Work

We introduced a new type of barycentric coordinates which we called higher order barycentric coordinates, as they allow the interpolation of not only function values, but linear functions. When used for space deformations, they introduce a new means of manipulating objects. They can specify rotations and other linear transformations directly without the need to "simulate" such a transformation by moving a group of close-by control points. They can also be used to manipulate only parts of an object since the derivative constraints ensure a smooth transition between deformed and undeformed parts of the model.

Furthermore, we suggested a method for modifying existing barycentric coordinates to create higher order barycentric coordinates. Therefore, they can be considered as a possible extension for existing coordinates rather than a completely new type. If we nevertheless compare higher order mean value coordinates and conventional barycentric coordinates, higher order mean value coordinates are the only non-negative, $C^1$-continuous (proven only for $\mathbb{R}^2$) barycentric coordinate functions, which are defined everywhere in $\mathbb{R}^n$, that we know of. If we compare them with Shepard's coordinates, our coordinates are clearly superior with regard to deformations since Shepard's coordinates satisfy our definition of higher order coordinates but can not capture the shape of an object.

Unfortunately, the higher order barycentric coordinates we introduced retain one disadvantage of conventional barycentric coordinates: If several polytopes are used for the control net, the coordinates are in general not smooth across polytope faces. This would be necessary to take full advantage of the method of partial deformations described for the armadillo model. To change this, the derivative property has to be extended such that the coordinate derivatives at the faces in the direction orthogonal to these faces are zero as well. Details of such a construction remain an opportunity for future work. We would also like to construct coordinates of still higher order. By specifying not only first, but also second derivatives, a bend, which is basically a change of rotations, could be determined at a single control point. Furthermore, the space of higher order coordinates should be systematically explored as has been done for generalized barycentric coordinates with respect to polytopes [FHK06, JLW07]. This would allow the selection of higher order barycentric coordinates that are tailored to particular needs.

# Part II

# Mean Value Bézier Maps

# Chapter 6

# Mean Value Bézier Surfaces over Quadratic Domains

Bernstein polynomials are a classical tool in Computer Aided Design to create smooth maps with a high degree of local control. In particular, they are used to define Bézier surfaces, which are parameterized surfaces whose parameterization is given by a linear combination of Bernstein polynomials in barycentric coordinates. Usually, Wachspress coordinates are used to obtain tensor product Bézier surfaces. In this chapter, we investigate the potential of mean value coordinates to design mean value Bézier surfaces over quadratic domains. When used to construct Bézier patches, they offer additional control points without raising the polynomial degree. In the next chapter, we will consider mean value Bézier surfaces over arbitrary (polygonal) domains and generalize mean value Bézier surfaces to mean value Bézier maps.

## 6.1 Bézier Theory and Barycentric Coordinates

Bézier surfaces are based on the works of Bézier, de Casteljau, and Forrest [Béz68, dC59, For72] and remain an important tool in Computer Aided Design today. We use the more general notion of *Bézier maps* to denote polynomial functions $f : \mathbb{R}^d \to \mathbb{R}^e$ in the form of simplicial Bézier maps

$$f(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda) \tag{6.1}$$

or tensor product Bézier maps

$$f(\mathbf{x}) = \sum_{i_1,\ldots i_d=0}^{n} b_{i_1\ldots i_d} \prod_{j=1}^{d} B_{i_j}^n(x_j) \tag{6.2}$$

where $\lambda := \lambda(\mathbf{x})$ are barycentric coordinates of $\mathbf{x} := (x_1, \dots x_d)$ with respect to a domain simplex (or polytope) $P \subset \mathbb{R}^d$ with vertices $\{\mathbf{v}_1, \dots \mathbf{v}_k\}$ ($k = d + 1$ if $P$ is a simplex) while (6.2) is defined over the domain $[0, 1]^d$. Furthermore, $n$ is the polynomial degree, $b_\alpha \in \mathbb{R}^e$ and $b_{i_1 \dots i_d} \in \mathbb{R}^e$ are the control points, and $B^n_\alpha$ and $B^n_i$ are the Bernstein polynomials defined by

$$B^n_\alpha(\lambda) = \frac{n!}{\alpha!}\lambda^\alpha , \qquad B^n_i(x) = \binom{n}{i}(1 - x)^{n-i}x^i \tag{6.3}$$

where we use the standard multi-index notation $\alpha := (\alpha_1, \dots \alpha_k) \in \mathbb{N}^k$ with $|\alpha| := \sum_i \alpha_i$, $\alpha! := \prod_i \alpha_i!$, and $\lambda^\alpha := \prod_i \lambda_i^{\alpha_i}$. In this chapter, we focus mainly on Bézier surfaces. In this case, $d = 2$, $e = 3$, and we denote the components of $\mathbf{x} = (x_1, x_2)$ by $(x, y)$.

The representations (6.1) and (6.2) are even more closely related to each other than it may appear at a first glance. If we set

$$\lambda^{\mathrm{W}}_1(\mathbf{x}) := (1 - x)(1 - y),\ \lambda^{\mathrm{W}}_2(\mathbf{x}) := x(1 - y),\ \lambda^{\mathrm{W}}_3(\mathbf{x}) := xy,\ \lambda^{\mathrm{W}}_4(\mathbf{x}) := (1 - x)y,$$

we can express tensor product Bézier surfaces in the form of (6.1) as has been noted in [LD89]. Using multi-indices $\alpha := (\alpha_1, \alpha_2, \alpha_3, \alpha_4) \in \mathbb{N}^4$, we obtain

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i,j=0}^{n} b_{ij} B^n_i(x) B^n_j(y) = \sum_{i,j=0}^{n} b_{ij} \binom{n}{i}\binom{n}{j}(1 - x)^{n-i}x^i(1 - y)^{n-j}y^j && (6.4)\\ &= \sum_{i,j=0}^{n} b_{ij} \sum_{\substack{\alpha_2+\alpha_3=i\\ \alpha_3+\alpha_4=j\\ |\alpha|=n}} \frac{n!}{\alpha!}(1 - x)^{n-i}x^i(1 - y)^{n-j}y^j && (6.5)\\ &= \sum_{i,j=0}^{n} b_{ij} \sum_{\substack{\alpha_2+\alpha_3=i\\ \alpha_3+\alpha_4=j\\ |\alpha|=n}} \frac{n!}{\alpha!}\lambda^{\mathrm{W}}(\mathbf{x})^\alpha = \sum_{|\alpha|=n} b_{\alpha_2+\alpha_3,\alpha_3+\alpha_4} B^n_\alpha(\lambda^{\mathrm{W}}(\mathbf{x})). \end{aligned}$$

Since $B^n_i(x)B^n_j(y)$ and $B^n_\alpha(\lambda^{\mathrm{W}}(\mathbf{x}))$ form both a partition of unity, the equality of (6.4) and (6.5) can be deduced by comparing the coefficients of

$$\begin{aligned} &\sum_{i,j=0}^{n} \binom{n}{i}\binom{n}{j}(1 - x)^{n-i}x^i(1 - y)^{n-j}y^j = 1\\ &\qquad = \sum_{|\alpha|=n} \frac{n!}{\alpha!}\lambda^{\mathrm{W}}(\mathbf{x})^\alpha = \sum_{i,j=0}^{n}\Bigg(\sum_{\substack{\alpha_2+\alpha_3=i\\ \alpha_3+\alpha_4=j\\ |\alpha|=n}} \frac{n!}{\alpha!}\Bigg)(1 - x)^{n-i}x^i(1 - y)^{n-j}y^j. \end{aligned}$$

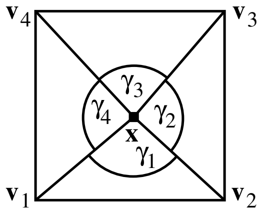


**Figure 6.1: Notation for the definition of mean value coordinates.**

The coordinate functions $\lambda_i^{\mathrm{W}}$ are just the Wachspress coordinates [Wac75] with respect to the unit square.

In this chapter, we explore the potential of substituting mean value coordinates [Hui91, Flo03], which we denote by $\lambda_i$, for Wachspress coordinates in the definition of Bézier surfaces. Using the notation from Fig. 6.1, they are defined by

$$\lambda_i = \frac{w_i}{\sum_j w_j}, \qquad w_i = \frac{\tan\frac{\gamma_{i-1}}{2} + \tan\frac{\gamma_i}{2}}{\|\mathbf{x} - \mathbf{v}_i\|} \tag{6.6}$$

in the interior of a convex polygon, and they have a continuous extension to the boundary [HF06]. We list some of their properties.

**6.1 Proposition (Hormann & Floater [HF06]).** *Mean value coordinates $\lambda_i$ are positive barycentric coordinates. In particular, they are positive everywhere within the unit square. In addition, they have the following properties:*

1. *Lagrange property: $\lambda_i(\mathbf{v}_j) = \delta_{ij}$.*
2. *Smoothness: The $\lambda_i$ are $C^\infty$ everywhere except at the vertices $\mathbf{v}_j$, where they are only $C^0$.*
3. *Linear independence: The functions $(\lambda_i(\mathbf{x}))_{i=1}^4$ are linearly independent.*
4. *Edge property: $\lambda_i$ is linear along the edges of the unit square.*

**6.2 Proposition.** *The four mean value coordinate functions with respect to the unit square $[0,1]^2$ are symmetric to themselves and to each other:*

$$\begin{aligned} \lambda_1(x,y) &= \lambda_1(y,x), \quad \lambda_3(x,y) = \lambda_3(y,x),\\ \lambda_2(x,y) &= \lambda_2(1-y,1-x), \quad \lambda_4(x,y) = \lambda_4(1-y,1-x),\\ \lambda_1(x,y) &= \lambda_2(1-x,y) = \lambda_3(1-x,1-y) = \lambda_4(x,1-y). \end{aligned}$$

Although mean value coordinates are only $C^0$ at the vertices, we can define partial derivatives by taking the limit over difference quotients within the square. This yields by Proposition 6.1.4 $\frac{\partial}{\partial x}\lambda_1(0,0) = \frac{\partial}{\partial x}\lambda_1(1,0) = \frac{\partial}{\partial y}\lambda_1(0,0) =$

$\frac{\partial}{\partial y}\lambda_1(0,1) = -1$, $\frac{\partial}{\partial x}\lambda_1(0,1) = \frac{\partial}{\partial x}\lambda_1(1,1) = \frac{\partial}{\partial y}\lambda_1(1,0) = \frac{\partial}{\partial y}\lambda_1(1,1) = 0$, and respectively for the other coordinate functions. But note that $\lim_{y\to 0, y>0} \frac{\partial}{\partial x}\lambda_1(0,y) \neq -1$ and similarly for the other $\lambda_i$.[1] Nevertheless, this is sufficient for our purposes since we can nevertheless achieve higher continuity by taking appropriate combinations of the $\lambda_i$. This is shown in the following section.

The advantage of mean value coordinates is that their Bernstein polynomials $B^n_\alpha(\lambda)$ are in general different for different multi-indices $\alpha$ while the Bernstein polynomials $B^n_\alpha(\lambda^W)$ and $B^n_{\alpha'}(\lambda^W)$ coincide if $\alpha_2+\alpha_3 = \alpha'_2+\alpha'_3$ and $\alpha_3+\alpha_4 = \alpha'_3+\alpha'_4$. This means that it is possible to define a different kind of Bézier patch by using the mean value coordinates of the unit square in (6.1) as suggested in [Flo03]. Moreover, these mean value Bézier patches have a greater number of control points than traditional Bézier patches with the same polynomial degree. In the following section, we will investigate the properties of mean value Bézier surfaces.

## 6.2 Mean Value Bézier Surfaces

Many of the properties of (triangular) Bézier surfaces can be proven by formal manipulations of the barycentric coordinates $\lambda_i$. Therefore, the respective proofs carry directly over to the case of generalized Bézier surfaces. In the following theorem, we summarize some of these results.

**6.3 Proposition.** *Let $\lambda_i$ be barycentric coordinates with respect to a polytope $P$, and let the Bernstein polynomials $B^n_\alpha$ and a Bézier map $f$ be defined as in (6.3) and (6.1). Then the following properties hold:*

1. $B^n_\alpha(\lambda) = \sum_{i=1}^k \lambda_i B^{n-1}_{\alpha-\mathbf{e}_i}(\lambda)$ *(for any Bernstein polynomial $B^m_\beta$, $|\beta| = m$, we use the convention $B^m_\beta(\lambda) := 0$ if one of the $\beta_i < 0$).*

2. *Let $(\mathbf{v}_{i_0}, \mathbf{v}_{i_1})$ be an edge of $P$, then the boundary curve $f(\lambda((1-t)\mathbf{v}_{i_0} + t\mathbf{v}_{i_1}))$ is a Bézier curve with control points $(b_{(n-j)\mathbf{e}_{i_0}+j\mathbf{e}_{i_1}})^n_{j=0}$.*

3. *$\{B^n_\alpha\}$ forms a partition of unity; if $P$ is convex and the $\lambda_i$ are positive coordinates, the partition of unity is positive within $P$. In particular, this is true for the mean value coordinates with respect to a square.*

4. *If $P$ is convex and the $\lambda_i$ are positive coordinates, the image of $P$ under $f(\lambda(\mathbf{x}))$ is contained in the convex hull of the $b_\alpha$. In particular, this is true for the mean value coordinates with respect to a square.*

[1] Our numerical experiments gave a limit value of approximately −0.7.

5. *The de Casteljau algorithm works: let* $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ *be a Bézier map with coefficients* $b_\alpha$*. For* $m \in \mathbb{N}$ *and a given* $\beta$ *with* $|\beta| = n - m$*, let* $b^m_\beta(\lambda) := \sum_{|\alpha|=m} b_{\beta+\alpha} B^m_\alpha(\lambda)$*. Then* $P(\lambda) = b^n_{\mathbf{0}}(\lambda)$ *can be computed from the* $b^0_\beta(\lambda) = b_\beta$ *via the recursive relation* $b^m_\beta(\lambda) = \sum_{i=1}^k \lambda_i b^{m-1}_{\beta+\mathbf{e}_i}(\lambda)$*.*

*(*$\mathbf{e}_i$ *denotes the multi-index with components* $(\mathbf{e}_i)_j = \delta_{ij}$*, and* $\mathbf{0}$ *denotes the multi-index with components* $\mathbf{0}_j = 0$*.)*

In the remainder of this chapter, we will consider the special case of planar mean value coordinates $\lambda_i$ with respect to a unit square $P$, $k = 4$. Next, we give the derivatives of the Bernstein polynomials in $\lambda$.

**6.4 Lemma.** *Let*

$$B^n_\alpha(\lambda) = \begin{cases} \frac{n!}{\alpha!}\lambda^\alpha & \text{if all } \alpha_i \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

*Then*

$$\frac{\partial^{|\beta|}}{\partial \lambda^\beta} B^n_\alpha(\lambda) := \frac{\partial^{|\beta|}}{\partial \lambda_1^{\beta_1} \partial \lambda_2^{\beta_2} \partial \lambda_3^{\beta_3} \partial \lambda_4^{\beta_4}} B^n_\alpha(\lambda) = \frac{n!}{(n - |\beta|)!} B^{n-|\beta|}_{\alpha-\beta}(\lambda).$$

The computation of derivatives of mean value Bézier patches with respect to $x$ and $y$, which is important to join several Bézier patches smoothly, is more challenging because these derivatives can not be expressed as a linear combination of Bernstein polynomials as it is the case with tensor product Bézier surfaces. Application of the chain rule yields:

**6.5 Lemma.** *Let*

$$f(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda),$$

*and define recursively*

$$\begin{aligned} \Delta_{0,0} b_\alpha(\mathbf{x}) &:= b_\alpha, \\ \Delta_{r+1,s} b_\alpha(\mathbf{x}) &:= \sum_{i=1}^4 \frac{\partial}{\partial x} \lambda_i(\mathbf{x}) \Delta_{r,s} b_{\alpha+\mathbf{e}_i}(\mathbf{x}), \\ \Delta_{r,s+1} b_\alpha(\mathbf{x}) &:= \sum_{i=1}^4 \frac{\partial}{\partial y} \lambda_i(\mathbf{x}) \Delta_{r,s} b_{\alpha+\mathbf{e}_i}(\mathbf{x}). \end{aligned}$$

*Then the first derivatives of* $f$ *are given by*

$$\begin{aligned} \frac{\partial}{\partial x} f(\lambda(\mathbf{x})) &= n \sum_{|\alpha|=n-1} \Delta_{1,0} b_\alpha(\mathbf{x}) B^{n-1}_\alpha(\lambda(\mathbf{x})), \\ \frac{\partial}{\partial y} f(\lambda(\mathbf{x})) &= n \sum_{|\alpha|=n-1} \Delta_{0,1} b_\alpha(\mathbf{x}) B^{n-1}_\alpha(\lambda(\mathbf{x})). \end{aligned}$$

*The second derivatives of f are given by*

$$\begin{aligned}
\frac{\partial^2}{\partial x^2} f(\lambda(\mathbf{x})) &= n \sum_{|\alpha|=n-1} \frac{\partial}{\partial x} \Delta_{1,0} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \\
&+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{2,0} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})), \\
\frac{\partial^2}{\partial y^2} f(\lambda(\mathbf{x})) &= n \sum_{|\alpha|=n-1} \frac{\partial}{\partial y} \Delta_{0,1} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \\
&+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{0,2} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})), \\
\frac{\partial^2}{\partial x \partial y} f(\lambda(\mathbf{x})) &= n \sum_{|\alpha|=n-1} \frac{\partial}{\partial y} \Delta_{1,0} b_\alpha(\mathbf{x}) \cdot B_\alpha^{n-1}(\lambda(\mathbf{x})) \\
&+ n(n-1) \sum_{|\alpha|=n-2} \Delta_{1,1} b_\alpha(\mathbf{x}) B_\alpha^{n-2}(\lambda(\mathbf{x})).
\end{aligned}$$

We can now give continuity conditions for connecting mean value Bézier patches.

**6.6 Theorem ($C^0$-continuity).** *Let*

$$f(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda), \quad f'(\lambda') = \sum_{|\alpha|=n} b'_\alpha B_\alpha^n(\lambda') \tag{6.7}$$

*where $\lambda(\mathbf{x})$ is defined with respect to the square $[0,1]^2$, and $\lambda'(\mathbf{x})$ is defined with respect to the square $[1,2] \times [0,1]$. These patches form a continuous surface iff the control points at the connecting edge coincide, that is*

$$b_{(n-i)\mathbf{e}_2 + i\mathbf{e}_3} = b'_{(n-i)\mathbf{e}_1 + i\mathbf{e}_4} \quad \forall i = 0 \ldots n, \tag{6.8}$$

*see Figure 6.2. Respective conditions hold to join patches along the other domain boundary edges.*

*Proof.* This is an immediate consequence of Proposition 6.3.2. □

To compare higher order derivatives, we need to know more about the partial derivatives of the mean value coordinates. Although the partial derivatives have been computed in [DF08] for transfinite coordinates, a closed formula for the polygonal case is not known. We circumvent the problem of finding an explicit expression for the derivatives with the following lemma.

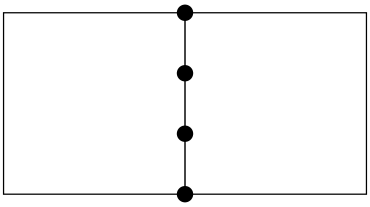


**Figure 6.2: Sketch of two cubic Bézier patches, which are connected with $C^0$-continuity. The control points marked with black dots must coincide.**

**6.7 Lemma.** *Let $\lambda_1, \dots \lambda_4$ be the mean value coordinates of the unit square. Then*

$$\begin{aligned}
\frac{\partial}{\partial x}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_2(\mathbf{x}) &= \frac{\partial}{\partial x}\lambda_3(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_4(\mathbf{x}) = 0, && (6.9)\\
-\left(\frac{\partial}{\partial y}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_2(\mathbf{x})\right) &= \frac{\partial}{\partial y}\lambda_3(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_4(\mathbf{x}) = 1,\\
-\left(\frac{\partial}{\partial x}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_4(\mathbf{x})\right) &= \frac{\partial}{\partial x}\lambda_2(\mathbf{x}) + \frac{\partial}{\partial x}\lambda_3(\mathbf{x}) = 1, && (6.10)\\
\frac{\partial}{\partial y}\lambda_1(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_4(\mathbf{x}) &= \frac{\partial}{\partial y}\lambda_2(\mathbf{x}) + \frac{\partial}{\partial y}\lambda_3(\mathbf{x}) = 0.
\end{aligned}$$

*Proof.* The linear precision property (2.2) of the mean value coordinates implies that linear functions $f$ are correctly interpolated by $\sum_{i=1}^{4} \lambda_i(\mathbf{x}) f(\mathbf{v}_i) = f(\mathbf{x})$ where $\mathbf{v}_i$ are the vertices of the unit square. By choosing $f(\mathbf{x}) := y$, we obtain $\lambda_3(\mathbf{x}) + \lambda_4(\mathbf{x}) = y$. Differentiating by $x$ yields the second equality in (6.9). Everything else follows completely analogous and by using the partition of unity property (2.1). □

**6.8 Theorem ($C^1$-continuity I).** *A mean value Bézier patch $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda)$ is $C^1$ everywhere (in particular at the vertices) iff it satisfies the smoothness condition*

$$b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i+2}} = b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i+1}} + b_{(n-1)\mathbf{e}_i+\mathbf{e}_{i-1}} - b_{n\mathbf{e}_i},\ i = 1 \dots 4 \tag{6.11}$$

*(indices of* $\mathbf{e}$ *modulo* 4*), see Figure 6.3.*

*Proof.* By the symmetry of the mean value coordinates, it is sufficient to prove the claim at one vertex and for one partial derivative. By Lemma 6.5 (compare also (6.16)), we obtain

$$\begin{aligned}
\lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x} f(\lambda(\mathbf{x})) = n\Bigg( &(b_{n\mathbf{e}_3} - b_{(n-1)\mathbf{e}_3+\mathbf{e}_4}) \lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x}\lambda_3(\mathbf{x})\\
&+ (b_{(n-1)\mathbf{e}_3+\mathbf{e}_2} - b_{(n-1)\mathbf{e}_3+\mathbf{e}_1}) \lim_{\substack{\mathbf{x}\to(1,1)\\ \mathbf{x}\neq 1}} \frac{\partial}{\partial x}\lambda_2(\mathbf{x})\Bigg) \qquad (6.12)
\end{aligned}$$

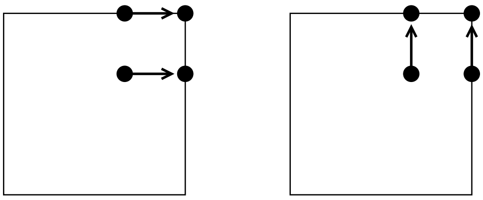


**Figure 6.3: Two equivalent sketches of a cubic Bézier patch, which is $C^1$-continuous at the top right corner. The control point differences indicated by the two arrows must coincide.**

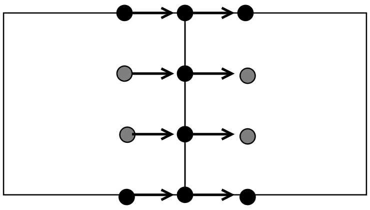


**Figure 6.4: Sketch of two cubic Bézier patches, which are connected with $C^1$-continuity. The control point differences indicated by pairs of arrows in a common row must coincide. Note that each of the grey-shaded dots actually represents two different control points. Therefore, six conditions have to be met.**

and

$$\frac{\partial}{\partial x} f(\lambda(1,1)) = n(b_{n\mathrm{e}_3} - b_{(n-1)\mathrm{e}_3+\mathrm{e}_4}). \tag{6.13}$$

We have $C^1$-continuity if and only if (6.12) and (6.13) are equal. Using (6.10), we obtain the claim. □

**6.9 Theorem ($C^1$-continuity II).** *Let $f$ and $f'$ be given as in (6.7). Let us denote $b_{ijkl} := b_{i\mathrm{e}_1+j\mathrm{e}_2+k\mathrm{e}_3+l\mathrm{e}_4}$ and correspondingly for $b'$. Then $f$ and $f'$ form a $C^1$-continuous surface if (6.8) and (6.11) are satisfied and*

$$b_{0,n-i,i,0} - b_{0,n-i,i-1,1} = b'_{n-i,0,1,i-1} - b'_{n-i,0,0,i} \quad \forall i = 1 \ldots n \tag{6.14}$$

*and*

$$b_{0,n-i,i,0} - b_{1,n-i-1,i,0} = b'_{n-i-1,1,0,i} - b'_{n-i,0,0,i} \quad \forall i = 0 \ldots n-1, \tag{6.15}$$

*see Figure 6.4. Respective conditions hold to join patches along the other domain boundary edges.*

*Proof.* The derivatives in $y$-direction coincide by Prop. 6.3.2. In $x$-direction, we obtain by Lemma 6.5 and (6.9)

$$
\begin{aligned}
&\frac{\partial}{\partial x} f(\lambda(1,y)) = \\
&\qquad n\Bigg(\sum_{i=0}^{n-1} (b_{0,n-1-i,i+1,0} - b_{0,n-1-i,i,1})\binom{n-1}{i}\left(\frac{\partial}{\partial x}\lambda_3 \cdot \lambda_2^{n-1-i}\lambda_3^{i}\right)(1,y) \\
&\qquad\qquad + (b_{0,n-i,i,0} - b_{1,n-1-i,i,0})\binom{n-1}{i}\left(\frac{\partial}{\partial x}\lambda_2 \cdot \lambda_2^{n-1-i}\lambda_3^{i}\right)(1,y)\Bigg)
\end{aligned}
\tag{6.16}
$$

and

$$
\begin{aligned}
&\frac{\partial}{\partial x} f'(\lambda'(1,y)) = \\
&\qquad n\Bigg(\sum_{i=0}^{n-1} (b'_{n-1-i,0,1,i} - b'_{n-1-i,0,0,i+1})\binom{n-1}{i}\left(\frac{\partial}{\partial x}\lambda'_3 \cdot {\lambda'_1}^{n-1-i}{\lambda'_4}^{i}\right)(1,y) \\
&\qquad\qquad + (b'_{n-1-i,1,0,i} - b'_{n-i,0,0,i})\binom{n-1}{i}\left(\frac{\partial}{\partial x}\lambda'_2 \cdot {\lambda'_1}^{n-1-i}{\lambda'_4}^{i}\right)(1,y)\Bigg).
\end{aligned}
$$

By the symmetry of the coordinate functions, we obtain $\lambda_2^{n-1-i}(1,y) = {\lambda'_1}^{n-1-i}(1,y)$, $\lambda_3^{i}(1,y) = {\lambda'_4}^{i}(1,y)$, and $\frac{\partial}{\partial x}\lambda_3(1,y) = -\frac{\partial}{\partial x}\lambda'_4(1,y) \overset{(6.9)}{=} \frac{\partial}{\partial x}\lambda'_3(1,y)$. By comparing the coefficients, we see that (6.14) and (6.15) are sufficient conditions for $C^1$-continuity. □

**6.10 Theorem ($C^2$-continuity I).** *Let $f$ and $f'$ be given as in (6.7). Then $f$ and $f'$ form a $C^2$-continuous surface ($C^1$-continuous at the vertices) if (6.8), (6.14) and (6.15) are satisfied, and, for all $i = 0 \ldots n-1$,*

$$
b_{0,n-1-i,i+1,0} + b_{1,n-1-i,i,0} - b_{0,n-1-i,i,1} - b_{0,n-i,i,0} = 0, \tag{6.17}
$$

*and, for all $i = 0 \ldots n-2$,*

$$
\begin{aligned}
b_{2,n-2-i,i,0} + b_{0,n-i,i,0} - 2b_{1,n-1-i,i,0} &= b'_{n-2-i,2,0,i} + b'_{n-i,0,0,i} - 2b'_{n-1-i,1,0,i}, \qquad (6.18) \\
b_{0,n-1-i,i+1,0} + b_{1,n-2-i,i,1} - b_{1,n-2-i,i+1,0} - b_{0,n-1-i,i,1}& \\
= b'_{n-1-i,0,0,i+1} + b'_{n-2-i,1,1,i} &- b'_{n-2-i,1,0,i+1} - b'_{n-1-i,0,1,i},
\end{aligned}
$$

