Length Constrained Multiresolution Deformation for Surface Wrinkling

Basile Sauvage
LMC-IMAG
Grenoble, France
Basile.Sauvage@imag.fr

Stefanie Hahmann
LMC-IMAG
Grenoble, France
Stefanie.Hahmann@imag.fr

Georges-Pierre Bonneau
GRAVIR-IMAG
Grenoble, France
Georges-Pierre.Bonneau@imag.fr

Abstract

We present a method for deforming piecewise linear 3D curves with constant length constraint. We show how this constraint can be integrated into a multiresolution editing tool allowing an intuitive control of the deformation's extent and aspect. The constraint is enforced following two steps. A first step consists in approximating the initial length by modifying the multiresolution decomposition at some specified scale. In a second step the constraint is exactly enforced by constrained minimization of a smoothness criterion. This process then provides the core of an integrated wrinkling tool for soft tissues modelling. A curve on the mesh is deformed, providing a deformation profile which is propagated in a user-defined neighbourhood on the surface.

1. Introduction

Multiresolution (MR) analysis has received considerable attention in recent years in many fields of geometric modeling, computer graphics and visualization. It provides an efficient representation of complex functions at multiple levels of detail. Thus convenient handling of geometrical objects is possible. General multiresolution editing or deformation techniques for parametric curves have been explored in detail by Finkelstein and Salesin [13], Gortler and Cohen [15], Elber and Gotsman [11] using B-splines. In contrast, *constrained* multiresolution editing techniques have not been explored so much in the past. However, there are many application areas, including CAD/CAM, computer graphics and computer animation, where deformations under constraints are needed. Constraint enforcement offers additional control of a shape during the modeling process. Constraining *linear geometric properties*, like position, normal and point tangent combined with minimizing curve or surface energy [14, 6, 34] proves to be an effective tool both for sculpting models and for animating real behaviours of objects. Some other works deal with *non-linear constraints*, e.g. length preserving deformation of Bézier curves [27], prescribing the length of rational Bézier curves [29], and volume preserving deformation of solids [28] and implicit surfaces [8]. None of these works integrate constraints into multiresolution editing. Only recently, MR editing of closed planar curves, coupled with the constraint of enclosed area preservation has been developed [10, 19]. In [5] a specific MR representation is proposed for volume preserving surfaces. It is based on the encoding volume elements between levels instead of basis function coefficients. While area and volume are bi-linear and tri-linear functionals, the length constraint in contrary is non-linear. This is the reason why for example the MR area preserving methods [10, 19] can't be directly adapted to the length. One of the purposes of the present paper is therefore to show how the non-linear constraint of length preservation can be incorporated into an MR editing tool.

Furthermore, the length constraint as well as constant area or constant volume belongs to the traditional principles of animation [22]. For example in computer animation of soft objects, where realistic deformations are required, realism can be achieved thanks to physical-based models [33, 7]. However, when low computation time has priority, typically for real-time animations, it is quite advantageous to mimic physical laws by some appropriated geometric properties. Area or volume preserving deformations can simulate inelastic material [28]. Length preserving deformation can mimic nearly non-compressible or non-stretch material like cloth [1, 9] or soft tissues. A characteristic behaviour of soft tissues is to form wrinkles. Wrinkles appear and disappear in order to compensate length changes. Therefore the second goal of the present paper is to handle length preserving curves specifically in order to control the generation of wrinkles through the MR representation when two curve points get closer.

Several models dedicated to wrinkle generation have been developped in the domain of computer graphics and computer animation. The first work [2] introduced bump mapping, that have often been used later on. The main drawback of bump mapping is that it doesn't modify the geometry of the object. Later works [37, 3] deal with *static* wrinkles for skin modelling which is significantly different from the the problem of *dynamic* wrinkles this paper focuses on. Modelling dynamic wrinkles with geometric deformations has been handled in [35, 37, 38] but all these methods require the ridges to be drawn by the user.

In [17] Hadap *et al* developped a method for simulating dynamic wrinkles on meshes. A displacement map is built in order to preserve the surface area: the underlying deformation of a coarse mesh is used for modulating predefined wrinkle patterns. Though it is convincing for cloth rendering it suffers drawbacks for a general use: the patterns have to be drawn by the user, and the geometry is modified through the displacement mapping extra layer.

Jing *et al* [20] presented a wrinkling tool for footwear design. It is based on the propagation of a wrinkling profile defined on the border of nurbs patches. The main drawback, relatively to our aims, is a tesselation step that prevents this method from being used for dynamic wrinkles. Moreover, only one single deformation is possible because the initial geometry is lost.

Larboulette and Cani [21] based their method on the propagation of a 2-dimensional profile curve, allowing wrinkles at different scales. But the profile is independant of the mesh's geometry, leading to very regular deformations. Moreover the propagation needs to project the mesh on a plane, hindering the use on high-curvature regions.

The contribution of this paper is to provide a multiresolution editing tool for 3D piecewise linear curves which allows to satisfy the non-linear constraint of length preservation. Beside general deformations through select-and-drag control points, the present paper focuses on a particular type of deformation: the generation of wrinkles in case of compressing the curve or part of it. The approach is purely geometric in the sense that the scale and frequency of the generated wrinkles are not controlled by some physical law, but by modifying curve coefficients at different multiresolution levels. Then we show how it can be used for general surface deformation through the extraction of a profile curve on the surface. Our algorithm is then applied to this curve. Eventually surface wrinkles are created by propagating and by attenuating smoothly the curve wrinkles on both sides of the curve.

The paper is organized as follows. Section 2 sets the basic tools and details the length preserving deformation method for 3D curves. In Section 3 the application to surface wrinkling is presented. Eventually, Section 4 concludes and gives possible future works.

2. 3D curves

In a classical MR editing environment the user chooses a resolution level at which the curve is modified by displacing one coarse control point. The shape of the curve changes more or less locally depending on whether a low or a high resolution level has been chosen.

