# Geometric Modeling with Conical Meshes and Developable Surfaces

Yang Liu [*] Univ. of Hong Kong
Helmut Pottmann [†] TU Wien
Johannes Wallner [‡] TU Wien
Yong-Liang Yang [§] Tsinghua Univ., Beijing
Wenping Wang [¶] Univ. of Hong Kong

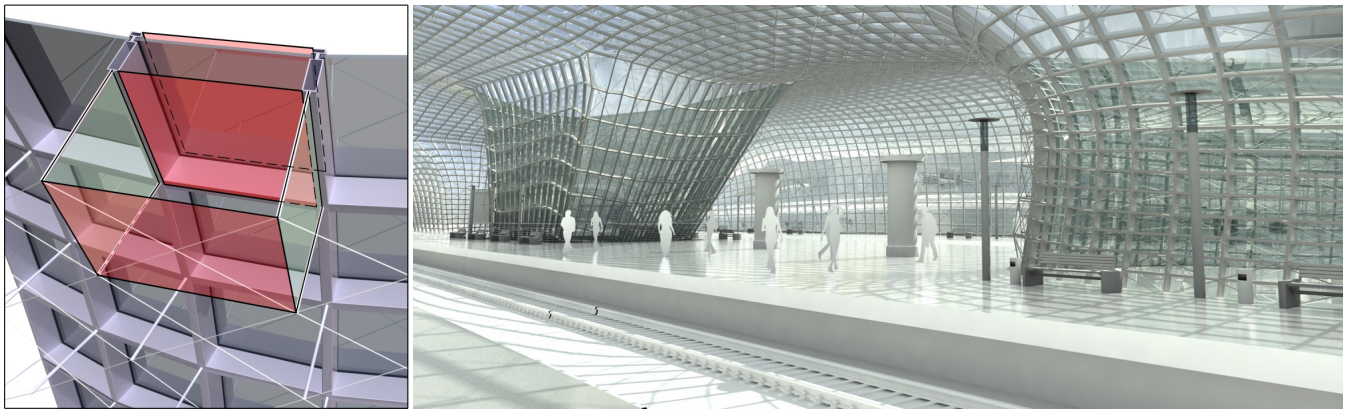

Figure 1: Conical meshes are planar quad meshes which discretize principal curvature lines, possess offset meshes at a constant distance as well as planar connecting elements supporting the offset meshes (left). Therefore they are especially suited for architectural design with glass structures (right). This student project of a railway station by B. Schneider was generated by a subdivision-type process (see also Fig. 14).

## Abstract

In architectural freeform design, the relation between shape and fabrication poses new challenges and requires more sophistication from the underlying geometry. The new concept of conical meshes satisfies central requirements for this application: They are quadrilateral meshes with planar faces, and therefore particularly suitable for the design of freeform glass structures. Moreover, they possess a natural offsetting operation and provide a support structure orthogonal to the mesh. Being a discrete analogue of the network of principal curvature lines, they represent fundamental shape characteristics. We show how to optimize a quad mesh such that its faces become planar, or the mesh becomes even conical. Combining this perturbation with subdivision yields a powerful new modeling tool for all types of quad meshes with planar faces, making subdivision attractive for architecture design and providing an elegant way of modeling developable surfaces.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

**Keywords:** discrete differential geometry, nonlinear subdivision, quad mesh, principal mesh, offset mesh, developable surface, developable subdivision surface, surfaces in architecture.

[*]e-mail:yliu@cs.hku.hk
[†]e-mail:pottmann@geometrie.tuwien.ac.at
[‡]email:wallner@geometrie.tuwien.ac.at
[§]email:yangyl@cg.cs.tsinghua.edu.cn
[¶]e-mail:wenping@cs.hku.hk

## 1 Introduction

The original motivation for this research comes from architecture, where freeform shapes are becoming increasingly popular, but the actual construction poses new demands on the underlying geometry. Gehry Partners and Schlaich Bergermann and Partners [Glymph et al. 2002] argue why freeform glass structures with planar quadrilateral facets are preferable over structures built from triangular facets or non-planar quads. The authors also show a few simple ways to construct quad meshes with planar faces. However, despite the huge amount of work on mesh processing and the interest in discrete differential geometry [Desbrun et al. 2005], we are not aware of a thorough investigation of this topic from the perspective of geometry processing.

The study of quad meshes with planar faces – called *PQ meshes* henceforth – will lead us to interesting geometric results, in particular to *conical meshes*, a discrete counterpart of principal curvature lines which have not been considered before. Algorithms which perturb a quad mesh into a PQ mesh can nicely be combined with subdivision. This makes subdivision a promising tool for architectural design and also provides a new and elegant approach to modeling and approximation with developable surfaces.

### 1.1 Previous work

**Discrete differential geometry.** PQ meshes have first been systematically addressed by R. Sauer, as summarized in his monograph [1970] on *difference geometry*, one of the precursors of discrete differential geometry [Bobenko and Suris 2005; Desbrun et al. 2005; Polthier 2002; Hildebrandt et al. 2005]. It has been observed that PQ meshes are a discrete counterpart of conjugate curve networks on surfaces. They appear in the mathematics literature under the name of *quadrilateral* meshes, which actually means quad meshes with the additional property that all quads are planar. The interesting case of *circular meshes* where all quads possess a circumcircle has been introduced in [Martin et al. 1986]. Like the conical meshes which are a focus of the present paper, circular meshes

are discrete analogues of the network of principal curvature lines. Pointers to the literature on PQ meshes and circular meshes, especially to higher-dimensional generalizations, are given in [Bobenko and Suris 2005] and [Bobenko et al. 2006]. Convergence of circular meshes towards the network of principal curvature lines is the topic of [Bobenko et al. 2003]. Here we discuss topics in planar quad meshes which have not been addressed previously, like conical meshes and offset meshes, as well as developable surfaces obtained by refinement and optimization.

**Quad meshes.** The computation of quad-dominant meshes from smoothed principal curvature lines has been presented in [Alliez et al. 2003]. Although the faces of these meshes are not exactly planar, one should expect that they are at least approximately planar. Thus such meshes can serve as an input to algorithms presented below, which compute numerically precise PQ meshes and conical meshes by optimization. Variational shape approximation according to [Cohen-Steiner et al. ] aims at the optimal placement of a given number of planar faces, which, in general, are not quadrilaterals. Other recent contributions to quadrilateral remeshing (see e.g. [Dong et al. 2005; Marinov and Kobbelt 2004; Ray et al. 2005]) do not try to achieve planarity of quads.

**Developable surfaces.** An arrangement of $n$ planar quads in a single row (see Fig. 2) is a discrete representation of a *developable surface*. In this way the study of PQ meshes is related to the computational geometry of developable surfaces. Recall a few facts from differential geometry [do Carmo 1976; Pottmann and Wallner 2001]: A developable surface $\Gamma$ is the envelope of a one-parameter family of planes. Each of these planes touches the surface along a straight line, a so-called ruling. There are three main types: Either rulings are parallel ($\Gamma$ is a cylinder surface), or they pass through a fixed point $\mathbf{s}$ ($\Gamma$ is a cone with vertex $\mathbf{s}$), or they are tangents of a space curve $r$ ($\Gamma$ is a tangent surface and $r$ is its singular curve). Because developable surfaces can be mapped into the plane without distortion, they possess a variety of applications, for example, in sheet-metal and plate-metal based industries and architecture. Modeling with developable surfaces is a nontrivial task, which is only weakly included in current 3D modelers. Several ways of geometric design with developables have been proposed. One can use B-spline ruled surfaces and express developability via nonlinear constraints [Aumann 2004; Chu and Sequin 2002]. Via duality such constraints can be avoided, at the cost of a less intuitive plane-based control structure [Pottmann and Wallner 2001]. There are also contributions based on constrained triangle meshes [Frey 2004; Wang and Tang 2004; Mitani and Suzuki 2004]. Singularities in crumpled sheets have also received attention, see e.g. [Cerda et al. 1999; Frey 2004]. Recently there has been interest in developable surfaces for mesh parametrization [Julius et al. 2005] and mesh segmentation [Yamauchi et al. 2005].

