

FREE DEFORMATION OF MULTIREOLUTION B-SPLINES CURVES

Luis A. Rivera ⁽¹⁾ Paulo C. P. Carvalho ⁽²⁾ Luiz Velho ⁽²⁾

(1) Departamento de Informática, PUC-Rio,
Marquês de São Vicente, 255, 22453-900, Rio de Janeiro, RJ, Brasil
rivera@inf.puc-rio.br

(2) IMPA- Instituto de Matemática Pura e Aplicada,
Estrada Dona Castorina, 110, 22460-320, Rio de Janeiro, RJ, Brasil
{[pcezar](mailto:pcezar@impa.br), [lvelho](mailto:lvelho@impa.br)}@impa.br

ABSTRACT. When a point of the curve is dragged to another position, a local curve segment is affected, whose size could be determined by propagating the deformation on lower resolution of the curve in multiresolution representation. Therefore, for a given point displacement, different relative local segments can be modified, by varying local parameters given by a function which simulates a stiffness constant. Using this method, the designer can interactively modify the curve, in a natural manner, by controlling the size of the affected segment, avoiding the use of energy terms for deformation. We have tested our ideas with a prototype system for modeling uniform B-spline curves in multiresolution, using biorthogonal B-spline wavelets.

1. INTRODUCTION

Free deformations tools are necessary in many geometric modeling systems, because, often, one needs to model objects having complex shapes, usually obtained by manipulating some region of the object. By deforming existing objects, the designer can obtain, in a natural manner, the desired shape. So, he may wish to modify a region superficially only. Or, he may intend to modify a small region, in order to simulate the behavior of an elastic material. He may also wish to make a large modification in the vicinity of a point, simulating a metallic material (such as a wire) with a given stiffness.

Recent work in interactive modelling has provided tools for the deformation of objects based on different approaches, but do not solve the problem completely. Control point manipulation [12, 2] is inappropriate, because the user does not know how to move control points in order to obtain a desired

form. The same problem is found in knot point operations in B-splines objects [4, 12, 11]. These techniques solve the problem through very tedious work. Direct manipulation techniques [1, 5, 6] allow for the intuitive deformation of objects, with implicit movement of control points, caused by direct edits of points on objects, but do not provide ways to control the region affected by the deformation. The interactive deformation based on variational theory [19, 20] is another alternative shape design, that works by minimizing the function energy. Variational design using multiresolution theory [16, 17] is a technique for controlling the smoothness of objects editing the effects at continuous levels. In these cases, the solution involves the manipulation of energy functions. Realistic effects are obtained using physics-based deformation of objects [8], but it is difficult to determine the adequate force for a desired deformation.

We formulate a new approach to manipulate the shape of objects (2D curves), controlling a local attribute that determines the size of the segment to be modified. We employ direct manipulation techniques to produce a local deformation on segments of a multiresolution representation, in such a way that the movement of control points will depend on the degree of deformation determined by the designer. For that, the section 2 is dedicated to a review of theory, focusing on multiresolution curve representation using biorthogonal wavelets. Section 3 treats the direct manipulation of multiresolution curves; for that, we define local deformations applied to a multiresolution curve. In section 4, we formulate the technique to control the deformation through low-resolution coefficients. In section 5, we define the distribution function that permits the distribution of displacement segments in all low-resolution coefficients, and Section 6 concludes this paper with a summary of results and future research directions.

2. REPRESENTATION AND MULTIRRESOLUTION OF CURVES.

In this section, we present the framework for multiresolution curve representation, using B-splines bases.

2.1. B-spline curves

The B-splines basis functions, $N_i^k(t)$, of order k , are defined by

$$N_i^1(t) = \begin{cases} 1 & \text{if } x_i \leq t < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$(2.1) \quad N_i^k(t) = \frac{(t - x_i)N_i^{k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1}^{k-1}(t)}{x_{i+k} - x_{i+1}}.$$

The values x_i , known as *knots*, are elements of the knot vector X , with $x_i \leq x_{i+1}$.

All points of a curve segment j are generated by k basis function with k constant control points C_j , by only varying the parameter t in a domain interval. For that, we use the basis function reparameterized to the interval $[0, 1)$, because the uniform periodic B-spline basis functions, used in this work, are all translates of each other and have the same form on each interval. See figure 1.

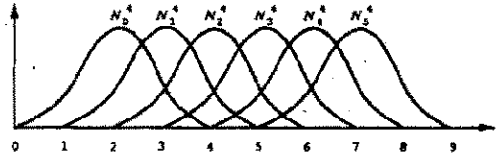


Figure 1. Periodic uniform cubic B-spline basis functions

The j^{th} segment of an uniform periodic B-spline curve of order k is defined as

$$(2.2) \quad f_j(t) = \tilde{N}^k(t).C_j,$$

where each column of the matrix $\tilde{N}^k(t)$ is formed by coefficients of the reparameterized basis function $N_r^k(t)$, $0 \leq r \leq 3$, to the interval $[0, 1)$ (see [13] for more detail). Specifically for $k = 4$, the matrix expression of the reparameterized basis function is

$$(2.3) \quad \tilde{N}^k(t) = \frac{1}{6} [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}.$$

To make a segment of curve $f_j(t)$, according to expression (2), the coefficient matrix columns are $C_j = [c_{j-1} \ c_j \ c_{j+1} \ c_{j+2}]^T$, and the basis is $\tilde{N}^4(t) = [N_0^4(t) \ N_1^4(t) \ N_2^4(t) \ N_3^4(t)]$.

