General discrete Laplace operators on polygonal meshes

Diplomarbeit

zur Erlangung des akademischen Grades
Diplominformatiker

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik

eingereicht von: Philipp Herholz
geboren am: 18.11.1985
in: Kiel

Gutachter: Prof. Dr. Marc Alexa
Prof. Dr. Peter Eisert

eingereicht am: ......
verteidigt am: ......

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Weiterhin erkläre ich, eine Diplomarbeit in diesem Studiengebiet erstmalig einzureichen.

Berlin, den December 5, 2012 . . . .

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Berlin, December 5, 2012 . . . .

Contents

1. Introduction **1**

1.1. Outline 2

1.2. Applications 2

2. Smooth Theory **5**

2.1. Surfaces 5

2.2. Integration over Surfaces 7

2.3. Differential Operators on Surfaces 7

2.3.1. The Gradient 8

2.3.2. The Divergence 9

2.3.3. The Laplacian 10

2.3.4. The Gauss Map, Curvature 11

3. Discrete Laplacians **15**

3.1. The Discrete Picture 15

3.2. Properties of Discrete Laplacians 16

3.3. Combinatorial Laplacians 17

3.4. Pinkall's Cotan Operator 18

3.5. Desbrun's Cotan Operator 21

3.6. The DEC Laplacian 22

3.7. Properties of the Cotan Operator 24

3.8. Weighted Cotan Operator 25

4. General Polygonal Meshes **28**

4.1. Orthogonal Duals for Polygonal Meshes 28

4.2. Alexa and Wardetzky's Polygon Laplacian 30

4.2.1. Implementation 33

4.2.2. Properties of the Polygon Laplacian 34

4.2.3. Weighting 36

4.2.4. Alternative Polygon Laplacian 38

A. Listings **43**

1. Introduction

The spectral approach to signal processing has led to a huge number of well known techniques, most prominently present in the area of analog and digital audio signal processing; every older child knows what it means to manipulate the high, low or mid frequencies at their audio player. Therefore it is a natural question to ask how one can formulate the notion of spectral processing on polygonal meshes. A polygonal mesh can be regarded as the discrete signal of the vertex coordinates together with connectivity information. In his seminal paper Taubin [Tau95] suggested to exploit the analogy between the second derivative for real valued functions over $\mathbb{R}$ and the so called Laplace-Beltrami operator on surfaces. This idea is motivated by the fact that spectral signal processing on periodic time-dependent discrete signals amounts to modifying the different frequency components of the signal. The frequency components are obtained by projecting the signal in a system of orthogonal sin and cos functions. Note, that $\sin(nx)$ and $\cos(nx)$ respectively are eigenfunctions of the second derivative, that is, they reproduce under the action of the second derivative up to a constant factor. The signal of mesh coordinates naturally lives on the surface itself. Therefore, it makes sense, in analogy to the 1-dimensional case, to project the coordinates onto the eigenfunctions of the second derivative of the surface, the Laplace-Beltrami operator. This analogy is one among many reason for the interest in the discretization of the Laplace-Beltrami operator (or Laplacian for short) on discrete surfaces. The Laplacian of a surface determines, to a certain extend, the topology and geometry of the surface itself. More specifically, the eigenvalues of the Laplacian encode properties of the surface like Betti numbers and area. On the other hand it captures the metric of a surface. The Laplacian is therefore a central tool for investigating how global geometry emerges from local geometry and has become one of the central elements in computational geometry processing. The academic world prefers to work with triangular meshes which admit easy theoretic treatment, the movie industry on the other hand generally prefers quad meshes for practical reasons. It is therefore an important area of research to find generalizations of well known constructions on triangular meshes to polygonal meshes.

1.1. Outline

This thesis deals with the discretization of the Laplace-Beltrami operator on meshes and investigates the generalization of common discretization schemes from triangular meshes to the general polygonal case. To understand the ideas used to discretize the Laplacian it is important to have a basic understanding of the Laplacian acting on smooth surfaces and its properties. Chapter 2 will give a brief introduction and motivates concepts used in the discrete theory. Chapter 3 and 4 are concerned with the discrete Laplacian on triangle and general polygonal meshes respectively. A popular way to discretize the Laplacian is by defining the discrete operator polygon-wise on a triangle mesh, the most prominent one being the cotan-operator, which has been discovered independently by Pinkall and Polthier [PP93] and Duffin [Duf59]. As it turns out many different approaches lead to the cotan-formula for the discrete Laplace-Beltrami operator. Alexa and Wardetzky [AW11] found a generalization of the cotan operator to general polygonal meshes sharing all the favorable properties of the cotan scheme. Unfortunately it also shares some flaws, resulting in unwanted behavior in certain applications. To overcome or at least alleviate these flaws is one of the main motivations for this thesis.

1.2. Applications

The rich set of mathematical properties of the Laplacian has led to a vast array of applications in the field of geometry processing. This section highlights some of them to illustrate the importance of discrete Laplace operators on polygonal meshes as introduced in section 3.1.

Mesh Filtering As described in the introduction, the eigendecomposition of the Laplacian can be used to process signals defined on meshes. The mesh coordinates themself can be interpreted as discrete signal over the mesh. Vallet and Lévy [VL08] explicitly filter the mesh by numerically computing a set of eigenvectors and modulating the frequency components (figure 1.1). The computation of eigenvectors is unfortunately very expensive. Even for moderate sized meshes and state of the art software interactive applications are not feasible. The visualization of the eigenfunctions (figure 1.1 below) illustrates the periodic nature of the eigenfunction. The spatial “frequency” increases with larger eigenvalues. Further details and applications can be found in the Siggraph course [LZ10].

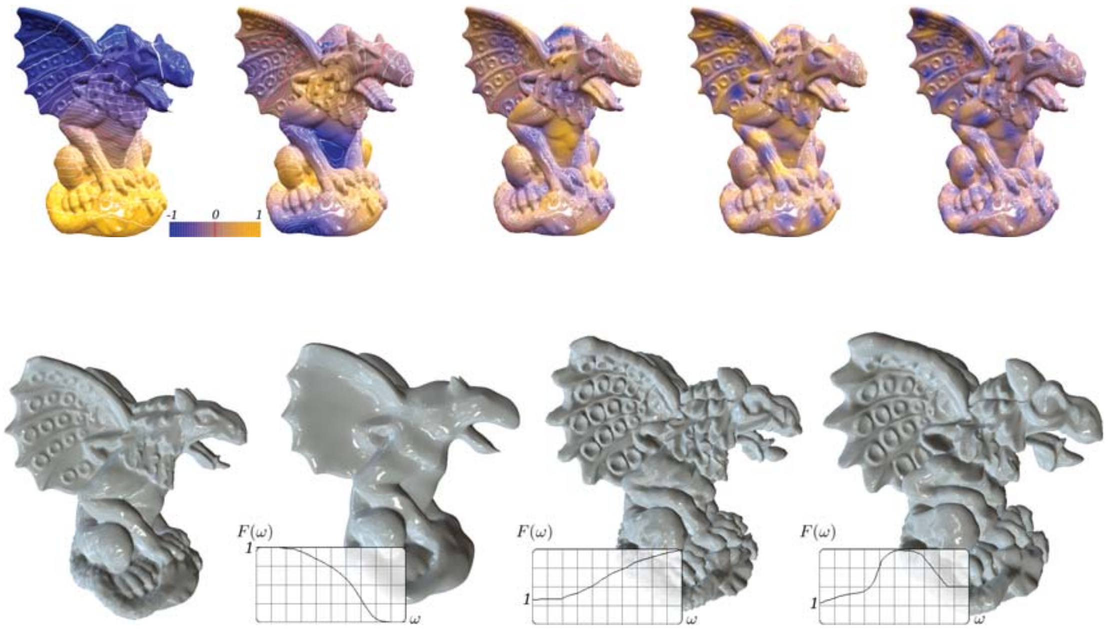

Figure (1.1): Eigenfunctions of the Laplacian on a mesh (above) and the result of increasing and attenuating specific frequency components of the mesh signal (below). Images taken from Vallet and Lévy [VL08].

Shape DNA The spectrum of the Laplacian encodes important information about the underlying surface. Kac [Kac66] posed the question whether it is possible to determine the shape of a 2-dimensional surface by the spectrum of its Laplacian. Since the eigenfunctions correspond to the natural vibration modes of the shape he formulated the question as “Can one hear the shape of a drum?”. The question was answered negative by Gordon et al. [Gor92] who found two simple isospectral domains. The spectrum gives nevertheless important information about the shape. Reuter et al. [Reu06] therefore use the spectrum as a “fingerprint” of a discrete surface. Applications of this method range from shape matching and watermarking to database queries. Jain and Zhang [JZ07] use the first eigenvectors of the Laplacian as coordinates of a spectral embedding to match articulated models (figure 1.2). They exploit the fact that this spectral embedding depends on distances measured on the surface rather than in the ambient space. The articulated models, even though not very similar in the spatial domain, become very similar in the spectral domain and can effectively be matched with standard methods.

Laplacian Coordinates The Laplacian applied to the coordinates of a mesh computes a vector representing the difference of a vertex and the average in a small neighborhood, as detailed later. These vectors per vertex are called *Laplacian coordinates* of the mesh. One can show that the geometry can be reconstructed from this representation up to a rigid

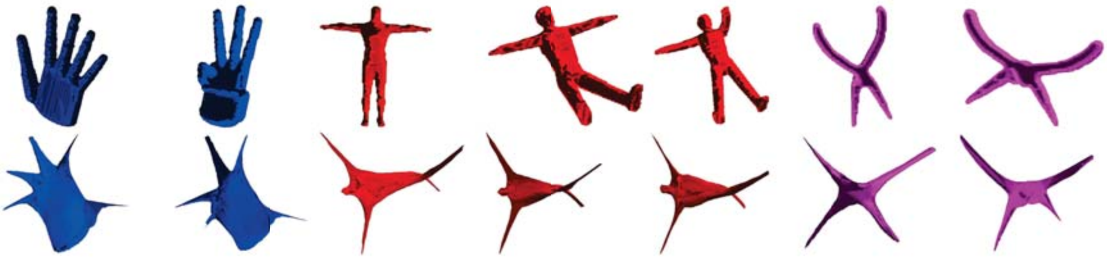

Figure (1.2): Spectral embeddings of articulated shapes. The coordinates of the spectral embeddings are given by the (eigenvalue normalized) eigenvectors corresponding to the lowest non-zero eigenvalues.

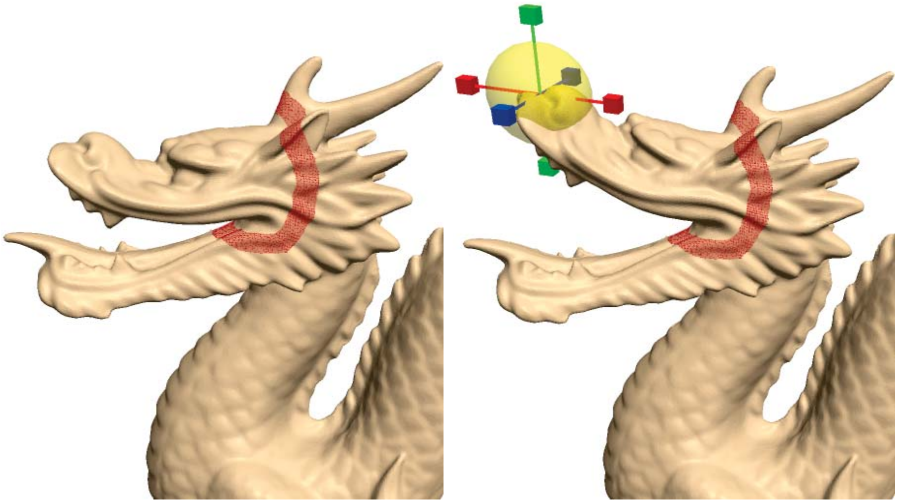

Figure (1.3): Constraining vertex positions (red) and a set of handle vertices (yellow sphere) yields a smooth deformed surface by solving a linear system.

transformation [Sor05]. By constraining the position of a set of vertices and solve for a mesh that has the same Laplacian coordinates in a least squares sense, the mesh can be smoothly deformed (figure 1.3).

2. Smooth Theory

Meshes are used to mimic the smooth theory of two-dimensional surfaces embedded in $\mathbb{R}^3$. Since meshes are inherently not smooth we cannot employ classical calculus methods on these surfaces. It is nevertheless very helpful to have a rough understanding of the smooth theory in order to develop analogous tools on meshes that reflect the behavior in the smooth setting.

2.1. Surfaces

Differential geometry is concerned with *manifolds*, smooth surfaces that look locally like the space $\mathbb{R}^n$. The earth for example can be thought of as a 2-dimensional manifold embedded in a 3-dimensional ambient space because it is locally flat. The surfaces we are interested in are called differentiable 2-manifolds. It is sufficient for this thesis to introduce a particular subset called regular surfaces.

