Curved Folding

Martin Kilian
TU Vienna
Evolute

Simon Flöry
TU Vienna
Evolute

Zhonggui Chen
TU Vienna
Zhejiang University

Niloy J. Mitra
IIT Delhi

Alla Sheffer
UBC

Helmut Pottmann
TU Vienna

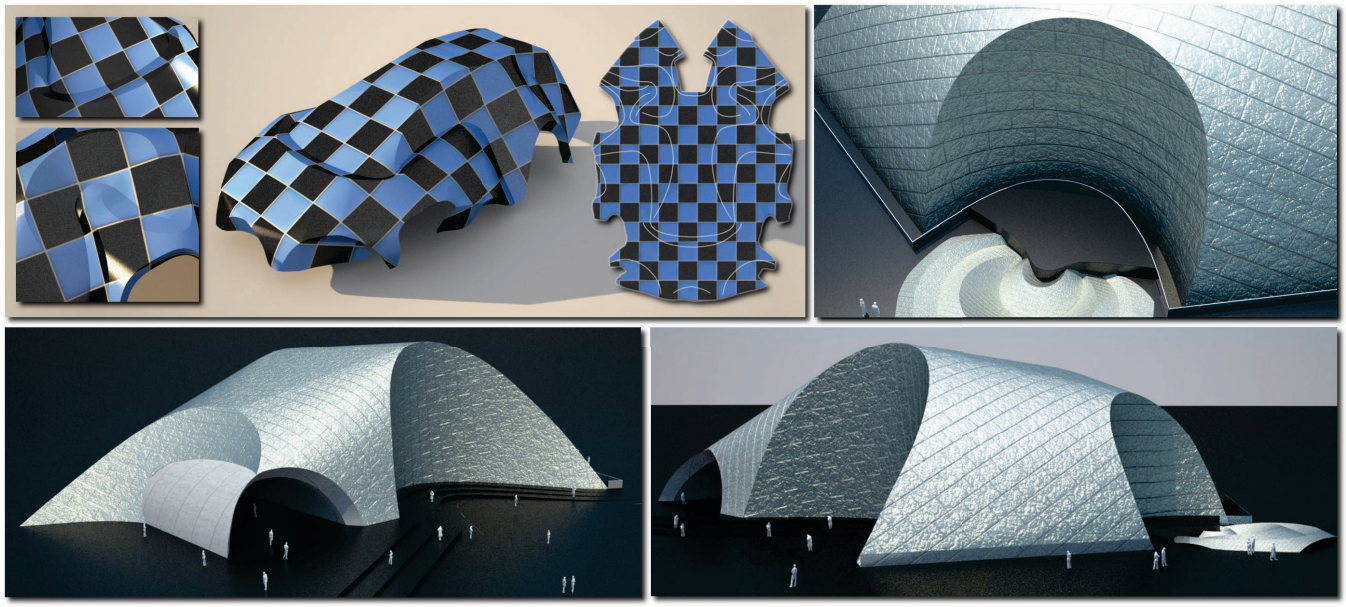

Figure 1: *Top left: Reconstruction of a car model based on a felt design by Gregory Epps. Close-ups of the hood and the rear wheelhouse are shown on the left. The fold lines are highlighted on the car's development. Top right and bottom: Architectural design. All shown surfaces can be isometrically unfolded into the plane without cutting along edges and can thus be texture mapped without any seams or distortions.*

Abstract

Fascinating and elegant shapes may be folded from a single planar sheet of material without stretching, tearing or cutting, if one incorporates curved folds into the design. We present an optimization-based computational framework for design and digital reconstruction of surfaces which can be produced by curved folding. Our work not only contributes to applications in architecture and industrial design, but it also provides a new way to study the complex and largely unexplored phenomena arising in curved folding.

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: computational differential geometry, computational origami, architectural geometry, industrial design, developable surface, folding, curved fold, isometry, digital reconstruction.

1 Introduction

Developable surfaces appear naturally when spatial objects are formed from planar sheets of material without stretching or tearing. Paper models such as origami art are prominent examples. The striking elegance of models folded from paper, such as those by David Huffman [Wertheim 2004], arises particularly from creases known as *curved folds*. Such folds can be generated from a single planar sheet. Early investigations of curved folds are due to Huffman [1976]. More recently, computational geometers became interested in folding problems and computational origami [Demaine and O'Rourke 2007]. Their work concentrates on piecewise linear structures; according to [Demaine and O'Rourke 2007], 'little is known' in the curved case. While industrial designers have started to explore the technique of curved folding (www.curvedfolding.com), current geometric modeling systems still lack any support for such a design process (in fact, most CAD systems are lacking a proper treatment of developable surfaces). As a result, Frank O. Gehry, who favors developable shapes for many of his architectural designs (cf. [Shelden 2002]), has initiated the development of a CAD module for developable surfaces by his technology company. To the best of our knowledge, curved folding is not present in that module either.

Motivated by the potential and interest in the use of curved folding for various geometric design purposes, we investigate this topic from the perspective of geometric modeling.

Related work. Developable surfaces are well studied in differential geometry [do Carmo 1976]. They are surfaces which can be unfolded into the plane while preserving the length of all curves on the

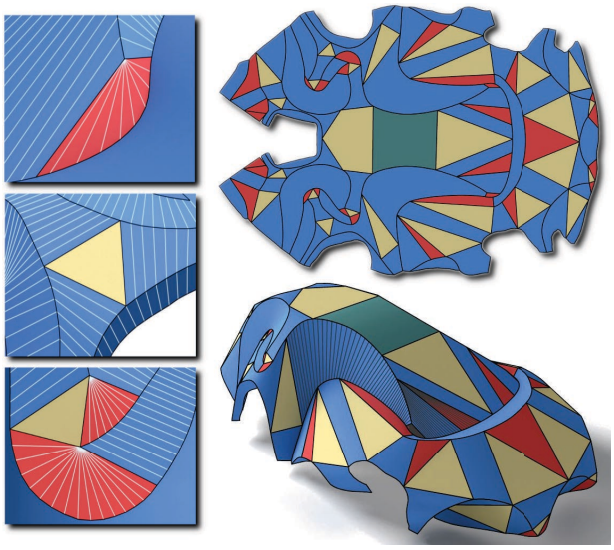

Figure 2: *The car model of Figure 1 and its development (top right). The patch decomposition into torsal ruled surfaces is shown using the following color scheme: planes are shown in yellow, cylinders in green, cones in red, and tangent surfaces in blue. Sample rulings are shown on some patches of the windshield and the side window. Such a segmentation is essential for NURBS surface fitting and manufacturing.*

surface. Developable surfaces are composed of planar patches and patches of ruled surfaces with the special property that all points of a ruling have the same tangent plane. Such *torsal ruled surfaces* consist of pieces of cylinders, cones, and tangent surfaces, i.e., their rulings are either parallel, pass through a common point, or are tangent to a curve (*curve of regression*), respectively. Whereas a torsal ruled surface has only one continuous family of rulings, general smooth developable surfaces are usually a much more complicated combination of patches. The presence of planar parts is the main source of this huge variety of possibilities. The level of difficulty is further increased if one admits creases, i.e., curved folds (see Figure 2).