**Surfaces in architecture and aesthetic design.** Freeform geometries are becoming increasingly popular in architecture, thus demanding adapted modeling methods which take the actual construction and fabrication into consideration. The Smart Geometry group (*http://www.smartgeometry.com*) promotes research in this direction; a good overview of the state of the art may be found in [Kilian 2006]. Geometric modeling for aesthetic design and 'optimal geometry' are the topics of [Sequin 2004] and [Sullivan 2005].

### 1.2 Contributions and overview

• We introduce *conical meshes* and demonstrate their superiority over other types of meshes for architectural design and other applications where planarity and exact offset property are demanded.

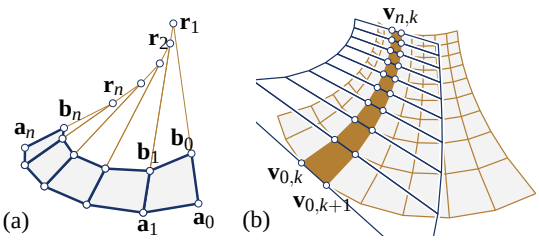


Figure 2: (a) PQ strip as discrete model for a developable surface. (b) Discrete developable tangent to PQ mesh along a row of faces.

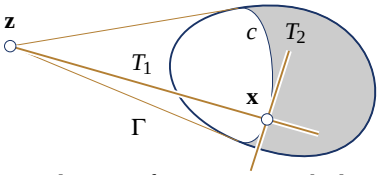


Figure 3: Visualization of conjugacy via shadow contours.

The conical mesh is a new type of *principal meshes* and it possesses the property that offsetting the face planes by a constant distance yields a planar mesh of the same connectivity, which is again a conical mesh. This is a very useful property in layer composition constructions for architecture, where each layer has to be covered by planar panel elements and the geometry of the outermost layer should also be valid for the offsets which represent the layer composition (see Figures 1, 11, 12a, and 15).

• We propose the *PQ perturbation algorithm* for computing a PQ mesh from an input quadrilateral mesh. By combining PQ perturbation with a surface subdivision scheme we obtain a powerful tool for modeling not only conical meshes, but also circular meshes and general PQ meshes. When applied to PQ strips, it leads to an effective and elegant approach to modeling developable surfaces.

In Sec. 2 we elaborate on the relation between PQ meshes to *conjugate curve networks* for understanding the variety of PQ meshes. Sec. 2.2 discusses the *PQ perturbation algorithm*. In Sec. 3 we combine subdivision and PQ perturbation to get a *hierarchical construction of PQ meshes*. In particular, we obtain *developable subdivision surfaces*. *Conical meshes* are introduced in Sec. 4, and their main properties are derived. Sec. 5 discusses how to approximate given data by a conical mesh via optimization of a quad mesh, possibly derived from robustly computed principal curves on an appropriate scale. We discuss our results in Sec. 6 and conclude the paper with some pointers to future research in Sec. 7.

## 2 PQ meshes and PQ perturbation

**Conjugate Curves.** Quad meshes with planar faces may be seen as a discrete version of so-called conjugate curve networks on a surface [Sauer 1970]. First we explain *conjugate surface tangents* at a point $\mathbf{x}$ of a surface $\Phi$ (see Fig. 3): Suppose that the straight line $T_1$ is tangent to the surface at $\mathbf{x}$. Choose a light source $\mathbf{z}$ on $T_1$. Then the line $T_2$ tangent to the shadow contour (contour generator) $c$ at $\mathbf{x}$ is conjugate to $T_1$. $T_1$ is contained in the conical surface $\Gamma$ of surface tangents passing through the light source $\mathbf{z}$. Here we could also use a parallel illumination, with $\mathbf{z}$ at infinity. An alternative definition of conjugate directions in terms of the second fundamental form of a surface is given by [do Carmo 1976, p. 150].

The above is a special case of the following more general property: If $\Gamma$ is the developable surface enveloped by the tangent planes along a curve $c \subset \Phi$, and $T_1$ is a ruling of $\Gamma$ passing through the

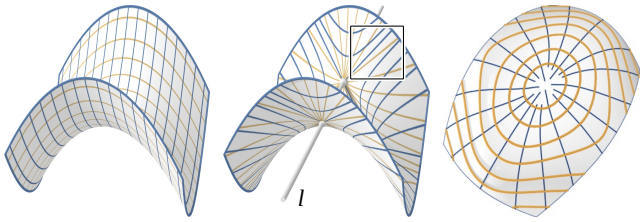


Figure 4: Various conjugate networks and their suitability for meshing purposes. Left: The network of generating curves in a translational surface $\Phi$ is conjugate. Center: For any surface $\Phi$, the intersection curves (yellow) of $\Phi$ with planes through a fixed line $l$ and the contour generators (blue) for viewpoints on $l$ form a conjugate network. Right: Isophotes (yellow) and curves of steepest descent (blue). Such networks may be unsuitable for meshing even for simple surfaces, if its curves do not intersect transversely. This is caused by asymptotic directions (see frame).

point $\mathbf{x} \in c$, then the line $T_2$ tangent to the curve $c$ at the point $\mathbf{x}$ is conjugate to $T_1$. This relation turns out to be symmetric (see e.g. [Pottmann and Wallner 2001]). *Asymptotic* directions are self-conjugate. A *conjugate network of curves* consists of two one-parameter families $A, B$ of curves which cover a given surface $\Phi$ such that for each point $\mathbf{p} \in \Phi$ there is a unique curve of $A$ and a unique curve of $B$ which pass through $\mathbf{x}$, and furthermore, the tangents of these two curves at $\mathbf{x}$ are conjugate. We may prescribe family $A$ and get family $B$ by integration of the vector field of directions conjugate to the tangents of family $A$.

Examples of conjugate networks on surfaces are:
— The network of principal curvature lines is always conjugate.
— In a translational surface of the form $\mathbf{x}(u,v) = \mathbf{p}(u) + \mathbf{q}(v)$, generated by a translatory motion of a profile curve $\mathbf{p}(u)$ along a directrix curve $\mathbf{q}(v)$, or vice versa, the isoparameter lines form a conjugate network (Fig. 4, left).
— The movement of a viewpoint $\mathbf{z}$ along some curve in space produces a family of *contour generators* "$c(\mathbf{z})$" on a given surface $\Phi$, where $\mathbf{z}$ is interpreted as a light source. The curves conjugate to the $c(\mathbf{z})$'s are called *epipolar curves* and are found by integrating the field of light rays tangent to the surface $\Phi$. These curves arise in 3D surface reconstruction from apparent contours in an image sequence [Cipolla and Giblin 2000]. Fig. 4 (center) shows the case where $\mathbf{z}$ moves along a straight line $l$.
— The condition that the surface normals form a constant angle with the $z$-axis defines an *isophotic curve*. These isophotes are conjugate to the system of *curves of steepest descent* with respect to the $z$-axis (see Fig. 4, right, and e.g. [Pottmann and Wallner 2001]).

## 2.1 PQ meshes

Let us start with a *PQ strip*, which means a single row of planar quadrilateral faces. The two rows of vertices are denoted by $\mathbf{a}_0, \ldots, \mathbf{a}_n$ and $\mathbf{b}_0, \ldots, \mathbf{b}_n$ (see Fig. 2). It is obvious and well known that such a mesh is a discrete model of a developable surface (see e.g. [Pottmann and Wallner 2001; Sauer 1970]). This surface is cylindrical, if all lines $\mathbf{a}_i\mathbf{b}_i$ are parallel. If the lines $\mathbf{a}_i\mathbf{b}_i$ pass through a fixed point $\mathbf{s}$, we obtain a model for a conical surface with vertex $\mathbf{s}$. Otherwise the PQ strip is a patch on the tangent surface of a polyline $\mathbf{r}_1, \ldots, \mathbf{r}_n$, as illustrated by Fig. 2: consecutive lines $\mathbf{a}_i\mathbf{b}_i$ and $\mathbf{a}_{i+1}\mathbf{b}_{i+1}$ are co-planar and thus intersect in a point $\mathbf{r}_{i+1}$. It follows that both $\mathbf{a}_i$ and $\mathbf{b}_i$ are contained in the line $\mathbf{r}_i\mathbf{r}_{i+1}$. This property is the direct analogue of the well known fact that, in general, a developable surface is part of the tangent surface of a space curve. The lines $\mathbf{r}_i\mathbf{r}_{i+1}$ serve as the rulings of the discrete tangent surface, which carries the given PQ strip. The planar faces of the strip represent tangent planes of the developable surface.

