The Visual Computer
(to appear 2007)

Johannes Wallner · Helmut Pottmann · Michael Hofer

Fair Webs

Abstract Fair webs are energy-minimizing curve networks. Obtained via an extension of cubic splines or splines in tension to networks of curves, they are efficiently computable and possess a variety of interesting applications. We present properties of fair webs and their discrete counterparts, i.e., fair polygon networks. Applications of fair curve and polygon networks include fair surface design and approximation under constraints such as obstacle avoidance or guaranteed error bounds, aesthetic remeshing, parameterization and texture mapping, and surface restoration in geometric models.

Keywords Curve network · Spline interpolation and approximation · Variational surface design · Curves in surfaces · Aesthetic meshes · Obstacle avoidance · Texture mapping · Mesh fairing

1 Introduction

We present a variational approach to the design of energy minimizing curve networks that are constrained to lie in a given surface or to avoid a given obstacle. Such constrained energy minimizing curve networks are called *fair webs*. These curve networks generalize the previous work of [14, 33] on energy-minimizing spline curves in surfaces. There are no restrictions on the dimension of the surface, the dimension of ambient space or the type of surface representation. In this paper we contribute theoretical results on properties of energy-minimizing curve networks in surfaces and their discrete counterparts, fair polygon networks. We illustrate the usefulness of fair webs and fair polygon networks by means of several applications such as the design of fair surfaces in the presence of obstacles, remeshing with fairness properties for applications in art and design (Fig. 1), surface parameterization, and hole filling in geometric models.

J. Wallner, H. Pottmann, and M. Hofer
Vienna University of Technology
Institute of Discrete Mathematics and Geometry
Wiedner Hauptstr. 8-10/104, 1040 Wien, Austria
Tel.: +43 1 58801 11327, Fax: +43 1 58801 11399
E-mail: {wallner,pottmann,hofer}@geometrie.tuwien.ac.at

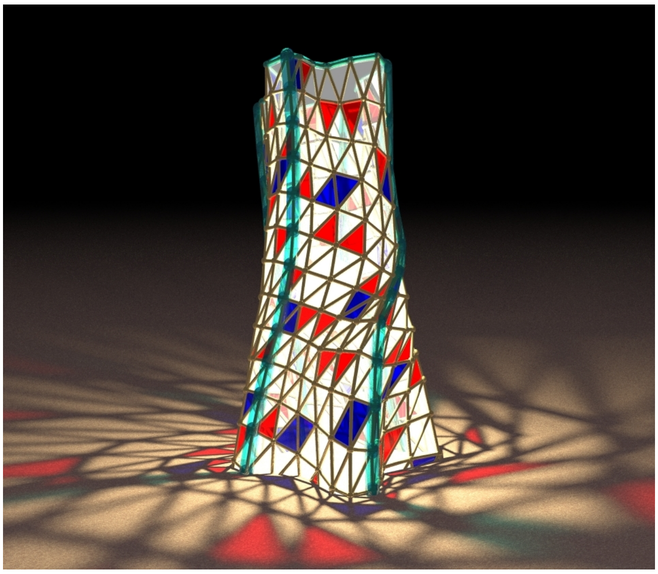


Fig. 1 Design based on a fair mesh.

1.1 Previous Work

Variational design of curves and surfaces is a well studied subject, see e.g. [6, 8, 26, 42] and the references therein. Energy-minimizing curve networks using the cubic spline energy or generalizations of it have served for the reconstruction of a bivariate function from scattered data [28, 29, 20]. There is also prior art on surface design based on energy-minimizing curve networks, e.g. [19, 26]. In the discrete setting variational subdivision has been addressed in [18].

Note that there are many contributions dealing with curve networks that do not explicitly use a variational approach. For completeness we cite a few recent contributions on that topic. 'Wires' [37] are sets of curves used for freeform deformations, combined subdivision schemes [22] exactly interpolate a network of curves given in any parametric representation, lofted subdivision surfaces [34] approximate a network of curves, and the multiresolution subdivision surfaces of [3] use a set of user-defined curves to create sharp features and trim regions along them. A set of polygonal curves embedded in a meshed surface has been used by [23] for mesh editing purposes.

While curve networks in 3-space are widely used and several contributions exist that use a variational approach, much less is known about energy-minimizing curves or curve networks in surfaces, and in the presence of obstacles. Some authors, e.g. [31], discuss the minimizers of the energy defined by the second covariant derivative with respect to arc length. These *intrinsic* splines will not be the topic of this paper. Our work is an extension of energy minimizing curves in surfaces as discussed at first by [5] and later by [14] where an *extrinsic* formulation of the energy is used. Fair polyline networks with an extrinsic energy formulation have been employed successfully for constrained smoothing of digital terrain elevation data [15].

The present work has also been inspired by research on active contours [4], especially by work on active curves and geometric flows of curves on surfaces, see e.g. [11,25]. Active curve *networks* have been used for image metamorphosis [21] and texture analysis [24]. The *geodesics* of a surface are those curves "c" which minimize the arc length $\int \|c'\|$ (the prime indicates differentiation with respect to the curve parameter). The same curves, if traversed with constant velocity, arise as minimizers of the functional $\int \|c'\|^2$. A variety of applications of geodesics has been described within Computer Vision and Image Processing [16]. We are going to minimize a similar functional defined on many curves simultaneously.

1.2 Contributions and Overview

In the present paper we study fair curve and polygon networks constrained to lie in surfaces or avoiding obstacles. In Section 2 we define curve networks and different energies for curve networks. Using variational calculus we prove theoretical results about fair webs. In Section 3 we introduce an appropriate notation which allows us to define discrete energies of polygon networks. Using the theoretical insights gained in the study of fair curve networks we are able to show properties of fair polygon networks, which can be seen as discretizations of fair curve networks. We also discuss a special case of fair polygon networks, namely fair meshes. In Section 4 we discuss computational issues that arise with the implementation of fair polygon networks. Furthermore, we illustrate that fair webs have a variety of interesting applications including fair remeshing, variational surface design in the presence of obstacles, surface approximation with guaranteed error bounds, surface parameterization and texture mapping, and surface restoration of geometric models containing holes. We conclude the paper in Section 4.3 with an outlook towards future research.

2 Fair Curve Networks

In this section we define curve networks, their connectivity, and the energies we are working with. Since this paper is about fair webs, i.e., energy minimizing curve networks, we use variational calculus to derive theoretical results about fair curve networks constrained to lie in a surface. These results are in the spirit of the paper [14], where similar results have been derived for the case of a single energy minimizing curve constrained to a surface.

2.1 Connectivity of a Curve Network

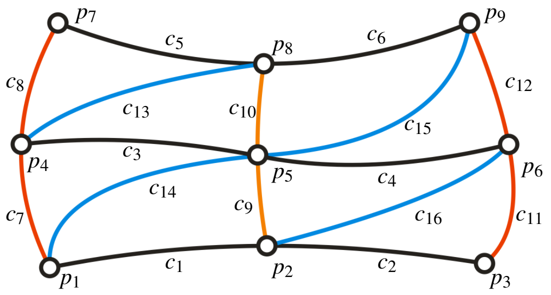
A *curve network* is a finite set of curves $\mathscr{C} = \{c\}$, each defined in its parameter interval $[a_c, b_c]$. We call points which are common to more than one curve *knots*. One can think that the curves are being knotted together at the knots. Note that by "curve" we actually mean a curve segment between two knot points of the curve network. As in the familiar case of splines, some of these curves will later be joined to larger curves, called *structure lines*.

