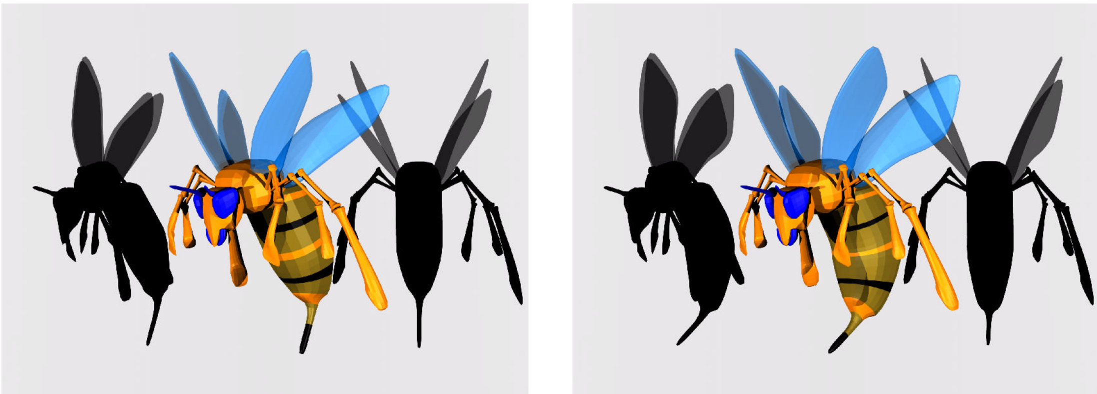
Fine Tuning: Curve and Surface Deformation by Scaling Derivatives

Kenjiro T. Miura
Shizuoka University
Department of Mechanical Engineering
Hamamatsu, Shizuoka, Japan
ktmiura@eng.shizuoka.ac.jp
Tel & Fax: 053-478-1074

Fuhua (Frank) Cheng
University of Kentucky, USA

Lazhu Wang
Raindrop Geomagic, Inc., USA



(a) Original surface (Doo-Sabin subdivision surface).

(b) Deformed surface (Non-uniform Catmull-Clark subdivision surface).

Figure 1. Example of subdivision surface fine tuning. The deformation process converts a Doo-Sabin subdivision surface to a non-uniform Catmull-Clark surface.

Abstract

A deformation-based fine tuning technique for parametric curves and surfaces is presented. A curve or surface is deformed by scaling its derivative, instead of manipulating its control points. Since only the norm of the derivative is adjusted, the resulting curve or surface keeps the basic shape of the original profile and curvature distribution. Therefore, the new technique is especially suitable for last minute fine tuning of the design process. Other advantages include: (1) the fine tuning process is a real local method, it can be performed on any portion of a curve or a surface, not just on a set of segments or patches; (2) by allowing a user to drag a scalar function to directly adjust the curvature (and, consequently, fairness) of a curve or surface, the new technique makes the shape design process more intuitive and effective; (3) the new technique is suitable for precise shaping and deforming such as making the curvature of a specific portion twice as big. In many cases, it can achieve results that other methods such as FFD can not; (4) the fine tuning process can also be used for subdivision curves and surfaces. Related techniques and test results are included.

Keywords: curve and surface deformation, subdivision curve and surface

1 Introduction

The generation of a smooth and fair curve or surface is an important and fundamental issue in computer graphics and computer aided geometric design. Smoothness is concerned with the functional and/or geometric continuity of a curve or surface (see, for example, [5]). Extensive research has been done in this area and various curve/surface generation methods have been proposed to ensure C^1 and C^2, or G^1 and G^2 continuity between adjacent curve segments or surface patches. Fairness is a more macroscopic concept than smoothness and its measure depends on fairness metrics. Many research works have been done to deal with fairness (cf.[14]). Nevertheless, it remains a difficult task to generate a fair curve or surface.

A major problem is the lack of fairness manipulation techniques. Curvature and variation of curvature are essential factors in determining the fairness of a curve or surface. Unfortunately, construction scheme of popular curve and surface representation schemes such as Bézier or B-spline, i.e., blending control points to generate curve or surface points, does not provide a user with direct manipulation capability of these quantities. Actually, since a simple relocation of the control points could change the curvature distribution dramatically, it is difficult to generate a fair curve or surface by directly manipulating their control points. The clothoid [6, 17] and QI [7] curves, on the other hand, have simple curvature profiles because the curves are defined by specifying their tangent vector directions. Hence, it is relatively easy to manipulate fairness of these curves. But, with limited shape range, these curves are not especially suitable for computer graphics applications.