*and*

$$
\begin{aligned}
&b_{0,n-2-i,i+2,0} + b_{0,n-2-i,i,2} - 2b_{0,n-2-i,i+1,1} \\
&\qquad = b'_{n-2-i,0,0,i+2} + b'_{n-2-i,0,2,i} - 2b'_{n-2-i,0,1,i+1},
\end{aligned}
\tag{6.19}
$$

*see Figure 6.5 (right). Respective conditions hold to join patches along the other domain boundary edges.*

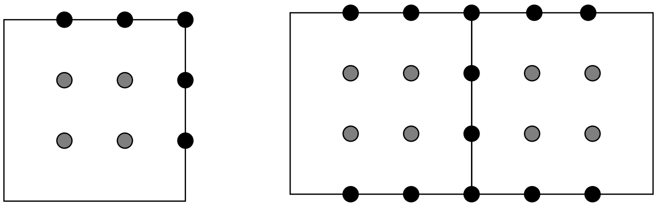


**Figure 6.5: The control points involved in the conditions for $C^2$-continuity. (The grey control points actually represent two different control points.) Left: the control points for $C^2$-continuity at the top right corner of a cubic Bézier patch. Right: the control points for two cubic Bézier patches connected with $C^2$-continuity.**

*Proof.* The second derivatives in $y$-direction coincide by Prop. 6.3.2. In $x$-direction, we obtain by Lemma 6.5 and by differentiating (6.9) and (6.10)

$$
\begin{aligned}
&\frac{\partial^2}{\partial x^2} f(\lambda(1,y)) = \\
&\quad n\Bigg(\sum_{i=0}^{n-1} (b_{0,n-1-i,i+1,0} - b_{0,n-1-i,i,1} - b_{0,n-i,i,0} + b_{1,n-1-i,i,0}) \\
&\qquad\qquad\qquad \cdot \binom{n-1}{i} \left(\frac{\partial^2 \lambda_3}{\partial x^2} \lambda_2^{n-1-i} \lambda_3^i\right)\Bigg)(1,y) \\
&\quad +n(n-1)\Bigg(\sum_{i=0}^{n-2} (b_{2,n-2-i,i,0} + b_{0,n-i,i,0} - 2b_{1,n-1-i,i,0}) \binom{n-2}{i} \left(\left(\frac{\partial \lambda_2}{\partial x}\right)^2 \lambda_2^{n-2-i} \lambda_3^i\right) \\
&\qquad + (b_{0,n-1-i,i+1,0} + b_{1,n-2-i,i,1} - b_{1,n-2-i,i+1,0} - b_{0,n-1-i,i,1}) \\
&\qquad\qquad\qquad \cdot 2\binom{n-2}{i} \left(\frac{\partial \lambda_2}{\partial x} \frac{\partial \lambda_3}{\partial x} \lambda_2^{n-2-i} \lambda_3^i\right) \\
&\qquad + (b_{0,n-2-i,i+2,0} + b_{0,n-2-i,i,2} - 2b_{0,n-2-i,i+1,1}) \\
&\qquad\qquad\qquad \cdot \binom{n-2}{i} \left(\left(\frac{\partial \lambda_3}{\partial x}\right)^2 \lambda_2^{n-2-i} \lambda_3^i\right)\Bigg)(1,y)
\end{aligned}
\tag{6.20}
$$

and a similar term for $\frac{\partial^2}{\partial x^2} f'(\lambda'(1,y))$. The rest follows again by comparing the coefficients of $f$ and $f'$. With (6.14), (6.15), and $\frac{\partial^2}{\partial x^2}\lambda_3(1,y) = \frac{\partial^2}{\partial x^2}\lambda'_4(1,y) \overset{(6.9)}{=} -\frac{\partial^2}{\partial x^2}\lambda'_3(1,y)$ we deduce (6.17) from the second line of (6.20). Note that (6.11) is a special case of (6.17) and that an analogon of (6.17) holds for the control points $b'_\alpha$ as well because of (6.14) and (6.15). The other three lines of (6.20) lead to (6.18)–(6.19). When computing the mixed derivatives of $f$ and $f'$, we observe that they already coincide by (6.14) and (6.15). □

**6.11 Theorem ($C^2$-continuity II).** *A mean value Bézier patch given by $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda)$ is $C^2$ everywhere (in particular at the vertices) if it satisfies the smoothness conditions (6.17) for $i = 0, 1, n-1, n$ and, additionally*

$$b_{2,0,n-2,0} + b_{0,2,n-2,0} - 2b_{1,1,n-2,0} = b_{0,0,n,0} + b_{0,0,n-2,2} - 2b_{0,0,n-1,1}, \qquad (6.21)$$

*see Figure 6.5 (left). Respective conditions hold for the other vertices.*

*Proof.* The proof proceeds along the lines of the proof of Theorem 6.8. We use Lemma 6.5 to compute $\frac{\partial^2}{\partial x^2} f(\lambda(1,1))$ and $\lim_{\mathbf{x}\to(1,1),\ \mathbf{x}\neq 1} \frac{\partial^2}{\partial x^2} f(\lambda(\mathbf{x}))$ and obtain (6.21) by comparing the coefficients. Note that we obtain the condition

$$b_{2,0,n-2,0} + b_{0,2,n-2,0} - 2b_{1,1,n-2,0} = b_{0,1,n-1,0} + b_{1,0,n-2,1} - b_{1,0,n-1,0} - b_{0,1,n-2,1}$$

as well, but this equation is already satisfied by (6.21) and (6.17). We get additional constraints from evaluating $\frac{\partial^2}{\partial y^2} f(\lambda(1,1))$ and $\lim_{\mathbf{x}\to(1,1),\mathbf{x}\neq 1} \frac{\partial^2}{\partial y^2} f(\lambda(\mathbf{x}))$ and from comparing $\frac{\partial^2}{\partial x \partial y} f(\lambda(1,1))$ with its respective limit values. But again, these constraints can already be deduced from (6.21) and (6.17). □

Theorems 6.10 and 6.11 together give sufficient conditions to join mean value patches to form a $C^2$-continuous surface.

## 6.3 Results

In Figure 6.6, we show a mean value Bézier surface that is constructed from four patches of order 2. Here, only the conditions for $C^1$-continuity along the common boundaries were met ((6.8), (6.14), and (6.15)). The central vertex is only $C^0$. As a result, the joining lines and the central vertex are visible in the shading.

In Figure 6.7, we demonstrate that mean value Bézier surfaces and tensor product Bézier surfaces are qualitatively comparable. These surfaces are constructed from four patches of order 3, and the same control points were used in both cases. Here, all constraints to achieve a $C^2$-continuous surface were enforced, except for the central vertex, which is only $C^1$ for the mean value Bézier surface. This results in two visually indistinguishable surfaces. Only if we compare the isolines of $\frac{\partial^2}{\partial x^2} f$, the $C^1$-continuous vertex can be discerned.

## 6.4 Summary and Future Work

We observe that mean value Bézier surfaces are well-suited for modeling surfaces. Although the advantage of the greater number of control points (compared to tensor product Bézier surfaces of the same degree) is diminished by a greater

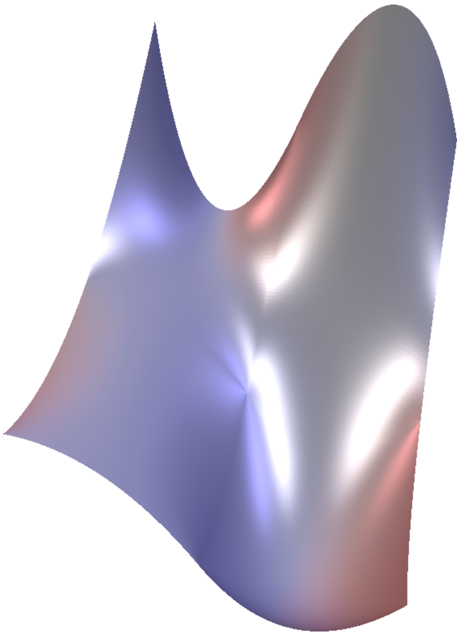


**Figure 6.6: A mean value Bézier surface constructed from four patches. Here, only the conditions for $C^1$-continuity along the common boundaries and $C^0$-continuity at the center vertex were met. This leads to an unpleasant appearance.**

number of constraints, we nevertheless obtain new, interesting possibilities for surface construction. However, to ensure that the greater number of control points corresponds to additional degrees of freedom, it has to be proven that the Bernstein polynomials in mean value coordinates are linearly independent. For linear Bernstein polynomials, this is stated in Proposition 6.1. For quadratic Bernstein polynomials on the square, we could show it by inspecting the ten different polynomials. The general case, however, is still open. One additional benefit, which does not depend on the linear independence of the mean value Bernstein polynomials, is that mean value coordinates, and hence mean value Bézier patches, are well-defined for all convex and non-convex polygons, and we are not restricted to rectangular domains. This is explored further in the subsequent chapter.

Mean value coordinates were recently generalized to higher dimensions [FKR05, JSW05, LBS06]. This allows the definition of mean value Bézier hyperpatches on (hyper-)cubes. A smoothness analysis similar to the one presented in this chapter can be done for the resulting hyper-surfaces.

Finally, we remark that most of our results are not only valid for constructing Bézier surfaces with mean value coordinates but also for all other differentiable barycentric coordinates as introduced in [FHK06]. This leads to a whole family of

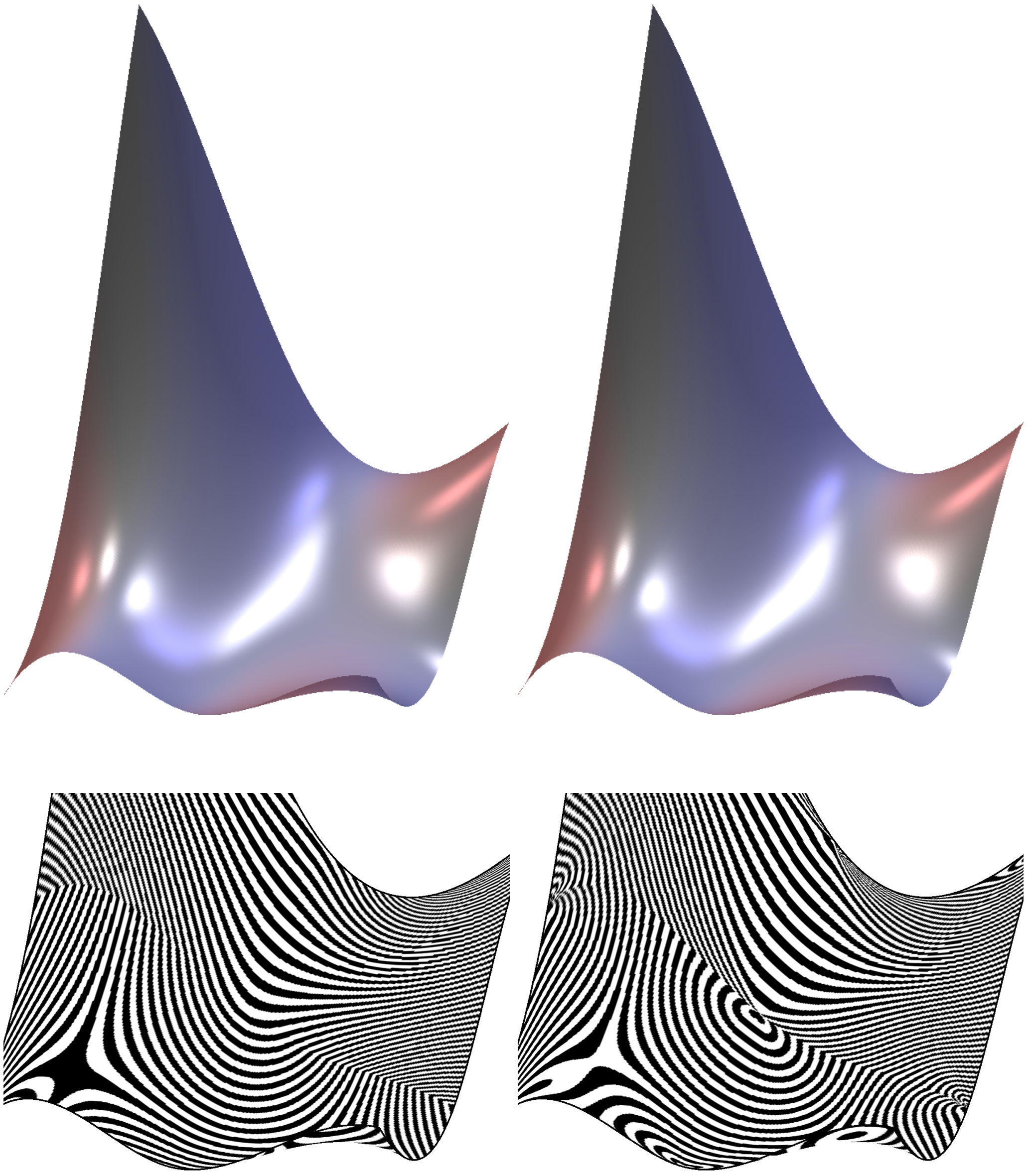


**Figure 6.7: Comparison of a tensor product Bézier surface (*left*) and a mean value Bézier surface (*right*). Both surfaces are visually indistinguishable, which shows the high quality that is achievable with mean value Bézier surfaces. Only the close-up, showing isolines of $\frac{\partial^2}{\partial x^2} f$, reveals the fine differences.**

different Bézier surfaces. Nevertheless, the convex hull property requires a special property of the mean value (and Wachspress) coordinates: positivity inside the domain. This condition is only guaranteed for few other barycentric coordinates [FHK06] and is the reason that we focused on mean value Bézier surfaces.

If Wachspress coordinates are used in our construction of Bézier surfaces, we obtain constant functions for $\frac{\partial}{\partial x}\lambda^{\mathrm{W}}$ and $\frac{\partial}{\partial y}\lambda^{\mathrm{W}}$, and the second derivatives vanish. What remains are the well-known conditions of tensor product Bézier surfaces.

# Chapter 7

# Mean Value Bézier Maps

In this chapter, we extend the approach of the previous chapter in several directions. Most importantly, we drop the restriction to quadratic domains. This requires the development of a new method to ensure smooth transitions between patch boundaries. Second, we consider not only parameterized surfaces $f : \mathbb{R}^2 \to \mathbb{R}^3$ but more general maps. This allows the use of these maps for space deformations and other purposes. For free-form deformations, it comes in handy that we can define them on arbitrary (polyhedral) domains since the classical Bernstein polynomials, defined only for simplices and parallelepipeds, can in general not directly capture the shape of arbitrary objects. Finally, we want to be able to apply our theory not only to maps based on mean value coordinates but also to maps based on other coordinates if desired.

With these extensions, we obtain smooth maps on arbitrary sets of polytopes such that the restriction to each of the polytopes is a Bernstein polynomial in mean value coordinates (or any other generalized barycentric coordinates).

## 7.1 Introduction

Bernstein polynomials are at the core of classical Computer Aided Design. They were not only used for the construction of Bézier surfaces but also to define free-form deformations of 3D space [Béz78, SP86]. More generally, they can be used to construct any kind of smooth map that requires local control.

In this chapter, we consider general Bézier maps $f : \mathbb{R}^d \to \mathbb{R}^e$. Important special cases of Bézier maps are on the one hand Bézier curves and (hyper-)surfaces where $e > d$ and usually $d = 1$ or $d = 2$, which were considered in the previous chapter. On the other hand, if $d = e$, we obtain space deformations. Sederberg and Parry [SP86] used tensor product Bernstein polynomials defined on parallelepipeds in $\mathbb{R}^3$ to specify such free-form deformations. In this case, the control points $b_{ijk}$ indicate the position and shape of the deformed parallelepiped.

However, the restriction on the shape of the domain makes it sometimes difficult to adapt the deformation to complex real objects. This restriction can be overcome by generalizing the barycentric coordinates $\lambda_i$ in (6.1) from simplices to more general polytopes. A first step in this direction was done by Loop and DeRose [LD89] who introduced coordinate functions $l_i$ in order to define Bézier surfaces over regular $k$-gons. These coordinates are a special case of the Wachspress coordinates [Wac75] that are defined inside of arbitrary convex polygons and were introduced to computer graphics by Meyer et al. [MLBD02]. A further generalization led to the definition of Wachspress coordinates for convex polytopes of higher dimensions [War96, JSWD05].

Another generalization of classical barycentric coordinates are the mean value coordinates [Hui91, Flo03], which were extended to higher dimensions later on [FKR05, JSW05]. They have the advantage of being defined for arbitrary, convex and non-convex, polytopes. Unfortunately, mean value coordinates are only $C^0$-continuous at vertices [HF06]. In the previous chapter, we addressed the latter problem and showed that the higher order discontinuities at the vertices vanish in the context of Bézier maps if the control points $b_\alpha$ satisfy certain continuity constraints. This solution, however, is only valid for Bézier maps defined on a square or (hyper-)cube. Thus, the mean value coordinates lost their greatest strength: to be defined with respect to arbitrary polytopes. Another kind of coordinate that have been used to define generalized Bézier surfaces are the Sibson coordinates [Sib80, Far90]. Since they are defined on point clouds instead of polytopes, they are not dependent on the connectivity of a particular polytope, but they can not take advantage of such a structure either.

When constructing a smooth map consisting of several polynomials that are defined on adjoining polytopes, we have to ensure that the respective polynomials connect smoothly. For connecting simplicial and tensor product polynomials, a well-developed theory is available. In [CG84], a smooth joint for a regular pentagon is constructed. Loop and DeRose [LD89, Sections 6 and 7] show how regular $k$-gons and triangles can be smoothly connected if Bernstein polynomials in Wachspress coordinates are used. This approach is extended in [LD90] where the control net for a complete surface is constructed. Unfortunately, their polynomial representation algorithm requires coordinates that are rational polynomial functions. The reason, in short, is that their proof uses the polarization of a polynomial. Hence, it cannot be carried over to mean value Bézier maps (Bézier maps based on mean value coordinates). Furthermore, their method does not cover the case of general domain polygons but only regular $k$-gons.

In this chapter, we derive constraints on the control points of Bézier maps in arbitrary generalized barycentric coordinates to obtain smooth transitions between arbitrary domain polytopes. One essential requirement, as noted in [Gol04], is to adopt an indexing scheme that is adapted to the given polytopes. We chose to use multi-indices (as has been done before in [LD89]). They correspond to the Minkowski sum approach in [Gol04]. (The set of multi-indices of degree $n$ for a polytope with vertex set $V$ corresponds to $\bigoplus_{i=1}^{n} V$.)

## 7.2 Theoretical Foundation

The basic idea for the construction of (generalized) Bézier maps is to insert (generalized) barycentric coordinates for polytopes in (6.1). Wachspress coordinates and mean value coordinates are the most prominent positive barycentric coordinates. Mean value Bézier maps have the advantage that their domain is not restricted to convex polygons. But other positive (or even non-positive) barycentric coordinates can be used as well to obtain further types of Bézier maps. A list of important properties of these Bézier maps is given in Proposition 6.3.

To join several Bézier maps smoothly, it is important to know their derivatives. In the remainder of the paper, we will assume that the $\lambda_i$ are differentiable everywhere apart from the vertices $\mathbf{v}_i$. This is true in particular for Wachspress and mean value coordinates. Using the chain rule, Lemma 6.4 implies immediately

**7.1 Lemma.** *Let*

$$f(\lambda) = \sum_{|\alpha|=n} b_\alpha B_\alpha^n(\lambda).$$

*Then the first derivatives of $f$ are given by*

$$\frac{\partial}{\partial x_i} f(\lambda(\mathbf{x})) = n \sum_{|\alpha|=n-1} \sum_{j=1}^{k} \frac{\partial}{\partial x_i} \lambda_j(\mathbf{x}) b_{\alpha+\mathbf{e}_j} B_\alpha^{n-1}(\lambda(\mathbf{x})).$$

However, the derivatives $\frac{\partial}{\partial x_i}\lambda_j$ are in general not easy to compute. The approach of the previous chapter is restricted to coordinates defined with respect to a square. Since we want to have smooth transitions of Bézier maps defined on arbitrary polytopes, we need a more general approach. In the following, we give sufficient conditions for the control points $b_\alpha$ to join arbitrary polytopes smoothly.

Basically, the control points at the common (hyper-)faces and adjacent to it must be determined by affine functions $A_\beta$ and these functions must coincide across these faces. This is visualized in Figure 7.1. The figure shows a Bézier surface and its control net from several viewpoints. The domain consists of a pentagon and an L-shaped hexagon that share two common edges (shown in black below the surface). On the right, the control net is colored to indicate the smoothness conditions. The parts of the control net that correspond to the three common vertices of the two polygons are affine images of the domain polygons. They are colored in blue, red, and green, respectively.

We make this idea more precise in the following theorems. We begin by expressing the derivatives of a Bézier map with respect to the control points.

**7.2 Theorem.** *Let $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ be a Bézier map defined with respect to a polytope $P$ with vertices $\mathbf{v}_i$. Assume that for every multi-index $\beta$ with $|\beta| = n - 1$ an affine function $A_\beta$ exists such that $b_{\beta+\mathbf{e}_i} = A_\beta(\mathbf{v}_i)$ for all $i = 1 \ldots k$.*

*Then, the derivative of $f$ with respect to a differential operator $\partial \in \left\{\frac{\partial}{\partial x_i}\right\}$ is*

$$\partial f(\lambda(\mathbf{x})) = n \sum_{|\beta|=n-1} \partial A_\beta \cdot B^{n-1}_\beta(\lambda(\mathbf{x})).$$

*Proof.* The proof consists of a direct calculation.

$$\begin{aligned}
\partial f(\lambda(\mathbf{x})) &= n \sum_{|\beta|=n-1} \sum_{i=1}^{k} \partial\lambda_i(\mathbf{x}) b_{\beta+\mathbf{e}_i} B^{n-1}_\beta(\lambda(\mathbf{x})) && \text{Lemma 7.1}\\
&= n \sum_{|\beta|=n-1} \sum_{i=1}^{k} \partial\lambda_i(\mathbf{x}) A_\beta(\mathbf{v}_i) B^{n-1}_\beta(\lambda(\mathbf{x})) && \text{definition of } A_\beta\\
&= n \sum_{|\beta|=n-1} B^{n-1}_\beta(\lambda(\mathbf{x}))\partial \sum_{i=1}^{k} \lambda_i(\mathbf{x}) A_\beta(\mathbf{v}_i) && \text{linearity of } \partial\\
&= n \sum_{|\beta|=n-1} B^{n-1}_\beta(\lambda(\mathbf{x}))\partial A_\beta\Big(\sum_{i=1}^{k} \lambda_i(\mathbf{x})\mathbf{v}_i\Big) && \text{affine linearity of } A_\beta\\
&= n \sum_{|\beta|=n-1} B^{n-1}_\beta(\lambda(\mathbf{x}))\partial A_\beta(\mathbf{x}) && \text{linear precision (2.2) for } \lambda(\mathbf{x})\\
&= n \sum_{|\beta|=n-1} B^{n-1}_\beta(\lambda(\mathbf{x}))\partial A_\beta && A_\beta(\mathbf{x}) \text{ has constant derivative.}
\end{aligned} \tag{7.1}$$

□

In the same way, we can compute higher derivatives:

**7.3 Corollary.** *In the situation of Theorem 7.2, assume that for every multi-index $\gamma$ with $|\gamma| = n - 2$ an affine function $A'_\gamma$ exists such that $\partial A_{\gamma+\mathbf{e}_i} = A'_\gamma(\mathbf{v}_i)$ for all $i = 1 \ldots k$.*

*Then, the derivative $\partial'\partial f$ of $f$ with respect to a differential operator $\partial' \in \left\{\frac{\partial}{\partial x_i}\right\}$ is*

$$\partial'\partial f(\lambda(\mathbf{x})) = n(n-1) \sum_{|\gamma|=n-2} \partial' A'_\gamma \cdot B^{n-2}_\gamma(\lambda(\mathbf{x})).$$

*Respective statements hold for the higher derivatives of $f$.*

*Proof.* Since $\frac{1}{n}\partial f(\lambda) = \sum_{|\beta|=n-1} \partial A_\beta B^{n-1}_\beta(\lambda)$ is a Bézier map with coefficients $\partial A_\beta$, the claim follows immediately from Theorem 7.2. □

**7.4 Corollary (Smooth mean value Bézier maps).** *Let $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ be a Bézier map where the $\lambda_i$ are the mean value coordinates with respect to a polytope $P$ with vertices $\mathbf{v}_i$. Assume that an affine function $A_i$ exists such that $b_{(n-1)\mathbf{e}_i+\mathbf{e}_j} = A_i(\mathbf{v}_j)$ for all $j = 1\dots k$.*

*Then, the derivative of $f$ with respect to any differential operator $\partial \in \left\{\frac{\partial}{\partial x_i}\right\}$ has a continuous extension to $\mathbf{v}_i$ and*

$$\lim_{\mathbf{x}\to\mathbf{v}_i} \partial f(\mathbf{x}) = n\,\partial A_i.$$

*Respective statements hold for the higher derivatives of $f$.*

*Proof.* We observe that the outer sum in (7.1) collapses to a single summand if the limit $\mathbf{x} \to \mathbf{v}_i$ is considered. We obtain the claim from the remaining term. □

Finally, we obtain sufficient constraints on the $b_\alpha$ to achieve smooth Bézier maps across common (hyper-)faces of polytopes (common vertices or edges in the case of polylines or polyhedra). Note that the extent, to which these constraints are necessary as well, is not yet known.

**7.5 Corollary (Continuity across polytope boundaries).** *Let $f(\lambda) = \sum_{|\alpha|=n} b_\alpha B^n_\alpha(\lambda)$ and $f'(\lambda') = \sum_{|\alpha|=n} b'_\alpha B^n_\alpha(\lambda')$ be Bézier maps defined with respect to polytopes $P$ and $P'$ that share a common (hyper-)face $\mathbf{f}$ (without loss of generality, let corresponding vertices have the same indices; this implies $\lambda_i(\mathbf{x}) = \lambda'_i(\mathbf{x})$ for all $i$ and $\mathbf{x} \in \mathbf{f}$). Let $V := \{\mathbf{v}_{i_j}\}^l_{j=1} = \{\mathbf{v}'_{i_j}\}^l_{j=1}$ be the vertex set of $\mathbf{f}$, and let $I_V$ be the set of all multi-indices $\beta$ with $|\beta| = n-1$ such that all non-zero entries $\beta_i$ of $\beta$ correspond to vertices in $V$, $i \notin V \Rightarrow \beta_i = 0$. Assume that, for every multi-index $\beta \in I_V$, an affine function $A_\beta$ exists such that $b_{\beta+\mathbf{e}_i} = A_\beta(\mathbf{v}_i)$ for all $i = 1\dots k$ and $b'_{\beta+\mathbf{e}_i} = A_\beta(\mathbf{v}'_i)$ for all $i = 1\dots k'$.*

*Then, the derivative of $f$ and $f'$ at points $\mathbf{x} \in \mathbf{f}$ with respect to a differential operator $\partial \in \left\{\frac{\partial}{\partial x_i}\right\}$ is*

$$\partial f(\lambda(\mathbf{x})) = \partial f'(\lambda'(\mathbf{x})) = n \sum_{\beta\in I_V} \partial A_\beta \cdot B^{n-1}_\beta(\lambda(\mathbf{x})).$$

*Respective statements hold for the higher derivatives of $f$ and $f'$.*

*Proof.* Observe that (7.1) remains valid if we substitute the sum over all $\beta \in I_V$ for the sum over all $\beta$ with $|\beta| = n-1$ (for $\mathbf{x} \in \mathbf{f}$). This implies the claim. □

For Bézier surfaces, it is often sufficient that the tangent plane varies smoothly without requiring smoothness of the parameterization (“geometric continuity”). In this case, slightly weaker constraints on the control points are sufficient.

