Eurographics Symposium on Geometry Processing 2012
Eitan Grinspun and Niloy Mitra
(Guest Editors)

Volume 31 (2012), Number 5

Flexible Developable Surfaces

Justin Solomon[1] Etienne Vouga[2] Max Wardetzky[3] Eitan Grinspun[2]

[1]Department of Computer Science, Stanford University, USA
[2] Computer Science Department, Columbia University, USA
[3] Institute for Numerical and Applied Mathematics, Univerity of Göttingen, Germany

Abstract

We introduce a discrete paradigm for developable surface modeling. Unlike previous attempts at interactive developable surface modeling, our system is able to enforce exact *developability at every step, ensuring that users do not inadvertently suggest configurations that leave the manifold of admissible folds of a flat two-dimensional sheet. With methods for navigation of this highly nonlinear constraint space in place, we show how to formulate a discrete mean curvature bending energy measuring how far a given discrete developable surface is from being flat. This energy enables relaxation of user-generated configurations and suggests a straightforward subdivision scheme that produces admissible smoothed versions of bent regions of our discrete developable surfaces.*

1. Introduction

If a surface can be bent to lie flat on the plane without tearing or stretching, it is said to be *developable*. The study of developable surfaces goes hand in hand with the study of isometric deformations, since any developable surface can be created by isometrically bending a flat sheet. This constructive view explains why many manufactured objects, such as those made of sheet metal, paper, cardboard, and plywood, are made by attaching together developable pieces. In general, such constructions are well-approximated by developable surfaces that contain smooth regions possibly interrupted by sharp creases.

Deformation, discretization, and locking To *design* a developable surface easily, it can be helpful to deform an existing developable surface *smoothly and isometrically*, tracing a continuous trajectory strictly within the space of developable surfaces.

One of the critical challenges to such smooth isometric deformation is the *locking* induced by discretization of the smooth surface. Locking takes on many forms, and there are numerous definitions and discussions (see [CB98] and references therein). For our purposes, locking can be understood as the *failure of a discretization to navigate the full space of isometric deformations of a developable surface smoothly*. Discretizations that lock are typically forced to leave the admissible space of developable configurations, even for simple or common deformations. For example, a square sheet discretized by an irregular triangle mesh consisting of conforming, planar triangles in general cannot be bent (at all) without stretching some triangles.

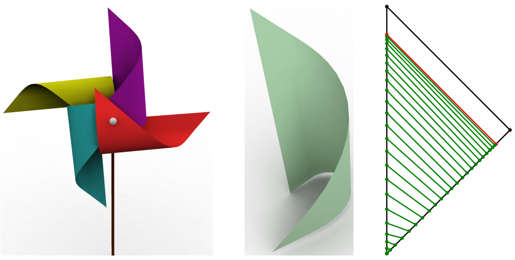

Figure 1: ***Developable pinwheel*** *modeled in our system (left). Each developable piece (middle), defined by its planar fold configuration (right), was modeled by prescribing the positions of the three corner vertices in three-space, constraining the left border to remain flat (*red ruling*), and allowing the developable surface to relax to the curved (*green rulings*) configuration that minimizes the nonlinear elastic thin plate energy.*

Desiderata As such, when the goal is to deform a surface smoothly and isometrically, the choice of discretization is paramount. We seek discretizations that are

- *geometrically exact*, in the sense that the discrete geometry is itself developable (without altering the definition of developability);

- *versatile*, in the sense that it can approximate *any* developable surface;
- *flexible*, in the sense that it is possible to navigate from any configuration to any other in the space of all valid discretizations smoothly, without leaving the space; and
- *as simple as possible*, so that efficient and robust numerics are easier to implement.

In focusing on geometrically-exact discretizations, we are in a sense seeking to build a discrete theory of developability from the ground up, rather than trying to approximate developable behavior on existing non-developable structures. In particular, we consider discrete developable surfaces represented by a planar domain decorated with embedded fold lines, bend angles, and rulings, similar in spirit to [KGK94, PB07]. We believe that this is a promising avenue to locking-free discretization of smoothly deforming developable surfaces, striking the right balance between simplicity and efficiency of numerical treatment, exact geometric flexibility, and versatility.

Mechanics After a suitable discretization is chosen, a natural step is to endow the discrete representation with an elastic potential; such potentials serve as intuitive metrics in the context of interactive modeling or as natural objective functions for offline optimization.

Physical realizations of developable surfaces are typically governed by the *thin plate* equations [CB98, AP09], which account for both the stretching and the bending of a thin (but finitely thick) sheet of *elastic* material. For sufficiently thin plates and appropriate boundary/loading conditions, the plate may be treated as *inextensible*, its shape governed by the stored bending energy and boundary conditions [CB98, AP09]. Identifying geometrically-exact flexible discretizations for developable surfaces and formulating a consistent discrete thin plate bending energy is an important step toward addressing the open problem of *locking* in numerical treatment of the thin plate equations [CB98, EB08].

Overview After surveying related work (§2), we recall the smooth theory of developable surfaces (§3) and develop a corresponding discrete theory (§4), which we endow with a consistent discrete bending energy (§5). We describe a numerical implementation in the context of deformation and interactive editing (§6), in which the surface can be manipulated, pinned, and elastically relaxed. While our representation is always discrete, subdivision (§6.2) in tandem with elastic relaxation allow arbitrarily fine approximation of smoothly curved developable surfaces with smoothly curved sharp creases.

2. Related Work

Previous works have considered developable surfaces as a geometric representations for design of plush toys [JKS05], ship hulls [PS07], clothing [KG90, EB08, TC09, CT10], folding maps in augmented reality [MS11], papercraft [Huf76, KGK94, MS04, STL06], and origami [BGW06, Tac09b, Tac09a, Cha09, SG11, Tac10a, IT10, Tac10b, Tac11, WC11].

Computational treatments of developable surfaces typically focus on synthesis, approximation, or deformation. The first goal seeks to *synthesize* a developable surface satisfying prescribed conditions, such as a given boundary contour [Fre02, Fre04, DJW*06, RSW*07], pair of opposing boundary curves [WF88, CWT08], prescribed geodesic [ZW08], or Gaussian image [Par02].