Now we consider a general PQ mesh with vertices $\mathbf{v}_{i,j}$, $i = 0, \ldots, n$, $j = 0, \ldots, m$. For theoretical investigations we will always assume that interior mesh vertices have valence four; vertices with valence $\neq 4$ are like singularities in a curve network and require special treatment. In practice, meshes will not consist of quads only – $n$-gons with $n \neq 4$ likewise are treated as singularities.

Recall the property mentioned above which characterizes conjugate curve networks: the envelope of tangent planes along a curve of family $A$ is a developable surface, whose rulings are tangent to curves of family $B$. We can easily see that the row and column polylines of a PQ mesh enjoy a discrete version of this property: Each row of faces $\mathbf{v}_{i,j}$ (we let $j = k, k+1$) is a PQ strip, which represents a discrete developable surface tangent to the mesh (Fig. 2). The row of vertices $\mathbf{v}_{0,k}, \ldots, \mathbf{v}_{n,k}$ can be seen as the polyline of tangency between the mesh and this developable surface. The rulings of the developable surface are spanned by the edges $\mathbf{v}_{i,k}, \mathbf{v}_{i,k+1}$ for $i = 1, \ldots, n$. The same lines occur as tangents of the column polylines $\mathbf{v}_{i,0}, \ldots, \mathbf{v}_{i,m}$. It follows that the system of row and column polylines are a *discrete conjugate network* of polylines. Moreover, a discrete developable surface tangent to a PQ mesh along a polyline is given by a row (or a column) of quad faces.

Consequently, *if a subdivision process, which preserves the PQ property, refines a PQ mesh and produces a curve network in the limit, then the limit is a conjugate curve network on a surface.*

## 2.2 PQ perturbation

Given a quad mesh with vertices $\mathbf{v}_{i,j}$, we want to minimally perturb the vertices into new positions such that the resulting mesh is a PQ mesh. One way to solve this problem is by a Sequential Quadratic Programming method (SQP, see e.g. [Madsen et al. 2004]), which minimizes fairness and closeness functionals subject to the planarity condition. Another way is a penalty method which optimizes a linear combination of functionals which express planarity, fairness, and closeness to the original mesh, weighted in a way which ensures numerically exact planarity.

In order to express planarity of a quad face $Q_{ij}$, we consider the four angles $\phi^1_{i,j}, \ldots, \phi^4_{i,j}$ enclosed by the edges of $Q_{ij}$, measured in the interval $[0, \pi]$. It is known that $Q_{ij}$ is planar and convex if and only these angles sum up to $2\pi$. We use the notation

$$c_{pq,i,j} := \phi^1_{i,j} + \ldots + \phi^4_{i,j} - 2\pi = 0. \qquad (1)$$

Below we need sums of the form $\sum_{i,j} \lambda_{pq,i,j} c_{pq,i,j}$, which we write as $\lambda^T_{pq} c_{pq}$, i.e., the inner product of the vectors $\lambda_{pq} = (\lambda_{pq,i,j})$ and $c_{pq} = (c_{pq,i,j})$.

For modeling developable surfaces it is important that the planarity criterion also works for a thin planar quad which converges to a straight line segment. Here, the constraints in (1) serve to maintain convexity and thereby avoid singularities, but they cannot express planarity in the limit (the angle sum will tend to $2\pi$ in any case, assuming convexity). Therefore we add another planarity term: Denote the unit vectors along the edges in quad $Q_{i,j}$ by $\mathbf{e}_{i,j} := (\mathbf{v}_{i,j+1} - \mathbf{v}_{i,j})/\|\mathbf{v}_{i,j+1} - \mathbf{v}_{i,j}\|$, $\mathbf{e}_{i+1,j}$, $\mathbf{f}_{i,j} := (\mathbf{v}_{i+1,j} - \mathbf{v}_{i,j})/\|\mathbf{v}_{i+1,j} - \mathbf{v}_{i,j}\|$, and $\mathbf{f}_{i,j+1}$. Then, using the four vertices in an equal way, the planarity of $Q_{i,j}$ is enforced by the following constraints,

$$c^1_{det,i,j} := \det(\mathbf{e}_{i,j}, \mathbf{e}_{i+1,j}, \mathbf{f}_{i,j}) = 0,\ c^2_{det,i,j} := \det(\mathbf{e}_{i,j}, \mathbf{e}_{i+1,j}, \mathbf{f}_{i,j+1}) = 0,$$
$$c^3_{det,i,j} := \det(\mathbf{e}_{i,j}, \mathbf{f}_{i,j}, \mathbf{f}_{i,j+1}) = 0,\ c^4_{det,i,j} := \det(\mathbf{e}_{i+1,j}, \mathbf{f}_{i,j}, \mathbf{f}_{i,j+1}) = 0.$$

A linear combination of these constraints as used below is denoted by $\lambda_{det}^T c_{det}$. Note that these terms are included as effective planarity constraints only when computing PQ strips.

In addition, we introduce two energy terms to ensure that the resulting PQ mesh has a fair shape and stays close to the input mesh. For aesthetic design we use the fairness term $f_{fair}$, which includes simplified bending energies of the mesh's row and column polygons.

$$f_{fair} := \sum_{i,j}[(\mathbf{v}_{i+1,j} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i-1,j})^2 + (\mathbf{v}_{i,j+1} - 2\mathbf{v}_{i,j} + \mathbf{v}_{i,j-1})^2].$$

At the boundary not all vertices required by the sum exist, so in addition we define that any undefined square is set to zero. For the PQ mesh to remain close to the surface $\Phi$ defined by the original mesh, we need to minimize the distances of the perturbed mesh vertices from the original mesh surface $\Phi$ by minimizing

$$f_{close} := \sum_{i,j} \|\mathbf{v}_{i,j} - \mathbf{y}_{i,j}\|^2,$$

where $\mathbf{y}_{i,j}$ is the closest point (i.e., footpoint) on $\Phi$ to $\mathbf{v}_{i,j}$. We put the above terms together and define the Lagrangian function

$$f_{PQ} := w_1 f_{fair} + w_2 f_{close} + \lambda_{pq}^T c_{pq} + \lambda_{det}^T c_{det}. \qquad (2)$$

Note that the term $\lambda_{det}^T c_{det}$ is needed only when computing a PQ strip. SQP minimizes the energy term $w_1 f_{fair} + w_2 f_{close}$ subject to the constraints $c_{pq} = 0$ and $c_{det} = 0$. That is, the minimizer gives a PQ mesh that has a fair shape and is close to the original surface $\Phi$. The desired minimum is a stationary point of the Lagrangian $f_{PQ}$. Note that $\lambda_{pq}$ and $\lambda_{det}$ are determined automatically by the SQP method, while $w_1$ and $w_2$ are user specified constants to control relative weighting of fairness and geometric fidelity.

SQP uses a sequence of Newton-like iterations. In each round we compute the Hessians and gradients of the four terms which occur in the Lagrangian $f_{PQ}$ of (2) to form a local quadratic approximation of $f_{PQ}$ at the current point. Computation of the Hessians is straightforward, except for the squared distance term $\|\mathbf{v}_{i,j} - \mathbf{y}_{i,j}\|^2$ in $f_{close}$, which involves the footpoints $\mathbf{y}_{i,j}$ as dependent variables, since $\mathbf{v}_{i,j} - \mathbf{y}_{i,j}$ is always perpendicular to the tangent plane of $\Phi$ at $\mathbf{y}_{i,j}$. We use $[(\mathbf{v}_{i,j} - \mathbf{y}_{i,j}) \cdot \mathbf{n}_{i,j}]^2$ as a quadratic approximation of $\|\mathbf{v}_{i,j} - \mathbf{y}_{i,j}\|^2$. This approximation arises from Gauss-Newton minimization of the squared distance of $\mathbf{v}_{i,j}$ from $\Phi$ and has been successfully used for registration [Chen and Medioni 1991] and curve and surface approximation (see e.g. [Blake and Isard 1998]).