A popular method in achieving fairness is to minimize a fairness functional of curvature or its variation to produce a fair free-form surface (Moreton and Séquin [8]). Such a method, however, tends to take large processing time and is not suitable for an interactive environment. Another problem with minimizing a fairness functional is the difficulty for a designer to perform local modification because of the global nature of the functional minimization process.

In the design process, it is desirable (sometimes necessary) to have a toolkit that, without changing the basic shape profile and curvature distribution of the object, allows the designer to slightly modify specific portions of the object to ensure a last minute fine tuning. One possible technique for this purpose is the so called free-form deformation (FFD) proposed by Sederberg and Parry [15]. FFD deforms an object by deforming a bounding volume defined by a 3D lattice of control vertices and a set of basis functions. This approach can easily control the shape profile and curvature distribution of the object by scaling the bounding volume as a whole, but it is not suitable for precise shaping and deforming, such as making the curvature of a specific portion of a curve twice as big. Other shortcomings of the FFD method include (1) the result may be unpredictable when the model is complex, and (2) it does not provide an analytic expression of the deformed model.

In this paper, we will address the above problems by proposing a deformation based *fine tuning* technique for parametric curves and surfaces. The deformation process is different from the previous approaches in that the deformation is performed by scaling the derivative of the curve or surface, instead of manipulating its control points, as shown in Figure 2. Since the scaling process changes the norm of the derivative only, the deformation fine tunes the curve or surface without changing the basic shape of its profile and curvature distribution. For instance, the red curve in Figure 2(a) is obtained by decreasing the norm of the green curve's derivative. Such a result can not be achieved by other deformation methods. The red curve in Figure 2(b) is similar to the red curve in (a) except it is also rotated and scaled to ensure the end points of the fine tuned curve remain the same as the original curve.

Other advantages of the new fine tuning method include: (1) the fine tuning process is a real local method, it can be performed on any portion of a curve or a surface, not just on a set of segments or patches; (2) by allowing a user to drag a scalar function to directly manipulate the curvature (and, consequently, fairness) of a curve or surface, the new technique makes the shape design process more intuitive and effective; (3) the new technique is suitable for precise shaping and deforming such as making the curvature of a specific portion twice as big. In many cases, it can achieve results that would not be possible using methods such as FFD; (4) the fine tuning techniques can also be used for subdivision surfaces as shown in Figure 1.

The remainder of this paper is organized as follows. The basics of the fine tuning technique for parametric curves is presented in Section 2. Fine tuning techniques for parametric surfaces are given in Section 3. The process of using the fine tuning technique on subdivision curves and surfaces is shown in Section 4. Concluding remarks and future research directions are given in Section 5.

(a) Original curve (green) and fine-tuned curve (red, the start point is fixed).

(b) Original curve (green) and fine-tuned curve (red, both end points are fixed).

Figure 2. Examples of curve fine tuning. The green curves are deformed by scaling their derivatives gradually to get the red curves.

2 Curve Fine Tuning

Basic ideas of the fine tuning technique for curves will be presented in this section. The technique can be used for all parametric curves. However, for simplicity of presentation, we will consider B-spline curves in this section only.

The idea of curve fine tuning is motivated by the following observation: given a curve $\boldsymbol{C}(t)$ with $0 \leq t \leq 1$, if one multiply the derivative $d\boldsymbol{C}(t)/dt$ by a scalar function $\alpha(t)$ and then integrate the product, one gets a curve $\boldsymbol{D}(t)$ with similar shape if the scalar function is close to 1. $\boldsymbol{D}(t)$ is defined as follows:

$$\begin{aligned} \boldsymbol{D}(t) &= \boldsymbol{C}(0) + \int_0^t \alpha(t) \frac{d\boldsymbol{C}(t)}{dt} dt \\ &= \boldsymbol{P}_0 + \int_0^t \boldsymbol{E}(t) dt. \end{aligned} \tag{1}$$

If $\boldsymbol{C}(t)$ and $\alpha(t)$ are both in the B-spline form, then the product $\alpha(t) d\boldsymbol{C}(t)/dt$ and $\boldsymbol{D}(t)$ are also B-spline curves.