In *geometric design*, various ways of treating developability have been pursued: as a constraint in tensor product B-spline surfaces of degree $(n, 1)$ [Chu and Sequin 2002; Aumann 2004], aiming only at approximate developability [Pérez and Suárez 2007], viewing the surfaces as sets of their tangent planes [Pottmann and Wallner 2001], subdividing strips of planar quads [Liu et al. 2006], or computing with triangle meshes and a local convexity constraint [Frey 2004; Wang and Tang 2004; Subag and Elber 2006; Wang 2008]. Bo and Wang [2007] model paper strips as rectifying developables of one of their geodesics. Digital reconstruction of torsal ruled surfaces employing a plane-geometric approach is the topic of [Peternell 2004].

Mesh parametrization and segmentation using developable surfaces has been investigated in [Julius et al. 2005] and [Yamauchi et al. 2005]. Rose et al. [2007] show how to compute developable surfaces from boundary curves, and present a strategy for selecting an optimal solution. Several algorithms have been proposed for the construction of papercraft models [Mitani and Suzuki 2004; Massarwi et al. 2006; Shatz et al. 2006] using folds along line segments. In all these papers, triangle meshes are used to represent developable surfaces.

Only a few contributions deal with the difficult analysis and computation of creases in developable surfaces. Most of them concentrate on conical creases [Kergosien et al. 1994; Cerda et al. 1999; Frey 2004]. Starting from conical folds, Cerda et al. [2004] investigated gravity-induced draping of naturally thin flat sheets.

Contributions and overview. We present an optimization-based computational framework for the design and reconstruction of general developable surfaces with a strong focus on curved folding applications. Our main contributions are as follows:

• We employ quad meshes with planar faces as a discrete differential geometric representation of developable surfaces, and for this representation introduce new ways of computing curvatures and bending energy. Moreover, we discuss curved folds from the discrete perspective (Section 2).

• In Section 3, we introduce the core of our work, a novel optimization algorithm which allows us to compute developable surfaces with curved folds that are isometric to a given planar sheet, while at the same time achieving additional objectives such as approximation of given geometric data, aesthetics, and minimization of bending energy.

• Section 4 presents various ways in which the basic optimization algorithm can be used for design. Even simple applications lead to new results, such as modeling developable strips of minimal bending energy.

• Another main contribution of our work is digital reconstruction of objects exhibiting developable surfaces with curved folds. Algorithms for preprocessing the input data in order to make the algorithm of Section 3 applicable to digital reconstruction are discussed in Section 5.

• Combining digital reconstruction, optimization and recently introduced algorithms for computing in shape space [Kilian et al. 2007] we have a rich toolbox for geometry processing with curved folds. This is demonstrated by means of a few application scenarios in Section 6. Finally, we summarize our main results, and address directions for future research within the largely unexplored area of curved folding.

2 Discrete developable surfaces

Developable surfaces. As our basic representation of developable surfaces we employ quad-dominant meshes with planar faces, which is also the representation of choice for discrete differential geometry [Sauer 1970; Bobenko and Suris 2005].

A strip of planar quadrilaterals (Figure 3, left) is a discrete model of a torsal ruled surface. Such a 'PQ strip' can be trivially unfolded into the plane without distortions. The edges where successive quads join together give us the discrete rulings. In general they form the edge lines of the regression polyline $\mathbf{r}_0, \mathbf{r}_1, \ldots$; in special cases the discrete rulings are parallel, or pass through a fixed point. A refinement process which maintains planarity of quads generates, in the limit, a torsal ruled surface Σ (Figure 3, right). Its rulings are the limits of the discrete rulings, which in general are tangent to the regression curve $\mathbf{r}(t)$, and in special cases are parallel (cylinder), or pass through a fixed point (cone).

The representation of developable surfaces as PQ strips provides various advantages over triangle meshes: (i) developability is guaranteed by planarity of faces and the development is easily obtained, (ii) subdivision applied to PQ strips provides a simple and computationally efficient multi-scale approach [Liu et al. 2006], (iii) the regression curve – which is singular on the surface and thus needs to be controlled – is present in a discrete form, and (iv) the curvature behavior can be easily estimated as shown next.

Curvatures and bending energy. The rulings on a smooth developable surface constitute one family of principal curvature lines corresponding to vanishing principal curvature. The second family is given by the orthogonal trajectories of the rulings which integrate the directions of non-vanishing principal curvatures κ_2. We are interested in a discrete definition of κ_2, and the related bending energy $E_{bend} = \int \kappa_2^2 dA$.

The rulings on a PQ strip are given by the edge lines $R_i := \mathbf{p}_i\mathbf{q}_i$ (cf. Figure 4a). As a discrete principal curvature line we take a polyline C with vertices $\mathbf{c}_i \quad R_i$ whose edges $\mathbf{c}_i\mathbf{c}_{i+1}$ are orthogonal to the inner bisectors S_i of R_i and R_{i+1}. In other words, each edge of C intersects consecutive rulings at the same angle (this definition is also motivated by an analogous definition in the context of circular meshes [Bobenko and Suris 2005]). We want to attach a surface normal vector $\mathbf{n}_i$ to the edge midpoint $\mathbf{m}_i = (\mathbf{c}_i + \mathbf{c}_{i+1})/2$, and take the normal vector of the plane P_i spanned by edges R_i, R_{i+1} for that purpose. The unit vectors $\mathbf{n}_i$ form the Gaussian image of C and thus it is natural to define the principal curvature κ_2 at $\mathbf{c}_i$ via:

$$\kappa_2(\mathbf{c}_i) := \frac{\mathbf{n}_i - \mathbf{n}_{i-1}}{\mathbf{c}_i - \mathbf{m}_{i-1} \;+\; \mathbf{c}_i - \mathbf{m}_i} = N_i/L_i. \qquad (1)$$

This definition has the advantage that the denominator L_i can be computed in the planar development of the strip, while only the numerator $N_i := \mathbf{n}_i - \mathbf{n}_{i-1}$ requires the embedding in space. Note that the curvature κ_2 always has a positive sign, in contrast to usual definitions. As we used it in its squared form only, this does not matter.

Using the notation of Figure 4 the discrete bending energy $E_i = \int \kappa_2^2 dA$ of a region bounded by two discrete principal curvature lines $C, \bar{C}$ at distance $h := \bar{\mathbf{c}}_i - \mathbf{c}_i$ and two bisectors $S_{i-1}S_i$ (depicted by the brown highlight in Figure 4a) is given as

$$E_i = w_i \; \mathbf{n}_i - \mathbf{n}_{i-1} \;^2. \qquad (2)$$

The weight w_i associated with the ruling R_i is given by

$$w_i = h(\ln \bar{L}_i - \ln L_i)/(\bar{L}_i - L_i), \qquad (3)$$

where L_i is the denominator of Equation (1). As $\bar{L}_i \quad L_i$, in the limit, we get $w_i = h/L_i$. Thus the bending energy of a general PQ strip can be simply approximated by a sum of energies E_i of the type (2). Note that we cannot directly apply known formulae for discrete bending energies [Desbrun et al. 2005] since those assume that all edge lengths tend to zero if one passes to the limit. This is not the case for the rulings in our approach.