Each *knot* k of the curve network has a collection C_k^s of curves *starting* there and another set C_k^e of curve segments *ending* there. The location of the knot in space is some point p_k. So if a curve c, defined in the interval $[a_c, b_c]$ starts in the knot p_k, i.e., $c \in C_k^s$, then $c(a_c) = p_k$, and analogously, if a curve c ends in the knot p_k, i.e., $c \in C_k^e$, then $c(b_c) = p_k$. The knots p_k together with the sets C_k^s, C_k^e define the *connectivity* of the network. Later on we will refer to the curves in the set C_k^s also as the *outgoing* curves of the knot p_k, and the curves in the set C_k^e as the *incoming* curves of p_k.

Figure 2 shows an example of a curve network defined by sixteen curves $c_1, \ldots, c_{16}$. The connectivity of the curve network is given by the nine knots $p_1, \ldots, p_9$, the nine sets of incoming curves $C_1^e, \ldots, C_9^e$ and the nine sets of outgoing curves $C_1^s, \ldots, C_9^s$:

2.2 Structure Lines

To obtain nice polygon networks we often want that two curve segments joining in a knot (an incoming curve and an outgoing one) actually belong to one larger curve. More formally we state: a curve $c_e \in C_k^e$ ending in a knot p_k together with a curve $c_s \in C_k^s$ starting in p_k may be required to form a single smooth curve. These two curve segments (c_e, c_s) are



$C_1^s = \{c_1, c_7, c_{14}\}$, $C_1^e = \{\}$,
$C_2^s = \{c_2, c_9, c_{16}\}$, $C_2^e = \{c_1\}$,
$C_3^s = \{c_{11}\}$, $C_3^e = \{c_2\}$,
$C_4^s = \{c_3, c_8, c_{13}\}$, $C_4^e = \{c_7\}$,
$C_5^s = \{c_4, c_{10}, c_{15}\}$, $C_5^e = \{c_3, c_9, c_{14}\}$,
$C_6^s = \{c_{12}\}$, $C_6^e = \{c_4, c_{11}, c_{16}\}$,
$C_7^s = \{c_5\}$, $C_7^e = \{c_8\}$,
$C_8^s = \{c_6\}$, $C_8^e = \{c_5, c_{10}, c_{13}\}$,
$C_9^s = \{\}$, $C_9^e = \{c_6, c_{12}, c_{15}\}$.

Fig. 2 Example of a curve network given by curves $c_1, \dots, c_{16}$. The connectivity is defined by the knots $p_1, \dots, p_9$, the sets $C_1^s, \dots, C_9^s$ of outgoing curves, and the sets $C_1^e, \dots, C_9^e$ of incoming curves.

part of what we call a *structure line* of the curve network. If two curves sharing a knot belong to a structure line, we say that these two curves have property (S). For a knot p_k, we collect all outgoing curves that have the property (S) in a set C_k^*. All outgoing curves $c \in C_k^s$ without property (S) are collected in a set C_k^{s*}, and the incoming curves $c \in C_k^e$ without property (S) form the set C_k^{e*}.

For the example shown in Fig. 2 we would e.g. choose the following structure lines: (c_1, c_2), (c_3, c_4), (c_5, c_6), (c_7, c_8), (c_9, c_{10}), (c_{11}, c_{12}), (c_{14}, c_{15}). Note that structure lines can also consist of more than two curve segments. Actually in our examples we use structure lines that connect many more than two curve segments, see e.g. Fig. 3.

2.3 Energy of a Curve Network

Energies E_1 and E_2 of a curve c are defined by

$$E_1(c) := \int_{a_c}^{b_c} \|c'(t)\|^2 dt, \quad E_2(c) := \int_{a_c}^{b_c} \|c''(t)\|^2 dt. \qquad (1)$$

E_2 is the cubic spline energy. Here $c'(t)$ and $c''(t)$ denote the first and second derivative vectors of the curve $c(t)$ with respect to the parameter t. Further we use a linear combination of E_1 and E_2, the tension energy E_τ,

$$E_\tau = E_2 + \tau E_1 \qquad (2)$$

where τ is a tension parameter. The energy of the entire curve network $\mathscr{C}$ is the sum of energies of all single curves of the curve network:

$$E(\mathscr{C}) = \sum\nolimits_{c \in \mathscr{C}} E(c). \qquad (3)$$

Here E is either of E_1, E_2, or E_τ. We refer to energy minimizing curve networks also as *fair* curve networks or short *fair webs*. Fair webs minimizing the energies E_2, $E_1 + 0.2E_1$, and E_1 are shown in Fig. 3.

2.4 Geometric Theory of Fair Webs in Surfaces

Fair webs constrained to surfaces have a number of nice properties. The most important from the fairness point of view is, that for E_2-and E_τ-minimizing fair webs, a structure line (which originally is required to be smooth only) is actually C^2. Further, the derivatives of the curve segments which join in a free knot (whose location is not fixed as a side condition) fulfill some balance equations, which are detailed below. A discretized network (see Section 3) has analogous properties. We consider fair webs whose curves are constrained to a given surface Φ. Although we assume that the connectivity of the fair web is maintained, we have the freedom to choose which knot points shall be *fixed* (their position is chosen, e.g. by the user), and which knot points shall be *free* (their position will be determined by the energy minimization procedure). In Fig. 3, fixed knots are marked.

The spirit of the theoretical results is like the familiar one from differential geometry that E_1-minimizing curves have the second derivative vectors c'' orthogonal to Φ, see e.g. [10]. We consider small variations of the curve network which obey the constraints, i.e., the connectivity of the curve network does not change, all curves stay in the surface Φ, and the fixed knots are maintained.

A smooth variation of the network — so that for each curve c, the point $c(t)$ depends on a second *variation parameter* u and we write $c^{(u)}(t)$ — leads to a total energy $E(u) = \sum_c E(c^{(u)})$ dependent on u. If at $u = 0$ the network is in a minimum position, then $\frac{dE}{du} = 0$ regardless of the particular variation we choose. There might be other, non-minimal, positions of the network with the same property. They are called *stationary* positions. It is impossible to tell a minimum position from other stationary ones without analysis of $\frac{d^2}{du^2}E(0)$, which already in the case of E_1-stationary curves is a rather involved theory (cf. do Carmo [9], §9.2).

We confine ourselves to C^2 curve segments in the case of E_1 and to C^4 segments in the case of E_2 and the tension energy E_τ. A vector V attached to a point p in Φ can be written as $V = V^\top + V^\perp$, where $V^\top$ is tangent and $V^\perp$ is orthogonal to Φ in the point p, see Fig. 4. The vectors $c'(t), c''(t), \dots$ are attached to the point $c(t)$. Now we prove theorems that give a characterization of E-stationary curve networks for the different energies that we consider in this paper. We begin with the energy E_1.