2.2 Multiresolution curve

We use the multiresolution curve representation as presented in [3, 14], built from B-spline wavelets, because it provides effective hierarchical representation of curves. However, we use biorthogonal wavelets instead of orthogonal or semi-orthogonal wavelets.

Let $f(t)$ be a curve with 2^n control points in column matrix $C^n = [c_0^n \ c_1^n \ c_2^n \ \dots \ c_{2^n-1}^n]^T$, where each element $c_j^n \in R^2$, i.e. $c_j^n = (x_j, y_j)$. For a multiresolution representation of the curve, we denote by $f^n(t)$ the refined curve, where n represents the highest-resolution level, whereas $f^j(t)$, for $j < n$, corresponds to low-resolution representations of the curve.

2.2.1 B-spline wavelets and curves in multiresolution

In order to decompose a curve f^j into a low-resolution curve f^{j-1} and a difference curve ϕ^{j-1} it is necessary to use some form of linear filtering and down-sampling on the entries of C^j , which are the control points of f^j , to obtain the low-resolution *scaling coefficients* C^{j-1} and *detail coefficients* D^{j-1} .

The coefficients C^{j-1} are obtained by using a *scaling filter matrix* A^j , such that $C^{j-1} = A^j C^j$, and D^{j-1} are obtained by using a *detail filter matrix* B^j ($D^{j-1} = B^j C^j$). Both matrices, A^j and B^j , are known as *analysis filter matrices*. For constructing the object from their multiresolution coefficients, we use the *synthesis filter matrices* P^j and Q^j , such that $C^j = P^j C^{j-1} + Q^j D^{j-1}$.

The curve f^j is represented by the linear sum of the *scaling basis function* $\Phi^j(t)$ with the scaling coefficients C^j : as in (2.2), we have $f^j(t) = \Phi^j(t)C^j$, where the vector $\Phi^j(t) = [\phi_1^j(t), \dots, \phi_{2^j}^j(t)]$ contains the basis functions of space V^j . In this case, each $\phi_i^j(t)$ is a cubic B-spline function. The $\Phi^j(t)$ is refined by the synthesis filter matrix P^j (i.e. $\Phi^{j-1}(t) = \Phi^j(t)P^j$). The *wavelets* $\Psi^{j-1}(t)$ are the basis function of space W^{j-1} , and are related with Φ^j by the matrix Q^j satisfying $\Psi^{j-1}(t) = \Phi^j(t)Q^j$.

When W^{j-1} is the orthogonal complement of V^{j-1} in V^j , $\Psi^{j-1}(t)$ are orthogonal wavelets. The spaces $\{V^j\}$, such that $\dots \subset V^{j-1} \subset V^j \subset \dots$, define the *multiresolution analysis* of f .

2.2.2 Biorthogonal wavelets

Biorthogonal wavelets are a generalization of orthogonal wavelets; they have some attributes of orthogonal wavelets and are more flexible. These attributes are essential to express geometric objects in multiresolution and their manipulations with the fast transformation using B-splines basis functions.

In biorthogonal multiresolution, biorthogonal bases are used in place of orthogonal bases: one basis is used for analysis and another for synthesis. For analysis, to calculate C^{j-1} and D^{j-1} , the *primal* basis Φ^j and Ψ^j are used, respectively; for synthesis, the *dual* basis $\tilde{\Phi}^j$ and $\tilde{\Psi}^j$ are used. The functions Φ^j and $\tilde{\Phi}^j$ are basis of spaces V^j and \tilde{V}^j , respectively; on the other hand, Ψ^j and $\tilde{\Psi}^j$ are basis of W^j and \tilde{W}^j , respectively. So, we have two multiresolution spaces

$$\dots \subset V^{j-1} \subset V^j \subset \dots \quad \text{and} \quad \dots \subset \tilde{V}^{j-1} \subset \tilde{V}^j \subset \dots,$$

with $V^{j-1} + W^{j-1} = V^j$ and $\tilde{V}^{j-1} + \tilde{W}^{j-1} = \tilde{V}^j$. Orthogonality is enforced between primal and dual basis (for more detail, see [14, 7, 15]).

A scaling function $\phi_s^j(t)$ is defined by shifting and scaling a single function $\phi(t)$, called *mother function*. In general, it is expressed as $\phi_s^j(t) = \phi(2^j t - s)$.

