Modeling 3D Curves of Minimal Energy

Remco C. Veltkamp[‡] [*] and Wieger Wesselink[‡]

[‡] Technical University of Eindhoven, Department of Computing Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, e-mail: {remco,wieger}@win.tue.nl

[*] CWI, Department of Interactive Systems, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Abstract

Modeling a curve through minimizing its energy yields an overall smooth curve. A common way to model shape features is to perform the minimization subject to a number of interpolation constraints. This way of modeling is attractive because the designer is not bothered with the precise representation of the curve (e.g. control points). However, local shape specification by means of interpolation constraints is very limited. On the other hand, local deformation by repositioning control points is powerful but very laborious, and destroys the minimal energy property. In this paper, deform operators are introduced for 3D curve modeling that have built-in energy terms that have an intuitive effect. These operators allow local shape modification and do justice to the energy minimization way of modeling.

Keywords: Curve design, minimal energy, variational modeling.

1. Introduction

A curve that minimizes some suitable energy functional subject to a number of interpolation constraints is usually considered smooth. Modeling these minimal energy curves is a powerful way of modeling. It allows the design of smooth curves satisfying a number of interpolation constraints, in a very easy way. The associated energy functional typically depends on local properties of the curve such as the tangent vector and curvature, in which case it is called the internal energy. A well known example of such a functional is the bend energy. Because of its physical background, the bend energy is commonly used (see [1], [9], [10], [2], [3]). Other energy functionals with a physical interpretation have been studied as well [8]. Also, internal energy functionals can be developed that have some desired effect, see for example the metrics described in [17].

There are two basic ways of interaction in the design process of a minimal energy curve. Firstly, constraints can be imposed upon the curve. The most common constraints are point and tangent interpolation. They can be used to sketch a rough outline of a curve. Secondly, an internal energy functional can be chosen. With the choice of an internal energy the designer has a certain kind of global control over the shape of the curve.

These two ways of interaction are useful, but sometimes they are not flexible enough. Flexibility can be provided by operators that deform the curve locally. In the case of control point based curves (e.g. Bézier curves or B-spline curves) this can easily be done by repositioning a few control points. However, the deformed curve does in general not minimize an energy functional.

This paper introduces some design operators that do not suffer from this problem. The effect of such an operator is defined by an energy functional that has a desired deformation effect. Since this functional depends on properties from outside the curve, it is called external. The internal and external energies E_{int} and E_{ext} are combined into one functional, the total energy E_{tot} of the curve, which has to be minimized:

$$E_{tot} = E_{int} + E_{ext}. \tag{1}$$

One of the few examples of minimal energy curves with external energy are active contours in image processing [12] [22]. They are guided towards contours in an image, under the influence of 'image forces'. These image forces deform a curve and they are expressed by means of external energy terms. However, they depend on image data and are not used as a tool for interactive modeling. Another difference with our approach is that active contours are time-dependent. This results in an increase of complexity and therefore in most practical implementations the active contours are discrete curves. Another example of modeling minimal energy curves by means of external energy is given in [6], but only spring forces are used to shape the curve. By contrast, in this paper we will present a variety of shape operators modeled by external energy components, which are suitable for interactive design of 3D minimal energy curves.

We have tested the concepts in a prototype system for modeling B-spline curves. However, the idea is also applicable to Bézier curves or virtually any other class of curves. In order to improve the speed of the computation the energy functionals are composed of quadratic expressions. This results in a minimization problem with linear constraints, which is solved using a standard method. Experiments show that these design operators are useful tools for interactive modeling.

In the following three sections we will discuss various means to model minimal energy curves. Section 2 discusses the choice of internal energy. Section 3 is about modeling using constraints. Then, in section 4, we introduce shape operators that work by adding a suitable external energy term to the total energy functional. Section 5 discusses computational aspects. Some concluding remarks are made in section 6.

2. Internal energy

The internal energy of a curve is the part of the total energy (1) that depends only on properties of the curve itself. It determines the global shape of the curve. The internal energy that occurs most in the literature is the bend energy (E_{bend}):

$$E_{bend}(x) = \int \kappa^2(t)\|x'(t)\|\,dt, \tag{2}$$

with κ the curvature of the curve x (see [5]).

Experience has shown that this functional yields smooth curves, so it seems an appropriate choice. However, if the length of the curve is not restricted in any way, an absolute minimum of (2) need not exist (see [1]): in some cases the bend energy can be decreased by introducing large loops. In order to deal with this unwanted effect the length of the curve has to be restricted. A common way to do this is to combine the bend energy with another functional, the length of a curve. As this functional makes the curve resist stretching, it is called stretch energy ($E_{stretch}$):

$$E_{stretch}(x) = \int \|x'(t)\|\,dt. \tag{3}$$

For 3D curves, torsion describes the amount of twisting (planar curves have zero torsion). Twisting can be reduced by adding the following term E_{twist} to the internal energy:

$$E_{twist}(x) = \int \tau^2(t)\|x'(t)\|\,dt, \tag{4}$$

with τ the torsion of the curve (see [5]). Minimizing this twist energy restrains the curve at every point from coming out of a plane. The result is a curve that is as flat as possible.