Theorem 1 *E_1-stationary curve networks in a surface Φ are characterized by:*

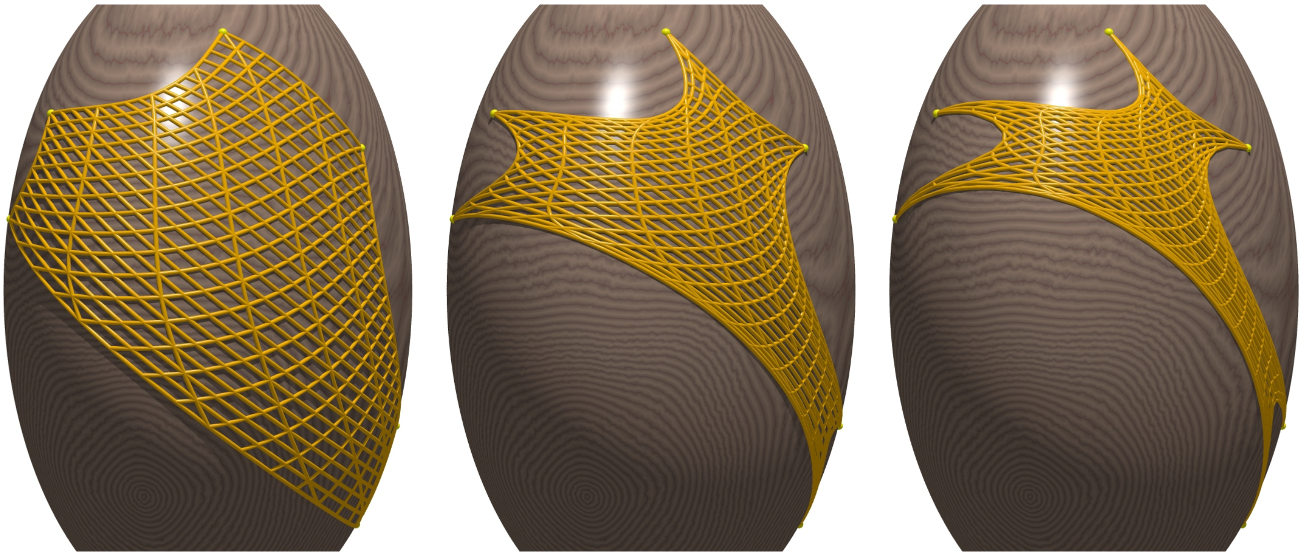


Fig. 3 Curve networks on a surface which minimize an energy E. From left to right: $E=E_2$, $E=E_2+0.2E_1$, $E=E_1$. Fixed knots are marked by a ball. All other knots are free. The structure lines of the fair web reveal themselves beautifully.

(i) The second derivative vectors c'' of all curves c are orthogonal to the surface Φ, i.e.,

$$c''^{\top}=0$$

for all curves.

(ii) For each free knot k,

$$T_{1,k}:=\sum_{c\in C_k^e} c'(b_k)-\sum_{c\in C_k^s} c'(a_k)=0, \qquad (4)$$

i.e., the first derivative vectors of incoming and outgoing curves are in equilibrium.

Proof A standard method in the calculus of variations is to consider a smooth variation of the network as described above and express $\frac{dE_1}{du}$ in terms of the *variation vector field*

$$V_c(t):=\frac{\partial c^{(u)}(t)}{\partial u}$$

at $u=0$, i.e., the initial velocity of the curve point $c(t)$ as it is subject to variation. For the curve c, we use integration by parts to compute

$$\begin{aligned}\frac{dE_1(c^{(u)})}{du} &= \frac{d}{du}\int_{a_c}^{b_c} c'\cdot c'\,dt=\int_{a_c}^{b_c} V_c'\cdot c'\,dt\\ &=[V_c\cdot c']_{a_c}^{b_c}-\int_{a_c}^{b_c} V_c\cdot c''\,dt.\end{aligned}$$

We sum up the contributions of the single curves in order to compute the derivative $\frac{dE_1}{du}(0)$ of the total energy. At each knot k where curves come together, the variation vector field has the same value $V(k)$ for all curves, as the variation must respect the fact that these curves are meeting at p_k. So we get

$$\begin{aligned}\frac{dE_1}{du} = &\sum_k V(k)\cdot\Big(\sum_{c\in C_k^e} c'(b_k)-\sum_{c\in C_k^s} c'(a_k)\Big)\\ &-\sum_{c\in\mathscr{C}}\int_{a_c}^{b_c} V_c\cdot c''. \end{aligned} \qquad (5)$$

Note that $V_c\cdot c''=V_c\cdot c''^{\top}$. We use a variation where V_c is a positive multiple of $c''^{\top}$ outside the knots and zero in the knots. Then the first part of Equ. (5) vanishes and we get a nonzero value of $\frac{dE_1}{du}$ unless $c''^{\top}=0$. This proves (i) of Th. 1. Now we know that the $\int\ldots$ term in Equ. (5) is zero for an E_1-stationary network, and we use another type of variation to show part (ii) of Th. 1. The $V(k)'s$ at different knots are independent of each other and can assume any value if the knot is not fixed. So we conclude that the right hand term in brackets $(\ldots)$ in Equ. (5) must vanish if $\frac{dE_1}{du}$ is to be zero. This is (ii) of Th. 1. It is obvious that the geometric characterization derived here is not only necessary but also sufficient for a network to be in an E_1-stationary position. □

For curves in C_k^* which represent an incoming/outgoing pair and are therefore part of a structure line, the symbol $\Delta c''$ means the jump in c'' when leaving the incoming curve and continuing at the outgoing one, and similar for $\Delta c'''$. The next theorem describes curve networks which make the energy E_2 stationary.

Theorem 2 *E_2-stationary curve networks in a surface Φ are characterized by:*

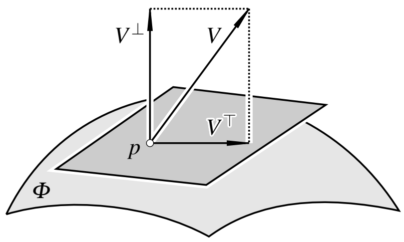


Fig. 4 A vector V attached to a point p on a surface Φ has a component $V^\perp$ orthogonal to Φ and a component $V^\top$ tangential to Φ.

(i) The fourth derivative vectors c'''' of all curves c are orthogonal to the surface Φ, i.e.,

$$c''''^\top = 0$$

for all curves.

(ii) For each knot k, curves in C_k^{e*} and C_k^{s*} have $c''^\top = 0$ there, and the curves in C_k^* which are part of a structure line have $\Delta c'' = 0$ in the knot.

(iii) For each free knot k, $T_{2,k}^\top = 0$, where

$$T_{2,k} := (\sum_{c\in C_k^{e*}} - \sum_{c\in C_k^{s*}})(c''^{\perp\prime} + c''') - \sum_{c\in C_k^*} \Delta c'''. \tag{6}$$

Proof We proceed in a way similar to the proof of Th. 1 for the energy E_1. The difference to E_1 is that we do integration by parts twice, and that the contributions of the single knots are somewhat more involved. It is elementary that for a single curve

$$\frac{dE_2(c^{(u)})}{du} = [V_c' \cdot c'']_{a_c}^{b_c} - [V_c \cdot c''']_{a_c}^{b_c} + \int_{a_c}^{b_c} V_c \cdot c'''' \, dt.$$

For the entire curve network, we get

$$\begin{aligned}\frac{dE_2}{du} = \quad & \sum_k (\sum_{c\in C_k^e} - \sum_{c\in C_k^s})(V_c' \cdot c'' - V_c \cdot c''') \\ & + \sum_{c\in\mathscr{C}} \int_{a_c}^{b_c} V_c \cdot c'''' \, dt. \end{aligned} \tag{7}$$

Here the vector fields V_c, c'' and so on are evaluated at the parameter value corresponding to the knot, i.e., at a_c or b_c depending on whether the curve is outgoing or incoming. V_c is the same for all curves meeting in a knot, but V_c' may be different for each. An argument completely analogous to the case of E_1 shows (i) of Th. 2, and we know that the expression involving integrals in Equ. (7) is zero for any E_2-stationary network.