The refinement of a scaling function and a wavelet function could be rewritten as

$$\phi(t) = \sum_m h_m \phi(2t - m) \text{ and } \psi(t) = \sum_m g_m \phi(2t - m).$$

Similarly, the dual basis functions are expressed by using the dual mother function $\tilde{\phi}(t)$, as

$$\tilde{\phi}(t) = \sum_m \tilde{h}_m \tilde{\phi}(2t - m) \text{ and } \tilde{\psi}(t) = \sum_m \tilde{g}_m \tilde{\phi}(2t - m).$$

Finally, considering the orthogonality condition of the basis, one obtains the relation between coefficients and filters, for decomposition:

$$(2.4) \quad c_m^j = \sum_s h_{s-2m} c_s^{j+1} \text{ and } d_m^j = \sum_s g_{s-2m} c_s^{j+1},$$

and for reconstruction:

$$(2.5) \quad c_m^{j+1} = \sum_s \tilde{h}_{m-2s} c_s^j + \sum_s \tilde{g}_{m-2s} d_s^j.$$

Observe that, in the computation of coefficients c_m^j and d_m^j , we capture elements relative to even positions ($2m$) of C^{j+1} . This means that, in each decomposition recursion, half the number of elements of C^{j+1} are generated. The filter coefficients, which are entries of matrices A, B, P and Q , can be computed using a formulation in the frequency domain [7] or in the spatial domain [15]. They can be interchanged, so they can be used for analysis or synthesis.

For the implementation of (4) and (5), we use the *fast wavelet transform* [7].

3. DIRECT MANIOPULATION OF MULTIREOLUTION CURVES

It is possible to modify interactively a curve segment by moving one point to a target position. This can be done by moving the k corresponding control points, that influence a total of $k + (k - 1)$ segments. To compute the movement of k control points of B-splines curves we use results from Bartels and Beaty [1], and Bartels and Fowler [5].

The operation consists in computing the variation Δc_{j+i-1} of the control point $\{c_{j+i-1}\}_{i=0, \dots, k-1}$ relative to curve segment j ,

$$(3.1) \quad \Delta c_{j+i-1} = \Delta f_j(t) \frac{N_{i-1}^k(t)}{\sum_{l=0}^{k-1} (N_{l-1}^k(t))^2}.$$

The basis functions $N_{i-1}^k(t)$ are the same as in (2.3). The new position of point c_{j+i-1} is given by $\tilde{c}_{j+i-1} = c_{j+i-1} + \Delta c_{j+i-1}$.

When the control points of the object are very close to each other, the effect of the manipulation does not look natural. This effect is a local deformation, where the manipulation of one point affects only a small segment, influenced by k control points.

In order to extend the deformation to a larger segment, one needs to modify a larger number of control points. Care must be taken, however, to do so in such a way that the resulting deformation looks natural. Multiresolution representations provide an adequate framework to achieve this goal. In this approach, as described below, we still adjust only one curve segment (controlled by k control points) at a time, but this is done over the different representation levels. Controlling the amount of change at each level makes the deformation more or less global.

3.1 Manipulation in decomposition process

Here, we use the ideas on multiresolution curve editing, described briefly in [3, 10]. In order to make some modification of the curve shape using local deformation in multiresolution representation, we manipulate all low-resolution coefficients. The amount of change varies according to the resolution level. These modifications will be propagated to the highest-resolution curve after the reconstruction process.

The deformation of the curve f^n can be controlled by suitably propagating to all low-resolution curves the displacement vector Δd of a point $f^n(t)$. If Δd_j is the corresponding displacement applied to a point of f^j relative of the point $f^n(t)$, the resulting low-resolution curve will be $\tilde{f}^j = f^j + \Delta f^j$. The Δd_j is determined from Δd by a distribution function defined in section 5. The modified curve, \tilde{f}^j , affects low and high-resolution levels. When the modification is made in the decomposition process, the coefficients generated from this modified curve include these variations.

Given the refined curve control points C^n , the method consists of applying a local deformation, Δd_n , to obtain $\tilde{C}^n = C^n + \Delta C^n$. These new elements will be different to the ones obtained by reconstruction, because the lower level elements are also modified locally. In general, we could denote the scaling coefficients to be decomposed by \tilde{C}_d^j , and the reconstruction result by \tilde{C}_a^j . After decomposing \tilde{C}_d^j we obtain \tilde{C}^{j-1} and \tilde{D}^{j-1} . We then apply local change to obtain \tilde{C}_d^{j-1} and \tilde{D}_d^{j-1} . This process is applied recursively down to level m (minimum level permitted for decomposition). When \tilde{C}^j is modified in $\Delta \tilde{C}^j$, detail orientations are updated in $\Delta \tilde{D}^j$. When $m = 2$, for example, the process is as follows:

$$\text{Modification: } \tilde{C}_d^n = C^n + \Delta C^n.$$

$$\text{Decomposition: } \tilde{C}^{n-1} = C^{n-1} + A^n \Delta C^n,$$

$$\tilde{D}^{n-1} = D^{n-1} + B^n \Delta C^n.$$

$$\text{Modification: } \tilde{C}_d^{n-1} = C^{n-1} + A^n \Delta C^{n-1} + A^n \Delta C^n + \Delta(A^n \Delta A^n),$$

$$\tilde{D}_d^{n-1} = D^{n-1} + B^n \Delta C^{n-1} + B^n \Delta C^n + \Delta(B^n \Delta C^n).$$