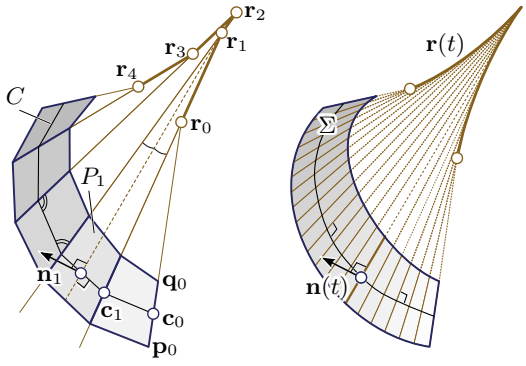

Figure 3: *A PQ strip (left) is a discrete model of a developable surface Σ (right). The intersections of edges $\mathbf{p}_i\mathbf{q}_i$ of adjacent planar quads generate the regression polyline $\mathbf{r}_i$. In the limit of a refinement process, this regression polyline becomes the regression curve $\mathbf{r}(t)$. Polylines C, whose edges $\mathbf{c}_i\mathbf{c}_{i+1}$ intersect inner bisectors of consecutive discrete rulings at right angles, are discrete versions of principal curvature lines, and serve for the definition of discrete curvatures. The unit normals to planar quads P_i are denoted by $\mathbf{n}_i$.*

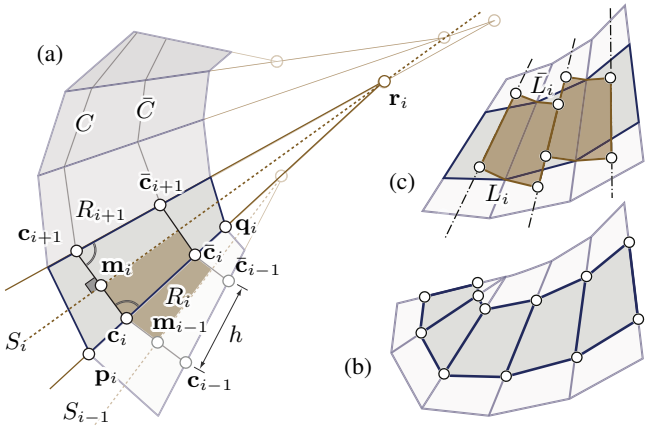

Figure 4: *(a) The rulings of a PQ strip are given by the edge lines $R_i := \mathbf{p}_i\mathbf{q}_i$. The inner bisector of R_iR_{i+1} is denoted by S_i. Edges $\mathbf{c}_i\mathbf{c}_{i+1}$ of the polyline C intersect S_i orthogonally. If $\mathbf{m}_i$ is the mid-point of $\mathbf{c}_i\mathbf{c}_{i+1}$, and $\mathbf{n}_i$ is the normal to the plane $P_i := R_iR_{i+1}$, then the principal curvature at $\mathbf{c}_i$ is naturally defined by Equation* (1)*. (b) To avoid infinite curvatures at cone vertices, computation takes place on a slightly shrunken strip. (c) The total bending energy of a region (brown) bounded by $S_{i-1}S_i$ and two principal curvature lines $C\bar{C}$ is given by Equations* (2)*,* (3)*.*

Curved folds. In the smooth setting, the following fact about curved folds is well known (see e.g. [Huffman 1976]): At each point of a fold curve c, the osculating plane of c is a bisecting plane of the tangent planes on either side of the fold. This follows immediately from the identical geodesic curvatures of the fold curve c with respect to the two adjacent developable surfaces S_1 and S_2. Hence, given the surface on one side of a fold curve, we can compute (part of) the other as the envelope of planes, obtained by reflecting the tangent planes about the osculating planes of c. This is discussed in some detail in [Pottmann and Wallner 2001], but one finds only that part of S_2 whose rulings meet c. Thus, the approach is not sufficient for most of our tasks where, in addition, multiple folds may appear, and the locations of such fold curves only become known in the process of optimization. In contrast to the smooth setting, in the discrete case there are more degrees of freedom in choosing the surface S_2 as described next.

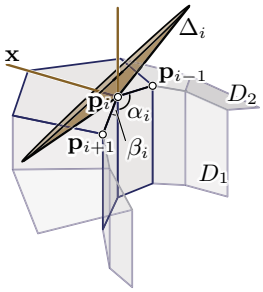

Figure 5: *Two PQ strips meeting at a discrete curved fold. For a given PQ strip D_1, the discrete ruling of the adjacent strip D_2 in a point $\mathbf{p}_i$ lies on a quadratic cone Δ_i.*

As a discrete model M of a general developable surface we use a quad-dominant mesh with planar faces, where the sum of inner angles at each vertex is equal to 2π. This means that we have a bijective mapping to a planar mesh of the same combinatorics such that corresponding faces are isometric.

Suppose a curved fold appears as a common polygon P of two PQ strips D_1, D_2 on the model M. Given the strip D_1 on one side of

P, we ask about the degrees of freedom in choosing the adjacent strip D_2. Clearly, we have to choose the discrete rulings in D_2 so that each vertex $\mathbf{p}_i$ P exhibits the angle sum 2π. Let $\mathbf{e}_- := (\mathbf{p}_{i-1} - \mathbf{p}_i)/\ \mathbf{p}_{i-1} - \mathbf{p}_i\ $ and $\mathbf{e}_+ := (\mathbf{p}_{i+1} - \mathbf{p}_i)/\ \mathbf{p}_{i+1} - \mathbf{p}_i\ $. Let the angle sum of D_1 at $\mathbf{p}_i$ be $\gamma_i = \alpha_i + \beta_i$ (see Figure 5). Then, the discrete ruling vector $\mathbf{x}$ on D_2 must form the angle sum $\sphericalangle(\mathbf{e}_-, \mathbf{x}) + \sphericalangle(\mathbf{x}, \mathbf{e}_+) = 2\pi - \gamma_i$. With $c := \cos\gamma_i$, this reads:

$$(c^2 - 1)\mathbf{x}^2 + (\mathbf{x}\cdot\mathbf{e}_-)^2 + (\mathbf{x}\cdot\mathbf{e}_+)^2 - 2c(\mathbf{x}\cdot\mathbf{e}_-)(\mathbf{x}\cdot\mathbf{e}_+) = 0. \quad (4)$$

Hence, the rulings have to lie on a quadratic cone Δ_i. Note that the ruling of D_1 also lies on this cone since its straight extension satisfies our requirement, though it does not describe a surface with a curved fold but a smooth extension. To obtain D_2, we may take one ruling on the cone Δ_i and compute further rulings at vertices $\mathbf{p}_j$ by keeping planarity of consecutive rulings. However, most of these solutions will not be suitable since they do not discretize a smooth surface. One has to take rulings which yield 'small' directional changes when passing from one cone to the next one. This necessitates an optimization approach as described next.

3 The basic optimization algorithm

The basic optimization algorithm *simultaneously* optimizes a discrete developable surface M and its planar development P. To maintain isometry between corresponding faces of M and P, we originally let M be a quad-dominant soup of planar polygons M^i in space. These polygons are isometric to the corresponding faces P^i in the planar mesh P, see Figures 6 and 7. During the optimization, the polygon soup M will become a mesh via a registration procedure which bears some similarity to that used in the PRIMO mesh deformation tool [Botsch et al. 2006]. However, our optimization requires more sophistication since we have to simultaneously optimize the development P while satisfying various other constraints.

Optimization starts with an initial set of pairs (M^i, P^i) of isometric planar polygons (primarily quads in our setting). The faces P^i form a planar mesh P, while in space the corresponding polygons M^i are assumed to roughly represent a developable shape D. They are not yet precisely aligned along edges. Thus M is not a mesh but a polygon soup. Later, in Sections 4 and 5, we describe how to compute initial positions P^i for different applications.