Figure 1: Left: minimal energy curve with point interpolation constraints. Right: with additional tangent constraint.

For an interactive modeling program the evaluation and minimization of the expressions (2), (3), and (4) would be computationally too expensive. Therefore they are approximated with the following ones:

$$\bar{E}_{stretch}(x) = \int \|x'(t)\|^2 \, dt, \tag{5}$$

$$\bar{E}_{bend}(x) = \int \|x''(t)\|^2 \, dt, \tag{6}$$

$$\bar{E}_{twist}(x) = \int \|x'''(t)\|^2 \, dt. \tag{7}$$

These approximations are frequently used (see for instance [8]). The advantage of these approximations is that for e.g. B-spline curves they are quadratic functions of the control points (see section 5). This is probably the reason for their popularity, since quadratic functions can be efficiently minimized. A disadvantage of the approximations (5), (6), and (7) is that they are parameterization dependent. Generally speaking, the approximations will be worse when the absolute value of the derivative of the curve is fluctuating more. Note that for an arclength parameterized curve ($\|x'(t)\| \equiv 1$), the approximations would be exact.

The following three subsections define operators based on the stretch, bend, and twist energy that affect a part of the curve. In all the following energy terms, $f(t)$ is a weight function, which determines the strength of the operator along the curve.

2.1. Tightener

The tightener operator controls the tightness of a part of the curve by reducing the length of that part. This is done by adding the following term to the energy of the curve:

$$E_{tight}(x) = \int_v^w f(t)\|x'(t)\|^2 \, dt. \tag{8}$$

This is similar to the approximated stretch energy, but the strength (weight function) and the part of the curve to which the operator applies is determined by the user.

2.2. Straightener

The straightener operator reduces the bending of a part of the curve. Applying this operator is done by adding the following term to the energy of the curve:

$$E_{str}(x) = \int_v^w f(t)\|x''(t)\|^2 \, dt. \tag{9}$$

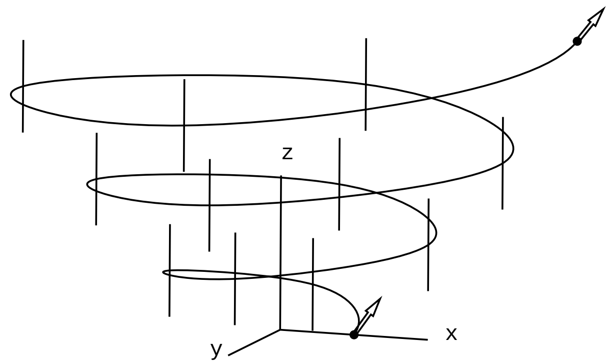

Figure 2: Minimal energy curve with point-on-line constraints.

This is essentially a weighted approximation to the bend energy, applied to only a part of the curve. The weight function and the affected curve segment is under user control, while the bend energy applies to the whole curve.

2.3. Flattener

The flattener operator flattens the curve, i.e. it tries to force the curve to lie in a plane. However, it does not prescribe *which* plane, in contrast to the plane attractor (see section 4.1).

An energy term that has this effect is:

$$E_{flat}(x) = \int_v^w f(t) \|x'''(t)\|^2 \, dt. \tag{10}$$

This is essentially a weighted approximation to the twist energy, applied to only a part of the curve. Similar to the operators above, the weight function and the affected curve segment is under user control, while the twist energy applies to the whole curve.

3. Constraints

Constraints can be used to sketch a rough outline of the curve. The most commonly used constraints are point and tangent interpolation. The point interpolation constraint $x(t_0) = p$ is illustrated in figure 1 (left), which shows a minimal energy curve interpolating prescribed points. In order to prevent the curve from collapsing to a single point in its strive to minimize its energy, at least two specified endpoints should be interpolated.

A tangent interpolation constraint $x'(t_0) = r$ forces the curve to attain a shape so as to have the prescribed tangent at some point of the curve. This is illustrated in figure 1 (right), which shows the left curve with an additional tangent constraint. Note that the length of the curve is increased somewhat.

In addition to the common point and tangent interpolation constraints we have designed a point-in-plane constraint $n \cdot x(t_0) = a$, which forces a point $x(t_0)$ of the curve to lie in the plane $n \cdot x = a$. A point-on-line constraint is the combination of two point-in-plane constraints, illustrated in figure 2. The (infinite) lines are denoted by line segments. The curve points $x(t_i)$ are free to slide along the lines so as to minimize the internal energy. Apart from the point-on-line constraints, there are two point and tangent interpolation constraints at the endpoints of the curve.