Rewrite $f_{PQ}$ in (2) in the form $f_{PQ}(x,\lambda) = f(x) - \lambda^T c(x)$, where $x$ denotes the unknown vertex coordinates, $f = w_1 f_{fair} + w_2 f_{close}$ and $-\lambda^T c(x) = \lambda_{pq}^T c_{pq} + \lambda_{det}^T c_{det}$. Let $J$ denote the Jacobian matrix of the constraints $c(x)$ and $H$ denote the Hessian matrix of $f_{PQ}(x,\lambda)$ w.r.t. $x$; (note that the contribution to $H$ by the $f_{close}$ term is a Gauss-Newton approximation). The update step $x \to x + h$ is solved from

$$\begin{bmatrix} H & -J^T \\ -J & 0 \end{bmatrix} \begin{bmatrix} h \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ c(x) \end{bmatrix}. \qquad (3)$$

We use a soft line-search strategy [Madsen et al. 2004] to determine the actual update step size $\alpha h$, $0 < \alpha \leq 1$, to ensure stable convergence and sufficient descent — so $x$ is updated by $x^* = x + \alpha h$.

Figure 5: PQ perturbation without a closeness term applied to a highly un-planar mesh consisting of only a few quads.

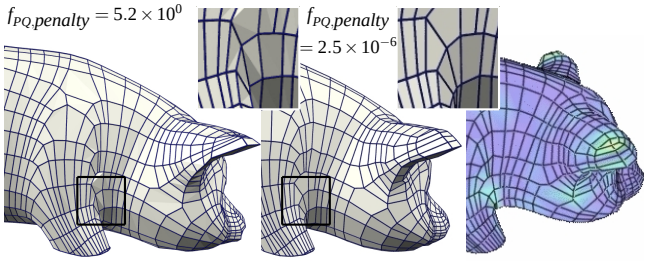


Figure 6: PQ perturbation acting on a quad-dominant mesh (left) extracted from principal curves (cf. Section 5). All faces, not only quads, are planarized (planarity is visualized via flat shading). The shape change noticeable at the ears is due to concentration of highly nonplanar quads in the original mesh. Higher geometric fidelity at the cost of fairness is easily possible. At right: color coded deviation from original mesh (max. 3% of object size).

The above coefficient matrix is highly sparse and has the size $(3M+5N) \times (3M+5N)$ or $(3M+N) \times (3M+N)$ if not including the terms $\lambda_{det}^T c_{det}$, where $M$ is the number of vertices and $N$ the number of faces of the input mesh. We use the sparse matrix packages *TAUCS* and *UMFPACK*. Our implementation of SQP works efficiently for meshes of small or medium size (up to 1000 vertices). Our experience shows that for larger meshes it is more efficient to use a penalty method: We combine the angle constraints in (1) in the function

$$f_{angle} := \sum_{i,j} (\phi_{i,j}^1 + \ldots + \phi_{i,j}^4 - 2\pi)^2, \qquad (4)$$

and similarly the determinant terms in $f_{det}$. Then, we minimize the objective function

$$f_{PQ,penalty} = w_1 f_{fair} + w_2 f_{close} + \mu f_{det} + f_{angle}. \qquad (5)$$

This is an unconstrained least squares problem which can be solved effectively by the Gauss-Newton method with L-M regularization [Kelley 1999]. We let $\mu = 1$ when considering a PQ strip, and $\mu = 0$ otherwise. For stability reasons, the coefficients $w_1$ and $w_2$ have higher values at the start of the iteration. Later, we let $w_1$ and $w_2$ tend to zero, so that planarity can be achieved with high accuracy. The following strategy works in practice: At the beginning, $w_1, w_2$ are chosen such that $f_{angle}$ dominates in (5), and they are divided by 2 in every iteration. In case of small values of $w_1$ and $w_2$, near-singular linear systems are solved via SVD.

In many cases the method exhibits much faster convergence than the SQP method. An inherent problem of the penalty method are stability deficiencies near the optimum. Therefore in practice we use the penalty method to quickly derive an initial mesh near a local minimum, and then employ SQP. PQ perturbation works very well if the input quad mesh is reasonably close to a PQ mesh (cf. Fig. 6). Fig. 5 shows an input mesh which is far from planar, so PQ perturbations results in large deviations from the original. In order to planarize $n$-gons with $n > 4$, we use the fact that condition (1) generalizes to $n$-gons.

## 3 Subdividing developables and PQ meshes

To generate a PQ mesh from a coarse control mesh, we combine the PQ perturbation algorithm with a quad based subdivision algorithm like Doo-Sabin or Catmull-Clark in an alternating way: We subdivide a given PQ mesh once, and then apply PQ perturbation to make the resulting faces planar (see Fig. 7). These two steps are iterated to generate a hierarchical sequence of PQ meshes.

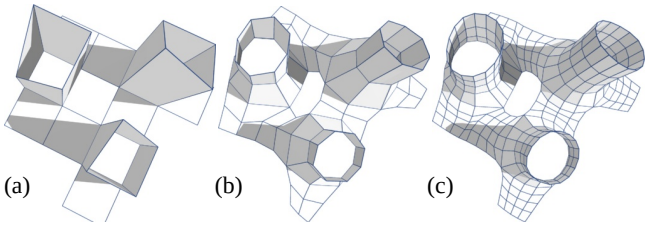


Figure 7: (a)–(c): Hierarchy of PQ meshes obtained by iterative application of Catmull-Clark subdivision and PQ perturbation.

A single PQ strip can be subdivided by applying a curve subdivision rule like Chaikin's to its boundaries, and subsequent application of PQ perturbation in order to achieve face planarity. Alternating application of these two steps is a *subdivision algorithm which generates developable surfaces*. Because of our treatment of PQ perturbation as a black box it is in general not possible to write down the limit of this subdivision process explicitly. Nevertheless it is a much simpler design tool than developable B-spline surfaces, whose control points have to satisfy a set of nonlinear constraints.

As illustrated in Fig. 8, the relation of the input PQ strip to the final developable surface is very intuitive – certainly more so than the dual control structure in terms of tangent planes, which can be used to avoid nonlinear constraints (cf. [Pottmann and Wallner 2001]).

In the perturbation phase of the algorithm, the term $f_{det}$ in (5) is important for maintaining planarity. The term $f_{angle}$ discourages self-intersecting quads and thus acts against the common problem that the singular curve enters the designed patch. Finally, $f_{fair}$ helps to prevent a zig-zag effect in adjacent quads.

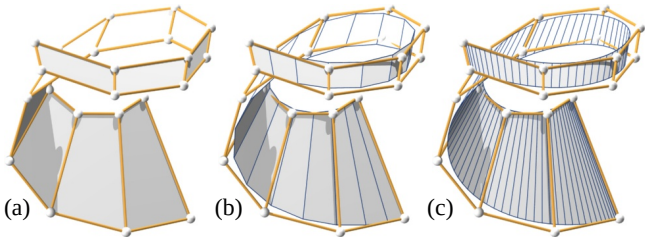


Figure 8: Developable subdivision surfaces generated with the perturbed cubic Lane-Riesenfeld algorithm; this nonlinear subdivision scheme keeps the planarity of quads and thus achieves developability of the limit. The control entity (a) is a piecewise-planar PQ strip. (b) and (c): 1 and 3 rounds of subdivision.

