Developable Surfaces with Curved Creases

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

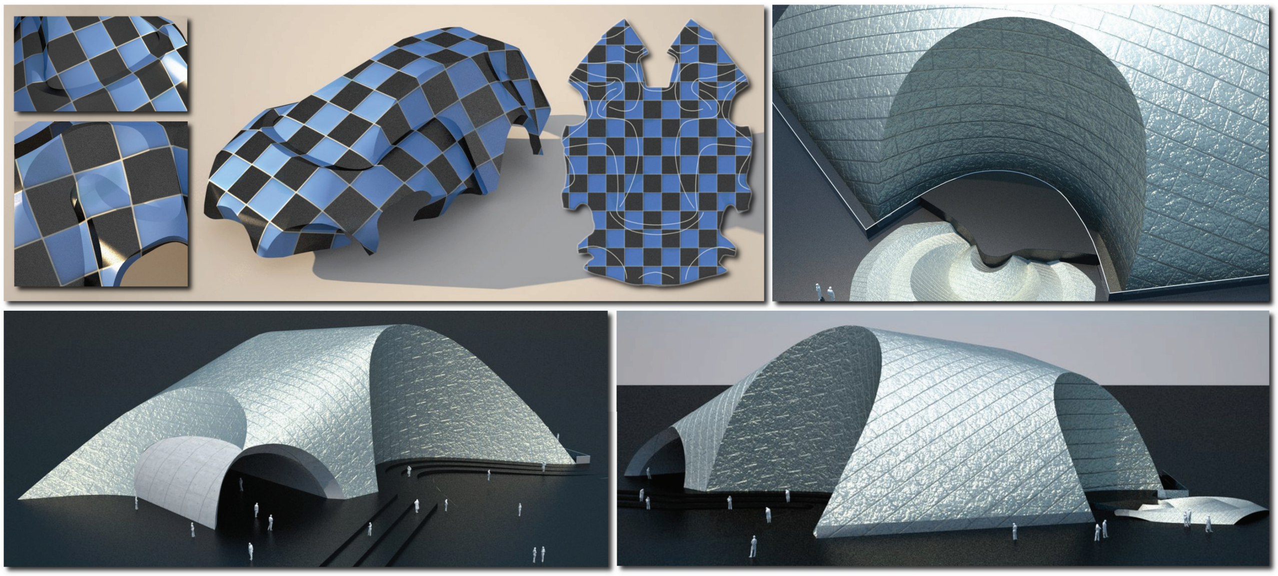

Figure 1: *Top left: Reconstruction of a car model based on a felt design by Gregory Epps. Close-ups of the hood and the rear wheelhouse are shown on the left. The fold lines are highlighted on the car's development. Top right and bottom: Architectural design. All shown surfaces can be isometrically 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.

Keywords: curved fold, developable surface, computational origami, architectural geometry, industrial design.

1 Introduction

This paper is an excerpt from [Kilian et al. 2008]. More details on curved folding can be found in the aforementioned paper.

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* (see Figure 2). 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.robofold.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.

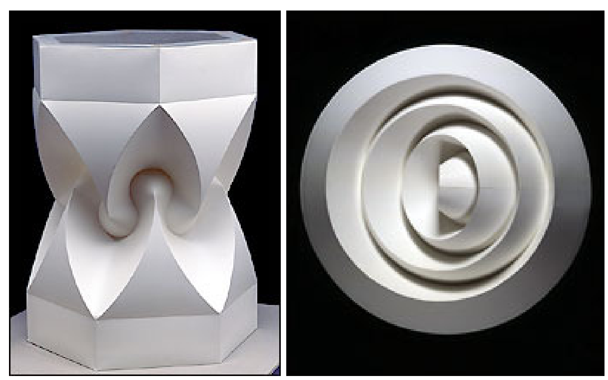

Figure 2: *Two examples of paper models featuring curved folds that were created by David Huffman.*