Definition 1 (Regular surface)**.** *A regular 2-surface embedded into $\mathbb{R}^3$ is a subset $S \subset \mathbb{R}^3$. For each point $p \in S$ there is an open neighborhood $p \in V_p \subset \mathbb{R}^3$ and a map $x : U \to V_p \cap S$ of an open set $U \in \mathbb{R}^2$ onto $V_p \cap S \subset \mathbb{R}^3$ such that:*

1. *x is differentiable.*
2. *x is a homeomorphism.*
3. *For each $q \in U$ the differential $dx_q : \mathbb{R}^2 \to \mathbb{R}^3$ is one-to-one ($\Leftrightarrow$ has rank 2).*

The maps x_p are called *local parameterizations* or *local charts* of the surface. They identify subsets of $\mathbb{R}^2$ with subsets of S in a continuous and invertible fashion (property 2). Properties 1 and 3 ensure that the surface is smoothly embedded into the ambient space $\mathbb{R}^3$, that is, the surface will have no cusps. Are cube for example cannot be represented by a regular surface. The third property makes it possible to combine a set of parameterizations into an *atlas* spanning the whole surface. Such a (finite) covering always exists, since surfaces are compact subsets of $\mathbb{R}^n$. For a local parameterization $x(s,t)$ at a point

$x(q) \in S$ the differential dx_q is represented by the Jacobian of $x(s,t)$

$$dx_q = \begin{pmatrix} \partial_s x_1|_q & \partial_t x_1|_q \\ \partial_s x_2|_q & \partial_t x_2|_q \\ \partial_s x_3|_q & \partial_t x_3|_q \end{pmatrix}. \tag{2.1}$$

The Jacobian maps directions in the parameter domain to elements in the so called *tangent space* $T_{x(q)}S$ at the point $x(q)$.

Definition 2 (Tangent space)**.** *The tangent space* T_pS *to a regular surface* S *at a point* $x(q) = p \in S$ *with a parameterization* x *is spanned by the two vectors* $\tilde{e}_1 = dx_q(e_1)$ *and* $\tilde{e}_2 = dx_q(e_2)$*.*

Since the parameterizations have full rank Jacobians (property 3), the tangent space is 2-dimensional everywhere. The tangent space is the space of all directional derivatives of curves on the manifold passing through the point $x(p)$ at $x(p)$ [dCa76, 2-4, Prop. 1]. Given a curve $c(t) : (-\varepsilon, \varepsilon) \subset \mathbb{R} \to S$ with $c(0) = p$, the vector $\frac{\partial c(t)}{\partial t}\big|_0$ is part of the tangent space. This motivates the notation

$$\partial_i = \frac{\partial}{\partial e_i} = \tilde{e}_i. \tag{2.2}$$

The scalar product of the ambient space $\mathbb{R}^3$, restricted to the tangent space, is a bilinear form

$$g_p = \begin{cases} T_pS \times T_pS \to \mathbb{R} \\ u, v \mapsto u^T g_p v \end{cases} \tag{2.3}$$

and is represented in the basis $\tilde{e}_1, \tilde{e}_2$ by the matrix

$$g = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \langle \tilde{e_1}, \tilde{e_1} \rangle & \langle \tilde{e_1}, \tilde{e_2} \rangle \\ \langle \tilde{e_2}, \tilde{e_1} \rangle & \langle \tilde{e_2}, \tilde{e_2} \rangle \end{pmatrix} \tag{2.4}$$

The subscript p will be dropped for the rest of the chapter. This scalar product on the tangent space is also called *first fundamental form* and is commonly abbreviated by I. The scalar product g can be used to measure metric properties on the surface like length of curves and angles. Consider for example a smooth curve $c : [0,1] \subset \mathbb{R} \to S$ on the surface. The length of c on the surface is given by

$$\int_0^1 |c'(t)|_g \, dt = \int_0^1 c'(t)^T g_{c(t)} c'(t) \, dt \tag{2.5}$$

2.2. Integration over Surfaces

With the help of an atlas we can define (differentiable) functions on a surface S, $f : S \to \mathbb{R}$. The question arises how to integrate these functions. As noted before, one can measure the length of a curve on the manifold by integrating the length of derivatives over an interval $I \in \mathbb{R}$. The length of a vector on the surface is the norm of the vector in the parameter space with respect to the metric induced by g. The same is true for functions. The integral of a function on an open subset of $\mathbb{R}^2$ is the integral over the function weighted by the infinitesimal area element $dx_1 dx_2$. To integrate a function over a surface we have to deform the area element accordingly. The area spanned by e_1 and e_2 in parameter space is mapped to the area $|\tilde{e}_1 \times \tilde{e}_2|$ on the tangent space of the surface. The calculation

$$\begin{aligned} & \langle \tilde{e}_1, \tilde{e}_2 \rangle = |\tilde{e}_1||\tilde{e}_2| \cos \alpha \\ & |\tilde{e}_1 \times \tilde{e}_2| = |\tilde{e}_1||\tilde{e}_2| \sin \alpha \\ \Rightarrow \quad & \frac{\langle \tilde{e}_1, \tilde{e}_2 \rangle^2 + |\tilde{e}_1 \times \tilde{e}_2|^2}{|\tilde{e}_1|^2 |\tilde{e}_2|^2} = \cos^2 \alpha + \sin^2 \alpha = 1 \\ \Rightarrow \quad & |\tilde{e}_1 \times \tilde{e}_2| = \sqrt{|\tilde{e}_1|^2 |\tilde{e}_2|^2 - \langle \tilde{e}_1, \tilde{e}_2 \rangle^2} = \sqrt{EG - F^2} = \sqrt{\det g} \end{aligned}$$

shows, that the mapped area on the tangent space is related by $\sqrt{\det g}$ to an area element in the parameter domain. The integration over one coordinate chart is thus defined as.

$$\int_u \int_v f(u,v) \sqrt{\det g(u,v)} \; dv \; du \tag{2.6}$$

One can show that this definition of integral is independent of the parameterization and can be extended to a whole atlas. A proof using a partition of unity argument can be found in [dCa94, Chapter 4]. Note, that we can measure surface area by integrating the function $f(x,y) = 1$. Moreover, it is possible to define an inner product on the space of square-integrable functions on the surface by

$$\langle f, g \rangle = \int_u \int_v f(u,v) g(u,v) \sqrt{\det g} \; dv \; du. \tag{2.7}$$

2.3. Differential Operators on Surfaces

Vector calculus is concerned with the generalization of the calculus on functions over $\mathbb{R}$ to functions over $\mathbb{R}^n$. For a function $f : \mathbb{R}^n \to \mathbb{R}$ one can define operators known as *gradient* and *divergence*. They can be used to define the Laplacian. We are interested in further generalizations of these concepts to surfaces in order to define the Laplace-Beltrami operator. This section gives some insight into the formulations of differential operators on surfaces in local coordinates.

2.3.1. The Gradient

The gradient maps functions $f : \mathbb{R}^n \to \mathbb{R}$ to vector fields $\mathbb{R}^n \to \mathbb{R}^n$ representing the direction of largest increase in the function f

$$\operatorname{grad} f = \nabla f = \left(\tfrac{\partial f}{\partial x_1}, \quad \dots \quad , \tfrac{\partial f}{\partial x_n} \right). \tag{2.8}$$

The gradient is characterized by the relation

$$\langle \operatorname{grad}\ f, v \rangle = df(v), \quad \forall v \in \mathbb{R}^3 \tag{2.9}$$

Note, that $\mathbb{R}^n$ is the tangent space for every point in $\mathbb{R}^n$. To generalize this expression to surfaces the space $\mathbb{R}^n$ is replaced by the tangent space of the surface.

$$\langle \operatorname{grad}_p\ f, v \rangle = (df)_p(v), \quad \forall v \in T_p S \tag{2.10}$$

In analogy to $\mathbb{R}^n$, the differential of a function on a surface maps elements of the tangent space of S at p to the directional derivative. Therefore we have in local coordinates

$$\left. \frac{\partial f}{\partial \tilde{e}_j} \right|_p = \langle \operatorname{grad}_p, \tilde{e}_j \rangle. \tag{2.11}$$

The gradient of a function at a point $p \in S$ is an element of the tangent space and can therefore be represented by

$$\operatorname{grad}_p f = \sum_i a_i \tilde{e}_i. \tag{2.12}$$

The coefficients a_i depend on p and describe a vector field on the surface. By plugging 2.12 into 2.11 we get

$$\left. \frac{\partial f}{\partial \tilde{e}_j} \right|_p = \left\langle \sum_i a_i \tilde{e}_i, \tilde{e}_j \right\rangle = \sum_i a_i \langle \tilde{e}_i, \tilde{e}_j \rangle = \sum_i a_i g_{ij}. \tag{2.13}$$

This expression is nothing more than the matrix equation

$$df = g\ a \tag{2.14}$$

where $a = (a_1, ..., a_n)^T$. By inverting g we obtain an expression for the elements of a. Denoting the elements of the inverse of g with g^{ij} we get

$$a_i = \sum_j g^{ij} \frac{\partial f}{\partial \tilde{e}_j} \tag{2.15}$$

and the gradient of f in local coordinates is given by

$$\operatorname{grad}_p f = \sum_i \sum_j g^{ij} \frac{\partial f}{\partial \tilde{e}_j} \tilde{e}_i \tag{2.16}$$

2.3.2. The Divergence

The divergence on $\mathbb{R}^n$ is an operator acting on a vector field $\mathcal{X} : \mathbb{R}^n \to \mathbb{R}^n$ and is given by

$$\operatorname{div} v = \nabla \cdot v = \sum_i \frac{\partial \mathcal{X}_i}{\partial x_i}. \tag{2.17}$$

Physically one can think of a vector field as flow of material. The divergence at a point measures how much material gets injected or removed into the field at that point. For smooth functions f with compact support and vector fields $\mathcal{X}$, the divergence is characterized by the relation (cf. [Ros97, Sec. 1.2.3])

$$\begin{aligned}
\langle \mathcal{X}, \operatorname{grad} f \rangle &= \sum_i \int_{\mathbb{R}^n} \mathcal{X}_i \frac{\partial f}{\partial x_i} dx \\
&\underset{\substack{\uparrow \\ \text{Integration by parts}}}{=} \sum_i [X_i f]_{\partial \mathbb{R}^n} - \sum_i \int_{\mathbb{R}^n} \frac{\partial \mathcal{X}_i}{\partial x_i} f dx \\
&\underset{\substack{\uparrow \\ \textit{supp}(f) \text{ is compact}}}{=} -\sum_i \int_{\mathbb{R}^n} \frac{\partial \mathcal{X}_i}{\partial x_i} f dx \\
&= \int_{\mathbb{R}^n} - \operatorname{div} \mathcal{X} \cdot f dx \\
&= \langle - \operatorname{div} \mathcal{X}, f \rangle.
\end{aligned}$$

Note that the inner products in the first and last line are acting on different objects. We can think of the divergence as negative (formal) adjoint operator to the gradient. By using the relation for the divergence operator on $\mathbb{R}^n$ one can deduce and appropriate expression for an divergence operator acting on vector fields $\mathcal{X}$ defined over a surface [Ros97, Sec.

1.2.3]:

$$\operatorname{div} \mathcal{X} = \sum_i \frac{1}{\sqrt{\det g}} \frac{\partial}{\partial \tilde{e}_i} (\mathcal{X}_i \sqrt{\det g}). \tag{2.18}$$

One can show, that this expression is indeed independent of the parametrization.

2.3.3. The Laplacian

The Laplacian on $\mathbb{R}^n$ is defined to be

$$\Delta = \operatorname{div} \circ \operatorname{grad} = \frac{\partial^2}{\partial x_1^2} + ... + \frac{\partial^2}{\partial x_n^2} \tag{2.19}$$

This equation generalizes to functions $f : S \to \mathbb{R}$ on surfaces by using the surface gradient and divergence (equation 2.16 and 2.18)

$$\Delta f = \operatorname{div} \circ \operatorname{grad}\ f = \frac{1}{\sqrt{\det g}} \frac{\partial}{\partial \tilde{e}_i} \left(g^{ij} \frac{\partial f}{\partial \tilde{e}_j} \sqrt{\det g} \right). \tag{2.20}$$

Since the divergence and gradient on surfaces are by construction adjoint to each other, the Laplacian is self-adjoint.

$$\begin{aligned} \langle f, \Delta f \rangle &= \langle f, \operatorname{div} \operatorname{grad} f \rangle \\ &= -\langle \operatorname{grad} f, \operatorname{grad} f \rangle \\ &= \langle \operatorname{div} \operatorname{grad} f, f \rangle \\ &= \langle \Delta f, f \rangle \end{aligned}$$

This shows furthermore that the negative Laplacian $-\Delta$ is positive semi definite, since $\langle \operatorname{grad} f, \operatorname{grad} f \rangle \geq 0$.

As mentioned before gradient and divergence have illustrative interpretations for surfaces embedded in $\mathbb{R}^n$. But what about the Laplacian? To get an intuition of the action of the Laplacian one can take a step back and think of the Laplacian as the generalization of the second derivative $\frac{\partial^2}{\partial x^2}$ of functions $f(x)$ defined over $\mathbb{R}$. The second derivative of f is given by

$$f''(x) = \lim_{h \to 0} \frac{f'(x+h) - f'(x-h)}{2h}. \tag{2.21}$$

This equation is related to

$$f''(x) = \lim_{h \to 0} \frac{f(x+h) + f(x-h) - 2f(x)}{2h^2}. \tag{2.22}$$

by the rule of L'Hospital. Equation (2.22) highlights the relation of the value $f''(x_0)$

at a specific x_0 to the value of the function $f(x_0)$. The second derivative measures the difference of $f(x_0)$ to the average of f in an infinitesimal neighborhood.

The second derivative in the 1-dimensional case is also related to *curvature*. Given an arc-length parameterizedcurve $C : \mathbb{R} \to \mathbb{R}^2$ the quantity $C''(t) = \kappa_t$ is called curvature at t. But how do these concepts extend to higher dimensions?

2.3.4. The Gauss Map, Curvature

The concept of curvature on regular surfaces is a little bit more involved. It is related to the notion of a normal defined by the Gauss map.