## 4 Conical meshes

Principal curvature lines form a special network of conjugate curves on a surface. Apart from umbilic points, where this network possesses singularities, it behaves nicely, since its curves intersect at right angles. This is not necessarily true for an arbitrary conjugate curve network; asymptotic (self-conjugate) directions give rise to degenerate situations that make such networks unsuitable for meshing purposes (Fig. 4).

A particular discretization of the network of principal curvature lines are the *circular meshes*, which are quad meshes whose quads are not only planar, but also have a circumcircle [Martin et al. 1986; Bobenko and Suris 2005]. Even though they are not the focus of the present paper, it is however easy to extend our PQ perturbation algorithm to the computation of circular meshes (see Section 5 and Fig. 17). It turns out that another discrete analogue of the principal

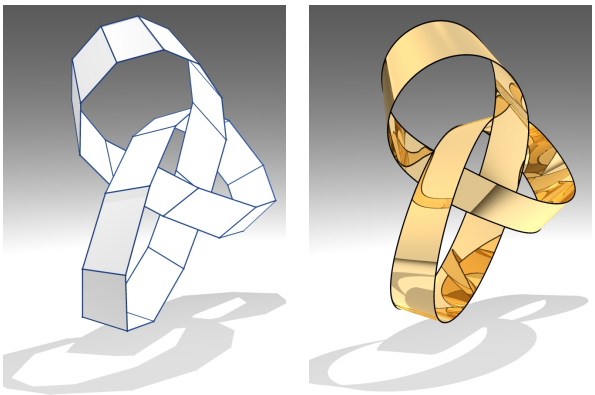

Figure 9: Developable Möbius band in the shape of a trefoil knot. Left: PQ strip as control structure. Right: Result of subdivision augmented by PQ perturbation. Numerical smoothness is $C^2$, as seen from smooth reflection lines ($f_{PQ,penalty} = 2.9 \times 10^{-11}$).

curvature lines – the conical meshes to be introduced in this section – have geometric properties essential for architectural design of freeform structures. For their computation via an augmented PQ perturbation algorithm, see Sec. 5.

A vertex $\mathbf{v}$ of a quad mesh is a *conical vertex* if all the four (oriented) face planes meeting at $\mathbf{v}$ are tangent to a common (oriented) sphere. This is equivalent to saying that these oriented face planes are tangent to a common *oriented cone of revolution* $\Gamma$ (see Fig. 10a). The axis $G$ of $\Gamma$ can be regarded as a discrete surface normal at that vertex.

We call a PQ mesh a *conical mesh* if all of its vertices of valence four are conical. For theoretical investigations, we consider only regular quadrilateral meshes whose vertices have valence 4, except for valence-2 or valence-3 vertices on the boundary. A conical mesh is in some sense dual to a circular mesh. Instead of requiring the four vertices of a quad to be co-circular, we require that the four (oriented) faces incident with a mesh vertex be tangent to an (oriented) cone of revolution. We will see that conical meshes, like circular meshes, discretize the network of principal curvature lines.

There are exactly three types of conical mesh vertices, which can be characterized geometrically as follows. A small sphere $S$ centered in a mesh vertex $\mathbf{v}$ intersects the mesh in a simple 4-sided spherical polygon $P$. If the four vertices $\mathbf{p}_i$ of $P$ cannot be contained in the same hemisphere, $\mathbf{v}$ is of the *hyperbolic* type. Otherwise (i.e., the four vertices $\mathbf{p}_i$ are contained a hemisphere) $\mathbf{v}$ is either of *elliptic type* (see Fig. 10a) or of *parabolic type*, depending on whether $P$

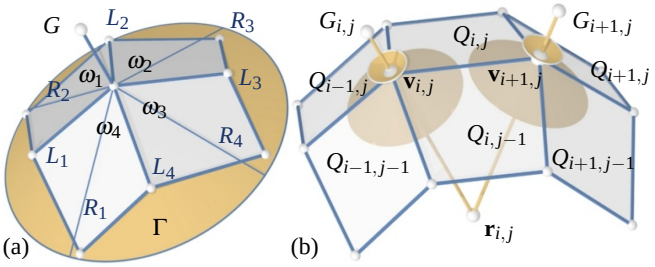


Figure 10: (a) Configuration of the faces of a conical mesh at a vertex. The faces touch the common cone $\Gamma$ along rulings $R_1, \ldots, R_4$, and have interior angles $\omega_1, \ldots, \omega_4$. (b) Faces of a conical mesh at two adjacent vertices $\mathbf{v}_{i,j}$ and $\mathbf{v}_{i,j+1}$, and the intersection point $\mathbf{r}_{i,j}$ of neighboring axes $G_{i,j}$, $G_{i,j+1}$.

is convex or not. These three types of mesh vertices are discrete analogues of hyperbolic points, elliptic points and parabolic points on a smooth surface.

**An angle criterion for conical meshes.** There is a simple condition characterizing a conical mesh in terms of the interior angles of its quads. This characterization is also important for computing conical meshes (see Equ. (6) in Section 5).

**Geometry Fact 1** *A vertex of a quad mesh is a conical vertex if and only if the angle balance $\omega_1+\omega_3=\omega_2+\omega_4$ is satisfied (see Fig. 10 for notation).*

Here we assume that no two adjacent faces incident with a mesh vertex are co-planar, for otherwise the vertex is always conical. The (oriented) great circles that carry the edges of the spherical polygon $P$ are tangent to a common (oriented) circle if and only if the vertex is conical. For elliptic vertices, Geometry Fact 1 follows from a result by A. J. Lexell which states that a convex spherical quadrilateral has an incircle if and only if the sums of opposite sides are equal. A proof of Geometry Fact 1 for all types of mesh vertices is given in [Wang et al. 2006]. As an example we now use the elliptic vertex in Fig. 10a to illustrate why the angle balance holds in this case.

Consider the right circular cone $\Gamma$ tangent to all the four faces incident with $\mathbf{v}$, the vertex of $\Gamma$. Suppose that the face plane $Q_i$ touches $\Gamma$ along the ruling $R_i$. Let $L_i$ denote the intersection line of $Q_i$ and $Q_{i+1}$. Denote $\alpha_i=\sphericalangle(L_i,R_i)$. Then, by symmetry, $\sphericalangle(L_i,R_{i+1})=\sphericalangle(L_i,R_i)=\alpha_i$. Since $\omega_i=\alpha_i+\alpha_{i+1}$, we have $\omega_1+\omega_3=\alpha_1+\alpha_2+\alpha_3+\alpha_4=\omega_2+\omega_4$.

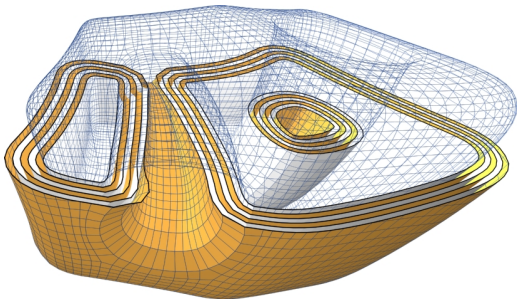

Figure 11: A conical mesh has conical offset meshes, here illustrated by a planar cut through a sequence of offsets.

**Offsetting conical meshes.** Meshes with planar faces (including triangle meshes, cf. e.g. [Kim and Yang 2005]) in general do not have the property that offsetting all faces by a fixed distance leads again to a mesh with the same connectivity, since planes meeting at a common point in general do not meet again at a common point after offsetting. Conical meshes, however, have this property — they possess conical quad meshes as offset meshes, as illustrated by Figures 11 and 12: The faces of a conical mesh incident with a vertex $\mathbf{v}_{i,j}$ are tangent to an oriented cone with axis $G_{i,j}$. After offsetting, they are still tangent to a cone with the *same* axis. This behavior of the discrete surface normal $G_{i,j}$ is consistent with the behavior of the ordinary surface normal of a smooth surface under offsetting (which also does not change).

*Remark:* It is easy to show that any PQ mesh having the offsetting property is a conical mesh; that is, the offsetting property is a characterizing property of conical meshes. Offsetting planes by a fixed distance along their normal vector is a simple instance of a Laguerre transformation [Cecil 1992]. It is not difficult to see that general Laguerre transformations map conical meshes to conical meshes, whereas the property of a mesh being circular is preserved under Möbius transformations [Bobenko and Suris 2005].

