Folded Paper Geometry from 2D Pattern and 3D Contour

Damien Rohmer[1,2,3], Marie-Paule Cani[1,2], Stefanie Hahmann[1], Boris Thibert[1]

[1]University of Grenoble, Laboratoire Jean-Kuntzmann. [2] INRIA. [3]University of Lyon, CPE Lyon.

Abstract

Folded paper exhibits very characteristic shapes, due to the presence of sharp folds and to exact isometry with a given planar pattern. Therefore, none of the physically-based simulators developed so far can handle paper-like material. We propose a purely geometric solution to generate static folded paper geometry from a 2D pattern and a 3D placement of its contour curve. Fold lines are explicitly identified and used to control a recursive, local subdivision process, leading to an efficient procedural modeling of the surface through a fold-aligned mesh. Contrary to previous work, our method generates paper-like surfaces with sharp creases while maintaining approximate isometry with the input pattern.

1. Introduction

Although extremely common around us, folded paper is almost never represented in 3D films or games. Indeed, none of the existing physically-based simulators, generally designed for cloth, can handle this challenging material. Let us first list the specificities required for modeling paper: Paper material is subject to exact length preservation when it deforms from its 2D, unfolded shape. Therefore, a piece of paper always remains exactly developable onto a plane, whatever the deformations applied to it. Secondly, these deformations are quite different from those of cloth-like material, since they include sharp fold lines. Note that the later would be very difficult to generate using the mesh-based physical simulation used for cloth, since the long, straight sharp folds of arbitrary orientation that may appear under deformation cannot be all well represented by a given predefined mesh. Therefore, modeling paper largely remains an open problem.

In this work, we introduce the first method to generate visually realistic, non regular folded paper geometry, from a 2D pattern and some 3D positioning of its contour curve. Our approach is fully procedural and restricted to the generation of a static shape: the basic idea is to iteratively identify fold lines and use them to recursively subdivide the geometry until isometry with the pattern is approximately restored. The main advantages of our approach are:

- The intuitive shape control we provide through the specification of the 3D contour curve while nearly preserving length with respect to a given pattern.
- The generation of general paper-like surfaces, not restricted to the convex hull of their contour points and where folds are automatically created without the need for any extra user intervention.
- The automatic construction of an adapted mesh, aligned with folds and therefore able to accurately represent sharp fold lines. With our method, even complex surfaces can be accurately represented using few polygons.
- The efficiency of computations, since no global optimization is required.

2. Related Work

Surfaces that preserve length with respect to a 2D pattern were already well studied. As standard simulation was shown to be ill-conditioned when stiff elements are used to limit stretch [CK02], specific physically-based models were introduced [Pro97,VMT07,LTJ07,BB08,TPS09] or the simulation was enhanced with geometric length preservation constraints [EB08, SSBT08]. However, these approaches, developed for cloth-like material, are expensive and none of them is well suited for non-smooth surfaces such as folded paper exhibiting sharp creases. Burgoon [BGW06] included such creases into a thin plate model, but the folding curves were to be traced manually by the user.

In parallel, some purely geometric methods were specifically designed for developable surfaces. Developable tensor product surfaces were first studied [PF95] but do not handle junction between patches. Procedural methods were developed to generate a ruled developable triangulation when the input data is a 3D contour curve [Fre04,RSW*07], positional constraints [Pet04], surfaces geodesics [BW07], or a coarse quadrilateral mesh [LPW*06]. All these methods are able to generate exact developable triangulation. However, they do not take into account length's preservation with respect to a given pattern. Moreover, the resulting surfaces remain very close to C2 developable patches, i.e. ruled surfaces without singular points and with no sharp edges. Therefore such surfaces always lie in the convex hull of the input contour. In consequence, these models cannot represent the general folded paper shapes we are looking for.