Definition 3 (Gauss Map)**.** *The Gauss map* $N : S \to \mathbb{S}_2$ *maps a point* $p \in S$ *with a local parameterization* x *to its normal*

$$N(p) = \frac{\tilde{e}_1 \times \tilde{e}_2}{|\tilde{e}_1 \times \tilde{e}_2|}(p). \tag{2.23}$$

The differential dN_p of the Gauss map at a point $p \in S$ is a self-adjoint operator [dCa76, Sec. 3-2, Prop. 1] on the tangent space T_pS. It is intuitively clear that a point with $dN_p \equiv 0$ is locally flat. Otherwise, it captures the local curvature behavior. Therefore dN_p is also called *Shape Operator*.

Definition 4 (Second Fundamental Form)**.** *The quadratic form*

$$\mathrm{II}_p : T_pS \to \mathbb{R}, \;\; v \mapsto -\langle dN_p v, v\rangle \tag{2.24}$$

is called second fundamental form.

The second fundamental form maps (normalized) tangent vectors v of curves on the surface to their so called *normal curvature*. The minimal and maximal normal curvature at a point on the surface are called *principal curvatures* κ_1, κ_2. Different parameterizations will lead to different basis for the tangent space and shape operator. The matrix representation of shape operators in different coordinate charts is therefore related by a similarity transform and the eigenvalues κ_1, κ_2 of the shape operator are invariant under the change of parameterizations. The *gauss curvature*

$$K = \kappa_1 \kappa_2 \tag{2.25}$$

and *mean curvature*

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{2.26}$$

can be defined in terms of these eigenvalues. Since the second fundamental form is a bilinear form it might be represented by a symmetric matrix

$$\mathrm{II}_p \equiv \begin{pmatrix} e & f \\ f & g \end{pmatrix} = - \begin{pmatrix} \langle \tilde{n}_1, \tilde{e}_1 \rangle & \langle \tilde{n}_2, \tilde{e}_1 \rangle \\ \langle \tilde{n}_2, \tilde{e}_1 \rangle & \langle \tilde{n}_2, \tilde{e}_2 \rangle \end{pmatrix} \tag{2.27}$$

with $\tilde{n}_i = \frac{\partial N}{\partial e_i}$. Moreover, one can show that

$$H = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2} \tag{2.28}$$

holds [dCa76, Sec. 3-3].

Mean Curvature Flow The quantity $H(p)N(p)$ is called *mean curvature vector* (the explicit dependence on p will be dropped from now on). It has the interesting property of being the gradient of surface area with respect to the point p. To see this, we need the notion of *normal variation* (cf. [dCa76, Sec. 3-5]). If $x : \mathbb{R}^2 \to \mathbb{R}^3$ is the parameterization of a regular surface patch, we call

$$\phi(u, v, t) = x(u, v) + th(u, v)N(u, v) \tag{2.29}$$

(with $t \in (-\varepsilon, \varepsilon)$) normal variation. A normal variation describes a family of "small" deformation of the surface in normal direction parameterizedby t. The constant ε can be chosen small enough to guarantee that $\phi(u, v, t)$ is a regular surface patch for every t within $(-\varepsilon, \varepsilon)$. The basis of the tangent plane and the fundamental forms of ϕ dependent on the parameter t. For a fixed constant time t, we obtain

$$\tilde{e}_i^t = d\phi(u, v, t)e_i = \tilde{e}_i + th\tilde{n}_i + t\tilde{h}_i N \tag{2.30}$$

and can write the entries of the first fundamental form at time t as

$$\begin{aligned} E^t &= E - 2the + \mathcal{O}\left(t^2\right) \\ F^t &= F - 2thf + \mathcal{O}\left(t^2\right) \\ G^t &= G - 2thg + \mathcal{O}\left(t^2\right) \end{aligned}$$

with e, f, g defined by equation 2.27. Note, that the differentiation in 2.30 is with respect to u and v only because t is constant. The squared infinitesimal area element A^t at time

t within an sufficiently small ε is then given by

$$\left(A^t\right)^2 = E^tG^t - \left(F^t\right)^2 = EG - F^2 - 2th(Eg - 2Ff + Ge) + \mathcal{O}\left(t^2\right) \tag{2.31}$$

which can be written in terms of the mean curvature H (cf. equation 2.28) as

$$A^t = A^0\sqrt{1 - 4thH} \tag{2.32}$$

and gives

$$\left(A^t\right)'|_{t=0} = -2hHA^0. \tag{2.33}$$

Because variation in normal direction yields a maximal change in surface area we obtain by setting $h \equiv 1$:

$$\nabla A = -2HNA^0. \tag{2.34}$$

The *mean curvature flow* $\frac{\partial x}{\partial t} = -HN$ is thus an area minimizing deformation of the surface. In other words, evolving the surface in direction of the mean curvature normal will maximally decrease surface area.

An *isothermal parametrization* is characterized by $E = G = \lambda^2, F = 0$. Such a parameterization induces an orthogonal basis on the tangent space. In case of an isothermal parameterization x, the Laplace-Beltrami operator (2.20) simplifies to

$$\frac{1}{\lambda^2}\sum_i \frac{\partial^2}{\partial x_i^2}. \tag{2.35}$$

Moreover, the mean curvature 2.28 becomes

$$H = \frac{1}{2}\frac{g + e}{\lambda^2}. \tag{2.36}$$

It follows by a short calculation [dCa76, Sec. 3-5, Prop. 2], that

$$\frac{\partial^2}{\partial x_1^2}x + \frac{\partial^2}{\partial x_2^2}x = 2\lambda^2 HN \tag{2.37}$$

or equivalently

$$\Delta x = 2HN \tag{2.38}$$

The Laplacian applied to isothermal coordinates therefore yields (twice) the mean curvature normal. In combination with 2.34 we obtain the relationship

$$A^0\Delta x = -\nabla A \tag{2.39}$$

Figure (2.1): Costas minimal surface. Rendering by Paul Nyander.

which will lead to a straightforward discretization of the Laplacian.

Dirichlet's Principle Given an open bounded domain $\Omega \subset \mathbb{R}^2$, Dirichlet's principle states that one can find a function on Ω with prescribed values on the boundary $\partial\Omega$ that fulfills $\Delta f = 0$ in the interior of Ω by minimizing *Dirichlet's energy*

$$E_d(f) = \int_\Omega |\nabla f|^2 dV \tag{2.40}$$

in a variational sense. Solutions to this problem are called *harmonic functions*. Dirichlet's principle is just a special case of the Euler-Lagrange equation known from variational calculus. A solution to the problem is nevertheless not guaranteed. One has to define the space of admissible functions and the shape of the domain Ω carefully [Cou77].

A direct consequence from the definition of harmonic function is the *maximum principle*. The maximum principle for harmonic functions states, that harmonic functions have no local extrema in Ω or equivalently, they reach their extrema at the boundary $\partial\Omega$. Every point inside the domain is a saddle point.

Fixing the boundary of a surface and solving for a surface with harmonic coordinate functions leads to *minimal surfaces*. It follows from the preceding section that minimal surfaces have extremal area, which explains the name. Minimizing a discrete Dirichlet energy leads to an efficient numerical scheme for computing minimal surfaces. Minimal surfaces appear for example as soap bubbles with fixed boundary (figure 2.1).

3. Discrete Laplacians

3.1. The Discrete Picture

The discrete analogue to a regular surface is called *simplicial* 2*-complex.* Simplicial 2-complexes are constructed out of so called simplices: points, edges and triangular faces. The following definitions assert that the simplicial complex fulfills some reasonable properties mimicking the smooth setting.

Definition 5 (k-Simplex)**.** *A k-simplex is the convex hull of $k+1$ affinely independent points. k points are affinely independent if they span a k dimensional affine subspace.*

Theses simplices are glued together in order to get triangulated surfaces.

Definition 6 (Simplicial 2-Complex)**.** *A simplicial* 2*-complex is a collection of* 0*-,* 1*- and* 2*-simplices $\mathcal{K}$ with the properties:*

1. *Every simplex t with $t \subset s$ for a $s \in \mathcal{K}$ is contained in $\mathcal{K}$.*
2. *The intersection $t \cap s$ of two simplices $s, t \in \mathcal{K}$ is contained in $\mathcal{K}$.*

This definition asserts that the edges of every triangle are in the complex and triangles intersect in edges also contained in the complex. In order to ensure the "manifoldness" of the simplical complex I will assume for the rest of the text that the following condition also holds.

3 Every 1-simplex $s \in \mathcal{K}$ is contained in exactly two $2-$simplices $t, u \in \mathcal{K}$.

Simplical 2-complexes can be represented by triangular meshes. A triangular mesh $\mathcal{M}$ is composed of a triple of indexed lists (X, Y, Z) representing the vertex coordinates i.e. $X = (x_1, x_2, \ldots, x_n)$ and a set of simplices $\mathcal{S}$ represented by indices into the list of coordinates. The set of 0-simplices (vertices) will be denoted by $\mathcal{V}$, the set of 1-simplices (edges) by $\mathcal{E}$ and the set of 2-simplices (faces) by $\mathcal{F}$.

In order to represent general polygonal meshes and dual meshes we need a generalization of simplical 2-complexes. Informally we obtain a *cell complex* by replacing k-simplices by open k-balls. That is, we do not require the elements to be a convex hull but rather to be homeomorphic to the open k-ball. We represent 2-dimensional cell complexes in the discrete setting by points, edges and general polygons defined by a closed loop of edges.

3.2. Properties of Discrete Laplacians

Motivated by the continuous definition of the Laplacian the question arises whether an corresponding discrete operator on polygonal meshes exists. Real valued functions on meshes are generally given by a discrete mapping $f : \mathcal{V} \to \mathbb{R}$. As in the smooth setting a discrete Laplacian should be a linear map of functions to functions. A discrete Laplacian can therefore be represented by a real valued $|V| \times |V|$ matrix L. Such a matrix can be defined by the real coefficients ω_{ij} in

$$(Lu)_i = \sum_j \omega_{ij}(u_i - u_j). \tag{3.1}$$

Wardetzky et al. [War07] identify a set of core properties that should be fulfilled by a discrete Laplacian in order to mimic the smooth setting. As it turns out there is, under reasonable assumptions, no discretization having all desired properties.

1. SYMMETRY (SYM): The continuous Laplace operator is self-adjoint, so should the matrix L. This leads to real eigenvalues and orthogonal eigenvectors.
2. LOCALITY (LOC): In order to mimic a differential operator the support for evaluating the Laplacian at a vertex should be "small". The following discussion will be restricted to discrete Laplacians with 1-ring support. This means $\omega_{ij} = 0$ if $(i, j) \notin \mathcal{E}$.
3. POSITIVE WEIGHTS (POS): Positive weights ω_{ij} for $i \neq j$ ensure a maximum principle for discrete harmonic functions. Consider a discrete function f with $\Delta f = 0$ and $\tilde{\omega}_{ij} = \frac{\omega_{ij}}{\sum_{k \in \mathcal{N}(i)} \omega_{ik}}$. Therefore $f_i = \sum_{j \in \mathcal{N}(i)} \tilde{\omega}_{ij} f_j$. With $0 \leq \tilde{\omega}_{ij} \leq 1$ we have for all i indices $j', j'' \in \mathcal{N}(i)$ such that $f_{j'} \leq f_i \leq f_{j''}$ holds. The value at every interior vertex is therefore not a local extremum.
4. POSITIVE SEMI-DEFINITENESS (PSD): L should be positive semi-definite, like the continuous operator in order to ensure the non-negativity of the *Dirichlet energy* $E_D(u) = u^T Lu$. The smooth Dirichlet energy vanishes for linear functions. Since linear functions on a closed mesh are constant, the kernel should be 1-dimensional.
5. LINEAR PRECISION (LIN): This property should be adopted to ensure reasonable behavior on planar meshes e.g. in parameterization applications. A discrete Laplace operator L is linear precise if $(Lu)_i = 0$ whenever the following conditions hold
    - all incident triangles to vertex i are coplanar.

- the discrete function u is linear on the 1-ring of i.
- i is not a boundary vertex.

Note, that this condition does not conflict with (Psd), because a planar embedded mesh will have boundary vertices and linear functions are not required to vanish at the boundary. Linear functions on a mesh with boundary will thus not appear in the null-space of the Laplacian.

6. Convergence (Con): When the mesh resolution becomes finer the discrete operator should converge to the smooth Laplacian (under reasonable conditions on the refinement).

3.3. Combinatorial Laplacians

Combinatorial Laplacians represent the simplest yet powerful class of discrete Laplace operators. Their coefficients are solely based on connectivity information (also called combinatoric of the mesh). The main advantage of this class of discrete Laplacians is, that the matrix does not have to be rebuild when altering the mesh i.e. in an iterative process. This advantage is even more striking if computational expensive decompositions of the matrix are used. The downside is of course the lack of any geometrical information. Especially in the presence of nonuniform sampling problems can arise. This section reviews the most prominent instances of combinatorial Laplacians.