The second goal is to *approximate* given geometric data, such as a scattered point clouds [CLL*99, Pet04, PB07, TC09, MS11], sets of approximating planes [PW99], or a nearby non-developable surfaces; the latter may require segmenting the input surface into patches [Wan04, JKS05, STL06, Wan08, SKKO02] or strips [MS04, LLH09, PSB*08] and treating the case of curved creases [KFC*08].

The third goal is to *deform* a given developable surface, for example by following the deformation of its contour [RCHT11] or by constrained energy relaxation [KGK94, BGW06, LTJ07, EB08, CT10]. One of the critical challenges to smooth deformation is the *locking* induced by discretization of the smooth surface. The success of a discretization in problems of synthesis or approximation is in general *not* indicative of success in smooth isometric deformation.

Discretization of developable surfaces may be inexact, as in the case of *quasi*-developables [JKS05, PS07, Wan08, TC09] or non-conforming meshes [EB08], or *exact*, in the sense that the discrete representation is piecewise smooth and developable in the usual sense. The latter category includes *reduced* (or “minimal”) discretizations based on cone splines [LP98, CLL*99], generalized cones [Red89, SF96], curves in dual projective three-space [PW99], embedded geodesics [Pet04, BW07, ZW08], and characteristic points based on singularity theory [KG90]. Some theoretical work also has been carried out characterizing properties of exact folded developables constructed by gluing together planar polygonal pieces [BI08].

Exact discretizations can also be developed by augmenting more general discretizations with *developability constraints*, an approach considered for Bézier and B-spline surfaces [CC90, FB93, Pot95, CS02, Aum03, Aum04, FJ07], triangle meshes [WF88, Fre02, Fre04, Wan04, MS04, BGW06, RSW*07, LTJ07, LLH09, CT10, RCHT11, SKKO02], and planar quadrilateral meshes [WF88, LPW*06, KFC*08]. The majority of these discretizations have been demonstrated successfully in the context of synthesis and approximation, but their performance in problems of smooth deformation remains less understood.

Our investigation has been motivated by recent works on *interactive design* of developables and origami [SF96, BGW06, BW07, TI08, Ida08, IT10, Tac09b, Tac09a, Cha09, Tac10a, Tac11, WC11]. Indeed, our decisions concerning geometric discretization, numerical implementation, and interaction metaphors seek to prioritize the numerical robustness, economy of computation, and versatility required for interactive design.

3. Smooth Developable Surfaces

Developable surfaces have a variety of fascinating properties, see e.g. [Car76] for a general discussion; we sketch those properties here that we require in the discrete setting later.

By definition, developable surfaces are characterized by having vanishing Gauß curvature, that is, they are locally isometric to the Euclidean plane. Any smooth developable surface Σ locally is comprised of a one-parameter family of straight line segments called *rulings* that do not cross and run across the surface. Likewise, there exists a one-parameter family of smooth curves on Σ that run orthogonally to the rulings. Any curve γ in this family has the property that the orthonormal frame formed by (i) the unit tangent of γ, (ii) the (properly oriented) unit ruling direction, and (iii) the surface normal is a so-called natural (or Bishop) frame [Bis75]. This observation can be reversed: Consider a smooth space curve $\gamma(s)$ equipped with its orthonormal natural frame (T,U,V). Choosing $\varepsilon > 0$ small enough yields a developable surface

$$\Sigma(s,t) := \gamma(s) + tU(s)$$

for all $t \in (-\varepsilon, \varepsilon)$, with rulings formed by U and surface normals along γ formed by V.

In this case, the curvature of the resulting surface conveniently can be described in terms of the framed curve. Since it is perpendicular to the flat rulings, γ runs along a principal curvature line with normal curvature equal to $T' \cdot V$; this normal curvature is in turn equal to half the mean curvature $H = \kappa_1 + \kappa_2$ of Σ since the other principal curvature must be exactly zero. Likewise, for *fixed t* consider the curve $\gamma_t(s) = \Sigma(s,t)$. Re-parameterizing γ_t to arc-length, there is a natural frame for γ_t from which one can infer the mean curvature at $\Sigma(s,t)$ from the curvature of $\gamma_t(s)$. Although we do not explicitly derive the requisite expressions here, this discussion serves as a motivation for the case of discrete developable surfaces, where (discrete) mean curvatures can be inferred from polygonal analogs of γ.

4. Discrete Developable Surfaces

Informally, we think of discrete developable surfaces as pieces of origami that can be folded from a sheet of paper without tearing or gluing. While such a discrete structure is piecewise flat, we do not view our surface as a collection of planar segments bordered by sharp folds. Instead, in many practical situations, external forces and other spatial constraints naturally lead to curved configurations that should be represented in our setup. We account for this situation by approximating smoothly-curved developable pieces using discrete developable pieces that consist of a *finite* number of rulings.

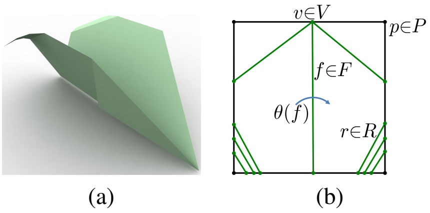

Figure 2: *(a) A simple developable surface with two-dimensional configuration (b).*

Formally, in our setup, a discrete developable surface is comprised of the following parts:

- A *polygonal domain* $P \subset \mathbb{R}^2$ with corners $p_1, \ldots, p_m$. We assume that the edges of P do not self-intersect. For notational convenience, we let P include its own boundary.
- A collection of vertices $V = \{v_1, \ldots, v_n\} \subset P$, which can reside both on ∂P and in int P.
- A collection of *folds* $F \subseteq V \times V$ representing line segments contained within P parameterized by their starting and ending vertices.
- A subset $R \subseteq F$ of *rulings* representing the boundaries of smoothly-bent pieces.
- A folding angle $\Theta : F \to \mathbb{R}$ associated with each fold; for a given fold $f = (v_i, v_j) \in F$ we denote its associated folding angle as θ_f.

Figure 2 labels all these components on a simple two-dimensional configuration.