The above equation also represents a deformation process of the curve driven by manipulating the derivative and, consequently, the curvature of the curve. For instance, if $\alpha(t)$ is set to 2 near a given parameter t_0, then the derivative of the curve at $t = t_0$ is scaled up by two and the curvature is reduced by one half. More generally, the curvature is changed by $1/\alpha(t)$ if the derivative is scaled by $\alpha(t)$. This claim is proved as follows: the curvature $\kappa(t)$ of $\boldsymbol{C}(t)$ is given by:

$$\kappa(t) = \frac{|\frac{d\boldsymbol{C}(t)}{dt} \times \frac{d^2\boldsymbol{C}(t)}{dt^2}|}{|\frac{d\boldsymbol{C}(t)}{dt}|^3}. \tag{2}$$

The first and second derivatives of $\boldsymbol{D}(t)$ are

$$\begin{aligned} \frac{d\boldsymbol{D}(t)}{dt} &= \alpha(t) \frac{d\boldsymbol{C}(t)}{dt}, \\ \frac{d^2\boldsymbol{D}(t)}{dt^2} &= \frac{d\alpha(t)}{dt} \frac{d\boldsymbol{C}(t)}{dt} + \alpha(t) \frac{d^2\boldsymbol{C}(t)}{dt^2}. \end{aligned} \tag{3}$$

Then the curvature $\kappa_D(t)$ of $\boldsymbol{D}(t)$ is given by:

$$\kappa_D(t) = \frac{|\alpha(t)\frac{d\boldsymbol{C}(t)}{dt} \times (\frac{d\alpha(t)}{dt}\frac{d\boldsymbol{C}(t)}{dt} + \alpha(t)\frac{d^2\boldsymbol{C}(t)}{dt^2})|}{|\alpha(t)\frac{d\boldsymbol{C}(t)}{dt}|^3}. \tag{4}$$

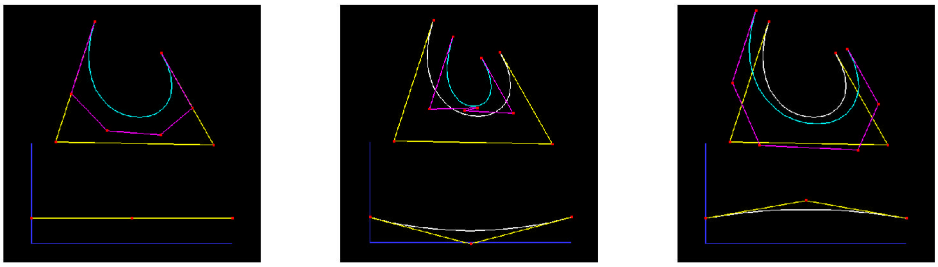
Since the cross product of two vectors oriented in the same direction is 0,

$$\kappa_D(t) = \frac{1}{\alpha(t)} \frac{|\frac{d\boldsymbol{C}(t)}{dt} \times \frac{d^2\boldsymbol{C}(t)}{dt^2}|}{|\frac{d\boldsymbol{C}(t)}{dt}|^3} = \frac{\kappa(t)}{\alpha(t)}. \quad (5)$$

The above equation proves the claim.

We call this type of shape manipulation technique *shape fine tuning* because, like fine tuning the pitch of a violin, shape manipulation through derivative scaling would gradually scale the curve and only induce minor change on the curvature distribution of the curve if the value of the scalar function does not change dramatically. This fine tuning process usually does not change the number of local extremes and would only change their locations slightly if the value of the scalar function does not exceed certain range.[1] Figure 3 shows several examples of the curve deformed by the fine tuning technique. If the scalar function is an identity function, the original and fine-tuned curves are identical as shown in Fig. 3(a). Figure 4 shows the curvature profile of these curves. The curvature of the fine-tuned curves are gradually scaled according to the scale function values as claimed above. Although the positions of the control points of the fine-tuned curve in Fig. 3(b) are strongly zigzaged, the curvature of the curve is only gradually scaled and it does not exhibit occurence of undulations.

A positive scalar function is usually used in the fine tuning process to avoid having a tangent in the opposite direction. For practical use of the fine tuning technique, a scalar function of degree $n-1$ is recommended for a curve of degree n. This is because a curve of degree n is C^{n-1} continuous. To maintain the same degree of continuity for the fine tuned curve, the scalar function $\alpha(t)$ should be at least C^{n-2} continuous, i.e. of degree $n-1$, as one can see from Eq.(1). Therefore, if the original curve is cubic, $\alpha(t)$ should be quadratic, and the fine-tuned curve is quintic. If the original curve is quadratic, $\alpha(t)$ should be linear and the fine-tuned curve is cubic. We use these combinations of degrees for all the examples in this paper, including the surface case.



(a) Original and fine-tuned curves are identical.

(b) Original (white) and fine-tuned (cyan) curves.

(c) Original (white) and fine-tuned (cyan) curves.