An alternative approach is to deform a preexisting mesh by increasing a developability criterion. Non linear minimization was used by Wang [WT04] for static surfaces and by Popa [PZB*09] for captured animated meshes. Tang [TC09, CT10] added an extra length preserving criterion into the energy formulation to preserve both developability and length with respect to a pattern. However, all these approaches require the use of a predefined surface tessellation into a mesh. Therefore, they are not suited to model folded paper with sharp creases (to which mesh edges would need to be aligned). Moreover the computational time always stays larger than a minute.

Lastly, Decaudin [DJW*06] proposed a procedural method to generate developable folds for surfaces wrapped around cylinders. Although this could be used to represent sharp paper-like folds (the mesh being predefined to be aligned with fold lines), only a specific, predefined family of folds was modeled.

In contrast with previous work, our method is especially designed for folded paper. It generates non-smooth surfaces that tend to respect isometry with a pattern while interpolating a contour curve. To do so, both smooth and sharp folds are automatically generated where needed to improve isometry, allowing the surface to bump out of the convex hull of its contour.

3. Recursive generation of folded paper

3.1. Overview and features

The inputs of our algorithm are a 2D convex pattern, whose closed polygonal 2D contour is noted $\overline{\Gamma}$, and a closed polygonal 3D curve representing a valid 3D positioning of $\overline{\Gamma}$, noted $\Gamma : t \mapsto \Gamma(t)$. The bijection from $\overline{\Gamma}$ to Γ is supposed to be given.

The output is a paper-like triangulated surface interpolating the 3D contour curve and almost isometric to the 2D pattern.

The algorithm, inspired by divide-and-conquer methods, iteratively inserts straight or curved segments between two selected points on the contour of the part of the surface under consideration. The choice of contour points and the shape of the inserted curve between them are based on length criteria to make the resulting surface more and more isometric to the pattern. When curve insertion stops the final surface is generated by smoothly triangulating all surface parts delimitated by the curves. The outline of the algorithm is as follows:

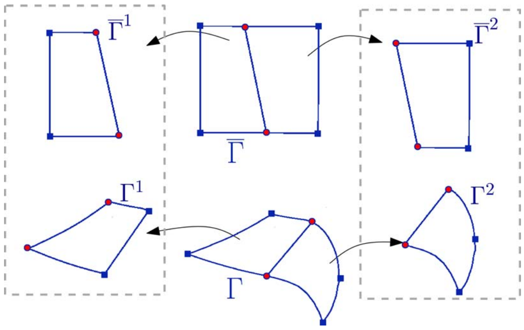

Figure 1: *Recursive subdivision step: the 2D pattern and the corresponding 3D contour (middle) are split into two pieces (left and right) along a fold line, which is straight in pattern space.*

1. Find the line segment $[\overline{\Gamma}(t_1), \overline{\Gamma}(t_2)]$ to be processed in priority on the 2D pattern. Compute its 3D mapping either as a straight line or as a 3D curve, fig. 1 middle.
2. Split the pattern region Γ into two parts $\overline{\Gamma}^1$ and $\overline{\Gamma}^2$, and associate to them the new local 3D contour curves, Γ^1 and Γ^2 respectively, fig. 1 left and right.
3. Repeat the subdivision steps recursively on the two independant regions by considering the new couples $(\overline{\Gamma}^1, \Gamma^1)$ and $(\overline{\Gamma}^2, \Gamma^2)$ respectively, as input.

Straight lines are inserted in priority when it's possible to find a pair of contour points of the pattern whose 3D counterpart has equal Euclidian distance. If this is not possible and the 3D distance is smaller than the corresponding distance on the pattern, then we have a folded surface inbetween. We therefore have to permit the surface to overpass the convex hull of its contour curve in order to restore isometry to the pattern. A curved fold curve is thus inserted to model this kind of surface. Its profile is determined by a local length minimization in order to improve length preservation through the subdivision steps. All inserted fold lines (curves and straight lines) are polygonal and will belong to the final mesh. No extra remeshing is therefore necessary in order to align the mesh edges with these fold lines. Depending on local mesh geometry they may represent sharp features. Figure 2 illustrates the steps of constructing a surface from a slightly deformed contour curve.