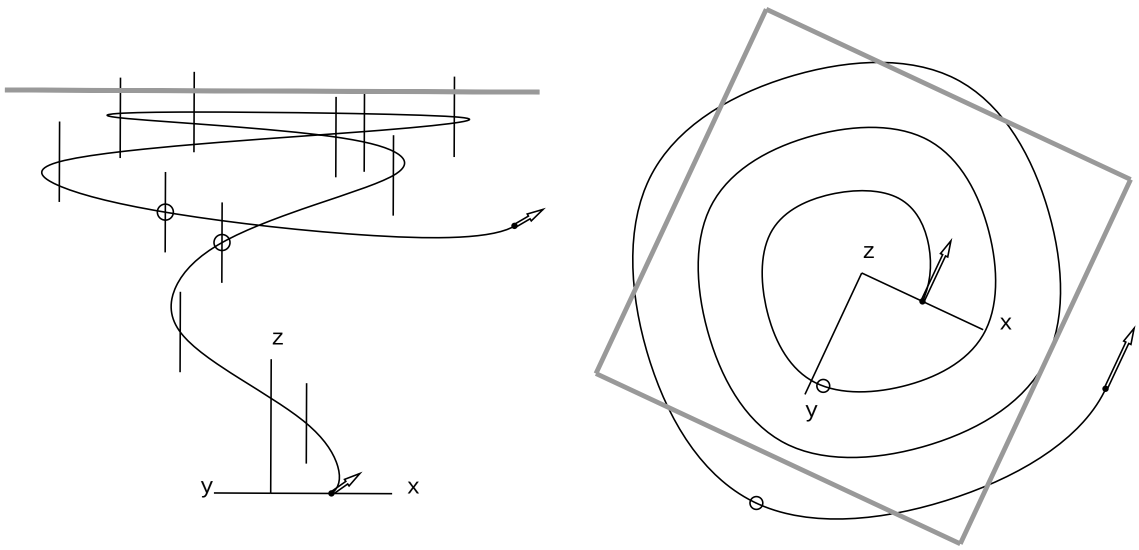

Figure 3: Plane attractor applied to the curve of figure 2.

4. External energy operators

We have designed a number of operators to add an energy term to the external energy functional: a plane attractor, a curve attractor, a point attractor, a director, a profiler, and a point repellor. These operators generate a corresponding energy term: E_{plat}, E_{cat}, E_{pat}, E_{dir}, E_{prof} and E_{prep} respectively. Again, in all the following energy terms, $f(t)$ is a weight function, which determines the strength of the operator along the curve.

4.1. Plane attractor

The plane attractor is an operator to pull a part of the curve towards a plane, as illustrated in figure 3. The figure shows the result of a plane attractor applied to the curve of figure 2, in two orientations. The part of the curve to which the attractor applies is indicated by the circles, and the (infinite) plane is represented by the square in grey.

Let the plane be given by the equation $n \cdot x = a$, where n is a unit vector normal to the plane, and a is a constant. For any point $x(t_0)$ of the curve, the quantity $(n \cdot x(t_0) - a)^2$ is the squared distance between that point on the curve and the plane. Therefore, minimizing the energy term:

$$E_{plat}(x) = \int_v^w f(t)(n \cdot x(t) - a)^2 \, dt \tag{11}$$

tries to pull the curve segment towards the plane. The minimal value of this expression is achieved when the curve segment between parameters v and w lies in the plane. However, other energy terms will often prevent this. For example, if the plane attractor will bend the curve, then the bend energy will increase. The bend energy component in the total energy of the curve will then compete with the energy of the plane attractor to obtain a minimum, and will often restrain the curve segment from lying completely in the plane.

4.2. Curve attractor

The curve attractor is a design operator that pulls the design curve towards another curve. We restrict ourselves to polynomial attractor curves (for instance line segments). This is illustrated in figure 4.

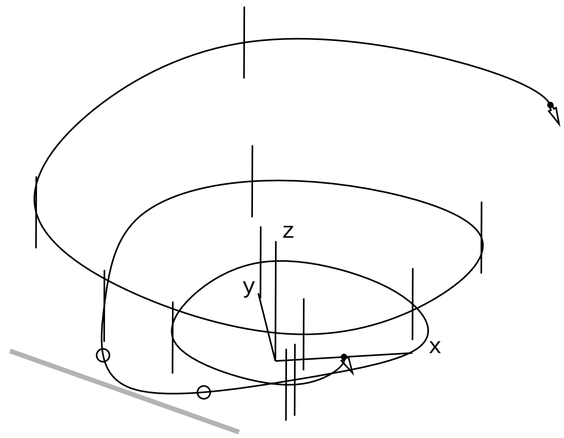

Figure 4: Minimal energy curve with a curve attractor applied.

The grey line is the attractor which is applied to the curve segment between the two circles.

Let ℓ be the attractor curve. The part of the curve x in the interval $[v, w]$ can be attracted to ℓ by an energy term of the following form:

$$E_{cat}(x) = \int_v^w f(t) \|x(t) - \ell(t)\|^2 \, dt. \tag{12}$$

Minimizing this expression amounts to minimizing the total deviation between the curve segment and the attractor curve. As for the plane attractor, this energy term has to compete with other terms, which may restrain the design curve from interpolating the attractor curve.

