# Developable Strip Approximation of Parametric Surfaces with Global Error Bounds

Yong-Jin Liu, Yu-Kun Lai, Shi-Min Hu
Department of Computer Science and Technology
Tsinghua University, Beijing, China
{liuyongjin, laiyk03, shimin}@tsinghua.edu.cn

## Abstract

*Developable surfaces have many desired properties in manufacturing process. Since most existing CAD systems utilize parametric surfaces as the design primitive, there is a great demand in industry to convert a parametric surface within a prescribed global error bound into developable patches. In this work we propose a simple and efficient solution to approximate a general parametric surface with a minimum set of $C^0$-joint developable strips. The key contribution of the proposed algorithm is that, several global optimization problems are elegantly solved in a sequence that offers a controllable global error bound on the developable surface approximation. Experimental results are presented to demonstrate the effectiveness and stability of the proposed algorithm.*

## 1. Introduction

Developable surfaces, which can be unfolded into plane without stretch, are widely used in engineering. While developable surfaces can be directly used to model some simple shapes such as cones and cylinders, most existing CAD/CAM systems use general parametric surfaces (e.g. NURBS) as primitives to model complicated free-form shapes. Therefore, there is a great demand in industries to convert a general parametric surface within a prescribed global error bound into a minimum set of developable pieces. These pieces are afterwards cut from planar material, bent back without stretch and distortion into their final positions and stitched together to form the final product.

If sufficient differentiability is assumed, developable surfaces can only be part of plane, cone, cylinder, tangent surface of a curve or a composition of them. Surface approximation using conical and cylindrical patches are studied in [8, 11, 14, 17]. Tangent surfaces of B-spline curves in dual space based on projective geometry are studied in [1, 2, 7, 13, 15]. Cone spline surfaces that can be used as transition of smooth joining developables are studied in [9]. For nearly developable surfaces, a spherical curve segmentation and fitting technique is presented in [3]. If the given surface is far from developables, Elber [4] proposes an approximation method that trims the surface using isolines and interpolates each trimmed piece by ruled patches. This method is extended by Subag and Elber [18] where a piecewise developable surface approximation is built, however, at the cost of involving complicated and possibly unstable numerical process.

Recent advances in computer-aided design have revealed that the developable surfaces can be in effect approximated by triangle strips [4, 5, 6, 8, 10, 19]. In this paper, we follow the direction and use triangle strips to simulate the real rolling processing in exactly the same $C^0$ way. We will also show that carefully constructing the triangle strips can lead to developable patches with the bending energy of the rolling process minimized. Our work is also related to the strip-based approximation algorithm in [12]; however, they are different in that graphical models are significantly different from CAD ones.

An optimal algorithm is proposed in this paper that achieves developable strip approximation by trimming an arbitrary parametric surface into a minimum set of strips; each strip consists of a chain of triangles that is clearly developable. The novelty of the presented algorithm is that we introduce a controllable global error bound into the trimming process such that the resulting piecewise strip approximation of the general parametric surface is globally optimal.

## 2. Algorithm overview

The basic idea of the presented algorithm is simple. Given a rectangular parametric surface with boundaries $e_1, e_2, e_3, e_4$ in counter clockwise order, from two pairs of opposite edges $(e_1, e_3)$ and $(e_2, e_4)$, an optimal pair is identified. Without loss of generality, assume the optimal pair is $(e_1, e_3)$. The curves $e_1$, $e_3$ are discretized by polylines with a prescribed tolerance: the number of vertices on each polyline is maintained to be the same. Then the corresponding vertices between two polylines are connected by geodesics on the parametric surface. These geodesics trim the surface into pieces which afterwards are optimally approximated by developable triangle strips. The overall algorithm, called **DevAppr**, is summarized below:

1. (Section 3) Optimally discretize curves $(e_1, e_3)$ into two polylines $(p_1, p_3)$ with the same number $n$ of vertices; the discretization error satisfies $\max\{\|e_1 - p_1\|, \|e_3 - p_3\|\} < \varepsilon$, where $\varepsilon$ is a prescribed tolerance.
2. (Section 4) Construct the geodesics by connecting the corresponding vertices $(v_i^1, v_i^3)$, $i = 2, \cdots, n-1$, in the polylines $p_1 = \{v_1^1, v_2^1, \cdots, v_n^1\}$ and $p_3 = \{v_1^3, v_2^3, \cdots, v_n^3\}$. Together with the boundary curves, these geodesic curves partition the parametric surface into strips $\{S_1, S_2, \cdots, S_{n-1}\}$.
3. Discretize curves $e_2$, $e_4$ and each geodesic $g_i$ into polylines $p_2$, $p_4$ and $l_i$, with the minimum number of vertices, respectively, such that $\|e_j - p_j\| < \varepsilon, j = 2, 4$ and $\|g_i - l_i\| < \varepsilon$.
4. (Section 5) For every pair $(l_i, l_{i+1})$ of adjacent polylines in the set $\{l_1 = p_2, l_2, \cdots, l_{n-1}, l_n = p_4\}$, construct an optimal strip of triangles $T_i$ that minimizes an elaborately designed energy functional.
5. Given a trimmed strip $S_i$ and its associated counterpart of triangles $T_i$, if the error $\|S_i - T_i\| > \varepsilon$, then subdivide the strip $S_i$ into two and repeat the process.
6. Develop all triangle strips into the same plane.

# 3. Polygonal approximation of parametric curves

The steps 1 and 3 in the Algorithm **DevAppr** require the solutions to two optimization problems: *Min-#* and *Min-$\varepsilon$*.

Given a parametric curve $c$, approximate it by a polyline $p$, the approximation error (denoted as $E(c, p)$) is defined in a particular metric $L_l$; $l = 2$ and $l = \infty$ are commonly used.

## 3.1 Solution to Min-# problem

Min-# problem approximates a given parametric curve $c$ by a polyline $p$, with the minimum number of segments, such that the approximation error $E(c, p)$ does not exceed a given tolerance $\varepsilon$. The solution to this problem is as follows. We start from one end of the curve and greedily add samples one by one, till we reach the other end of the curve. To add a new sample, we put a cylindrical surface with radius $\varepsilon/2$ centered at the current position, and oriented along with the tangent direction of the curve at the position. The nearest intersection point of the curve with the cylindrical surface is considered as the next sample. This simple greedy scheme is easily seen to be nearly optimal in worst case when the $L_\infty$ error measure is used.

## 3.2 Solution to Min-$\varepsilon$ problem

Min-$\varepsilon$ problem approximates a given parametric curve $c$ by a polyline $p$ with a given number of line segments $n$, such that the approximation error $E(c, p)$ is minimized. This can be achieved by optimizing a $n - 2$ vector $U = (u_2, \cdots, u_{n-1})^T$ in the optimization function defined by

$$Opt(U) = \sum_{i=1}^{n-1} \int_{u_i}^{u_{i+1}} \|c(u) - \frac{(u_{i+1} - u)c(u_i) + (u - u_i)c(u_{i+1})}{u_{i+1} - u_i}\|^2 du \tag{1}$$

where $c(u_i)$, $i = 1, \cdots, n$ are $n$ optimal positions of samples with $u_1 = 0, u_n = length(c)$. Minimization of function $Opt(U)$ is a typical multi-dimensional optimization problem which can be solved by the classical gradient or simplex methods.

Given the solutions to the above two problems, the step 1 in Algorithm **DevAppr** is performed first by optimally discretizing $e_1$ and $e_3$ with a given tolerance $\varepsilon$; that solves a Min-# problem. Denote the resulting numbers of vertices on $p_1$ and $p_3$ by $n_1$ and $n_3$, respectively. Without loss of generality, let $n_1 \le n_3$. To maintain the same vertex number on $p_1$ and $p_3$, we need to re-sample $e_1$ with a given number of line segments $n_3$, that solves a Min-$\varepsilon$ problem. The solution to Min-# problem also uses in step 3. Unless otherwise specified, all the parametric curves are parameterized by the arc-length.

The optimal pair determined from the boundary curves $e_1, e_2, e_3, e_4$ of the parametric surface is used to locate the endpoints of geodesics for trimming the surface. Two strategies are used to find the optimal pair, either the pair with the minimum common samples or the one interactively specified by the user.