It is well known that for any tangent vector field $V(t)$ along a curve, $V'^\perp$ depends on V and on c' in a bilinear way (see do Carmo [9], §6.2). Especially, if $V = 0$, $V'^\perp = 0$ and V' is tangent to the surface. Also, $c''^\perp$ depends only on c'. So if for a smooth curve in Φ the vector c'' jumps, this jump expresses itself in $c''^\top$ only, while $c''^\perp$ is always continuous. We now construct a variation with $V_c = 0$ in the knots. An incoming/outgoing pair of curves at a knot shares the same c' there. The vectors V_c' can assume any value tangent to Φ, independently of each other in the case of a curve without partner. A curve in C_k^* shares the same V_c' with its partner. The knot k contributes to Equ. (7) the expression

$$\sum_{c\in C_k^{e*}} V_c' \cdot c''^\top - \sum_{c\in C_k^{s*}} V_c' \cdot c''^\top - \sum_{c\in C_k^*} V_c' \cdot \Delta c''^\top. \tag{8}$$

This has to vanish for all variations, which leads to part (ii) of Th. 2: $c''^\top = 0$ for curves without partner, and $\Delta c''^\top = 0$ for curves in C_k^*. As was mentioned above, this actually means that $\Delta c'' = 0$. In case a knot is not fixed, there are variations where $V(k)$ is nonzero and equal for all curves meeting there. Note that

$$V_c' \cdot c'' = V_c' \cdot c''^\top + V_c' \cdot c''^\perp.$$

Further, differentiating the identity $V_c \cdot c''^\perp = 0$ yields

$$V_c \cdot c''^{\perp\prime} + V_c' \cdot c''^\perp = 0,$$

and thus

$$V_c' \cdot c'' = V_c' \cdot c''^\top - V_c \cdot c''^{\perp\prime}.$$

Using these equalities together with $\Delta c''^\perp = 0$ for $c \in C_k^*$ we express the contribution of a knot k to Equ. (7) as

$$V(k) \cdot ((\sum_{c\in C_k^{e*}} - \sum_{c\in C_k^{s*}})(c''^{\perp\prime} + c''') - \sum_{c\in C_k^*} \Delta c''').$$

This leads to (iii) of Th. 2: For all free knots, $T_{2,k}^\top = 0$. □

The next theorem gives a geometric characterization of E_τ-stationary curve networks in a surface.

Theorem 3 *For the tension energy $E_\tau = E_2 + \tau E_1$, E_τ-stationary curve networks are characterized by:*

(i) A linear combination of the fourth and the second derivative vector with the coefficients 1 and $-\tau$ vanishes,

$$(c'''' - \tau c'')^\top = 0$$

for all curves;

(ii) same as (ii) in Theorem 2;

(iii) at all free knots we have

$$(T_{2,k} - \tau T_{1,k})^\top = 0$$

where $T_{1,k}$ and $T_{2,k}$ are defined by Equ. (4) and Equ. (6) respectively.

Proof If E_τ is as in Th. 3, then $\frac{dE_\tau}{du}$ is a linear combination of Equ. (7) plus τ times Equ. (5). Consequently, a geometric characterization of E_τ-stationary networks is the one given by Theorem 3. □

2.5 Unconstrained Networks and Generalizations

Networks not constrained to a surface may be thought to be constrained to $\mathbb{R}^n$. Then for all vectors V, $V = V^\top$ and $V^\perp = 0$. Thus Theorem 2 turns into the following characterization of E_2-stationary networks:

(i) $c'''' = 0$, i.e., all curves of the network are cubic polynomials.

(ii) all structure lines are C^2 cubic splines with natural end conditions ($c'' = 0$ at the ends and $\Delta c'' = 0$ in the knots)

(iii) For all free knots k,

$$T_{2,k} := (\sum_{c\in C_k^{e*}} - \sum_{c\in C_k^{s*}})c''' - \sum_{c\in C_k^{*}} \Delta c''' = 0.$$

These conditions (and likewise those of Th. 1 and Th. 3) are linear. The usually unique E-minimal network is found as solution of a system of linear equations. Further we would like to mention that it is straightforward to derive similar properties for curve networks which minimize other quadratic functionals from spline theory, e.g., the smoothing spline energy or energies with higher derivatives.

As is done for *smoothing splines*, we penalize the distance of movable knots p_k from fixed target points q_k and consider the energy E_σ of a curve network $\mathscr{C}$,

$$E_\sigma(\mathscr{C}) = E_2(\mathscr{C}) + \sum\nolimits_{k \text{ free}} \mu_k \|p_k - q_k\|^2 \quad (\mu_k \geq 0 \text{ fixed}).$$

The choice of the smoothing parameters μ_k for the purpose of data approximation is not a simple problem [41]. Methods analogous to those used in the proofs of Theorem 1–3 show the following: *Theorem2 characterizes E_σ-stationary networks, except that in (iii) the vector $T_{2,k}$ is replaced by*

$$T_{2,k} + \sum_{k \text{ free}} \mu_k(p_k - q_k).$$

3 Fair Polygon Networks

In this section we describe *polygon networks* which are discrete analogues of curve networks. We first introduce a consistent notation and resolve some technicalities that later on allow us to describe theoretical properties of fair polygon networks and compute them using known optimization algorithms.

Theorem 2 says that structure lines have C^2 smoothness when passing through a knot, which means that demanding a small energy yields smoothness. This is the main reason why the optimization framework described below works and produces 'fair' curve networks.

3.1 Notation and Technicalities

We use the word *polygon* as a synonym for a 'sequence of vertices': We look at polygons as discrete curves. In this paragraph we only consider a single polygon $p = (p_1, \dots, p_K)$ with vertices $p_i \in \mathbb{R}^d$. The length (number of vertices) of a polygon with vertices $p = (p_1, \dots, p_K)$ is K. A polygon is either thought to be *closed* with $p_{K+1} = p_1$, $p_{K+2} = p_2, \dots$ or *open*. The *difference polygon* Δp consists of the vectors

$$\Delta p_k := p_{k+1} - p_k.$$

- If p is *open*, the length of the difference polygon $\Delta p = (\Delta p_1, \dots, \Delta p_{K-1})$ is $K-1$.
- If p is *closed*, indices are taken modulo K and the length of the difference polygon $\Delta p = (\Delta p_1, \dots, \Delta p_K)$ is K.

By iteration, we construct higher difference polygons $\Delta^2 p$, $\Delta^3 p$ that consist of vectors

$$\Delta^2 p_k := \Delta(\Delta p_k) = p_{k+2} - 2p_{k+1} + p_k,$$
$$\Delta^3 p_k := \Delta(\Delta^2 p_k) = p_{k+3} - 3p_{k+2} + 3p_{k+1} - p_k,$$

and so on. In the remainder of the section we want to describe the energy of a polygon without having to distinguish between the two cases of 'open' and 'closed' polygons. Thus we define a modified difference operator $\widehat{\Delta}$ as follows.