The unknowns. We introduce a Cartesian coordinate system in the plane of P, with origin $\mathbf{o}$ and basis vectors $\mathbf{e}_1, \mathbf{e}_2$. Each face P^i of P is congruent to the respective face M^i in space. For each such face, the image of $(\mathbf{o}; \mathbf{e}_1, \mathbf{e}_2)$ under the isometric transformation P^i M^i is a Cartesian frame $(\mathbf{o}^i, \mathbf{e}_1^i, \mathbf{e}_2^i)$ in the plane of the face M^i. If (p_x, p_y) are the coordinates of a vertex $\mathbf{p}$ of P^i, then the corresponding vertex $\mathbf{m}$ of M^i is $\mathbf{m} = \mathbf{o}^i + p_x\mathbf{e}_1^i + p_y\mathbf{e}_2^i$. During the optimization, the frames $(\mathbf{o}^i, \mathbf{e}_1^i, \mathbf{e}_2^i)$ undergo a spatial motion, and the coordinates (p_x, p_y) can also vary since we allow the polygons P^i to change.

We linearize the spatial motion of any face M^i using an instantaneous velocity vector field: The velocity of a point $\mathbf{x}$ can be represented as $\mathbf{v}(\mathbf{x}) := \bar{\mathbf{c}}^i + \mathbf{c}^i \times \mathbf{x}$, where $\bar{\mathbf{c}}^i, \mathbf{c}^i$ are vectors in 3-space.

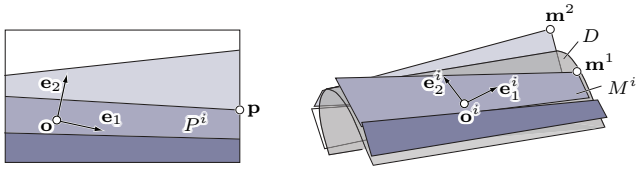

Figure 6: *Basic setup for the optimization when a reference surface D is used. Faces with the same color are congruent.*

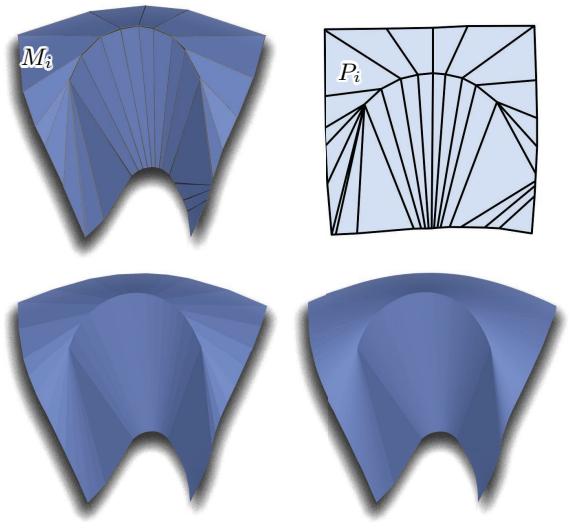

Figure 7: *Top left: Initial polygon soup M. Top right: Development P. Bottom left: M after subdivision and optimization. Bottom right: M after three rounds of subdivision and optimization.*

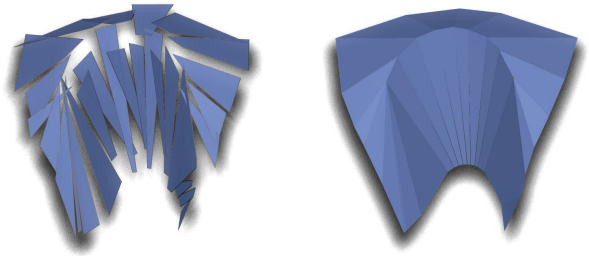

Figure 8: *Stability of the proposed optimization strategy. After artificial perturbation of faces (left),* 10 *rounds of optimization yield an almost aligned polygon soup (right). The stability of the procedure allows us to use rough estimates of ruling directions and planar development to initialize the algorithm*

Thus a vertex $\mathbf{m}_+$ of the displaced quad face is given by:

$$\mathbf{m}_+ = \mathbf{m} + \bar{\mathbf{c}}^i + \mathbf{c}^i \times \mathbf{o}^i + p_x(\mathbf{c}^i \times \mathbf{e}_1^i) + p_y(\mathbf{c}^i \times \mathbf{e}_2^i).$$

The new vertex position is linear in the unknown parameters $\bar{\mathbf{c}}^i, \mathbf{c}^i$ $\mathbb{R}^3$ of the velocity field, and also linear in the unknown coordinates p_x, p_y. We optimize over *both* the velocity parameters and the coordinates. The products $p_x\mathbf{c}^i$ and $p_y\mathbf{c}^i$ result in nonlinear terms if we insist on simultaneously optimizing them. To avoid nonlinear optimization, we alternately optimize for displacements $\bar{\mathbf{c}}^i, \mathbf{c}^i$ and for vertex coordinates p_x, p_y. Since our objective function is quadratic in both types of unknowns this amounts to alternately solving two sparse systems of linear equations.

Applying displacements corresponding to $\mathbf{c}, \bar{\mathbf{c}}$ destroys the exact isometric relation between corresponding faces P_i and M_i. It is therefore necessary to further modify the vertices of M^i. This can either be done by rigid registration of the face P^i to the estimated vertex locations $\mathbf{m}_+^j$ as proposed by Botsch et al. [2006], or by using a helical motion as described in [Pottmann et al. 2006] – we use the former approach.

The objective function. Our objective function is designed to simultaneously ensure that M becomes a mesh, fits the input data, and satisfies the aesthetic requirements of the application.

If a vertex $\mathbf{p}$ in the planar mesh P is shared by k faces, then $\mathbf{p}$ corresponds to k different vertices $\mathbf{m}^1, \ldots, \mathbf{m}^k$ of the corresponding k faces in M. Since these vertices should agree in the final mesh, we use a *vertex agreement term* of the form:

$$F_{vert} := \sum (\mathbf{m}^i_+ - \mathbf{m}^j_+)^2,$$

where the sum extends over all $\binom{k}{2}$ combinations per vertex $\mathbf{p} \quad P$, and over all vertices in P.

For M to approximate an underlying data surface D, we include a *fitting term* F_{fit} which is quadratic in the vertex coordinates $\mathbf{m}$. Let $\mathbf{m}_c$ denote the closest point in D to $\mathbf{m}$, and let $\mathbf{n}_c$ denote the unit normal at $\mathbf{m}_c$ to the underlying surface. We use a linear combination of the squared distance $(\mathbf{m}-\mathbf{m}_c)^2$ and the squared distance to the tangent plane $[(\mathbf{m}-\mathbf{m}_c)\cdot\mathbf{n}_c]^2$ as the data fitting term. When fitting curves, especially near boundaries, we use tangent lines instead of tangent planes.

Finally, we need a *fairness term* F_{fair}. For each pair of adjacent quads M^i and M^j of the PQ strip, we use the discrete bending energy of the corresponding developable surface $w_{ij}(\mathbf{n}^i_+ - \mathbf{n}^j_+)^2$, as given by Equations (2) and (3), as the fairness term. The normal of a quad M^i of M is given by $\mathbf{n}^i = \mathbf{e}^i_1 \times \mathbf{e}^i_2$. Under small displacements, this normal linearly varies as $\mathbf{n}^i_+ = \mathbf{n}^i + \mathbf{c}^i \times \mathbf{n}^i$. Given a polyline $(\mathbf{p}_1, \ldots \mathbf{p}_n)$ representing a fold line, i.e., a crease or a segment of a boundary curve, the contribution to F_{fair} is a sum of squared second differences $\sum(\mathbf{p}_{i-1}-2\mathbf{p}_i+\mathbf{p}_{i+1})^2$. Fairness terms are also applied to the respective polylines in the planar domain P.