For a straight line as attractor curve, we can also model the attraction operator as the combination of two plane attractors. Given a straight line, we can construct two perpendicular planes through the line. The resulting attraction is an attraction to the intersection line, which has the advantage that the attractor does not depend on a parameterization of the line and the correspondence with the parameterization of the curve, as for the arbitrary curve attractor in (12).

4.3. Point attractor

The point attractor is a design operator that pulls the curve towards a point, as illustrated in figure 5. The grey dot is the point that attracts the curve; the point on the curve to which the attractor applies is graphically represented by a circle.

The point $x(t_0)$ of the curve can be attracted to point p by an energy term of the following form:

$$E_{pat}(x) = c\|x(t_0) - p\|^2, \tag{13}$$

with c a constant weight. The minimal value of this expression is achieved when the curve's point $x(t_0)$ and p coincide.

4.4. Director

The director is a design operator that tries to push the tangent along a segment of the curve in a specified direction, as illustrated in figure 6. The small circles on the curve identify the parameter interval $[v, w]$ on the curve to which the operator is applied. Let r be the prescribed unit direction vector. In any point on the curve the quantity $\|x' \times r\|^2$ measures the orthogonality between the

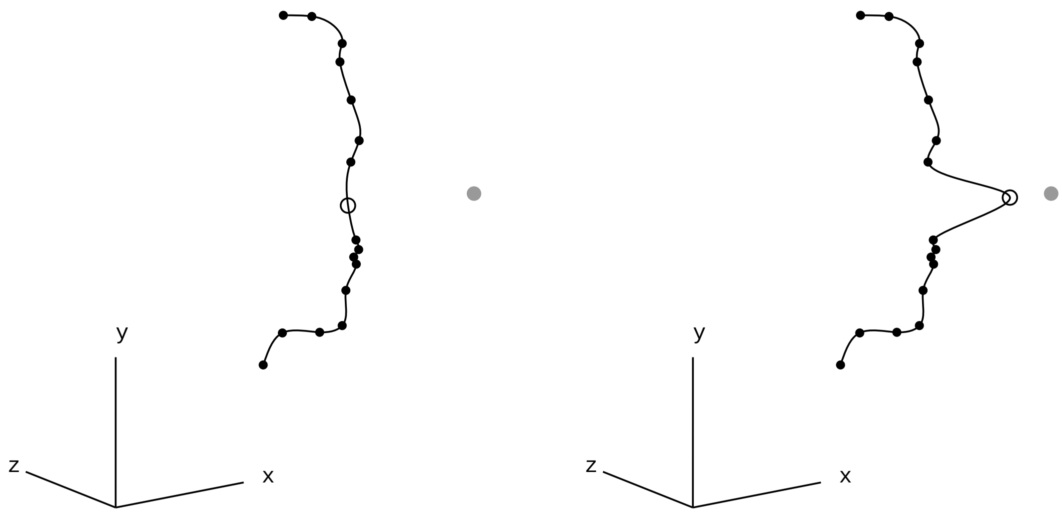

Figure 5: Left: curve with point attractor operator. Right: resulting minimal energy curve.

tangent vector x' at the point and the vector r. The smaller the deviation between prescribed and given tangent vector, the smaller the length of the cross product.

Therefore, a suitable energy term that tries to direct the tangent of the curve in the parameter interval $[v, w]$ into the specified direction is:

$$E_{dir}(x) = \int_v^w f(t) \|x'(t) \times r\|^2 \, dt. \tag{14}$$

Minimizing this expression amounts to minimizing the total deviation between the prescribed and given tangent along the curve segment.

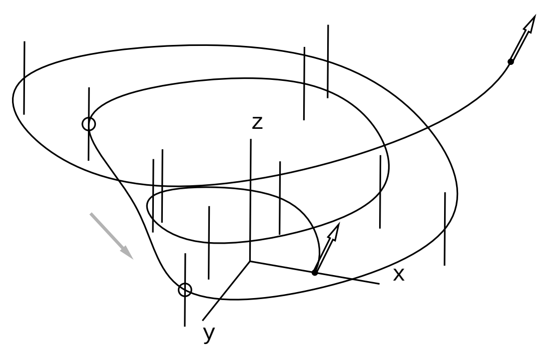

Figure 6: The direction $(2, 1, -1)$ imposed on the curve segment between the circles.