The same procedure can be applied when integrating the additional constraint of keeping the curve length constant during deformation. The contribution of the present section is a method for solving the length constraint in a MR editing environment. It is in particular used for creating wrinkles whose frequency can be controlled by the user through the MR representation of the curve. This method is a generalization of the 2D method presented in [30], with a specific solution of the 3D orientation problems.

In a first time we present the length measure and constraints in Section 2.1. It is followed in Section 2.2 by an overview of the method whose two main steps are detailed in Sections 2.3 and 2.4.

2.1. Curve length

Let $\mathbf{c}(t) = (x(t), y(t), z(t))$ be a parametric curve lying in a space having a multiresolution analysis (see Annexe A for details). The length of $\mathbf{c}(t)$ is given by $L = \int \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2 dt}$. The curves the present paper is dealing with are continuous and piecewise linear. In that case the length simplifies to

$$L = \sum_{i=0}^{N-2} ||\mathbf{c}_{i+1}^n - \mathbf{c}_i^n||_2,$$

where $(\mathbf{c}_i^n)_{i=0\cdots N-1} = (x_i^n, y_i^n, z_i^n)_i$ denote the control points at the finest level n following the notations of A.

For the multiresolution representation of piecewise linear curves the scheme based on the Lazy wavelets [32] (see Fig. 2) is used:

$$\left\{ \begin{array}{ll} \mathbf{c}_i^j & = \mathbf{c}_{2i}^{j+1} \\ \mathbf{d}_i^j & = \mathbf{c}_{2i+1}^{j+1} - \frac{1}{2}(\mathbf{c}_{2i}^{j+1} + \mathbf{c}_{2i+2}^{j+1}) \end{array} \right. . \qquad (1)$$

where $\mathbf{c}_i^j$ and $\mathbf{d}_i^j$ are the coefficient of the MR curve at level j (see Eq. 8).

The reasons for the particular choice of working only with piecewise linear curves are threefold. First, the length preserving deformation method presented here is intended to work dynamically. Linear curves in contrast to higher order polynomial curves simplify and accelerate the length computations. Second, the particular effect of wrinkle creation in response to the motion of control points is a purely visual effect. Therefore, on a computer screen no visual difference can be observed between displaying a dense polygonal curve or a rasterized polynomial curve. Third, the algorithm will be used in Section 3 to create dynamic surface wrinkles, the surface being represented as a fine triangular mesh. Thus all curves on the surface are piecewise linear.

In the case of piecewise linear curves one can choose either to keep the total length constant or to preserve the length of each segment. We choose the second way because of two main reasons:

- It ensures the balance between segment's length that is to say the control points don't gather in a small part of the curve.
- It allows the length constraints to be expressed in such a way that computationally inefficient square roots evaluations can be avoided.

The length constraints on a deformed curve defined by $C^n = (\mathbf{c}_i^n)_i^T$ can now be reformulated as follows:

$$f_i(C^n) = \Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2 - l_i^2 = 0; \quad i = 0, \ldots, N-2 \ , \tag{2}$$

where l_i is the reference length of segment $[\mathbf{c}_i; \mathbf{c}_{i+1}]$ before deformation, $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$ and $\Delta z_i = z_{i+1} - z_i$.

2.2. Overview of the deformation method

We present here how to deform a 3D curve while enforcing the previous length constraints. The multiresolution representation of the curve is at the center of this process. It provides a friendly manipulating tool for the user: the curve can be deformed globally through a few coarse control points while preserving the details. Moreover the use of several decomposition levels allows to precisely control the deformation scale at each step of the process.

Three levels of decomposition (corresponding to the three columns in figure 1) are involved: the finest level n, the editing level e and an intermediate level w corresponding to the wrinkling scale if required by the length constraint. Hence C^j denote the gathering of the MR coefficients $\mathbf{c}^j, \mathbf{d}^j, \mathbf{d}^{j+1}, \cdots, \mathbf{d}^{n-1}$ (see Annexe A). The deformation process follows the loop of figure 1. It can be iterated several times, basically at each time step in a select-and-drag editing process. During the deformation process the curve undergoes several transitions and thus admits different geometric stages i.e. different values of the MR coefficients. Four different stages of C are involved, labelized by a different index:

- C_R is the initial curve and also the reference curve for the length;
- C_D is the deformed curve, i.e. after it has been edited and before any length preservation action;
- C_A is the attracting curve that originates from the explicit length preservation step;
- C_F is the final curve. It will replace C_R for the next deformation.

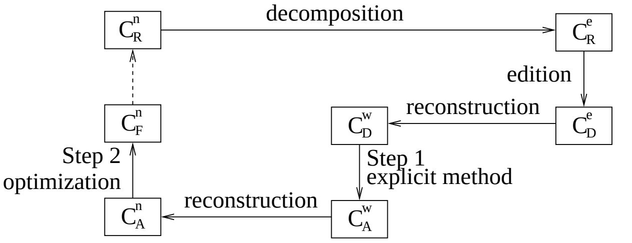


Figure 1. Editing loop: each box is a state of the curve; each horizontal transition is a change in the decomposition level; each vertical transition is a modification of the curve.

Let us describe the role of each transition in figure 1:

- **Decomposition:** The curve is decomposed into C_R^e at the level e, chosen by the user, following equation (1).
- **Edition:** The coarse control polygon C_R^e a level e is modified by the user. The corresponding deformed curve C_D has not the same length as the reference curve C_R. Notice that the choice of level e determines the extent of the deformation.
- **Reconstruction:** The deformed curve is then partially reconstructed at level w ($e \leq w < n$), chosen by the user to be the wrinkling scale.
- **Explicit length preserving:** An attracting curve C_A is constructed. This curve is obtained by modifying C_D^w at level w using the details in order to obtain a control polygon C_A^{w+1} whose edges have the same length as the control polygon segments of C_R at level $w+1$. This step is detailed in Section 2.4. Modifying details at level w is equivalent to modifying corresponding control points at level $w+1$. The choice of an intermediate level w different from e for length approximation increases the number of coarse control points defining the same portion of the curve. The closer to the highest level n is w, the higher is the number of control points for length approximation, and the higher is the frequency of the wrinkles in case of need. If the curve is stretched w acts as a kind of stiffness parameter.
- **Reconstruction:** The attractive curve is then completely reconstructed by re-inserting the details of C_R

at the levels $w+1,\dots,n$. After reconstruction, the length of C_A is close to satisfy the length constraint. Note that the use of a lazy wavelet scheme may lead to some sharp features in the reconstructed curve. Hence a smoothing is useful.