The Umbrella Operator The umbrella operator is a straightforward generalization of the five-point stencil Laplacian used in image processing on a regular grid. This discrete Laplacian is motivated by the fact, that the Laplacian evaluates the distance of a function value to the average of function values in a small neighborhood (see section 2.3.3). Denoting the *degree* of a vertex i by $d_i := |\{j : \; (i,j) \in \mathcal{E}\}|$, the umbrella operator L_U is given by

$$\omega_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{else.} \end{cases} \tag{3.2}$$

through the relation (3.1). The properties (Sym), (Loc) and (Pos) are fulfilled by construction. (Lin) does not hold. This follows from the fact that combinatorial operators are independent of an embedding, the notion of a planar embedding is therefore “invisible” to any combinatorial operator. The property (Psd) is implied by (Pos) and (Sym). For a discrete function u over the mesh with $u \neq const$:

$$\begin{aligned} u^T L_U u &= \sum_i u_i \sum_j \omega_{ij}(u_j - u_i) \\ &= \sum_{ij} u_i \omega_{ij}(u_j - u_i) \\ &\underset{(\textsc{Sym})}{=} \sum_{i>j} \omega_{ij}(u_i(u_j - u_i) + u_j(u_i - u_j)) \\ &= \sum_{i>j} \omega_{ij}(u_j - u_i)^2 \underset{(\textsc{Pos})}{>} 0 \end{aligned}$$

This fact is again valid for any combinatorial operator, since the specific structure of L_U is not used in the proof.

The Tutte Laplacian This operator is given by the row-wise normalized umbrella operator. The normalization destroys the symmetry for meshes with irregular connectivity. (Psd) is nevertheless preserved[Zha04]. The spectrum of the Tutte Laplacian is bounded within $[0, 2]$.

The normalized graph Laplacian This operator is an symmetrized version of the Tutte Laplacian. The weights are given by

$$\omega_{ij} = \begin{cases} \frac{1}{\sqrt{d_i d_j}} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{else.} \end{cases}$$

The property (Psd) is destroyed and the kernel can contain non-constant vectors.

Further details and more elaborate combinatorial Laplacians can be found in [Zha04]. The choice of a discrete Laplacian depends in general on the application and the properties that should be preserved during discretization.

3.4. Pinkall's Cotan Operator

Originally derived in the context of numerical computation of minimal surfaces (section 2.3.4), the cotan operator represents the most common discretization of the Laplacian on triangle meshes. Pinkall and Polthier [PP93] observe, that discrete minimal surfaces, interpreted as coordinate function defined on the mesh, can be characterized as minimizers of a discrete Dirichlet energy E_d (cf. 2.40). The dirichlet energy is derived in the more general case of an arbitrary geometry function $f : \mathcal{V} \to \mathbb{R}^3$ that maps every triangle (i, j, k)

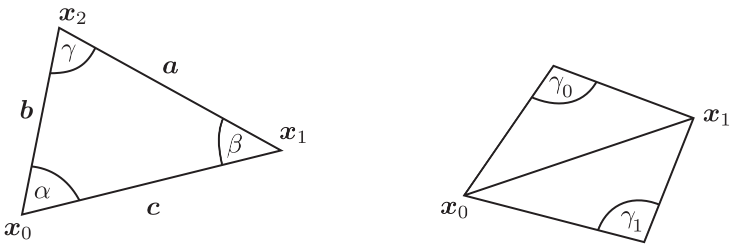

Figure (3.1): Notation used in the formulation of the discrete Laplacian.

in the mesh to the triangle $(f(i), f(j), f(k))$ (see figure 3.2). The discrete Dirichlet energy is given by

$$E_d(f) = \frac{1}{2}\int_{\mathcal{M}} |\nabla f|^2 dx. \tag{3.3}$$

analogous to the smooth case. This functional is the sum of the functionals $E_d(f)|_T$ over each triangle T in $\mathcal{F}$. We are thus interested in the differential of a (affine) function f_T (the explicit restriction to T will be dropped from now on) that maps a triangle T_g onto another triangle T_h.

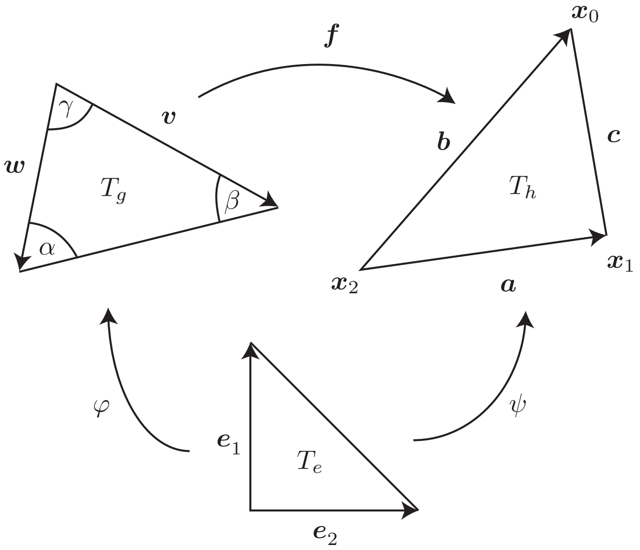

Figure (3.2): A function f mapping triangle T_g to T_h can be decomposed into mappings φ and ψ.

We can consider the map f as the composition of the maps ψ and φ^{-1}, mapping from and to the unit triangle T_e (figure 3.2). Since ψ is affine, $d\psi$ will be constant over the triangle. The differential takes the particularly simple form $d\psi = (a, b)$, because we are mapping from the unit triangle. This gives

$$d\psi^T d\psi = \begin{pmatrix} \langle a, a\rangle & \langle a, b\rangle \\ \langle a, b\rangle & \langle b, b\rangle \end{pmatrix} \tag{3.4}$$

It follows from the inverse function theorem, that differentation and the operation of taking the inverse can be interchanged for smooth functions:

$$d\varphi^{-1} = (d\varphi)^{-1} = \begin{pmatrix} v & w \end{pmatrix}^{-1} = \begin{pmatrix} v_0 & w_0 \\ v_1 & w_1 \end{pmatrix}^{-1} = \frac{1}{\det \begin{pmatrix} v & w \end{pmatrix}} \begin{pmatrix} w_1 & -w_0 \\ -v_1 & v_0 \end{pmatrix} \tag{3.5}$$

With the area of the triangle $A = \frac{1}{2} \det \begin{pmatrix} v & w \end{pmatrix}$ we get

$$d\varphi^{-1} d\varphi^{-1^T} = \frac{1}{4A^2} \begin{pmatrix} \langle w, w\rangle & -\langle v, w\rangle \\ -\langle v, w\rangle & \langle v, v\rangle \end{pmatrix}. \tag{3.6}$$

We can now express the integrand of the Dirichlet energy

$$\begin{aligned} |\nabla f|^2 &= \operatorname{tr}\left[(df)^t df\right] \\ &= \operatorname{tr}\left[(d\varphi^{-1})^T d\psi^T d\psi d\varphi^{-1}\right] \\ &= \operatorname{tr}\left[d\psi^T d\psi d\varphi^{-1} (d\varphi^{-1})^T\right] \end{aligned}$$

By using equations 3.4 and 3.6 and rearranging terms we obtain:

$$\begin{aligned} &= \frac{1}{4A^2}(\langle a, a\rangle\langle w, w\rangle - 2\langle a, b\rangle\langle v, w\rangle + \langle v, v\rangle\langle b, b\rangle) \\ &= \frac{1}{4A^2}((\langle w, w\rangle - \langle v, w\rangle)|a|^2 + (\langle v, v\rangle - \langle v, w\rangle)|b|^2 + \langle v, w\rangle|a - b|^2) \\ &= \frac{1}{4A^2}(\langle w - v, w\rangle|a|^2 + \langle v, v - w\rangle|b|^2 + \langle v, w\rangle|c|^2) \end{aligned}$$

The triangle area can be expressed with cross products:

$$= \frac{1}{2A}\left(\frac{\langle w - v, w\rangle}{|(v - w) \times w|}|a|^2 + \frac{\langle v, v - w\rangle}{|v \times (v - w)|}|b|^2 + \frac{\langle v, w\rangle}{|v \times w|}|c|^2\right)$$

Scalar and vector product can be expressed in terms of sin and cos:

$$\begin{aligned} &= \frac{1}{2A}\left(\frac{|w - v||w| \cos(\alpha)}{|w - v||w| \sin(\alpha)}|a|^2 + \frac{|v||v - w| \cos(\beta)}{|v||v - w| \sin(\beta)}|b|^2 + \frac{|v||w| \cos(\gamma)}{|v||w| \sin(\gamma)}|c|^2\right) \\ &= \frac{1}{2A}(\cot\alpha|a|^2 + \cot\beta|b|^2 + \cot\gamma|b|^2) \end{aligned}$$

And we finally obtain

$$E_d(f) = \frac{1}{4}\sum_{\mathcal{F}} \cot\alpha|a|^2 + \cot\beta|b|^2 + \cot\gamma|c|^2. \tag{3.7}$$

Note, that the angles refer to the base triangle whereas the lengths a, b, c refer to the mapped triangle. For the special case $f \equiv \mathrm{Id}$, that is, the geometry is mapped to itself, length and angles come from the same triangle. With the notation from figure 3.1 and reordering to sum over the edges of the mesh we get

$$E_d(f) = \frac{1}{4} \sum_{(x_0, x_1) \in \mathcal{E}} (\cot \gamma_0 + \cot \gamma_1) |x_1 - x_0|^2. \tag{3.8}$$

By differentiating (3.8) with respect to vertex positions we obtain

$$\frac{dE_d}{dx_0} = -\frac{1}{2} \sum_{x \in \mathcal{N}(x_0)} (\cot \gamma_0 + \cot \gamma_1)(x - x_0). \tag{3.9}$$

It follows, that $\frac{dE_d}{dx_0} = 0$ (with appropriate boundary conditions) is the condition for $\mathcal{M}$ to be minimal (the coordinate functions are discrete harmonic). In other words $\frac{dE_d}{dx_0}$ can be interpreted as a discrete Laplacian, since $\Delta x = 0$ is the condition for the Dirichlet energy to be minimal in the smooth case (cf. 2.40).

3.5. Desbrun's Cotan Operator

As seen in the smooth setting (section 2.3.4), the Laplacian is intimately related to the area gradient. For an isothermal parameterization x and a small surface patch of area A we have:

$$A^0 \Delta x = -\nabla A \tag{3.10}$$

(cf. 2.39). Desbrun et al. [Des99] find the (area normalized) cotan Laplacian by computing the area gradient explicitly in the discrete setting. They use the gradient of the 1-ring area with respect to its center vertex. The following proof shows this approach indeed yields the cotan-formula.

Proof. Let $T = (a, b, c)$ be a Triangle with $a, b, c \in \mathbb{R}^3$. The area of T can be written as

$$A = \frac{1}{2} \sqrt{|b - a|^2 |c - a|^2 - \langle b - a, c - a \rangle^2}. \tag{3.11}$$

The derivative with respect to a is given by

$$\begin{aligned} \nabla_a A &= \frac{1}{2A} \nabla_a (|b - a|^2 |c - a|^2 - \langle b - a, c - a \rangle^2) \\ &= -\frac{1}{2A} \left[2(b - a)|c - a|^2 + 2(c - a)|b - a|^2 + 2\langle b - a, c - a \rangle (2a - c - b) \right] \end{aligned}$$

Expanding the squared norms and rearranging gives:

$$
\begin{aligned}
&= -\frac{1}{A}\left[(b-a)\langle c-a+a-b, c-a\rangle + (c-a)\langle b-a, b-a-c+a\rangle\right] \\
&= -\frac{1}{A}\left[(b-a)\langle b-c, a-c\rangle + (c-a)\langle a-b, c-b\rangle\right]
\end{aligned}
$$

The area and scalar products can be expressed in terms of the side lengths and the enclosed angle.

$$
\begin{aligned}
&= -\frac{1}{2}\frac{|b-c|\ |a-c|\cos\gamma}{|b-c|\ |a-c|\sin\gamma}(b-a) - \frac{1}{2}\frac{|a-b|\ |c-b|\cos\beta}{|a-b|\ |c-b|\sin\beta}(c-a) \\
&= -\frac{1}{2}\cot\gamma(b-a) - \frac{1}{2}\cot\beta(c-a)
\end{aligned}
$$

This is the cotan-formula derived earlier through a totally different construction. Using formula 3.10 we get the area normalized cotan weights

$$\omega_{ij} = \frac{1}{4A_{Ring}}(\cot\gamma_{ij} + \cot\gamma_{ji}). \tag{3.12}$$

□

3.6. The DEC Laplacian

It is also possible to derive the cotan operator in the framework of *Discrete Exterior Calculus*. DEC does not try to directly discretize operators but to build them up from basic building blocks analogous to the smooth theory of exterior calculus [Hir03]. The Laplacian can be written as

$$\Delta = \operatorname{div}\operatorname{grad} = \delta d = d \star d$$

with $\star$ being the Hodge-Star-operator. With a discrete version of the Hodge-Star-operator and the differential in place we can use the smooth theory as recipe for a discrete Laplacian. The discrete differential is the difference of function values per edge. The DEC-Hodge-Star is defined with the help of a dual mesh (see figure 3.3).

Definition 7 (Circumcentric Dual). *For a given simplical 2-complex $\mathcal{K}$ the circumcentric dual complex is a cell 2-complex $\star\mathcal{K}$. Every k-simplex in $t \in \mathcal{K}$ has a corresponding $2-k$ cell $\star t \in \star\mathcal{K}$. For every $s \subset t$ we have $\star t \subset \star s$. The circumcenter of every 2-complex (primal face) determines the position of its dual 0-cell (dual vertex).*

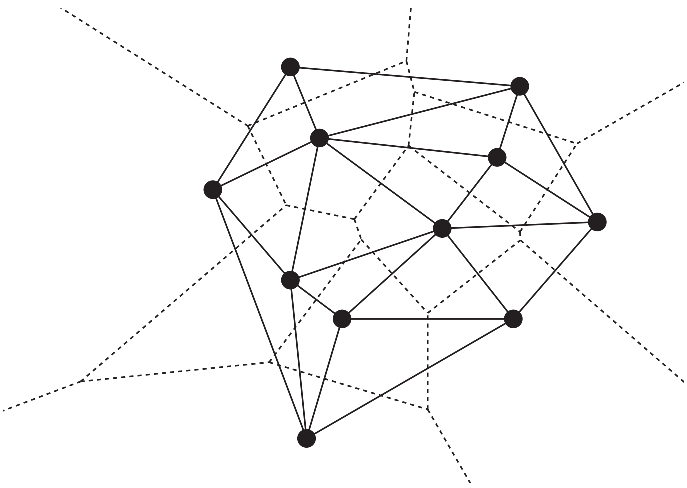

Figure (3.3): A planar simplicial complex and its cicumcentric dual (dotted). Dual simplices might be unbounded in case of a primal complex with boundary.