**7.6 Corollary (Geometric continuity across polytope boundaries).** *In the situation of Corollary 7.5, let $Q$ be any affine transformation of the domain $\mathbb{R}^d$ that keeps $\mathbf{f}$ fixed such that $b_{\beta+\mathbf{e}_i} = A_\beta(\mathbf{v}_i)$ for all $i = 1 \ldots k$ and $b'_{\beta+\mathbf{e}_i} = A_\beta(Q\mathbf{v}'_i)$ for all $i = 1 \ldots k'$.*

*Then $\partial f(\lambda) \cdot \partial Q = \partial f'(\lambda')$.*

*Proof.* Factoring out $Q$ in (7.1) yields the claim. □

## 7.3 Applications

In this section, we present several applications of mean value Bézier maps. Although the results obtained in the previous section are general and hold for any barycentric coordinates, Wachspress and mean value coordinates are the only known positive three-point coordinates [FHK06]. Wachspress coordinates, however, have already been used to some extent in the past in the form of tensor product Bézier maps (with parallelepipeds as domain) and S-patches [LD89] (with regular $k$-gons as domain). Therefore, it seemed more appropriate to us to use mean value Bézier maps to demonstrate our results.

In all our applications, we begin by specifying several domain polytopes and their respective control points to achieve a smooth Bézier map $f : \mathbb{R}^d \to \mathbb{R}^e$. To determine the polytope in which a point $\mathbf{x} \in \mathbb{R}^d$ lies, we can use another property of mean value coordinates: the mean value coordinates with respect to a polytope $P$ are defined in the whole space $\mathbb{R}^d$ and the denominator (for normalization) in the construction is positive if and only if $\mathbf{x}$ lies within $P$ [HF06, LBS07b]. Thus, we can automatically determine the polytope $P$ containing $\mathbf{x}$ when computing the mean value coordinates of $\mathbf{x}$ with respect to $P$.

### 7.3.1 Bézier curves and surfaces

If we choose $d = 1$ or $d = 2$ and $e > d$, Bézier maps specialize to Bézier curves and surfaces. In the case $d = 1$, however, barycentric coordinates on the unique 1-dimensional polytope, which is the 1-simplex or line segment, are uniquely determined ($t$ and $1 - t$ on $[0, 1]$). Our results coincide with the well-known theory for Bézier curves.

Therefore, we present an example of a mean value Bézier surface, that is a mean value Bézier map $f : \mathbb{R}^2 \to \mathbb{R}^3$. Figure 7.1 shows a $C^1$-continuous Bézier surface from several viewpoints. It consists of two patches of degree two. The domain is the union of a pentagon and an L-shaped hexagon, which share two common edges. Since the hexagon is not convex, some of the coordinates can become negative, and we cannot guarantee that the Bézier surface is contained in

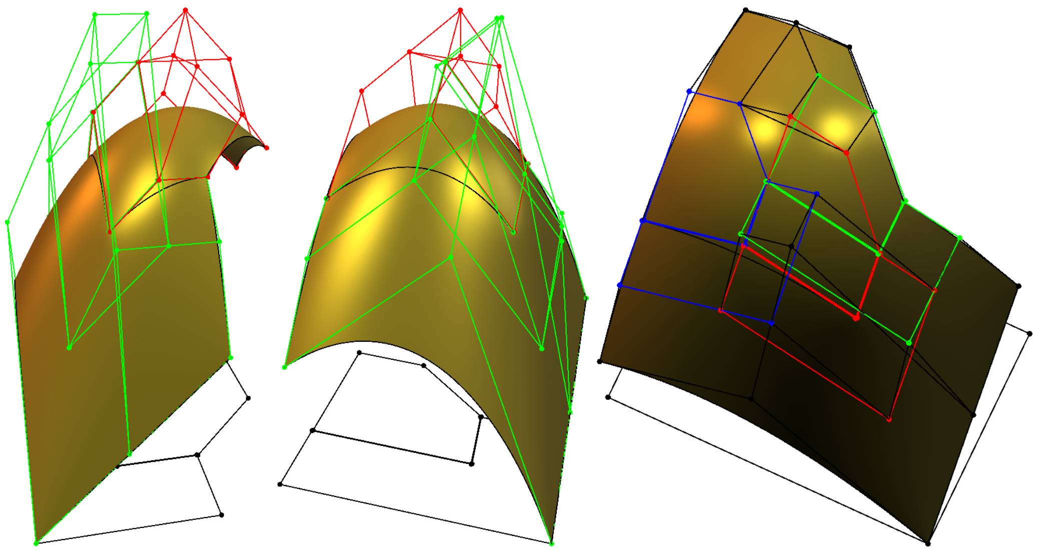


**Figure 7.1: Our method makes it possible to use non-convex polygons in the construction of Bézier surfaces. We present three views of a Bézier surface consisting of a pentagonal and an L-shaped hexagonal patch. Note that the highlights vary smoothly across the common edges.**

the convex hull of its control points. However, in practice, this posed no problem in our experiments. Nevertheless, it is probably possible to construct cases in which a mean value Bézier surface leaves the convex hull of its control points.

Note that the highlights vary smoothly across the common edges. The two domain polygons are shown in black below the surface. The control nets, which determine the shape of the surface, are also depicted. We followed the suggestion in [LD89] and drew all polygons $(b_{\beta+\mathbf{e}_i})_{i=1}^{k}$ with $|\beta| = n - 1 = 1$. (For drawing purposes, we shifted the control net belonging to the pentagon slightly to make sure that it does not overlap with the other one.) On the left and in the middle, we colored the control net for the pentagon red and the control net for the hexagon green. On the right, we chose common colors for those parts of the control net that belong to a common vertex of both polygons. They can be discerned as affine images of the domain.

### 7.3.2 Space deformations

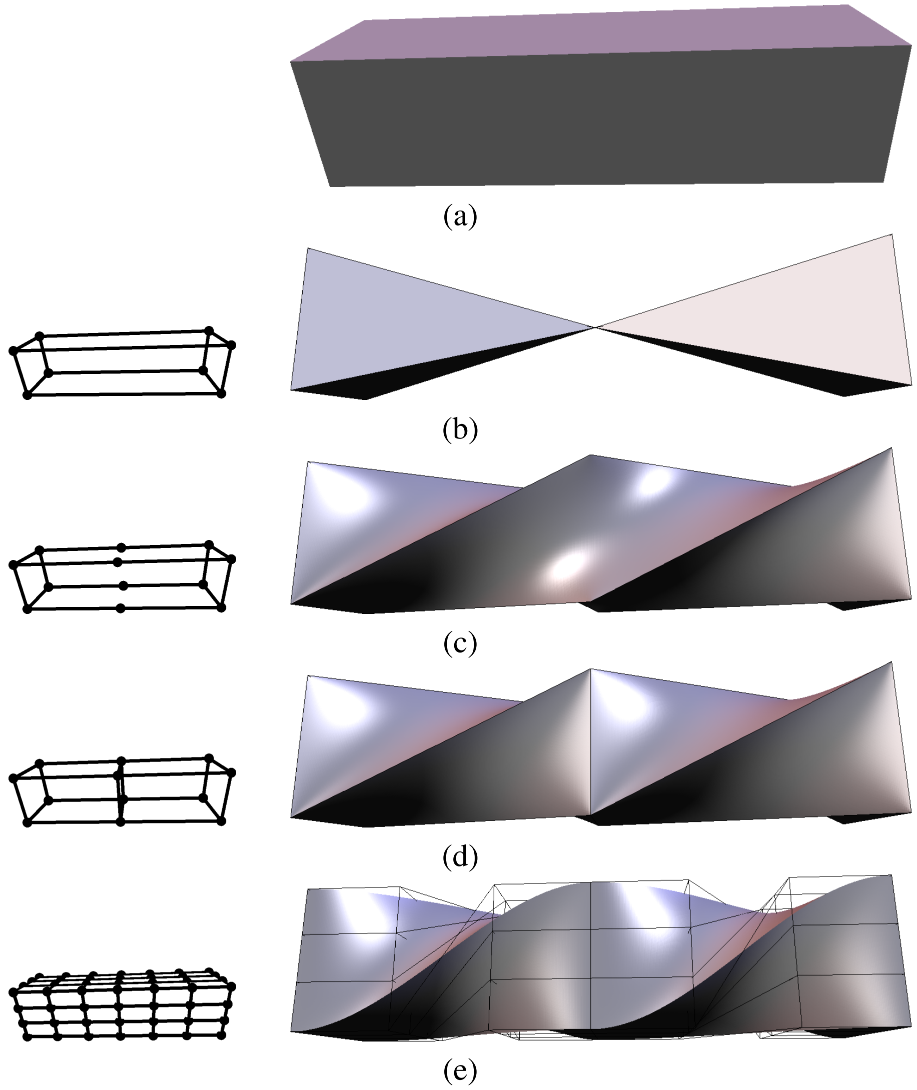
A Bézier map with $d = e$ is a space deformation of $\mathbb{R}^d$. While geometric continuity is often sufficient for Bézier curves and surfaces, we need “real” analytic continuity to obtain a smooth space deformation. Even a discontinuity of the absolute value of the derivative in a single direction may be clearly visible if a textured object is deformed.

To display a control polyhedron $P$, we note that each set $(b_{\beta+\mathbf{e}_i})_{i=1}^k$ with $|\beta| = n - 1$ corresponds naturally to the polyhedron with vertices $(\mathbf{v}_i)_{i=1}^k$. Therefore, we connect control points $b_{\beta+\mathbf{e}_i}$ and $b_{\beta+\mathbf{e}_j}$ if and only if $(\mathbf{v}_i, \mathbf{v}_j)$ is an edge in $P$.

Figure 7.2 demonstrates a space deformation of $\mathbb{R}^3$. In (a), we show the cuboid that we want to twist by 180°. We align the control polyhedron with the edges of the cuboid. (b) depicts the result if the twist is done directly with 3D mean value coordinates (that is Bernstein polynomials of degree one). The lack of local control leads to a singularity. In (c), we include four additional vertices in the middle of the long edges without changing the total shape of the control polyhedron. This allows us better local control, but $C^1$-discontinuities are introduced in the middle and at the vertices. (The bead shaped reflection at the top left corner of the cuboid indicates the $C^1$-discontinuity of mean value coordinates at the vertices.) In (d), we split the control net into two identical, adjoining control polyhedra and deform them independently of each other. This gives us the desired local control but we still have the $C^1$-discontinuities. In (e), we use a Bézier map of degree three to join the same two control polyhedra smoothly. It allows us to enforce $C^1$-continuity while maintaining local control. Observe that the $C^1$-discontinuities at the vertices have vanished as well. The control net shows how the continuity conditions are satisfied here. The left-most and right-most part is an affine image of the domain cuboids to make the deformation smooth at the respective vertices. (The left part is identically mapped, and the right part is rotated by 180° degree.) The two middle "columns" are mapped by a common affine map (both are rotated by 90° degree) to ensure a smooth transition between the adjoining control polyhedra.

As an example for stretching, we consider the cuboid once more and stretch its right half by a factor of two (Figure 7.3). To do this, we use the control net from Figure 7.2(e) again (shown in blue). This time, we depict the underlying triangle mesh of the cuboid as well (in black) to demonstrate that no discontinuity is introduced in the middle of the cuboid but the stretching increases gradually. Hence, a texture could be mapped to the stretched cuboid without artifacts due to the stretching.

Figure 7.4 shows how a complex model can be handled by specifying a control net that is adapted to the shape of the model. It also shows that Bézier maps of different degrees can be mixed under certain circumstances. (Here, the body is mapped identically.) While the body and left front leg is mapped by a degree one map, the Bézier maps for the head and the right leg have degree three. Note that we didn't need to specify all the control points manually. We explain this in the case of the head: The seven vertices of the head polyhedron can be classified into two groups: the four vertices at the neck, which are connected to the body of the cow, and the three remaining, "exterior" vertices. Since we consider a Bézier map of order three, all control points can be classified as neck control points (if two



(a)

(b)

(c)

(d)

(e)

**Figure 7.2: A cuboid shall be twisted by** $180^{\circ}$**. We present results of several methods. The small picture on the left shows the corresponding control net. (*a*) The undeformed cuboid. (*b*) Interpolation of the twist with 3D mean value coordinates. (*c*) Interpolation of the twist with 3D mean value coordinates using additional control points. (*d*) We split the cuboid into two halves and interpolate both halves with 3D mean value coordinates. (*e*) Our method. Although we use the same two halves as interpolation domains as in (*d*), the use of third order polynomials allows us to control the smoothness. If we had increased the number of control points without using higher order polynomials, we would have introduced new discontinuities as in (*c*) and (*d*).**

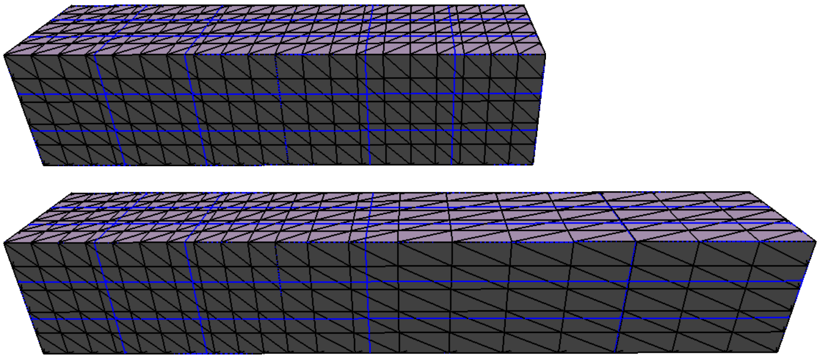


Figure 7.3: A stretched cuboid.

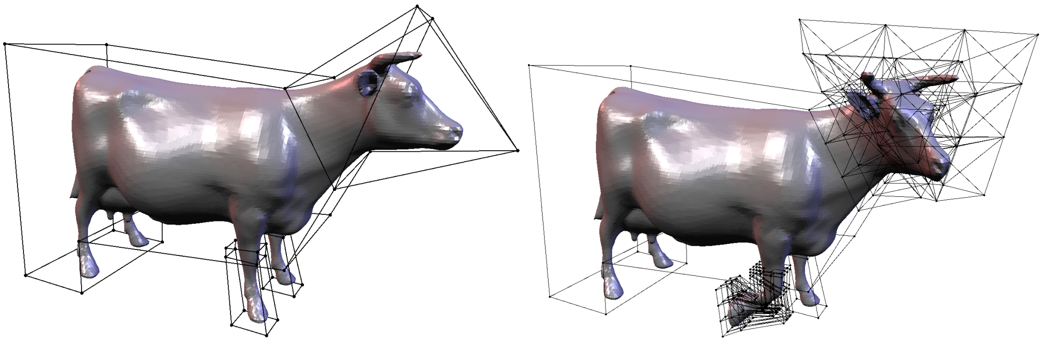


**Figure 7.4: The control net containing the cow consists of 6 polyhedra: one for the body, one for the head, two for the front knees, and two for the front legs (*left*). It demonstrates the ability of our method to handle complex control nets that are adapted to the shape of the object. We specified the deformation, which is $C^1$-continuous, by moving the vertices of the control polyhedra. The intermediate control points were computed automatically (*right*).**

or three entries of the multi-index belong to neck vertices) or as exterior control points (if two or three entries belong to exterior vertices). All neck control points are mapped identically to ensure a smooth connection to the undeformed body of the cow (Corollary 7.5), and all exterior control points are mapped by the linear function defined by the deformation of the head polyhedron to capture the total deformation and to make sure that the deformation is smooth at the three exterior vertices as well (Corollary 7.4). However, the task of providing a general and convenient method for the placement of the control points (for space deformations and for Bézier surfaces) remains a topic for future research.

## 7.4 Summary and Future Work

We developed criteria for the construction of smooth Bézier maps. A Bézier map is a map that is piecewise (on a given polytope) a homogeneous polynomial in generalized barycentric coordinates. We showed how the coefficients of the Bernstein polynomials can be chosen to enforce smoothness of any desired order across common (hyper-)faces of the polytopes. We decided to develop the theory in full generality although we mainly aim at Bézier maps in mean value coordinates. This allows the use of our results for any other barycentric coordinates that might come to the focus of attention in the future. Moreover, it shows that many results from the well developed field of simplicial and tensor product Bézier theory can be considered as a special case of our findings if Wachspress coordinates are used. Our indexing scheme, however, does not coincide with the traditional indexing scheme for tensor product Bézier maps. This sheds new light on the classical theory, which will hopefully lead to a better understanding of the tensor product Bézier maps as well.

Probably the most important examples of Bézier maps are Bézier curves and surfaces and space deformations. We presented examples of mean value Bézier surfaces and free-form deformations based on Bernstein polynomials in mean value coordinates as possible applications. Nearly without additional effort, we can ensure that our Bézier maps exhibit the desired smoothness even at the polytope vertices, although the mean value coordinates themselves are only $C^0$-continuous at these points. Thus, it is now possible to construct smooth mean value Bézier maps with arbitrary polytopes as domains.

Nevertheless, a number of open questions remain. Foremost, some kind of spline representation of Bézier maps has to be found that takes care of any continuity issues fully automatically. These splines should allow to place meaningful control points directly during the design of surfaces and deformations without the necessity to spend much time on the cumbersome process of satisfying the continuity constraints manually. Another issue that we did not discuss in the current chapter are rational Bézier maps. The use of rational Bézier maps expanded the capabilities of classical Bézier theory greatly. The same should be done for generalized Bézier maps.

# Part III

# Curvature Estimation and Barycentric Coordinates

# Chapter 8

# Asymptotic Analysis of Discrete Normals and Curvatures of Polylines

Accurate estimations of geometric properties of smooth curves and surfaces from discrete approximations are important for many computer graphics and computer vision applications. The (more complicated) case of surfaces is treated in the next chapter. In this chapter, we will focus on investigating estimations for curves.

To assess and improve the quality of such an estimation, we assume that the curve is known in general form. Then we can represent the curve by a Taylor series expansion and compare its geometric properties with the corresponding discrete approximations. In turn, we can either prove convergence of these approximations towards the true properties as the edge lengths tend to zero, or we can get hints on how to eliminate the error. In this chapter, we propose and study discrete schemes for estimating tangent and normal vectors as well as curvature and torsion of a smooth 3D curve approximated by a polyline. We thereby make some interesting findings about connections between (smooth) classical curves and certain estimation schemes for polylines.

## 8.1 Introduction

Reliable approximations of differential properties of a curve form the basis of many algorithms in computer graphics and computer vision. Curvature, for example, can be used to define the smoothness of a curve. Furthermore, the understanding of discrete normals and curvatures of curves is a precondition for the even more important—and more difficult—task of understanding discrete normals and curvatures of a surface. In this sense, this chapter also lays the foundation for reliable estimates of normals and curvatures on meshes.

The problem of estimating differential properties of discrete approximations has already been treated in the classical literature of differential geometry [Sau70]. But in that context, the speed of convergence was not an issue, and often very simple approximations were used. For example, the tangent vector at a vertex was approximated linearly by an incident edge. Today, usually a weighted average of the incident edges (or a weighted average of the edge normals, respectively) is used to approximate tangent and normal vectors at a vertex of a polyline. Various weights have been proposed for that purpose [ABS02, ABFH08]. Some popular schemes are uniform weighting, weighting by edge lengths, and weighting by inverse edge lengths. It was shown in [ABS02] that the last of these methods yields the best results for planar curves. But this result holds not necessarily for space curves. In particular, there exists no unique edge normal from which the (uniquely defined) curve normal can be computed. For the estimation of curvature and torsion, various methods have been suggested by Boutin [Bou00]. With our approach, we yield simpler formulae, which, nevertheless, exhibit at least the same accuracy.

There are basically two ways to evaluate the quality of any of these methods. On the one hand, they can be applied to a specific polyline that interpolates an analytical curve, and the result can be compared to the exact tangent vector (or any other approximated geometric property) at the corresponding point. On the other hand, an asymptotic analysis can be applied. In this case, the analytical curve is given in general form, usually represented by a Taylor series expansion. Then the outcome of the discrete approximation can again be compared to the real tangent vector. Both methods have advantages and drawbacks. The first one cannot state general results, but only for certain test curves. The second method holds for all (analytical) curves and can give clues for design and improvement of the approximations. But it is only helpful for dense polylines where dense is not well defined. It has successfully been applied for planar curves [ABS02]; for space curves, pioneering work has been done in [Bou00].

In real world applications, all these computations have often to be done in the presence of noise. In this chapter, we assume that all points lie exactly on a smooth curve since the definitions for differential properties are valid only in that case. Though we make this assumption for the development of our discrete approximation formulae, this does not mean that our work is useless for real data. The estimation error of every approximation scheme is composed of a systematic error, which is inherent in the utilized approximation scheme, and of an error introduced by noise. The goal of this chapter is to minimize the former.

The main focus of this chapter is developing a mathematical apparatus for the asymptotic analysis of arbitrary curves, and applying it to derive new, asymptotically correct estimations for tangents, normals, curvatures, and torsions of space curves. A uniform evaluation of existing approaches and our newly proposed

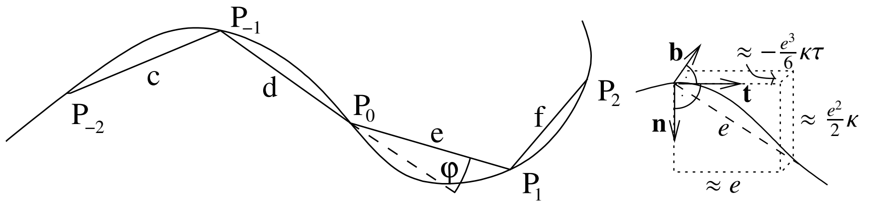


**Figure 8.1: A space curve and its Frenet frame.**

approximations is given. In particular, we prove the convergence of our approximations and can show their optimality in many cases. To estimate torsion, which is a third derivative, we need at least four points for the approximation, but we consider also estimations using five points to obtain better results. The case of planar curves [ABS02] is consistently included.

## 8.2 Approximation of Space Curves

As shown in [ABS02], the (2-dimensional) Frenet frame of a planar curve can be used to evaluate the discrete estimations of curve normals and curvatures. The same method carries over to the case of space curves using the 3-dimensional Frenet frame. We will show that the same formulae essentially still hold to compute tangent vectors and curvatures of 3D curves. Nevertheless, it is necessary to rederive these formulae for 3D curves since their asymptotic expansions are more complex. They contain, for example additional terms involving torsion. We suggest and assess several approximations for the Frenet frame, the curvature, and the torsion of a space curve.

Let a smooth curve **r** be interpolated by five points $P_{-2}$, $P_{-1}$, $P_0$, $P_1$, and $P_2$, with the corresponding edges $\overrightarrow{P_iP_{i+1}}$ denoted by **c**, **d**, **e**, and **f**, and their lengths denoted by $c$, $d$, $e$, and $f$, see Figure 8.1. Then these edges can be expressed by their Taylor expansions in the coordinate system given by the Frenet frame of **r** with tangent **t**, normal **n**, and binormal $\mathbf{b} = \mathbf{t} \times \mathbf{n}$. Let further $\kappa$ denote the curvature at $P_0$ and $\tau$ denote the torsion at the same point. The exact expansions can be found in Appendix B.

First, we consider inverse edge lengths as weights for the edges:

**8.1 Theorem (tangent vector).** *The tangent of the circle passing through $P_{-1}$, $P_0$ and $P_1$ (see Figure 8.2) is a second order approximation of the real tangent of the*

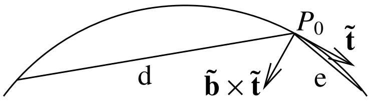


**Figure 8.2: The circle approximation.**

*curve* **r***:*

$$
\begin{aligned}
\tilde{\mathbf{t}} &:= \frac{de}{d+e}\Big(\frac{\mathbf{d}}{d^2}+\frac{\mathbf{e}}{e^2}\Big)\\
&= \mathbf{t}\Big(1-\frac{de}{8}\kappa^2+\frac{d^2e-de^2}{12}\kappa\kappa'+O(d,e)^4\Big)\\
&+\mathbf{n}\Big(\frac{de}{6}\kappa'-\frac{d^2e-de^2}{24}(\kappa''-\kappa\tau^2)+O(d,e)^4\Big) \qquad (8.1)\\
&+\mathbf{b}\Big(-\frac{de}{6}\kappa\tau+\frac{d^2e-de^2}{24}(2\kappa'\tau+\kappa\tau')+O(d,e)^4\Big).
\end{aligned}
$$

*This estimation is optimal among all three-point approximations of the tangent in the sense that the quadratic term in the normal component cannot be different from the one that shows up here. Also, this is the only linear combination of* **d** *and* **e** *that yields a second order approximation.*

*Proof.* The equation can be derived directly from the Taylor expansions in Appendix B. If there were curves with other quadratic terms, we could gain a tangent estimation and in turn an estimation of the normal for planar curves of the same accuracy, but this is not possible, see [ABS02].

The last statement of the proposition can easily be derived by using the Taylor expansions of **d** and **e** from Appendix B. □

Note that in the planar case the knowledge of the tangent and the normal is equivalent. Therefore, every tangent formula can be used to compute normals of plane curves. In 3D, however, the computation of normals is more difficult because the oscillating plane is unknown. It can be done after estimating the binormals which determine that plane. A more direct approach is to compute the curvature vector, for example by using finite differences.

**8.2 Theorem (curvature vector).** *The finite difference approach yields a linear approximation of the true curvature vector and thus of the true normal vector. If* *all edges have equal length, the convergence is quadratic.*

$$\begin{aligned}\bar{\mathbf{k}} &:= \frac{2}{d+e}\Big(\frac{\mathbf{e}}{e} - \frac{\mathbf{d}}{d}\Big)\\ &= \mathbf{t}\Big(\frac{d-e}{4}\kappa^2 - \frac{d^2 - de + e^2}{6}\kappa\kappa' + O(d,e)^3\Big),\\ &+ \mathbf{n}\Big(\kappa - \frac{d-e}{3}\kappa' + \frac{d^2 - de + e^2}{12}(\kappa'' - \kappa\tau^2) + O(d,e)^3\Big)\\ &+ \mathbf{b}\Big(\frac{d-e}{3}\kappa\tau - \frac{d^2 - de + e^2}{12}(2\kappa'\tau + \kappa\tau') + O(d,e)^3\Big).\end{aligned} \tag{8.2}$$

*Furthermore, this is the only linear combination of* **d** *and* **e**, *that yields a (at least) linear approximation of the real normal vector.*

*Proof.* Again all claims can directly be proven from the Taylor expansions given in Appendix B. □

From the curvature vector, we gain the curvature as the norm. Another possibility is to estimate the curvature by *angle approximation*. That approach is based on the definition of curvature as the rate of angular change of the tangent vector along the curve.

**8.3 Theorem (curvature).** *Let $\varphi$ be the angle between* **d** *and* **e**, *see Figure 8.1. Curvature, estimated using the discrete curvature vector (8.2) or angle approximation, respectively, converges linearly towards the true curvature. If all edges have equal length, the convergence is quadratic.*

$$\begin{aligned}\bar{\kappa} := \|\bar{\mathbf{k}}\| &= \kappa + \frac{e-d}{3}\kappa' + \frac{d^2 - de + e^2}{12}\kappa''\\ &\quad - \frac{d^2 + de + e^2}{36}\kappa\tau^2 + \frac{d^2 - 2de + e^2}{32}\kappa^3 + O(d,e)^3,\\ \hat{\kappa} := \frac{2\varphi}{d+e} &= \kappa + \frac{e-d}{3}\kappa' + \frac{d^2 - de + e^2}{12}(\kappa'' + \frac{\kappa^3}{2})\\ &\quad - \frac{d^2 + de + e^2}{36}\kappa\tau^2 + O(d,e)^3.\end{aligned}$$

*These estimations are optimal among all three-point approximations in the sense that the linear terms cannot be different from the ones that show up here.*

*Proof.* Again the equations can be derived from Appendix B and optimality can be reduced to the planar case [ABS02]. □

Yet another way to estimate the curvature is as the inverse of the radius of the circle passing through $P_{-1}$, $P_0$ and $P_1$. This has been done in [Bou00] and yields

$$\widetilde{\kappa} = \kappa + \frac{e-d}{3}\kappa' + \frac{d^2 - de + e^2}{12}\kappa'' - \frac{d^2 + de + e^2}{36}\kappa\tau^2 + O(d,e)^3.$$

Since $\sin\varphi$ equals $\varphi$ up to quadratic error, we can compute an approximation for $\kappa$ as

$$\frac{2\|\mathbf{d} \times \mathbf{e}\|}{de(d+e)} \approx \hat{\kappa}$$

without significant loss of accuracy.