Degrees of freedom (DOFs) For a given domain P, the configuration space of discrete developable surfaces is parameterized, modulo rigid motion, by the graph topology induced by F, the positions of the vertices V, and the folding angles Θ of the fold lines. We note that this set of DOFs is distinguished from those used in the origami literature (see [Tac09b] and references therein) in that we consider variable, rather than fixed, vertex positions.

To this essential set of variables, we add variables defining a set of rigid Euclidean transformations E mapping each face of P to its realization in $\mathbb{R}^3$. While these additional variables do not add expressive power to the discretization, they simplify the implementation of positional constraints and user manipulations in three-space.

Boundary conditions and constraints Fold lines and prescribed bending angles Θ are given as user input or through the subdivision schemes described in §6.2. A challenge is to ensure developability, which corresponds to a number of

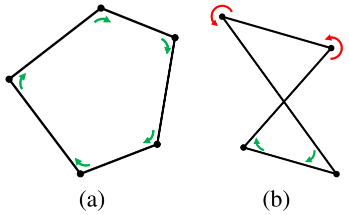

Figure 3: *Polygons satisfying (a) and violating (b) the interior angle constraint.*

constraints on V, F, and Θ that must be obeyed by admissible configurations:

4.1. Constraints

A number of constraints are needed to ensure that every configuration corresponds to a discrete developable surface with a realizable embedding in $\mathbb{R}^3$. These constraints can be divided into three groups:

- Geometric constraints on the vertices and edges of P: edges must intersect only at vertices, and vertices must remain within or on the boundary of the domain ∂P. (§4.2)
- Compatibility constraints imposed by 3D realizability: around every interior vertex, gluing together adjacent faces at their common edge with that edge's folding angle must yield a surface that locally "closes up." Moreover, the Euclidean transformations E must be compatible with the other degrees of freedom, so that neighboring faces meet seamlessly in 3D and at the correct folding angles. (§4.3)
- User-specified constraints on positions in 3D, positions of rulings, or folding angles. (§4.4)

4.2. Graph Constraints

On a developable surface, rulings intersect only at cone points; we thus require discrete folds to intersect each other only at (interior or boundary) vertices. Preventing the user from drawing folds that violate this constraint is straightforward, but preventing such intersections from occurring during optimization is much more difficult. Although non-intersection could be enforced with inequality constraints for each pair of folds, this approach requires up to $|F|^2$ nonlinear constraints for $|F|$ folds and quickly becomes prohibitively expensive.

We instead make two observations: first, we may assume, without loss of expressiveness, that the polygonal faces formed by cutting P along fold segments are convex. The folding angles of any folds around a concave interior vertex must be zero (it is impossible to fold along such lines without tearing the paper), so the surface is locally flat at that point. Second, given a graph in the plane with straight edges, whose faces are all convex and whose edges do not intersect, it is impossible to slide vertices around smoothly in a way that introduces an edge-edge intersection without causing one of the faces to become non-convex.

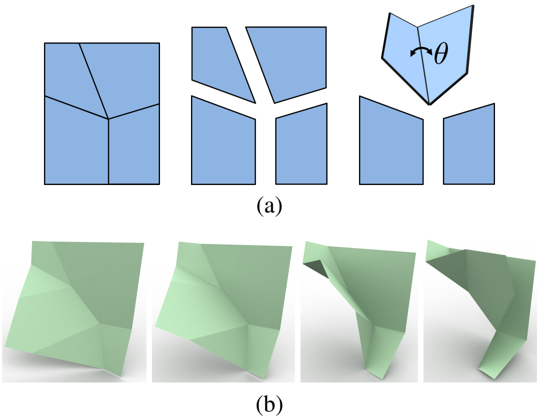

Figure 4: *(a) Assembling a developable surface; (b) multiple sets of compatible folding angles are possible with a fixed set of points in V and ruling topology.*

We thus achieve non-intersection of edges by instead enforcing the simpler condition that the *turning angle* between consecutive folds when traveling around each face of P lies in $[0,\pi]$. For polygons of winding number 1, this constraint is equivalent to convexity of the polygon. Polygons of higher winding number do not arise in practice, since it is impossible to arrive at such a confguration without violating the turning angle constraint of a neighboring face. See Figure 3 for examples of polygons satisfying and violating this constraint.

Since the boundary of a piece of paper does not grow or shrink during folding, we also cannot allow vertices on the boundary ∂P of the domain to leave ∂P. We enforce this condition by pinning the domain corners p_i, and restricting all other boundary vertices to slide along the lines between these corners.

4.3. Compatibility Constraints

As long as folds do not intersect or meet (and thus all vertices are on the boundary), any configuration is admissible. For interior vertices, however, only some choices of folding angles for the adjacent fold lines yield a locally realizable surface.

We can think of realizing a discrete developable surface as cutting P along its folds to form pieces (faces of P) to be assembled in $\mathbb{R}^3$ by gluing them together one at a time using angles in Θ, as illustrated in Figure 4(a). Often multiple such assemblies are possible with the same set of pieces, as in Figure 4(b). We must ensure that the pieces can be assembled in a way that does not tear the surface like the example in

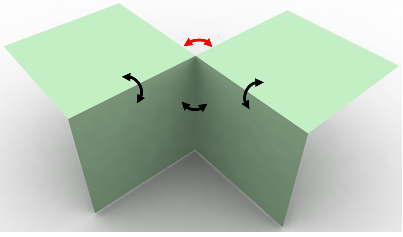

Figure 5: *Prescribing the black folding angles in a simple crossed fold configuration forces a tear at the red angle, which should represent a single ruling.*

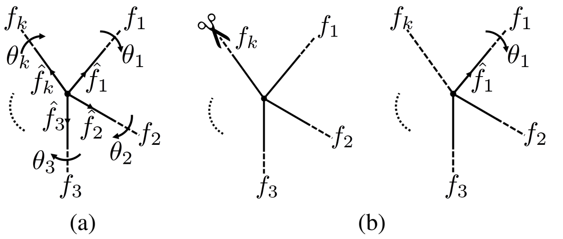

Figure 6: *(a) Notation for interior vertex compatibility; (b) construction for compatibility constraint.*

Figure 5. Unlike previous work on developable surface approximation using meshes, this constraint does *not* have to do with the requirement for vanishing Gaussian curvature, a property that follows automatically from our representation via rulings embedded on a flat sheet.