The DEC-Laplacian of a function $u : \mathcal{V} \to \mathbb{R}$ evaluated at vertex i is then given by

$$(L_D u)_i = \frac{1}{|\star i|} \sum_{e \in \mathcal{E}, e=(i,j)} \frac{|\star e|}{|e|}(u_i - u_j) \tag{3.13}$$

[Hir03]. This geometric construction can be shown to be equivalent to the area normalized cotan operator.

Lemma 1. *The DEC Laplacian (3.13) is the area normalized cotan operator.*

Proof. Following the notation from Glickenstein [Gli08] we denote the angle at vertex k in a triangle (i,j,k) with γ_{kij}. $C(i,j,k)$ is the circumcenter of the triangle and $C(i,j)$ the circumcenter of the edge (i,j) (cf. figure 3.4). Since the lines perpendicular to the edges through the midpoints meet in the circumcenter the angle $\gamma_{kC(i,j,k)j}$ and $\gamma_{jkC(i,j,k)}$ are equal. With the same argument we have $\gamma_{kiC(i,j,k)} = \gamma_{iC(i,j,k)k}$. This gives $\gamma_{C(i,j,k)jk} = \pi - 2\gamma_{kC(i,j,k)j}$ and $\gamma_{C(i,j,k)ki} = \pi - 2\gamma_{kiC(i,j,k)}$. Since $2\varphi = 2\pi - \gamma_{C(i,j,k)ki} - \gamma_{C(i,j,k)jk}$ we get

$$2\varphi = 2\gamma_{kC(i,j,k),j} + 2\gamma_{kiC(i,j,k)} \Rightarrow \varphi = \gamma_{kij}. \tag{3.14}$$

With $\frac{1}{2}|(i,j)| \cot\varphi = |(C(i,j,k), C(i,j))|$ we get

$$\frac{|\star (i,j)|}{|(i,j)|} = \frac{1}{2}(\cot\gamma_{ij} + \cot\gamma_{ji}) \tag{3.15}$$

which completes the proof. □

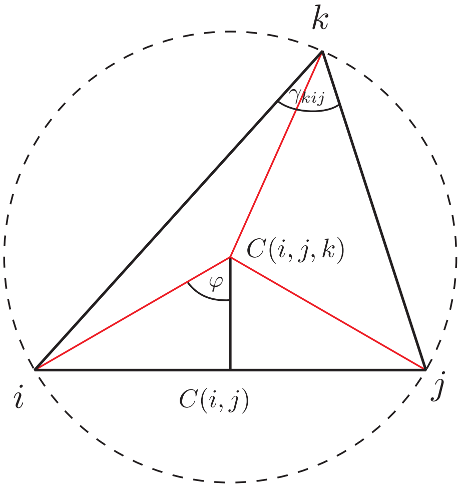

Figure (3.4): Weighted dual mesh (red). The distance between edge and triangle circumcenter, $C(i,j)$ and $C(i,j,k)$ respectively, can be calculated in terms of the angle γ_i and the distances d_{ij} and d_{ik}.

3.7. Properties of the Cotan Operator

The cotan operator is given by (3.9) as

$$\omega_{ij} = \frac{1}{2}(\cot(\gamma_0) + \cot(\gamma_1)) \tag{3.16}$$

following the notation from figure 3.1. The cotan operator is obviously symmetric (Sym). (Psd) is implied because the discrete Dirichlet energy is positive by construction.

Lemma 2. *The cotan operator L fulfills the property* (Lin)*.*

Proof. Consider wlog. a mesh with all incident triangles to the interior vertex i embedded in the x-y-plane. In order to fulfill (Lin) every linear function f on the 1-ring of i has to vanish at vertex i under the action of the Laplacian. The space of linear functions on this 1-ring is spanned by the functions u_x, u_y mapping every vertex to its own x- and y-coordinate and the constants. With the DEC formulation of the cotan Laplacian (3.13) and the vector-valued function $u = (u_x, u_y)$ we get

$$\begin{aligned} (Lu)_i &= \frac{1}{|\star i|} \sum_{(i,j)\in\mathcal{E}} \frac{|\star(i,j)|}{|(i,j)|}(u_i - u_j) \\ &= \frac{1}{|\star i|} \sum_{(i,j)\in\mathcal{E}} \frac{(i,j)}{|(i,j)|}|\star(i,j)|. \end{aligned}$$

Since the dual edges form a closed loop we have $\sum_{(i,j)\in\mathcal{E}} \star(i,j) = 0$ for a fixed i. Because dual and primal edges are perpendicular, the mapping

$$\star(i,j) \mapsto \frac{(i,j)}{|(i,j)|}|\star(i,j)| \tag{3.17}$$

is just a $\pi/2$-rotation and we can conclude that $(Lu_x)_i = (Lu_y)_i = 0$. Since the constants are in the kernel of the operator we have $(Lu)_i = 0$ for every linear function u and inner vertices i. □

The convergence behavior (Con) of the cotan Laplacian was discussed by Wardetzky [War08]. The point-wise convergence of a family of meshes to a smooth surface is in general not sufficient for the cotan-formula to converge to the smooth Laplacian. However, one can show that if the meshes converge in *Hausdorff distance* and the normals converge as well (which is not implied), the cotan operator converges to the smooth Laplacian with respect to an operator norm.

The positivity (Pos) is in general violated because $\cot(x) < 0$ for $x \in (\pi/2, \pi)$. For meshes with a bound of $\pi/2$ on the angles positivity is guaranteed and therefor a discrete maximum principle holds.

3.8. Weighted Cotan Operator

We have seen that the property (Lin) is intimately related to the dual mesh. The fact that the boundary loop of the dual face to each vertex is closed was crucial for the proof, but the choice of the circumcenter as dual vertex of each face was somewhat arbitrary. Varying the position of each dual vertex will lead to cotan operators that are guaranteed to maintain linear precision and symmetry. Glickenstein [Gli08] shows that the space of orthogonal dual meshes can be parameterizedby vertex weights leading to weighted discrete Laplace operators. They can be shown to be positive definite as long as the dual vertex lies inside the circumcircle of the triangle [Gli07].

As before, we denote by (i,j,k) the triangle formed by the vertices i, j and k and by (i,j) the edge from i to j. The points $C(i,j,k)$ and $C(i,j)$ are the circumcenter of the triangle and edge respectively. The triangle circumcenter induces a dual mesh with dual edges perpendicular to the primal edges. The dual edge $\star(i,j)$ intersects the edge (i,j) in the circumcenter $C(i,j)$. The definition of circumcenter as a point that is equidistant to a set of 2 or 3 points depends on the specific distance $d : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$. Consider the *power distance* defined by

$$\pi_p(x) = ||x-p||_2^2 - w_p \tag{3.18}$$

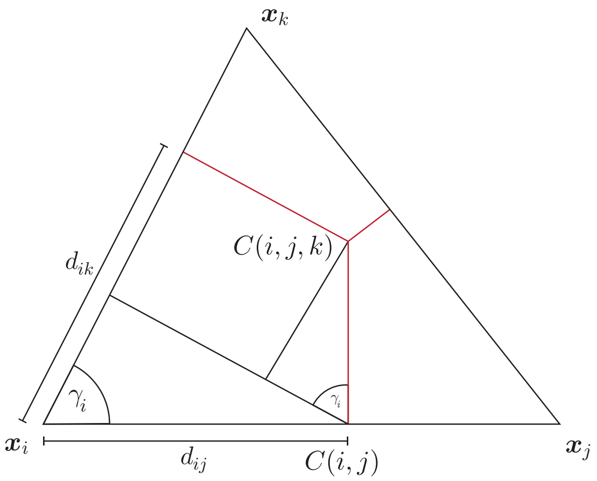

Figure (3.5): Weighted dual mesh (red). The signed distance between weighted edge and triangle circumcenter, $C(i,j)$ and $C(i,j,k)$ respectively, can be calculated in terms of the angle γ_i and the distances d_{ij} and d_{ik}.

with a weight $w_p \in \mathbb{R}$. Given a triangle (i,j,k) and weights w_i, w_j, w_k we can define the *weighted circumcenter* C_w of the triangle by

$$\pi_{x_i}(C_w(i,j,k)) = \pi_{x_j}(C_w(i,j,k)) = \pi_{x_k}(C_w(i,j,k)) \tag{3.19}$$

and accordingly the weighted circumcenter of an edge (i,j) as

$$\pi_{x_i}(C_w(i,j)) = \pi_{x_j}(C_w(i,j)) \text{ and } C_w(i,j) = sx_i + tx_j \text{ for some } s,t \in \mathbb{R}. \tag{3.20}$$

For a point p with equal power distance to x_i and x_j we have

$$||x_i - p||^2 - w_i = ||x_j - p||^2 - w_j \tag{3.21}$$

which can be rearranged to give

$$\langle p, x_j - x_i \rangle = w_i - w_j - ||x_i||^2 + ||x_j||^2. \tag{3.22}$$

The set of points p obeying this equation form a line in the plane of the triangle perpendicular to the primal edge (i,j). The three (weighted) dual edges of a triangle intersect in $C_w(i,j,k)$. The power distance with respect to a weight functions thus induces an orthogonal dual mesh. Given a weighted triangle mesh we can define a weighted Laplacian by using formula 3.13 together with a weighted dual mesh. In order to implement the weighted Laplace operator, it is necessary to evaluate the weighted dual edge length.

Denoting the distance of a vertex i to the the weighted circumcenter $C(i,j)$ by d_{ij} and the angle at vertex i by γ_i, [Gli07] gives the formula

$$d_{\pm}(C(i,j,k),C(i,j)) = \frac{d_{ik} - d_{ij}\cos(\gamma_i)}{\sin(\gamma_i)} \tag{3.23}$$

for the signed distance between triangle and edge circumcenter. Figure 3.5 illustrates this relationship. The coefficients of the Laplacian with respect to a triangle are given by

$$\omega_{ij} = \frac{d_{\pm}(C(i,j,k),C(i,j))}{|(i,j)|} = \frac{d_{ik} - d_{ij}\cos(\gamma_i)}{|(i,j)|\sin(\gamma_i)}. \tag{3.24}$$

For d_{ij} we have

$$\begin{aligned} d_{ij}^2 - w_i &= d_{ji}^2 - w_j = (|(i,j)| - d_{ij})^2 - w_j \\ &= (|(i,j)| - d_{ij})^2 - w_j \\ &= |(i,j)|^2 - 2|(i,j)|d_{ij} + d_{ij}^2 - w_j \end{aligned}$$

and therefore

$$d_{ij} = \frac{|(i,j)|^2 + w_i - w_j}{2|(i,j)|} \tag{3.25}$$

which leadss to a simple formula for the weighted Laplacian.

4. General Polygonal Meshes

The operators encountered so far are all defined on triangular meshes (simplical 2-complexes). This makes a lot of sense from a mathematical point of view as seen in the definition of simplical complexes. Since three vertices in general position describe a single plane, it is easy to incorporate ideas from the smooth theory like parameterization and normal vectors. From a aesthetically and practically point of view triangle meshes are not always the right choice. The movie industry generally prefers quad-meshes. This is attributed to the extensive use of Catmull-Clark subdivision surfaces. Moreover, it is important for a 3d-artist to have control over the *edge-flow* while modeling an object which is much easier in a semi-regular quad mesh (figure 4.1) .

A quick remedy would be to triangulate the quad meshes before applying operations involving a discrete Laplacian. This would destroy an extensive amount of information, especially for low-poly meshes like in figure 4.1. Depending on the chosen triangulation the result will consequently differ. Therefore we need a real polygon based geometric Laplacian. It is easy to extend the definition of combinatorial Laplacians to general polygonal meshes (cell 2-complexes). This is not the case for the cotan-Laplacian. We have seen that many derivations of the cotan-formula exist, all of which can potentially be used to generalize the cotan-formula. The main obstacle to a direct generalization of the ideas is the fact, that there is a priori no distinguished surface spanned by the polygon.