After the first stage of modifications and decompositions, we reconstruct recursively the curve. In this second stage, it is necessary, prior to each reconstruction, to convert each detail coefficient from the local coordinate system to a global coordinate system as detailed in section 3.3. Having both elements expressed in the same system, we can then reconstruct the curve as

$$\tilde{C}_a^{j+1} = P^{j+1}\tilde{C}_a^j + Q^{j+1}\tilde{D}_a^j.$$

Finally, we obtain the highest resolution coefficients as

$$\tilde{C}_a^n = C^n + \Delta C^n + \Delta_r C^n.$$

where $\Delta_r C^n$ is the projection on level n of all variations in lower levels. Considering $m = 2$, in the sequence of deformation and decomposition above, the reconstruction is

$$\begin{aligned} \tilde{C}_a^n &= P^n \tilde{C}_a^{n-1} + Q^n \tilde{D}_a^{n-1} \\ &= C^n + \Delta C^n + P^n \Delta C^{n-1} + Q^n \Delta C^{n-1} + \\ &\quad + P^n \Delta(A^n \Delta C^n) + Q^n \Delta(B^n \Delta C^n) \end{aligned}$$

Finally, the modified curve is

$$\begin{aligned} \tilde{f}^n &= \Phi^n [C^n + \Delta C^n + P^n \Delta C^{n-1} + Q^n \Delta C^{n-1} + \\ &\quad + P^n \Delta(A^n \Delta C^n) + Q^n \Delta(B^n \Delta C^n)] \\ &= f^n + \Delta f^n + proj_n \Delta f^{n-1} + proj_n (\Delta proj_{n-1} \Delta f^n) \\ &= f^n + \Delta f^n + \Delta_r f^n, \end{aligned}$$

where Δf^n is the local variation of f^n and $\Delta_r f^n$ is the sum of lowers variations projected in level n . The $proj_n \Delta f^{n-1}$ is the projection of Δf^{n-1} onto resolution level n .

Figure 2 illustrate the process described above. On the left we see the effect of applying the deformation only to C^n ; on the right we show the deformation obtained when the respective lower level coefficients are also modified.

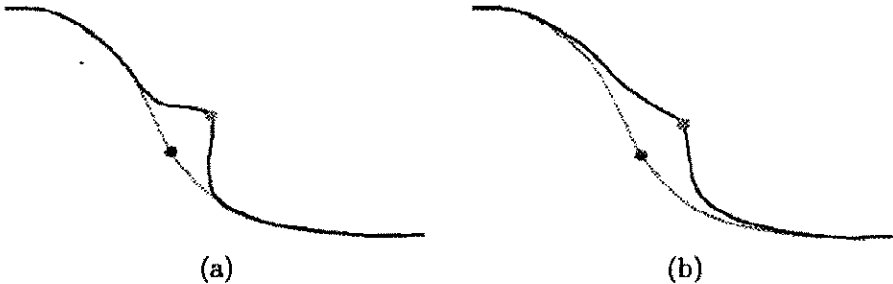


Figure 2. A curve deformed: (a) modifying only on the highest-resolution curve; (b) distributing displacement on three lower-resolution curves.

3.2 Displacement of points in multiresolution curves

When the point $f^n(t)$ is moved by $\Delta \mathbf{d}$, we obtain its final position $\tilde{f}^n(t) = f^n(t) + \Delta \mathbf{d}$. If the displacement vector $\Delta \mathbf{d}$ is applied in $f^{n-1}(t_1)$, the corresponding point on f^n is displaced too. The relation between t and t_1 is explained in section 4.

The relation between points of two levels of resolution could be given by

$$f^{n-\lambda}(t) = f^{n-1}(t) + (1 - \lambda)g^{n-1}(t), \quad \text{where } 0 \leq \lambda \leq 1.$$

So, for one point $f^{n-1}(t)$ and $g^{n-1}(t)$, it is possible to compute $f^{n-\lambda}(t)$ for $0 \leq \lambda \leq 1$.

Therefore, when a point $f^{n-1}(t_1)$ is displaced in $\Delta \mathbf{d}$, the corresponding point on f^n is displaced, after reconstruction, in $\Delta \mathbf{d}$,

$$\begin{aligned} f^{n-\lambda}(t) + \Delta \mathbf{d} &= f^{n-1}(t) + \Delta \mathbf{d} + (1 - \lambda)g^{n-1}(t) \\ \tilde{f}^{n-\lambda}(t) &= \tilde{f}^{n-1}(t) + (1 - \lambda)g^{n-1}(t). \end{aligned}$$

In particular, for $\lambda = 0$, we have

$$\tilde{f}^n(t) = \tilde{f}^{n-1}(t) + g^{n-1}(t),$$

where we could observe that the segment of the curve projected on level n , from the curve based on wavelet coefficients, g^{n-1} , is not affected by the displacement of point $f^{n-1}(t)$ by $\Delta \mathbf{d}$. However, it is necessary to preserve the original details characteristic of f^n , and thus we need to preserve the original orientation of the coefficient details, d^{n-1} , affected by the deformation of f^{n-1} . Section 3.3 gives further details on this.