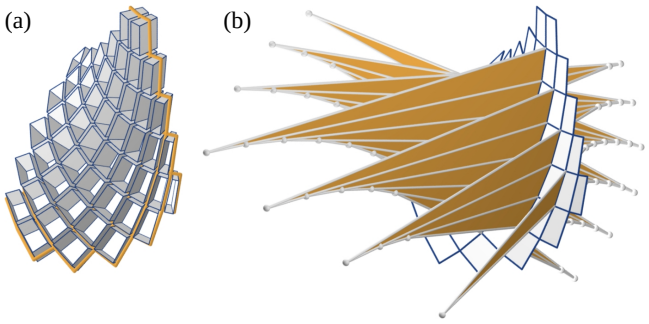


Figure 12: A conical mesh discretizes the network of principal curvature lines. (a) A conical mesh has conical offset meshes and a discretely orthogonal support structure connecting the offsets. (b) Cone axes of neighboring vertices intersect in the discrete principal curvature centers. Connecting these axes leads to discrete row and column developables orthogonal to the mesh.

**The normals of a conical mesh.** Starting from a planar mesh with quads $Q_{i,j}$, we construct a mesh in the unit sphere whose vertices $\mathbf{n}_{ij}$ are the unit normal vectors of $Q_{ij}$. It is called the *spherical image* of the original PQ mesh. For conical meshes, the spherical image has special properties: As the four faces which meet in a vertex $\mathbf{v}_{k,l}$ are tangent to a common cone $\Gamma_{k,l}$, the normal vectors of these faces enclose the same angle with the cone's axis. Thus these four normal vectors lie in a circle contained in the unit sphere; the spherical center of this circle gives the unit direction vector of $G_{k,l}$. Thus *the spherical image of a conical mesh is a circular mesh*. This property is a discrete analogue to the well known fact that the spherical image of the network of principal curvature lines is an orthogonal curve network on the sphere.

**Support structures of conical meshes.** Fig. 10b illustrates two neighboring vertices $\mathbf{v}_{i,j}$ and $\mathbf{v}_{i+1,j}$ of a conical mesh. There are two faces $Q_{i,j}$ and $Q_{i,j-1}$ containing both vertices. These two faces are tangent to both cones $\Gamma_{i,j}$ and $\Gamma_{i+1,j}$. It follows that their axes $G_{i,j}$ and $G_{i+1,j}$ lie in the bisector plane of the oriented faces $Q_{i,j}$ and $Q_{i,j-1}$. The important fact derived here is that neighboring axes (discrete surface normals) are co-planar, and they are contained in a plane orthogonal to the mesh in a discrete sense.

It follows that an edge of the mesh, the discrete normals at its endpoints, and the corresponding edge of any offset mesh, lie on a common plane. This property can be used to build 'orthogonal' support structures as shown in Figures 1, 12 and 15, which are important for the construction of freeform glass structures based on conical meshes. Co-planarity of axes $G_{i,j}$ and $G_{i,j+1}$ implies:

**Geometry Fact 2** *Successive discrete normals of a conical mesh along a row or column are co-planar and therefore form a discrete developable surface (see Fig. 12).*

Recall that the surface normals of a smooth surface along a curve constitute a developable surface if and only if that curve is a principal curvature line. Fact No. 2 is a discrete analogue of this classical result, and shows the following important property:

**Geometry Fact 3** *If a subdivision process, which preserves the conical property, refines a conical mesh and in the limit produces a curve network on a smooth surface, then this limit curve network is the network of principal curvature lines.*

**Focal surfaces with conical meshes.** In a conical mesh, neighboring axes (discrete surface normals) in a row intersect, and so do neighboring axes in a column. These intersection points are discrete row and column *curvature centers*, which define, in general, a two-sheet discrete focal surface. It is easy to see that the two quad meshes defined by the row centers and the column centers are

actually PQ meshes, and that singularities of discrete offsets occur at these two discrete focal sheets. This is analogous to the smooth case [Porteous 1994].

*Remark:* We might ask if there are meshes which are both circular and conical. The answer is in the affirmative, and it is not difficult to construct some. Interesting examples of conical meshes of constant cone opening angle, which are at the same time circular of constant circle radius, are derived from the discrete surfaces of constant negative Gaussian curvature of [Wunderlich 1951]. One of them is shown in Fig. 12. However, meshes with both properties may be too inflexible to be useful for modeling and approximation.

## 5 Computing conical meshes

We would like to approximate a surface $\Phi$, which is given in any representation, by a conical (or circular) mesh. Since both types of meshes converge to principal curvature lines under refinement, it is a good choice to use a quad mesh extracted from principal curvature lines (e.g., the meshes from [Alliez et al. 2003]) as input for an optimization algorithm which achieves the conical or circular property by perturbing the vertices as little as possible.

**Robust computation of principal curves.** For computing principal curves, we employ a method different from previous approaches [Cohen-Steiner and Morvan 2003; Clarenz et al. 2004]. In view of the desired average size of faces in a principal mesh, we find it appropriate to use as input *robust principal curves on a given scale r* (Fig. 13), which are computed as follows. The procedure analyzes neighborhoods of points $\mathbf{p}$ of the given surface $\Phi$. We choose a kernel radius $r$, which defines the scale on which we would like to work. The domain of space which, locally around $\mathbf{p}$, lies to one side of the surface is denoted by $D$. For each point $\mathbf{p} \in \Phi$ we perform a principal component analysis (PCA) of the set $N^r(\mathbf{p}) = B^r(\mathbf{p}) \cap D$, where $B^r(\mathbf{p})$ is a ball of radius $r$ centered in $\mathbf{p}$. This means that we compute the barycenter $\mathbf{s}^r$ of $N^r$ and the eigenvectors $\mathbf{t}_1^r, \mathbf{t}_2^r, \mathbf{t}_3^r$ and corresponding eigenvalues $\lambda_1^r \le \lambda_2^r \le \lambda_3^r$ of the covariance matrix $J := \int_{N^r} (\mathbf{x}-\mathbf{s}^r) \cdot (\mathbf{x}-\mathbf{s}^r)^T d\mathbf{x}$. In the limit $r \to 0$, $\mathbf{t}_3^r$ converges to the surface normal at $\mathbf{p}$, and $\mathbf{t}_1^r, \mathbf{t}_2^r$ converge to the principal directions. What we actually compute is a kind of average of these geometric characteristics over a small neighborhood of $\mathbf{p}$. Most importantly, directions $\mathbf{t}_1^r, \mathbf{t}_2^r$ are more robust against noise and minor perturbations than those of classical differential geometry or than those in [Clarenz et al. 2004], which are computed via PCA on the surface patch neighborhood $B^r(\mathbf{p}) \cap \Phi$. For proofs and details on efficient implementation we refer to [Pottmann et al. 2005].

The directions $\mathbf{t}_1^r, \mathbf{t}_2^r$ need not be tangent to the given surface at $\mathbf{p}$. However, we can still obtain meaningful principal directions at $\mathbf{p}$ if we just project them onto the tangent plane at $\mathbf{p}$. The direction of this projection shall be given by the third eigenvector $\mathbf{t}_3^r$, which estimates the surface normal. The projected directions do not have to be orthogonal anymore, which is actually no loss and rather enhances stability when we now integrate these two vector fields to obtain *principal curves at the chosen scale r* (see Fig. 13). Our algorithm for vector field integration and quad mesh extraction is based on ideas in [Alliez et al. 2003; Dong et al. 2005; Marinov and Kobbelt 2004]. We do not give more details, since this part is not considered the topic of the present paper.