The fairness term F_{fair} alone is not always sufficient to maintain convex quads, and to prevent flips in the planar mesh P, especially when the quads become thin after several steps of subdivision. Hence we add another term F_{conv} to enforce convexity. We assume that the orientation of each face of P coincides with the orientation of the plane induced by the frame $(\mathbf{o}; \mathbf{e}_1, \mathbf{e}_2)$. A corner $(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})$ of a planar polygon is convex if and only if the oriented area of the triangle $\Delta(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})$ is positive. This term also prevents flipping of faces.

The algorithm. Combining all individual terms, our basic optimization problem reads,

$$\begin{aligned} &\text{minimize} \quad F = F_{vert} + \lambda F_{fit} + \mu F_{fair} \\ &\text{subject to} \quad F_{conv} \geq 0. \end{aligned} \tag{5}$$

We alternately minimize the objective function over new positions of vertices in P, and displacements of faces in space, i.e., velocity vectors for the corresponding face planes. Observe that the weights w_{ij} of F_{fair}, which only depend on the planar mesh P, remain fixed when optimizing for displacements of faces in space and the side condition F_{conv} is also not needed. Hence, the spatial sub-problem amounts to solving a sparse linear system, and subsequent application of the corresponding rigid body motion per face. Optimizing the development P is more involved since the weights w_{ij} change in a non linear way as the geometry of P changes. Additionally we have a quadratic term F_{conv} to maintain convexity as a side constraint. With the meshes scaled to fit inside a unit cube, we found $\lambda = 1$ and $\mu = 10^{-4}$ to be good values to start the optimization.

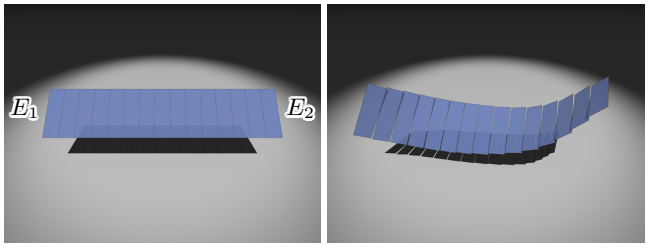

Figure 9: *Basic setup for bending energy minimization. We start with a regular grid (left). After prescribing point locations and tangent planes at the boundary the basic optimization is applied. (Right) The result after one round of optimization.*

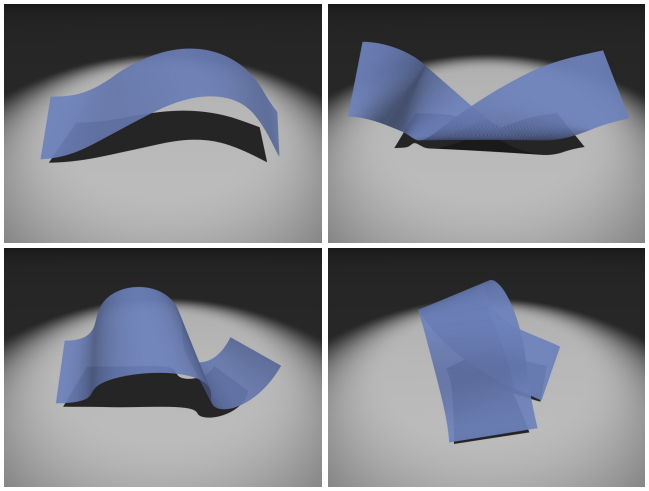

Figure 10: *Results of bending energy minimization for different boundary conditions. Given user constraints, the final models are obtained by alternately optimizing and subdividing.*

Given an initial mesh P and a polygon soup M that roughly approximates a developable shape, we alternately optimize for P and M. The optimization terminates when the vertex agreement term falls below a given threshold. For the next refinement level, we subdivide the current mesh P, and map the new faces to space using the rigid transformation associated with the faces of P at the current level. The refinement process splits each quad of P to form two new ones. Splitting is performed along the edges that do not correspond to ruling directions (see Figure 3, right). The process is repeated until desired accuracy is reached.

4 Applications to surface design

In this section we employ the basic optimization algorithm to the design of objects with curved folds.

Developable surfaces with minimal bending energy. As a simple application of our framework, without any relation to curved folding yet, we allow the user to take a planar strip of paper and attach it to some points and/or lines in space. The resulting shape is computed using a bending energy minimization, as popularly done for spline curves and double curved surfaces. Our approach extends the paper modeling tool of Bo and Wang [2007].

Since ruling directions are unknown, we initialize optimization from a soup of congruent quadrilaterals as shown in Figure 9. The user can prescribe new locations for the boundary edges E_1 and E_2 as well as the tangent planes at these edges, i.e., the planes of the outermost quads. We obtain the resulting shape by minimizing

$$F = F_{vert} + \mu F_{fair}. \tag{6}$$

Figure 10 shows several results obtained using our modeling tool. In all cases, the final maximal vertex disagreement is lower than 10^{-4} (with the models scaled to fit a unit box).

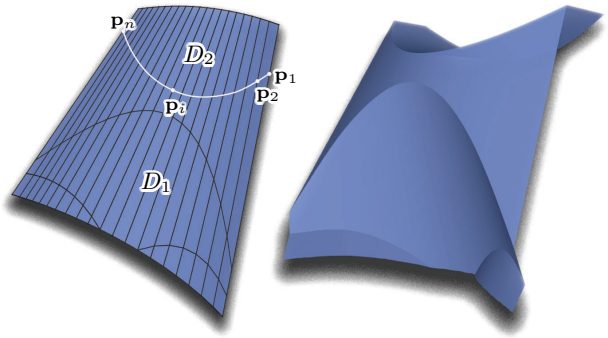

Figure 11: *Modeling curved creases. (Left) Crease curves are specified by the user on a developable surface. (Right) The resulting shape obtained with the curved folds along the prescribed curves.*

Bending in the presence of a curved fold. If a smooth developable surface along with the location of a future fold curve on it is specified, the shape of the folded developable is *uniquely* determined (up to those parts whose rulings do not intersect the fold curve). This is because tangent planes on the two sides of the fold are bisected by the osculating planes of the curve (cf. Section 2). However, no such uniqueness property exists for the discrete case (see Figure 11). By marking the location of a fold on a PQ strip with new vertices $\mathbf{p}_1, \mathbf{p}_2, \ldots$ on the edges, we segment the original strip into two strips, D_1 and D_2. There are, in theory, many possible strips D_2^* such that D_1, D_2^* form a curved fold. We use our optimization framework to filter out a *good* solution. Moreover, minimization of bending energy and fairness terms allows us to also compute parts of the surface whose rulings do not intersect the fold curve.

For each marked vertex $\mathbf{p}_i$, we approximate the discrete osculating plane of the polyline $\mathbf{p}_1, \mathbf{p}_2, \ldots$ by the plane spanned by the edges $\mathbf{p}_{i+1}\mathbf{p}_i$ and $\mathbf{p}_i\mathbf{p}_{i-1}$ (see Figure 5). We also attach an osculating plane to each edge, namely the bisector of the osculating plane at its end points. In order to construct the face of D_2^* adjacent to the edge $\mathbf{p}_i\mathbf{p}_{i+1}$, the plane of corresponding face in D_1 is reflected about the osculating plane associated with that edge. By intersecting neighboring mirrored planes we estimate the rulings of D_2^*. To ensure a vertex angle sum of 2π, we project these estimated rulings onto their respective cones Δ_i, given by Equation (4). From these projected rulings, we generate a mesh, which may contain non-planar faces at this stage. This mesh is then used to initialize our optimization, and also as the reference surface for the term F_{fit}. A typical modeling result obtained using this process is shown in Figure 11.