- For *closed* polygons p, we define the second difference polygon $\widehat{\Delta}^2 p := (\widehat{\Delta}^2 p_1, \dots, \widehat{\Delta}^2 p_K)$ and the forth difference polygon $\widehat{\Delta}^4 p := (\widehat{\Delta}^4 p_1, \dots, \widehat{\Delta}^4 p_K)$ that contain the vectors

  $$\widehat{\Delta}^2 p_k := \Delta^2 p_{k-1}, \quad \widehat{\Delta}^4 p_k := \widehat{\Delta}^2(\widehat{\Delta}^2 p_k)$$

  where the indices are to be taken modulo K.
- If p is *open*, we let $\widetilde{p} = (0, p_1, \dots, p_K, 0)$, and define

  $$\widehat{\Delta}^2 p := \Delta(\widetilde{\Delta p}), \quad \widehat{\Delta}^4 p := \Delta^2(\widetilde{\Delta^2 p}).$$

Then for both closed and open polygons, neither $\widehat{\Delta}^2$ nor $\widehat{\Delta}^4$ changes polygon lengths. Away from the polygon's boundary, we have

$$\begin{aligned}\widehat{\Delta}^2 p_k &= \qquad\quad p_{k+1} - 2p_k + p_{k-1} && (1<k<K),\\ \widehat{\Delta}^4 p_k &= p_{k+2} - 4p_{k+1} + 6p_k - 4p_{k-1} + p_{k-2} && (2<k<K-1).\end{aligned}$$

3.2 Energy of Polygons

Energies $E(p)$ of a polygon p with K vertices in $\mathbb{R}^d$ are real-valued functions defined in $\mathbb{R}^{Kd}$ that have gradients $\nabla E(p)$.

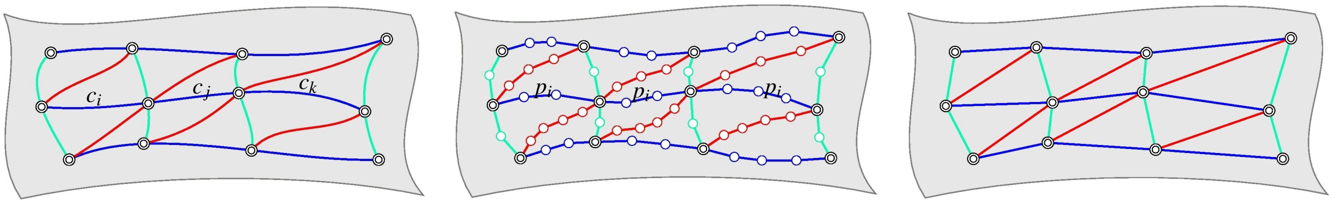


Fig. 5 (Left) Curve network, (middle) polygon network, (right) triangle mesh. Knots are marked with the symbol ⊚. While for curve networks we combine some curves c_i, c_j, c_k to form structure lines (c_i, c_j, c_k), for polygon networks all polygons p_i are already structure lines.

We define the discrete counterparts of the energies E_1, E_2, and E_τ by

$$E_1(p) := \sum_{k=1}^{K} \|\Delta p_k\|^2, \quad E_2(p) := \sum_{k=1}^{K} \|\Delta^2 p_k\|^2, \tag{9}$$

$$E_\tau(p) := \sum_{k=1}^{K} (\|\Delta^2 p_k\|^2 + \tau \|\Delta p_k\|^2). \tag{10}$$

The gradients ∇E_1, ∇E_2, and ∇E_τ of these energy functions are given by

$$\nabla E_1(p) = -2\widehat{\Delta}^2 p, \quad \nabla E_2(p) = 2\widehat{\Delta}^4 p \tag{11}$$

$$\nabla E_\tau(p) = 2(\widehat{\Delta}^4 p - \tau \widehat{\Delta}^2 p). \tag{12}$$

The expressions for the gradients hold in both the open and the closed cases. The modifications the second and fourth order central differences must undergo at the boundaries of open polygons are neatly hidden in the operators $\widehat{\Delta}^2$ and $\widehat{\Delta}^4$.

An energy gradient $\nabla E(p)$ of Equations. (11) and (12) may be visualized as a sequence of vectors $(\nabla E(p))_k$ attached to the vertices p_k of the polygon.

3.3 Fair Polygon Networks

A *fair polygon network* is a polygon network minimizing a discrete energy. We consider polygon networks whose vertices are constrained to surfaces, and also networks whose vertices avoid obstacles. If the fair polygon network is such that all vertices are knots, then we call it a *fair mesh*. A *knot* is a vertex that is shared by more than one polygon. The *connectivity* of a polygon network is given by the polygons and the knots. Note that in contrast to the curve network case we do not consider incoming and outcoming curves at a knot, and each polygon is already a structure line. This is illustrated in Fig. 5: In the curve network case, curve segments c_i, c_j, c_k may form a structure line (c_i, c_j, c_k), but in the polygon network counterpart we only have one polygon p_i for which four of its vertices are knots. The total energy of a polygon network $\mathscr{P}$ is defined as the sum of energies of all polygons of the network,

$$E(\mathscr{P}) := \sum_{p \in \mathscr{P}} E(p),$$

where $E(p)$ is one of the discrete energies E_1, E_2, or E_τ of a polygon that we have defined in Equ. (9) or Equ. (10).

Since the vertices of a polygon network are not all distinct, we start with a collection $V = (v_1, \ldots, v_N) \in \mathbb{R}^{Nd}$ of vertices. Then we consider polygons $p = (p_1, \ldots, p_K)$ whose vertices are taken from V:

$$p_1 = v_{i_1}, \ldots, p_K = v_{i_K}.$$

Using this setup, the energy of a single polygon p is also a function of the entire vertex collection $v_1, \ldots, v_N$ and we write $E_p(V)$. The gradient of the function $E_p(V)$ is a sequence of vectors attached to the vertices collected in V. If a vertex is not contained in p, then $\nabla E_p(V)$ will assign the zero vector to this vertex. Thus, the energy gradient $\nabla E_p(V)$ is interpreted as a sequence of N vectors.

For the total energy of a fair polygon network, and the gradient of the energy, seen as a function of the entire vertex set $V = (v_1, \ldots, v_N)$, we use the notation

$$E(V) = \sum_{p \in \mathscr{P}} E_p(V), \quad \nabla E(V) = \sum_{p \in \mathscr{P}} \nabla E_p(V).$$

3.4 Fair Polygon Networks in Surfaces

Suppose that the vertices v_i of a polygon network $\mathscr{P}$ are constrained to a surface Φ with implicit equation $F(x) = 0$. Then an E-stationary polygon network is found by letting the gradient of the Lagrange function

$$E(V) + \sum_{i=1}^{N} \lambda_i F(v_i)$$

equal zero under the constraints

$$F(v_i) = 0, \quad i = 1, \ldots, N.$$

When using the tension energy $E_\tau = E_2 + \tau E_1$, this condition reads

$$2\sum\nolimits_p (\widehat{\Delta}^4 p - \tau \widehat{\Delta}^2 p) = -\lambda_i \nabla F(v_i), \tag{13}$$

for $i = 1, \ldots, N$. The meaning of the sum is that each polygon which contains the vertex v_i contributes to it. We see that for each vertex v_i the sum on the left hand side of Equ. (13) must be orthogonal to Φ if the discrete network is to be E-minimizing.