**Conical and circular optimization.** It is not difficult to modify the PQ perturbation algorithm from Sec. 2.2 so that it produces conical meshes. For perturbation into a conical mesh, we keep the constraints of (1) and, according to Geometry Fact No. 1, for each vertex add the constraint

$$\omega_{i,j}^1 + \omega_{i,j}^3 - \omega_{i,j}^2 - \omega_{i,j}^4 = 0. \tag{6}$$

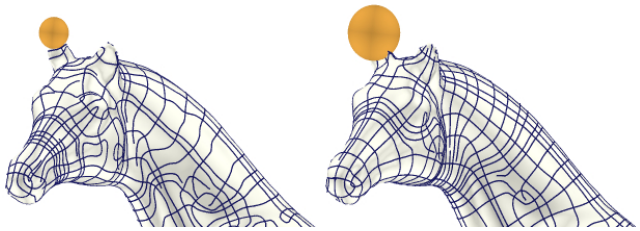

Figure 13: Principal curves computed with different kernel radii.

The perturbation algorithm for computing a circular mesh is similar. A quad is planar, convex, and has a circumcircle, if and only if the four angles enclosed by its four edges have the property

$$\phi_{i,j}^1 + \phi_{i,j}^3 - \pi = 0, \quad \phi_{i,j}^2 + \phi_{i,j}^4 - \pi = 0. \tag{7}$$

We therefore replace the planarity constraint $\phi_{i,j}^1 + \ldots + \phi_{i,j}^4 - 2\pi = 0$ in Equ. (1) by the two constraints in (7). The modifications of the penalty method proceed along the same lines. We do not enforce the conical condition for vertices of valence greater than four, as in architectural applications such vertices are expected to get special treatment anyway. We could however easily make such a vertex conical by imposing the conical condition for any four faces adjacent to the vertex. Similarly, for circular meshes the circular condition is not enforced for $n$-gons with $n > 4$, except for artificial 5-gons which arise from quads at T junctions, where the original quad's vertices are made cocircular.

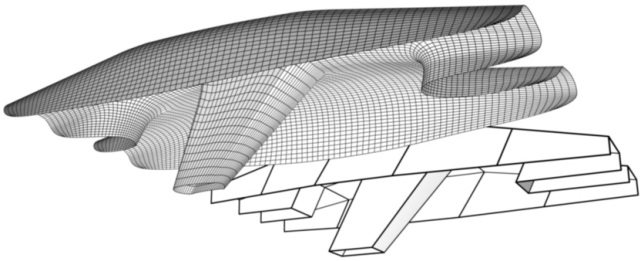

Figure 14: The conical mesh in front was obtained by a combination of Catmull-Clark subdivision and conical optimization from the control mesh behind. This conical mesh is the basis for the glass structure in Fig. 1.

## 6 Results and discussion

**Developable surfaces.** Our experiments show that the proposed subdivision approach to developable surface modeling is a powerful new tool (see Figures 9 and 16). The elimination of the singular curve from the actually designed patch is simplified by the multiscale approach inherent to subdivision: Planarization results in convex quads and thus eliminates singularities from the designed patch at each subdivision level (Fig. 16). This multiscale elimination of the singular curve appears to be more efficient than the methods known in the literature [Pottmann and Wallner 2001].

The PQ perturbation method described in Section 2.2 makes use of a reference surface. If only a coarse PQ strip is available as a control structure for a smooth developable, such a reference surface may be generated by applying the *unperturbed* subdivision rule to the strip. Numerical evidence for the $C^2$ smoothness of the perturbed cubic Lane-Riesenfeld rule is furnished by the apparent smoothness of reflection lines in Fig. 9.

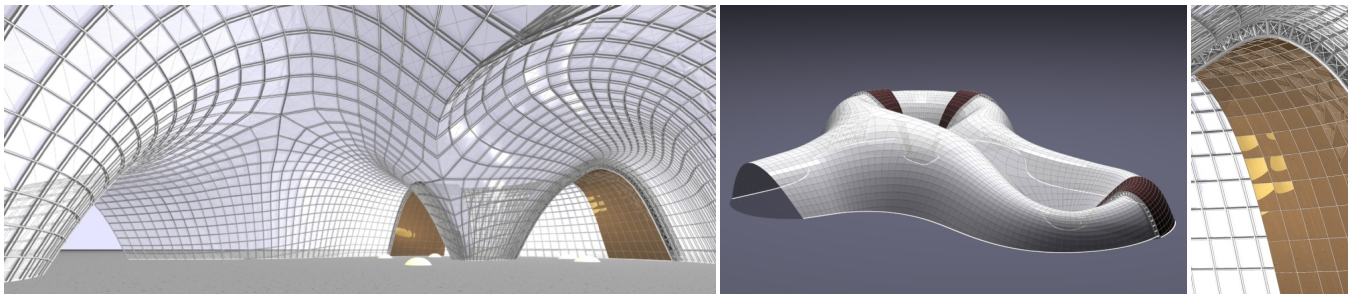

Figure 15: Design studies with conical meshes and their offset meshes produced by subdivision and conical perturbation. The figure shows a wide-angle perspective of the interior (left), an exterior view (center), and an offset detail (right).

**Conical and circular meshes.** The combination of subdivision and conical/circular perturbation produces high quality meshes suitable for aesthetic design (see Figures 1, 14, and 15). Without subdivision, it is essential that perturbation which aims at principal meshes (circular or conical) is applied to a mesh which is not too far away from a principal mesh. This is achieved by deriving a mesh from principal curves (cf. Fig. 17). Otherwise, we either get large deformations, or – if the surface is subject to further constraints such as fixed points or closeness to a reference surface – we may obtain self-intersections, creases, and other undesirable effects.

**Efficiency.** The performance of our non-optimized PQ perturbation code depends not only on the size of the input data, but also on the geometry and the nature of nearness constraints. To give a few numbers, on a 2 GHz PC we experienced computation times of 0.08 (penalty) and 0.75 seconds (sequential quadratic programming) for PQ perturbation applied to the trefoil knot with 336 faces in Fig. 9. Thus interactive modeling of developable surfaces is easily possible. Total mesh computation time for Fig. 15 was 13 seconds, of which 80% was for 4 iterations of conical perturbation (penalty) at the finest subdivision level with 5951 vertices. There is still room for improvement when processing large meshes. The number of iterations can be as high as 20–50 for the SQP method with a closeness term present. Surface approximation by conical and circular meshes would certainly benefit from even better initial meshes, but this aspect of quad-dominant remeshing is not a focus of the present paper.

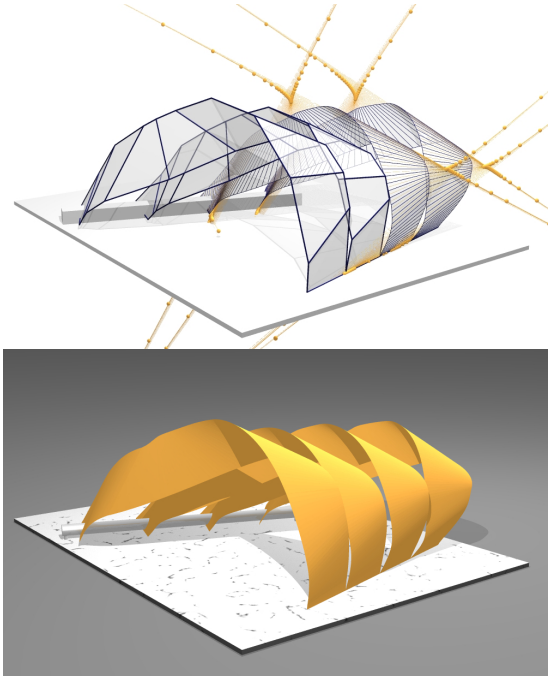

Figure 16: Design studies with developable surfaces. Two of four developable strips in the top figure show control structures, and the other two show the result of subdivision together with singular curves.

**Convergence.** It should be mentioned that there are meshes where PQ perturbation fails because of topological obstructions. On the other hand we did not encounter problems with meshes based on principal curves, and PQ perturbation is capable of large deformations when that is necessary for achieving planarity.