4.1. Orthogonal Duals for Polygonal Meshes

Orthogonal dual meshes are at the heart of the discrete Laplacian formulation in discrete exterior calculus and lead to linear precision. A straightforward generalization of the cotan operator to polygonal meshes would thus be to construct orthogonal dual meshes and use the cotan-formulation from DEC (equation 3.13). Note, that the problem of finding admissible dual vertices is linear. Given a mesh $(\mathcal{V}, \mathcal{E}, \mathcal{F})$ a solution to the problem consists of a set of dual vertices per face $\mathcal{V}' = \star\mathcal{F}$ and a set of distances d_{ij} per unoriented edge. The set of unoriented edges $\mathcal{E}_u$ is the set of edges modulo the relation $(i, j) \sim (j, i)$. Per face f_k with dual vertex v'_k and for each edge $(i, j) \in \mathcal{E}_u$ adjacent to that face we have the

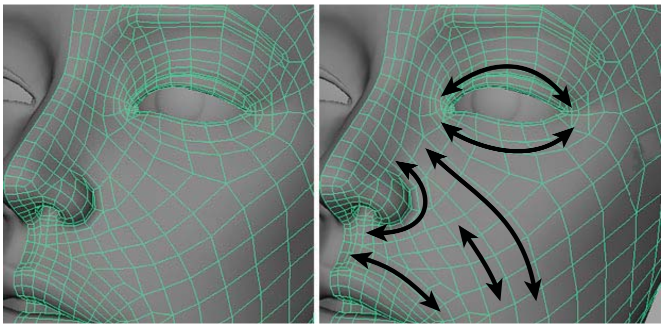

Figure (4.1): The edges of the base quad mesh follow smooth feature lines of the face (right). Carefully modeling the edge flow is important for the visual aesthetics of the final Catmull-Clark subdivision surface (left).

condition

$$\left\langle v_i + d_{ij}\frac{v_j - v_i}{||v_j - v_i||} - v'_k, v_j - v_i \right\rangle = 0 \tag{4.1}$$

expressing the orthogonality between primal and dual edge. The set of $|\mathcal{E}|$ equations forms a linear system in the $3|\mathcal{F}| + |\mathcal{E}_u|$ unknowns v'_k and d_{ij}. For a closed mesh of n-gons we have the relation

$$n|\mathcal{F}| = |\mathcal{E}| = 2|\mathcal{E}_u|. \tag{4.2}$$

Assuming the systems have almost maximal rank we have an under constraint system for n-gons with $n \leq 5$. By solving the system and computing the null space we can explore the space of admissible dual meshes. Note that the set of d_{ij}'s and the set of dual vertices determine each other. Numerical experiments suggest, that the dimension of the solution space is generally very close to $|\mathcal{F}|$ for quad meshes. A reasonable choice from the solution space is a mesh that minimizes the squared distance between the face barycenters and their dual vertices. An implementation of this strategy in Mathematica can be found in the appendix (listing A.2). Results of the optimization can be found in figures 4.2 and 4.3. The computational cost is unfortunately prohibitively large, even for moderate sized meshes (a couple of minutes for figure 4.3 on a 2.2 GHz dual core laptop). A simple parameterization of the space of orthogonal duals, like for the triangle case (section 3.8), would be necessary in order to use orthogonal duals for polygonal meshes in an interactive application.

The approach bears some similarity to the HOT energy introduced by Mullen et al. [Mul11]. They try to optimize the weights on a triangulation in order to move the circumcenters closer to the barycenters of each triangle. The application of HOT energy optimization for polygonal meshes is an interesting direction for future research.

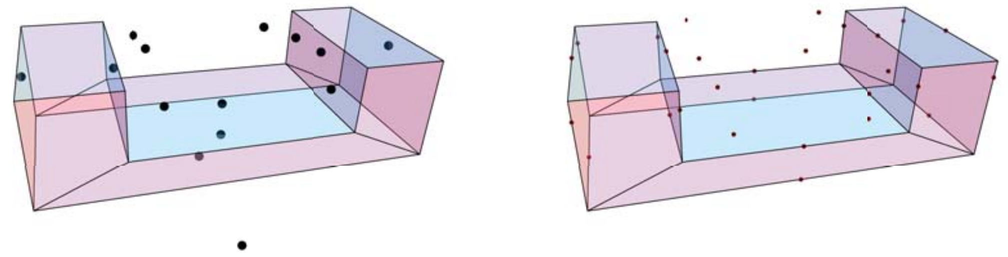

Figure (4.2): Dual vertices (left) and edge midpoints (right) of a quad mesh obtained by an optimization implemented in Mathematica (listing A.2).

4.2. Alexa and Wardetzky's Polygon Laplacian

To cope with the lack of a spanned surface it is natural to look for mathematical quantities of a surface that can be expressed in terms of its boundary. *Stoke's theorem* might come to mind stating

$$\int_S \langle \nabla \times \mathcal{X}, n \rangle dA = \int_{\partial S} \langle \mathcal{X}, dx \rangle. \tag{4.3}$$

for a smooth vector field $\mathcal{X}$ and a compact surface S with boundary ∂S. Integration of the normal component of the vector field $\nabla \times \mathcal{X}$ over the surface yields the same value as integrating the vector field $\mathcal{X}$ along the boundary. Given vector fields a_i, $i = 0, 1, 2$ with $\nabla \times a_i = e_i$ equation (4.3) gives

$$\int_S n_i \; dA = \int_S \langle \nabla \times a_i, n \rangle \; dA = \int_{\partial S} \langle a_i, dx \rangle \tag{4.4}$$

The integral $\int_S n \; dA$ is called *vector area* of the surface and is by equation (4.4) independent of the surface itself. It is easy to check that

$$A = (a_0, a_1, a_2)^T = \frac{1}{2} \begin{pmatrix} 0 & -x_2 & x_1 \\ x_2 & 0 & -x_0 \\ -x_1 & x_0 & 0 \end{pmatrix} \tag{4.5}$$

satisfies the condition $\nabla \times a_i = e_i$. By calculation we see that for every $y \in \mathbb{R}^3$ we have $Ay = \frac{1}{2} x \times y$. The position vector $\frac{1}{2} x$ is called Darboux vector of the matrix A. More generally, for every skew-symmetric matrix B the vector v is called *Darboux vector* if it satisfies

$$By = v \times y \quad \text{for every } y \in \mathbb{R}^3 \tag{4.6}$$

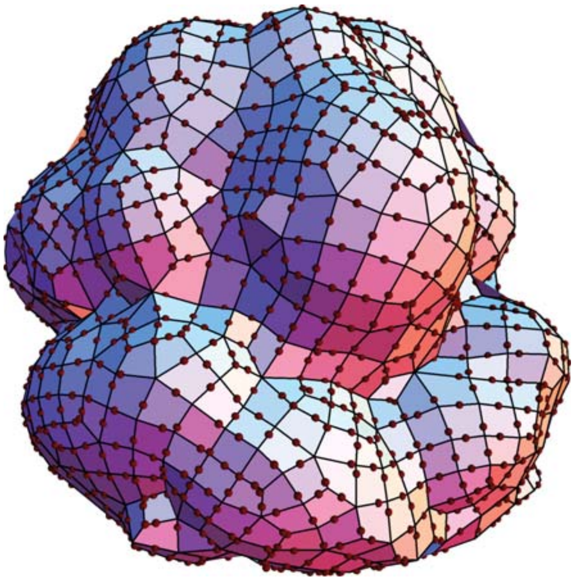

Figure (4.3): Edge midpoints inducing an orthogonal dual of a quad mesh.

and we write $[B] = v$. With this observation we obtain a compact expression for the vector area:

$$\int_S n \; dA = \sum_i \int_{\partial S} \langle a_i, dx \rangle = \int_{\partial S} A \; dx = \frac{1}{2} \int_{\partial S} x \times dx. \tag{4.7}$$

In the discrete case the right hand side of (4.7) becomes the sum of area-weighted triangle normal vectors (the vector area of a triangle). For the vector area of a general n-gon f with coordinates $X_f = (x_0, ..., x_{n-1})^T \in \mathbb{R}^{n \times 3}$ we have

$$a_f = \frac{1}{2} \sum_{i=0}^{n-1} x_i \times x_{i+1} \tag{4.8}$$

where the indices are cyclic, meaning $x_{i+1} = x_{((i+1) \mod n)}$. This convention will be used throughout this section. Note, that this formula is translation invariant, it does not depend on a specific choice of origin. With the vector area we have a generalization of area to non-planar polygons that is easy to compute — just choose an arbitrary tessellation of the polygon and sum up the individual vector areas. For a planar polygon the length of the vector area will consequently be the area of the polygon. For a non-planar polygon the length will correspond to the area of the polygon projected in the direction of the vector area. We will see in a moment that this projection direction is in fact the direction that produces the projection of largest area.

The idea of the polygon-Laplace operator is to compute the gradient of the vector area

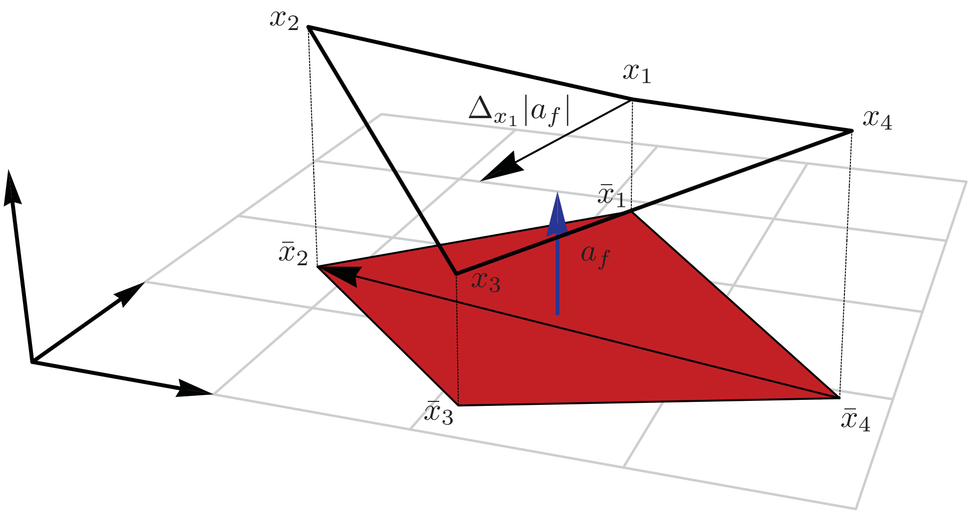

Figure (4.4): A nonplanar polygon $f = (x_1, x_2, x_3, x_4)$ is projected along its vector area a_f. Rotating the diagonal $(\bar{x}_4, \bar{x}_2)$ in the plane and scaling by 0.5 gives the vector area gradient with respect to x_1.

norm $|a_f|$ with respect to a vertex x_i analogous to Desbrun's approach (see section 3.5):

$$(\nabla_{x_i}|a_f|)^T = \left(\nabla_{x_i}\sqrt{\langle a_f, a_f\rangle}\right)^T = \frac{1}{2|a_f|}\,\langle\nabla_{x_i}(x_{i-1}\times x_i - x_{i+1}\times x_i), a_f\rangle \tag{4.9}$$

It is easy to construct matrices X_{i+1}, X_{i-1} with $[X_{i+1}] = x_{i+1}$ and $[X_{i-1}] = x_{i-1}$. We thus have

$$= \frac{1}{2|a_f|}\,\langle\nabla_{x_i}((X_{i-1} - X_{i+1})x_i), a_f\rangle = \frac{1}{2|a_f|}(X_{i-1} - X_{i+1})^T a_f \tag{4.10}$$

Using the definition of the Darboux vector we finally get for the area gradient

$$= \frac{1}{2|a_f|} a_f \times (x_{i-1} - x_{i+1}) = \frac{1}{2|a_f|} A(x_{i-1} - x_{i+1}) \tag{4.11}$$

A closer look at this last equation reveals the geometric meaning of the vector area (norm) gradient. Multiplication of the edge vector $(x_{i-1} - x_{i+1})$ by the matrix $\frac{1}{|a_f|}A$ comes down to calculating the cross product with $a_f/|a_f|$ yielding a vector orthogonal to both, the edge and the vector area, with length determined by the orthogonal projection of the edge onto the space orthogonal to a_f. The vector area gradient can thus be interpreted as a rotation of the projected edge $(x_{i-1} - x_{i+1})$ in the plane of maximal projection followed by scaling with $\frac{1}{2}$. This is exactly the area gradient of the projected polygon with respect to vertex x_i (figure 4.4). We can also see the fact, that the vector area points into the direction of largest projection. Since the gradient is just acting on the projected area, the projection direction has to be critical with respect to area. Moreover, it is by equation (4.8) apparent, that the polygon-Laplacian reduces to the cotan operator on triangle meshes since it is exactly the same construction as Desbrun's [Des99].

4.2.1. Implementation

The polygon Laplacian admits for a simple implementation in terms of the two matrices

$$E_f = (x_1 - x_0, \cdots, x_0 - x_{n-1})^T \text{ and } B_f = \frac{1}{2}(x_1 + x_0, \cdots, x_0 + x_{n-1})^T \tag{4.12}$$

encoding the polygon edges and edge-midpoints respectively. Note, that the operation $(x_i \times x_{i+1})\times$ gives the rotated component of a vector in the plane spanned by $(0, x_i, x_{i+1})$. This operation can be written as

$$p : v \mapsto x_{i+1}\langle x_i, v\rangle - x_i\langle x_{i+1}, v\rangle \tag{4.13}$$

since p projects a vector into the basis x_i, x_{i+1} and rotates it by $\pi/2$. Rotation by $\pi/2$ is archived by multiplication by the rotation matrix

$$\begin{pmatrix} \cos(\pi/2) & -\sin(\pi/2) \\ \sin(\pi/2) & \cos(\pi/2) \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{4.14}$$

For the operator $A = \frac{1}{2}\left(\sum_i x_i \times x_{i+1}\right) \times$ we can write

$$v \mapsto \frac{1}{2}\sum_i x_{i+1}\langle x_i, v\rangle - x_i\langle x_{i+1}, v\rangle = \frac{1}{2}\sum_i (x_{i+1} - x_i)\langle x_i + x_{i+1}, v\rangle \tag{4.15}$$

since the terms $x_i\langle x_i, v\rangle$ cancel each other cyclically. From this last equation we get

$$A_f = E_f^T B_f \tag{4.16}$$

Moreover, with the *coboundary operator* δ defined by $\delta X_f = E_f$ and by noting that $\left(B_f^T \delta\right)_i = x_{i-1} - x_{i+1}$ holds, we can rewrite equation (4.11) as

$$\left(\nabla_{X_f} |a_f|\right)^T = \frac{1}{|a_f|} A_f B_f^T \delta = \frac{1}{|a_f|} E_f^T B_f B_f^T \delta \tag{4.17}$$

Furthermore, by the definition of the coboundary operator we obtain

$$\nabla_{X_f}|a_f| = \frac{1}{|a_f|}\left((\delta X_f)^T B_f B_f^T \delta\right)^T = \frac{1}{|a_f|}\delta^T B_f B_f^T \delta X_f = \tilde{L}_f X_f. \tag{4.18}$$

The matrices $\tilde{L}_f$ can be assembled to build the matrix representation of the polygonal Laplacian. For triangle meshes this construction is equivalent to the traditional weak cotan operator. An implementation in Mathematica can be found in the Appendix A.1.