The effects of the displacement vector, $\Delta \mathbf{d}$, applied on a point in a low-resolution curve, as showed above, is formalized by a theorem.

Theorem 1. *The point $f^n(t)$ displaced by one vector $\Delta \mathbf{d}$ becomes another point, $p = f^n(t) + \Delta \mathbf{d}$ equivalent to the displacement of $f^{n-1}(t_1)$, the projection of $f^n(t)$ on f^{n-1} , when displaced by the same vector $\Delta \mathbf{d}$ projected on $f^n(t)$.*

Proof: Let $\Delta \mathbf{d} = \Delta f^n(t)$ be the displacement vector applied onto $f^n(t)$. A function f^n is reconstructed from their low-resolution coefficients as

$$f^n(t) = \Phi^n(t)(P^n C^{n-1} + Q^n D^{n-1}).$$

If we displace one point onto f^{n-1} , corresponding to $f^n(t)$, by $\Delta \mathbf{d}$, some relative coefficients of C^{j-1} are moved, as (3.1), in ΔC^{n-1} . After that, the expression above will be as

$$\tilde{f}^n(t) = \Phi^n(t) (P^n (C^{n-1} + \Delta C^{n-1}) + Q^n D^{n-1}).$$

Considering the expression (3.1) as $\Delta C^{n-1} = \Delta \mathbf{d} \frac{\Phi^{n-1}}{\Phi^{n-1} \cdot \Phi^{n-1}}$, because so k (curve degree) basis relating to the point are different from zero (subsection 2.1); the deformed curve can be rewritten as

$$\begin{aligned}
 \tilde{f}^n(t) &= \Phi^n(t) \left(P^n \left(C^{n-1} + \Delta d \frac{\Phi^{n-1}}{\Phi^{n-1}, \Phi^{n-1}} \right) + Q^n D^{n-1} \right) \\
 &= \Phi^n(t) (P^n C^{n-1} + Q^n D^{n-1}) + \Phi^n(t) P^n \Delta d \frac{\Phi^{n-1}}{\Phi^{n-1}, \Phi^{n-1}} \\
 &= \Phi^n(t) (P^n C^{n-1} + Q^n D^{n-1}) + \Phi^{n-1} \Delta d \frac{\Phi^{n-1}}{\Phi^{n-1}, \Phi^{n-1}} \\
 &= f^n(t) + \Delta d.
 \end{aligned}$$

◇

The theorem 1 is generalized to displace one point on arbitrary low-resolution curves, $f^j, j < n$. Another consequence of this theorem refers to the case in which the displacement vector Δd is segmented in various vectors, $\Delta d_1, \Delta d_2, \dots, \Delta d_m$, and each vector Δd_k is applied on a respective point of low-resolution curves, f^j . After reconstruction, the point $f^n(t)$ displaced is the same as $f^n(t) + \Delta d$. These extensions are formalized by corollaries.

Corollary 1. *If $p = f^n(t) + \Delta d$ is a displaced point of $f^n(t)$ by Δd , the same effect is obtained by moving $f^j(t_j), 0 < j < n$, corresponding to $f^n(t)$, by Δd .*

Proof: The function f^n could be expressed using their low-resolution coefficients and all wavelet coefficients as

$$f^n = \Phi^j C^j + \sum_{i=j}^{n-1} \Psi^i D^i.$$

After the point $f^j(t_j)$ is displaced by Δd , some relative coefficients of C^j are moved as (3.1), resulting, in general in ΔC^j ; thus, the expression above is rewritten as

$$\begin{aligned}
 \tilde{f}^n(t) &= \Phi^j(t) (C^j + \Delta C^j) + \sum_{i=j}^{n-1} \Psi^i D^i \\
 &= \Phi^j(t) C^j + \Phi^j(t) \Delta C^j + \sum_{i=j}^{n-1} \Psi^i D^i \\
 &= \Phi^j C^j + \sum_{i=j}^{n-1} \Psi^i D^i + \Delta d \\
 &= f^n(t) + \Delta d.
 \end{aligned}$$

◇

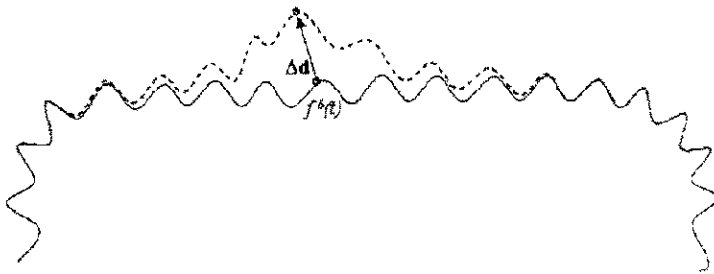
Corollary 2. *The displacement of point $f^n(t)$ by Δd is equivalent to $p = f^n(t) + \Delta d$. If Δd is segmented in $\Delta d_0, \Delta d_1, \dots, \Delta d_m$, and each Δd_i displaces one corresponding point of $f^n(t)$ onto low-resolution curves $f^{n-i}, 0 \leq i \leq m$, each one by time, the point $f^n(t)$ is moved to p .*