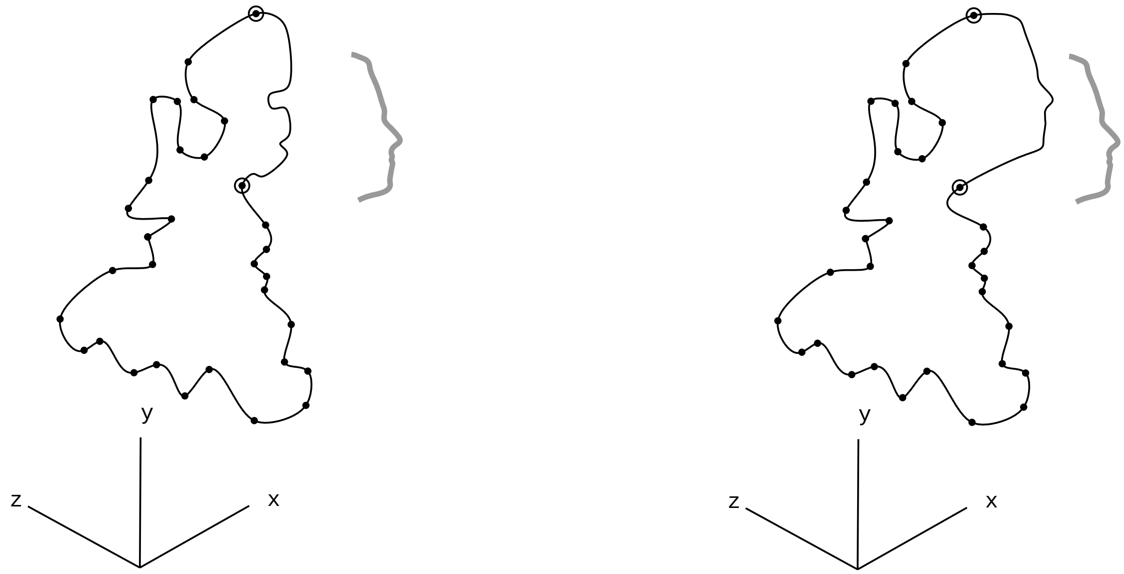

Figure 7: Left: design curve and the profile. Right: minimal energy curve after applying the profiler.

4.5. Profiler

When modeling a three dimensional curve, a designer is usually confronted with a two dimensional view of this curve, which is in fact a projection of the curve in a plane. A typical modeling operation is to modify this projection of the curve. The profiler is a modeling operator that has this effect. It attracts the projection of a portion of the curve in a given plane V towards another curve (a 'profile') in this plane. This is illustrated in figure 7. The profile is shown in grey, the part of the curve to which the operator applies is between the circles (which coincide with interpolation points).

The energy term E_{prof} corresponding to the profiler has the following form:

$$E_{prof}(x) = \int_v^w f(t)\|P(x) - c(t)\|^2 dt, \tag{15}$$

where $P(x)$ is the projection of x in V, and c is a given curve in V. The parameter interval $[v, w]$ defines the portion of the curve that is affected by the profiler.

4.6. Point repellor

In principle, the weight function of the attractor operators could be made negative in order to simulate repelling behavior. However, if the weight factor is too negative, the total energy of the curve is dominated by the negative term, and the curve tries to minimize its energy by shooting to infinity. To prevent this we have designed the repellor differently.

The point repellor is a modeling operator that can be compared with an electrical point charge in an electric field. It is a point in space that exerts a repelling force on a specific point of the curve. The force is parallel to the line between the two points and its magnitude decreases when the distance between the two points increases. (Note that for the attractors the effect increases with increasing distance.) The energy term E_{prep} for this operator is:

$$E_{prep}(x) = c\frac{1}{\|x(t_0) - p\|^2}, \tag{16}$$

where p is the repelling point, and c a constant weight factor.

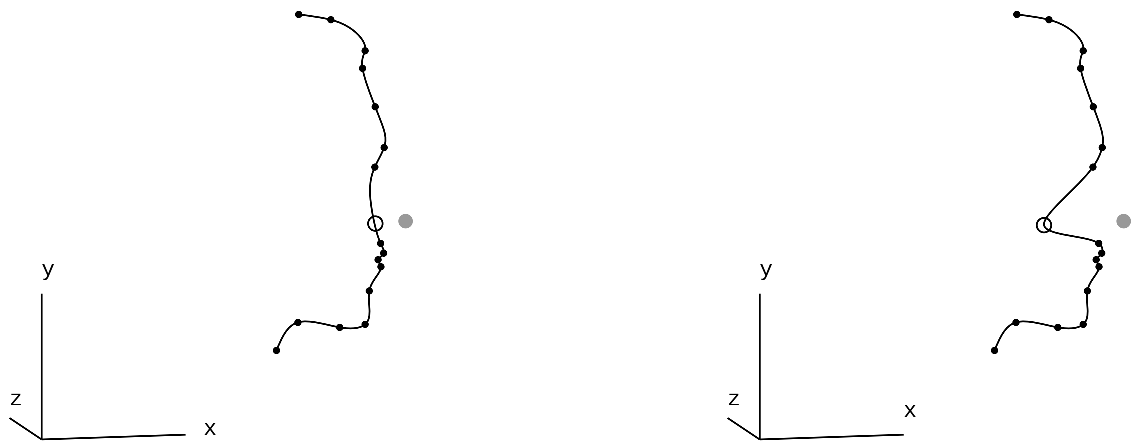

Figure 8: Left: curve with point repellor. Right: minimal energy curve after applying the repellor.

Most of the effect of this operator is that the point $x(t_0)$ slides along the curve to increase the distance to p, but the shape of the curve is hardly changed. We can prevent this behavior by combining the above energy term with the constraint that point $x(t_0)$ must remain on the line through p and $x(t_0)$. This point repellor is illustrated in figure 8.