- **Length preserving by smoothing:** An optimization method applied to C_A leads to the final curve C_F. It precisely satisfies the length constraints, has a smooth shape, and it is close to the attracting curve C_A. This step is described in Section 2.3.

2.3. Optimization step

We detail here the step of exact length enforcing via optimization. After presenting the basic ideas and the calculus we explain why it is useful to combine it with the explicit length preserving step.

Starting from C_A, the problem to solve consists in length enforcing, i.e. finding a final curve C_F:

1. whose length is L_R;
2. that is close to C_A;
3. that is smoother than C_A.

The length is a strong constraint while the closeness and the smoothness are soft constraints. Hence let us choose a minimization method constrained by the length. The objective function contains a smoothness term preventing the curve to have unwanted wriggles and a distance term minimizing the distance to the attracting curve C_A.

In variational design a physical model is used for the description of a "smooth" or a "fair" curve or surface [25, 36]. The most widely used fairness criteria originate from the observation that the shape of a thin elastic beam or a thin plate under deformation which minimizes the bending energy is always smooth, i.e. has a visual pleasing shape.

Since the bending energy for a parametric curve, $E_B = \int \kappa^2(t)dt$, is a non-linear functional, it is common to use instead the linearized version [12, 4]:

$$E_B = \int |\mathbf{c}''(t)|^2 dt = \int x''(t)^2 + y''(t)^2 + z''(t)^2 dt \; .$$

Both expressions are identical if $|c'(t)| \equiv 1$.

In the case of a piecewise linear curve, a discrete version is derived from a finite difference approximation of the derivatives:

$$\begin{aligned} E(X,Y,Z) &= \textstyle\sum_{i=1}^{N-2} \|\frac{1}{4}(\mathbf{c}_{i-1} - 2\mathbf{c}_i + \mathbf{c}_{i+1})\|^2 \\ &= \tfrac{1}{2}(X^T H X + Y^T H Y + Z^T H Z), \end{aligned}$$

where H is a banded matrix and X, Y, Z are the coordinate vectors of C.

Besides we want the distance to C_A to be minimal. Hence we introduce a second term in the objective function, defined as the quadratic distance between C and C_A:

$$D(X,Y,Z) = \|X - X_A\|^2 + \|Y - Y_A\|^2 + \|Z - Z_A\|^2 \; .$$

The problem we aim to solve is now the following optimization problem:

$$\begin{aligned} &(X_F, Y_F, Z_F) = \arg\min \, \{ \, (1-\beta)E(X,Y,Z) + \beta D(X,Y,Z) \, \} \\ &\text{subject to } f_i(X,Y,Z) = 0 \; , \; i = 0, \dots, N-2 \; , \end{aligned} \tag{3}$$

where the constraints f_i are defined in 2 (see Section 2.1). $0 \le \beta \le 1$ is a scalar value which balances between a smoother curve and a curve closer to C_A. Similar energy functional have been used in [18].

With the technique of Lagrange multipliers [16], the minimization problem is restated to finding a stationary point of the following function:

$$g(X,Y,Z,\Lambda) = (1-\beta)E + \beta D + \sum_{i=0}^{N-2} \lambda_i f_i$$

where $\Lambda = (\lambda_0, \dots, \lambda_{N-2})^T$ is the vector of Lagrange multipliers.

The length constraints are quadratic expressions. Since minimizing a quadratic cost function subject to quadratic constraints is costly and since one of our objectives is to provide a fast algorithm, let us approach the length constraints f_i by using linearized constraints $\tilde{f_i}$ instead. $\tilde{f_i}$ is the Taylor expansion of f_i with respect to Δx_i, Δy_i and Δz_i in the neighbourhood of Δx_i^A, Δy_i^A and Δz_i^A. Hence one gets

$$\begin{aligned} \tilde{f_i} = \; & 2\Delta x_i^A \Delta x_i - (\Delta x_i^A)^2 + 2\Delta y_i^A \Delta y_i - (\Delta y_i^A)^2 \\ & + 2\Delta z_i^A \Delta z_i - (\Delta z_i^A)^2 - l_i^2 . \end{aligned}$$

Assuming that the approximation of f_i by $\tilde{f_i}$ holds, the problem can be restated as solving:

$$\vec{\nabla} \tilde{g} = 0 \;\text{ where }\; \tilde{g}(X,Y,Z,\Lambda) = (1-\beta)E + \beta D + \sum \lambda_i \tilde{f_i} \; . \tag{4}$$

A symmetric square sparse system of linear equations has to be solved:

$$\left\{ \begin{array}{l} (1-\beta)HX + 2\beta X + 2\Delta_X \Lambda = 2\beta X_A \\ (1-\beta)HY + 2\beta Y + 2\Delta_Y \Lambda = 2\beta Y_A \\ (1-\beta)HZ + 2\beta Z + 2\Delta_Z \Lambda = 2\beta Z_A \\ 2\Delta_X^T X + 2\Delta_Y^T Y + 2\Delta_Z^T Z = b \end{array} \right. \tag{5}$$

where $b_i = l_i^2 + (\Delta x_i^A)^2 + (\Delta y_i^A)^2 + (\Delta z_i^A)^2$ for $i = 0 \dots N-2$,
and where Δ_X, Δ_Y and Δ_Z are 2-banded matrices of size $N \times (N-1)$ containing Δx_i^A, Δy_i^A and Δz_i^A (see [30] for details).

Let us note that the result is not exact for the length constraints due to the linearization of the length constraints. In order to increase the precision, the solving is encapsulated in a loop which iterates the system solving and replaces (X_A, Y_A, Z_A) by (X, Y, Z) at each loop. Since the system is very sparse iterative methods are efficient.