5 Approximation algorithm

Designing an object with curved folds when starting from scratch is not easy. Such a task can be daunting even for experienced users, specially in presence of multiple curved folds. However, it is much easier and intuitive for a designer to build a rough shape using paper or similar materials. The model can then be scanned and approximated using our approach. Subsequently, the user can edit or tweak the digital model using the proposed deformation tools (see Figure 14). During this process we also obtain a precise segmentation of the model into planes, cylinders, cones, and tangent surfaces (see e.g. Figure 2). Such a classification is useful for NURBS fitting and manufacturing. Therefore we address the following problem: Given scanned data D representing an almost developable surface, fit the data with an *exactly* developable surface which may exhibit (multiple) curved folds.

In order to initialize the optimization framework described in Section 3, we require the following: (1) A planar development P of the input data D, (2) estimates of the ruling directions on D, (3) a quad-dominant decomposition of P and (4) a corresponding polygon soup M which lies close to D in space.

Planar development of D. Using the constrained deformation tool by Kilian et al. [2007], we derive an as-isometric-as-possible mapping κ between the data mesh D and a plane, thus obtaining an approximate development of D. This general tool handles near-isometric deformations under constraints – in our case, the constraint is that the image points must have zero z coordinate. Such a procedure, unlike parameterization approaches [Sheffer et al. 2006], provides us with a sequence of intermediate meshes between D and its planar development. This additional information is useful for tracking persistent ridge lines or curves during unfolding which are used to initialize curved fold locations.

Estimating ruling directions on D. We first estimate approximate ruling directions on the given data mesh D as follows:

Stage A: At each vertex $\mathbf{p}$, we compute a *geodesic circle* $\mathcal{G}_\mathbf{p}$ as the set of points which are at constant geodesic distance $r_\mathbf{p}$ from $\mathbf{p}$. The radius $r_\mathbf{p}$ is chosen as 0.8 times the minimum distance from $\mathbf{p}$ to the mesh boundaries and all feature lines. We use ridge lines [Ohtake et al. 2004] as initial guess for curved folds, and mark them as feature lines. Points with radii smaller than a threshold are ignored. We compute a score for points $\mathbf{q}$ $\mathcal{G}_\mathbf{p}$, as:

$$\sigma(\mathbf{q}) := \mathbf{n_p} \cdot \mathbf{n_q} + \nu \ \ \mathbf{p} - \mathbf{q} \ \ / r_\mathbf{p},$$

where ν $[0, 1]$ and $\mathbf{n_p}, \mathbf{n_q}$ denote the unit normal vectors at $\mathbf{p}, \mathbf{q}$, respectively. In our experiments we use $\nu = 0.1$. Typically there are two strong maxima along diametrically opposite points on the geodesic circle; these points lie on a ruling. However, in nearly planar regions, the variation in the value of $\sigma(\mathbf{q})$ being small, we cannot detect a stable ruling direction. We explicitly mark such regions as planar (see Figure 12). Later we refine the boundaries of such planar regions using neighboring ruling information.

Stage B: In this step we extend the rulings. We use the following fact: For a torsal ruled surface, the surface normal remains constant along each ruling. Hence we extend the estimated ruling through a

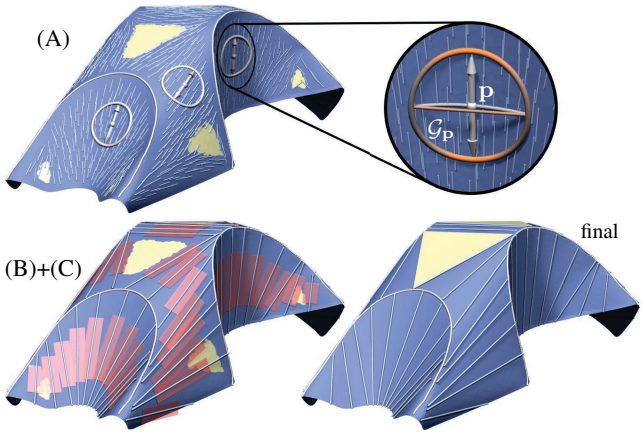

Figure 12: *Estimating ruling directions. In stage A, we guess ruling directions using geodesic circles, and also identify candidate planar regions. In stages B and C, the initial guesses at rulings are extended, and the entire ruling collection is thinned out. Bottom right: The final estimated rulings and the regions which have been established as planar (in yellow).*

point $\mathbf{p}$ until the surface normals in the end points deviate from the normal $\mathbf{n_p}$ in $\mathbf{p}$ more than a pre-defined threshold. Rulings are also terminated if they come close to feature lines or boundaries. For purposes of later pruning, we assign the negative mean deviation of surface normals along the ruling from $\mathbf{n_p}$ as a measure of quality to each extended ruling.

Stage C: The set of rulings obtained so far is thinned out while retaining rulings with high quality measure. We use a greedy approach: The ruling with highest quality measure is retained, and the ones intersecting a narrow band around it are removed. Here it is important to find the right measure of proximity of rulings, because the surface may exhibit conical parts where rulings intersect at a common vertex. Thus our band is shorter than the ruling and centered in its midpoint (in Figure 12 (B+C) these bands are marked in red and slightly widened for better visibility). All other rulings which intersect this band are considered 'close' and are removed.

We continue the process of pruning until we get a set of (roughly) evenly spaced rulings on the surface. Regions marked as planar are confirmed to be planar if they are bounded by three or more rulings or boundary edges.

Quad dominant decomposition of P. After estimating and pruning the rulings, we now deal with initializing the planar development P of D.

We use the development mapping κ to map the estimated ruling directions from the surface D to the plane. From this set of mapped rulings, we generate a coarse quad-dominant mesh P. Note that here a correct connectivity is much more important than the actual coordinates of the vertices. Subsequent optimization retains the connectivity of the initial mesh while updating the vertex positions.

The available input data for mesh generation are the estimated rulings mapped to the plane, the boundary of the planar mesh $\kappa(D)$, and the location of ridge lines in the original surface, which are used as candidate curved fold locations (see Figure 13, left). First, all end points of rulings are snapped to the closest boundary or ridge line — or, if the latter are too far away, are clustered in a point. Additionally short ridge lines are contracted to a single cone point (see Figure 13, center). Depending on the snapping target, we roughly classify an endpoint as *boundary point*, *fold point*, or *cone point*, respectively. In our examples, we frequently encountered combinations of two of these (e.g. a curved fold might extend to the boundary). The extension of rulings to boundary and ridge lines might introduce intersections close to ruling end points. Such intersections are resolved by swapping the corresponding end point vertex coordinates. We get a preliminary mesh by connecting ruling end points as they are traversed along the boundary and ridge lines, generating mostly long quadrilateral faces.