Also note that for $d = e$ the expansion of the angle approximation becomes

$$\hat{\kappa} = \kappa + \frac{e^2}{12}(\kappa'' + \frac{\kappa^3}{2} - \kappa\tau^2) + O(e^4),$$

see Appendix B, and here the quadratic term vanishes for a special class of curves called elastica, characterized by minimizing the bending energy

$$\int \kappa^2 ds \longrightarrow \min$$

while fixing end points. They were introduced by Euler [Eul44] and have applications in computer graphics as well as in computer vision today [CMP04, Mum94, Hor83].

**8.4 Theorem (Euler's elastica).** *The curvature estimation $\hat{\kappa}$ converges of fourth order for elastica if all edges have equal length.*

In fact, the lower order error terms vanish for an even broader class of curves, see Appendix D for a derivation.

Binormals $\mathbf{b}_i$ at $P_i$ can be estimated by the normal of the plane defined by three consecutive points, $P_{i-1}$, $P_i$, and $P_{i+1}$, for example $\mathbf{b}_0 = \frac{\mathbf{d}\times\mathbf{e}}{\|\mathbf{d}\times\mathbf{e}\|}$. Now we apply the method of angle approximation to these binormals to compute the torsion $\hat{\tau}_\mathbf{e}$ (located at the edge $\mathbf{e}$) from the angle $\eta_\mathbf{e}$ between $\mathbf{b}_0$ and $\mathbf{b}_1$. A straight-forward approximation of $\eta_\mathbf{e}$ is $\|\mathbf{b}_1 \times \mathbf{b}_0\| = \sin\eta_\mathbf{e}$. However, computing the norm of the cross product is computationally rather expensive, and, even worse, yields always positive values whereas torsion is a signed property. Hence, we exploit the fact from the Frenet equations that $\frac{d\mathbf{b}}{ds} = \tau\mathbf{n}$. This implies that $\mathbf{b}_1 \times \mathbf{b}_0$ is approximately orthogonal to $\mathbf{n}$ and, since $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ is an orthonormal basis, approximately aligned with $\mathbf{t}$, and we define $\hat{\eta}_\mathbf{e} := \langle \mathbf{b}_1 \times \mathbf{b}_0, \tilde{\mathbf{t}}\rangle$ where $\tilde{\mathbf{t}}$ denotes the tangent approximation from equation (8.1). In fact $\hat{\eta}_\mathbf{e} = \eta_\mathbf{e} + O(d,e,f)^3$ (because $\eta_\mathbf{e}$ depends linearly on $d$, $e$ and $f$, and $\sin\eta_\mathbf{e}$ approximates $\eta_\mathbf{e}$ up to second order). We define analogously $\hat{\eta}_\mathbf{d}$ from $\mathbf{b}_{-1}$ and $\mathbf{b}_0$ and get (see Appendix B for the Taylor expansion of $\hat{\eta}_\mathbf{e}$)

**8.5 Theorem (torsion, four points).** *Using four of the five points $P_{-2}$, $P_{-1}$, $P_0$, $P_1$, and $P_2$, torsion can be approximated linearly as follows:*

$$\begin{aligned}\hat{\tau}_{\mathbf{d}} &:= \frac{3\hat{\eta}_{\mathbf{d}}}{c+d+e} = \tau - \frac{c-e}{6}\frac{\kappa'}{\kappa}\tau - \frac{c+2d-e}{4}\tau' + O(c,d,e)^2,\\ \hat{\tau}_{\mathbf{e}} &:= \frac{3\hat{\eta}_{\mathbf{e}}}{d+e+f} = \tau + \frac{f-d}{6}\frac{\kappa'}{\kappa}\tau - \frac{d-2e-f}{4}\tau' + O(d,e,f)^2.\end{aligned}$$

It is interesting to compare the above estimation $\hat{\tau}_{\mathbf{e}}$ with the results from Boutin [Bou00], who uses the same four points. Let $g$ be the distance $\|\overrightarrow{P_0P_2}\|$. Then

$$\begin{aligned}\widetilde{\tau}_1 &= \tau + \frac{d-e+3g}{6}\frac{\kappa'}{\kappa}\tau + \frac{e-d+g}{4}\tau' + O(d,e,f)^2\\ \text{and}\quad \widetilde{\tau}_2 &= \tau + \frac{d+e+g}{6}\frac{\kappa'}{\kappa}\tau + \frac{e-d+g}{4}\tau' + O(d,e,f)^2.\end{aligned}$$

Our approximation is more symmetric in the sense that the first linear error term vanishes if all edge lengths are equal. By estimating torsion using the angle between $\mathbf{b}_{-1}$ and $\mathbf{b}_1$, we can even get a (rational) expression completely without linear terms if $d = e$ and $c = f$.

$$\begin{aligned}\tilde{\tau} := \frac{3\tilde{\eta}}{c+2(d+e)+f} &= \tau - \frac{c^2+cd+d^2-e^2-ef-f^2}{6(c+2(d+e)+f)}\frac{\kappa'}{\kappa}\tau\\ &\quad - \frac{c^2+3cd+3d^2-3e^2-3ef-f^2}{4(c+2(d+e)+f)}\tau' + O(c,d,e,f)^2.\end{aligned}$$

A better way to obtain such a symmetric expression is to take the (unique) weighted average of $\hat{\tau}_{\mathbf{d}}$ and $\hat{\tau}_{\mathbf{e}}$ such that the term involving $\tau'$ vanishes completely and the term involving $\frac{\kappa'}{\kappa}\tau$ vanishes for $d = e$ and $c = f$:

$$\begin{aligned}\hat{\tau} &:= \frac{1}{c+d+e+f}((f+2e-d)\tau_{\mathbf{d}} + (c+2d-e)\tau_{\mathbf{e}})\\ &= \tau - \frac{ce-e^2+d^2-df}{3(c+d+e+f)}\frac{\kappa'}{\kappa}\tau + O(c,d,e,f)^2.\end{aligned}$$

It can be improved further by estimating $\frac{\kappa'}{\kappa}\tau$ and eliminating the corresponding error term. In that way, we can get a five-point approximation of the torsion at $P_0$ that converges quadratically for arbitrary edge lengths. For this purpose, we approximate curvatures at $P_{-1}$ and $P_1$ from the angles $\varphi_{-1}$ between $\mathbf{c}$ and $\mathbf{d}$, and $\varphi_1$ between $\mathbf{e}$ and $\mathbf{f}$:

$$\begin{aligned}\kappa_{-1} &:= \frac{2\varphi_{-1}}{c+d} = \kappa - \frac{c+2d}{3}\kappa' + O(c,d)^2\\ \text{and}\quad \kappa_1 &:= \frac{2\varphi_1}{e+f} = \kappa + \frac{f+2e}{3}\kappa' + O(e,f)^2.\end{aligned}$$

From this, we get five-point estimates for the curvature

$$\kappa_5 := \frac{1}{c + 2(d+e) + f}((2e + f)\kappa_{-1} + (c + 2d)\kappa_1) = \kappa + O(c, d, e, f)^2,$$

for its derivative (this formula was also suggested in [Bou00])

$$\kappa_5' := \frac{3}{c + 2(d+e) + f}(\kappa_1 - \kappa_{-1}) = \kappa' + O(c, d, e, f),$$

and finally

**8.6 Theorem (torsion, five points).** *Using the five points $P_{-2}$, $P_{-1}$, $P_0$, $P_1$, and $P_2$, we can obtain a second order approximation for torsion:*

$$\tau_5 := \hat{\tau} + \frac{ce - df + d^2 - e^2}{3(c + d + e + f)} \frac{\kappa_5'}{\kappa_5} \hat{\tau} = \tau + O(c, d, e, f)^2.$$

*Here, $\hat{\tau}$, $\kappa_5$, and $\kappa_5'$ are defined as above.*

## 8.3 Summary and Future Work

We have presented a mathematical framework to develop and evaluate approximation schemes to estimate differential properties of discrete curves. Its application yielded several formulae to estimate curvature, torsion, and the Frenet frame of a space curve, such that they converge towards their smooth counterparts as edge lengths tend to zero. Furthermore, we proved the optimality of our estimates in many cases. Thus, we provided a useful toolbox for the analysis of polylines in the three-dimensional space.

Possible future work is the extension of our research on the asymptotic properties of estimations of curvature and torsion of curves on surfaces. Also, the influence of noise on normal and curvature estimations has to be taken into account.

# Chapter 9

# Exact and Interpolatory Quadratures for Curvature Tensor Estimation

The computation of the curvature of smooth surfaces has a long history in differential geometry and is essential for many geometric modeling applications such as feature detection. We present a novel approach to calculating the mean curvature from arbitrary normal curvatures. Then, we demonstrate how the same method can be used to obtain new formulae to compute the Gaussian curvature and the curvature tensor. We compute the respective curvature integrals as a weighted sum. To obtain the correct weights in this quadrature, we make use of the periodic structure of the normal curvatures and apply barycentric coordinates. Finally, we derive an approximation formula for the curvature of discrete data like meshes and show its convergence if quadratically converging normals are available.

## 9.1 Introduction

The first attempts to define and determine the curvature of a surface date back at least to the eighteenth century. While a curve has only a single curvature value at each point, there are an infinite number of curvatures for every surface point: one curvature value for every tangential direction of the surface at a given point. The average of all these curvatures is defined as *mean curvature $H$*.

Euler was the first who recognized that the *principal directions* $\mathbf{t}_{\max}$ and $\mathbf{t}_{\min}$ of maximal and minimal curvature $\kappa_{\max}$ and $\kappa_{\min}$ are orthogonal to each other and that every other directional curvature (also called normal curvature) $\kappa(\phi)$ can be derived from these curvatures and the angle $\phi$ between $\mathbf{t}_{\max}$ and the curvature direction of $\kappa(\phi)$

$$\kappa(\phi) = \kappa_{\max} \cos^2 \phi + \kappa_{\min} \sin^2 \phi. \tag{9.1}$$

This relationship is now known as *Euler's formula*. In the nineteenth century, Gauß introduced a local surface quantity that received the name *Gaussian curvature K* and measures the deviation from flatness at a given point. In computer graphics, mean curvature, Gaussian curvature, and principal directions together are often referred to as the *curvature tensor* [MDSB03, Tau95a]. We will use this notion throughout this chapter but want to remark that it does not coincide with the (Riemann) curvature tensor used in differential geometry but rather with the shape operator or Weingarten map. We will present new closed form formulae for mean and Gaussian curvature and give a new formula for the curvature tensor (shape operator) matrix as well.

Nowadays, reliable curvature computation is important for many computer graphics and geometric modeling applications, and several recent articles present closed form curvature formulae [Gol05, XB03]. The mean curvature vector is related to the Laplace-Beltrami operator, and its computation is the core of many smoothing and fairing algorithms [DMSB99, Tau95b]; the principal directions can be used for "geometrically meaningful" remeshing [ACSD*03]. The curvature tensor is essential for many more tasks, like mesh segmentation, feature detection, and non-photorealistic rendering [DFRS03, GG01, HPW05, OBS04].

The field of finding and defining discrete counterparts of these smooth notions is often called Discrete Differential Geometry. Roughly, two approaches can be distinguished: either differential properties are redefined for discrete data while trying to preserve *global* characteristics or formulae from (smooth) differential geometry are discretized with the focus on preserving *local* (pointwise) properties. We elaborate further on these approaches in Section 9.1.2.

Our contribution belongs to classical and discrete differential geometry. We derive a novel curvature formula and use it to propose a new curvature approximation for discrete data. This is explained in more detail in the following section.

### 9.1.1 Main contributions and discussion

A quadrature of an integral is an approximation of this integral as a weighted sum, $\int f(x)\,\mathrm{d}x \approx \sum_i \omega_i f(x_i)$. In this chapter, we are particularly interested in integral representations of surface curvatures. We consider the mean curvature

$$H = \frac{1}{2\pi} \int_0^{2\pi} \kappa(\phi)\,\mathrm{d}\phi, \tag{9.2}$$

the Gaussian curvature (the integral representation of which appeared before in [MT98])

$$K = 3H^2 - \frac{1}{\pi}\int_0^{2\pi} \kappa(\phi)^2 \,\mathrm{d}\phi, \tag{9.3}$$

and the Taubin integral representation of the curvature tensor ("Taubin tensor")

$$\mathbf{M} = \frac{1}{2\pi}\int_0^{2\pi} \kappa(\phi)\mathbf{t}(\phi)\mathbf{t}(\phi)^t \,\mathrm{d}\phi. \tag{9.4}$$

It is defined as the integral of the curvature weighted covariance matrix of the tangent vector $\mathbf{t}(\phi)$ over the angle $\phi$ between $\mathbf{t}(\phi)$ and $\mathbf{t}_{\max}$, and it has the principal directions as eigenvectors with the eigenvalues equal to linear combinations of the principal curvatures [Tau95a].

We will derive novel weights that enable for the first time to obtain exact quadratures of these (and further) curvature integrals by making use of their periodic structure. Only minor restrictions regarding the number and positions of the samples $\phi_i$ will be made.

Let $P$ be a point of a smooth surface, let $\kappa_i := \kappa(\phi_i)$ be the normal curvatures at $P$ in the tangential directions $\mathbf{t}_i$, and let $\beta_i := \phi_{i+1} - \phi_i$ be the angle between $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$ (compare Figure 9.1 on the left hand side). First, we will present a formula to compute the mean curvature $H$ at the surface point $P$ by a quadrature of integral (9.2). It is a generalization of the not-so-well-known formula

$$H = \frac{1}{n}\sum_i \kappa_i$$

that holds if all angles $\beta_i$ are equal. This, again, is a generalization of

$$H = \frac{1}{2}(\kappa_{\max} + \kappa_{\min}).$$

Our formula will be proven in Theorem 9.2; it is already stated here to give an outlook on our results.

*Mean curvature is given by the weighted sum of normal curvatures:*

$$H = \sum_i \omega_i \kappa_i, \qquad \omega_i := \frac{\tan\beta_{i-1} + \tan\beta_i}{\sum_j(\tan\beta_{j-1} + \tan\beta_j)}. \tag{9.5}$$

*Similar results are obtained for the Gaussian curvature and the Taubin tensor* **M**.

The $\omega_i$ can be seen as a kind of barycentric coordinate: they determine $H$ as a linear combination of normal curvatures $\kappa_i$ within the line segment $[\kappa_{\min}, \kappa_{\max}]$. If necessary (for example, because all the $\kappa_i$ are smaller than $H$, that is, they belong

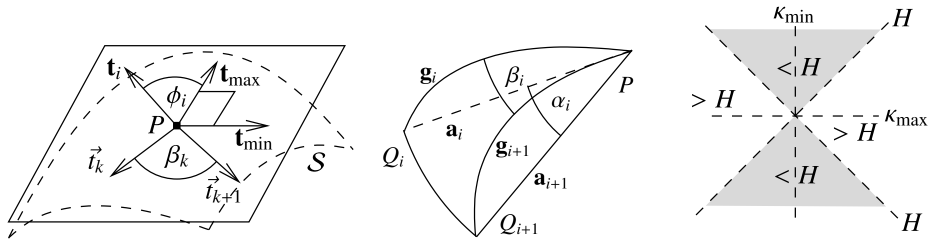


**Figure 9.1: Left and Middle: Angles in the tangent plane and a mesh. Right: Change of the normal curvature around a point.**

to the grey area in Figure 9.1 (right)), then some of the weights $\omega_i$ become negative automatically.

In practice, the quadrature formula (9.5) leads to the following discrete approximation of $H$ (given for a mesh to simplify the notation):

$$H \approx \sum_i \frac{\tan \alpha_{i-1} + \tan \alpha_i}{\sum_j (\tan \alpha_{j-1} + \tan \alpha_j)} k_i \qquad (9.6)$$

where $\alpha_i$ is the angle between two consecutive outward edges $\mathbf{a}_i$ in the one-ring around $P$ (see Figure 9.4 (left) and Figure 9.1 (middle) for the relationship of $\alpha_i$ and $\beta_i$) and $k_i := 2\langle \frac{\mathbf{a}_i}{a_i^2}, \mathbf{n} \rangle$ is an approximation of the normal curvature $\kappa_i$ using the edges $\mathbf{a}_i$ of length $a_i$ and the normal $\mathbf{n}$ at $P$. We will discuss the influence of normal approximations for $\mathbf{n}$ in Section 9.3.

Our quadrature can be used to calculate integrals over all “Euler-like” formulae

$$M(\phi) = M_1 \cos^2 \phi + M_2 \sin^2 \phi.$$

Such formulae occur repeatedly in the literature to describe surface related quantities. Blaschke [Bla56, §24.III] showed that the curvature radius of the osculating cylindrical surface along a given tangent vector is described this way. (This statement can be considered as dual to Euler’s formula (9.1).) Pottmann et al. [PHYK05, 7.1] derived this relationship for curvature related moments at a surface point. An especially important example is the Laplacian of a function $f$ defined on the surface. After expressing it as the integral $\int \frac{\partial^2 f}{\partial \mathbf{t}(\phi)^2} \, \mathrm{d}\phi$, we get the following quadrature identity:

$$\Delta f = 2 \sum_i \omega_i \frac{\partial^2 f}{\partial \mathbf{t}_i^2}.$$

The same technique extends also to other integrals with “trigonometric” integrands. We applied it to the integrals (9.3) and (9.4) to obtain exact formulae and, in turn, discrete approximations of the Gaussian curvature and of the curvature tensor.

**9.1 Remark.** Now, the question arises whether the previous results generalize to higher dimensions. An analog of the Euler formula (9.1) for the directional curvature in direction $\mathbf{t}$ for three-dimensional manifolds is given by

$$\kappa(\mathbf{t}) = \kappa_{\max} \cos^2 \phi_{\max} + \kappa_{\mathrm{med}} \cos^2 \phi_{\mathrm{med}} + \kappa_{\min} \cos^2 \phi_{\min}$$

where $\phi_{\max}$, $\phi_{\mathrm{med}}$, and $\phi_{\min}$ are the angles between $\mathbf{t}$ and the respective principal directions $\mathbf{t}_{\max}$, $\mathbf{t}_{\mathrm{med}}$, and $\mathbf{t}_{\min}$ with principal curvatures $\kappa_{\max} \geq \kappa_{\mathrm{med}} \geq \kappa_{\min}$. Corresponding formulae hold for n-dimensional manifolds. The integral formula for the mean curvature follows easily:

$$H = \frac{\kappa_{\max} + \kappa_{\mathrm{med}} + \kappa_{\min}}{3} = \frac{1}{4\pi} \int_{\mathbb{S}^2} \kappa(\mathbf{t})\, \mathrm{d}S.$$

Similar integral representations can be obtained for the other coefficients of the characteristic polynomial of the shape operator. Nevertheless, we were not able to derive a discrete quadrature formula similar to (9.5) to calculate the above integral. This topic needs still further research.

Let us now summarize our contributions:

- we derive quadrature identities for trigonometric integrals; in particular, we present exact formulae for the mean curvature, the Gaussian curvature, the curvature tensor, and the Laplacian (Section 9.2),

- we show how these quadratures can be used to obtain fast and reliable approximations of the curvature of a smooth surface interpolated by discrete data like a triangle mesh (Section 9.2.6),

- extending the work of Chapter 8 from curves to surfaces, we present a framework designed to enable an easy and meaningful representation of the Taylor series expansion of an analytic surface interpolated by discrete data (Appendix C),

- we use this framework to analyze the convergence speed of common estimations for normal vectors and our curvature approximations, and we suggest a new normal estimation based on our analysis (Section 9.3),

- and we prove the convergence of our curvature approximations for arbitrary aspect ratios if the normal vector is known with quadratic accuracy and confirm the result experimentally (Section 9.4).

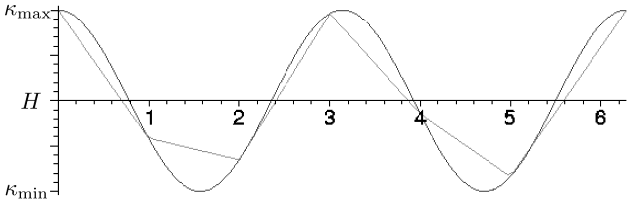


**Figure 9.2: The curvature function whose integral (9.2) defines the mean curvature. The piecewise linear function used by the trapezoidal rule shows considerable deviation.**

## 9.1.2 Related work

### Quadratures

A classical quadrature formula is given by the trapezoidal rule that integrates the piecewise linear function that is given by connecting sample points by straight lines as shown in Figure 9.2. It was used for curvature estimation in [WB01] but yields a relatively large error for highly periodic functions with only a few samples which is the case for our curvature integrals. More sophisticated methods like the Newton-Cotes formulae, Gaussian quadrature, or Romberg integration have higher precision and are even exact on certain function classes, but they require the evaluation of the given function at a high number of prescribed sample positions [PVTF02]. Our quadrature formula is the first one that is exact for curvature integrals, given an arbitrary number ($\geq$ 3) of (nearly) arbitrary samples. Consequently, the derived formulae to compute surface curvature are new as well.

### Curvature estimation

A great number of papers on normal and curvature approximation on polygonal meshes exist. Various types of convergence theorems are proven therein. Meek and Walton [MW00] showed the pointwise convergence of triangle normals towards the real normals provided that the shape of the triangles doesn't change too much. Morvan and Thibert [MT04] proved that it is sufficient for convergence of triangle mesh normals that no triangle angle approaches $\pi$. This condition had already been imposed earlier in the context of finite elements [BA76].

In the field of curvature estimation, "local" algorithms try to recover the curvature of a smooth surface, which is interpolated by a mesh at the vertices. This is done either by interpolating a small environment by a polynomial and computing its curvature or by discretizing a more or less special curvature definition for smooth surfaces. Cazals and Pouget [CP03] showed that polynomial fitting leads to asymptotically correct results. But this method needs at least five neighboring points and is comparatively time consuming since a (small) system of equations has to be solved. The former problem was solved by Goldfeather and Interrante [GI04] by taking the normals at each point into account. This came at the cost of losing the proven convergence, especially if the normals are not exactly known. Another approach is taken by Pottmann et al. [PWHY08] where differential quantities are extracted from integral invariants.

The cotangent formula [MDSB03] is probably the most widely used discrete approximation for the Laplace-Beltrami operator (and the mean curvature vector). Since the formula arises naturally when finite elements are used to approximate the Dirichlet energy [Duf59, PP93], it was rediscovered several times. Zayer [Zay07] gives a historical survey of the formula. Given convergent normals, a weak convergence of the cotangent formula in an appropriate Sobolev space was proven [Dzi88, HPW06]. This weak convergence justifies the successful use of the cotangent formula in finite element approximations of the Laplace-Beltrami operator theoretically. However, the cotangent formula, like many other common definitions of the discrete mean curvature vector, converge pointwise to the Laplace-Beltrami operator only if the mesh satisfies certain regularity conditions at the vertices as stated in Theorem 9.10 and shown in [Xu04]. If the cotangent weights are used in combination with the classical angle deficit formula, convergence is only given for special cases as well, see also [Xu06a].

This motivates the search for further curvature approximations that converge in norm. These "global" algorithms focus on preserving global characteristics like the Gauß-Bonnet theorem and are usually rather stable but not necessarily pointwise convergent for denser meshes with smaller edge lengths. The prime example is the work of Cohen-Steiner and Morvan [CSM03] who gave discrete definitions for the mean curvature, the Gaussian curvature, and the curvature tensor. If they are integrated over a (small) area and the interpolating points are connected by a restricted Delaunay triangulation, the resulting values converge towards the integrals over the respective smooth curvatures. A similar definition was suggested by Hildebrandt and Polthier [HP04].

A third type of approximations of the Laplace-Beltrami operator and therefore of the mean curvature vector is used in the field of machine learning, see [HAvL07] and references therein. The so called graph Laplacian at a certain point is usually influenced by all point samples of the surface while the before mentioned methods use only samples from a small neighborhood (the one-ring of a vertex in the case of meshes). Convergence results using graph Laplacians usually consider convergence in probability. A special case is the mesh Laplacian, which is actually a family $L^h$ of Laplacian operators such that $L^h\mathcal{M}_h$ converges pointwise towards $\Delta\mathcal{S}$ if $\mathcal{M}_h$ is a family of meshes, which converges towards a given surface $\mathcal{S}$ [BSW08]. A comparison of several discrete Laplace operators and their properties is given in [WMKG07].

Opposed to most of the other methods, we require stronger (quadratic) convergence of the normals to derive a stronger result: our mean curvature (function) estimate is pointwise convergent. The same is true for our approximations of the Gaussian curvature and the curvature tensor. A similar result was proven by Meek and Walton [MW00] for the spherical image method to compute the Gaussian curvature. But the spherical image method is restricted to the computation of the Gaussian curvature while our quadrature method is much more general.

There have been earlier approaches to approximate the integrals (9.2), (9.3), and (9.4) [HS03, Tau95a, WB01]. But since they lacked exact weights for the discretization of these integrals by a sum, they introduced systematic errors at irregular vertices. We believe that our quadrature formulae can initiate renewed interest in the classical approach by Taubin, especially as the progress in acquisition techniques makes it possible to obtain better and better normals directly from the scanning process [NRDR05]. Many more approaches for curvature computation exist. While numerical experiments demonstrate sometimes good results, a comprehensive theoretical analysis of convergence properties remains usually an open issue [MS92, Mar98, Rus04, TRZS04, GGRZ06].

Nevertheless, when applied to spheres, considerably better results can be achieved: Max [Max99] derived weights for the triangle normals that yield exact vertex normals under arbitrary conditions instead of linear convergence. Xu [Xu06b] suggested a modification of the mean curvature vector by Meyer et al. [MDSB03] that converges on a sphere for each mesh configuration. The angle deficit formula in the version of Meyer et al. [MDSB03] (modified in the same way as the mean curvature vector) is also convergent on spheres as has been shown in [MD02].

## 9.2 Exact Quadratures for Curvature Integrals

In this section, we derive the exact quadrature formula for the mean curvature integral (9.2) after introducing the necessary notation. Then we show how this technique can be used to obtain exact formulae for the Gaussian curvature and the curvature tensor. Finally, we deduce approximations for curvature computation on discrete data.

### 9.2.1 Notation

Given a point $P$ on a smooth surface $\mathcal{S}$, consider its tangent plane that is spanned by the two unit vectors in the principal directions $\mathbf{t}_{\max}$ and $\mathbf{t}_{\min}$, see Figure 9.1 on the left hand side. The curvatures in these directions are given by $\kappa_{\max}$ and $\kappa_{\min}$, respectively. Assume now that $\mathbf{t}_i$, $i = 1 \ldots n$, are further unit tangent vectors. Let $\phi_i$ be the angle between $\mathbf{t}_{\max}$ and $\mathbf{t}_i$ measured counter-clockwise and let $\beta_i :=$

$\phi_{i+1} - \phi_i$ (indices modulo $n$) be the angle between $\mathbf{t}_{i+1}$ and $\mathbf{t}_i$. Let $\kappa_i$ be the normal curvature of $\mathcal{S}$ at $P$ in direction $\mathbf{t}_i$. More general, let $\mathbf{t}(\phi)$ be the unit tangent vector with angle $\phi$ to $\mathbf{t}_{\max}$ and $\kappa(\phi)$ the normal curvature in that direction. Now we will show how the surface curvature can be computed only from the knowledge of the $\kappa_i$ and $\beta_i$. Discrete estimations for $\kappa_i$ and $\beta_i$ are given in Section 9.2.6.

## 9.2.2 Mean curvature

First, we present a quadrature formula $\sum_j \omega_j \kappa_j$ for the mean curvature integral $H = \frac{1}{2\pi} \int_0^{2\pi} \kappa(\phi)\, \mathrm{d}\phi$ with certain weights $\omega_j$ that add up to one.

**9.2 Theorem (mean curvature).** *The mean curvature is given by the weighted sum of normal curvatures*

$$H = \sum_j \omega_j \kappa_j, \quad \omega_j := \frac{\tan\beta_{j-1} + \tan\beta_j}{\sum_k (\tan\beta_{k-1} + \tan\beta_k)} \tag{9.7}$$

*if the denominator of the $\omega_j$ is non-zero.*

The fact that our formula depends on the denominator being non-zero is a sign of its correctness. In this way, configurations with insufficient data for the curvature computation are detected. See Section 9.3.4 for a discussion.