In order to validate the approximation of f_i by the linearized expression $\tilde{f}_i$ we have to ensure that $\Delta x_i,\ \Delta y_i,\ \Delta z_i$ are close to $\Delta x_i^A,\ \Delta y_i^A,\ \Delta z_i^A$. In other words the curve C_A must nearly satisfy the length constraints. Hence the choice of C_A is crucial. We explain in the next section how to find a curve satisfying these conditions.

2.4. Expicit length preservation step for 3D curves

In this section we present a method computing in linear time an attracting curve suitable for the optimization step since it sufficiently approximates the length constraints. Though based on the same idea as the 2D curve [30] we will see that the 3D solution is more complex, involving the intersection of two spheres and a plane instead of two circles.

There are three different curves involved in this section: C_R, the reference curve, C_D, the curve deformed by the user, and C_A, the resulting curve that nearly satisfies the length constraint and that therefore is a good starting curve for the optimization of Section 2.3.

Suppose the user has modified the initial multiresolution curve C_R by displacing one control point at resolution level e, $0 \leq e < n$, leading to C_D which differs from C_R only in a localized portion. This classical multiresolution editing operation generally doesn't preserve the length of C_R. But in case of keeping the curve length constant one desired effect would be the generation of wrinkles at some scale w depending on the object. In order to approximate the reference length, the idea is to compute C_A from C_D by computing new detail coefficients of level w. All other detail coefficients of level $w+1$ up to n and all coarse coefficients (if possible) of level w are kept fixed.

Let us explain this particular choice. The detail coefficients at level w are used because they encode the geometric information at that scale. Assume $\mathbf{c}_i^w$ and $\mathbf{c}_{i+1}^w$ (see Fig. 2) have been moved closer by the user. A new detail $\mathbf{d}_i^w$ is computed such as the lengths $\tilde{l}_{2i}$ and $\tilde{l}_{2i+1}$ equal the reference lengths. Hence the norm of $\mathbf{d}_i^w$ increases, creating a wrinkle at scale w. Through the choice of w the user controls explicitly the wrinkling scale.

Fixing all other detail coefficients ensures that fine details of the initial curve are preserved. To not modify the coarse coefficients of C_D at level w ensures that the global shape of the resulting curve C_A differs as less as possible from C_D, i.e. it respects as much as possible the user's original deformation.

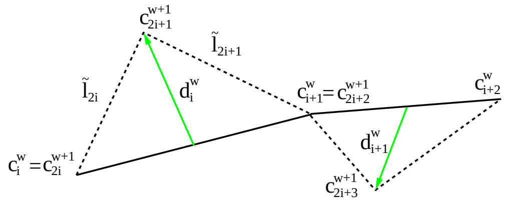


Figure 2. Interpolating scheme.

Let us now describe how in practice this idea has been implemented. Let $\tilde{l}_i$ be the length of the control polygon edges of the initial curve C_R at level $w+1$. Then starting at index i (corresponding to the coarse control point of C_R that has been modified by the user) the algorithm traverses all indices in increasing order, computes for each coarse polygon edge $[\mathbf{c}_i^w, \mathbf{c}_{i+1}^w]$ a new value of the corresponding detail coefficient $\mathbf{d}_i^w$ and possibly modifies the coarse coefficients $\mathbf{c}_i^w$ of C_A such that at level $w+1$, the length of the edges of C_R's and C_A's control polygons match. In other words we have to find the head of the vector $\mathbf{d}_i^w$: it lies in the intersection circle of the two spheres with centers $\mathbf{c}_i^w$ and $\mathbf{c}_{i+1}^w$, and with radii $\tilde{l}_{2i}$ and $\tilde{l}_{2i+1}$. Moreover we want the new detail to have the same orientation as the reference one relatively to the coarse edge $[\mathbf{c}_i^w, \mathbf{c}_{i+1}^w]$. Hence the problem turns out to intersect two spheres and a half-plane going through the centers. The solution is detailed and illustrated in Annexe B. It may occur that the spheres do not intersect:

- if one sphere is included in the other: we equalize the radii while keeping the total length $\tilde{l}_{2i} + \tilde{l}_{2i+1}$ constant;
- if the two spheres are distinct: we move $\mathbf{c}_{i+1}^w$ along the edge until the spheres are tangent to each other.

The same procedure as above is applied symmetrically to the other half of the curve, starting at index i and traversing all indices in decreasing order.

Discussion:

- An important advantage of this method is that the resulting curve C_A is close to C_D because their MR coefficients match whenever possible, except the details at level w (level of expected wrinkles). That is to say the curve C_A follows as much as possible the modification applied by the user.
- Let us recall the main condition for using the optimization method efficiently: a starting curve nearly satisfy-

ing the length constraints. Thanks to the length preservation at the scale $w + 1$, the reconstructed curve C_A (i.e. after adding back the fine details) has almost the same length as the initial curve. Hence it can be used as a good starting point for the second step (see Section 2.3) in order to precisely approach the length constraint.

- Following this method, a main role of the previous optimization method, whose cost dramatically drops, is to smooth the curve. It is useful since C_A generally may have sharp features.

3. Surfaces

Modelling inelastic soft tissues is a hard task because it involves area preserving, length preserving and minimizing some energies on the surface. In this section we show how the length preserving curve deformations presented in the previous section can be used to mimic the dynamic behaviour of materials like skin or cloths. We take advantage of the computational efficiency of the curve model in order to generate wrinkles dynamically with an easy control on the frequency and the extent.

The main idea of our proposal is to extract a curve on the surface that will be deformed following the algorithm of Section 2 and to re-inject it into the surface. The wrinkles are then propagated in a pre-defined neighbourhood of this curve. The wrinkles are smoothly attenuated inside the neighbourhood. The propagation is obtained by displacing each vertex of the neighbourhood, depending on its distance to the curve and on the curve's nearest points. Though all examples are given with manifold triangular meshes it can be applied to quadrilateral meshes and non-manifold meshes as well.