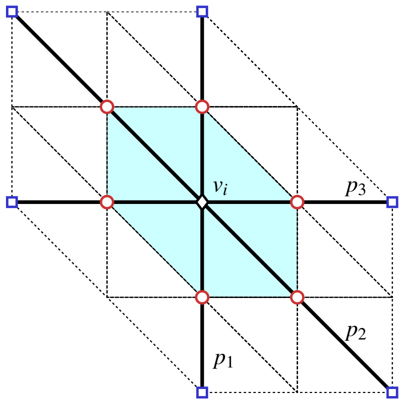


Fig. 6 Illustration of the polygons and points involved in the computation of the first and second umbrella vector. Three polygons p_1, p_2, p_3 pass through the vertex v_i. The points of the sets V_i' and V_i'' are marked by circles ○ and squares □, respectively. The triangle mesh defined by the fair polygon network is marked with dotted lines. The triangles of the first umbrella are shown shaded.

3.5 Fair Meshes

We now move to a special case and consider a *mesh* with vertices $V = (v_1, \ldots, v_N)$ taken from a surface Φ. The vertices can be chosen such that the mesh is decomposed into polygons made from edges of the mesh (see Fig. 6 for the decomposition of a small patch and Fig. 7 (b) for a whole surface). In our applications, the decomposition will be part of the design (Fig. 1). Thus a fair mesh is a special fair polygon network in a surface.

If a vertex v_i takes part in exactly L polygons passing through, it has $2L$ first order neighbours collected in the set V_i' and $2L$ second order neighbours, two to each polygon, collected in V_i'', see Fig. 6. Kobbelt introduced the so called *umbrella vector* for a vertex of a mesh, see e.g. [17]. Following this idea we define *first and second umbrella vectors* $u_{i,1}$, $u_{i,2}$ by

$$u_{i,1} := v_i - \frac{1}{2L} \sum_{v \in V_i'} v, \quad u_{i,2} := v_i - \frac{1}{2L} \sum_{v \in V_i''} v.$$

Note that our first umbrella vector is the same as the one defined by Kobbelt. However, $u_{i,2}$ is different in that we use only those points of the 2-ring neighborhood that appear in the polygons passing through the vertex v_i. For a fair mesh it is easy to show that Equ. (13) reduces to the condition that

$$u_{i,2} - (4 + \tau) u_{i,1}$$

is orthogonal to Φ, whereas an E_1-stationary network has the first umbrella vectors orthogonal to Φ (the latter condition is used in mesh fairing algorithms based on Laplacian smoothing, see e.g. [40]).

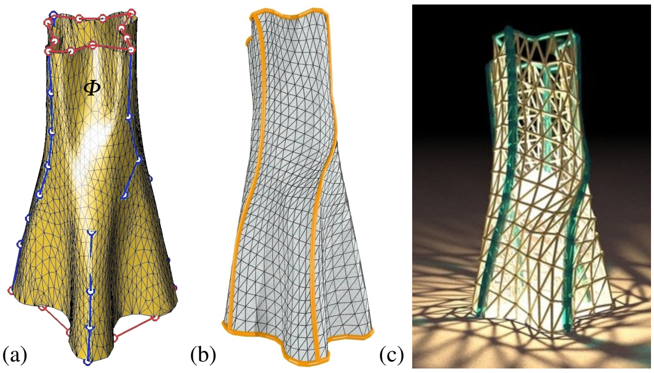


Fig. 7 (a) User input on original mesh Φ are four polygons in vertical direction and two polygons in horizontal direction. (b) Fair mesh interpolating the input polygons. (c) Design based on coarser fair mesh interpolating the given polygons.

4 Results and Discussion

4.1 Implementation Issues

The energy gradient of a single polygon is computable with Equ. (11) and the total energy gradient is found by summing up the contributions of the single polygons. Thus setting up a gradient descent algorithm or conjugate gradient method for energy minimization is straightforward (in the case of networks constrained to surfaces, "gradient" means "projected gradient"; in the case of obstacles gradients have to be modified so as not to move points into the obstacle). In order to treat the various numerical representations of surfaces (parametric/implicit/triangle mesh/point cloud) and obstacles, it is sufficient that a projection-type mapping onto the surface or onto the obstacle's boundary is available, and that tangent spaces can be computed.

If the network connectivity is refinable, an energy-minimizing coarse network can be used to initialize computation of a finer one, so the multigrid idea applies. By locality of the gradient, perturbations propagate through the network at a speed of two vertices per iteration. Total computation times depend on the nature of surfaces and obstacles used — we experienced about 1/100 sec per point of the discrete network on a 2GHz PC.

4.2 Examples

Figure 7 illustrates how a user specifies coarse polygons on a surface Φ which are interpolation conditions for the fair mesh (i.e., all knots are fixed). Note that in this example Φ is itself a triangle mesh. The vertices of the fair mesh lie on the surface Φ, but there is no correspondence of the vertices of Φ to the vertices of the fair mesh. With the interpolation constraints as input our algorithm computes a fair mesh on Φ where the edges of the triangles nicely follow the directions given by the user.

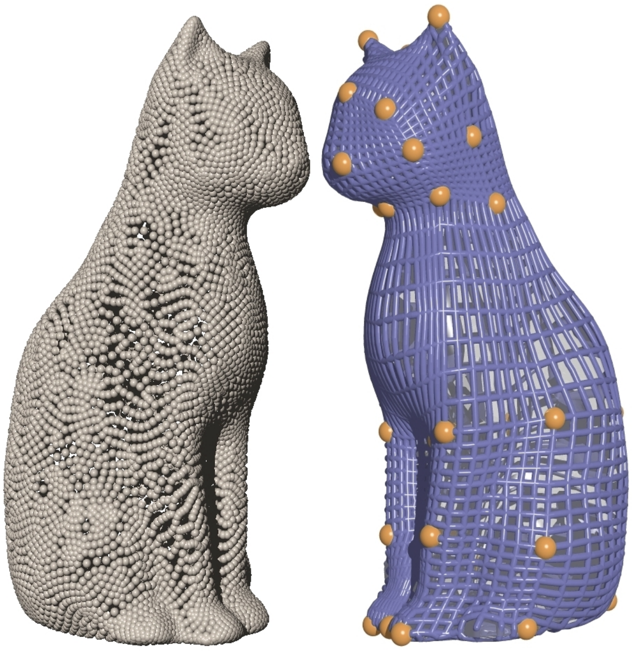


Fig. 8 Minimal energy network of cube topology wrapped around a point cloud (fixed knots yellow, original cube corners have valence 3).

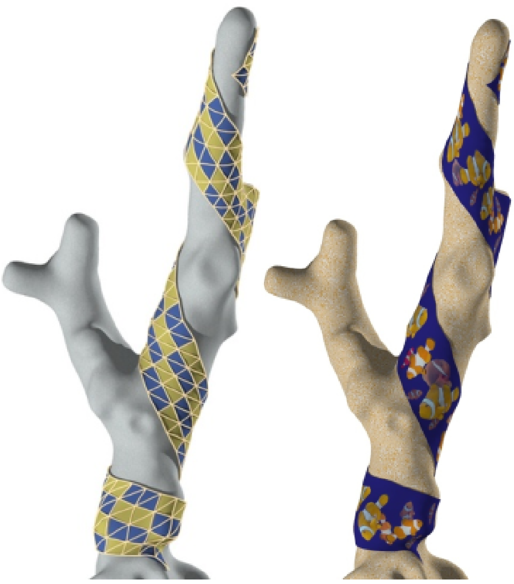


Fig. 9 Fair polygon network used for texture mapping. The mean size of features of the coral branch implies that hat the network is just dense enough to avoid being stuck in an insignificant local minimum. Experimental evidence supports this. The colored band itself lies on the given surface.