The proof of the corollary 2 is immediate. If we displace one point $f^{n-i}(t_i)$, corresponding to $f^n(t)$, in Δd_i once by each level, the corollary (1) ensures that $f^n(t)$ is moved in Δd_i ; then, the total amount of displacements of $f^n(t)$ is $\Delta d = \Delta d_0 + \Delta d_1 + \dots + \Delta d_m$ finally.

The figure 3(a) shows the results of corollary (1) after displacing the point $f^6(t) = (333.72, 506.37)$ by $\Delta d = (-19.72, 64.62)$, interior dashed curve. The curve highest-level is $n = 6$. When a corresponding point of $f^6(t)$ on f^3 is displaced by Δd , the final position of $\tilde{f}^6(t)$, after reconstruction, is the same (external dotted curve of the figure). The figure ??(b) shows the effect after reconstruction when the displacement vector Δd is segmented in various vectors, $\Delta d_1, \Delta d_2, \Delta d_3$ and Δd_4 (corollary (2)), and $f^6(t)$ displaced by Δd_1 , $f^5(t_5)$ by Δd_2 , $f^4(t_4)$ by Δd_4 and $f^3(t_3)$ by Δd_3 , as explained in section 3.1.



(a)



(b)

Figure 3. Displacement vector Δd applied onto low-resolution curves: (a)

Effects of displacement point by Δd onto $f^j(t_j)$, $3 \leq j \leq 6$, once in each resolution (external dotted curve is after the point $f^3(t_3)$ is displaced, the internal dashed curve is after the point $f^6(t)$ is displaced); (b) dashed curve is produced after displacing the segmented vectors, $\Delta d_1, \Delta d_2, \dots, \Delta d_m$, which are applied onto each low-resolution curves, f^6, f^5, f^4 and f^3 .

3.3 Orientation of detail

Since each detail coefficient d_s^j of D^j is composed by two elements $(\Delta x_s^j, \Delta y_s^j)$, it can be treated as a two-dimensional vector δ_s^j . For each δ_s^j it will be assigned a Local Coordinate System (LCS). Any operation with the scaling coefficients could produce local operations with relative detail coefficients, such as local rotation.

When an object is deformed, it is necessary to keep detail characteristics controlling its orientation. In this way a real deformation effect is achieved, which would be difficult to obtain if the operation of rotation were executed in the Global Coordinate System (GCS). These principles were suggested in [14, 3, 4, 18], with different purposes.

One alternative for the representation of the LCS for each δ_s^j is to use a similar method, formulated by Forsey and Bartels [4], considering the x -axis tangent to the curve of level $j - 1$ in t_0 , where the wavelets function $\psi_1^j(t_0)$ has the maximum value. In this work, we considered the x -axis of the LCS of each element δ_s^{j-1} as being parallel to the edge s of the control polygon of f^{j-1} formed from point c_s^{j-1} to c_{s+1}^{j-1} . The y -axis is a vector orthogonal to the edge s .

We use polar coordinates to express detail coefficients, which are computed in each decomposition process. The polar coordinates for detail δ_s^{j-1} is composed by the pair $(r_s^{j-1}, \theta_s^{j-1})$, where r_s^{j-1} is the module of δ_s^{j-1} and θ_s^{j-1} is the angle formed by δ_s^{j-1} with the x -axis of their LCS. This process is linear relative to the number of scaling coefficients, so it does not change the computational complexity of the algorithm.

4. MULTIREOLUTION DISTRIBUTION OF DEFORMATIONS

A point on the segment f_s^n influenced by $B_s^n = [c_{s-1}^n c_s^n c_{s+1}^n c_{s+2}^n]^T$, converges, just as the decomposition, to a point of segment f_h^{n-1} . The relation between their indices is given by $h = \lfloor s/2 \rfloor$. This allows us to relate adjacent segments of the same level n with another segment of level $n - 1$. All points of the segment f_h^{n-1} are influenced by their control point matrix $B_h^{n-1} = [c_{h-1}^{n-1} c_h^{n-1} c_{h+1}^{n-1} c_{h+2}^{n-1}]^T$. Thus, it is possible to establish recursively the correspondence between segments and control points of multiresolution curves.

We show, in figure 6, that the segment f_s^{j-1} with reference point c_s^{j-1} , is the convergence segment of the high-level segments f_{2s}^j and f_{2s+1}^j , whose reference points are c_{2s}^j and c_{2s+1}^j respectively. The relation between the control points of C^j and C^{j-1} was established in section 2.2.2, that shows how c_{2s}^j and its neighbourhood generate c_s^{j-1} .