3.1. Curve extracting

The first step consists in extracting a curve defined as a sequence of edges of the mesh. Hence we get a piecewise linear 3D curve whose control points correspond to the vertices of the mesh. Depending on the application several extracting methods are possible, including user defined curve, shortest path between specified ends [24], intersection with other objects, extracting ridges [26], etc.

Then the curve is deformed at constant length using the MR method presented in Section 2. The deformed curve thus defines a dispacement of the corresponding mesh vertices. In order to create a surfacic deformation we have to propagate these displacements on the mesh in a predefined neighbourhood.

3.2. Propagation on the surface

Firstly we specify an area of influence defining the part of the mesh that will be modified. In the literature, different appraoches to solve this problem have been used. In [20]*...bref description de leur methodes de propagation ...*?? Another possibility is to *????*. Thee geometry-based approaches are probably the most accurate but it is much time consuming. This is the reason why we choose the following topology-based approach. Herein, we choose the region of extend for the wrinkle propagation by selecting the vertices at a uniformly bounded topological distance of the extracted curve (Fig. 3). The topological distance between two vertices is the minimal number of edges for all paths connecting this two vertices. We define the area of influence as the set of mesh triangles whose vertices have a topological distance to a curve's vertex less or equal than a specified number. The choice of using topological distance is made for efficiency reasons. Highly non-uniform meshes would however require using geometric distance instead of a topological distance. But these meshes don't occur so much in practice.

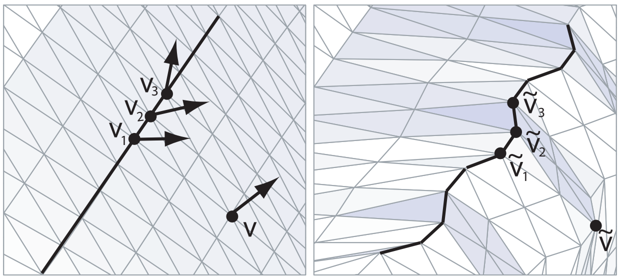


Figure 3. Propagation of the deformation.

Let us now explain how to compute the new position of all vertices belonging to the area of influence so that the new shape of the extracted 3D curve is propagated smoothly inside that area. Let $\mathbf{v}$ be a vertex (see Fig. 3) inside the area. The corresponding vertex $\tilde{\mathbf{v}}$ on the deformed surface is obtained by $\tilde{\mathbf{v}} = \mathbf{v} + \vec{\delta}_{\mathbf{v}}$. The displacement $\vec{\delta}_{\mathbf{v}}$ has to be defined. This vertex may have several closest points on the curve (named $\mathbf{v}_1, \cdots, \mathbf{v}_k$) with respect to the topological distance. It means that there is a path from $\mathbf{v}$ to each of them with topological length equal to the minimal topological distance $l_{\mathbf{v}}$ from $\mathbf{v}$ to the curve. Remember we know the displacement vector of each curve point. The displacement $\vec{\delta}_{\mathbf{v}}$ is defined as an average of $\vec{\delta}_{\mathbf{v}_1}, \cdots, \vec{\delta}_{\mathbf{v}_k}$. But the edges of the mesh may have different lengths involving important differences between the geometric distances $\|\mathbf{v} - \mathbf{v}_1\|, \cdots, \|\mathbf{v} - \mathbf{v}_k\|$. In order to overcome that problem the average is weighted by the inverse of the geometric

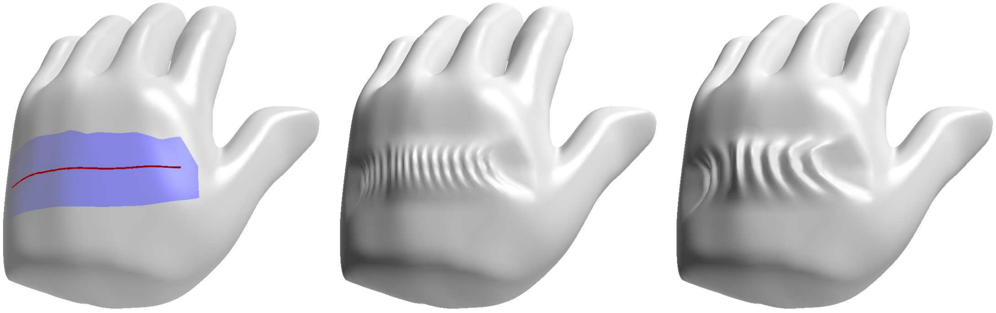


Figure 4. Selected area and profile curve on the initial mesh (left); resulting small (middle) and large (right) wrinkles on the back of the hand.

distance between the points and $\mathbf{v}$:

$$\vec{\delta}_{\mathbf{v}} = a(l_{\mathbf{v}}) \left(\sum_{i=1}^{k} \frac{1}{\|\mathbf{v} - \mathbf{v}_i\|} \right)^{-1} \sum_{i=1}^{k} \frac{1}{\|\mathbf{v} - \mathbf{v}_i\|} \vec{\delta}_{\mathbf{v}_i} , \quad (6)$$

where $a(l)$ is a transverse attenuation function making the deformation more realistic. It must decrease from 1 for $l_{\mathbf{v}} = 0$ (i.e. $\mathbf{v}$ belongs to the curve) to 0 for $l_{\mathbf{v}}$ maximum (i.e. when $\mathbf{v}$ is on the boundary of the selected area). In our examples a is made up of a step $a(l) = 1$ for l small, followed by a smooth cubic polynomial join.

3.3. Results

We present here examples of mesh editing that illustrates the process previously presented. Once the mesh is loaded and the deformation area is defined, the deformation process works dynamically.

Figure 4 shows the wrinkling of a mesh. The initial mesh (left) has $50\,000$ triangles and the selected area (blue area) has $1\,400$ triangles. The extracted curve (red) is pinched by the user and the length is enforced following the process of Sections 2 and 3. Two different values for w (see Section 2.2) are used: $w = 4$ (middle) and $w = 3$ (right), providing small and large wrinkles on the back of the hand. It illustrates the control of the wrinkling scale thanks to the MR representation. The propagation process automatically creates realistic surface wrinkles.