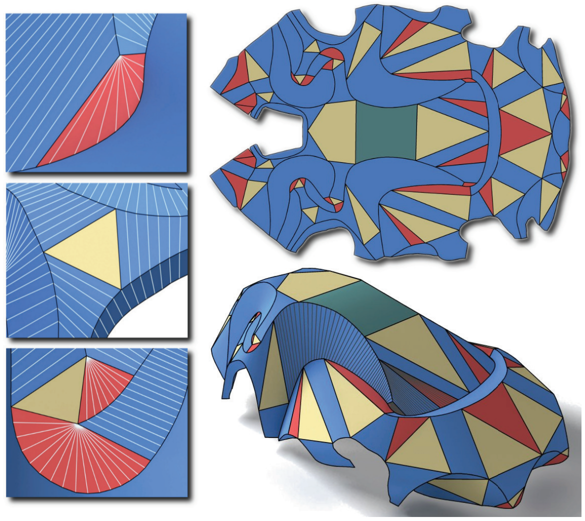

Figure 3: *The car model of Figure 1 and its development (top right). The patch decomposition into torsal ruled surfaces is shown using the following color scheme: planes are shown in yellow, cylinders in green, cones in red, and tangent surfaces in blue. Sample rulings are shown on some patches of the windshield and the side window. Such a segmentation is essential for NURBS surface fitting and manufacturing.*

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. 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 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 3).

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 4, 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 4, 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 in [Kilian et al. 2008].

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. This fact 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 5 and 6. 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.

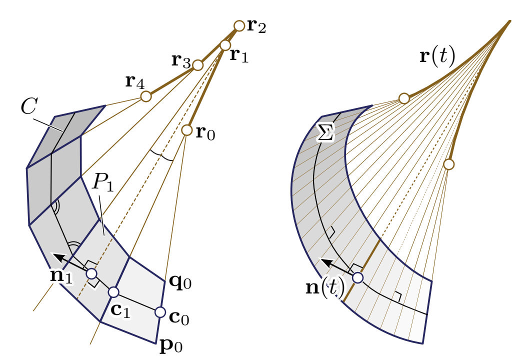

Figure 4: *A PQ strip (left) is a discrete model of a developable surface Σ (right). The intersections of edges $\mathbf{p}_i\mathbf{q}_i$ of adjacent planar quads generate the regression polyline $\mathbf{r}_i$. In the limit of a refinement process, this regression polyline becomes the regression curve $\mathbf{r}(t)$. 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 $\mathbf{n}_i$.*

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. See [Kilian et al. 2008] on 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 $P^i \mapsto M^i$ 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. 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 $\bar{\mathbf{c}}^i, \mathbf{c}^i \in \mathbb{R}^3$ 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:

$$F_{\text{vert}} := \sum(\mathbf{m}_+^i - \mathbf{m}_+^j)^2,$$

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

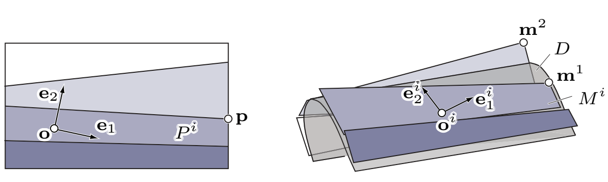

Figure 5: *Basic setup for the optimization when a reference surface D is used. Faces with the same color are congruent.*

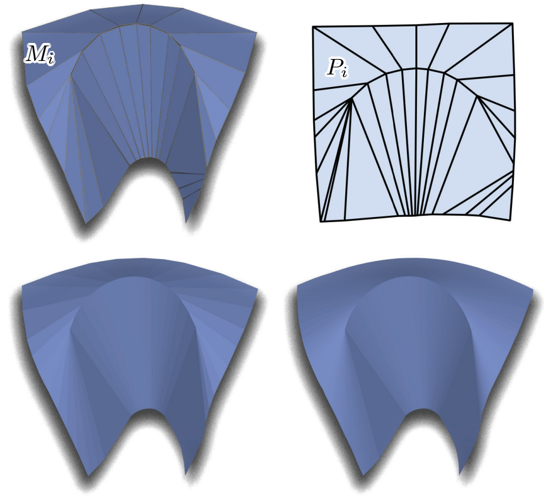

Figure 6: *Top left: Initial polygon soup M. Top right: Development P. Bottom left: M after subdivision and optimization. Bottom right: M after three rounds of subdivision and optimization.*