Aesthetic remeshing. Alliez et al. give a state-of-the-art report on remeshing [2], a topic that has attracted a lot of attention in recent years. Optimization criteria mainly have been taken from needs in simulation or high accuracy approximation [1,7,38]. Fair webs add another perspective and thus contribute to aesthetic design [36]: They are well suited to compute visually pleasing meshes in the sense that they are

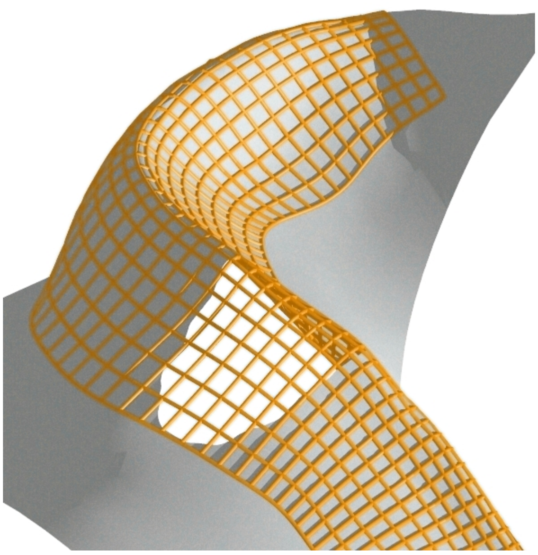


Fig. 10 Model restoration: A fair curve network is used to fill the hole present in the model. Note that the curve network beautifully fills the hole in a meaningful way.

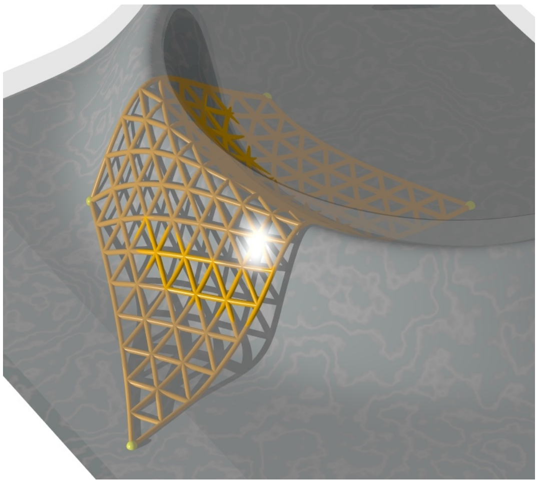


Fig. 11 Surface approximation with prescribed error bounds. This is achieved by confining the network to the space between two surfaces which are close together.

formed by sequences of fair discretized curves. By fixing certain knots or even some curves, the designer has an influence on the visual appearance. Special fair meshes arise in various problems of computational differential geometry [32], but there also the surface is subject to optimization. This is not the case in our approach, where the mesh vertices stay on a given surface or, if desired, just close to it. Figs. 1, 7 show a remeshing example.

Fair parameterization. Fair webs may also be applied to surface parameterization. Most known parameterization algorithms aim at isometry, and since this is in general not achievable, at angle preservation, area preservation or reasonable trade-offs between those [12,13]. We would like to

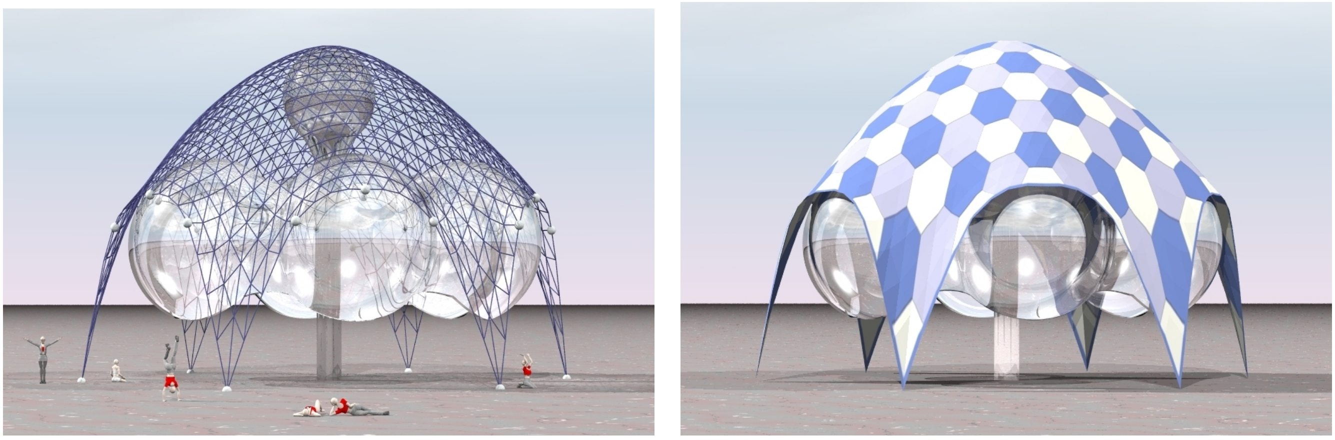


Fig. 12 Architectural design with a fair polygon network that avoids the given obstacles. The small balls in the left hand figure represent interpolation conditions.

point to [30], where isosurfaces are parameterized via special curve networks. Parameterization via fair webs maps chosen line families of the parameter domain to fair curves on a surface. For applications such as mapping regular texture onto a surface, this fairness criterion plays an important role. For instance, it is easy to wind a visually pleasing textured band around an object (Fig. 9) or to compute a fair parameterization of a surface over a simpler object of arbitrary topology, e.g., a cube (Fig. 8) or a polycube [39]. Fig. 7 may be seen as reparametrization over a domain of cylinder topology.

Surface restoration and editing. Surface restoration in 3D models has a number of applications such as completion of an incomplete scan, 3D model beautification, or surface blending. Several variational formulations for the solution of this problem have been proposed, see [6,35,43]. Fair webs are well suited for this purpose. We do not just incorporate continuity conditions at the boundary of the hole, but take a larger web, which partially lies in the original surface. There, it picks up the surface behavior around the hole and therefore fills the hole in a natural manner (Fig. 10). Control handles for *editing* fair webs (i.e., surfaces) are the placement of fixed knots, the temporary introduction of new fixed knots, and the location of obstacles (Fig. 12).

Fair surface design in the presence of obstacles. Although limited design space is a practical issue, only recently there has been some research on the design of fair curves in the presence of obstacles [14,27]. Fair webs provide a simple approach to the design of fair surfaces which avoid given obstacles (Fig. 12).

Surface approximation with guaranteed error bounds. By deleting an admissible strip around a surface Φ from $\mathbb{R}^d$ and considering the rest as an obstacle, we can approximate Φ within guaranteed error bounds (see Fig. 11). We have successfully employed *fair polyline networks* for smoothing digital terrain elevation data with guaranteed error bounds [15].

4.3 Conclusion and Future Research

In this paper we contribute results on energy-minimizing curve and polygon networks which are either constrained to lie in a given surface or are to avoid obstacles. We illustrate our theoretical results by means of several examples that indicate the possible applications of fair webs and fair polygon networks. In future research we plan to investigate the use of constraints putting a penalty on heavy distortions such as accumulations of network curves (see Fig. 8), or pushing away of network curves from feature regions.

Acknowledgements The work of M. Hofer has been supported by the Austrian Science Fund (FWF) under grant P16002.