Figure 5 shows the superposition of three deformations (lower row). The initial surface (upper left) has $93\,000$ triangles but only $13\,000$ triangles have been modified, creating several wrinkles around the cow's neck (upper right). This example illustrates the ability of our model to be integrated into a surface designing tool.

4. Conclusion

We have first presented a method for length preserving deformation of multiresolution 3D curves. It combines a step of length approximating and a smoothing step via constrained optimization. The MR representation allows easy editing and control of the deformation scale. This method is then used for mesh wrinkling by extracting a deformation profile which is propagated on the surface. The whole process works dynamically. It is illustrated on triangular meshes.

Future works concern the adaptation of the method to adaptative meshes. Refining the mesh on the wrinkling area while keeping a coarse sampling on flat areas could improve the results with minor computation overloading.

Acknowledgements

The work was partially supported by the European Community 6-th framework programm, with the Network of Excellence *Aim@Shape* (http://www.aimatshape.net).

References

[1] D. Baraff and A. Witkin. Large steps in cloth simulation. *Computer Graphics*, 32(Annual Conference Series):43–54, 1998.

[2] J. F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH Computer Graphics*, pages 286–292, 1978.

[3] L. Boissieux, G. Kiss, N. Magnenat-Thalmann, and P. Kalra. Simulation of skin aging and wrinkles with cosmetics insight. In *Computer animation and simulation*, pages 15–27, 2000.

[4] G.-P. Bonneau and H. Hagen. Variational design of rational bézier curves and surfaces. *Curves and Surfaces in Geometric Design*, pages 51–58, 1994.

[5] M. Botsch and L. Kobbelt. Multiresolution surface representation based on displacement volumes. *Computer Graph-*

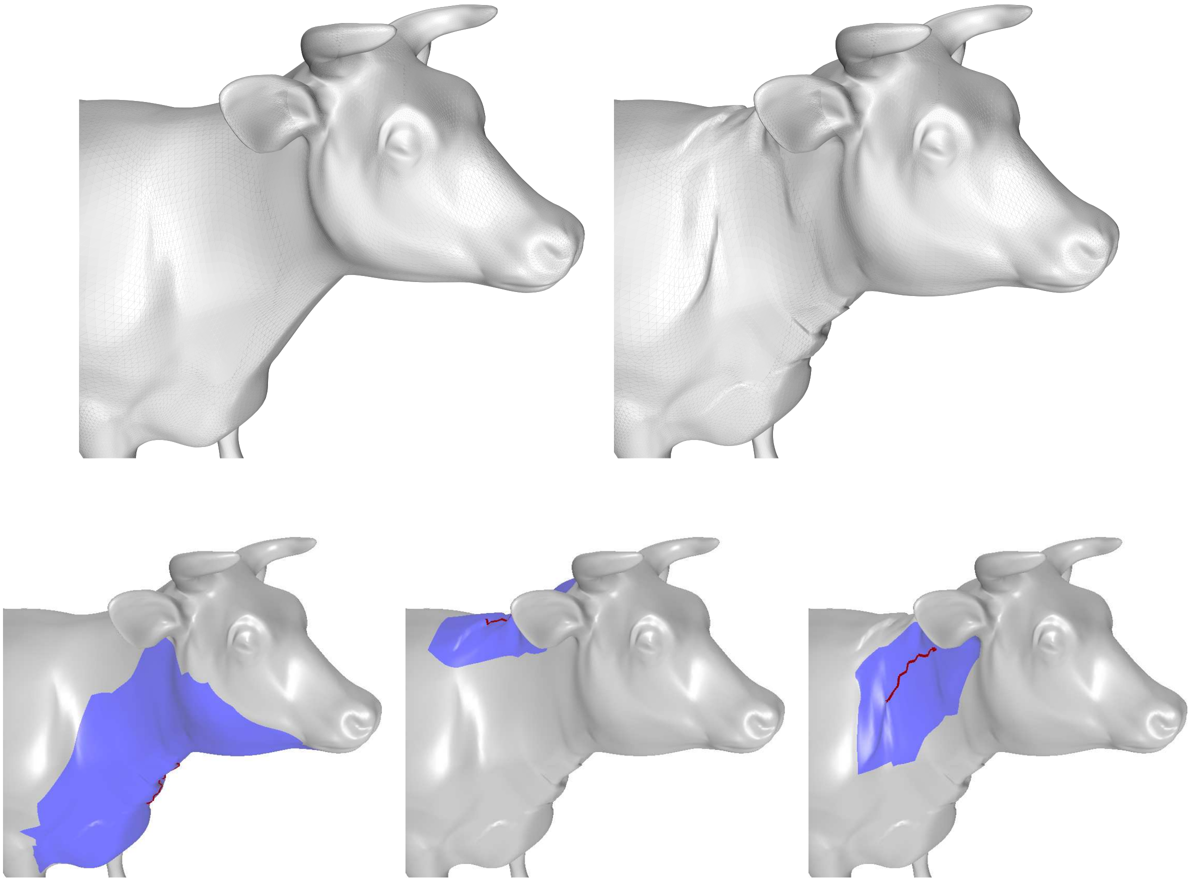


Figure 5. Wrinkles around the neck. The initial mesh (upper row left) is wrinkled (upper row right) trough three deformations (lower row).

ics Forum, 22(3):483–491, 2003. (Proceedings Eurographics ’03).

[6] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *SIGGRAPH Computer Graphics*, pages 257–266, 1991.

[7] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr. Adaptive simulation of soft bodies in real-time. In *Computer Animation 2000, Philadelphia, USA*, pages 133–144, May 2000.

[8] M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *SIGGRAPH Computer Graphics*, pages 287–290. ACM Press, 1995.

[9] M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Graphics Interface*, pages 1–8, 1999.

[10] G. Elber. Multiresolution curve editing with linear constraints. In *6th ACM/IEEE Symposium on Solid Modeling and Applications*, pages 109–119. Ann Arbor, Michigan, June 2001.