# 4. Compute geodesics on parametric surface

In our scenario, we need to find a solution to the following problem: *given two points $\mathcal{P}$, $\mathcal{Q}$ on surface, compute the exact geodesic passing $\mathcal{P}$ and $\mathcal{Q}$.* Denote the given parametric surface as $S = \{x(u, v), y(u, v), z(u, v)\}$. Let the prescribed source points $\mathcal{P}$, $\mathcal{Q}$ be specified by parameters $(u_p, v_p), (u_q, v_q)$ and, let any curve on surface connecting them be specified by function $v = v(u)$ with $v_p = v(u_p)$

and $v_q = v(u_q)$. The length of curve specified by $v(u)$ is

$$I = \int_{u_1}^{u_2} \sqrt{E(u,v) + 2F(u,v)v' + G(u,v)v'^2} du$$

where $v' = dv/du$ and $E, F, G$ are the first fundamental forms. Due to the nature of geodesics, our problem is to find the function $v(u)$ that minimizes the above functional. By calculus of variations, the solution must satisfy the Euler-Lagrange equation:

$$\frac{\partial f}{\partial v} - \frac{d}{du}\left(\frac{\partial f}{\partial v'}\right) = 0, \quad (2)$$

$$where f = \sqrt{E + 2Fv' + Gv'^2}.$$

With the boundary conditions $v_p = v(u_p)$ and $v_q = v(u_q)$, we need to solve for $v(u)$ a *two point boundary value problem*. Relaxation method [16] is applied here; leading to a set of finite difference equations (FDEs) which can then be solved by the multi-dimensional Newton's method. The produced set of samples are then interpolated with a B-spline curve to obtain continous, parametric representation of geodesics.

## 5. Triangle strip generation

The computed geodesics trim the parametric surface into strips $(S_1, S_2, \cdots, S_{n-1})$. Each strip $S_i$ is bounded by two parametric curves $c_i, c_{i+1}$ (either geodesics or boundary curves $e_2, e_4$) and two approximate polylines $P = \{p_1, \cdots, p_n\}$ and $Q = \{q_1, \cdots, q_m\}$ (within the given tolerance $\varepsilon$). Inspired by the recent work [5, 19], the minimal bending energy on all bridge edges (refer to as *MinBend*) can be used as an objective function for optimization. The bending energy related to $p_i q_j$ is calculated by the dihedral between two triangles adjacent to $p_i q_j$; here we denote it by $Bend_{xy}(p_i, q_j)$, where $x$ denotes the sample which forms the preceding triangle with $p_i, q_j$ and $y$ forms the succeeding one.

**Initial triangulation.** A greedy approach is proposed in [19] to find a locally optimal solution. On the contrary, we propose a dynamic-programming-based approach that guarantees output a globally optimal solution with the same constraints as in [19], producing improved and more stable results. Assume that $MinBend_x(p_i, q_j)$ is the minimal bending energy for sequences $p_1, \cdots, p_i$ and $q_1, \ldots, q_j$, with the last triangle having two samples on $x$ side ($x = p$ or $q$). The rule to compute a particular $MinBend$ is given as:

$$\begin{aligned} &MinBend_p(p_i, q_j) = \\ &\min\{MinBend_p(p_{i-1}, q_j) + Bend_{pp}(p_{i-1}, q_j), \\ &\quad MinBend_q(p_{i-1}, q_j) + Bend_{qp}(p_{i-1}, q_j)\} \end{aligned}$$

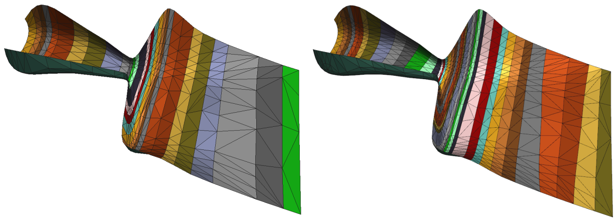

**Figure 1. Triangulation of the Ex1 model with $\varepsilon = 1.0cm$ (left) and $\varepsilon = 0.5cm$ (right).**

$$\begin{aligned} &MinBend_q(p_i, q_j) = \\ &\min\{MinBend_p(p_i, q_{j-1}) + Bend_{pq}(p_i, q_{j-1}), \\ &\quad MinBend_q(p_i, q_{j-1}) + Bend_{qq}(p_i, q_{j-1})\} \end{aligned}$$

The global minimum solution is then derived from $\min\{MinBend_p(p_n, q_m), MinBend_q(p_n, q_m)\}$ with back tracing.

