曲線と曲面の局所的なフェアリング・曲率最適化のアルゴリズム

東京大学　○サルビ・ペーテル、ハンガリー科学アカデミー　ヴァーラディ・タマーシュ、東京大学 鈴木宏正

**Local Fairing of Curves and Surfaces - Algorithms for Curvature Optimization**

P. Salvi (University of Tokyo), T. Várady (Hungarian Academy of Sciences), H. Suzuki (University of Tokyo)

*After reviewing different approaches, a new algorithm is presented for fairing B-spline curves and surfaces. It is based on a novel fairness measure, which is derived from a notion called "target curvature". The target curvature is computed from the not-yet-faired curve or surface automatically, but with optional user-interaction to make it flexible. The method itself is parameter invariant and local. We introduce two implementations, a slower, iterative method emphasizing locality and a faster algorithm involving discrete integration and fitting. The results are illustrated by a few examples.*

## 1. Introduction

Digital Shape Reconstruction (DSR) is a fast growing area in CAGD, which deals with the creation of geometric models using measured data. In many practical applications of DSR, surface fairness is a crucial matter, in particular in the automobile industry. Although fairness does not have an exact mathematical definition, researchers agree that the curvature of fair surfaces must be evenly distributed. A wide range of graphical *interrogation tools* has been developed to detect small surface artifacts, but even with these, fairing is a laborious manual process. This is why there is a natural need for (semi) automatic fairing algorithms that smooth the surface while preserving the highly curved features of the original shapes.

One widely used criterion of fairness is the smoothness and smooth distribution of reflection lines. If a fair object was placed into a room lit by parallel lights, the reflections of the light source should bend smoothly and evenly over the surface. Isophote lines are very similar to reflection lines, since their smoothness depends on the change of the first derivative of the surface. An isophote line is a set of surface points where the angle between the normal vector and the viewing direction is the same (within a given tolerance). Since this map is much simpler than the previous, but reveals just about the same flaws, it is more often seen in practice.

Isophote lines are an example of interrogation tools, because they help the user to find minor discontinuities or wiggles on the surface. Other tools include the curvature maps and the curvature combs, which depend on the second derivatives. Curvature maps color-code the curvature values and can have various types (Gaussian, mean, minimum, maximum, etc.), curvature combs display the values as orthogonal straight line segments along a curve.

In order to implement a fairing algorithm, it is common to define a fairness measure, i.e. a functional that represents the fairness of the surface. In other words, we can say that a surface $S$ is fair, if $F(S) < \tau$ applies, where $\tau$ is a user-defined tolerance.

We propose a new fairness measure, along with two algorithms that make use of it: a simple, iterational procedure focusing on locality and a fast, direct method involving integration and fitting.

## 2. Previous Work

One "classical" definition of fairness by Farin and Sapidis is as follows: *A curve is fair if its curvature plot is continuous and consists of only a few monotone pieces* [2]. But as Roulier and Rando point out, we cannot hope to have a universal fairness measure or algorithm. Still, we should strive to create new ones, in order to give designers the freedom of choosing the most suitable algorithms for their tasks [6].

### 2.1. Fairness Measures

A natural measure for curve fairness is the strain energy, which is based on a drawing technique used in ship design. To create a smooth curve, metal weights were placed at the interpolation points and a flexible spline was spanned between them. The resulting curve $c$ minimizes the strain energy, yielding the measure $E = \int (\kappa(s))^2 ds$, where $\kappa(s)$ is the curvature of the curve as a function of the arc length. This minimizes the mean curvature, while giving a penalty to the extreme values by squaring [6].

Computing the curvature can be difficult, so it is often replaced by a simpler, parameter-dependent formula $\hat{E} = \int (c''(t))^2 dt$. This has the drawback that in cases where the parameterization substantially differs from the arc-length parameterization, fairness is not guaranteed, and unexpected results may occur.

Moreton and Séquin introduced another measure [4], called Minimum Variation Curve, optimizing the variation of the curvature: $E_{MVC} = \int (\kappa'(s))^2 ds$. This has the advantage that it does not create unnecessary inflection points.

The curve-fairing measures introduced so far all have their surface-fairing equivalents. Similar to the strain energy, in the surface case we can minimize the thin plate energy, which also has a simpler variant that is parameter dependent and can only be used safely for surfaces with isometric parameterization.

Moreton and Séquin suggested an alternative measure based on the variation of the principal curvatures. It vanishes on spheres, cones and tori. Although this measure gives excellent results, it requires very complex computations.

### 2.2. Fairing Algorithms

One of the simplest, widely used curve-fairing methods is the *knot removal and reinsertion* (KRR), originally conceived by Kjellander and later made local by Sapidis and Farin [2].