References

1. Alliez, P., Cohen-Steiner, D., Devillers, O., Levy, B., Desbrun, M.: Anisotropic polygonal remeshing. ACM Transactions on Graphics **22**(3), 485–493 (2003). (Proc. SIGGRAPH 2003)
2. Alliez, P., Ucelli, G., Gotsman, C., Attene, M.: Recent advances in remeshing of surfaces. State-of-the-art report, AIM@SHAPE EU network (2005). URL *http://www-sop.inria.fr/geometrica/team/Pierre.Alliez/*
3. Biermann, H., Martin, I.M., Zorin, D., Bernardini, F.: Sharp features on multiresolution subdivision surfaces. Graph. Models **64**(2), 61–77 (2002)
4. Blake, A., Isard, M.: Active Contours. Springer (1998)
5. Bohl, H.: Kurven minimaler Energie auf getrimmten Flächen. Ph.D. thesis, Stuttgart Univ. (1999)
6. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. ACM Transactions on Graphics **23**(3), 630–634 (2004). (Proc. SIGGRAPH 2004)
7. Botsch, M., Kobbelt, L.: A remeshing approach to multiresolution modeling. In: Proc. Symposium on Geometry Processing, pp. 189–196. Eurographics (2004)

8. Brunnett, G., Hagen, H., Santarelli, P.: Variational design of curves and surfaces. Surveys Math. Ind. **3**, 1–27 (1993)
9. do Carmo, M.: Riemannian Geometry. Birkhäuser (1992)
10. do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)
11. Cheng, B.T., Burchard, P., Merriman, B., Osher, S.: Motion of curves constrained on surfaces using a level set approach. J. Comp. Phys. **175**, 604–644 (2002)
12. Clarenz, U., Litke, N., Rumpf, M.: Axioms and variational problems in surface parameterization. Computer-Aided Geometric Design **21**, 727–749 (2004)
13. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: N.A. Dodgson, M.S. Floater, M.A. Sabin (eds.) Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer (2005)
14. Hofer, M., Pottmann, H.: Energy-minimizing splines in manifolds. ACM Transactions on Graphics **23**(3), 284–293 (2004). (Proc. SIGGRAPH 2004)
15. Hofer, M., Sapiro, G., Wallner, J.: Fair polyline networks for constrained smoothing of digital terrain elevation data. IEEE Trans. Geosc. Remote Sensing **44**(10/2), 2983–2990 (2006)
16. Kimmel, R.: Numerical Geometry of Images. Springer (2003)
17. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: Proc. SIGGRAPH '98, pp. 105–114. ACM Press (1998)
18. Kobbelt, L., Schröder, P.: A multiresolution framework for variational subdivision. ACM Trans. Graphics **17**(4), 209–237 (1998)
19. Kolb, A.: Optimierungsansätze bei der Interpolation verteilter Daten. Ph.D. thesis, Univ. Erlangen (1995)
20. Kolb, A., Seidel, H.P.: Interpolating scattered data with C^2 surfaces. Computer-Aided Design **27**, 277–282 (1995)
21. Lee, S.Y., Chwa, K.Y., Shin, S.Y., Wolberg, G.: Image metamorphosis using snakes and free-form deformations. In: ACM SIGGRAPH 95, pp. 439–448. ACM (1995)
22. Levin, A.: Interpolating nets of curves by smooth subdivision surfaces. In: Proc. SIGGRAPH '99, pp. 57–64. ACM Press (1999)
23. Li, W.C., Levy, B., Paul, J.C.: Mesh editing with an embedded network of curves. In: Proc. International Conf. Shape Modeling (2005)
24. Liu, Y., Lin, W.C., Hays, J.: Near-regular texture analysis and manipulation. ACM Transactions on Graphics **23**(3), 368–376 (2004)
25. Memoli, F., Sapiro, G.: Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. J. Comput. Phys. **173**, 730–764 (2001)
26. Moreton, H., Sequin, C.: Functional optimization for fair surface design. In: ACM SIGGRAPH 92, pp. 167–176. ACM (1992)
27. Myles, A., Peters, J.: Threading splines through 3D channels. Computer-Aided Design **37**, 139–148 (2005)
28. Nielson, G.: A method for interpolating scattered data based upon a minimum norm network. Math. Comp. **40**, 253–271 (1983)
29. Nielson, G., Franke, R.: A method for construction of surfaces under tension. Rocky Mountain J. Math. **14**, 203–221 (1984)
30. Nielson, G., Lee, K., Zhang, L.: Lifting curve parameterization methods to isosurfaces. Computer-Aided Geometric Design **21**, 751–766 (2004)
31. Noakes, L.: Null cubics and Lie quadratics. J. Math. Phys. **44**, 1436–1448 (2003)
32. Polthier, K.: Polyhedral surfaces of constant mean curvature. Habilitationsschrift TU Berlin (2002)
33. Pottmann, H., Hofer, M.: A variational approach to spline curves on surfaces. Comput. Aided Geom. Design **22**(7), 693–709 (2005)
34. Schaefer, S., Warren, J., Zorin, D.: Lofting curve networks using subdivision surfaces. In: Proc. Symposium on Geometry Processing, pp. 103–114. Eurographics (2004)
35. Schneider, R., Kobbelt, L.: Geometric fairing of irregular meshes for free-form surface design. Computer-Aided Geometric Design **18**, 359–379 (2001)
36. Sequin, C.: CAD tools for aesthetic engineering. CAD & Appl. **1**, 301–309 (2004)
37. Singh, K., Fiume, E.: Wires: A geometric deformation technique. In: ACM SIGGRAPH 98, pp. 405–414. ACM (1998)
38. Surazhsky, V., Alliez, P., Gotsman, C.: Isotropic remeshing of surfaces: a local parameterization approach. In: Proceedings 12th Int. Meshing Roundtable (2003)
39. Tarini, M., Hormann, K., Cignoni, P., Montani, C.: Polycube-maps. ACM Transactions on Graphics **23**(3), 853–860 (2004). (Proc. SIGGRAPH 2004)
40. Taubin, G.: A signal processing approach to fair surface design. In: Proc. SIGGRAPH '95, pp. 351–358. ACM Press (1995)
41. Wahba, G.: Spline Models for Observational Data. SIAM (1990)
42. Welch, W., Witkin, A.: Variational surface modeling. In: Proc. SIGGRAPH '92, pp. 157–166. ACM Press (1992)
43. Xu, G.: Discrete Laplace-Beltrami operators and their convergence. Computer-Aided Geometric Design **21**, 767–784 (2004)

Johannes Wallner is a member of the Institute of Discrete Mathematics and Geometry at Vienna University of Technology. He received his Ph.D. from TU Wien in 1997 and researches in geometry. He coauthored the monograph *Computational Line Geometry* published by Springer in 2001. A recent focus of his work is smoothness analysis of nonlinear subdivision schemes, tolerance analysis, and geometry processing. See also the web pages at *http://www.geometrie.tuwien.ac.at/wallner*.

Helmut Pottmann is professor of Geometry at Vienna University of Technology. His research interests include classical geometry and its applications, especially to Computer Aided Geometric Design, Computer Vision, industrial geometry, kinematics, and the relations between geometry, numerical analysis and approximation theory. At list of recent publications can be found *http://www.geometrie.tuwien.ac.at/pottmann*.

Michael Hofer received his Ph.D. in 2005 from Vienna University of Technology and then worked as a Postdoc at the University of Minnesota. Currently he is a Postdoc at TU Wien. His research focuses on the solution of geometric problems in Geometric Modeling and Processing, Computer Aided Geometric Design, Computer Graphics and Computer Vision. For more information and a list of publications see the webpage *http://www.geometrie.tuwien.ac.at/hofer*.