Consider the interior vertex $v \in V$ shown in Figure 6(a); without loss of generality we can assume that v is at position $(0,0)$. We draw unit vectors $\hat{f}_1,\ldots,\hat{f}_k$ in the directions of the outgoing folds $f_1,\ldots,f_k$ from v in clockwise order. Embed P in the xy plane of $\mathbb{R}^3$ and define M_i to be the rotation about $\hat{f}_i$ by angle θ_i.

Suppose we cut outward from v along $\hat{f}_k$. We hold the piece of P counterclockwise from f_k flat on the xy plane and fold along $\hat{f}_1$ by angle θ_1, then along $\hat{f}_2$ by angle θ_2, and so on as in Figure 6(b). The product $M_1M_2\cdots M_i$ represents the transformation from points in the segment clockwise from f_1 to f_i in the xy plane to its position in the three-dimensional configuration after i folds. Completing the loop, the product $M_1M_2\cdots M_k$ should put the segment clockwise from f_1 back in the xy plane, since we have completed the folding process and then rotated around the final fold using M_k. This process yields the compatibility condition $M_1M_2\cdots M_k = I_{3\times3}$.

We should mention here that constraints in this form are redundant. In particular, the product $\prod_i M_i$ is guaranteed to be a rotation regardless of whether the configuration is compatible. It is thus sufficient to constrain only the three upper-off-diagonal entries to be zero. Strictly speaking, however, this constraint is weaker since it does not distinguish between the identity and 180° rotations about the axes, but this dinstinction is not important in practice for a surface that starts in a realizable configuration and deforms smoothly.

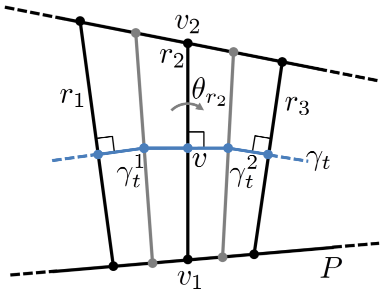

Figure 7: *Notation for the mean curvature bending energy.*

It is *possible* to construct a three-dimensional developable surface from any two-dimensional configuration satisfying all of the above constraints. To make it easier to realize the surface as it is being edited and to enforce user-supplied constraints on the 3D positions of parts of the surface, we introduce as auxiliary degrees of freedom Euclidean transformations E mapping faces to their positions and orientations in $\mathbb{R}^3$. We then require additional compatibility conditions ensuring that these transformations do indeed reconstruct a seamless surface.

To guarantee that the transformations E realize the developable surface, it is sufficient to build a minimum spanning tree of the dual graph of P and to enforce that two faces connected by an edge in the spanning tree abut in $\mathbb{R}^3$ with the correct folding angle; that other pairs of adjacent faces meet up correctly is already enforced by the interior vertex compatibility constraints. Thus, for each edge of our spanning tree we simply add constraints that endpoints of the adjacent pieces meet up and that the resulting fold is at the angle from Θ; the latter can be accomplished as a constraint on $\tan{}^{\theta_f}\!/_2$ analogous to that in [BWR*08].

4.4. Additional Constraints

We allow the user to specify additional constraints on the configuration: vertex positions in P, folding angles, and the 3D positions of the vertices on the realization can all be pinned.

5. Mean Curvature Bending Energy

While the constructions in Section 4 can be used to obtain admissible developable surfaces, in the end we wish to approximate smooth surfaces rather than ones obtained solely using oragami constructions. To this end, we introduce a squared mean curvature bending energy measuring a surface's deviation from flatness.

As in the smooth setting, we may infer curvature and thus bending energies using curves that run perpendicular to rulings. To this end, consider three rulings r_1, r_2, r_3 that border two neighboring planar pieces as in Figure 7. Assume the ends of r_2 are given by vertices v_1 and v_2. Then, for every point $v = (1-t)v_1 + tv_2$ on r_2, there exists a piecewise linear curve γ_t, comprised of three segments that are orthogonal to the rulings, whose vertices lie along the angle bisectors between r_1, r_2 and r_2, r_3. The bending energy induced by a folding angle θ_{r_2} about r_2 is associated with the part of the surface flanking either side of r_2 through the vertices of γ_t. Denote these vertices as γ_t^1 and γ_t^2.

The curve γ_t is a discrete principal curve of the surface $\forall t \in [0,1]$, because it is perpendicular to the zero-curvature ruling direction. Thus, we can regard θ as the mean curvature of our discrete developable integrated along the segment from γ_t^1 to γ_t^2. In particular, defining $s(t) = \|\gamma_t^2 - \gamma_t^1\|$ we write the *discrete mean curvature* of the surface at v as:

$$H(t) = \frac{\theta_{r_2}}{s(t)} \tag{1}$$

Of course, $H(t)$ is rarely constant as a function of t, representing the fact that the same folding angle θ_{r_2} gives different mean curvatures as the rulings become more and more spread out.

The curve γ_t can leave the boundary of the sheet. In this case we still take s to be the length of the segment of γ_t intersecting r_2 extended beyond the boundary of P; this way curvature does not shrink simply because P was cut to have a particular shape. Take $l(t)$ to be the length of γ_t between vertices adjacent to r_2 intersected with P; obviously we always have $l(t) \leq s(t)$.

Since our rulings and the border of the planar sheet are both comprised of line segments, $s(t)$ and $l(t)$ are piecewise linear functions of t. Thus, we can identify a sequence $t_0 = 0, t_1, \ldots, t_k = 1$ such that $s(t)$ and $l(t)$ are linear between adjacent values t_i; denote $s_i = s(t_i)$ and $l_i = l(t_i)$. We divide the computation of the bending energy associated with r_2 into computation of bending energies for each segment $[t_i, t_{i+1})$.