An order $k$ B-spline is $C^{k-1}$ continuous at every point except the knot points, where it has only $C^{k-2}$ continuity. We can add knots to a B-spline in a way that its shape does not change. On the other hand, if we take out a knot, we can only preserve the shape if the curve was originally $C^{k-1}$ continuous at that knot point. The knot removal problem breaks down to an over-defined equation that can have several approximate solutions. Farin gives the most local solution for third degree B-splines [2], repositioning only one control point.

This gives the idea of the KRR algorithm, i.e. to find, remove and then reinsert the knot where the third derivative has the largest discontinuity, thus ensuring $C^3$ continuity at the knot. The process may be iterated until a suitable end condition is met. Finding such a condition is not a trivial task. A vast range of heuristics can be applied, including best-first-search [3] and simulated annealing [5].

Eck and Hadenfeld fix all but one control points and locally minimize the fairness measure $E_l = \int_{t_{k-1}}^{t_{k+1}} (c^{(l)}(t))^2 dt$, where $l$=2 or 3, while keeping the deviation from the original curve under a predefined tolerance by constraining the new control points to be in the vicinity of the original control polygon.

Both of these methods have equivalents in surface fairing. The main disadvantage of the KRR algorithm is that removing a knot changes a whole line of control points in the other parametric direction. Hahmann proves that it is sufficient to remove and reinsert a knot in only three rows or columns [3], but the generalized KRR ensures $C^3$ continuity in only one parametric direction, which is not satisfactory in real-life applications.

Hadenfeld proposed a fairing method using the thin plate energy metric [1]. As above, only one control point is moved at a time and the largest deviation is constrained from the original.

## 3. The New Algorithm

In this section we first sketch a fairing algorithm for curves, then we generalize it for surfaces. We can expect that the curvature comb of a fair curve is smooth, without any jumps or sudden changes. Therefore we can smooth the curve defined by the curvature comb's endpoints, which is practically the same as the evolute. We will call the smoothed curve the *target evolute*.

Now we want to find a curve that is close to the original, but whose

evolute is the target evolute. This also defines a fairness measure: the closer the evolute is to the target evolute, the fairer is the curve. Let $\rho$ denote the radius of the osculating circle, $n$ the normal and $e$ the target evolute, then our fairness measure will be $E = \int \|(c(t) + \rho n(t)) - e(t)\|^2 \, dt$, assuming that the two curves have a common parameterization. The algorithm for finding the minimum of this functional will be presented later. Controlling the deviation from the original curve can be managed by the convex hull property, as in the Eck-Hadenfeld algorithm.

Since the evolute (and the curvature comb) may be self-intersecting, we use directly the curvatures instead, so our measure becomes $\tilde{E} = \sum_i |\kappa(t_i) - g(t_i)|^2$, where $g$ is the target curvature.

In the surface case the curvature is replaced by the principal curvatures. Let $g_1$ and $g_2$ be the target curvatures based on $\kappa_1$ and $\kappa_2$, then $\tilde{\Pi} = \sum_i \sum_j \left( |\kappa_1(u_i, v_j) - g_1(u_i, v_j)|^2 + |\kappa_2(u_i, v_j) - g_2(u_i, v_j)|^2 \right)$ is a meaningful fairness measure.

### 3.1. Determining the Target Curvature

Any simple and fast smoothing method can be effectively used for defining the target curvature, e.g. averaging consecutive sampled points of the evolute. Global averaging can remove parts of the curvature that represent features, so the user should be allowed to restrict the smoothing or edit the target curvature manually.

Another possibility is to fit a NURBS curve over the sampled points. For surfaces this means fitting surfaces over the points of the target curvatures. To get smooth results, we should also minimize the curvature of the fitted surface.

### 3.2. An Iterative Algorithm

Our first algorithm is local in the sense that it moves one control point at a time. As most iterative methods, it has a drawback of speed, which we tried to counter by using the relatively fast downhill simplex method [5]. In every iteration we select a control point and move it to a position where the fairness measure is at a local minimum.

Selection of the next control point has great influence on the quality of fairness. Selecting the control point where the largest deviation of the target curvature occurs is a natural choice. However, this can lead to a deadlock, if the same control point is chosen over and over again. A list of the recently moved control points may be kept in order to avoid this. Also, boundary control points should not be selected for most applications.

### 3.3. Integration Method

The main idea is that the curvature of a curve is the same as the length of the second derivative in arc-length parameterization, so we can take the target curvatures as second derivatives and integrate twice to get the faired curve.