Similarly, we can build a point repellor applying to a curve segment, with energy term

$$\frac{1}{\int_v^w f(t)\|x(t)-p\|^2}.$$

Just as we have the curve and plane attractor, we can construct a curve and plane repellor analogous to the point repellor.

5. Solving the optimization problem

We have implemented the constraints and energy operators in a prototype system for modeling uniform B-splines of minimal energy.

5.1. Curve representation

A uniform B-spline curve of degree d with knot sequence $\{0,\ldots,m+2d-2\}$ and control points $\{P_0,\ldots,P_{m+d-1}\}$ is defined as (see [7]):

$$x(t) = \sum_{i=0}^{m+d-1} P_i N_i^d(t), \tag{17}$$

where the basis functions N_i^d are defined by the de Boor recursion relations:

$$\begin{aligned} N_i^d(t) &= \frac{t-i+1}{d} N_i^{d-1}(t) + \frac{i+d-t}{d} N_{i+1}^{d-1}(t), \\ N_i^0(t) &= \begin{cases} 1 & \text{if} \quad i-1 \le t < i, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The curve is defined over the domain $[d-1, m+d-1]$. m is the number of curve segments if all knots in the domain are simple (do not coalesce).

In the rest of this section, the vector that contains the components of the control points of a B-spline curve is denoted as $\mathcal{P}$:

$$\mathcal{P} = [P_{0x} \dots P_{(m+d-1)x} \; P_{0y} \dots P_{(m+d-1)y} \; P_{0z} \dots P_{(m+d-1)z}]^T. \tag{18}$$

The subscripts x, y and z stand for the x, y and z-coordinates respectively. The dimension of $\mathcal{P}$ will be denoted as g (so $g = 3m + 3d$). In our implementation we have used cubic B-spline curves ($d = 3$).

5.2. Constraint representation

In case of a curve in B-spline representation, a point interpolation constraint $x(t_0) = p$ takes the following form :

$$\sum_{i=0}^{m+d-1} P_i N_i^d(t_0) = p.$$

Hence a point interpolation constraint consists of three equations $C^T \mathcal{P} = q$, one for the each coordinate, where the C's are suitably chosen g-vectors and $\mathcal{P}$ is the vector defined in (18). Note that these equations are linear in $\mathcal{P}$, if t_0 is fixed. The value of t_0 is determined automatically as the value corresponding to the point of the curve closest to p.

A tangent constraint $x'(t_0) = r$ takes the following form:

$$\sum_{i=0}^{m+d-1} P_i \dot{N}_i^d(t_0) = r.$$

If t_0 is fixed, the tangent constraint can be written in the form of three linear constraints $C^T \mathcal{P} = q$, one for the each coordinate, with the C's suitably chosen g-vectors. In practice we use the tangent constraint in combination with a point interpolation constraint, which determines the value of t_0.

A point-in-plane constraint $n \cdot x(t_0) = a$ expands into

$$\sum_{i=0}^{m+d-1} (n \cdot P_i) N_i^d(t_0) = a.$$

For a user-specified value of t_0, this equation is linear in the control points and can be written as a single vector equation $C^T \mathcal{P} = a$. The point-on-line constraint is a combination of two point-in-plane constraints. (The point interpolation is a combination of three point-in-plane constraint, but more efficiently represented as done above.)

5.3. Energy representation

The energy functionals can be conveniently expressed in the vector $\mathcal{P}$. First it will be shown that all functionals except for the point repellor are quadratic functions of $\mathcal{P}$ and then the integrals that actually have to be computed will be reviewed.

For the internal energy we use a convex combination of the approximated stretch, bend, and twist energy:

$$E_{int}(x) = \alpha_1 \bar{E}_{stretch}(x) + \alpha_2 \bar{E}_{bend}(x) + \alpha_3 \bar{E}_{twist}, \tag{19}$$

for some $\alpha_1, \alpha_2, \alpha_3$ in the interval $[0, 1]$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Added to this are the energy terms from the applied tighten, straighten, and flatten operators.

Substitution of the B-spline representation (17) into the bend energy functional (6) gives the following expression:

$$\bar{E}_{bend}(x) \quad = \quad \int_{t_{d-1}}^{t_{m+d-1}} \|x''(t)\|^2 \, dt$$

$$= \sum_{i=0}^{m+d-1} \sum_{j=0}^{m+d-1} P_i \cdot P_j \int_{t_{d-1}}^{t_{m+d-1}} \ddot{N}_i^d(t) \ddot{N}_j^d(t)\, dt.$$

This can be written in matrix form as $\bar{E}_{bend}(x) = \mathcal{P}^T A_{bend} \mathcal{P}$, which is quadratic in $\mathcal{P}$. The elements of the $g \times g$-matrix A_{bend} contain integrals of products of B-spline basis functions.