Taking into account this piecewise linear structure, for a given segment i, we can write our bending energy using the following integral:

$$\begin{aligned} E_i &= d\Delta_t \int_0^1 \frac{\theta_{r_2}^2((1-u)l_i + ul_{i+1})}{((1-u)s_i + us_{i+1})^2}\, du \\ &= \frac{d\Delta_t \theta_{r_2}^2 \left[\Delta_s(l_i s_{i+1} - l_{i+1}s_i) - \Delta_l s_i s_{i+1} \log \frac{s_{i+1}}{s_i})\right]}{s_i s_{i+1} \Delta_s^2} \end{aligned} \tag{2}$$

where $d = \|v_2 - v_1\|$ and $\Delta_\chi = \chi_{i+1} - \chi_i$ for any variable χ; assume $s_i \leq s_{i+1}$ and otherwise swap accordingly. The total mean curvature bending energy associated with r_2 is simply $E = \sum_{i=1}^{k-1} E_i$.

While the terms of (2) are fairly complex, they can be computed in closed form from the local geometry of the discrete developable. Our implementation computes the derivatives using the open source automatic differentiation library FADBAD++.

5.1. Dealing with Singularities

General developable surfaces may contain singular points of unbounded mean curvature. While the mathematical objects may contain such points, their physical counterparts cannot. Evidence is found in the localized permanent scars formed when a sheet of paper is crumpled [Huf76, BP97, CM05, AP09]. Near the singular points, the physical sheet must stretch to remain smooth, and this strain causes permanent damage.

The thin plate elastic potential—the sum of the squared mean curvature bending energy and a membrane stretching energy—is bounded only for smooth surfaces [AP09]. Surfaces with singular points are outside the space of minimizers of this potential. However, admissible smooth surfaces are always nearby, and in particular the discrepancy is observed only in an arbitrarily small neighborhood of the singular points, where the squared mean curvature elastic potential becomes unbounded. This phenomenon is most clear in the geometry of a circular cone, whose rulings all emanate from a point; approaching the tip of the cone, the radius of the circle vanishes, as does the radius of curvature.

Because the discrepancy arises in an arbitrarily small neighborhood, it does not affect the solution in a macroscopic sense. Our implementation *implicitly* cuts a small hole around each singularity when evaluating the energy, by setting $t_0 = \varepsilon > 0$ and correspondingly interpolating s_0 and l_0.

Some physical experimentation on real-world developable materials like paper and sheet metal remains to be done to verify if this hole-cutting approximation aligns with the infinitesimal stretching that occurs in these materials at singular points; we have yet to identify a reasonable test case in which it yields nonphysical behavior. Furthermore, the form of (2) indicates that our approximation will at least not have strong dependence on the choice of ε, since the divergence of E_0 occurs like $O(\log \varepsilon)$ as $s_0 = \varepsilon \to 0$.

6. Discrete Developable Surface Editing

We have implemented an interactive tool for exploring the manifold of discrete developable surfaces. The interface displays views of the developable surface both in its two-dimensional configuration on P as well as the final three-dimensional shape. The user manually can introduce folds or rulings by drawing segments or snapping to the endpoints of existing folds; additional rulings can be introduced by the subdivision techniques described in Section 6.2. Constraints can be added by pinning vertices in the folded or unfolded configurations, fixing folding angles, and so on.

As the user changes the configuration, his or her motions are projected back onto the manifold of acceptable configurations. Introduction of folds violating basic topological and geometric constraints explicitly is prohibited. More interestingly, when folding angles or vertex positions are changed by the user, an iterative primal-dual interior point method with moderately tight error tolerances is used to find a nearby admissible configuration. In particular, the tool attempts to minimize $E_{config}(V,\Theta,E) = \sum_i \|v_i - v_i^*\|^2 + \alpha\sum_j(\theta_j - \theta_j^*)^2$, where * denotes the configuration after applying the user's desired edits, subject to the constraints in §4. Since v^*, θ^* are nearly admissible after incremental edits, we find there to be minimal dependence on α and choose $\alpha = 1$. Since these projections occur for each small user-introduced change, few iterations are needed to return to a compatible configuration starting from $v_i = v_i^*, \theta_j = \theta_j^*\ \forall i, j$. Thus, our system can deal with moderately-sized ruling topologies including interior vertices interactively.

6.1. Curvature-Based Relaxation

We can substitute the mean curvature bending energy (2) for the proximity measure from §6 to formulate an energy minimization problem for relaxing rulings on discrete developable surfaces. Without additional constraints, this minimization simply will flatten the surface, since the lowest possible bending energy $E = 0$ is obtained when all fold angles are exactly zero. By pinning points in the folded configuration or by fixing even just one folding angle, however, the solution of this variational problem becomes far from trivial, attempting to smooth out the surface while moving only in the space of admissible developables.

We allow the user to specify where on the developable sheet the bending energy should be evaluated, distinguishing between "folds" and "rulings," the latter only appearing on smooth regions. Not all regions of the surface can be marked as smoothly folded, such as interior polygons within fold segments, which are provably flat; the user is restricted to smoothing those sections with or without cone singularities for which the bending energy can be evaluated. While constraints can be added to localize the relaxation to a single portion of the developable surface, often times higher degrees of flexibility can be obtained by allowing more parts to move.

Figure 8 shows an example of curvature-based relaxation applied to a sharply-folded surface after applying the subdivision technique described below. As expected, this process smooths sharp corners while obeying user-specified constraints, providing a simple way to obtain a smooth developable from a rough configuration.

6.2. Subdivision

We provide a subdivision operator for refining smoothly-bent regions, similar in spirit to Liu et al. [LPW*06]. We identify rulings whose folding angle is above some threshold $\hat{\theta}$ and add rulings with 0° folding angles flanking them on either side. We apply mean curvature relaxation to the subdivided surface and repeat until all fold angles are below $\hat{\theta}$. Combining subdivision, relaxation, and modeling yields an effective workflow for dealing with discrete developable surfaces, in which the user makes a rough model using few rulings as an initial step and then subdivides and relaxes to obtain the final surface, as exemplified in Figure 8.