3.2. Fold line computation

Case of a straight 3D line. First note that the segment $[\Gamma(t_1), \Gamma(t_2)]$ is lying on the 3D surface iff

$$\|\Gamma(t_1) - \Gamma(t_2)\| = \|\overline{\Gamma}(t_1) - \overline{\Gamma}(t_2)\| \, . \qquad (1)$$

Therefore, given Γ and $\overline{\Gamma}$, we search for the couple (t_1, t_2) which best satisfies eq.(1). If $\|\Gamma(t_1) - \Gamma(t_2)\| - \|\overline{\Gamma}(t_1) - \overline{\Gamma}(t_2)\| \geq 0$ then there is no compression in this part of the

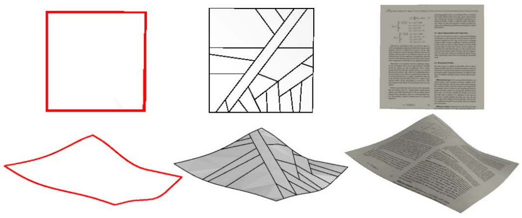

Figure 2: *Reconstruction of a sheet of paper. Left to right: initial contours (2D pattern and 3D), recursive insertion of straight fold lines, resulting surface.*

surface, and a straight 3D line would be the best solution to connect the two points in 3D. Otherwise, we have the case of local compression, meaning that the surface between both positions is not planar. A straight line would not be sufficient.

Case of a curved 3D line. If no straight segment can be found, we search for the pair of vertices with least compression in Euclidean distance. Let's suppose that the mapping of the least compressed segment $[\overline{\Gamma}(t_1), \overline{\Gamma}(t_2)]$ is given by the 3D curve c. We choose c to be a cubic polynomial for the following reasons:

- Cubics can accurately approximate cylindrical or conical sections forming C^2 developable patches.
- Cubics are minimizing the bending energy, therefore no spurious oscillation is introduced.
- Efficient fitting is possible due to the limited number of degrees of freedom.

A good candidate for the fold curve c is a curve which improves the preservation of length through the recursive splitting algorithm. We therefore look for the optimal cubic curve such that the mesh computed after subdivision minimizes the error in length with respect to the pattern.

To compute such an error, we define two pairs of trian-

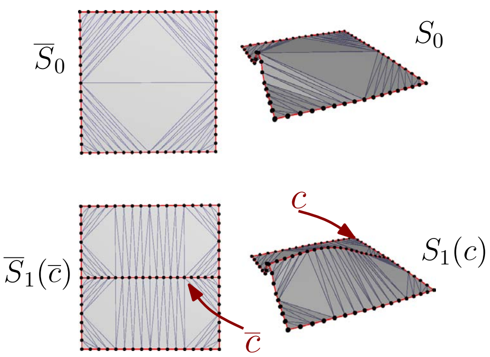

Figure 3: *Pair of meshes* $(\overline{S}_0, S_0)$ *without interior points and* $(\overline{S}_1(\overline{c}), S_1(c))$ *after adding the 3D curve c.*

gulated surfaces such as illustrated in fig. 3. The first pair $(\overline{S_0}, S_0)$ corresponds to the mesh at the current step. This mesh is defined by the flat Delaunay triangulation, without interior vertices, based on the pair of curves $(\overline{\Gamma}, \Gamma)$. The second pair $(\overline{S_1}(\overline{c}), S_1(c))$ corresponds to the mesh obtained after adding the interior fold curve c and its associated 2D segment on the pattern $\overline{c} = [\overline{\Gamma}(t_1), \overline{\Gamma}(t_2)]$. Finally, the length error E is mesured as the L^2 norm on the edges of the mesh:

$$E(\overline{S}, S) = \sum_{\text{edges } i} \left(L_i - \overline{L}_i\right)^2 ,$$

with L_i (resp. $\overline{L}_i$) the i$^{\text{th}}$ edge of the mesh S (resp. $\overline{S}$). Note that $E(\overline{S}, S) = 0$ means that the lengths are exactly preserved everywhere, which means the surface is developable.