4.2.2. Properties of the Polygon Laplacian

By the representation (4.18) of the polygon Laplacian we immediately see that the operator is symmetric. Like in the triangle case positivity cannot be ensured. Note, that in the planar case the applying the polygon Laplacian to the geometry yields the same result as applying the cotan operator on an arbitrary tessellation of the mesh. Therefore the property (Lin) holds also for arbitrary polygons. Since the constant functions are in the kernel of δ and

$$u^T B B^T u = (B^T u)^T (B^T u) \geq 0 \tag{4.19}$$

the matrix $\tilde{L}$ is positive semi definite and the constants are in the kernel of the operator. Unfortunately the kernel is even bigger leading to unintuitive behavior of the Dirichlet energy. For a general n-gon with $n > 3$ the edge-midpoint matrix B_f^T will have rank ≤ 3 therefore admitting for a null space of dimension $\geq n - 3$. In order to overcome this problem, Alexa and Wardetzky [AW11] propose to add a matrix F to $B_f B_f^T$ that keeps the properties of the operator but fills up the null space leading to a positive semi definite Laplacian. In order to keep the symmetry, the additional matrix F should be symmetric. We have seen in section 3.7 that linear precision for a discrete Laplacian L amounts to having $(LX)_i = 0$ for every vertex i with planar 1-ring and the discrete coordinate function X. Denoting the planar projection of the polygon f in direction a_f as $\bar{f}$, the height vectors h, h_e, h_b are defined by

$$X_f = X_{\bar{f}} + h a_f^T, \quad B_f = B_{\bar{f}} + h_b a_f^T \quad \text{and} \quad E_f = E_{\bar{f}} + h_e a_f^T \tag{4.20}$$

Linear precision is preserved if we have for a planar region around vertex i

$$\begin{aligned} (F\delta X)_i = (FE_f)_i = (FE_{\bar{f}})_i &= 0 \\ \Leftrightarrow (E_{\bar{f}}^T F^T)_i &= 0 \end{aligned} \tag{4.21}$$

it is thus sufficient to require $\operatorname{im} F^T = \ker E_{\bar{f}}$ or equivalently $\ker F = \operatorname{im} E_{\bar{f}}$. With matrices C_f spanning the kernel of $E_{\bar{f}}^T$ and a positive definite, symmetric matrix U_f we have

$$F_f = C_f U_f C_f^T \tag{4.22}$$

as positive semi definite candidates for the additional term. To demonstrate that this approach successfully fills up the null space we have to show

Lemma 3. *For an arbitrary 1-form v we have*

$$v^T (B_f B_f^T + C_f U_f C_f^T) v > 0 \tag{4.23}$$

with matrices C_f whose kernel is spanning the image of $E_{\bar{f}}$ and U_f positive definite and symmetric.

Proof. It is sufficient to require

$$||B^T v|| > 0 \text{ and } ||C^T v|| > 0 \tag{4.24}$$

for every 1-form v. Let us assume that this is not true. From $C^T v = 0$ we get

$$v \in \ker C_f^T = \operatorname{im} E_{\bar{f}}. \tag{4.25}$$

Therefore there exists a $u \in \mathbb{R}^3$ with $v = E_{\bar{f}} u$. We have for the vector area:

$$E_f^T B_f = A_f = A_{\bar{f}} = E_{\bar{f}}^T B_{\bar{f}} \tag{4.26}$$

which cancels each other out in

$$\begin{aligned} E_f^T B_f &= \left(E_{\bar{f}}^T + a_f h_e^T\right)\left(B_{\bar{f}} + h_b a_f^T\right) \\ &= E_{\bar{f}}^T B_{\bar{f}} + E_{\bar{f}}^T h_b a_f^T + a_f h_e^T B_f \end{aligned}$$

Alexa and Wardetzky [AW11, Lemma 2] show that $E_{\bar{f}}^T h_b \equiv 0$ and we therefore have

$$B_f^T h_e \equiv 0. \tag{4.27}$$

With this in mind we have because of $B_f^t v = 0$:

$$0 = B_f^T E_{\bar{f}} u = B_f^T (E_{\bar{f}} + h_e a_f^T) u = B_f^T E_f u = A_f^T u = -A_f u = -a_f \times u. \tag{4.28}$$

Therefore u is parallel to a_f in contradiction to $v = E_{\bar{f}} u$ since a_f is perpendicular to the projected edges $E_{\bar{f}}$. $\square$

The discrete polygon Laplacian

$$L_f = \delta^t \left(\frac{1}{|a_f|} B_f B_f^T + C_f U_f C_f^T \right) \delta \tag{4.29}$$

is therefore linear precise, symmetric and positive semi definite with a 1-dimensional kernel consisting of the constant functions. Moreover, for a non-degenerate triangle f we have $\ker E_{\bar{f}}^T = \ker E_f^T \equiv 0$ and therefore $L_f = \tilde{L}_f$ which is equivalent to the classical cotan-formula. Alexa and Wardetzky [AW11] suggest to use an orthonormalized null-space of $E_{\bar{f}}$ for C_f and to set $U_f = \lambda \mathrm{Id}$.

4.2.3. Weighting

Can we find a weighting scheme for general polygons that generates a family of discrete polygon Laplacian analogous to the weighted cotan operator for triangle meshes (section 3.8)? For triangle meshes the answer is positive. With a matrix of weighted edge-midpoints

$$(B_f^w)^T = (b_0^w, \dots, b_{n-1}^w) \tag{4.30}$$

where b_i^w is the circumcenter of the i-th edge with respect to the power distance. With a real weight per vertex w_i we get (cf. equation 3.25)

$$b_i^w = x_i + d_{ij} \frac{(x_{i+1} - x_i)}{|x_{i+1} - x_i|} = x_i + \frac{(|x_{i+1} - x_i|^2 + w_i - w_j)(x_{i+1} - x_i)}{2|x_{i+1} - x_i|^2} \tag{4.31}$$

And we can show

Lemma 4. *The polygon Laplace construction yields the weighted cotan operator of Glickenstein [Gli08] if the term* $B_f B_f^T$ *is replaced by* $B_f (B_f^w)^T$

$$\tilde{L}_w = \delta^T B_f (B_f^w)^T \delta \tag{4.32}$$

Proof. The value w_{ij} of the operator is given by the dot product of a difference of edge-points.

$$\omega_{ij} = \frac{1}{|a_f|} \left\langle b_{i-1}^w - b_i^w, b_j - b_{j+1} \right\rangle$$

By expanding the edge mid-points (equations 4.31 and 4.12)

$$
\begin{aligned}
&= \frac{1}{|a_f|}\left\langle d_{i,i+1}\frac{x_i - x_{i+1}}{|x_1 - x_{i+1}|} + d_{i,i-1}\frac{x_{i-1} - x_i}{|x_{i-1} - x_i|}, \frac{1}{2}(x_{i+1} - x_i) + \frac{1}{2}(x_{j+1} - x_j)\right\rangle \\
&= \frac{1}{2|a_f|}\left\langle d_{i,i+1}\frac{x_i - x_{i+1}}{|x_1 - x_{i+1}|} + d_{i,i-1}\frac{x_{i-1} - x_i}{|x_{i-1} - x_i|}, x_{j+1} - x_i\right\rangle
\end{aligned}
$$

Note, that we have for a triangle $x_{i+1} = x_j$ and $x_{j+1} = x_{i-1}$. Moreover by definition of the scalar product

$$
\begin{aligned}
&= \frac{1}{2|a_f|}\left(d_{i,i-1}|x_{i-1} - x_i| - d_{i,i+1}|x_{j+1} - x_i|\cos\gamma_i\right) \\
&= \frac{d_{i,i-1}|x_{i-1} - x_i| - d_{i,i+1}|x_{j+1} - x_i|\cos\gamma_i}{|x_{i-1} - x_i||x_{i+1} - x_i|\sin\gamma_i} \\
&= \frac{d_{i,i-1} - d_{i,i+1}\cos\gamma_i}{|x_{i+1} - x_i|\sin\gamma_i}
\end{aligned}
$$

which is exactly the formula of Glickenstein (cf. equation 3.24). □

Unfortunately, this construction does not extended to higher polygons because the symmetry is violated.

Example. Consider the quadrilateral

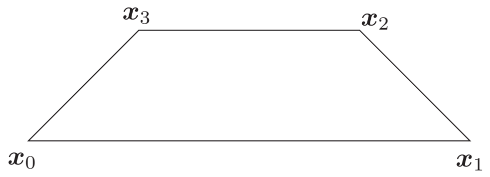

Figure (4.5): A quadrilateral.

x_0	(-2, 0)
x_1	(2, 0)
x_2	(1, 1)
x_3	(-1, 1)

Table (4.1): Coordinates

We have the edge-midpoint matrix B. With the length $\tilde{d}_{ij} = d_{ij}/|x_i - x_j|$ we get the general weighted midpoint matrix B_W.

$$
B = \begin{pmatrix} 0 & 0 \\ \frac{3}{2} & \frac{1}{2} \\ 0 & 1 \\ -\frac{3}{2} & \frac{1}{2} \end{pmatrix} \qquad B_W = \begin{pmatrix} -2 + 4\tilde{d}_{01} & 0 \\ 2 - \tilde{d}_{12} & \tilde{d}_{12} \\ 1 - 2\tilde{d}_{23} & 1 \\ -1 - \tilde{d}_{30} & 1 - \tilde{d}_{30} \end{pmatrix}
$$

Symmetry occurs if $L_W - L_W^T = 0$ holds. This is true for

$$\boxed{\begin{aligned}\tilde{d}_{12} &= 2 - 3\tilde{d}_{01} \\ \tilde{d}_{23} &= 1 - \tilde{d}_{01} \\ \tilde{d}_{30} &= 2 - 3\tilde{d}_{01}\end{aligned}}$$

We have one degree of freedom (i.e. choosing $\tilde{d}_{01}$). B is obviously a special case of B_W ($\tilde{d}_{01} = 1/2$). Fixing one weighted edge-point pins down all other midpoints of that polygon. In order to archive linear precision we have to choose weighted edge-points on two edge-neighboring polygons consistently. For a general quad-mesh this leaves no other choice than $B_W = B$. A notable exception is the cube, where one can choose consistent weights because of the symmetry of the mesh. The question whether it is possible to choose weights that yield a *symmetrizable matrix*, a matrix that can be decomposed into a full-rank diagonal matrix and a symmetric matrix, remains unanswered.

4.2.4. Alternative Polygon Laplacian

The main idea for the construction of a cotan-like Laplacian on general polygonal meshes was to require that the operator applied to the geometry vector should yield the vector area gradient. In a second step an additional term was added in order to ensure the property Psd. Let $X_f = X_{\bar{f}} + h_f a_f^T$ be the vertex vector of a polygon f. We have

$$\begin{aligned} L_f X_f &= \delta^T \left(B_f B_f^T + C_f U_f C_f^T\right) \delta \left(X_{\bar{f}} + h_f a_f^T\right) \\ &= \delta^T B_f B_f^T E_{\bar{f}} + \delta^T C_f U_f C_f^T h_e a_f^T + \underbrace{\delta^T B_f B_f^T h_e a_f^T}_{=0 \text{ by eq. } 4.27} + \underbrace{\delta^T C_f U_f C_f^T E_{\bar{f}}}_{=0 \text{ by def. of } C_f}. \end{aligned} \tag{4.33}$$

The operator has two components, one acting on the maximal planar projection of the polygon $E_{\bar{f}}$ and computing the vector-area gradient and one component acting on the height vector of the polygon over the maximal projection $h_e a_f^T$.

Following this approach we can construct a polygonal Laplacian L by ensuring that the planar component of LX yields the vector area gradient.

1. Project the polygon in direction of the area vector.
2. Triangulate the projected polygon.
3. Build the matrix of cotangent weights for the polygon by summing over the triangles.

Note, that the weights will differ depending on the chosen triangulation, however, the mean curvature vector will be independent of the triangulation since the projected polygon is planar. The Laplacian for the whole mesh can be assembled from the individual face operators. The operator is symmetric because the weights ω_{ij} are summed up from the

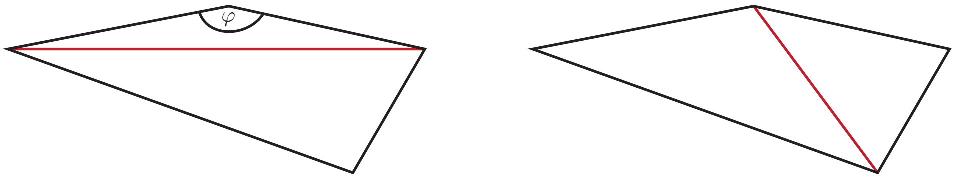

Figure (4.6): Two possible triangulations of a quadrilateral leading to different maximal interior angles.