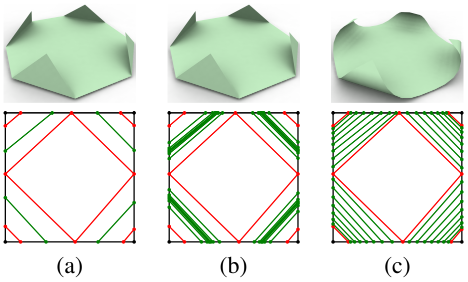

Figure 8: *The rough user-input configuration (a) is subdivided (b) to add rulings but preserve the geometry and then relaxed (c) using the mean curvature bending energy.*

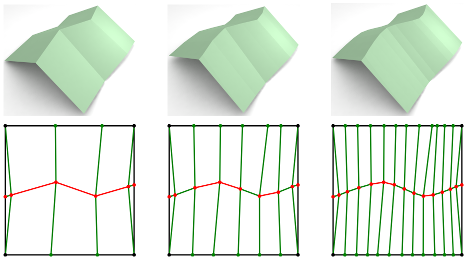

Figure 9: *Successive applications of our curved fold subdivision scheme generates smoother and smoother interior folds across the interior.*

We also provide a *crease* subdivision operator for approximating curved folds rather than curved pieces. In this case, the user identifies chains of segments in $F \backslash R$ marked as belonging to a discrete curved fold. We apply four-point subdivision to the curve, yielding new vertices in the two-dimensional configuration. These vertices are connected using additional rulings to boundary segments on their neighboring faces, and the resulting configuration is relaxed. Two iterations of curved fold subdivision are shown in Figure 9. Note that the curved fold as well as the developable pieces on either side become smoother with each application of our operator.

6.3. Implementation Details

We use the IPOPT library for nonlinear optimization, with the Watson Sparse Matrix Package for linear algebra and the

Eigen library for matrix operations. On a 2.4 GHz Xeon processor, we can deal with editing configurations with up to 50 rulings at interactive rates; better speeds likely could be achieved by making use of parallel computations to evaluate derivatives and project.

One aspect of our formulation that requires special attention during optimization is the enforcement of the fold-fold interaction constraints from §4.2. These constraints always guarantee admissibility but prevent folds from moving from one edge of P to another. To circumvent this issue, during optimization we simply check for fold vertices coincident with vertices of P; if their energy gradient directs them into a different segment of P, we make that localized change to the topology of the surface and continue with optimization. Such a method can be regarded as an “active set” technique leaving constraints that folds do not leave far-away parts of P in the inactive set.

Results Figures 1 and 10 show examples of discrete developable surfaces modeled using our tool. We find that we can achieve a variety of smooth and oragami effects within our system while navigating completely within the space of admissible folded configurations of the domain P.

7. Discussion

We have proposed a flexible discrete structure for representing developable surfaces. Our structure acknowledges the unique requirements of developability rather than attempting to approximate them on meshes or other less well-suited domains. For this reason, we are able to allow users to explore the space of developables interactively. Additionally, we provide a mean curvature bending energy on our flexible discrete developables allowing for relaxation of parts of the surface corresponding to smooth bends rather than folds; together with schemes for curved fold and ruling subdivision we can obtain approximations of smooth developables with straight and curved creases as well as point singularities.

Limitations While our system is among the first incorporating concepts from smooth and non-smooth geometry of developable surfaces for design and editing, it remains subject to many computational challenges surrounding developability. Principally, projection to the manifold of admissible configurations is a nonlinear operator solving a complex constrained optimization problem. Our system remains interactive at moderate scales by moving incrementally as the user changes the configuration, providing initial conditions for the optimization that already nearly satisfy the constraints; the objective discourages snapping to far-away configurations by punishing large steps. Taking larger steps or handling general non-convex domains would require a more robust and efficient optimization. Additionally, there is potential for getting caught in local minima.

Other limitations are related to the communication of the stringent developability constraints to the user. Although with practice it becomes straightforward to predict which fold configurations will be flexible, the space of admissible folds and linked degrees of freedom can be less than obvious while inspecting the unfolded and folded configurations. An indicator of the stiffness of each apparent degree of freedom would make constructions easier to understand without experimentation. Similarly, soft constraints on degrees of freedom like folding angles and vertex positions might enable trade-offs between the user’s desired outcomes when they are not exactly achievable while satisfying developability; this feature would avoid instabilities that result when the user specifies constraints that are inadmissible or stiff.

Extensions of the tool may improve its functionality and the steps needed to model certain types of developables. For instance, in some applications it is desirable to avoid self-intersecting surfaces, and some additional work will be required to incorporate such nonlocal constraints into our optimization. Additionally, combining our subdivision techniques with the bending energy-based optimization may enable dynamic modification of ruling topologies to relax sharp edges automatically. Other methods for inserting singularities also might help to model crumpled paper and other highly-folded or buckled configurations with many interior vertices. Finally, our current metaphor involving the insertion and editing of folds and bends mimics physical modeling with a sheet of paper but may be tedious for large-scale models; a user study observing developable modeling workflows may determine more scalable operations.

Future Work This exploratory work suggests the potential for an assortment of follow-up research. Within applications to modeling, a tuned implementation paired with a careful choice of nonlinear solvers may make it possible to work with higher numbers of interacting rulings; a related option for achieving higher complexity would be to allow for gluing of multiple developable patches to achieve isolated points or curves of nonzero Gauß curvature. On the less mathematical side, user studies ascertaining the best way to communicate the highly nonlinear manifold of admissible developables to a novice user may make our tool more intuitive.

Our work has potential application in a variety of disciplines. Developable surfaces appear in manufacturing and other fields in which molding materials is more easily accomplished using bending rather than stretching. Relaxing our model to allow for limited stretching at isolated singularities may be sufficient for the simulation of cloth, paper, and related materials; initial tests show that our mean curvature energy can be used to generate animations of developable surfaces bending under different stresses. Such applications likely will be able to demonstrate unique phenomena associated with developable surfaces at both coarse and fine scales using our flexible structure, which does not suffer from locking, stiffness, or other artificial issues created by conventional discrete approximations of developability.