[11] G. Elber and C. Gotsman. Multiresolution control for nonuniform bspline curve editing. In *The third Pacific Graphics Conference on Computer Graphics and Applications, Seoul, Korea*, pages 267–278, 1995.

[12] G. Farin, G. Rein, N. Sapidis, and A. Worsey. Fairing cubic b-spline curves. *Computer Aided Geometric Design*, 4:91–103, 1987.

[13] A. Finkelstein and D. H. Salesin. Multiresolution curves. *SIGGRAPH Computer Graphics*, pages 261–268, 1994.

[14] B. Fowler. Geometric manipulation of tensor product surfaces. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 101–108. ACM Press, 1992.

[15] S. J. Gortler and M. F. Cohen. Hierarchical and variational geometric modeling with wavelets. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 35–42. ACM Press, 1995.

[16] B. S. Gottfried and J. Weisman. *Introduction to Optimization Theory*. Prentice Hall, Englewood Cliffs, New Jersey, 1973.

[17] S. Hadap, E. Bangerter, P. Volino, and N. Magnenat-Thalmann. Animating wrinkles on clothes. In *VIS '99: Proceedings of the conference on Visualization*, pages 175–182, 1999.

[18] H. Hagen, S. Hahmann, and G. Bonneau. Variational surface design and surface interrogation. *Computer Graphics Forum*, 12(3):447–459, 1993. (Proceedings Eurographics'93).

[19] S. Hahmann, B. Sauvage, and G.-P. Bonneau. Area preserving deformation of multiresolution curves. *CAGD*, 22(4):349–367, 2005.

[20] F. Jing, A. Joneja, and K. Tang. Modeling wrinkles on smooth surfaces for footwear design. *Computer-Aided Design*, 37:815–823, 2005.

[21] C. Larboulette and M.-P. Cani. Real-time dynamic wrinkles. In *Computer Graphics International*. IEEE Computer Society Press, 2004. Greece.

[22] J. Lasseter. Principles of traditional animation applied to 3d computer animation. *SIGGRAPH Computer Graphics*, pages 35–44, 1987.

[23] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[24] J. Mitchell. Geometric shortest paths and network optimization. In *Handbook of Computational Geometry*, volume 334, pages 633–702. 2000.

[25] H. Nowacki and D. Reese. Design and fairing ship surfaces. *Surfaces in CAGD*, pages 121–134, 1983.

[26] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 23(3):609–612, 2004.

[27] Q. Peng, X. Jin, and J. Feng. Arc-length-based axial deformation and length preserved animation. In *Computer Animation*, pages 86–92, 1997.

[28] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. In *IEEE Transactions on Visualization and Computer Graphics*, volume 2(1), pages 19–27, 1996.

[29] J. Roulier and B. Piper. Prescribing the length of rational bezier curves. *Computer-Aided Geometric Design*, 13(1):23–43, 1996.

[30] B. Sauvage, S. Hahmann, and G.-P. Bonneau. Length preserving multiresolution editing of curves. *Computing*, 72:160–171, 2004.

[31] E. J. Stollnitz, T. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.

[32] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[33] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Computer Graphics*, pages 205–214, 1987.

[34] D. Terzopoulos and H. Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *Transactions on Graphics*, 13(2):103–136, 1994.

[35] M.-L. Viaud and H. Yahia. Facial animation with wrinkles. In *Eurographics Workshop on Animation and Simulation*, 1992.

[36] W. Welch and A. Witkin. Variational surface modeling. *SIGGRAPH Computer Graphics*, pages 157–166, 1992.

[37] Y. Wu, P. Kalra, and N. Magnenat-Thalmann. Simulation of static and dynamic wrinkles of skin. In *CA '96: Proceedings of the Computer Animation*, page 90, Washington, DC, USA, 1996. IEEE Computer Society.

[38] Y. Wu, P. Kalra, L. Moccozet, and N. Magnenat-Thalmann. Simulating wrinkles and skin aging. *The visual computer*, 15(4):183–198, 1999.

A. Multiresolution curves

Let us briefly sketch the notation of the wavelet based multiresolution analysis used here. For more details see [23], [13], and [31]. Suppose we have a certain functional space E and some nested linear *approximation spaces* $V^j \subset E$ with $V^0 \subset V^1 \subset \cdots \subset V^n$. Let V^n be of dimension N. Since we are dealing with finite curves, these spaces have finite dimension. Let V^j be spanned by a set of basis functions $\varphi^j = [\varphi^j_1, \ldots, \varphi^j_m]^T$, called *scaling functions*. A space W^j being the complement of V^j in V^{j+1} is called the *detail space*. Its basis functions $\psi^j = [\psi^j_1, \ldots, \psi^j_{N-m}]^T$ are such that together with φ^j they form a basis of V^{j+1}. The functions ψ^j_i are called *wavelets*. The space V^n can therefore be decomposed as follows:

$$\begin{aligned} V^n &= V^{n-1} \oplus W^{n-1} = V^{n-2} \bigoplus_{j=n-2}^{n-1} W^j \\ &= \cdots = V^0 \bigoplus_{j=0}^{n-1} W^j. \end{aligned} \tag{7}$$

A *multiresolution curve* is then defined as a parametric curve $\mathbf{c}(t) = (\mathbf{c}^n)^T(\varphi^n)$, element of V^n, where $\mathbf{c}^n$ is a column of its N control points in $\mathbb{R}^3$. Due to property 7 the same curve can be expressed in terms of the basis functions of the different decompositions of V^n, each of it corresponding to a certain resolution of the curve. The multiresolution curve at any *level of resolution* $e \in [0, n]$, i.e. element of $V_e \bigoplus_{j=e}^{n-1} W_j$ is then given by some coarse control points $\mathbf{c}^e$ that form approximations of the initial control polyline and by the detail coefficients $\mathbf{d}^e, \ldots, \mathbf{d}^{n-1}$ as follows:

$$\begin{aligned} \mathbf{c}(t) &= (\mathbf{c}^e)^T(\varphi^e) + (\mathbf{d}^e)^T(\psi^e) + \cdots + (\mathbf{d}^{n-1})^T(\psi^{n-1}), \\ & e = 0, \ldots, n. \end{aligned} \tag{8}$$

The *filter bank* algorithm [23, 13] is used to compute the coefficients of all levels of resolutions from the initial coefficients $\mathbf{c}^n$ and vice versa.