First we create a pseudo arc-length parameterization by sampling points at equal distances, then use a discrete integration method like the Euler or Runge-Kutta algorithms, starting from both sides of the curves, resulting in two sets of points. Starting point and normal data is extracted from the original curve. To control the deviation, we add an extra term to the integration equations that "pulls back" the curve towards the original. Finally we blend the two set together with a suitable blending function, and fit a curve over the blended points.

The algorithm can be extended to surfaces by first fairing $u$ and $v$ isocurves of the surface using the above integration method, then fitting a surface over the blended points.

## 4. Conclusions

Fairing curves and surfaces is a complex problem. Unfortunately, the goal of generating perfectly fair shapes cannot be unambiguously formulated with mathematical terms, and there are many alternatives. Authors propose a method where a smoothed target curvature function is approximated either by a local iterative algorithm or by a direct method involving discrete integration and fitting. Figures 1-4 show some results. Our future research will focus on fairing multiple surfaces together while preserving or enhancing the continuity between them.

## References


[1] M. Eck, J. Hadenfeld, *Local Energy Fairing of B-Spline Curves*. Computing Supplement 10, pp. 129–147, 1995.

[2] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*. A Practical Guide. Academic Press, 5th edition, 2002.

[3] S. Hahmann, S. Konz, *Knot-Removal Surface Fairing using Search Strategies*. Computer Aided Design 30, pp. 131–138, 1998.

[4] H. P. Moreton, C. H. Sequin, *Minimum Variation Curves and Surfaces for Computer-Aided Geometric Design*. In: N. S. Sapidis (Ed.), Designing Fair Curves and Surfaces, pp. 123–159, SIAM, ISBN 0-898-71332-3, 1994.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2nd Edition, ISBN 0-521-43108-5, 1992.

[6] J. Roulier, T. Rando, *Measures of Fairness for Curves and Surfaces*. In: N. S. Sapidis (Ed.), Designing Fair Curves and Surfaces, pp. 75–122, SIAM, ISBN 0-898-71332-3, 1994.


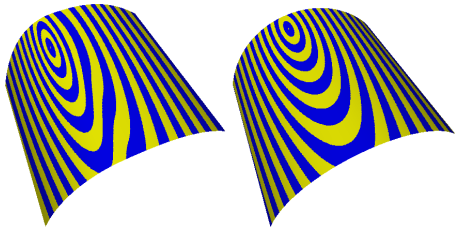

Figure 1. Isophotes of the extrusion surfaces of a curve before (left) and after (right) fairing by the iterative method.

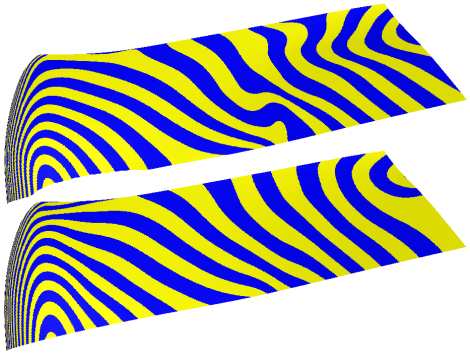

Figure 2. Isophotes of a Fiat body part before (top) and after (bottom) fairing using the iterative method.

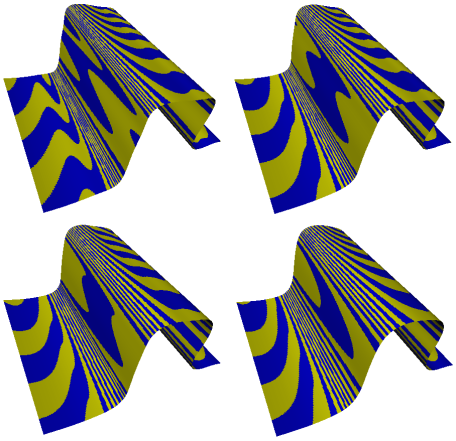

Figure 3. Isophotes of the extrusion surface of a curve before fairing (top left) and after fairing by the integration method with tight (top right), medium (bottom left) and loose (bottom right) tolerances.

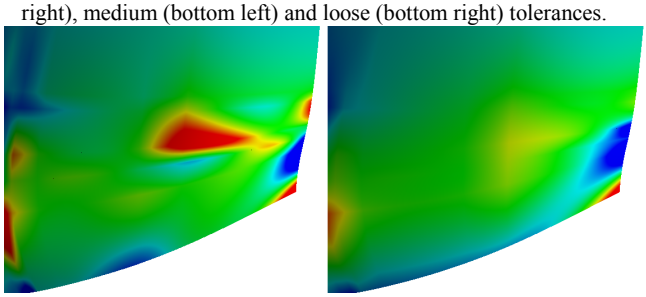

Figure 4. Mean map of a Fiat body part before (left) and after (right) local fairing on the center region using the integration method.