## 7 Conclusion and future work

We have shown how to construct and approximate surfaces with meshes composed of planar quadrilaterals. To our knowledge, approximation with conical, circular, or even just PQ meshes has not been treated before. Combining an optimization algorithm for the computation of these PQ meshes with quad-based subdivision algorithms results in a powerful modeling tool. It adapts subdivision for applications in architecture and also provides a new way of modeling developable surfaces. In particular, we have introduced and studied conical meshes, which discretize the network of principal curvature lines. They are well suited for designing free-form glass structures in architecture, and provide a simple and natural offsetting operation and the construction of a support structure from discrete surface normals.

The many directions for future research include studies in discrete differential geometry such as the investigation and computation of special discrete surfaces (such as surfaces of constant mean or Gaussian curvature, Willmore surfaces, etc.) in a principal mesh and especially conical mesh representation. This would also be very welcome for aesthetic design [Sullivan 2005]. Further research directions are the incorporation of statics, stability and other aspects of construction and fabrication into the computation of quad meshes for architecture and the use of conical meshes and their offsets as a discrete model for simulation problems with shells and plates.

**Acknowledgements** We gratefully acknowledge the architectural input by Sigrid Brell-Cokcan. The railway station of Fig. 1 was created by Benjamin Schneider. Markus Forstner is the designer of both the conical mesh in Fig. 15 and the developables in Fig. 16. The 3D data sets of animals used in this paper have been obtained from the *aim@shape* repository. This research was support by the Austrian Science Fund (Grant No. S9206-N12).

## References


ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Trans. Graphics 22*, 3, 485–493.

AUMANN, G. 2004. Degree elevation and developable Bézier surfaces. *Comp. Aided Geom. Design 21*, 661–670.

BLAKE, A., AND ISARD, M. 1998. *Active Contours*. Springer.

BOBENKO, A., AND SURIS, Y., 2005. Discrete differential geometry. Consistency as integrability. preprint, *http://arxiv.org/abs/math.DG/0504358*.

BOBENKO, A., MATTHES, D., AND SURIS, Y. 2003. Discrete and smooth orthogonal systems: $C^\infty$-approximation. *Int. Math. Res. Not.*, 45, 2415–2459.

BOBENKO, A., HOFFMANN, T., AND SPRINGBORN, B. A. 2006. Minimal surfaces from circle patterns: Geometry from combinatorics. *Annals of Mathematics 164*, 231–264.

CECIL, T. 1992. *Lie Sphere Geometry*. Springer.

CERDA, E., CHAIEB, S., MELO, F., AND MAHADEVAN, L. 1999. Conical dislocations in crumpling. *Nature 401*, 46–49.

CHEN, Y., AND MEDIONI, G. 1991. Object modeling by registration of multiple range images. In *Proc. IEEE Conf. on Robotics and Automation*.

CHU, C. H., AND SEQUIN, C. 2002. Developable Bézier patches: properties and design. *Computer-Aided Design 34*, 511–528.

CIPOLLA, R., AND GIBLIN, P. 2000. *Visual Motion of Curves and Surfaces*. Cambridge University Press.

CLARENZ, U., RUMPF, M., AND TELEA, A. 2004. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Trans. Visual. Comp. Graphics 10*, 516–524.

COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted Delaunay triangulations and normal cycle. In *Proc. 19th annual symposium on Computational geometry*, ACM, 312–321.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. Variational shape approximation. *ACM Trans. Graphics 23*, 3, 905–914.

DESBRUN, M., GRINSPUN, E., AND SCHRÖDER, P. 2005. *Discrete Differential Geometry*. SIGGRAPH Course Notes.

DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

DONG, S., KIRCHER, S., AND GARLAND, M. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comp. Aided Geom. Design 22*, 392–423.

FREY, W. 2004. Modeling buckled developable surfaces by triangulation. *Computer-Aided Design 36*, 4, 299–313.

GLYMPH, J., SHELDEN, D., CECCATO, C., MUSSEL, J., AND SCHOBER, H. 2002. A parametric strategy for freeform glass structures using quadrilateral planar facets. In *Acadia 2002*, ACM, 303–321.

HILDEBRANDT, K., POLTHIER, K., AND WARDETZKY, M. 2005. On the convergence of metric and geometric properties of polyhedral surfaces. Tech. Rep. 05-24, Zuse Institute Berlin.

JULIUS, D., KRAEVOY, V., AND SHEFFER, A. 2005. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum (Proc. Eurographics 2005) 24*, 3, 581–590.

KELLEY, C. T. 1999. *Iterative Methods for Optimization*. SIAM.

KILIAN, A. 2006. *Design exploration through bidirectional modeling of constraints*. PhD thesis, Massachusets Inst. Technology.

KIM, S.-J., AND YANG, M.-Y. 2005. Triangular mesh offset for generalized cutter. *Computer-Aided Design 37*, 10, 999–1014.

MADSEN, K., NIELSEN, H. B., AND TINGLEFF, O., 2004. Optimization with constraints. Lecture Notes.

MARINOV, M., AND KOBBELT, L. 2004. Direct anisotropic quad-dominant remeshing. In *Proc. Pacific Graphics*, 207–216.

MARTIN, R. R., DE PONT, J., AND SHARROCK, T. J. 1986. Cyclide surfaces in computer aided design. In *The mathematics of surfaces*, J. A. Gregory, Ed. Clarendon Press, Oxford, 253–268.

MITANI, J., AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graphics 23*, 3, 259–263.

POLTHIER, K. 2002. *Polyhedral surfaces of constant mean curvature*. Habilitationsschrift TU Berlin.

PORTEOUS, I. R. 1994. *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge Univ. Press.

POTTMANN, H., AND WALLNER, J. 2001. *Computational Line Geometry*. Springer.

POTTMANN, H., HUANG, Q.-X., YANG, Y.-L., AND KÖLPL, S., 2005. Integral invariants for robust geometry processing. Geometry Preprint 146, TU Wien.

RAY, N., LI, W.-C., LEVY, B., SHEFFER, A., AND ALLIEZ, P., 2005. Periodic global parameterization. INRIA preprint.

SAUER, R. 1970. *Differenzengeometrie*. Springer.

SEQUIN, C. 2004. CAD tools for aesthetic engineering. *CAD & Appl. 1*, 301–309.

SULLIVAN, J. 2005. The aesthetic value of optimal geometry. In *In The Visual Mind II*, M. Emmer, Ed. MIT Press, 547–563.

WANG, C., AND TANG, K. 2004. Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *Vis. Computer 20*, 521–539.

WANG, W., WALLNER, J., AND LIU, Y., 2006. An angle criterion for conical mesh vertices. Geometry Preprint 157, TU Wien. *http://www.geometrie.tuwien.ac.at/ig/papers/tr157.pdf*.

WUNDERLICH, W. 1951. Zur Differenzengeometrie der Flächen konstanter negativer Krümmung. *Sitz. Öst. Ak. Wiss. 160*, 41–77.

YAMAUCHI, H., GUMHOLD, S., ZAYER, R., AND SEIDEL, H.-P. 2005. Mesh segmentation driven by Gaussian curvature. *Visual Computer 21*, 659–668.


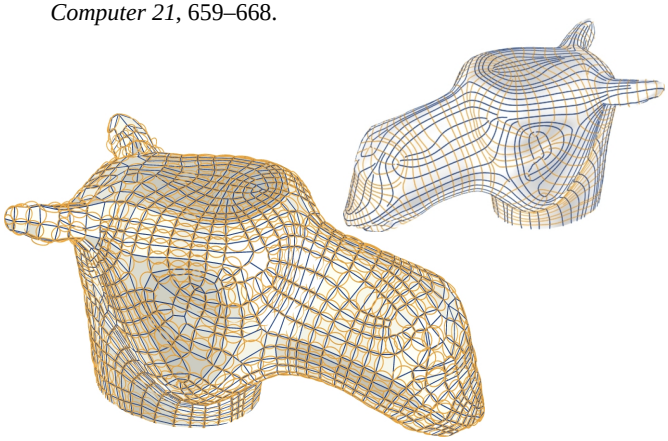

Figure 17: Circular mesh generated by optimizing a mesh generated from principal curves (top right).