**Adaptive triangle strip refinement.** Based on our motivation, generated triangle strip $T_i$ needs to further satisfy a prescribed tolerance $\varepsilon$ when compared to the original strip $S_i$. This is achieved by adaptive triangle strip refinement. The overall $L_\infty$ error is computed for the initial triangulation. If it is above $\varepsilon$, the strip with the largest $L_\infty$ error is picked out and subdivided by a new added geodesic connecting two points located at the midpoints of two boundary line segments. The two subdivided strips are then triangulated and checked again for the approximation error. The process repeats until the error is within the given tolerance $\varepsilon$.

**Triangle strip flattening.** Given the triangle strips, flattening them is straightforward. For each strip, the first triangle is flattened at some place in the $X-Y$ plane. Adjacent triangles are then flattened one by one along the bridge edges, as long as the flattened triangles do not overlap each other in the plane.

## 6 Experimental results

The first example **Ex1** is shown in Fig. 1. The largest extent of this model is $130cm$. Two error bounds $\varepsilon = 1.0cm$ and $\varepsilon = 0.5cm$ in $L_\infty$ metric are tested. The strip triangulation results are given in Fig. 1 and The flattened triangulation with $\varepsilon = 1.0cm$ is shown in Fig. 2. Another example **Ex2** is a swung surface which has iso-parameter curves closed in one direction (see Fig. 3). The largest extent from top to bottom is $63.9cm$. The surface is first cut by a shortest iso-curve from top to bottom, and then the Algorithm **DevAppr** is applied. Two results are produced, with error bounds $\varepsilon = 1.0cm$ and $\varepsilon = 0.5cm$, respectively, both in $L_\infty$ metric (see Fig. 3). Performance data of output triangle strips in the two models is summarized in Table 1. The experimental data shows that triangle strips within the given error bounds can be produced appropriately.

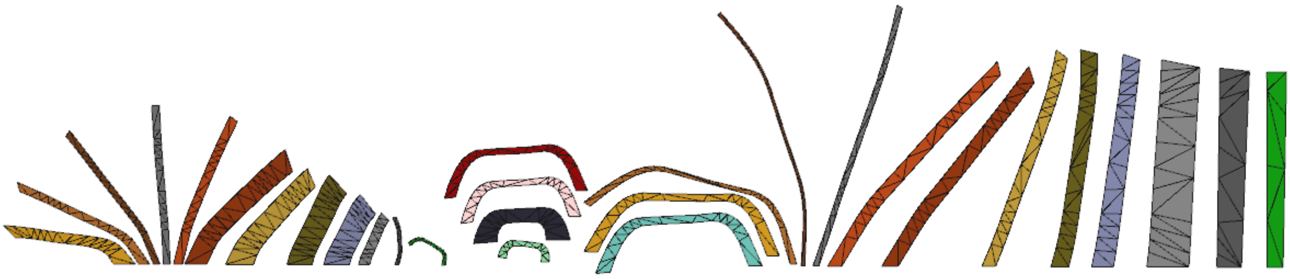

**Figure 2. Triangulation flattening of the Ex1 result in Fig. 1 (left).**

| example | $\varepsilon$ | final error | #. strips | #. triangles |
|---|---|---|---|---|
| **Ex1** | $1.0cm$ | $0.801cm$ | 29 | 681 |
| **Ex1** | $0.5cm$ | $0.477cm$ | 42 | 1410 |
| **Ex2** | $1.0cm$ | $0.823cm$ | 14 | 1575 |
| **Ex2** | $0.5cm$ | $0.381cm$ | 20 | 3251 |

**Table 1. Performance data of experiments with different error bounds.**

## 7 Conclusions

In this paper, a simple and efficient algorithm is proposed to approximate a general parametric surface using a minimum set of $C^0$-joint triangle strips. The approximation error to the given parametric surface can be well controlled by a user specified error bound $\varepsilon$; this property is often desirable in manufacturing process. Finally we conclude that previous solutions proposed in [4, 8] can be viewed as two special cases of our general solution: in [4] isolines are used for trimming while we use the more general geodesics; in [8] only surfaces of revolution are considered while we handle the general parametric surfaces.