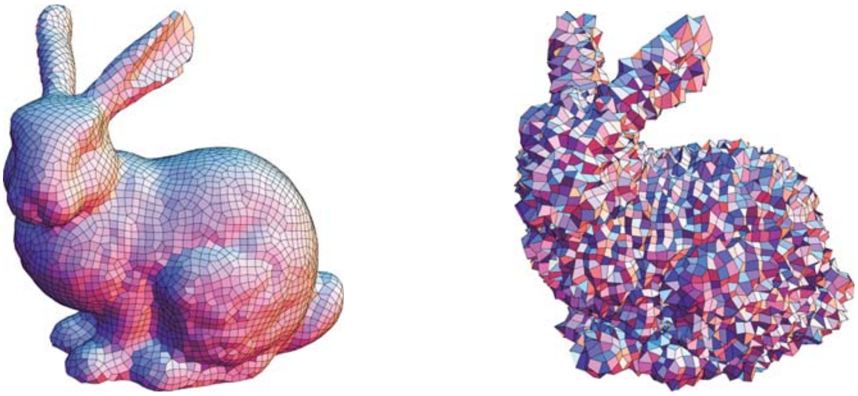

Figure (4.7): A quad mesh of the Stanford bunny with 5k vertices. For the left version noise in the normal direction was added.

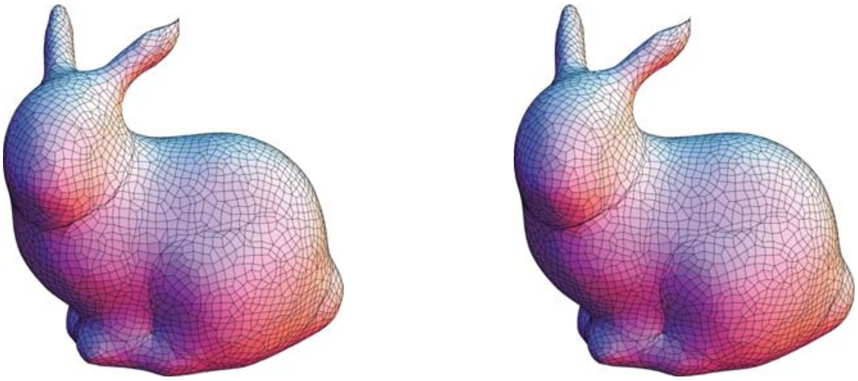

Figure (4.8): Result of implicit mean curvature flow applied to the noisy bunny (figure 4.7) with the planar operator (left) and the polygon Laplacian (right).

two adjacent polygons. Linear precision is ensured because the operator equals the cotan operator on planar meshes. The property Psd is also inherited from the triangle cotan operator. As for triangles we cannot guarantee positive weights, however, it is possible to significantly reduce negative weights by choosing a "good" triangulation of the polygon.

E.g. for a quadrilateral there are two ways of triangulating by choosing one of the two diagonals (see figure 4.6). The value $\cot\varphi$ will be negative since $\varphi > \pi/2$. The second triangulation will produce no negative weight. For triangles with good aspect ratio this

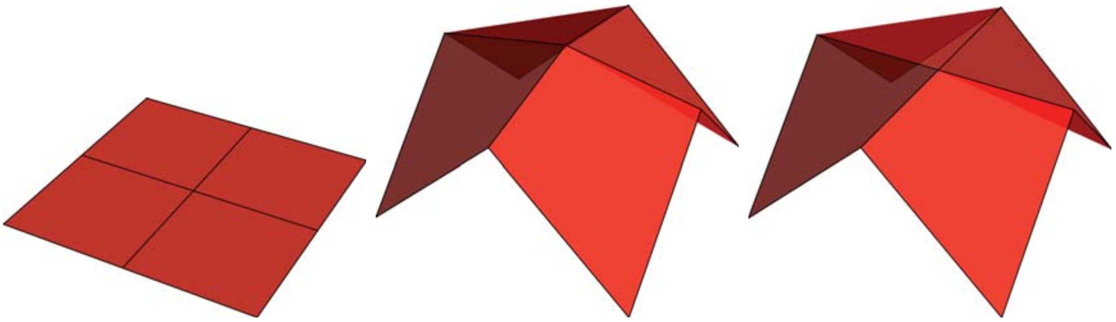

Figure (4.9): A simple quadrilateral mesh with a single interior vertex (left), prescribed values at the boundary and vanishing Laplacian at the interior vertex for the polygon Laplacian (center), the same boundary values with the planar Laplacian vanishing for the interior vertex (right).

operator can significantly reduce the number of negative weights. For the noisy bunny (4.7 right) with 5000 vertices the polygon cotan operator has 23950 negative weights whereas the planar formulation has 680 negative weights. For the smooth bunny (4.7 right) the difference is even more striking with 20714 for the polygon Laplacian and 30 for the planar version. The following example illustrates the effect of negative weights.

Example. The positivity of the Laplacian coefficients ω_{ij} is related to the discrete maximum principle, as detailed in section 3.2. For a surface with fixed boundary and vanishing discrete Laplacian at all interior vertices the discrete maximum principle requires that the extrema of each coordinate function are attained at the boundary. Since the polygon Laplacian of Alexa and Wardetzky [AW11] is not guaranteed to have only positive entries the discrete maximum principle might be violated as depicted in figure 4.9 (center). The planar polygon Laplacian has no negative entries in this particular case and consequently obeys the maximum principle (4.9 right).

Figure 4.8 compares the original polygon Laplacian and the proposed version in a mesh smoothing application. The results are almost indistinguishable.

Bibliography

[AW11] Marc Alexa and Max Wardetzky. “Discrete Laplacians on general polygonal meshes”. *ACM Trans. Graph.* 30 (4 Aug. 2011), 102:1–102:10.

[Cou77] R. Courant. *Dirichlet’s principle, conformal mapping, and minimal surfaces: reprint*. Springer, 1977.

[dCa76] M.P. do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall, 1976.

[dCa94] M.P. do Carmo. *Differential Forms and Applications*. Universitext (1979). Springer-Verlag GmbH, 1994.

[Des99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. “Implicit fairing of irregular meshes using diffusion and curvature flow”. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 317–324.

[Duf59] R.J. Duffin. “Distributed and lumped networks”. *Journal of Mathematics and Mechanics* 8.5 (1959), 793–826.

[Gli07] David Glickenstein. “A Monotonicity Property for Weighted Delaunay Triangulations”. *Discrete Comput. Geom.* 38.4 (Dec. 2007), pp. 651–664.

[Gli08] David Glickenstein. “Geometric triangulations and discrete Laplacians on manifolds” (2008), pp. 1–43.

[Gor92] Carolyn Gordon, David L. Webb, and Scott Wolpert. “One cannot hear the shape of a drum” (1992).

[Hir03] Anil N. Hirani. “Discrete Exterior Calculus”. PhD thesis. California Institute of Technology, Mar. 2003.

[JZ07] Varun Jain and Hao Zhang. “Zhang H.: A spectral approach to shapebased retrieval of articulated 3d models”. *CAD* 39 (2007), pp. 398–407.

[Kac66] Mark Kac. “Can One Hear the Shape of a Drum?” *The American Mathematical Monthly* 73.4 (1966), pp. 1–23.

[LZ10] Bruno Lévy and Hao (Richard) Zhang. “Spectral mesh processing”. *ACM SIGGRAPH 2010 Courses*. SIGGRAPH ’10. Los Angeles California: ACM, 2010, 8:1–8:312.

[Mul11] Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. “HOT: Hodge-optimized triangulations”. *ACM SIGGRAPH 2011 papers*. SIGGRAPH ’11. Vancouver British Columbia Canada: ACM, 2011, 103:1–103:12.

[PP93] Ulrich Pinkall and Konrad Polthier. “Computing discrete minimal surfaces and their conjugates”. *Experim. Math.* 2 (1993), pp. 15–36.

[Reu06] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. “Laplace-Beltrami spectra as ”Shape-DNA” of surfaces and solids”. *Computer-Aided Design* 38.4 (2006), pp. 342–366.

[Ros97] S. Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds*. London Mathematical Society Student Texts. Cambridge University Press, 1997.

[Sor05] Olga Sorkine. “Laplacian Mesh Processing”. Ed. by Yiorgos Chrysanthou and Marcus Magnor. Eurographics 05 STAR. Dublin, Ireland: Eurographics Association, Sept. 2005, pp. 53–70.

[Tau95] Gabriel Taubin. “A signal processing approach to fair surface design”. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’95. New York, NY, USA: ACM, 1995, pp. 351–358.

[VL08] Bruno Vallet and Bruno Lévy. “Spectral Geometry Processing with Manifold Harmonics”. *Computer Graphics Forum (Proceedings Eurographics)* (2008).

[War07] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. “Discrete laplace operators: no free lunch”. *Proceedings of the fifth Eurographics symposium on Geometry processing*. Barcelona Spain: Eurographics Association, 2007, pp. 33–37.

[War08] Max Wardetzky. “Convergence of the Cotan formula - an overview”. *Discrete Differential Geometry*. Ed. by Alexander I. Bobenko, John M. Sullivan, and Peter Schröder ans Günter Ziegler. Oberwolfach Seminars. 2008.

[Zha04] Hao Zhang. “Discrete combinatorial Laplacian operators for digital geometry processing”. *in SIAM Conference on Geometric Design, 2004*. Press, 2004, pp. 575–592.

A. Listings

```
PolyCotan[x_, faces_] := Module[{dmat, cnt, res, lambda},
   cnt = 0;
   lambda = 1.;

   (* Compute per polygon data *)
   res =
    Module[{n, E, B, A, a, area, Xp, Ep, M1, C, L, ret, coords, d, rules},
       n = Length@#;
       (* Get the coordinates per face *)
       coords = Extract[x, Partition[#, 1]];
       E = ListConvolve[{{1.}, {-1.}}, coords, -1];
       B = ListConvolve[{{.5}, {.5}}, coords, -1];

       (* Assemble M1, compute vector area 'a' *)
       A = Eᵀ.B;
       a = {-A[[2, 3]], A[[1, 3]], -A[[1, 2]]};
       area = Norm[a];
       a /= area;
       M1 = 1.0/area * B.Bᵀ;

       (* Project polygon along the vector area *)
       Xp = (# - a (a.#)) & /@ coords;
       Ep = ListConvolve[{{1.}, {-1.}}, Xp, -1];
       C = Transpose@Orthogonalize@NullSpace[Epᵀ];

       M1 += lambda C.Cᵀ;

       (* Return M1 and d in triplet form *)
       rules = Drop[ArrayRules[M1], -1];
       rules[[;;, 1]] += cnt;
       d = Table[{{cnt + j, #[[j]]} -> 1., {cnt + j, #[[Mod[j, n] + 1]]} -> -1.}
         , {j, n}];

       ret = {d, rules};

       cnt += n;
       ret
      ] & /@ faces;

   (* Assemble & return final matrix *)
   dmat = SparseArray@Flatten@res[[;;, 1]];
   Return@(dmatᵀ.(SparseArray@Flatten@res[[;;, 2]]).dmat);
];
```

Listing (A.1): Implementation of the polygonal Laplace operator by Alexa and Wardetzky [AW11] in Mathematica. The input arguments are assumed to be a list of coordiate triples and a list of faces represented by indices into the vertex list.

```
dualMesh[file_String] := Module[{mesh, v, f, edgeIndizes, cnt, faceCnt, b,
    triplets, bary, sys, sol, ker, c},

   (* load mesh *)
   mesh = Import[file];
   v = mesh[[1, 2, 1]];
   f = mesh[[1, 2, 2, 1, 1]];

   (* choose an integer index per unoriented edge *)
   edgeIndizes = Flatten[♯, 1] &@(Table[{♯[[i]], ♯[[ Mod[i, Length[♯]] + 1]]},
         {i, Length[♯]}] & /@ f);

   edgeIndizes = Select[edgeIndizes, ♯[[1]] > ♯[[2]] &];
   edgeIndizes = MapIndexed[♯1 -> First@♯2 &, edgeIndizes];

   (* build the system *)
   cnt = 1;
   faceCnt = 0;
   b = Table[0, {Length@Flatten@f}];
   triplets =
    Flatten@
     (Module[{ret},
        ret = Table[Module[{e, edgeInd, k, l},
           edgeInd = Sort[{♯[[i]], ♯[[ Mod[i, Length[♯]] + 1]]}, Greater];
           e = v[[edgeInd[[2]]]] - v[[edgeInd[[1]]]];
           b[[cnt]] = v[[edgeInd[[1]]]].e;
           {{cnt, 3 Length[f] + (edgeInd /. edgeIndizes)} -> -Norm[e],
            { cnt, 3 faceCnt + 1} -> e[[1]],
            { cnt, 3 faceCnt + 2} -> e[[2]],
            { cnt++, 3 faceCnt + 3} -> e[[3]]}
          ], {i, Length[♯]}];
        ++faceCnt;
        ret
       ] & /@ f);

   (* barycenters *)
   bary = (1. / Length[♯] Plus @@ Extract[v, Partition[♯, 1]]) & /@ f;
   sys = SparseArray@triplets;

   (* solve system *)
   sol = LeastSquares[sys, b];
   ker = NullSpace[sys];

   (* optimize dual vertex positions *)
   c = LeastSquares[ker[[ ;; , 1 ;; 3 Length[f]]]ᵀ,
     Flatten@bary - sol[[1 ;; 3 Length[f]]]];
   sol += kerᵀ.c;
   Return[sol];
  ];
```

Listing (A.2): Implementation of the dual mesh computation for polygonal meshes as described in section 4.1.