Finding the curve such that the surfaces created on both sides best preserve the lengths with respect to their pattern can therefore be expressed as a local minimization problem $\min E(\overline{S}(\overline{c}), S(c))$ where the curve c is the unknown.

In practice this non-linear minimization is solved using a gradient descent method, the BFGS method, starting from an initial guess given by a straight line. This is the most costly part of the algorithm. However this minimization is performed only locally and in a space of low dimension, so the method remains extremely cheap compared to global minimization approaches.

3.3. Surface generation

The recursive splitting ends when the surface parts are almost planar. A surface mesh interpolationg the initial contour curve and all inserted segments is generated by triangulating each surface part individually. As for each part the mapping onto the 2D pattern is known, a standard constrained planar Delaunay is performed in the pattern space as illustrated in fig. 3 and then mapped onto 3D. The fold lines are thus embedded into the mesh. This yields a good representation of sharp creases, which are a typical feature of folded paper.

4. Results

Fig. 2 shows our result in a simple case where no compression is detected: only straight lines are generated so the surface remains in the convex hull of its contour. Note that despite of the quite large 3D deformation, good corresponding rulings are found and the texture is not deformed due to length preservation.

Further results of folded paper are shown in fig.4. Firstly, note that our results are not constrained to stay within the convex hull defined by the 3D contour curve. Secondly, the algorithm automatically generates a mesh with edges aligned along the folds. Therefore very few polygons are required to accurately model non-smooth surfaces. Finally, we can observe that our results are actually closer to the real deformation of a piece of paper (at bottom right) than to cloth simulations with the same contour (yellow surfaces). The reason for that is that we couple a strong constraint on lengths along the surface with the possibility to produce non smooth geometry.

All presented results were computed in less than 1s thanks to our local subdivision approach. Moreover, when only straight lines are required, the method is about 10 times faster. In addition, all the resulting surfaces have relative length error below 5%.

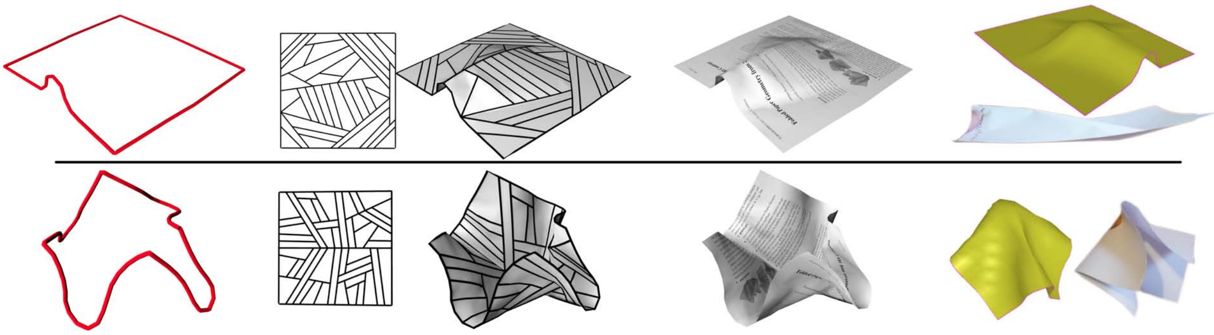

Figure 4: *Results for folded surfaces. From left to right: Input 3D contour curve; Pattern and 3D surface with separation segments; Final rendering; Comparison between cloth simulator (yellow) and real experiment (white).*

The input 3D contour curves were computed using a standard cloth simulator using high stiffness mass-spring elements.