Substitution of the B-spline representation into the plane attractor energy functional (11) gives

$$E_{plat}(x) = \int_v^w f(t)(n \cdot x(t) - a)^2\, dt$$

$$= \sum_{i=0}^{m+d-1} \sum_{j=0}^{m+d-1} (P_i \cdot n)(P_j \cdot n) \int_v^w f(t) N_i^d(t) N_j^d(t) - 2a \sum_{i=0}^{m+d-1} (P_i \cdot n) \int_v^w N_i^d(t_0)\, dt + a^2.$$

This can be written as $E_{plat}(x) = \mathcal{P}^T A_{plat} \mathcal{P} + B_{plat}^T \mathcal{P} + c_{plat}$, with A_{plat} a $g \times g$-matrix, and B_{plat} a g-vector. Similarly, substitution of the B-spline representation into all the other energy functionals except for the point repellor gives expressions that can be written in the form

$$E(x) = \mathcal{P}^T A \mathcal{P} + B^T \mathcal{P} + c. \tag{20}$$

The point repellor is a special case: its energy functional is not quadratic, but the reciprocal of a quadratic function, which is easily verified by comparing the energy functional of the point repellor (16) with the energy functional (13) of the point attractor. Substitution of the B-spline representation into E_{prep} gives an expression of the form

$$\frac{1}{\mathcal{P}^T A_{prep} \mathcal{P} + 2B_{prep}^T \mathcal{P} + c_{prep}}.$$

Now that we have seen that the energy functionals can be written as (reciprocals of) quadratic functions in $\mathcal{P}$, it's time to take a closer look at the matrices A and vectors B that constitute these functions. We restrict ourselves to piecewise constant weight functions f (step functions), and the curve ℓ in E_{cat} is polynomial. With this in mind, a careful examination of the energy functionals learns that the computation of the matrices and vectors involves the computation of the following primitive functions:

$$\begin{aligned}
&N_i^d(t), \\
&N_i^d(t) N_j^d(t), \\
&\int \dot{N}_i^d(t) \dot{N}_j^d(t)\, dt, \\
&\int \ddot{N}_i^d(t) \ddot{N}_j^d(t)\, dt, \\
&\int N_i^d(t) N_j^d(t)\, dt, \\
&\int t^k N_i^d(t)\, dt, \qquad k = 0, \ldots, \text{degree}(\ell) + 1,
\end{aligned} \tag{21}$$

for $0 \le i, j \le m + d - 1$. Since the B-spline basis functions N_i^d are piecewise polynomial, these primitive functions have to be computed for individual intervals $[k, k+1]$.

A basis function N_i^d is in the interval $[k, k+1]$ a degree d polynomial defined on the knots $\{k - d + 1, \ldots, k + d\}$. We have computed these coefficients with Mathematica [23] for the case $d = 3$

(cubic curves). They consist of large sums of terms, each of them depending on the knots. The integrals (21) only have to be computed for $0 \leq i, j \leq 3$, because of the following property of uniform B-spline functions:

$$N_i^d(t) = N_{i-k}^d(t - kh), \qquad k \in \mathbb{N}, k \leq i,$$

for some h. All these polynomials have to be computed only once. Once they are stored, the computation of the matrices A and vectors B is nothing more than substitution of the parameter values v and w in these polynomials. See for more details [21], which describes the case for planar non-uniform B-splines.

5.4. Calculation

In the previous subsection we have seen that all the energy functionals that we use are (reciprocals of) quadratic functions of the control points of a curve. Therefore the total energy E_{tot}, which is a linear combination of these functionals, can be written as a function in $\mathcal{P}$ of the following form:

$$E_{tot}(\mathcal{P}) = \mathcal{P}^T A \mathcal{P} + 2B^T \mathcal{P} + c + \sum_i \frac{1}{\mathcal{P}^T A_i \mathcal{P} + 2B_i^T \mathcal{P} + c_i} \tag{22}$$

where the A's are $g \times g$-matrices, the B's are g-vectors, and the c's are numbers.

We have also seen that all constraints are linear in $\mathcal{P}$. So we have a minimization problem with linear constraints of the following form:

$$\text{Minimize} \quad E_{tot}(\mathcal{P}) \tag{23}$$

$$\text{subject to} \quad C\mathcal{P} = D, \tag{24}$$

where C is an $h \times g$-matrix, and D is an h-vector, with h the number of constraints.

We solve this system in two steps. First the constraint equations (24) are used to eliminate some variables from the minimization problem (23). This results in an unconstrained minimization problem with an object function of the same form as (22). This unconstrained minimization problem is solved using the conjugate gradient method, described in [13].

6. Concluding remarks

We have introduced a way of modeling minimal energy curves by means of operators that affect the total energy of the curve. These operators effectively control groups of control points at once. The tightener, straightener, flattener, and director directly influence the local behavior of the curve, the other operators directly influence the position of a part of the curve. In this way we can deform the curve locally, while the energy minimizing principles remain satisfied. By contrast, the internal energy of a curve always has a global effect on its shape. This gives additional flexibility over the standard way of minimal energy curve modeling (with constraints and internal energy only). The effectiveness for planar curves has been demonstrated in [21]. Experiments show that these operators offer an intuitive way of modeling.