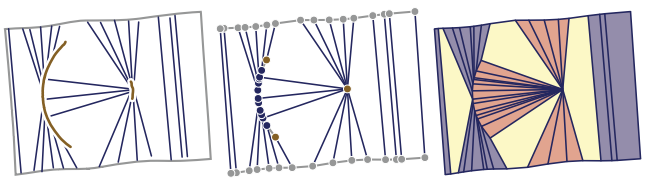

Figure 13: *Initial mesh layout. Left: A given collection of rulings (blue), ridge lines (brown), and mesh boundaries (gray). Center: Ruling endpoints are snapped and classified as boundary points (gray), fold points (blue) and cone points (brown). Right: By inserting and deleting rulings, a valid mesh connectivity without T-junctions is obtained. The planar parts of the original shape are marked in yellow.*

The resulting mesh is next modified by deleting or inserting rulings based on the following observations: (i) From any point on a fold, two rulings must emanate to prevent any T-junctions on the fold. (ii) Planar regions must be bounded either by rulings or a boundary curve. (iii) Boundary corner points should be included to preserve the shape of the base mesh. (iv) Faces adjacent to cone points might have more than four vertices. To ensure an optimal approximation of these regions, such faces need to be split into triangles or quads for our subdivision stage to apply. If a face holds more than a single cone point and the connecting lines lie entirely in the face, rulings are inserted connecting the cone points. If necessary, the faces originating from this step are further split by inserting rulings emanating from cone points. Finally, we obtain a quad dominant planar mesh P (see Figure 13, right).

Initialization of the polygon soup M. Initialization of our optimization procedure is complete when a polygon soup M, corresponding to the development P and close to the original shape D, is found. We find a face M^i of M corresponding to a face P^i of P by applying κ^{-1} to the vertices of P^i. Since the resulting vertices, in general, do not form a planar polygon which is isometric to P^i, we register a copy of P^i to these mapped vertices to initialize M^i.

Now we can apply the optimization alogrithm of Section 3 and obtain a mesh which approximates the given data D and has the optimized version of P as its precise development. In order to efficiently achieve high approximation quality, we start with a coarse approximation which is subsequently refined (by splitting quads in ruling direction) and optimized again. Results are shown in Figures 1, 2, 7, 14 and 16.

6 Further applications and discussion

As illustrated by Figure 14, surface reconstruction can nicely be combined with deformation tools such as [Kilian et al. 2007]. We first compute a digital reconstruction of a physical model and then vary its shape by an as-isometric-as possible deformation. The deformation will introduce deviations from a true developable surface, but it turns out that our reconstruction works very well on such deformed data sets. Note that even precisely isometric deformations in general do not preserve rulings and therefore rulings have to be re-estimated. In the example of Figure 14 it turned out that the optimization worked well with the initialization for the reconstruction of the physical model. In other cases, one may have to re-initialize for intermediate positions in a deformation sequence.

We emphasize here that the design and reconstruction of objects with curved folds is *not* simply solvable by a parameterization method. Parameterization will not yield any information about the precise location of folds, rulings and types of ruled patches, nor will it modify a data set to become precisely developable.

The nature of curved folds. Digital reconstruction of physical paper models yields a segmentation into torsal ruled patches. This provides insight into the typical behavior of a developable surface near curved folds. Some frequently occurring situations are depicted in Figure 2. In this way, our work can further contribute both to the theory of curved folding and to applications, e.g., to the development of interactive CAD tools for modeling objects with curved folds.

Architectural freeform structures. Developable surfaces are prominently visible in architectural design [Shelden 2002; Glaeser and Gruber 2007; Pottmann et al. 2007]. In particular, Frank O. Gehry has been using these surfaces quite extensively. The presence of rulings simplifies the actual construction. Panelization, e.g.

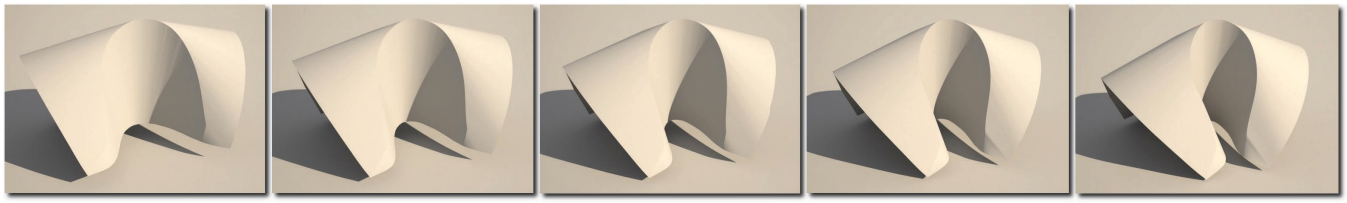

Figure 14: *A bending sequence which exhibits a curved fold. The left hand mesh is the result of approximating a 3D scan of a paper model. The other shapes have been computed by combining our reconstruction algorithm with an as-isometric-as possible shape modification of the reference surface, i.e., the reference data set in F_{fit} has changed, but the remaining data for optimization are taken from the left hand mesh.*

by metal tiles, is easy due to developability. The aesthetic continuation of a tiling over a general sharp edge is a difficult problem. However, at a curved fold the tile continuation is optimal (cf. Figures 1 and 15) and the design of the tiling can be done in the development. Note that our segmentation into torsal ruled patches is of high importance for manufacturing such architectural structures.

Industrial design. The shapes shown in our paper hopefully provide a first impression of the wide applicability of curved folding in industrial design. Such applications may require a high quality NURBS representation which is very easy to compute from our segmentation into torsal ruled patches. The exact locations of rulings (which cannot be seen in the triangle mesh of a 3D scan of a physical model) are important for manufacturing as well. Moreover, due to the fairness measures in our optimization framework we obtain aesthetically pleasing digital models while maintaining the hard constraint of a precise planar development.

Limitations. We performed a large number of experiments on data sets obtained by scanning models built from fabric or materials with a similar stretching behavior. It turned out that these models hardly behave like developable surfaces, particularly in regions with drastic folds. Hence, we must leave the task of (roughly) approximating such data by a single developable surface with curved folds to future research. The fully automatic generation of the initial planar mesh P worked well for all considered models. The only exception was the car model, a significantly more complex model, where we interactively modified a few ruling directions to ensure a suitable mesh for the adaptive subdivision we employ.

Implementation and run times. In our current implementation we use CHOLMOD [Davis and Hage 2001] to solve a sparse linear system and the KNITRO optimization package for constrained nonlinear optimization. Average runtimes for the models of Figure 16 are 160 seconds for ruling extraction (on 50K reference mesh), 20 seconds mesh layout and 140 seconds for optimization. The objective function was reduced to order of 10^{-4}. In particular the vertex agreement term is less than 10^{-4}. The fitting weight λ was reduced by a factor of 0.1 after each step of subdivision to favor fair solution surfaces instead of best approximating ones.

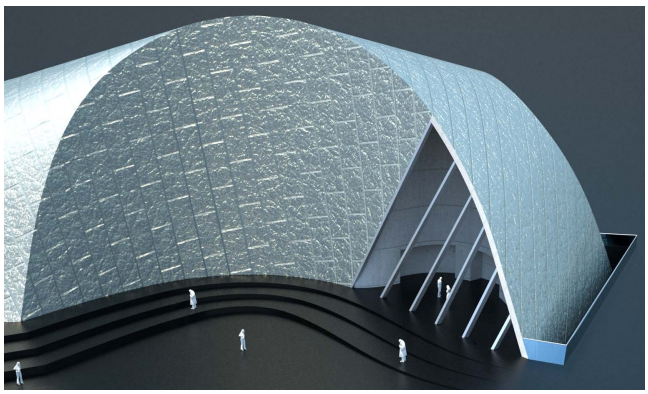

Figure 15: *Architectural design that features rulings as part of the support structure.*

Conclusion and Future research. We presented a computational framework for the design and digital reconstruction of developable surfaces with curved folds. Our work contributes to the discrete differential geometry of developable surfaces, to the discrete geometry of curved folds, and to the geometric optimization of surfaces with curved folds. Moreover, we illustrated the potential of our developments on a number of examples motivated by applications in architecture, industrial design and manufacturing. Given the limited amount of prior research in this area, there is still a lot of work to be done. Open problems include the reconstruction of models where a high approximation error has to be admitted such as scanned fabric, a careful analysis and classification of typical ruled patch arrangements at curved folds, and the development of novel interactive modeling tools for curved folding.