*Proof.* First, let the $\omega_j$ be arbitrary weights with $\sum_j \omega_j = 1$. Euler's formula (9.1) yields

$$\begin{aligned}\sum_j \omega_j \kappa_j &= \kappa_{\max} \sum_j \omega_j \cos^2 \phi_j + \kappa_{\min} \sum_j \omega_j \sin^2 \phi_j \\ &= \kappa_{\max} \sum_j \omega_j \frac{1 + \cos 2\phi_j}{2} + \kappa_{\min} \sum_j \omega_j \frac{1 - \cos 2\phi_j}{2} \\ &= \frac{\kappa_{\max} + \kappa_{\min}}{2} + \frac{\kappa_{\max} - \kappa_{\min}}{2} \operatorname{Re}\Big(\sum_j \omega_j \mathrm{e}^{2\mathrm{i}\phi_j}\Big).\end{aligned}$$

The term on the left is equal to the mean curvature; therefore, it is sufficient to find weights such that $\sum_j \omega_j \mathrm{e}^{2\mathrm{i}\phi_j} = 0$.

In Figure 9.3, the situation is sketched. On the left hand side, the unit vectors $\mathrm{e}^{2\mathrm{i}\phi_j}$ are shown. After a rotation of these vectors by an angle of $\frac{\pi}{2}$ (shown on the right hand side), we obtain immediately that $\omega_j := \frac{\tan\beta_{j-1} + \tan\beta_j}{\sum_k (\tan\beta_{k-1} + \tan\beta_k)}$ is a possible solution. (This is a special case of a theorem of Minkowski.)

This can also be seen as a problem of finding generalized barycentric coordinates for the origin of the complex plane, given the vertices $\mathrm{e}^{2\mathrm{i}\phi_j}$. Mean value

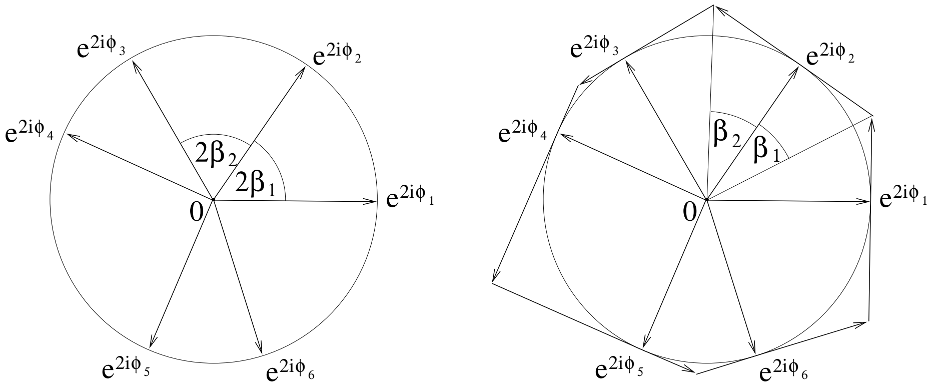


**Figure 9.3: Computation of weights for the mean curvature quadrature.**

coordinates [Flo03], Wachspress coordinates [Wac75, MLBD02], Sibson's natural neighbors [Sib80], and discrete harmonic coordinates (Section 9.3.2) all yield the same weights as above in our situation. □

Note, that $H$ does not necessarily lie in the convex hull of the $\kappa_j$ (that is the minimal interval $I \subset \mathbb{R}$ containing all the $\kappa_j$). In these cases, some of the weights $\omega_j$ automatically get negative values. This is necessary to represent $H$ as a linear combination of the $\kappa_j$. In Theorem 9.10, we indicate an example where negative weights are inevitable.

It is well known that the mean curvature vector can be obtained as (half of) the Laplacian applied to the position function $\mathbf{x}$ that parameterizes the surface, $H\mathbf{n} = \frac{1}{2}\Delta\mathbf{x}$. And in fact, our proof can also be applied to yield a new formula to calculate the two-dimensional Laplacian.

**9.3 Corollary.** *Let $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ be a smooth function. Let $f_{jj} = \dfrac{\partial^2 f}{\partial \mathbf{t}_j{}^2}$ be the second derivative of $f$ in direction $\mathbf{t}_j$. Then the Laplacian of $f$ can be computed with the weights given in (9.7)*

$$\Delta f = 2 \sum_j \omega_j f_{jj}$$

*if the denominator of the $\omega_j$ is non-zero.*

It may seem strange that we used coordinates in the 2-dimensional plane to prove the correctness of weights in a 1-dimensional sum ($\sum_j \omega_j \cos 2\phi_j \overset{!}{=} 0$). However, we do not know the principal directions. Therefore, the $\phi_j$ are only known up to a constant, and a second dimension is implicitly added. We will now give a proof of Corollary 9.3 that makes this relationship more explicit.

*Proof.* We consider again arbitrary weights $\omega_j$ with $\sum_j \omega_j = 1$.

$$\begin{aligned}\sum_j \omega_j f_{jj} &= f_{xx}\sum_j \omega_j \cos^2\phi_j + f_{yy}\sum_j \omega_j \sin^2\phi_j + 2f_{xy}\sum_j \omega_j \sin\phi_j \cos\phi_j \\ &= f_{xx}\sum_j \omega_j \frac{1+\cos 2\phi_j}{2} + f_{yy}\sum_j \omega_j \frac{1-\cos 2\phi_j}{2} + 2f_{xy}\sum_j \omega_j \frac{\sin 2\phi_j}{2} \\ &= \frac{1}{2}\Delta f + \frac{f_{xx}-f_{yy}}{2}\operatorname{Re}\Big(\sum_j \omega_j \mathrm{e}^{2\mathrm{i}\phi_j}\Big) + f_{xy}\operatorname{Im}\Big(\sum \omega_j \mathrm{e}^{2\mathrm{i}\phi_j}\Big).\end{aligned}$$

Again, the weights in (9.7), yield $\sum_j \omega_j \mathrm{e}^{2\mathrm{i}\phi_j} = 0$ as desired. □

### 9.2.3 Gaussian curvature

Next, we want to compute the integral $G := \frac{1}{2\pi}\int_0^{2\pi} \kappa(\phi)^2 \,\mathrm{d}\phi = \frac{3}{2}H^2 - \frac{1}{2}K$ to obtain the Gaussian curvature $K = 3H^2 - 2G$ as defined in (9.3). Again with Euler's formula (9.1), we obtain an approximation of this integral:

$$\begin{aligned}\widetilde{G} := \sum_j \widetilde{\omega}_j\kappa_j^2 &= \frac{3}{2}H^2 - \frac{1}{2}K \\ &+ \sum_j \widetilde{\omega}_j\Big(\frac{4\cos 2\phi_j + \cos 4\phi_j}{8}\kappa_{\max}^2 - \frac{4\cos 2\phi_j - \cos 4\phi_j}{8}\kappa_{\min}^2 - \frac{\cos 4\phi_j}{4}K\Big).\end{aligned}$$

Now the weights

$$\widetilde{\omega}_j := \frac{\tan 2\beta_{j-1} + \tan 2\beta_j}{\sum_k(\tan 2\beta_{k-1} + \tan 2\beta_k)} \tag{9.8}$$

yield $\sum_j \widetilde{\omega}_j \mathrm{e}^{4\mathrm{i}\phi_j} = 0$ and therefore $\sum_j \widetilde{\omega}_j \sin 4\phi_j = \sum_j \widetilde{\omega}_j \cos 4\phi_j = 0$. With the notation $\Sigma_{\cos} := \frac{1}{2}\sum_j \widetilde{\omega}_j \cos 2\phi_j$, we get

$$\widetilde{G} = \frac{3}{2}H^2 - \frac{1}{2}K + 2H(\kappa_{\max} - \kappa_{\min})\Sigma_{\cos}.$$

If we use the weights $\widetilde{\omega}_j$ for the estimation of $H$ we obtain the same error term:

$$\widetilde{H} := \sum_j \widetilde{\omega}_j\kappa_j = H + (\kappa_{\max} - \kappa_{\min})\Sigma_{\cos}.$$

Therefore, we can compute the exact value of $K$:

**9.4 Theorem (Gaussian curvature).** *The Gaussian curvature can be computed from the mean curvature H and the corrected weighted sum of squares of normal curvatures with the weights defined in (9.8)*

$$K = 3H^2 - 2\widetilde{G} + 4H(\widetilde{H} - H) = 3H^2 - 2\sum_j \widetilde{\omega}_j\kappa_j^2 + 4H(\sum_j \widetilde{\omega}_j\kappa_j - H)$$

*if the denominator of the $\widetilde{\omega}_j$ is non-zero.*

### 9.2.4 The Taubin tensor

Now, we show how the matrix **M**, defined by integral (9.4), which was introduced by Taubin [Tau95a] to compute the curvature tensor, can be evaluated exactly. Afterwards, we will give an alternative method to compute the principal curvatures and directions.

Let $\mathbf{T} := (\mathbf{n}, \mathbf{t}_{\max}, \mathbf{t}_{\min})$ be the orthogonal matrix with the columns given by the unit normal and the unit tangent vectors of the principal directions at the surface point $P$. In [Tau95a], it was shown that $\mathbf{M} = \mathbf{T}\left(\begin{smallmatrix} 0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{smallmatrix}\right)\mathbf{T}^t$ and that the eigenvalues $\lambda_1$ and $\lambda_2$ are related to the principal curvatures by $\kappa_{\max} = 3\lambda_1 - \lambda_2$ and $\kappa_{\min} = 3\lambda_2 - \lambda_1$. To compute a discrete approximation of integral (9.4), we use the weights (9.8) together with Euler's formula (9.1) and obtain

$$\widetilde{\mathbf{M}} := \sum_j \widetilde{\omega}_j \kappa_j \mathbf{t}_j \mathbf{t}_j^{\,t} = \mathbf{T} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda_1 + \kappa_{\max}\Sigma_{\cos} & H\Sigma_{\sin} \\ 0 & H\Sigma_{\sin} & \lambda_2 - \kappa_{\min}\Sigma_{\cos} \end{pmatrix} \mathbf{T}^t$$

with $\Sigma_{\cos} := \frac{1}{2}\sum_j \widetilde{\omega}_j \cos 2\phi_j$ and $\Sigma_{\sin} := \frac{1}{2}\sum_j \widetilde{\omega}_j \sin 2\phi_j$. Again, we want to compute the error terms and subtract them afterwards, as we did when computing the Gaussian curvature. However, the task is trickier this time since the error terms are given in a coordinate system yet unknown.

But let us define the error matrix

$$\widetilde{\mathbf{E}} := \sum_j \widetilde{\omega}_j \mathbf{t}_j \mathbf{t}_j^{\,t} = \mathbf{T} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2} + \Sigma_{\cos} & \Sigma_{\sin} \\ 0 & \Sigma_{\sin} & \frac{1}{2} - \Sigma_{\cos} \end{pmatrix} \mathbf{T}^t.$$

After restriction of the matrices to the tangent plane by a Householder transformation **Q** as done in [Tau95a], we can eliminate most of the error terms and control the remaining ones by computing:

$$\begin{aligned} &\mathbf{Q}\widetilde{\mathbf{M}}\mathbf{Q}^t - H(\mathbf{Q}\widetilde{\mathbf{E}}\mathbf{Q}^t - \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}) \\ &\qquad = \mathbf{QT} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda_1 + \frac{\kappa_{\max}-\kappa_{\min}}{2}\Sigma_{\cos} & 0 \\ 0 & 0 & \lambda_2 + \frac{\kappa_{\max}-\kappa_{\min}}{2}\Sigma_{\cos} \end{pmatrix} (\mathbf{QT})^t. \end{aligned} \tag{9.9}$$

Now all the desired information can be extracted:

**9.5 Theorem (curvature tensor).** *The matrix defined in Equation (9.9) has the surface normal and the principal directions as eigenvectors. The eigenvalues* $\mu_1$

*and $\mu_2$ that correspond to the latter two eigenvectors are related to the principal curvatures by*

$$\kappa_{\max} = 3\mu_1 - \mu_2 - (\widetilde{H} - H), \quad \kappa_{\min} = 3\mu_2 - \mu_1 - (\widetilde{H} - H).$$

### 9.2.5 Direct computation of principal curvatures and directions

Once we know mean and Gaussian curvature, there is also a more direct approach to obtain principal curvatures and directions. First, we can compute the principal curvatures by $\kappa_{\max} = H + \sqrt{H^2 - K}$ and $\kappa_{\min} = H - \sqrt{H^2 - K}$. Now we use Euler's formula (9.1) in the form $\kappa_i = (\kappa_{\max} - \kappa_{\min}) \cos^2 \phi_i + \kappa_{\min}$. When solving for $\phi_i$ we get two solutions for every $i$ (modulo $\pi$). Since the correct solutions indicate the same principal direction $\mathbf{t}_{\max}$ for all $i$, they can easily be determined. The principal directions are then uniquely given by the $\phi_i$.

### 9.2.6 Practical curvature tensor estimation

In this section, we describe how the above quadratures can be used to obtain curvature approximations for discrete data. To simplify the notation (in particular the definition of a neighborhood and a normal), we specify our estimates for (triangular) meshes only. But note that the formulae are also valid for point clouds since we rely only on the relative position of points and not on the connectivity that is defined by edges and faces of a mesh. We will prove the convergence of these formulae in Section 9.3.3.

Let $P$ be a vertex of a mesh whose one-ring is given by vertices $Q_i$ such that $P$ and the $Q_i$ interpolate the surface $\mathcal{S}$. Let $\mathbf{n}$ be the (possibly estimated) normal at $P$ and let $\mathbf{a}_i$ be the edge connecting $P$ and $Q_i$ with length $a_i := \|\mathbf{a}_i\|$. Furthermore, let $\alpha_i$ be the angle between $\mathbf{a}_i$ and $\mathbf{a}_{i+1}$, see Figure 9.4 (left). The middle part of Figure 9.1 indicates the relationship of $\alpha_i$ and $\beta_i$.

Using this notation, we can define the following approximations for the normal curvature $\kappa_i$ and the weights $\omega_i$ and $\widetilde{\omega}_i$:

$$k_i := 2\frac{\langle \mathbf{a}_i, \mathbf{n} \rangle}{a_i^2}, \; w_i := \frac{\tan \alpha_{i-1} + \tan \alpha_i}{\sum_j (\tan \alpha_{j-1} + \tan \alpha_j)}, \text{ and } \widetilde{w}_i := \frac{\tan 2\alpha_{i-1} + \tan 2\alpha_i}{\sum_j (\tan 2\alpha_{j-1} + \tan 2\alpha_j)}. \tag{9.10}$$

Thus, we get the curvature estimates

$$H \approx \sum_i w_i k_i \quad \text{and} \quad K \approx 3H^2 - 2\sum_i \widetilde{w}_i k_i^2 + 4H(\sum_i \widetilde{w}_i k_i - H). \tag{9.11}$$

Using the normalized projection of $\mathbf{a}_i$ to the tangent plane as an estimation for the tangent vector

$$\mathbf{t}_i \approx \frac{\mathbf{a}_i - \langle \mathbf{a}_i, \mathbf{n}\rangle \mathbf{n}}{\|\mathbf{a}_i - \langle \mathbf{a}_i, \mathbf{n}\rangle \mathbf{n}\|}$$

we can calculate $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{E}}$ in the (discrete) Taubin integral representation (9.9) of the curvature tensor. Means to increase the numerical stability are discussed in Section 9.3.4.

The approach of Section 9.2.5 can be used with approximated values for $H$ and $K$ as well. In this case, we have to check whether $H^2 - K \geq 0$. If this is not the case, we assume $\kappa_{\min} = \kappa_{\max} = H$. Furthermore, the principal directions, obtained from the correct solution for the $\phi_i$, do not coincide exactly. Therefore, we look for an "accumulation point" of the possible principal directions and choose this direction as $\mathbf{t}_{\max}$.

## 9.3 Asymptotic Analysis

Our approach to assess discrete approximation schemes for normal vectors and curvatures is to compare the normal vector (curvature) of an arbitrary analytical surface $\mathcal{S}$ with the estimated normal (curvature) at the same point. For this purpose we express the discrete normal (curvature) with respect to the Darboux frame, that is the coordinate system consisting of the exact normal vector $\mathbf{n}$, a tangential vector $\mathbf{t}$, and $\mathbf{v} = \mathbf{n} \times \mathbf{t}$. Then, good approximations for the normal and the curvature would look like this:

$$\mathbf{n}_{\text{estimated}} = (1 + \epsilon)\mathbf{n} + \epsilon'\mathbf{t} + \epsilon''\mathbf{v} \quad \text{and} \quad \kappa_{\text{estimated}} = \kappa + \epsilon'''$$

where each $\epsilon$ is supposed to be small and should tend to zero if a denser mesh is regarded. In contrast to other methods, we express everything in terms of mesh and surface properties without depending on a specific parameterization domain.

Consider again a mesh interpolating $\mathcal{S}$ and a mesh vertex $P$. Let the one-ring be given by vertices $Q_i$, edges $\mathbf{a}_i$ with length $a_i$ and included angles $\alpha_i$ as indicated in Figure 9.4. For each $Q_i$, we use the Darboux frame $(\mathbf{t}_i, \mathbf{v}_i, \mathbf{n})$ where $\mathbf{t}_i$ is the tangential vector in the direction of $\mathbf{a}_i$. Finally, we denote the normal curvature and geodesic torsion along $\mathbf{t}_i$, with $\kappa_i$ and $\tau_i$, respectively (see Appendix C for details).

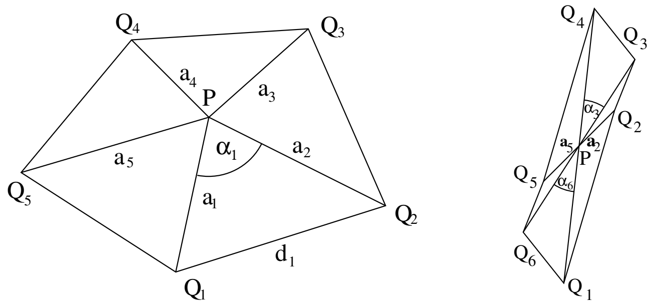


**Figure 9.4: Left: One-ring of a vertex. Right: Example of a regular vertex.**

### 9.3.1 Averaging of face normals

Using these notations, the normal of an incident triangle at $P$ is given as

$$\begin{aligned}\frac{\mathbf{a}_i \times \mathbf{a}_{i+1}}{\|\mathbf{a}_i \times \mathbf{a}_{i+1}\|} &= \mathbf{n}\Big(1 + O(a_i, a_{i+1})^2\Big) \\ &- \mathbf{v}_i\Big(\frac{a_{i+1}}{2\sin\alpha_i}\kappa_{i+1} + O(a_i, a_{i+1})^2\Big) + \mathbf{t}_i\Big(O(a_i, a_{i+1})^2\Big) \\ &+ \mathbf{v}_{i+1}\Big(\frac{a_i}{2\sin\alpha_i}\kappa_i + O(a_i, a_{i+1})^2\Big) + \mathbf{t}_{i+1}\Big(O(a_i, a_{i+1})^2\Big), \qquad (9.12)\end{aligned}$$

see Appendix C. This shows that the triangle normal, and therefore every normal computed as a weighted average of the triangle normals in the one-ring of $P$, converges linearly to the real normal.

Since the coordinate systems $(\mathbf{t}_i, \mathbf{v}_i, \mathbf{n})$ chosen by us are adapted to the mesh structure, it is now easy to derive further convergence results. First, we prove quadratic convergence for regular vertices. These shall be vertices of even valence, where opposing edges have the same length, and opposing angles are equal, see Figure 9.4 (right). More rigorously, we define

**9.6 Definition (regular vertex).** Let $P$ be a mesh vertex of valence $n = 2m$ with incident edges $\mathbf{a}_i$ of length $a_i$. Let $\alpha_i$ be the angle between $\mathbf{a}_i$ and $\mathbf{a}_{i+1}$. Then $P$ is called *regular* iff

$$a_i = a_{i+m} \quad \text{and} \quad \alpha_i = \alpha_{i+m}$$

for all $i = 1 \ldots m$.

For regular vertices we can conclude

$$\mathbf{v}_{i+m} = -\mathbf{v}_i + O(a_j)_j^2 \quad \text{and} \quad \kappa_{i+m} = \kappa_i + O(a_j)_j^2,$$

see Appendix C. Therefore, the linear terms cancel out when summing up all facet normals of the faces incident at $P$. Summarizing, we generalize a result by Meek and Walton [MW00]:

**9.7 Theorem (mesh normals).** *Let $P$ be a mesh vertex with incident edges $\mathbf{a}_i$ of length $a_i$ connecting $P$ with $Q_i$. Let $\alpha_i$ be the angle between $\mathbf{a}_i$ and $\mathbf{a}_{i+1}$ and let $\mathbf{n}_i := \frac{\mathbf{a}_i \times \mathbf{a}_{i+1}}{\|\mathbf{a}_i \times \mathbf{a}_{i+1}\|}$. Let $w_i$ be weights depending only on edge lengths and included angles. Then*

$$\Big(\sum_{i=1}^{n} w_i \mathbf{n}_i\Big) \Big/ \Big\| \sum_{i=1}^{n} w_i \mathbf{n}_i \Big\|$$

*converges linearly to the real normal as the edge lengths $a_i$ tend to zero if all $\alpha_i$ are bounded within $(0, \pi)$ (meaning that there is a $\delta > 0$ such that always $\delta < \alpha_i < \pi - \delta$ for all $\alpha_i$).*

*If $P$ is a regular vertex, the convergence is quadratic.*

*Proof.* We have already shown the linear convergence. For regular vertices, the linear terms in (9.12) cancel, and we obtain the claim. □

We have also computed the angular error of the facet normal $\mathbf{n}_i$, for details see Appendix C. It seems sensible to us to interpret the inverse of that approximated error as confidence values for the respective normals, giving higher confidence to triangles whose normals are closer to the real normal. Using these values as weights for the faces incident at $P$ yields

$$\mathbf{n} \approx \Big(\sum_i \frac{\sin \alpha_i}{d_i} \mathbf{n}_i\Big) \Big/ \Big\| \sum_i \frac{\sin \alpha_i}{d_i} \mathbf{n}_i \Big\|$$

where $d_i$ is the length of the triangle edge opposing $P$, see the left hand side of Figure 9.4.

Further examples for the weights $w_i$ of Theorem 9.7 are area weights, uniform weights, and spherical weights [Max99].

## 9.3.2 Averaging of edges

Now we turn our attention to the mean curvature vector $\mathbf{k}$ at $P$. We use an approximation similar to the one in the paper by Meyer et al. [MDSB03]. That is, we approximate the integral over the mean curvature vector in a certain environment of $P$ and divide by the corresponding area $A$. Using the notation of Figure 9.6 (left), let $\overline{w}_i := \cot \gamma_i + \cot \delta_i$ and let $A := \frac{1}{8} \sum_i \overline{w}_i a_i^2$ be the Voronoi area of $P$ within the one-ring of $P$. The area $A$ can be computed in terms of $\gamma_i$ and $\delta_i$ because $\eta_i = \frac{\pi}{2} - \gamma_i$. We define

$$\mathbf{k} := \frac{1}{4A} \sum_i \overline{w}_i \mathbf{a}_i = 2 \frac{\sum_i \overline{w}_i \mathbf{a}_i}{\sum_i \overline{w}_i a_i^2} \approx H\mathbf{n}.$$

The same formula has been investigated by Xu [Xu06b].

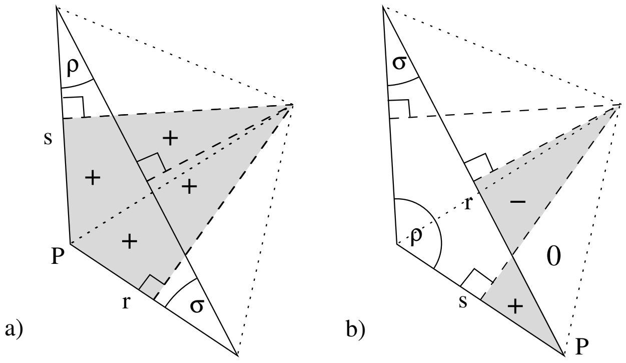


**Figure 9.5: Voronoi areas, as computed by the formula $\frac{1}{8}(r^2 \cot\rho + s^2 \cot\sigma)$, give an exact tiling of the triangle. Shown are the "Voronoi areas" of an a) obtuse angle, and b) acute angle.**

In contrast to Meyer et al. [MDSB03] where a "mixed region" for $A$ has been used to ensure that it lies completely within the one-ring of $P$, we use always the Voronoi region. The reason for this becomes clear in Theorems 9.9 and 9.10 where we show why these regions determine the "right" area.

**9.8 Remark.** Note that our approach is still compatible with the derivation given in [MDSB03]: the boundary of our "Voronoi region" (dashed lines in Figures 9.5 and 9.6) passes through the midpoints of the edges $\mathbf{a}_i$, and all "Voronoi regions" of the complete mesh together guarantee a perfect tessellation without overlapping if we admit "negative areas": even though the Voronoi cell at the obtuse angle exceeds the area of the triangle in Figure 9.5 a), this is compensated for by the "Voronoi cells" at the acute angles that count the same area negative (Figure 9.5 b)).

**9.9 Theorem (mean curvature vector I).** *Let $P$ be a mesh vertex and $\mathbf{k}$ be the approximated mean curvature vector at $P$ as above. Then $\mathbf{k}$ converges linearly towards a normal vector of $\mathcal{S}$ at $P$.*

*If $P$ is regular, the convergence is quadratic.*

*Proof.* Using the Taylor series of $\frac{\mathbf{a}_i}{a_i}$ in Appendix C, we get

$$\begin{aligned} 4A \cdot \mathbf{k} = \sum_i \overline{w}_i \mathbf{a}_i &= \sum_i \mathbf{t}_i \Big( \overline{w}_i a_i - \frac{\overline{w}_i a_i^3}{8} \kappa_i^2 + O(a_i^4) \Big) \\ &+ \sum_i \mathbf{v}_i \Big( - \frac{\overline{w}_i a_i^3}{6} \kappa_i \tau_i + O(a_i^4) \Big) \\ &+ \mathbf{n} \sum_i \Big( \overline{w}_i \frac{a_i^2}{2} \kappa_i + \overline{w}_i \frac{a_i^3}{6} \kappa_i' + O(a_i^4) \Big). \end{aligned}$$

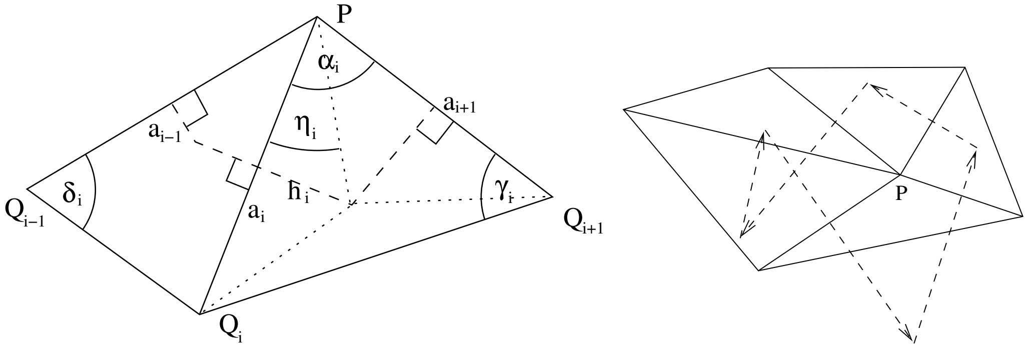


**Figure 9.6: Left: Cotangent weights and Voronoi region. Right: Complete Voronoi region.**

Therefore, all that remains to show (for linear convergence) is $\sum_i \overline{w}_i a_i \mathbf{t}_i = O(a_i)^3$. Since all the angles and consequently the weights $\overline{w}_i$ converge quadratically to the respective values in the tangent plane, we may assume that the one-ring of $P$ is already planar and $a_i \mathbf{t}_i = \mathbf{a}_i$. In that sense, we have to show that the $\overline{w}_i$ are homogeneous coordinates. We sketch the proof in the following.