5. Discussion and Future Work

We have presented an efficient method to generate paper-like geometry that approximately preserves length with respect to a 2D pattern. The method seamlessly handles non-smooth surfaces thanks to a recursive meshing process that aligns edges along fold curves.

Despite of its good visual results, our method still has a number of limitations, which we hope will inspire future research: Firstly, providing valid 3D contour curves as an input is not an easy task: most closed curves of the right length would not be a solution for the contour of a surface isometric with the pattern. Therefore, we used a cloth-simulator to generate our input curves, which is a bad solution in terms of shape control. Being able to input just a few position constraints would be highly desirable. Secondly, our algorithm is a greedy one: nothing insures that it converges towards the best possible solution, and the latter might not be reachable with the cubic curves we use: therefore, our isometry error generally stops decreasing after while, which mean we reached a local minima. Thirdly, the current method does not insure that the generated surface is self-intersection-free, and nor avoids collisions with obstacles during the generation process, so the result is not always visually plausible. Lastly, setting up an animation method for dynamically folding virtual paper is still an unsolved challenge. The method we just presented is only a first step in this direction.

References

[BB08] BENDER J., BAYER D.: Parallel simulation of inextensible cloth. *Proc. of VRIPhys.* (2008).

[BGW06] BURGOON R., GRINSPUN E., WOOD Z.: Discrete shells origami. *Proc. of Comp. and their App.* (2006), 180–187.

[BW07] BO P., WANG W.: Geodesic-controlled developable surfaces for modeling paper bending. *Comp. Graph. Forum. 26* (2007).

[CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Trans. on Graph. 21* (2002), 604–611.

[CT10] CHEN M., TANG K.: A fully geometric approach for developable cloth deformation simulation. *The Vis. Computer.* (2010).

[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 Graph. Forum.* (2006), 625–634.

[EB08] ENGLISH E., BRIDSON R.: Animating developable surfaces using nonconforming elements. *ACM Trans. on Graph. 27* (2008), 1–5.

[Fre04] FREY W.: Modeling buckled developable surfaces by triangulation. *Comp. Aided Design. 36* (2004).

[LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Trans. on Graph. 25* (2006).

[LTJ07] LIU Y.-J., TANG K., JONEJA A.: Modeling dynamic developable meshes by the hamilton principle. *Comp. Aided Design. 39* (2007).

[Pet04] PETERNELL M.: Developable surface fitting to point clouds. *Comp. Aided Geometric Design. 21* (2004).

[PF95] POTTMANN H., FARIN G.: Developable rational bézier and b-spline surface. *Comp. Aided Geometric Design. 12* (1995).

[Pro97] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface.* (1997), 147–154.

[PZB*09] POPA T., ZHOU Q., BRADLEY D., KRAEVOY V., FU H., SHEFFER A., HEIDRICH W.: Wrinkling captured garments using space-time data-driven deformation. *Comp. Graph. Forum. 28* (2009).

[RSW*07] ROSE K., SHEFFER A., WITHER J., CANI M.-P., THIBERT B.: Developable surfaces from arbitrary sketched boundaries. *Symp. on Geometry Processing.* (2007).

[SSBT08] STUMPP T., SPILLMANN J., BECKER M., TESCHNER M.: A geometric deformation model for stable cloth simulation. *Proc of VRIPhys.* (2008).

[TC09] TANG K., CHEN M.: Quasi-developable mesh surface interpolation via mesh deformation. *IEEE Trans. on Vis. and Comp. Graph. 15* (2009).

[TPS09] THOMASZEWSKI B., PABST S., STRASSER W.: Continuum-based strain limiting. *Comp. Graph. Forum. 28* (2009), 569–576.

[VMT07] VOLINO P., MAGNENAT-THALMANN N.: Stop-and-go cloth draping. *The Visual Computer.* (2007), 669–677.

[WT04] WANG C., TANG K.: Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *The Visual Computer. 20* (2004).