To relate the detail coefficients of level j with level $j - 1$, we can derive an expression similar to the one obtained for scaling coefficients. Any action

that affects a point of segment f_{2s}^j or f_{2s+1}^j is transmitted to the vicinity of d_s^{j-1} , as its best representative. Detail coefficient d_s^{j-1} is related with the scaling coefficient c_{2s+1}^j . In general, f_s^j and d_h^{j-1} are related by the index expression $h = \lfloor (s+2)/2 \rfloor$.

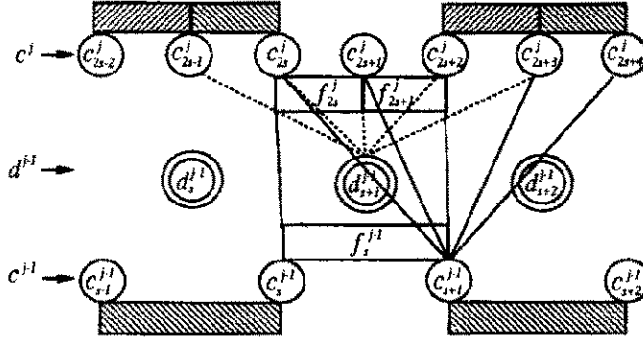


Figure 4. Convergence of curve segments from resolution j to resolution $j - 1$, and the relation of their coefficients.

As two adjacent segments of a level j curve converge to a segment of level $j - 1$, the point $p_{h_m} = f_h^j(t_{h_m})$ and its neighbourhood converges to a point $p_{s_i} = f_s^{j-1}(t_{s_i})$, where the respective local deformation will be applied. The relation between the index h and s is given by $s = \lfloor h/2 \rfloor$. The parameter t_{h_m} is known, while t_{s_i} is calculated by analyzing the convergence relation of segments from level j to $j - 1$. We compute t_{s_i} using the propagation proportion, because two adjacent segments, even and odd, of f^j converge to another segment of f^{j-1} . The parameter t_{s_i} is calculated as

$$t_{s_i} = t_{h_m}/2 + (h/2 - s), \text{ with } s = \lfloor h/2 \rfloor,$$

and the parameter t varies as $0 \leq t_{s_i} < 1$ in all segments.

5. DISTRIBUTION FUNCTION FOR DEFORMATION

In this section we discuss how to distribute a certain displacement vector Δd to all resolution levels of the curve. It is natural that modifications at higher resolution levels be larger than those at lower levels. An appropriate way of dealing with this problem is by considering a continuous distribution function $g(t)$, defined on $[0, 1]$ (with 0 corresponding to the highest level and 1 to the lowest), such that $\int_0^1 g(t)dt = 1$. The proportion of displacement corresponding to level i will be given by:

$$a_i = \int_{\frac{i}{m}}^{\frac{i+1}{m}} g(t)dt,$$

where m is the number of levels. The displacement applied at level $n - i$ will be given by

$$\Delta d_i = a_i \Delta d.$$

We adopt the distribution function of the form

$$g(t) = \frac{1}{A}e^{-wt}, \quad \text{where} \quad A = \int_0^1 e^{-wt} dt.$$

The parameter w determines how fast the amount of change decreases at the less refined levels. Figure 5 shows how the choice of w influences the behavior of the deformation.