Let $()^\perp$ be the $\frac{\pi}{2}$ counter-clockwise rotation. Then we can show equivalently that $(\sum_i \overline{w}_i \mathbf{a}_i)^\perp = \sum_i \overline{w}_i \mathbf{a}_i^\perp = 0$. From $\eta_i = \frac{\pi}{2} - \gamma_i$, we conclude $(\frac{1}{2}\cot\gamma_i \mathbf{a}_i)^\perp = \mathbf{h}_i$, where $\mathbf{h}_i$ is the part of the Voronoi edge separating $P$ and $Q_i$ that connects $\overrightarrow{PQ_i}$ to the circumcenter of the triangle $\triangle PQ_iQ_{i+1}$, see Figure 9.6 (left). This means that we need to show that the sum of the boundary edges of the Voronoi region is zero. This can easily be seen from Figure 9.6 (right) (and has previously been used in [PP93]). Note that this may not be a Voronoi region in the usual sense but it retains the important property that it forms a closed curve. We obtain

$$\begin{aligned}4A \cdot \mathbf{k} = &\sum_i \mathbf{t}_i\Big(-\frac{\overline{w}_i a_i^3}{8}\kappa_i^2 + O(a_{i-1}, a_i, a_{i+1})^3\Big)\\ &+\sum_i \mathbf{v}_i\Big(-\frac{\overline{w}_i a_i^3}{6}\kappa_i\tau_i + O(a_i^4)\Big)\\ &+\mathbf{n}\sum_i\Big(\overline{w}_i\frac{a_i^2}{2}\kappa_i + \overline{w}_i\frac{a_i^3}{6}\kappa_i' + O(a_i^4)\Big).\end{aligned} \tag{9.13}$$

For regular vertices, we know $a_i = a_{i+m}$ and $\alpha_i = \alpha_{i+m}$, and therefore, $\overline{w}_i = \overline{w}_{i+m}$, $\kappa_i \approx \kappa_{i+m}$, $\tau_i \approx \tau_{i+m}$, $\mathbf{t}_i \approx \mathbf{t}_{i+m}$, and $\mathbf{v}_i \approx \mathbf{v}_{i+m}$, see Appendix C. This means that the third order terms, which are explicitly given in the tangential components of (9.13), vanish, and a careful examination shows that the $O(a_{i-1}, a_i, a_{i+1})^3$ terms, stemming from the computations above, cancel as well. □

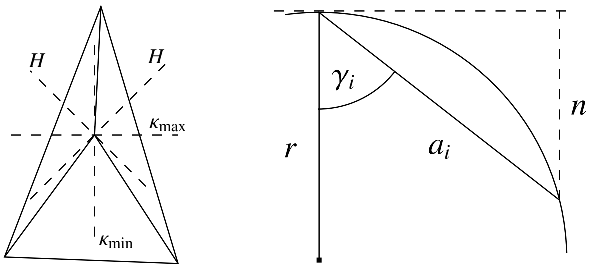


**Figure 9.7: Left: A vertex of valence 3 for which negative weights are required. Right: Estimation of the curvature of a sphere.**

Note, that it is also immediately possible to derive the convergence of $\mathbf{k}$ on spheres to the true mean curvature vector (particularly $||\mathbf{k}|| = H$) from (9.13). This follows from the fact that $\frac{\sum_i \overline{w}_i a_i^2/2}{4A} = 1$. Nevertheless, on arbitrary surfaces, $\mathbf{k}$ converges towards the correct mean curvature vector only under additional assumptions.

**9.10 Theorem (mean curvature vector II).** *Let*

$$\mathbf{k} = 2\frac{\sum_i(\cot\gamma_i + \cot\delta_i)\mathbf{a}_i}{\sum_i(\cot\gamma_i + \cot\delta_i)a_i^2}$$

*be the mean curvature vector at $P$ as defined above. Let all edge lengths $a_i$ be equal and let all angles $\alpha_i$ be equal. Then $||\mathbf{k}||$ converges linearly towards the real mean curvature.*

*If the edge lengths $a_i$ and angles $\alpha_i$ are varying then $||\mathbf{k}||$ will in general not converge towards the correct mean curvature.*

*Proof.* Under the assumption that all the $a_i$ and $\alpha_i$ are equal, the angles $\gamma_i$ and the angles $\delta_i$ (given by Figure 9.6, left) and therefore the weights $\overline{w}_i = \cot\gamma_i + \cot\delta_i$ are equal as well. Using (9.13), we obtain $||\mathbf{k}|| = \sum_i \frac{\overline{w}_i a_i^2}{\sum_i \overline{w}_i a_i^2}\kappa_i + O(a_i) = \frac{1}{n}\sum_i \kappa_i + O(a_i) = H + O(a_j)_j$. The last equation is well-known. It is a special case of Theorem 9.2 with all $\beta_i$ equal up to a quadratic error ($\beta_i = \alpha_i + O(a_i, a_{i+1})^2$, see Appendix C).

For the general case, Figure 9.7 (left) indicates a counter-example. If $\kappa_i < H$ along all (three) edges, then $||\mathbf{k}|| < H$ as well (because all the weights $\overline{w}_i$ are positive in this example). □

Further special cases, in which $\mathbf{k}$ converges towards the mean curvature vector are discussed in [Xu04]. We conclude that the cotangent weights are not perfectly appropriate to get the correct mean curvature for differing edge lengths and angles, even though they are well suited to obtain the correct normal. (Using the original, unmodified formula in [MDSB03] does not ameliorate the situation.) This motivates the question: can we approximate the mean curvature better if the normal vector is already known? The answer is "yes" (this is shown in Section 9.3.3).

**9.11 Remark.** Another drawback of the cotangent weights is, that they fail at boundary vertices. Since our analysis shows that only the normal component is important, it is most likely that the result of the cotangent weights can be improved (especially for boundary vertices) by taking only the normal component instead of the norm of the curvature vector (with respect to the true or an estimated normal vector).

### 9.3.3 Curvature estimation using quadratures

The Taylor expansions in Appendix C lead to the following proposition.

**9.12 Proposition.** *With the notations from Sections 9.2.1 and 9.2.6 the approximations*

$$\alpha_i \approx \beta_i, \quad w_i \approx \omega_i, \quad \textit{and } \widetilde{w}_i \approx \widetilde{\omega}_i$$

*converge quadratically as the edge lengths $a_i$ tend to zero. The normalized edge*

$$\frac{\mathbf{a}_i}{a_i} \approx \mathbf{t}_i$$

*converges linearly, and its projection to the tangent plane*

$$\frac{\mathbf{a}_i - \langle \mathbf{a}_i, \mathbf{n} \rangle \mathbf{n}}{\|\mathbf{a}_i - \langle \mathbf{a}_i, \mathbf{n} \rangle \mathbf{n}\|} \approx \mathbf{t}_i \tag{9.14}$$

*converges quadratically if the normal vector approximation* **n** *converges quadratically towards the true normal. Furthermore, the curvature approximation*

$$k_i = 2\frac{\langle \mathbf{a}_i, \mathbf{n} \rangle}{a_i^2} \approx \kappa_i$$

*converges linearly if* **n** *converges quadratically.*

**9.13 Remark.** If $\beta_i$ is approximated by the angle between the estimated tangent vectors (9.14), a slightly smaller, but still quadratic, error is achieved. In our experiments, we didn't observe a significant difference; therefore, we used the simpler approximation $\alpha_i$.

Now, we can derive corresponding results for our curvature approximations.

**9.14 Corollary.** *Let $P$ be a mesh vertex interpolating a smooth surface $\mathcal{S}$ as in Section 9.2.6. If $\mathbf{n}$ is a quadratic approximation of the true normal vector of $\mathcal{S}$ at $P$, then the approximations for $H$ and $K$ as given in Equation (9.11) converge linearly towards the real values of $H$ and $K$:*

$$\sum_i w_i k_i = H + O(a_i)_i \quad \textit{and} \quad 3H^2 - 2\sum_i \widetilde{w}_i k_i^2 + 4H(\sum_i \widetilde{w}_i k_i - H) = K + O(a_i)_i.$$

*If we use the approximations from Proposition 9.12 for $\widetilde{\omega}_i$, $\kappa_i$, and $\mathbf{t}_i$ to approximate the matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{E}}$, then the corresponding approximation of the Taubin integral representation of the curvature tensor (9.9) satisfies Theorem 9.5 up to a linear error, and we can obtain a linear approximation of the curvature tensor.*

*The preceding statements hold for arbitrary, but bounded (during the convergence) aspect ratios in the one-ring of $P$ provided that the weights $\omega_i$ and $\widetilde{\omega}_i$ exist (that is, the denominators are non-zero; see Section 9.3.4 for a discussion).*

*If $P$ is regular, then the vector*

$$2\sum_i \frac{\tan\alpha_{i-1} + \tan\alpha_i}{\sum_j(\tan\alpha_{j-1} + \tan\alpha_j)} \frac{\mathbf{a}_i}{a_i^2}$$

*converges linearly towards the mean curvature vector at $P$ (regardless of $\mathbf{n}$).*

*Proof.* The first paragraph follows immediately from Proposition 9.12 and Theorems 9.2, 9.4, and 9.5. The last statement holds, because the lower order tangential components of $\frac{\mathbf{a}_i}{a_i^2}$ vanish for regular vertices as can be seen in Appendix C and the normal component contains the approximation of $H$ from Equation (9.11). □

**9.15 Proposition.** *If the normals are approximated with spherical weights [Max99] in combination with (9.11) to estimate the curvature of a sphere, they yield the exact values of mean and Gaussian curvature.*

*Proof.* For a sphere, spherical weights yield exact normals. Let $\gamma_i$ be the angle between the inward pointing normal $\mathbf{n}$ and an edge $\mathbf{a}_i$. Then it is easy to see that the length of $\mathbf{a}_i$ is $a_i = 2r\cos\gamma_i$ and the normal component $n := \langle \mathbf{a}_i, \mathbf{n} \rangle$ of $\mathbf{a}_i$ is $n = 2r\cos^2\gamma_i$ for a sphere of radius $r$, see Figure 9.7 (right). Therefore, $k_i = 2\frac{n}{a_i^2} = \frac{1}{r} = \kappa_i$ and the claims follow. □

## 9.3.4 Necessary conditions and implementation details

A general problem when computing the curvature of a surface from few samples is that the data may be insufficient to determine the curvature uniquely. This was discussed by Cazals and Pouget [CP03] as the necessity that the interpolating problem is *poised* in order to find a unique fitting polynomial. In our quadrature formulae, the denominators of the weights (9.7) and (9.8), respectively, become zero if insufficient data is available. This happens for example if only two samples are available or if two consecutive angles add up to $\pi$, that is $\beta_{i-1} + \beta_i = \pi$, for all $i$. Nevertheless, in some of these cases, the data would be sufficient to compute at least the *mean* curvature, for example if $\beta_i = \frac{\pi}{2}$ for all $i$. In the future, we would like to find weights that are well defined for these cases as well.

If we compute the approximations (9.10) and the value of the denominator is zero or close to zero, this leads to numerical problems. To cope with a situation of a very small denominator for a vertex, we omit some of the edges in the one-ring of that vertex. Since our quadratures are exact (if the $\kappa_i$ and $\beta_i$ are known), we can apply our formulae for any subset ($\geq 3$) of the edges, and the quadratures yield the same result. In practice, however, numerical issues can lead to quite different results for different edge (sub)sets. Our experiments yielded a proportional relationship between the absolute value of the denominator and the reliability of the result. Therefore, we chose that subset of edges in the one-ring such that the denominator becomes maximal. (Remember that we only need three edges to apply our method.) This improved our results for these cases substantially.

## 9.4 Experimental Results

It remains to show that our new weights are not only of theoretical value, but also of practical use. To that end, we present at first detailed results of our quadrature method to compute the mean curvature; afterwards, we briefly compare our method and existing approaches to compute the curvature tensor.

### 9.4.1 Convergence properties

The top row of Figure 9.8 demonstrates that our method converges even for irregular meshes. We sampled the graph of $f(u, v) = 0.1\, \mathrm{e}^{2u+v-v^2}$ (the same function was used by Cazals and Pouget [CP03]) and computed the root-mean-square error and maximum error while increasing the sampling density. More precisely, the graph of $f$ is sampled at $(x, y, f(x, y))$ where the points $(x, y)$ lie on a randomly perturbed rectangular grid of side length $h$ within $[-2, 2] \times [-2, 2]$. The perturbation is done by adding a vector of random direction and length $0.3h$ to each point $(x, y)$. During the experiments, we decreased $h$ from 0.128 to 0.002. Figure 9.8 depicts the graph of $f$ for $h = 0.128$ at the top left and the measured errors at the top right. Our tangent weights show a linear convergence as predicted in Theorem 9.14 while conventional cotangent weights [MDSB03] do not converge.

## 9.4.2 Error distribution

We applied our newly proposed formula for mean curvature approximation to several meshes whose vertices interpolate smooth surfaces of constant mean curvature. For these meshes, it is especially easy to compare the estimated values with the true values. We studied the following methods: cotangent weights using a mixed area as proposed by [MDSB03] and tangent weights as defined in Theorem 9.2. Since, for the last method, knowledge of the normal vectors is necessary, we compared the results when using the true normals which were available for our test surfaces, and using estimated normals with spherical weights [Max99] to study the case where the true normals are unknown. The mesh of a catenoid (1575 vertices, 126 on the boundary) and the corresponding histogram of computed curvature values are depicted in Figure 9.8 (bottom).

Here, the result of the preceding convergence analysis is confirmed. While the cotangent weights mean curvature estimations are scattered over a large range of values (**R**oot **M**ean **S**quare error: 0.37, **max**imum error: 10.6), the tangent weights method gave good approximations of the correct value zero (RMS: 0.10, max: 0.53). Also, the importance of a good normal approximation could be seen here: using the estimated normals, the tangent and cotangent weights led to similar results (RMS: 0.20, max: 1.7 for tangent weights with estimated normals). But note that the correct zero value seems to be avoided by the cotangent method. This is probably an inherent problem of computing the mean curvature as the norm of a vector (compare Remark 9.11 for a possible solution).

We also computed the curvature of several spheres to verify Proposition 9.15. In all cases, the maximum error was smaller than $10^{-10}$.

## 9.4.3 Curvature tensor

We tested our algorithm to compute the Taubin integral representation of the curvature tensor (9.4) on a model of a torus with Gaussian curvature varying from $-1$ to $\frac{1}{3}$ (Figure 9.9). The results are given in Table 9.1. We measured the root-mean-square error and the maximum error for the mean curvature $H$, the Gaussian curvature $K$, and the angular deviation $\theta$ of the principal directions in radians, and the time in seconds computed on a 1.8 GHz CPU (Intel Xeon). No optimization to achieve especially short running times was done for our algorithm. We compared the following methods: (a) cotangent weights to compute the mean curvature [MDSB03], (b) our tangent weights to compute the mean curvature (Section 9.2.2), the Gaussian curvature (Section 9.2.3), and the principal directions (Section 9.2.5) (the time is measured for computing one, two, or all three columns (from left), respectively) (c) the original Taubin method [Tau95a], (d) the Taubin tensor computed by exact quadrature as presented in this chapter (Section 9.2.4),

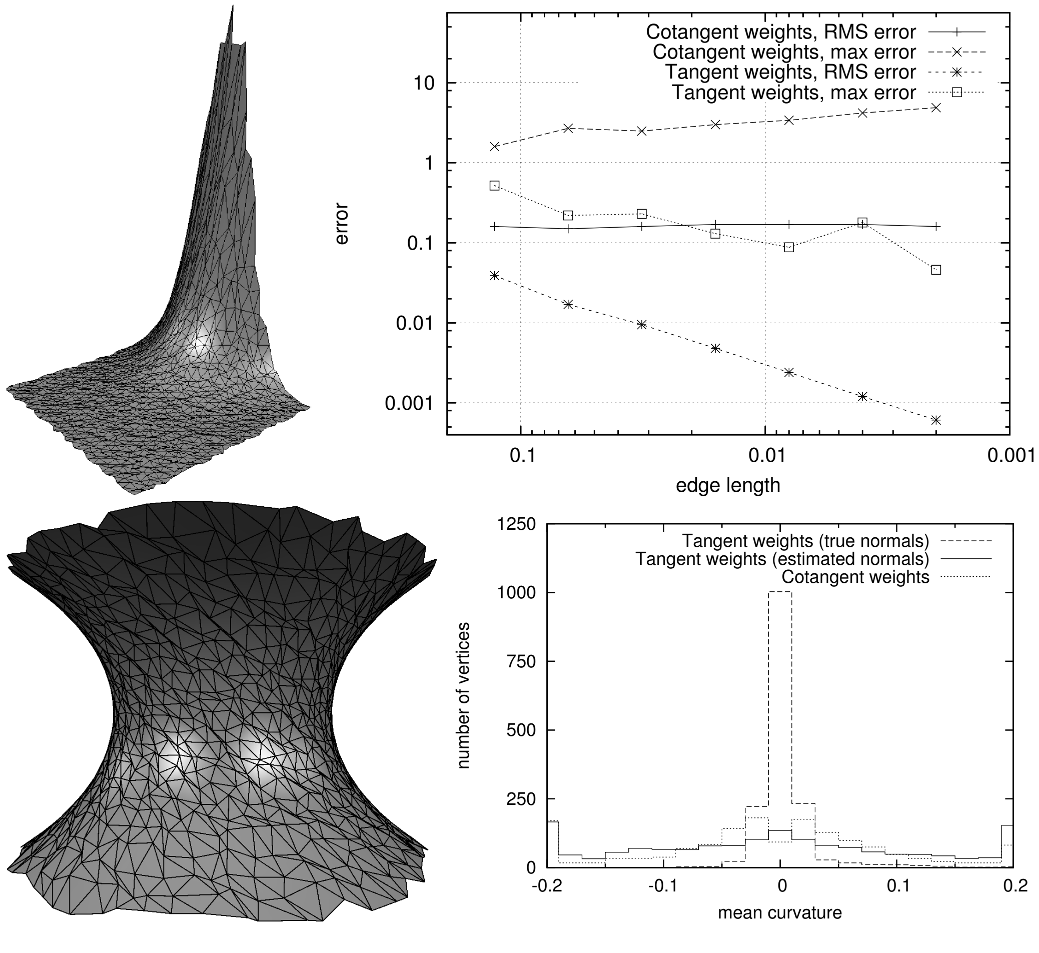


**Figure 9.8: Comparison of cotangent weights and our tangent weights. Top: The graph of $f(u,v) = 0.1\,\mathrm{e}^{2u+v-v^2}$ and the measured root-mean-square and maximum error when the sampling density is increased. Bottom: A catenoid (constant mean curvature zero) and the corresponding curvature distribution. The leftmost and rightmost columns in the histogram count not only the vertices of the specified curvature but also all vertices with lower or higher curvature, respectively.**

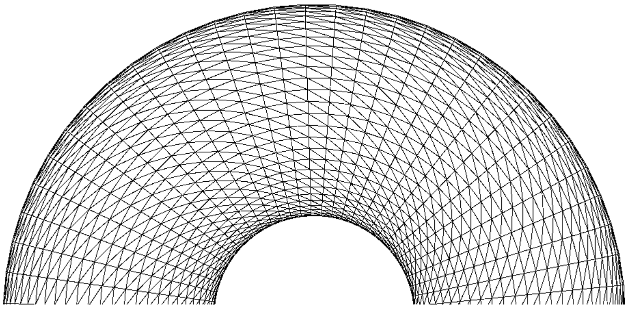


**Figure 9.9: The mesh of a rectangularly tessellated torus with 3969 vertices.**

**Table 9.1: RMS error (maximum error in brackets) and time for the torus model; italicized methods are developed in this chapter.**

| | $H$ | | $K$ | | $\theta$ | | T (s) |
|---|---|---|---|---|---|---|---|
| (a) cotangent weights | 0.00057 | (0.0049) | – | | – | | 0.10 |
| (b) *tangent weights* | 0.00081 | (0.0013) | 0.042 | (0.11) | 0.14 | (0.29) | 0.10/0.18/0.27 |
| (c) Taubin (classic) | 0.082 | (0.10) | 0.38 | (0.80) | 0.16 | (0.30) | 0.10 |
| (d) *Taubin (quadr.)* | 0.00081 | (0.0013) | 0.052 | (0.18) | 0.14 | (0.19) | 0.26 |
| (e) GI (cubic) | 0.0046 | (0.012) | 0.0079 | (0.017) | 0.0018 | (0.0024) | 0.57 |

(e) cubic polynomial fitting as done by Goldfeather and Interrante [GI04]. Here, all computations were done with the exact normals known.

If only the mean curvature is of interest, our method is competitive with all other methods. The real power of our method can be observed by comparing (c) and (d) in Table 9.1. The only difference lies in the discretization of the integral. The small, but decisive, change in the weights improves the Taubin tensor by orders of magnitude of accuracy while it retains relatively short running times. Nevertheless, method (b) yields similar results in a shorter time for the mean and Gaussian curvature. Comparing our method (d) with the approach of Goldfeather and Interrante (e), it can be seen that method (e) offers the best results for computing the principal directions. But if only the mean and Gaussian curvature are of interest, our approach yields good results and short running times, especially if good normals are available. Our approach combines high accuracy and relatively fast computation.

We ran additional tests with exact mean and Gaussian curvature to find out how much the angular error in our methods is affected by the errors in the computation of $H$ and $K$. It turned out that method (d) improves only to a small degree (angular error RMS: 0.12, max: 0.16) while method (b) profits strongly from correct values of $H$ and $K$ (angular error RMS: 0.00049, max: 0.0028). Note, that we still used approximated values for the normal curvatures in both cases.

## 9.5 Summary and Future Work

We have presented an approach for the exact computation of curvature integrals that leads to quite impressive results and relies only on elementary mathematics. Our experiments showed short running times and often superior results compared to existing methods. While many approaches yield satisfactory results for regular meshes (only), our weights can deal with arbitrary aspect ratios. Furthermore, the method is proven to converge if normal vectors of at least quadratic accuracy are available.

The power of our approach becomes especially clear when comparing the error of the Taubin tensor computed with our weights and the original weights. The change in the discretization of the integral improves the result by orders of magnitude of accuracy. While at present, we are not able to compete with Goldfeather and Interrante (except for the mean curvature), our formula exhibits mathematical elegance and simplicity. By the modular structure of method (b), the already short running time can be further reduced if only the curvatures, and not the principal directions, are of interest.

Furthermore, we introduced a framework for the asymptotic analysis of differential properties of discrete surfaces. Using it, we have shown that all commonly used weighting schemes for estimating vertex normals on a mesh from the normals of the incident faces converge linearly towards the true normal in general and quadratically at regular vertices. The same is true for the mean curvature vector in the variant suggested in this chapter.

In the future, we hope to find a way to compute the tangent weights without explicit evaluation of numerator and denominator to make their computation more stable. Furthermore, we plan to examine the influence of noise on normal and curvature estimations. It would also be interesting to obtain extensions of our main formula (9.5) to nonlinear Laplacians similar to those considered in [Wei94, DW06].

# Chapter 10

# Conclusions

Barycentric coordinates constitute a fascinating research topic, the investigation of which is not yet complete. In this final chapter, we summarize our results once more and point out possible directions for future research. We intend to complement the summaries of the individual chapters and put them into a broader context.

In the first part, we aimed to remove shortcomings of previously existing barycentric coordinate systems. The *spherical barycentric coordinates* transfer the concept of barycentric coordinates from Euclidean spaces to spheres. Although spherical barycentric coordinates are not new, very little work exists in this direction, for example Möbius' coordinates for spherical triangles [Möb46] and Ju et al.'s vector coordinates [JSWD05], which are defined for convex spherical polygons. Our scheme, in contrast, allows the definition of counterparts of any Euclidean barycentric coordinates in the spherical domain. In particular, we introduced spherical mean value coordinates, which are defined for arbitrary spherical polygons without antipodal points. Besides being valuable on their own, for example for the interpolation of vectors, these coordinates proved to be useful in the construction of general *3D barycentric coordinates*, in particular 3D mean value coordinates.

While only 3D mean value coordinates for triangular polyhedra were known so far, we gave a definition of 3D mean value coordinates for arbitrary polygonal polyhedra. We noted that these coordinates are no longer uniquely determined by the integral definition, which was used in previous work. This definition for mean value interpolation is based on a distance weighted integral of a function on the polyhedron. In the discrete scenario, function values are only given at the vertices of the polyhedron. For triangular meshes, this function could be extended to the faces by linear interpolation. This is in general not possible for general polygonal faces. Therefore, we chose to define 3D mean value coordinates for polygonal meshes in a way that 2D mean value interpolation is used on the faces.

Our approach allows us to define other types of 3D barycentric coordinates for polygonal meshes as well. In these cases, it seems promising to choose the same type of interpolation on polygonal faces as shall be achieved for the polyhedron. Nevertheless, determining the “best” choice remains an open problem. For example, we could not prove that the 3D Wachspress coordinates obtained with this approach coincide with the known 3D Warren-Wachspress coordinates. Although we cannot know how these issues will be solved in the future, our framework to construct barycentric coordinates is versatile enough to support all possible choices, and it allows an easy generalization to polytopes in higher dimensions.

An extension of the notion of barycentric coordinates, which we called *higher order barycentric coordinates*, allows us to interpolate not only function values but linear functions. When used for space deformations, they introduce new means to manipulate objects. They can specify rotations and other linear transformations directly without the need to “simulate” such a transformation by moving a group of nearby control points. We suggested a method to modify existing barycentric coordinates to create higher order barycentric coordinates. Therefore, they might rather be considered a possible extension of existing coordinates than a completely new type. If we, nevertheless, compare higher order mean value coordinates and conventional barycentric coordinates, higher order mean value coordinates retain the good properties of conventional mean value coordinates without most of their shortcomings.

However, our higher order barycentric coordinates retain one disadvantage of classical barycentric coordinates as well: if the control net consists of several adjoining parts, the coordinates are in general not smooth across polytope faces. This would be necessary to deform several parts of a model independently of each other with barycentric space deformation. Although this problem can be resolved by changing the derivative axiom, the actual construction of such coordinates remains *future work*.

Our definitions of spherical coordinates, 3D coordinates, and higher order coordinates allow the construction of a wide range of related coordinates. However, it would be desirable to explore the space of these coordinates systematically, similar to the analysis done in [FHK06, JLW07]. In this way, it would be possible to select barycentric coordinates that are tailored to particular needs.

Furthermore, we considered generalized *Bézier maps*. A generalized Bézier map is a map that is piecewise (on a given polytope) a homogeneous polynomial in generalized barycentric coordinates. We showed how the coefficients of the Bernstein polynomials have to be chosen to enforce smoothness of any desired order across common (hyper-)faces of polytopes. We decided to develop the theory in full generality although we mainly aimed at Bézier maps in mean value coordinates which are defined with respect to polytopes of arbitrary shape. This allows the use of our results for any other barycentric coordinates that might become the focus of attention in the future. Moreover, it shows that many results from the well developed field of simplicial and tensor product Bézier theory can be considered as a special case of our findings if Wachspress coordinates are used. Our indexing scheme, however, does not coincide with the traditional indexing scheme for tensor product Bézier maps. This sheds new light on the classical theory, which will hopefully lead to a better understanding of the tensor product Bézier maps as well.

Probably the most important examples of Bézier maps are Bézier curves and surfaces, and space deformations. We observe that mean value Bézier surfaces are qualitatively comparable to tensor product surfaces, and hence well-suited for modeling surfaces. Although the advantage of the greater number of control points (compared to tensor product Bézier surfaces of the same degree) is diminished by a greater number of constraints, we nevertheless obtain new, interesting possibilities for surface construction.

However, a number of *open questions* remain. Foremost, some kind of spline representation of Bézier maps has to be found that resolves any continuity issues fully automatically. These splines should allow the placement of meaningful control points directly during the design of surfaces and deformations without needing to spend much time on the cumbersome process of satisfying the continuity constraints manually. Another issue that we did not discuss are rational Bézier maps. The use of rational Bézier maps greatly expanded the capabilities of classical Bézier theory. The same should be done for generalized Bézier maps.