## Acknowledgment

The work was supported by the National Natural Science Foundation of China (Project No. 60603085) and the National Basic Research Program of China (Project No. 2006CB303102).

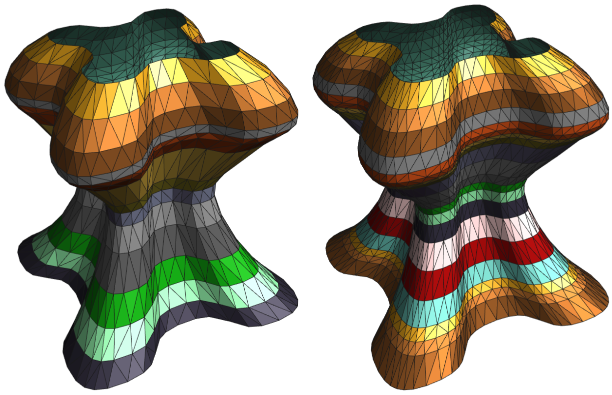

**Figure 3. Triangulation of the Ex2 model with $\varepsilon = 1.0cm$ (left) and $\varepsilon = 0.5cm$ (right).**

## References


[1] R. Bodduluri, B. Ravani. Geometric design and fabrication of developable surfaces, *ASME Adv. Design Autom.* **2** (1992), pp.243–250.

[2] R. Bodduluri, B. Ravani. Design of developable surfaces using duality between plane and point geometries, *Computer-Aided Design* **25** (1993), pp.621–632.

[3] H. Chen, I. Lee, S. Leopoldseder, H. Pottmann, T. Randrup, J. Wallner. On surface approximation using developable surfaces, *Graphical Models and Image Processing* **61** (1999), pp.110–124.

[4] G. Elber. Model fabrication using surface layout projection, *Computer-Aided Design* **27** (1995), pp.283–291.

[5] W. Frey. Boundary triangulations approximating developable surfaces that interpolate a close space curve, *International Journal of Foundations of Computer Science* **13** (2002), pp.285–302.

[6] W. Frey. Modeling buckled developable surface by triangulation, *Computer-Aided Design* **36** (2004), pp.299–313.

[7] J. Hoschek, H. Pottmann. Interpolation and approximation with developable B-spline surfaces, in Dahlen M, Lyche T and Schumaker LL, eds., *Mathematical Methods for Curves and Surfaces* (1995), pp.255–264.

[8] J. Hoschek. Approximation of surfaces of revolution by developable surfaces, *Computer-Aided Design* **30** (1998), pp.757–763.

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

[10] Y.J. Liu, K. Tang, A. Joneja. Modeling dynamic developable meshes by Hamilton principle, *Computer-Aided Design* doi:10.1016/j.cad.2007.02.013.

[11] G. Lukacs, A. Marshall, R. Martin. Faithful least squares fitting spheres, cylinders, cones, and tori for reliable segmentation, *Computer Vision - ECCV* (1998), pp.671–686.

[12] J. Mitani and H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics* (SIGGRAPH'04) **23** (2004) pp.259–263.

[13] H. Pottmann, G. Farin. Developable rational Bezier and B-spline surfaces, *Computer Aided Geometric Design* **12** (1995), pp.513–531.

[14] H. Pottmann, T. Randrup. Rotational and helical surface approximation for reverse engineering, *Computing* **60** (1998), pp.307–323.

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

[16] W. Press, S. Teukolsky, W. Vetterling, B. Flannery. *Numerical Recipes in C++*, Cambridge University Press, 2nd, 2002.

[17] T. Randrup. Approximation of surfaces by cylinders, *Computer-Aided Design* **30** (1998), pp.807–812.

[18] J. Subag, G. Elber. Piecewise developable surface approximation of general NURBS surfaces, with global error bounds, *GMP2006* **LNCS 4077** (2006), pp.143–156.

[19] K. Tang, C. Wang. Modeling developable folds on a strip, *Journal of Computing and Information Science in Engineering* **5** (2005), pp.35–47.