Acknowledgments. This work is supported by the Austrian Science Fund (FWF) under grants S92 and P18865. Niloy is also supported by a Microsoft outstanding young faculty fellowship. We are grateful to Heinz Schmiedhofer for his help with building and scanning the paper models and rendering the reconstructed models. We also thank Martin Peternell and Johannes Wallner for their thoughtful comments on the subject.

References

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

BO, P., AND WANG, W. 2007. Geodesic-controlled developable surfaces for modeling paper bending. *Comp. Graphics Forum 26*, 3, 365–374.

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

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. Primo: coupled prisms for intuitive surface modeling. In *Symp. Geom. Processing*, 11–20.

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

CERDA, E., MAHADEVAN, L., AND PASINI, J. M. 2004. The elements of draping. *Proc. Nat. Acad. Sciences 101*, 7, 1806–1810.

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

DAVIS, T. A., AND HAGE, W. W. 2001. Multiple-rank modifications of a sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications 22*, 4, 997–1013.

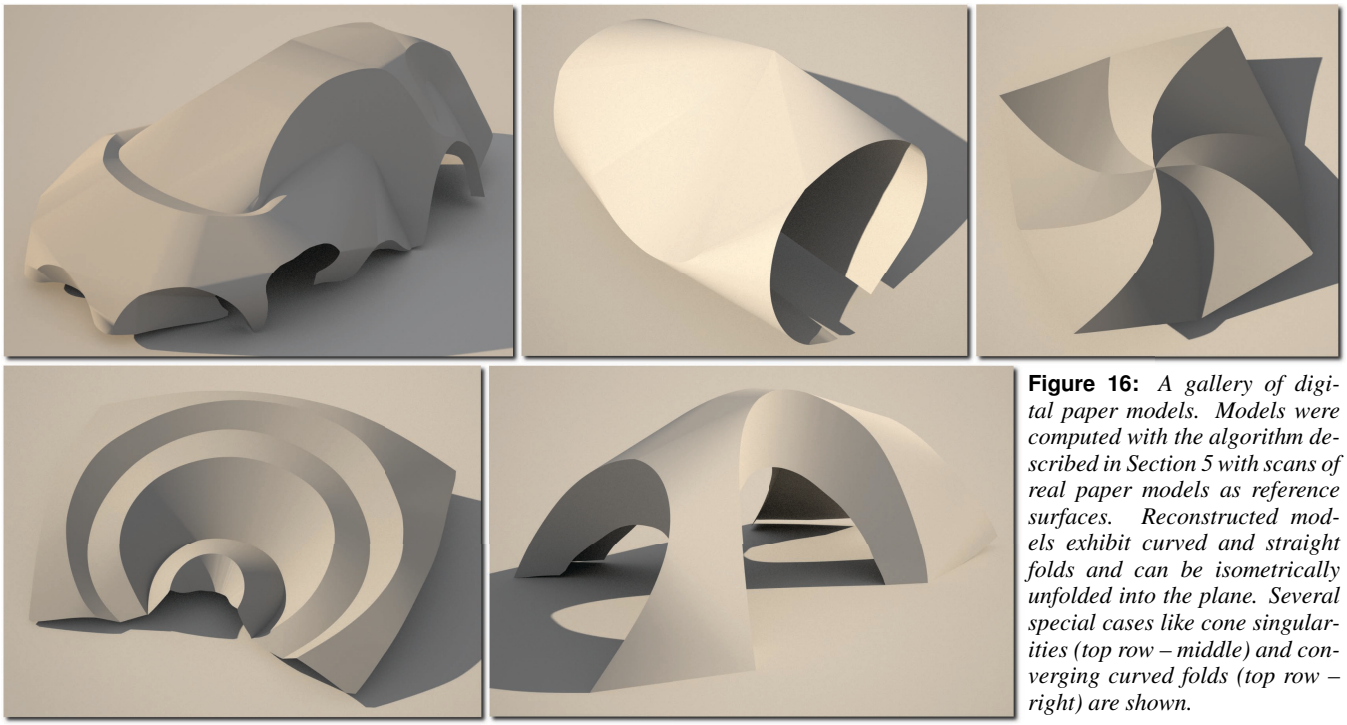

Figure 16: *A gallery of digital paper models. Models were computed with the algorithm described in Section 5 with scans of real paper models as reference surfaces. Reconstructed models exhibit curved and straight folds and can be isometrically unfolded into the plane. Several special cases like cone singularities (top row – middle) and converging curved folds (top row – right) are shown.*

DEMAINE, E., AND O'ROURKE, J. 2007. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge Univ. Press.

DESBRUN, M., POLTHIER, K., AND SCHRÖDER, P. 2005. *Discrete Differential Geometry*. Siggraph Course Notes.

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

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

GLAESER, G., AND GRUBER, F. 2007. Developable surfaces in contemporary architecture. *J. of Math. and the Arts 1*, 1–15.

HUFFMAN, D. A. 1976. Curvature and creases: a primer on paper. *IEEE Trans. Computers C-25*, 1010–1019.

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

KERGOSIEN, Y., GOTUDA, H., AND KUNII, T. 1994. Bending and creasing virtual paper. *IEEE Comp. Graph. Appl. 14*, 1, 40–48.

KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric modeling in shape space. *ACM Trans. Graphics 26*, 3, 64.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graphics 25*, 3, 681–689.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2006. Papercraft models using generalized cylinders. In *Pacific Graph.*, 148–157.

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

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. 23*, 3 (August), 609–612.

PÉREZ, F., AND SUÁREZ, J. A. 2007. Quasi-developable B-spline surfaces in ship hull design. *Comp.-Aided Design 39*, 853–862.

PETERNELL, M. 2004. Developable surface fitting to point clouds. *Comput. Aided Geom. Des. 21*, 8, 785–803.

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

POTTMANN, H., HUANG, Q.-X., YANG, Y.-L., AND HU, S.-M. 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Computer Vision 67*, 3, 277–296.

POTTMANN, H., ASPERL, A., HOFER, M., AND KILIAN, A. 2007. *Architectural Geometry*. Bentley Institute Press.

ROSE, K., SHEFFER, A., WITHER, J., CANI, M.-P., AND THIBERT, B. 2007. Developable surfaces from arbitrary sketched boundaries. In *Symp. Geometry Processing*. 163–172.

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

SHATZ, I., TAL, A., AND LEIFMAN, G. 2006. Papercraft models from meshes. *Vis. Computer 22*, 825–834.

SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis. 2*, 2, 105–171.

SHELDEN, D. 2002. *Digital surface representation and the constructibility of Gehry's architecture*. PhD thesis, M.I.T.

SUBAG, J., AND ELBER, G. 2006. Piecewise developable surface approximation of general NURBS surfaces with global error bounds. In *Proc. Geometric Modeling and Processing*. 143–156.

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, C. C. L. 2008. Towards flattenable mesh surfaces. *Comput. Aided Des. 40*, 1, 109–122.

WERTHEIM, M. 2004. Cones, Curves, Shells, Towers: He Made Paper Jump to Life. *The New York Times, June 22*.

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