Let us now briefly compare the two approaches for space deformation presented in this thesis: (higher order) barycentric coordinates and Bézier maps. We believe that both have their merits. Higher order barycentric coordinates yield an easy and intuitive user interface to control the deformation. Furthermore, no additional complexity is imposed if control nets of complex topology are used. However, if a high degree of local control is desired, and the control net is divided into several smaller control nets, usually only $C^0$-continuity is guaranteed across the boundaries of the polyhedra. Bézier maps can achieve continuity of arbitrary order, and further guarantees, like the convex hull property, can be given. This comes at the cost of having a high number of continuity conditions for the control points. Satisfying all of them simultaneously can be difficult unless and until a better representation for them is found.

Finally, we have presented several approximations for differential properties of curves and surfaces. Our main result in this field is the development of *quadrature formulae* for surface curvatures, which lead to novel approximations for the curvature of discrete surfaces. Our experiments showed short running times and often superior results compared to existing methods. While many approaches yield satisfactory results for regular meshes (only), our weights can deal with arbitrary aspect ratios. The power of our approach becomes apparent when the Taubin tensor, which is based on a curvature integral, is computed with our weights instead of the original weights. The change in the discretization of the integral improves the result by orders of magnitude of accuracy. Moreover, our method is proven to converge if normal vectors of at least quadratic accuracy are available.

Furthermore, we introduced a framework for the asymptotic analysis of differential properties of discrete curves and surfaces. Its application yielded several formulae to estimate curvature, torsion, and the Frenet frame of a space curve, such that they converge towards their smooth counterparts as edge lengths tend to zero. We proved the optimality of our estimates in many cases. Moreover, we have shown that all commonly used weighting schemes for estimating vertex normals on a mesh from the normals of the incident faces converge linearly towards the true normal in general and quadratically at regular vertices. The same is true for the mean curvature vector in the variant suggested in this paper.

An important task for *future work* is to find a way to compute the tangent weights without explicit evaluation of the numerator and the denominator in our curvature formulae in order to make their computation more stable. Furthermore, the asymptotic properties of estimations of curvature and torsion of curves on surfaces and the influence of noise on our estimations should be explored.

# Appendix A

# Proof of Theorem 5.1

To prove the theorem, we check the axioms.

**Lagrange property.** Using the Lagrange property of $\lambda_i$ and $m(0) = 0,\ m(1) = 1$, we obtain

$$\frac{m \circ \lambda_i}{\sum_k m \circ \lambda_k}(\mathbf{v}_j) = \frac{m(\delta_{ij})}{\sum_k m(\delta_{kj})} = \delta_{ij}.$$

**Partition of unity.**

$$\sum_i \frac{m \circ \lambda_i}{\sum_k m \circ \lambda_k} = \frac{\sum_i m \circ \lambda_i}{\sum_k m \circ \lambda_k} = 1.$$

**Derivative property.** If the $\lambda_i$ are differentiable at the vertices, the derivative property can be proven by

$$\begin{aligned}
&\nabla \frac{m \circ \lambda_i}{\sum_k m \circ \lambda_k}(\mathbf{v}_j) \\
&\quad = \frac{\nabla(m \circ \lambda_i)(\mathbf{v}_j) \cdot \sum_k m \circ \lambda_k(\mathbf{v}_j) - m \circ \lambda_i(\mathbf{v}_j) \cdot \sum_k \nabla(m \circ \lambda_k)(\mathbf{v}_j)}{(\sum_k m \circ \lambda_k(\mathbf{v}_j))^2} \\
&\quad = m'(\lambda_i(\mathbf{v}_j)) \cdot \nabla\lambda_i(\mathbf{v}_j) - m(\lambda_i(\mathbf{v}_j)) \cdot \sum_k m'(\lambda_k(\mathbf{v}_j)) \cdot \nabla\lambda_k(\mathbf{v}_j) = \mathbf{0}
\end{aligned}$$

since $m'(0) = m'(1) = 0$. If the $\lambda_i$ are not differentiable, we have to resort to computing the derivatives of $m \circ \lambda_i(\mathbf{v}_j)$ in the above equation as the limit of difference quotients. In the following, we use the definition of the difference quotient, we add 0, we use the Taylor formula of $m$, we use the Hölder condition, we combine the terms, and use $\alpha > \frac{1}{2}$.

$$\begin{aligned}
|\frac{\partial}{\partial x_k}(m \circ \lambda_i)(\mathbf{v}_j)| &= \lim_{h\to 0} \frac{|m \circ \lambda_i(\mathbf{v}_j + h\mathbf{e}_k) - m \circ \lambda_i(\mathbf{v}_j)|}{|h|} \\
&= \lim_{h\to 0} \frac{\Big|m\Big(\lambda_i(\mathbf{v}_j) + \Big(\lambda_i(\mathbf{v}_j + h\mathbf{e}_k) - \lambda_i(\mathbf{v}_j)\Big)\Big) - m\Big(\lambda_i(\mathbf{v}_j)\Big)\Big|}{|h|} \\
&= \lim_{h\to 0} \frac{1}{|h|}\Big|m\Big(\lambda_i(\mathbf{v}_j)\Big) + \Big(\lambda_i(\mathbf{v}_j + h\mathbf{e}_k) - \lambda_i(\mathbf{v}_j)\Big)m'\Big(\lambda_i(\mathbf{v}_j)\Big) \\
&\quad + \frac{1}{2}\Big(\lambda_i(\mathbf{v}_j + h\mathbf{e}_k) - \lambda_i(\mathbf{v}_j)\Big)^2 m''(\xi) - m\Big(\lambda_i(\mathbf{v}_j)\Big)\Big| \\
&\leq \lim_{h\to 0} \frac{\frac{1}{2}C|h|^{2\alpha}|m''(\xi)|}{|h|} = \lim_{h\to 0} \frac{1}{2}C|h|^{2\alpha-1}|m''(\xi)| = 0.
\end{aligned}$$

**Domain.** Since $m$ is defined everywhere in $\mathbb{R}$, $m \circ \lambda_i$ is defined on the domain of $\lambda_i$. Furthermore, the denominator is always greater than zero, because $m$ is non-negative, maps positive numbers to positive numbers, and at least one $\lambda_i(\mathbf{x})$ is positive for every $\mathbf{x}$ because the $\lambda_i$ form a partition of unity.

**Non-negativity.** Since $m(x) > 0$ for $x > 0$ and the denominator is greater than zero, positivity is preserved.

**Smoothness.** Since $m$ is $C^1$-continuous everywhere, the $C^1$-continuity of the coordinate functions apart from the vertices is preserved. At the vertices, $C^1$-continuity has been proven for the derivative property.

# Appendix B

# Taylor Series Expansion of Space Curves

In this appendix, we conduct an asymptotic analysis of an arbitrary space curve $\mathbf{r}(s)$, with $\mathbf{r}(s_i) = P_i$, $s_0 = 0$, interpolated by a polyline as in Figure 8.1. Our treatment is based on the work of Anoshkina et al. [ABS02], but we have to take into account the higher complexity of three-dimensional space; in particular, the notion of torsion has no meaning for planar curves.

We assume without loss of generality that the curve is parameterized by arc length. This facilitates the problem to express discrete properties in geometrically meaningful terms like curvature and torsion by using Taylor series along with the well known Frenet equations [dC76, Koe90]

$$\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n}, \quad \frac{d\mathbf{b}}{ds} = \tau\mathbf{n}, \quad \frac{d\mathbf{n}}{ds} = -\kappa\mathbf{t} - \tau\mathbf{b}, \tag{B.1}$$

where $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{b}$ are the unit tangent, the unit normal and the unit binormal vector, respectively, and $\kappa$ and $\tau$ are curvature and torsion, respectively. (We omit the position $s$ since the equations hold for all (fixed) $s$ and we are interested only in the case $s = 0$, anyway.) Differentiating the curve $\mathbf{r}(s)$ then yields

$$\begin{aligned}
\mathbf{r}' = \mathbf{t}, \quad \mathbf{r}'' &= \mathbf{t}' = \kappa\mathbf{n}, \quad \mathbf{r}''' = (\kappa\mathbf{n})' = \kappa'\mathbf{n} - \kappa^2\mathbf{t} - \kappa\tau\mathbf{b}, \\
\mathbf{r}^{(4)} &= -3\kappa\kappa'\mathbf{t} + (\kappa'' - \kappa^3 - \kappa\tau^2)\mathbf{n} - (2\kappa'\tau + \kappa\tau')\mathbf{b}, \\
\mathbf{r}^{(5)} &= (\kappa^4 + \kappa^2\tau^2 - 4\kappa\kappa'' - 3(\kappa')^2)\mathbf{t} \\
&\quad - (6\kappa^2\kappa' + 3\kappa\tau\tau' + 3\kappa'\tau^2 - \kappa''')\mathbf{n} \\
&\quad + (\kappa^3\tau + \kappa\tau^3 - \kappa\tau'' - 3\kappa'\tau' - 3\kappa''\tau)\mathbf{b},
\end{aligned}$$

and so on. Now we can use Taylor expansion to express the edge $\mathbf{e} = \overrightarrow{P_0P_1} =$

$\mathbf{r}(s_1) - \mathbf{r}(0)$ in the local canonical form [Kre59, Sau70]:

$$
\begin{aligned}
\mathbf{e} &= s_1\mathbf{r}' + \frac{s_1^2}{2}\mathbf{r}'' + \frac{s_1^3}{6}\mathbf{r}''' + \frac{s_1^4}{24}\mathbf{r}^{(4)} + \frac{s_1^5}{120}\mathbf{r}^{(5)} + O(s_1^6)\\
&= \mathbf{t}\Big(s_1 - \frac{s_1^3}{6}\kappa^2 - \frac{s_1^4}{8}\kappa\kappa'\frac{s_1^5}{120}(\kappa^4 + \kappa^2\tau^2 - 4\kappa\kappa'' - 3(\kappa')^2) + O(s_1^6)\Big)\\
&\quad + \mathbf{n}\Big(\frac{s_1^2}{2}\kappa + \frac{s_1^3}{6}\kappa' + \frac{s_1^4}{24}(\kappa'' - \kappa^3 - \kappa\tau^2)\\
&\qquad - \frac{s_1^5}{120}(6\kappa^2\kappa' + 3\kappa\tau\tau' + 3\kappa'\tau^2 - \kappa''') + O(s_1^6)\Big)\\
&\quad + \mathbf{b}\Big(-\frac{s_1^3}{6}\kappa\tau - \frac{s_1^4}{24}(2\kappa'\tau + \kappa\tau')\frac{s_1^5}{120}(\kappa^3\tau + \kappa\tau^3 - \kappa\tau'' - 3\kappa'\tau' - 3\kappa''\tau) + O(s_1^6)\Big).
\end{aligned}
$$

In the next step, we express $\mathbf{e}$ in terms of its length $e$ without using the possibly unknown geodesic length $s_1$. Since $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ is an orthonormal basis, we can compute $e$ in terms of $s_1$ by

$$
\begin{aligned}
\|\mathbf{e}\|^2 &= s_1^2 - \frac{s_1^4}{12}\kappa^2 - \frac{s_1^5}{12}\kappa\kappa' + \frac{s_1^6}{360}(\kappa^4 + \kappa^2\tau^2 - 9\kappa\kappa'' - 8(\kappa')^2) + O(s_1^7),\\
e := \|\mathbf{e}\| &= s_1 - \frac{s_1^3}{24}\kappa^2 - \frac{s_1^4}{24}\kappa\kappa' + \frac{s_1^5}{5760}(3\kappa^4 + 8\kappa^2\tau^2 - 72\kappa\kappa'' - 64(\kappa')^2) + O(s_1^6).
\end{aligned}
$$

After inverting the Taylor series for $e$, we obtain

$$
s_1 \;=\; e + \frac{e^3}{24}\kappa^2 + \frac{e^4}{24}\kappa\kappa' + \frac{e^5}{5760}(27\kappa^4 \;-\; 8\kappa^2\tau^2 \;+\; 72\kappa\kappa'' \;+\; 64(\kappa')^2) \;+\; O(e^6).
$$

Substituting the expansion of $s_1$ into the formula for $\mathbf{e}$ and dividing by $e$ yields

$$
\begin{aligned}
\frac{\mathbf{e}}{e} &= \mathbf{t}\Big(1 - \frac{e^2}{8}\kappa^2 - \frac{e^3}{12}\kappa\kappa' - \frac{e^4}{1152}(9\kappa^4 - 8\kappa^2\tau^2 + 24\kappa\kappa'' + 16(\kappa')^2) + O(e^5)\Big)\\
&\quad + \mathbf{n}\Big(\frac{e}{2}\kappa + \frac{e^2}{6}\kappa' + \frac{e^3}{24}(\kappa'' - \kappa\tau^2) + \frac{e^4}{240}(3\kappa^2\kappa' - 6\kappa\tau\tau' - 6\kappa'\tau^2 + 2\kappa''') + O(e^5)\Big)\\
&\quad + \mathbf{b}\Big(-\frac{e^2}{6}\kappa\tau - \frac{e^3}{24}(2\kappa'\tau + \kappa\tau')\\
&\qquad - \frac{e^4}{240}(3\kappa^3\tau - 2\kappa\tau^3 + 2\kappa\tau'' + 6\kappa'\tau' + 6\kappa''\tau) + O(e^5)\Big).
\end{aligned}
$$

In a similar fashion, we obtain $\frac{\mathbf{f}}{f}$ from the difference of two Taylor expansions, $\mathbf{f} = \mathbf{r}(s_1 + (s_2 - s_1)) - \mathbf{r}(s_1)$,

$$\begin{aligned}
\frac{\mathbf{f}}{f} = {} & \mathbf{t}\Big(1 - \frac{f^2 + 4fe + 4e^2}{8}\kappa^2 - \frac{f^3 + 5f^2e + 9fe^2 + 6e^3}{12}\kappa\kappa' + O(e,f)^4\Big) \\
& + \mathbf{n}\Big(\frac{f + 2e}{2}\kappa + \frac{f^2 + 3fe + 3e^2}{6}\kappa' - \frac{f^2e + 2fe^2 + e^3}{8}\kappa^3 \\
& \qquad + \frac{f^3 + 4f^2e + 6fe^2 + 4e^3}{24}(\kappa'' - \kappa\tau^2) + O(e,f)^4\Big) \\
& + \mathbf{b}\Big(-\frac{f^2 + 3fe + 3e^2}{6}\kappa\tau - \frac{f^3 + 4f^2e + 6fe^2 + 4e^3}{24}(2\kappa'\tau + \kappa\tau') + O(e,f)^4\Big),
\end{aligned}$$

and in the same way we get the expressions for $\frac{\mathbf{d}}{d}$ and $\frac{\mathbf{c}}{c}$:

$$\begin{aligned}
\frac{\mathbf{d}}{d} = {} & \mathbf{t}\Big(1 - \frac{d^2}{8}\kappa^2 + \frac{d^3}{12}\kappa\kappa' + O(d^4)\Big) \\
& + \mathbf{n}\Big(-\frac{d}{2}\kappa + \frac{d^2}{6}\kappa' - \frac{d^3}{24}(\kappa'' - \kappa\tau^2) + O(d^4)\Big) \\
& + \mathbf{b}\Big(-\frac{d^2}{6}\kappa\tau + \frac{d^3}{24}(2\kappa'\tau + \kappa\tau') + O(d^4)\Big), \\
\frac{\mathbf{c}}{c} = {} & \mathbf{t}\Big(1 - \frac{c^2 + 4cd + 4d^2}{8}\kappa^2 + \frac{c^3 + 5c^2d + 9cd^2 + 6d^3}{12}\kappa\kappa' + O(c,d)^4\Big) \\
& + \mathbf{n}\Big(-\frac{c + 2d}{2}\kappa + \frac{c^2 + 3cd + 3d^2}{6}\kappa' + \frac{c^2d + 2cd^2 + d^3}{8}\kappa^3 \\
& \qquad - \frac{c^3 + 4c^2d + 6cd^2 + 4d^3}{24}(\kappa'' - \kappa\tau^2) + O(c,d)^4\Big) \\
& + \mathbf{b}\Big(-\frac{c^2 + 3cd + 3d^2}{6}\kappa\tau + \frac{c^3 + 4c^2d + 6cd^2 + 4d^3}{24}(2\kappa'\tau + \kappa\tau') + O(c,d)^4\Big).
\end{aligned}$$

Using these series, we can compute the cross product of $\frac{\mathbf{d}}{d}$ and $\frac{\mathbf{e}}{e}$:

$$\begin{aligned}
\frac{\mathbf{d}}{d} \times \frac{\mathbf{e}}{e} = {} & \mathbf{t}\Big(\frac{d^2e + de^2}{12}\kappa^2\tau + O(d,e)^4\Big) \\
& + \mathbf{n}\Big(\frac{e^2 - d^2}{6}\kappa\tau + \frac{d^3 + e^3}{24}(2\kappa'\tau + \kappa\tau') + O(d,e)^4\Big) \\
& + \mathbf{b}\Big(\frac{d + e}{2}\kappa + \frac{e^2 - d^2}{6}\kappa' + \frac{d^3 + e^3}{24}(\kappa'' - \kappa\tau^2) - \frac{d^2e + de^2}{16}\kappa^3 + O(d,e)^4\Big).
\end{aligned}$$

Note that the quadratic terms vanish for $d = e$. The same is true for fourth order terms:

$$\begin{aligned}\frac{\mathbf{d}}{d}\times\frac{\mathbf{e}}{e} \overset{d=e}{=} &\mathbf{t}\Big(\frac{e^3}{6}\kappa^2\tau + O(e^5)\Big)\\ &+\mathbf{n}\Big(\frac{e^3}{12}(2\kappa'\tau + \kappa\tau') + O(e^5)\Big)\\ &+\mathbf{b}\Big(e\kappa + \frac{e^3}{12}(\kappa'' - \kappa\tau^2) - \frac{e^3}{8}\kappa^3 + O(e^5)\Big).\end{aligned}$$

Since the norm of the above vector equals $\sin\varphi$, we obtain

$$\begin{aligned}\sin\varphi &= \frac{d+e}{2}\kappa - \frac{d^2-e^2}{6}\kappa' + \frac{(d-e)(d^2-e^2)}{36}\kappa\tau^2\\ &\quad + \frac{d^3+e^3}{24}(\kappa'' - \kappa\tau^2) - \frac{d^2e+de^2}{16}\kappa^3 + O(d,e)^4,\\ \varphi &= \frac{d+e}{2}\kappa - \frac{d^2-e^2}{6}\kappa' + \frac{d^3+e^3}{48}(\kappa^3 - \frac{2}{3}\kappa\tau^2 + 2\kappa'')\\ &\quad - \frac{d^2e+de^2}{36}\kappa\tau^2 + O(d,e)^4,\end{aligned}$$

and for $d = e$

$$\varphi \overset{d=e}{=} e\kappa + \frac{e^3}{24}(2\kappa'' + \kappa^3 - 2\kappa\tau^2) + O(e^5).$$

We can also compute the normalized binormals at $P_0$, $P_{-1}$, and $P_1$ by

$$\begin{aligned}\mathbf{b}_0 := \frac{\mathbf{d}\times\mathbf{e}}{||\mathbf{d}\times\mathbf{e}||} = \frac{\frac{\mathbf{d}}{d}\times\frac{\mathbf{e}}{e}}{\sin\varphi} &= \mathbf{t}\Big(\frac{de}{6}\kappa\tau + O(d,e)^3\Big)\\ &\quad + \mathbf{n}\Big(\frac{e-d}{3}\tau + \frac{d^2-de+e^2}{12}\tau'\\ &\qquad\qquad + \frac{d^2+de+e^2}{18}\frac{\kappa'}{\kappa}\tau + O(d,e)^3\Big)\\ &\quad + \mathbf{b}\Big(1 - \frac{(d-e)^2}{18}\tau^2 + O(d,e)^3\Big),\\ \mathbf{b}_{-1} := \frac{\mathbf{c}\times\mathbf{d}}{||\mathbf{c}\times\mathbf{d}||} &= \mathbf{t}\Big(-\frac{cd+d^2}{6}\kappa\tau + O(c,d)^3\Big)\\ &\quad + \mathbf{n}\Big(-\frac{c+2d}{3}\tau + \frac{c^2+3cd+3d^2}{12}\tau'\\ &\qquad\qquad + \frac{c^2+cd+d^2}{18}\frac{\kappa'}{\kappa}\tau + O(c,d)^3\Big)\\ &\quad + \mathbf{b}\Big(1 - \frac{c^2+4cd+4d^2}{18}\tau^2 + O(c,d)^3\Big),\end{aligned}$$

and

$$\begin{aligned}
\mathbf{b}_1 := \frac{\mathbf{e}\times\mathbf{f}}{\|\mathbf{e}\times\mathbf{f}\|} = {} & \mathbf{t}\Big(-\frac{fe+e^2}{6}\kappa\tau + O(e,f)^3\Big) \\
& + \mathbf{n}\Big(\frac{f+2e}{3}\tau + \frac{f^2+3fe+3e^2}{12}\tau' \\
& \qquad + \frac{f^2+fe+e^2}{18}\frac{\kappa'}{\kappa}\tau + O(e,f)^3\Big) \\
& + \mathbf{b}\Big(1 - \frac{f^2+4fe+4e^2}{18}\tau^2 + O(e,f)^3\Big).
\end{aligned}$$

With those, in turn, we can estimate the angle between two consecutive binormals as

$$\begin{aligned}
\hat{\eta}_{\mathbf{e}} = \langle \mathbf{b}_1 \times \mathbf{b}_0, \tilde{\mathbf{t}} \rangle = {} & \frac{d+e+f}{3}\tau - \frac{d^2+de-ef-f^2}{18}\frac{\kappa'}{\kappa}\tau \\
& - \frac{d^2-de-2e^2-3ef-f^2}{12}\tau' + O(d,e,f)^3
\end{aligned}$$

where $\tilde{\mathbf{t}}$ is the tangent approximation from equation (8.1).

# Appendix C

# Taylor Series Expansion of Geodesics

Given an arbitrary, smooth surface curve $\mathbf{r}(s)$, parameterized by arc length $s$, we can analyze it using Taylor series along with the so called Darboux frame $(\mathbf{t}(s), \mathbf{v}(s), \mathbf{n}(s))$ where $\mathbf{t}(s)$ is the unit tangent vector of the curve, $\mathbf{n}(s)$ is the unit surface normal, and $\mathbf{v}(s) := \mathbf{n}(s) \times \mathbf{t}(s)$ is defined by the cross product. The behavior of the Darboux frame along the curve is governed by the following modified Frenet-Serret equations [dC76, Koe90]

$$\begin{pmatrix} \mathrm{d}\mathbf{t}/\,\mathrm{d}s \\ \mathrm{d}\mathbf{v}/\,\mathrm{d}s \\ \mathrm{d}\mathbf{n}/\,\mathrm{d}s \end{pmatrix} = \begin{pmatrix} 0 & \kappa_g & \kappa_n \\ -\kappa_g & 0 & \tau \\ -\kappa_n & -\tau & 0 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{v} \\ \mathbf{n} \end{pmatrix}$$

where $\kappa_n$ is the normal curvature, $\kappa_g$ is the geodesic curvature, and $\tau$ is the geodesic torsion of the curve. (We omit the position $s$, since the equations hold for all (fixed) $s$ and we are interested only in the case $s = 0$, anyway.) Thus for the geodesic $\mathbf{g}_i(s)$ with $\mathbf{g}_i(0) = P$ and $\mathbf{g}_i(s_1) = Q_i$ as defined in Section 9.3, compare also Figure 9.1 (middle), we have

$$\frac{\mathrm{d}\mathbf{t}_i}{\mathrm{d}s} = \kappa_i \mathbf{n}, \qquad \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}s} = \tau_i \mathbf{n}, \qquad \frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} = -\kappa_i \mathbf{t}_i - \tau_i \mathbf{v}_i,$$

where $\kappa_i(s)$ and $\tau_i(s)$ are the normal curvature and geodesic torsion of $\mathbf{g}_i(s)$, respectively.

Note that these are exactly the same equations as for the space curves given in (B.1), only with a slightly different meaning of $\kappa_i$ and $\tau_i$ compared to $\kappa$ and $\tau$. Therefore, we have (formally) exactly the same Taylor series expansions for $\mathbf{a}_i = \overrightarrow{PQ_i} = \mathbf{g}_i(s_1) - \mathbf{g}_i(0)$ and $a_i := \|\mathbf{a}_i\|$ as for $\mathbf{e}$ and $e$ in Appendix B. Finally, we arrive at

$$\begin{aligned}\frac{\mathbf{a}_i}{a_i} =& \mathbf{t}_i\Big(1 - \frac{a_i^2}{8}\kappa_i^2 + O(a_i^3)\Big)\\ &+\mathbf{v}_i\Big(-\frac{a_i^2}{6}\kappa_i\tau_i + O(a_i^3)\Big)\\ &+\mathbf{n}\Big(\frac{a_i}{2}\kappa_i + \frac{a_i^2}{6}\kappa_i' + O(a_i^3)\Big).\end{aligned}$$

Let $\beta_i$ be the angle between $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$ (indices taken modulo $n$), see Figure 9.1 on the left hand side. Now we can compute the normal of an incident triangle at $P$ as

$$\begin{aligned}\frac{\mathbf{a}_i}{a_i} \times \frac{\mathbf{a}_{i+1}}{a_{i+1}} =& \sin\beta_i\mathbf{n}\Big(1 - \frac{a_i^2}{8}\kappa_i^2 - \frac{a_{i+1}^2}{8}\kappa_{i+1}^2 + O(a_i, a_{i+1})^3\Big)\\ &+ \cos\beta_i\mathbf{n}\Big(-\frac{a_{i+1}^2}{6}\kappa_{i+1}\tau_{i+1} + O(a_i, a_{i+1})^3\Big)\\ &- \mathbf{v}_i\Big(\frac{a_{i+1}}{2}\kappa_{i+1} + \frac{a_{i+1}^2}{6}\kappa_{i+1}' + O(a_i, a_{i+1})^3\Big)\\ &- \cos\beta_i\mathbf{n}\Big(-\frac{a_i^2}{6}\kappa_i\tau_i + O(a_i, a_{i+1})^3\Big)\\ &+ \sin\beta_i\mathbf{n}\Big(O(a_i, a_{i+1})^3\Big)\\ &+ \mathbf{t}_i\Big(O(a_i, a_{i+1})^3\Big)\\ &+ \mathbf{v}_{i+1}\Big(\frac{a_i}{2}\kappa_i + \frac{a_i^2}{6}\kappa_i' + O(a_i, a_{i+1})^3\Big)\\ &- \mathbf{t}_{i+1}\Big(O(a_i, a_{i+1})^3\Big).\end{aligned}$$

The norm of this vector

$$\begin{aligned}\Big\|\frac{\mathbf{a}_i}{a_i} \times \frac{\mathbf{a}_{i+1}}{a_{i+1}}\Big\| =& \sin\beta_i + (\frac{1}{\sin\beta_i} - \sin\beta_i)\Big(\frac{a_i^2}{8}\kappa_i^2 + \frac{a_{i+1}^2}{8}\kappa_{i+1}^2\Big)\\ &+ \cos\beta_i\Big(\frac{a_i^2}{6}\kappa_i\tau_i - \frac{a_{i+1}^2}{6}\kappa_{i+1}\tau_{i+1}\Big) - \cot\beta_i\Big(\frac{a_i a_{i+1}}{4}\kappa_i\kappa_{i+1}\Big) + O(a_i, a_{i+1})^3\end{aligned}$$

equals the sine of the angle $\alpha_i$ between $\mathbf{a}_i$ and $\mathbf{a}_{i+1}$ and we conclude $\sin\alpha_i = \sin\beta_i + O(a_i, a_{i+1})^2$ and $\alpha_i = \beta_i + O(a_i, a_{i+1})^2$ (compare also [MD02]).

