Area preserving deformation of multiresolution curves

Stefanie Hahmann[*], *Basile Sauvage*[†], *Georges-Pierre Bonneau*[‡]

February 2, 2005

— Article to be published in Computer Aided Geometric Design —

Abstract

We describe a method for multiresolution deformation of closed planar curves that keeps the enclosed area constant. We use a wavelet based multiresolution representation of the curves that are represented by a finite number of control points at each level of resolution. A deformation can then be applied to the curve by modifying one or more control points at any level of resolution. This process is generally known as multiresolution editing to which we add the constraint of constant area. The key contribution of this paper is the efficient computation of the area in the wavelet decomposition form: the area is expressed through all levels of resolution as a bilinear form of the coarse and detail coefficients, and recursive formulas are developed to compute the matrix of this bilinear form. A similar result is also given for the bending energy of the curve. The area constraint is maintained through an optimization process. These contributions allow a real time multiresolution deformation with area constraint of complex curves.

Keywords: multiresolution analysis, curves, multiresolution editing, constrained deformation, area preservation, wavelets.

1 Introduction

Multiresolution analysis has received considerable attention in recent years in many fields of computer graphics, geometric modeling and visualization. It provides a powerful tool

---

[*]Laboratoire LMC-IMAG, Institut National Polytechnique de Grenoble, France `Stefanie.Hahmann@imag.fr`

[†]Laboratoire LMC-IMAG, Institut National Polytechnique de Grenoble, France `Basile.Sauvage@imag.fr`

[‡]Laboratoire GRAVIR-IMAG, Université Joseph Fourier Grenoble, France `Georges-Pierre.Bonneau@imag.fr`

for efficiently representing functions at multiple levels of detail. Herein, a complex function is decomposed into a "coarser" low resolution part, together with a collection of detail coefficients, necessary to recover the original function. Deformation of complex objects with a lot of detail can be tricky and expensive to compute. In a multiresolution setting however complex objects can be edited at a chosen scale with mainly two effects: First, modifying some low-resolution control points and add back the details modifies the overall shape of the object. Second, modifying a set of fine detail coefficients modifies the character of the object without affecting its overall shape.

General multiresolution editing or deformation techniques for parametric curves have been explored in detail by Finkelstein and Salesin [12] using B-splines. In contrast, constraint multiresolution editing techniques have not been explored so much in the past. However, there are application areas, like computer animation, where deformations of soft objects under constraints are needed. An example are physical based deformations that incorporate simplified physical laws into the deformation process [22, 7]. Many physical based deformations are related to some geometric properties of curves and surfaces, e.g. area or volume preserving deformations for curves and surfaces simulating inelastic material [19], or length preserving deformations (bending) in cloth simulation [1, 8]. In cases where low computation time has priority over the physical exactness of the animation, it could be advantageous to use geometric constraints instead of time consuming numerical integration of physical differential equations. Fast algorithms can then be obtained by linearizing the non-linear geometric constraints. This is one of the purposes of the present paper. Other application areas could be geometric modeling with shape preservation, or in computer vision, multi-scale representations and smoothing of boundaries of bounded planar shapes without shrinkage [20].

The present paper introduces a wavelet based multiresolution curve editing/deformation environment that allows to satisfy a non linear constraint related to the area enclosed by a planar periodic curve. Area preservation is known as an important visual effect for realistic-looking computer animations of deformable objects as pointed out by Lasseter [15]. Closely related work has been previously done by Elber. In [10] he shows how it is possible to linearize the area constraint. The displacement vector of a curve's control point, which defines the deformation, is decomposed into several small x- and y-steps, such that for each step the quadratic area constraint becomes linear. The resulting algorithm for curve deformation is therefore interactive. Later Elber [11] developed a MR version of that algorithm. Herein the user is able to edit a curve at an arbitrary coarse scale. All details finer than the scale at which the editing takes place are preserved. The editing takes place by adding a vector function defined at some scale. In contrast with the present work, the algorithm developed in [11] is not based on a wavelet MR analysis of the curve. In our setting the curve is first decomposed in a MR-form, and then the editing is performed by modifying this decomposition at some scale. Therefore, the present framework can benefit from all advantages inherent to MR-decomposition, including compression, Level of Detail (LOD) displaying, progressive transmission and LOD editing, while [11] aims solely at LOD editing.

In this paper an efficient computation of the area in the wavelet decomposition form is introduced. Herein the area is expressed through all levels of resolution as a bilinear form of the coarse and detail coefficients. We develop recursive fomulas to compute the matrices of this bilinear form. The area constraint is maintained through an optimization process. Several deformation techniques are then introduced, which make benefit from the multiresolution decomposition of the curve and which ensure that the enclosed area of the curve is always kept unchanged. All presented formulas and algorithms are independent of the particular multiresolution representation of the curve. They apply to any kind of wavelet based multiresolution curves, e.g. B-spline wavelets. All algorithms we present have been developed with the objective of getting a real-time interactive editing tool. Simplicity and speed are crucial.