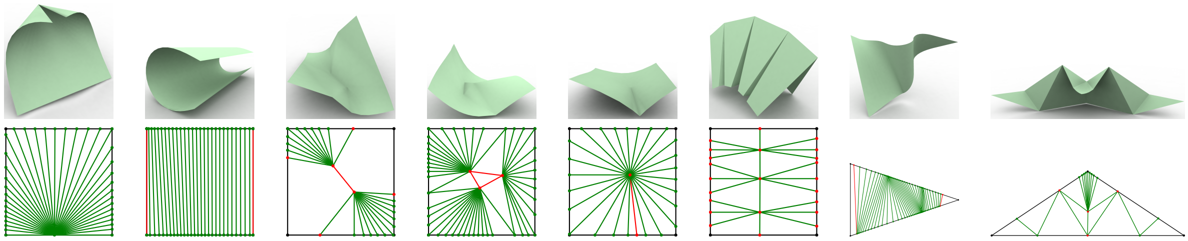

Figure 10: *Developable surfaces produced by the proposed method and their corresponding two-dimensional fold configuration. Folds or vertices highlighted in red are aspects of the two-dimensional configuration fixed by the user before smoothing.*

Acknowledgments

This research was made with government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate Fellowship 32CFR168a. It also received support from the Sloan Foundation, NSF (CCF-06-43268, IIS-09-16129, IIS-10-48948, IIS-11-17257, CMMI-11-29917, GRF), DoD, AFOSR, Hertz Foundation, and generous gifts from Adobe, Autodesk, Intel, mental images, NVIDIA, Side Effects Software, and The Walt Disney Company.

References

[AP09] AUDOLY B., POMEAU Y.: *Elasticity and Geometry: From hair curls to the nonlinear response of shells*. Oxford University Press, USA, Dec. 2009. 2, 6

[Aum03] AUMANN G.: A simple algorithm for designing developable Bézier surfaces. *Computer Aided Geometric Design 20*, 8-9 (2003), 601–619. 2

[Aum04] AUMANN G.: Degree elevation and developable Bézier surfaces. *Computer Aided Geometric Design 21*, 7 (2004), 661–670. 2

[BGW06] BURGOON R., GRINSPUN E., WOOD Z.: Discrete Shells Origami. In *Proceedings of Computers And Their Applications* (Mar. 2006), pp. 180–187. 2

[BI08] BOBENKO A., IZMESTIEV I.: Alexandrov's Theorem, Weighted Delaunay Triangulations, and Mixed Volumes. *Ann. Inst. Fourier 58*, 2 (2008), 447–505. 2

[Bis75] BISHOP R. L.: There is More than One Way to Frame a Curve. *American Mathematical Monthly 82*, 3 (1975), 246–251. 3

[BP97] BEN AMAR M., POMEAU Y.: Crumpled paper. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 453*, 1959 (Apr. 1997), 729–755. 6

[BW07] BO P., WANG W.: Geodesic-controlled developable surfaces for modeling paper bending. *Computer Graphics Forum 26*, 3 (2007), 365–374. 2

[BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete Elastic Rods. *SIGGRAPH (ACM Transactions on Graphics) 27*, 3 (Aug 2008), 1–12. 5

[Car76] CARMO M. P. D.: *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976. 3

[CB98] CHAPELLE D., BATHE K. J.: Fundamental considerations for the finite element analysis of shell structures. *Computers and Structures 66*, 1 (1998), 19–36. 1, 2

[CC90] CHU C.-H., CHEN J.-T.: Geometric Design of Developable Composite Bézier Surfaces. *Computer Aided Design And Applications* (1990), 531–540. 2

[Cha09] CHANG S. H.-H.: Origami Simulator: a Multi-Touch Experience. *Building* (2009), 3889–3894. 2

[CLL*99] CHEN H.-Y. Y., LEE I.-K. K., LEOPOLDSEDER S., POTTMANN H., RANDRUP T., WALLNER J.: On Surface Approximation Using Developable Surfaces. *Graphical Models and Image Processing 61*, 2 (Mar. 1999), 110–124. 2

[CM05] CERDA E., MAHADEVAN L.: Confined developable elastic surfaces: cylinders, cones and the Elastica. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 461*, 2055 (Mar. 2005), 671–700. 6

[CS02] CHU C.-H., SÉQUIN C. H.: Developable Bézier patches: properties and design. *Computer-Aided Design 34*, 7 (June 2002), 511–527. 2

[CT10] CHEN M., TANG K.: A fully geometric approach for developable cloth deformation simulation. *The Visual Computer 26*, 6-8 (2010), 853–863. 2

[CWT08] CHU C.-H., WANG C. C., TSAI C.-R.: Computer aided geometric design of strip using developable Bézier patches. *Computers in Industry 59*, 6 (Aug. 2008), 601–611. 2

[DJW*06] DECAUDIN P., JULIUS D., WITHER J., BOISSIEUX L., SHEFFER A., CANI M.-P.: Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum 25*, 3 (Sept. 2006), 625–634. 2

[EB08] ENGLISH E., BRIDSON R.: Animating developable surfaces using nonconforming elements. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 2

[FB93] FREY W., BINDSCHADLER D.: *Computer aided design of a class of developable Bézier surfaces*. Tech. rep., GM research publication R&D-8057, 1993. 2

[FJ07] FERNÁNDEZ-JAMBRINA L.: B-spline control nets for developable surfaces. *Computer Aided Geometric Design 24*, 4 (May 2007), 189–199. 2

[Fre02] FREY W.: Boundary triangulations approximating developable surfaces that interpolate a closed space curve. *International Journal of Foundations of Computer Science 13*, 2002 (2002), 285–302. 2

[Fre04] FREY W.: Modeling buckled developable surfaces by triangulation. *Computer-Aided Design 36*, 4 (2004), 299–313. 2

[Huf76] HUFFMAN D. A.: Curvature and creases: A primer on paper. *IEEE Transactions on Computers 25*, 10 (1976), 1010–1019. 2, 6

[Ida08] IDA T.: Graph Rewriting in Computational Origami. In *2008 10th International Symposium on Symbolic and Numeric*

Algorithms for Scientific Computing (2008), vol. 1, IEEE, pp. 20–27. 2

[IT10] Ida T., Takahashi H.: Origami fold as algebraic graph rewriting. *Journal of Symbolic Computation 45*, 4 (Apr. 2010), 393–413. 2

[JKS05] Julius D., Kraevoy V., Sheffer A.: D-Charts: Quasi-Developable Mesh Segmentation. *Computer Graphics Forum 24*, 3 (2005), 581–590. 2

[KFC*08] Kilian M., Flöry S., Chen Z., Mitra N. J., Sheffer A., Pottmann H.: Curved folding. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 1. 2