But we can also use this result to compute the face normal $\frac{\mathbf{a}_i\times\mathbf{a}_{i+1}}{\|\mathbf{a}_i\times\mathbf{a}_{i+1}\|} = (\frac{\mathbf{a}_i}{a_i} \times \frac{\mathbf{a}_{i+1}}{a_{i+1}})/\sin\alpha_i$ (as done in Section 9.3.1). Now suppose $a_i = a_{i+m}$ and $\alpha_i = \alpha_{i+m}$. We know

$$2\sum_{j=i}^{i+m-1}\alpha_j = \sum_{j=1}^{2m}\alpha_j = \sum_{j=1}^{2m}\beta_j + O(a_j)^2 = 2\pi + O(a_j)_j^2.$$

Therefore, we get

$$\beta_{i,i+m} := \sum_{j=i}^{i+m-1} \beta_i = \sum_{j=i}^{i+m-1} \alpha_i + O(a_j)^2 = \pi + O(a_j)_j^2,$$

$$\mathbf{v}_{i+m} = \cos\beta_{i,i+m}\mathbf{v}_i - \sin\beta_{i,i+m}\mathbf{t}_i = -\mathbf{v}_i(1 + O(a_j)_j^2) + \mathbf{t}_i(O(a_j)_j^2),$$

$$\text{and } \mathbf{t}_{i+m} = \cos\beta_{i,i+m}\mathbf{t}_i + \sin\beta_{i,i+m}\mathbf{v}_i = -\mathbf{t}_i(1 + O(a_j)_j^2) + \mathbf{v}_i(O(a_j)_j^2),$$

where $\beta_{ij}$ is the counter-clockwise angle between $\mathbf{t}_i$ and $\mathbf{t}_j$. Using Euler's formula

$$\kappa = \kappa_{\max}\cos^2\theta + \kappa_{\min}\sin^2\theta$$

for curvature and the corresponding

$$\tau = (\kappa_{\max} - \kappa_{\min})\cos\theta\sin\theta$$

for torsion, we obtain

$$\kappa_{i+m} = \kappa_i + O(a_j)_j^2 \quad \text{and} \quad \tau_{i+m} = \tau_i + O(a_j)_j^2.$$

We can also use our computations to gain an estimation of the angular error of the triangle normal $\mathbf{n}_i := \frac{\mathbf{a}_i \times \mathbf{a}_{i+1}}{\|\mathbf{a}_i \times \mathbf{a}_{i+1}\|}$. We compute the scalar product with the normal $\mathbf{n}$ at $P$ and obtain, up to a third order term,

$$\langle \mathbf{n}_i, \mathbf{n} \rangle = 1 - \frac{1}{8\sin^2\alpha_i}(a_i^2\kappa_i^2 + a_{i+1}^2\kappa_{i+1}^2 - 2a_i a_{i+1}\kappa_i\kappa_{i+1}\cos\alpha_i).$$

This is the cosine of the angle. Therefore, the angular error has to be

$$\frac{1}{2\sin\alpha_i} \cdot \sqrt{a_i^2\kappa_i^2 + a_{i+1}^2\kappa_{i+1}^2 - 2a_i a_{i+1}\kappa_i\kappa_{i+1}\cos\alpha_i}$$

up to a second order error. Because the curvature is unknown anyway, we simplify the term by setting $\kappa_i = \kappa_{i+1} = 1$, and we get

$$\frac{1}{2\sin\alpha_i}\sqrt{a_i^2 + a_{i+1}^2 - 2a_i a_{i+1}\cos\alpha_i}$$

as an approximated error term. We note that the square root equals the length of the edge $\mathbf{d}_i$ opposing $P$, see Figure 9.4 (left).

# Appendix D

# Euler's Elastica for Space Curves

In this section, we will derive necessary conditions for a space curve $\mathbf{r}(s)$ to be an elastica, that means

$$\int \kappa^2 ds \longrightarrow \min$$

while fixing positions and tangents of the two end points. Thereby, we follow the treatment given in [BAYY99] and [Mum94] for elastica in the plane. Nevertheless, the situation for elastica in the three-dimensional space is more complex since torsion has to be considered as well. Therefore, we have to derive the criteria for 3D elastica anew.

We consider a small perturbation of $\mathbf{r}(s)$

$$\hat{\mathbf{r}}(s) := \mathbf{r}(s) + \epsilon(h(s)\mathbf{n} + k(s)\mathbf{b})$$

where $\mathbf{r}(s)$ is an elastica parameterized by arc length $s$, $h(s)$ and $k(s)$ are real functions with compact support, and $\epsilon$ is a real number. Using the Frenet equations (B.1) we get

$$\frac{d\hat{\mathbf{r}}}{ds} = \mathbf{t} + \epsilon(-h\kappa\mathbf{t} + (h' + k\tau)\mathbf{n} + (k' - h\tau)\mathbf{b}).$$

Let $\hat{\mathbf{r}}(\hat{s})$ be a parameterization of $\hat{\mathbf{r}}$ by arc length. Then

$$d\hat{s} = \left\|\frac{d\hat{\mathbf{r}}}{ds}\right\| ds = (1 - \epsilon h\kappa + O(\epsilon^2))ds.$$

Therefore, we have

$$\begin{aligned}
\hat{\mathbf{t}} &= \frac{d\hat{\mathbf{r}}}{d\hat{s}} = \frac{d\hat{\mathbf{r}}}{ds}\frac{ds}{d\hat{s}} = \mathbf{t} + \epsilon((h' + k\tau)\mathbf{n} + (k' - h\tau)\mathbf{b}) + O(\epsilon^2), \\
\hat{\kappa}\hat{\mathbf{n}} &= \frac{d\hat{\mathbf{t}}}{d\hat{s}} = \frac{d\hat{\mathbf{t}}}{ds}\frac{ds}{d\hat{s}} = \kappa\mathbf{n} + \epsilon(-h'\kappa - k\kappa\tau)\mathbf{t} + \epsilon(h(\kappa^2 - \tau^2) + h'' + k\tau' + 2k'\tau)\mathbf{n} \\
&\qquad + \epsilon(-h\tau' - 2h'\tau - k\tau^2 + k'')\mathbf{b} + O(\epsilon^2),
\end{aligned}$$

and

$$\hat{\kappa}^2 = ||\hat{\kappa}\hat{\mathbf{n}}||^2 = \kappa^2 + 2\epsilon\kappa(h(\kappa^2 - \tau^2) + h'' + k\tau' + 2k'\tau) + O(\epsilon^2).$$

Now we can compute, using integration by parts:

$$\begin{aligned}
\int \hat{\kappa}^2 d\hat{s} &= \int \kappa^2 d\hat{s} + \epsilon \int h(2\kappa^3 + 2\kappa'' - 2\kappa\tau^2) - 2k(\kappa\tau' + 2\kappa'\tau)d\hat{s} + O(\epsilon^2) \\
&= \int \kappa^2 ds + \epsilon \int h(\kappa^3 + 2\kappa'' - 2\kappa\tau^2) - 2k(\kappa\tau' + 2\kappa'\tau)ds + O(\epsilon^2).
\end{aligned}$$

Because $h(s)$ and $k(s)$ are arbitrary functions with compact support and the integral $\int \kappa^2 ds$ is minimal for $\mathbf{r}(s)$, this shows:

$$\kappa'' + \frac{\kappa^3}{2} - \kappa\tau^2 = 0 \quad \text{and} \quad \kappa'\tau + \frac{\kappa\tau'}{2} = 0,$$

which is a generalization of the conditions for 2D elastica.

# Bibliography

[ABFH08] ALBRECHT G., BÉCAR J.-P., FARIN G., HANSFORD D.: On the approximation order of tangent estimators. *Computer Aided Geometric Design 25*, 2 (Feb. 2008), 80–95.

[ABS02] ANOSHKINA E. V., BELYAEV A. G., SEIDEL H.-P.: Asymptotic analysis of three-point approximations of vertex normals and curvatures. In *Vision, Modeling, and Visualization 2002* (Erlangen, Germany, Nov. 2002), Greiner G., Niemann H., Ertl T., Girod B., Seidel H.-P., (Eds.), pp. 211–216.

[ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. *ACM Transactions on Graphics 22*, 3 (2003), 485–493. (SIGGRAPH 2003).

[Alf89] ALFELD P.: Scattered data interpolation in three or more variables. In *Mathematical Methods in Computer Aided Geometric Design*, Lyche T., Schumaker L. L., (Eds.). Academic Press Professional, Inc., San Diego, CA, USA, 1989, pp. 1–34.

[ANS96] ALFELD P., NEAMTU M., SCHUMAKER L. L.: Bernstein-Bézier polynomials on spheres and sphere-like surfaces. *Computer Aided Geometric Design 13*, 4 (1996), 333–349.

[AO06] ARROYO M., ORTIZ M.: Local *maximum-entropy* approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering 65*, 13 (2006), 2167–2202.

[BA76] BABUŠKA I., AZIZ A. K.: On the angle condition in the finite element method. *SIAM Journal of Numerical Analysis 13*, 2 (Apr. 1976), 214–226.

[BAYY99] BELYAEV A. G., ANOSHKINA E. V., YOSHIZAWA S., YANO M.: Polygonal curve evolutions for planar shape modeling and analysis. *International Journal of Shape Modeling 5*, 2 (1999), 195–217.

[BBU06] BOBACH T., BERTRAM M., UMLAUF G.: Issues and implementation of $C^1$ and $C^2$ natural neighbor interpolation. In *Advances in Visual Computing, Part II* (Lake Tahoe, NV, USA, Nov. 2006), Bebis G., Boyle R., Parvin B., Koracin D., Remagnino P., Nefian A. V., Gopi M., Pascucci V., Zara J., Molineros J., Theisel H., Malzbender T., (Eds.), vol. 4292 of *Lecture Notes in Computer Science*, Springer, pp. 186–195.

[Bel06] BELYAEV A.: On transfinite barycentric coordinates. In Sheffer and Polthier [SP06], pp. 89–99.

[Béz68] BÉZIER P.: *How Renault uses numerical control for car body design and tooling*. Paper SAE 680010, Society of Automotive Engineers Congress, Detroit, Mich., 1968.

[Béz78] BÉZIER P.: General distortion of an ensemble of biparametric surfaces. *Computer-Aided Design 10*, 2 (Mar. 1978), 116–120.

[BF01] BUSS S. R., FILLMORE J. P.: Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics 20*, 2 (2001), 95–126.

[BG07] BELYAEV A., GARLAND M. (Eds.):. *SGP07: Eurographics Symposium on Geometry Processing* (Barcelona, Spain, 2007), Eurographics Association.

[BIK*97] BELIKOV V. V., IVANOV V. D., KONTOROVICH V. K., KORYTNIK S. A., SEMENOV A. Y.: The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics 37*, 1 (1997), 9–15.

[Bla56] BLASCHKE W.: *Kreis und Kugel*, second ed. de Gruyter, 1956.

[Bou00] BOUTIN M.: Numerically invariant signature curves. *International Journal of Computer Vision 40*, 3 (2000), 235–248.

[BSW08] BELKIN M., SUN J., WANG Y.: Discrete Laplace operator on meshed surfaces. In *24th Annual Symposium on Computational Geometry (SOCG)*. 2008.

[BW92] BROWN J. L., WORSEY A. J.: Problems with defining barycentric coordinates for the sphere. *Mathematical Modelling and Numerical Analysis 26* (1992), 37–49.

[CFL82] CHRIST N. H., FRIEDBERG R., LEE T. D.: Weights of links and plaquettes in a random lattice. *Nuclear Physics B 210*, 3 (Oct. 1982), 337–346.

[CG84] CHARROT P., GREGORY J. A.: A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design 1*, 1 (1984), 87–94.

[CMP04] CERDA E., MAHADEVAN L., PASINI J. M.: The elements of draping. *Proceedings of the National Academy of Sciences of the United States of America 101*, 7 (February 2004), 1806–1810.

[CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aachen, Germany, 2003), Eurographics Association, pp. 177–187.

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted Delaunay triangulations and normal cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry* (2003), ACM Press, pp. 312–321.

[dC59] DE CASTELJAU P.: *Outillage méthodes calcul*. Paris: André Citroën Automobiles S.A., 1959.

[dC76] DO CARMO M. P.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976. 503 pages.

[DF08] DYKEN C., FLOATER M. S.: Transfinite mean value interpolation. *Computer Aided Geometric Design* (2008). In Press.

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. *ACM Transactions on Graphics 22*, 3 (July 2003), 848–855. (SIGGRAPH 2003).

[DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum 21* (2002), 209–218.

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. *Computer Graphics (Proceedings of SIGGRAPH 99)* (1999), 317–324.

[DP05] DESBRUN M., POTTMANN H. (Eds.):. *Third Eurographics Symposium on Geometry Processing* (Vienna, Austria, July 2005), Eurographics Association.

[Duf59] DUFFIN R. J.: Distributed and lumped networks. *Indiana University Mathematics Journal 8*, 5 (1959), 793–826.

[DW06] DIDAS S., WEICKERT J.: From adaptive averaging to accelerated nonlinear diffusion filtering. In *Pattern Recognition* (Berlin, Germany, Sept. 2006), K. Franke et al., (Ed.), vol. 4174 of *Lecture Notes in Computer Science*, Springer, pp. 101–110.

[Dzi88] DZIUK G.: Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial differential equations and calculus of variations*, Hildebrandt S., Leis R., (Eds.), vol. 1357 of *Lecture Notes in Mathematics*. Springer, Berlin Heidelberg, 1988, pp. 142–155.

[Eul44] EULER L.: Additamentum 'De Curvis Elasticis'. In *Methodus Inveniendi Lineas Curvas Maximi Minimive Probprietate Gaudentes* (Lausanne, 1744).

[Far86] FARWIG R.: Rate of convergence of Shepard's global interpolation formula. *Mathematics of Computation 46*, 174 (1986), 577–590.

[Far90] FARIN G.: Surfaces over Dirichlet tessellations. *Computer Aided Geometric Design 7*, 1–4 (1990), 281–292.

[FHK06] FLOATER M. S., HORMANN K., KÓS G.: A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics 24*, 1–4 (Jan. 2006), 311–331.

[FKR05] FLOATER M. S., KÓS G., REIMERS M.: Mean value coordinates in 3D. *Computer Aided Geometric Design 22*, 7 (Oct. 2005), 623–631. Special Issue on Geometric Modelling and Differential Geometry.

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

[For72] FORREST A. R.: Interactive interpolation and approximation by Bézier polynomials. *The Computer Journal 15*, 1 (1972), 71–79.

[GG01] GOOCH B., GOOCH A.: *Non-Photorealistic Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[GGRZ06] GRINSPUN E., GINGOLD Y., REISMAN J., ZORIN D.: Computing discrete shape operators on general meshes. *Computer Graphics Forum 25*, 3 (2006), 547–556.

[GI04] GOLDFEATHER J., INTERRANTE V.: A novel cubic-order algorithm for approximating principal direction vectors. *ACM Transactions on Graphics 23*, 1 (2004), 45–63.

[Gol02] GOLDMAN R.: *Pyramid Algorithms: A Dynamic Programming Approach to Curves and Surfaces for Geometric Modeling*. Morgan Kaufmann, San Francisco, CA, 2002.

[Gol04] GOLDMAN R.: Multisided arrays of control points for multisided Bézier patches. *Computer Aided Geometric Design 21*, 3 (Mar. 2004), 243–261.

[Gol05] GOLDMAN R.: Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design 22*, 7 (Oct. 2005), 632–658.

[Gou71] GOURAUD H.: Continuous shading of curved surfaces. *IEEE Transactions on Computers C-20*, 6 (1971), 623–629.

[Grü67] GRÜNBAUM B.: *Convex Polytopes*, vol. XVI of *Pure and Applied Mathematics*. Interscience, London, 1967.

[HAvL07] HEIN M., AUDIBERT J.-Y., VON LUXBURG U.: Graph Laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research 8* (2007), 1325–1370.

[HF06] HORMANN K., FLOATER M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics 25*, 4 (Oct. 2006), 1424–1441.

[Hor83] HORN B. K. P.: The curve of least energy. *ACM Transactions on Mathematical Software 9* (1983), 441–460.

[HP04] HILDEBRANDT K., POLTHIER K.: Anisotropic filtering of non-linear surface features. *Computer Graphics Forum 23*, 3 (2004), 391–400.

[HPW05] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: Smooth feature lines on surface meshes. In Desbrun and Pottmann [DP05], pp. 85–90.

[HPW06] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: On the convergence of metric and geometric properties of polyhedral surfaces. *Geometriae Dedicata 123* (Dec. 2006), 89–112.

[HS00] HIYOSHI H., SUGIHARA K.: Voronoi-based interpolation with higher continuity. In *SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry* (New York, NY, USA, 2000), ACM Press, pp. 242–250.

[HS03] HAMEIRI E., SHIMSHONI I.: Estimating the principal curvatures and the Darboux frame from real 3-d range data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 33*, 4 (2003), 626–637.

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *ACM Transactions on Graphics 25*, 3 (2006), 1126–1134.

[HT04] HORMANN K., TARINI M.: A quadrilateral rendering primitive. In *HWWS '04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware* (Grenoble, France, Aug. 2004), Akenine-Möller T., McCool M., (Eds.), Eurographics Symposium Proceedings, ACM, pp. 7–14.

[Hui91] HUISKAMP G.: Difference formulas for the surface Laplacian on a triangulated surface. *Journal of Computational Physics 95*, 2 (1991), 477–496.

[JLW07] JU T., LIEPA P., WARREN J.: A general geometric construction of coordinates in a convex simplicial polytope. *Computer Aided Geometric Design 24*, 3 (Apr. 2007), 161–178.

[JMD*07] JOSHI P., MEYER M., DEROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *ACM Transactions on Graphics 26*, 3 (2007), 71.

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics 24*, 3 (2005), 561–566.

[JSWD05] JU T., SCHAEFER S., WARREN J., DESBRUN M.: A geometric construction of coordinates for convex polyhedra using polar duals. In Desbrun and Pottmann [DP05], pp. 181–186.

[Koe90] KOENDERINK J. J.: *Solid Shape*. MIT Press, 1990.

[Kre59] KREYSZIG E.: *Differential Geometry*. University of Toronto Press, 1959.

[LBS05a] LANGER T., BELYAEV A., SEIDEL H.-P.: Exact and approximate quadratures for curvature tensor estimation. In *Vision, Modeling, and Visualization 2005*, Greiner G., Hornegger J., Niemann H., Stamminger M., (Eds.). Akademische Verlagsgesellschaft Aka GmbH, Berlin, Nov. 2005, pp. 421–428.

[LBS05b] LANGER T., BELYAEV A. G., SEIDEL H.-P.: Asymptotic analysis of discrete normals and curvatures of polylines. In *Spring Conference on Computer Graphics SCCG 2005* (Bratislava, May 2005), Jüttler B., (Ed.), Comenius University, pp. 221–224.

[LBS06] LANGER T., BELYAEV A., SEIDEL H.-P.: Spherical barycentric coordinates. In Sheffer and Polthier [SP06], pp. 81–88.

[LBS07a] LANGER T., BELYAEV A., SEIDEL H.-P.: Exact and interpolatory quadratures for curvature tensor estimation. *Computer Aided Geometric Design 24*, 8–9 (Nov. 2007), 443–463. Special Issue on Discrete Differential Geometry.

[LBS07b] LANGER T., BELYAEV A., SEIDEL H.-P.: Mean value coordinates for arbitrary spherical polygons and polyhedra in $\mathbb{R}^3$. In *Curve and Surface Design: Avignon 2006* (2007), Chenin P., Lyche T., Schumaker L. L., (Eds.), Modern Methods in Mathematics, Nashboro Press, pp. 193–202.

[LBS08] LANGER T., BELYAEV A., SEIDEL H.-P.: Mean value Bézier maps. In *Advances in Geometric Modeling and Processing* (Hangzhou, China, Apr. 2008), Chen F., Jüttler B., (Eds.), vol. 4975 of *Lecture Notes in Computer Science*, Springer, pp. 231–243.

[LD89] LOOP C. T., DEROSE T. D.: A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics 8*, 3 (1989), 204–234.

[LD90] LOOP C., DEROSE T. D.: Generalized B-spline surfaces of arbitrary topology. In *SIGGRAPH '90: Proceedings of the 17th annual* *conference on Computer graphics and interactive techniques* (New York, NY, USA, 1990), ACM, pp. 347–356.

[LKCOL07] LIPMAN Y., KOPF J., COHEN-OR D., LEVIN D.: GPU-assisted positive mean value coordinates for mesh deformations. In Belyaev and Garland [BG07], pp. 117–123.

[LLCO08] LIPMAN Y., LEVIN D., COHEN-OR D.: Green coordinates. Accepted for SIGGRAPH 2008.

[LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics 25*, 3 (2006), 681–689.

[LS07] LANGER T., SEIDEL H.-P.: Mean value Bézier surfaces. In *Mathematics of Surfaces XII* (Sheffield, England, Sept. 2007), Martin R., Sabin M., Winkler J., (Eds.), vol. 4647 of *Lecture Notes in Computer Science*, Springer, pp. 263–274.

[LS08] LANGER T., SEIDEL H.-P.: Higher order barycentric coordinates. In *Eurographics 2008* (Crete, Greece, 2008), Drettakis G., Scopigno R., (Eds.), The European Association for Computer Graphics (Eurographics), Foundation for Research and Technology - Hellas (FORTH), pp. 459–466.

[Mar98] MARTIN R. R.: Estimation of principal curvatures from range data. *International Journal of Shape Modeling 4*, 3&4 (1998), 99–109.

[Max99] MAX N.: Weights for computing vertex normals from facet normals. *Journal of Graphics Tools 4*, 2 (1999), 1–6.

[MD02] MALTRET J.-L., DANIEL M.: *Discrete curvatures and applications : a survey.* Rapport de recherche LSIS.RR.2002.002, Laboratoire des Sciences de l'Information et des Systèmes, 2002.

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, Heidelberg, 2003, pp. 35–57.

[MLBD02] MEYER M., LEE H., BARR A., DESBRUN M.: Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools 7*, 1 (2002), 13–22.

[Möb27] MÖBIUS A. F.: *Der barycentrische Calcul.* Johann Ambrosius Barth, Leipzig, 1827.

[Möb46] MÖBIUS A. F.: Ueber eine neue Behandlungsweise der analytischen Sphärik. In *Abhandlungen bei Begründung der Königl. Sächs. Gesellschaft der Wissenschaften.* Jablonowski Gesellschaft, Leipzig, 1846, pp. 45–86. Reappeared in *A. F. Möbius, Gesammelte Werke*, F. Klein (ed.), vol. 2, Leipzig, 1886, 1–54.

[MS92] MORETON H. P., SÉQUIN C. H.: Functional optimization for fair surface design. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), ACM, pp. 167–176.

[MT98] MEHLUM E., TARROU C.: Invariant smoothness measures for surfaces. *Advances in Computational Mathematics 8*, 1–2 (Mar. 1998), 49–63.

[MT04] MORVAN J.-M., THIBERT B.: Approximation of the normal vector field and the area of a smooth surface. *Discrete & Computational Geometry 32*, 3 (2004), 383–400.

[Mum94] MUMFORD D.: Elastica and computer vision. In *Algebraic Geometry and its Applications* (Springer, 1994), Bajaj C. L., (Ed.), pp. 491–506.

[MW00] MEEK D. S., WALTON D. J.: On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design 17* (2000), 521–543.

[NRDR05] NEHAB D., RUSINKIEWICZ S., DAVIS J., RAMAMOORTHI R.: Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics 24*, 3 (Aug. 2005), 536–543. (SIGGRAPH 2005).

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics 23* (2004), 609–612. (SIGGRAPH 2004).

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Communications of ACM 18*, 6 (1975), 311–317.

[PHYK05] POTTMANN H., HUANG Q.-X., YANG Y.-L., KÖLPL S.: *Integral invariants for robust geometry processing.* Tech. Rep. 146, Geometry Preprint Series, Vienna Univ. of Technology, November 2005.

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, 1 (1993), 15–36.

[PVTF02] PRESS W. H., VETTERLING W. T., TEUKOLSKY S. A., FLANNERY B. P.: *Numerical Recipes in C++ — The Art of Scientific Computing*. Cambridge University Press, 2002.

[PWHY08] POTTMANN H., WALLNER J., HUANG Q.-X., YANG Y.-L.: Integral invariants for robust geometry processing. *Computer Aided Geometric Design* (2008). In Press.

[Rat07] Ratatouille. Bird B., Pinkava J. (Dirs.), Pixar Animation Studios, 2007.

[Rus04] RUSINKIEWICZ S.: Estimating curvatures and their derivatives on triangle meshes. In *3DPVT '04: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission* (Thessaloniki, Greece, Sept. 2004), IEEE Computer Society, pp. 486–493.

[Rus07] RUSTAMOV R. M.: *Boundary Element Formulation of Harmonic Coordinates*. Tech. rep., Purdue University, 2007.

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Transactions on Graphics 23*, 3 (2004), 870–877.

[Sau70] SAUER R.: *Differenzengeometrie*. Springer, Berlin, 1970.

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *SGP 2004—Symposium on Geometry Processing* (Nice, France, July 2004), Scopigno R., Zorin D., (Eds.), pp. 175–184.

[She68] SHEPARD D.: A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (New York, NY, USA, 1968), ACM Press, pp. 517–524.

[Sib80] SIBSON R.: A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of Cambridge Philosophical Society 87* (1980), 151–155.

[SJW07] SCHAEFER S., JU T., WARREN J.: A unified, integral construction for coordinates over closed curves. *Computer Aided Geometric Design 24*, 8–9 (Nov. 2007), 481–493. Special Issue on Discrete Differential Geometry.

[SM06] SUKUMAR N., MALSCH E. A.: Recent advances in the construction of polygonal finite elements interpolants. *Archives of Computational Methods in Engineering 13*, 1 (2006), 129–163.

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1986), ACM Press, pp. 151–160.

[SP06] SHEFFER A., POLTHIER K. (Eds.):. *Fourth Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy, June 2006), Eurographics Association.

[ST04] SUKUMAR N., TABARRAEI A.: Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering 61*, 12 (2004), 2045–2066.

[Sug99] SUGIHARA K.: Surface interpolation based on new local coordinates. *Computer-Aided Design 31*, 1 (Jan. 1999), 51–58.

[Tau95a] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Procedings of the Fifth International Conference on Computer Vision* (1995), pp. 902–907.

[Tau95b] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH 95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 351–358.

[TRZS04] THEISEL H., RÖSSL C., ZAYER R., SEIDEL H.-P.: Normal based estimation of the curvature tensor for triangular meshes. In *12th Pacific Conference on Computer Graphics and Applications, PG 2004* (Seoul, South Korea, Oct. 2004), Cohen-Or D., Ko H.-S., Terzopoulos D., Warren J., (Eds.), IEEE, pp. 288–297.

[Wac75] WACHSPRESS E. L.: *A Rational Finite Element Basis*, vol. 114 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.

[War96] WARREN J.: Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics 6*, 2 (1996), 97–108.

[WB01] WATANABE K., BELYAEV A. G.: Detection of salient curvature features on polygonal surfaces. *Computer Graphics Forum 20*, 3 (Sept. 2001), 385–392. (Eurographics 2001).

[WBG07] WICKE M., BOTSCH M., GROSS M.: A finite element method on convex polyhedra. *Computer Graphics Forum 26*, 3 (2007), 355–364.

[Wei94] WEICKERT J.: Anisotropic diffusion filters for image processing based quality control. In *Proc. Seventh European Conf. on Mathematics in Industry* (Stuttgart, 1994), Teubner, pp. 355–362.

[WMKG07] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete Laplace operators: No free lunch. In Belyaev and Garland [BG07], pp. 33–37.

[WSHD07] WARREN J. D., SCHAEFER S., HIRANI A. N., DESBRUN M.: Barycentric coordinates for convex sets. *Advances in Computational Mathematics 27*, 3 (2007), 319–338.

[XB03] XU G., BAJAJ C. L.: Curvature computations of 2-manifolds in $\mathbb{R}^k$. *Journal of Computational Mathematics 21*, 5 (2003), 681–688.

[Xu04] XU G.: Convergence of discrete Laplace-Beltrami operators over surfaces. *Computers & Mathematics with Applications 48*, 3–4 (Aug. 2004), 347–360.

[Xu06a] XU G.: Convergence analysis of a discretization scheme for Gaussian curvature over triangular surfaces. *Computer Aided Geometric Design 23*, 2 (Feb. 2006), 193–207.

[Xu06b] XU G.: Discrete Laplace-Beltrami operator on sphere and optimal spherical triangulations. *International Journal of Computational Geometry and Applications 16*, 1 (2006), 75–93.

[Zay07] ZAYER R.: *Numerical and variational aspects of mesh parameterization and editing*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 2007.

Please note that the bibliography is sorted with respect to the abbreviated citation keys, not with respect to the full author names.