In the current paper we have extended the set of operators. Some of the extra operators are also applicable to planar curves, some are only sensible for 3D curves. The computational aspects of the implementation have been described in section 5. The implemented system has no satisfactory user interface for intuitive interaction. This nontrivial aspect is part of future work.

The same way of modeling applies to surfaces. Modeling is again commonly done by means of constraints and by choosing the internal energy functional [16] [4] [14] [20] [11]. The use of external energy components is again limited to spring forces [18]. We have designed a number of more flexible operators that add external energy terms [19].

One may wonder what the actual physical meaning is of the constructed total energy of a curve. The bend energy component E_{bend} has a physical justification, although it is an approximation. The external energy components are designed heuristically so as to have the desired effect. The composition of the internal and external energy functionals is a rather arbitrary weighted sum of the individual components.

The modeling by means of external energy functionals also has its limitations. When the weight factor of an external energy functional is increased to a very high level, the corresponding part of the curve will almost exactly meet the specifications of the operator. But the neighborhood of this segment of the curve can be heavily disturbed when the energy in this portion of the curve dominates the total energy of the curve so much that the rest is neglected. Then the neighborhood of this portion will do anything to decrease the energy in this portion, even if this causes them to oscillate wildly.

When the user applies an external energy operator to a segment of the curve between certain parameter values, the lengths of the curve segment corresponding to the parameter range before and after minimization are generally not the same. This is caused by the fact that the curves are not arclength parameterized.

To alleviate these two problems we can use hierarchical spline curves. A hierarchical spline is the sum of a base curve and additional spline segments. The segments added to the base curve are typically used to model local detail. Each external energy operators could add a new spline segment to the hierarchical spline to provide the local refinement. However, this would make the final curve dependent of the order in which the operators are applied. In our current concept, the operators have no order dependent effect, since they affect one total energy functional for the whole curve.

References

[1] Garret Birkhoff and Carl R. de Boor. *Piecewise Polynomial Interpolation and Approximation*, pages 164–190. Elsevier, 1965.

[2] Guido Brunnett, Hans Hagen, and Paolo Santarelli. Variational design of curves and surfaces. *Surveys on Mathematics for Industry*, 3(1):1–27, 1993.

[3] Guido Brunnett and Johannes Kiefer. Interpolation with minimal-energy splines. *Computer Aided Design*, 26(2):137–144, February 1994.

[4] George Celniker and Dave Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics*, 25(4):257–266, July 1991.

[5] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[6] Lian Fang and David C. Gossard. Fitting 3D curves to unorganized data points using deformable curves. In Tosiyasu L. Kunii, editor, *Visual Computing*, pages 535–543. Springer-Verlag, 1992.

[7] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 3rd edition, 1993.

[8] Hans Hagen and G. Schulze. Variational principles in curve and surface design. In Hans Hagen and Dieter Roller, editors, *Geometric Modelling–Methods and Applications*. Springer-Verlag, 1991.

[9] B. K. P. Horn. The curve of least energy. *ACM Transactions on Mathematical Software*, 9(4):441–460, December 1983.

[10] Emery D. Jou and Weimin Han. Minimal-energy splines I: Plane curves with angle constraints. *Mathematical Methods in the Applied Sciences*, 13:351–372, 1990.

[11] Michael Kallay. Constrained optimization in surface design. In *Proceedings IFIP TC5/WG5.10 Working Conference on Modeling in Computer Graphics, Methods and Applications*, 1993.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[13] David G. Luenberger. *Linear and Non-Linear Programming*. Addison-Wesley, 2nd edition, 1989.

[14] Henry Moreton and Carlo H. Séquin. Functional optimization for fair surface design. *Computer Graphics*, 26(2), July 1992.

[15] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 1988.

[16] T. Rando and J. A. Roulier. Designing faired parametric surfaces. *Computer Aided Design*, 23(7):492–497, September 1991.

[17] John Roulier, Thomas Rando, and Bruce Piper. Fairness and monotone curvature. In Charles K. Chui, editor, *Approximation Theory and Functional Analysis*, pages 177–199. Academic Press, 1991.

[18] Demetri Terzopoulos and Hong Qin. Dynamic NURBS with geometric constraints and for interactive sculpting. *Transactions on Graphics*, 13(2):103–136, April 1994.

[19] Remco C. Veltkamp and Wieger Wesselink. Interactive variational spline surface design. *Technical Report*, Technical University of Eindhoven, 1995.

[20] William Welch and Andrew Witkin. Variational surface modeling. *ACM Computer Graphics*, 26(2):157–166, July 1992.

[21] Wieger Wesselink and Remco C. Veltkamp. Interactive design of constrained variational curves. *Computer Aided Geometric Design* (accepted for publication), 1995.

[22] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature information. *CVGIP: Image Understanding*, 1(55):14–26, 1992.

[23] S. Wolfram. *Mathematica: A System for Doing Mathematics by Computer*. Addison Wesley, 1991.