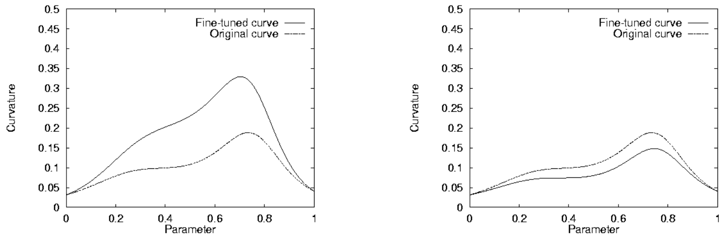
Figure 3. Examples of curve fine tuning. The white curves are deformed by scaling their derivatives gradually to get the cyan curves.

Implementation of the fine tuning technique for B-spline curves is straightforward except for the calculation of the product of a B-spline curve and a B-spline scalar function. The production can be regarded as a degree elevation process and implementation details can be found in standard textbooks on B-spline curves (see, for instance, [11]). A complete discussion on the product of two NURBS functions can be found in Morken [9]. Implementation of the fine tuning technique for other parametric curves is similar except clothoid curves whose derivatives can not be integrated analytically.

2.1 Boundary Constraints for Curves

The fine tuning technique preserves tangent direction of the curve at the endpoints. But it does not preserve the endpoints of the curve if the scalar function is not equal to 1. It is often necessary to keep the positions of the end points of the curve, especially to deform it locally. One simple method to satisfy the end-point constraint without changing the basic shape of the curvature profile is to apply a rotation and a scaling after the deformation

[1] This statement is rather qualitative. If $\alpha(t)$ is overly modified, some extreme values would vanish and new ones would be generated.

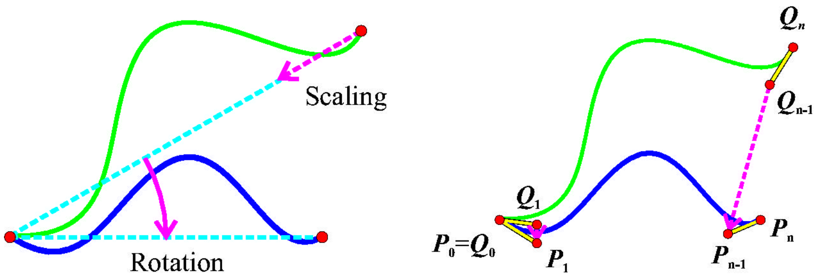


(a) Original curve (dashed) and fine-tuned curve (solid).

(b) Original curve (dashed) and fine-tuned curve (solid).

Figure 4. Curvature profiles.

process, as shown in Figure 5(a). The red curves in Figures 2(a) and 2(b) are obtained by fine tuning the given curve in green. The curve in (a) is deformed without constraint on the endpoints. The curve in (b) is deformed by the same scalar function, but also rotated and scaled to satisfy the end-point constraint. Hence the red curves in (a) and (b) are geometrically similar. Curvature profile is invariant under rotation and is only globally scaled by global scaling of the curve. Therefore none of these geometric transformations would cause undesirable undulations to the curvature profile.



(a) End-point constraint.

(b) Tangent constraint.

Figure 5. Rotation and Scaling. (a) End-point constraint. One simple method to hold the endpoints unchanged is to apply a rotation and a scaling after the fine tuning process. (b) Tangent constraint. The second control points Q_1 and Q_{n-1} from both endpoints Q_0 and Q_n of the fine-tuned curve are translated, rotated and scaled to match the corresponding control point P_1 and P_{n-1} of the invariant curve.

In addition to having fixed endpoints, sometimes it is also necessary to have fixed tangent and curvature at the endpoints. These constraints can be satisfied in a similar way as the end-point constraint. For the tangent constraint, first, multiply the given curve's derivative by an identity scalar function $\beta(t)$ of the same order and knot vector as $\alpha(t)$ and compute the control points of the fine-tuned curve (the blue curve in Figure 5(b)) which is the same as the original curve (but of higher degree). Then, apply $\alpha(t)$ to the original curve, and rotate and scale the fine tuned curve (the green curve in Figure 5(b)) so that the second control points on both ends also match the second control points on both ends of the above (blue) curve.

The curvature constraint is satisfied by matching both the second and third control points on both ends. Figure 6 shows examples of the end-point, tangent, and curvature constraints. Note that the more the constraints, the less freedom the deformation process.

The above method for the end-point constraint does not work for closed curves because an adequate scaling factor for the scaling process can not be determined for a closed curve. A different approach, based on a physical model described in Section 4 for recursive subdivision curves, has to be used.

2.2 Local Fine Tuning of a Curve

With the capability of fixing the endpoints, the tangents and the curvature of a curve segment at the endpoints, one can actually perform fine tuning on any portion of a parametric curve. For instance, to fine tune a portion