[KG90] Kunii T. L., Gotoda H.: Singularity theoretical modeling and animation of garment wrinkle formation processes. *The Visual Computer 6*, 6 (Nov. 1990), 326–336. 2

[KGK94] Kergosien Y. L., Gotoda H., Kunii T. L.: Bending and Creasing Virtual Paper. *IEEE Comput. Graph. Appl. 14*, 1 (Jan. 1994), 40–48. 2

[LLH09] Liu Y.-J., Lai Y.-K., Hu S.: Stripification of free-form surfaces with global error bounds for developable approximation. *IEEE Transactions on Automation Science and Engineering 6*, 4 (2009), 700–709. 2

[LP98] Leopoldseder S., Pottmann H.: Approximation of developable surfaces with cone spline surfaces. *Computer-Aided Design 30*, 7 (1998), 571–582. 2

[LPW*06] Liu Y., Pottmann H., Wallner J., Yang Y.-L., Wang W.: Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics 25*, 3 (July 2006), 681. 2, 7

[LTJ07] Liu Y.-J., Tang K., Joneja A.: Modeling dynamic developable meshes by the Hamilton principle. *Computer-Aided Design 39*, 9 (Sept. 2007), 719–731. 2

[MS04] Mitani J., Suzuki H.: Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics 23*, 3 (Aug. 2004), 259. 2

[MS11] Martedi S., Saito H.: Foldable augmented papers with a relaxed constraint. *2011 1st International Symposium on Access Spaces (ISAS)* (June 2011), 127–131. 2

[Par02] Park F. C.: Design of Developable Surfaces. *Journal of Mechanical Design 124*, December (2002), 602–608. 2

[PB07] Perriollat M., Bartoli A.: A Quasi-Minimal Model for Paper-Like Surfaces. *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), 1–7. 2

[Pet04] Peternell M.: Developable surface fitting to point clouds. *Computer Aided Geometric Design 21*, 8 (2004), 785–803. 2

[Pot95] Pottmann H.: Developable rational Bézier and B-spline surfaces. *Computer Aided Geometric Design 12*, 5 (1995), 513–531. 2

[PS07] Pérez F., Suárez J.: Quasi-developable -spline surfaces in ship hull design. *Computer-Aided Design 39*, 10 (Oct. 2007), 853–862. 2

[PSB*08] Pottmann H., Schiftner A., Bo P., Schmiedhofer H., Wang W., Baldassini N., Wallner J.: Freeform surfaces from single curved panels. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 76:1–76:10. 2

[PW99] Pottmann H., Wallner J.: Approximation algorithms for developable surfaces. *Computer Aided Geometric Design 16*, 6 (1999), 539–556. 2

[RCHT11] Rohmer D., Cani M.-p. P., Hahmann S., Thibert B.: Folded Paper Geometry from 2D Pattern and 3D Contour. In *Eurographics* (2011), pp. 1–4. 2

[Red89] Redont P.: Representation and deformation of developable surfaces. *Computer-Aided Design 21*, 1 (1989), 13–20. 2

[RSW*07] Rose K., Sheffer A., Wither J., Cani M.-p., Thibert B.: Developable surfaces from arbitrary sketched boundaries. *Proc. Symposium on Geometry Processing* (2007), 163–172. 2

[SF96] Sun M., Fiume E.: A Technique for Constructing Developable Surfaces. In *Proc. Conference on Graphics Interface* (1996), pp. 176–185. 2

[SG11] Schenk M., Guest S. D.: Origami Folding: A Structural Engineering Approach. *Origami 5 Fifth International Meeting of Origami Science Mathematics and Education* (2011), 1–16. 2

[SKKO02] Shinagawa Y., Kawamichi R., Kunii T., Ohwada S.: Developing surfaces. In *Proceedings SMI. Shape Modeling International 2002* (2002), IEEE Comput. Soc, pp. 253–260. 2

[STL06] Shatz I., Tal A., Leifman G.: Paper craft models from meshes. *The Visual Computer 22*, 9-11 (2006), 825–834. 2

[Tac09a] Tachi T.: Generalization of rigid foldable quadrilateral mesh origami. *Journal Of The International Association For Shell And Spatial Structures 50*, October (2009), 2287–2294. 2

[Tac09b] Tachi T.: Simulation of rigid origami. In *Origami 4*, Peters A. K., (Ed.), no. October. Editorial de la Universitat Politécnica de Valencia., 2009, p. 175. 2, 3

[Tac10a] Tachi T.: Geometric Considerations for the Design of Rigid Origami Structures. In *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2010* (Shanghai, 2010), vol. 12, Elsevier Ltd, pp. 458–460. 2

[Tac10b] Tachi T.: Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics 16*, 2 (2010), 298–311. 2

[Tac11] Tachi T.: Rigid-Foldable Thick Origami. In *Origami 5: Fifth International Meeting of Origami Science Mathematics and Education* (2011), Wang-Iverson P., Lang R. J., Yim M., (Eds.), Taylor & Francis, pp. 253–264. 2

[TC09] Tang K. T. K., Chen M. C. M.: Quasi-developable mesh surface interpolation via mesh deformation. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (2009), 518–528. 2

[TI08] Takahashi H., Ida T.: Modeling of Origami by Graph Rewriting. *Engineering* (2008), 1–13. 2

[Wan04] Wang C. C. L.: Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *The Visual Computer 26*, 8-9 (2004), 167–539. 2

[Wan08] Wang C.: Computing length-preserved free boundary for quasi-developable mesh segmentation. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 25–36. 2

[WC11] Wang K., Chen Y.: Folding a Patterned Cylinder by Rigid Origami. In *Origami 5: Fifth International Meeting of Origami Science Mathematics and Education* (2011), Wang-Iverson P., Lang R. J., Yim M., (Eds.), Taylor & Francis Group, pp. 265–276. 2

[WF88] Weiss G., Furtner P.: Computer-aided treatment of developable surfaces. *Computers & Graphics 12* (1988), 39–51. 2

[ZW08] Zhao H., Wang G.: A new method for designing a developable surface utilizing the surface pencil through a given curve. *Progress in Natural Science 18*, 1 (2008), 105–110. 2