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 $w_{ij}(\mathbf{n}_+^i - \mathbf{n}_+^j)^2$ of the corresponding developable surface as described in [Kilian et al. 2008] as the fairness term. The normal of a quad M^i of M is given by $\mathbf{n}^i = \mathbf{e}_1^i \times \mathbf{e}_2^i$. 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

$$\begin{aligned} &\text{minimize} \quad F = F_{\text{vert}} + \lambda F_{\text{fit}} + \mu F_{\text{fair}} \\ &\text{subject to} \quad F_{\text{conv}} \geq 0. \end{aligned} \tag{1}$$

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. Note that the weights w_{ij} (see [Kilian et al. 2008]) of F_{fair}, which only depend on the planar mesh P, remain fixed when optimizing for displacements of faces

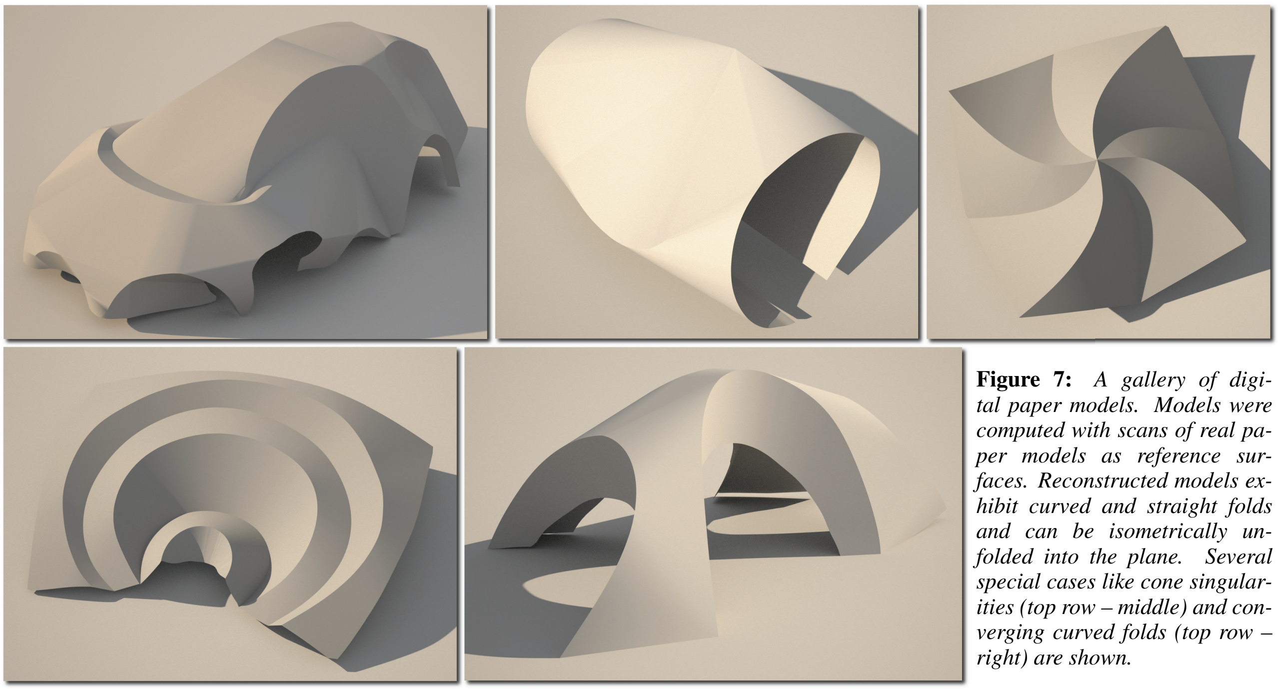

Figure 7: *A gallery of digital paper models. Models were computed with scans of real paper models as reference surfaces. Reconstructed models exhibit curved and straight folds and can be isometrically unfolded into the plane. Several special cases like cone singularities (top row – middle) and converging curved folds (top row – right) are shown.*

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.

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 4, right). The process is repeated until desired accuracy is reached.

Acknowledgements

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

References

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.

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

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

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

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. *ACM Transactions on Graphics 27*, 3. to appear.

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.

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.

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

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

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