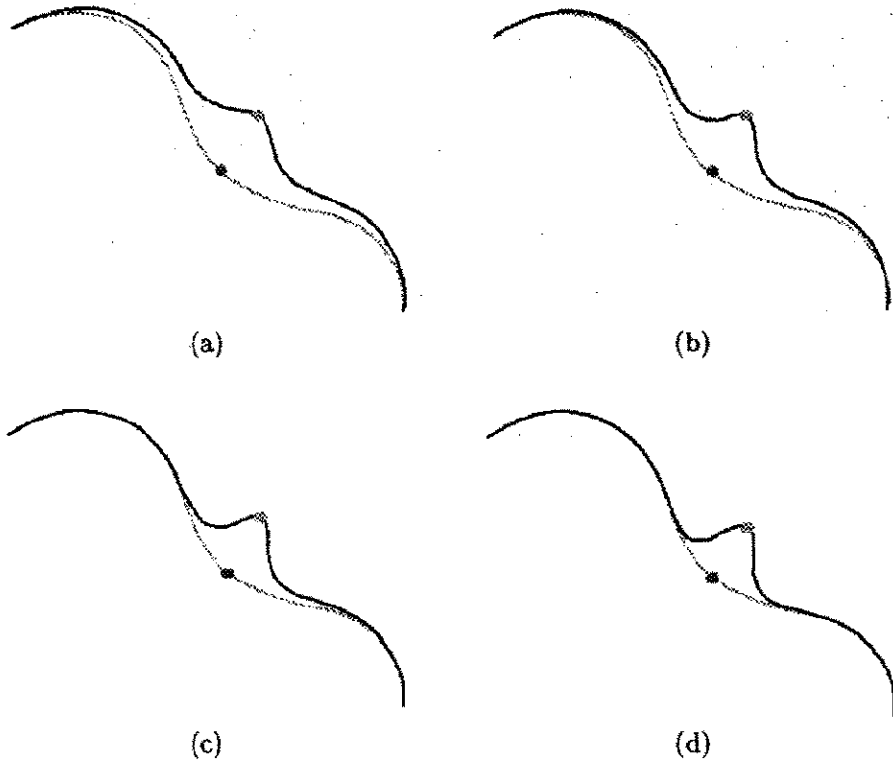


Figure 5. A curve deformed in several ways depending of the weight w that acts as the stiffness of the curve. (a) $w = 1$; (b) $w = 3$; (c) $w = 5$; and (d) $w = 12$.

6. CONCLUSION AND FUTURE WORK

We described a technique for manipulation of multiresolution B-spline curves. This technique could be used for direct manipulation of curves and surfaces represented in multiresolution. A designer, using our method, can model interactively the curve, by deforming segments of variable size defined by a local parameter such as a stiffness constant. In this way, he does not need to deal with non-intuitive parameters as required by others techniques.

Our method conserves the initial number of control points, because it does not use any knot operations. Thus, the same space used to store the control points are used in the manipulation of the object in multiresolution. One advantage of the wavelet transform is the conservation of the space used in all decomposition and reconstruction process.

The technique described in this work may be generalized to automatic deformation of objects, considering the extern agent as a force generated by collision between objects, as proposed in [9], for physical simulation effects. Another work could be to extent this approach using geometric constraints formulated in [5, 6] to obtain deformation of curves with geometric restrictions.

BIBLIOGRAFIA

- [1] R. Bartels and J. Beatty, A Technique for the Direct Manipulation of Spline Curves, *Graphics Interface'89 (Proceedings)*, 1989, 33-39.
- [2] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A practical guide*, San Diego: Academic Press Inc, 1990.
- [3] A. Finkelstein and D. Salesin, Multiresolution Curves, *Computer Graphics Proceedings (SIGGRAPH'94)*, 1994, 261-268.
- [4] D. Forsey and R. Bartels, Hierarchical B-spline Refinement, *Computer Graphics Proceedings (SIGGRAPH'88)*, 1988, 205-212.
- [5] B. Fowler and R. Bartels, Constraint Based Curve Manipulation, *IEEE Computer Graphics and Applications*, 13(5), 1993, 43-49.
- [6] B. Fowler, Geometric Manipulation of Tensor Product Surfaces, *Computer Graphics*, 22(4), 1992, 101-108.
- [7] J. Gomes and L. Velho, From Fourier Analysis to Wavelets, *Course Notes (SIGGRAPH'98)*, 1998.
- [8] Z. Guan, J. Ling, N. Tao, X. Ping and T. Rongxi, Study and Application of Physics-Based Deformable Curves and Surfaces, *Computer & Graphics*, 21(3), 1997, 305-313.
- [9] W. Hsu, J. Hughes and H. Kaufman, Direct Manipulation of Free-Form Deformations, *Computer Graphics Proceedings (SIGGRAPH'92)*, 1992, 177-184.
- [10] R. Kazinnik and G. Elber, Orthogonal Decomposition of Non-Uniform B-spline Space Using Wavelets, *Computer Graphic Forum (Eurographics'97)*, 16(3), 1997, c26-c38.
- [11] T. Lyche and K. Morken, Knot Removal for Parametric B-spline Curves and Surfaces, *Computer Aided Geometric Design*, 4(2), 1987, 217-231.
- [12] L. Piegl and W. Tiller, *The NURBS Book*, New York: Springer Editions, 1995.
- [13] D. Rogers and J. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill International Editions, 1990.
- [14] E. Stollnitz, T. Deroose and D. Salesin, *Wavelets for Computer Graphics: Theory and applications*, California: Morgan Kaufmann Publishers, Inc, 1996.
- [15] W. Sweldens and P. Schroder, Building Your Own Wavelets at Home, *Technical Report 1995.5*, Department of Mathematics, Univ. of South Carolina, 1995.
- [16] S. Takahashi, Variational Design of Curves and Surfaces Using Multiresolution Constraints, *The Visual Computer*, 14(5/6), 1998, 208-227.

- [17] S. Takahashi, Y. Shinagawa and T. Kunii, Continuous-Resolution-Level Constraints in Variational Design of Multiresolution Shapes, *The Visual Computer*, 14(4), 1998, 177-192.
- [18] G. Thurmer and A. Wuthrich, Normal Computation for Discrete Surfaces in 3D Space, *Computer Graphic Forum (Eurographics'97)*, 16(3), 1997, c15-c26.
- [19] W. Wesselink, Variational Modeling of Curves, *PhD Thesis*, Department of Computer Science, Eindhoven University of Technology, 1996.
- [20] W. Welch, Serious Putty: Topological Design for Variational Curves and Surfaces, *PhD Thesis*, Department of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1995.