The function so represented can be edited in an intuitive way: editing the coarse coefficients modifies the global shape without affecting the details; in contrast editing the details modifies the character of the object without changing its overall shape.

B. Intersecting 2 spheres and a half-plane

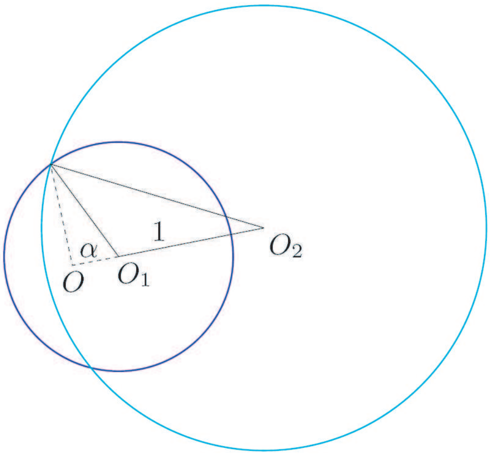


Figure 6. Intersecting two spheres.

The problem of section 2.4 can be stated as finding the intersections of (see Fig. 6):

1. the sphere $\mathcal{S}_1(O_1, r_1)$,
2. the sphere $\mathcal{S}_2(O_2, r_2)$,
3. the plane $\mathcal{P}:\ (O_1, O_2, \vec{\delta})$,

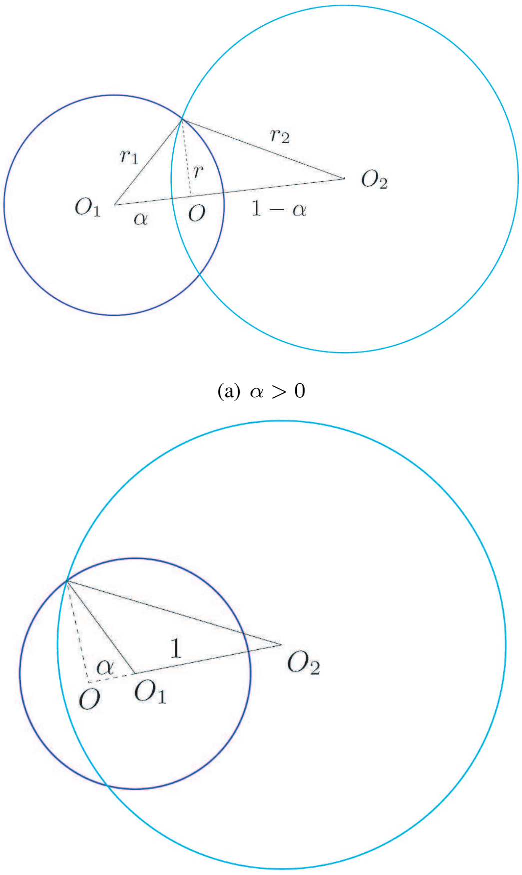
instantiated with $O_1 = \mathbf{c}_i^w$, $O_2 = \mathbf{c}_{i+1}^w$, $r_1 = \tilde{l}_{2i}$, $r_2 = \tilde{l}_{2i+1}$ and $\vec{\delta} = \mathbf{d}_i^w$ (detail of the reference curve C_R).
Let us define $d = dist(O_1, O_2) = \|\overrightarrow{O_1O_2}\|$.
Assuming $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$ (i.e. $r_1 + r_2 \leq d$ and $|r_1 - r_2| \leq d$), the intersection is a circle $\mathcal{C}(O, r)$:

1. lying in a plane orthogonal to $\overrightarrow{O_1O_2}$ (then also to $\mathcal{P}$),
2. with center $O \in (O_1O_2)$,
3. with radius $r \leq \min\,(r_1, r_2)$.

Let $(1-\alpha, \alpha)$ be the homogenous barycentric coordinates of O with respect to $\{O_1, O_2\}$ i.e. $\overrightarrow{O_1O} = \alpha\ \overrightarrow{O_1O_2}$ (see Fig. 7).
Applying Pythagorean theorem in the triangles (O_1, O, F) and (O_2, O, F) with F belonging to the circle, we deduce O and r:

$$\left\{\begin{array}{ll} r^2 + (1-\alpha)^2 d^2 & = r_2^2 \\ r^2 + \alpha^2 d^2 & = r_1^2 \end{array}\right. \Leftrightarrow \left\{\begin{array}{ll} \alpha & = (d^2 + r_1^2 - r_2^2)\ /\ 2d^2 \\ r^2 & = r_1^2 - \alpha^2 d^2 \end{array}\right.$$

Hence $\mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{P} = \mathcal{C}(O, r) \cap \mathcal{P}$ is the set of 2 points $\{F, F'\}$ such as:



(a) $\alpha > 0$

(b) $\alpha < 0$

Figure 7. The intersection circle.

1. $\overrightarrow{OF}$ and $\overrightarrow{OF'}\ \perp\ \overrightarrow{O_1O_2}$,
2. F and $F' \in \mathcal{P}$, i.e. $\overrightarrow{OF}$ and $\overrightarrow{OF'} \perp\ \vec{\delta} \times \overrightarrow{O_1O_2}$,
3. $\|\overrightarrow{OF}\| = \|\overrightarrow{OF'}\| = r$,
4. $\overrightarrow{OF} = -\overrightarrow{OF'}$.

Then $\overrightarrow{OF}$ and $\overrightarrow{OF'}$ are the only 2 vectors *colinear* with $\overrightarrow{O_1O_2} \times (\vec{\delta} \times \overrightarrow{O_1O_2})$ whose norm equals r. The one lying in the same half-plane as $\vec{\delta}$ has the same *direction* as $\overrightarrow{O_1O_2} \times (\vec{\delta} \times \overrightarrow{O_1O_2})$.