The paper is organized as follows. Section 2 gives the basics on multiresolution curves. Section 3 explains how to compute efficiently the area enclosed by a curve given in a multiresolution setting. Section 4 introduces the general method for deforming multiresolution curves subject to area constraints. Section 5 presents some extensions of the method. Finally Section 6 concludes and gives some possible future work.

2   Multiresolution curves

Wavelets have found a wide variety of applications in the last years: numerical analysis [2], signal analysis [18], image processing [17, 6, 16], visualization [14, 3] and computer graphics [21, 13]. The wavelet based multiresolution analysis framework we want to sketch in the present section is slightly more general than the classical multiresolution analysis introduced by Mallat [17]. This section is not intended to give a complete presentation of multiresolution curves. More details can be found in [21, 12].

Suppose we have a certain functional space E and some nested linear *approximation spaces* $V^j \subset E$ with $V^0 \subset V^1 \subset \cdots \subset V^n$. Since we are dealing with closed curves, these spaces have finite dimension. Let V^j be spanned by a set of basis functions $(\varphi_i^j)_i$, 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 $(\psi_i^j)_i$ are such that together with φ^j they form a basis of V^{j+1}. The functions ψ_i^j are called *wavelets*. The space V^n can therefore be decomposed as follows:

$$V^n = V^{n-1} \oplus W^{n-1} = V^{n-2} \bigoplus_{j=n-2}^{n-1} W^j = \cdots = V^0 \bigoplus_{j=0}^{n-1} W^j. \tag{1}$$

Condition (1) implies that the scaling functions are refinable, that is for all $j \in \{0, \ldots, n\}$ there must exist some matrices P^j and Q^j such that the following *refinement equations*

hold

$$\varphi^{j-1} = (P^j)^T \varphi^j$$
$$\psi^{j-1} = (Q^j)^T \varphi^j. \tag{2}$$

On the other hand, the "fine" scaling functions φ^j can be constructed from the coarser scaling functions and wavelets with the aid of some matrices A^j and B^j:

$$\varphi^j = (A^j)^T \varphi^{j-1} + (B^j)^T \psi^{j-1}. \tag{3}$$

Note that $[P^j \mid Q^j]$ and $\begin{bmatrix} A^j \\ B^j \end{bmatrix}$ are both square matrices, and that

$$\begin{bmatrix} P^j & Q^j \end{bmatrix} \begin{bmatrix} A^j \\ B^j \end{bmatrix} = I. \tag{4}$$

(4) is sometimes referred to as the *reconstruction condition*. The choice of the scaling functions determines the structure of the matrices P^j, Q^j, A^j, and B^j . Sparse matrices are desirable for most of the applications.

Based on this framework let a *multiresolution (MR) curve* $\mathbf{c}(t)$, $t \in I$ be defined as an element of a certain functional space V^n. Let $D2^n$ be the dimension of the space V^n, with $D = dim(V^0)$. $\mathbf{c}(t)$ can be written as:

$$\mathbf{c}(t) = \sum_{i=0}^{D2^n-1} x_i^n \varphi_i^n = (\mathbf{x}^n)^T (\varphi^n), \tag{5}$$

where $\mathbf{x}^n$ is a column vector of $D2^n$ scaling coefficients $x_0^n, \ldots, x_{D2^n-1}^n \in \mathbb{R}^2$. In a more general setting, the vector of scaling coefficients $\mathbf{x}^n$ can be thought of as a discrete signal with $D2^n$ samples.

The relations (2) and (3) allow now to create a low-resolution signal $\mathbf{x}^{n-1}$ with less samples by using the low pass filter matrix A^n:

$$\mathbf{x}^{n-1} = A^n \mathbf{x}^n,$$

where A^n is of size $(D2^{n-1} \times D2^n)$. The details which have been lost in this filtering process can be captured as another signal $\mathbf{d}^{n-1}$ using the high pass filter matrix B^n:

$$\mathbf{d}^{n-1} = B^n \mathbf{x}^n,$$

where B^n is a $(D2^{n-1} \times D2^n)$ matrix. The process of splitting a signal $\mathbf{x}^n$ into a coarser signal $\mathbf{x}^{n-1}$ and detail $\mathbf{d}^{n-1}$ is called **decomposition**. The matrices A^n and B^n are called *decomposition (analysis) filters*. Decomposition can now recursively be repeated with the new signal $\mathbf{x}^{n-1}$. Finally, the original signal will be decomposed into a low-resolution signal $\mathbf{x}^0$ and details $\mathbf{d}^0, \ldots, \mathbf{d}^{n-1}$. This recursive process is known as *filter bank* [17],

see Table 1. At any intermediate level of resolution L the original signal is decomposed into a coarser signal $\mathbf{x}^L$ and details $\mathbf{d}^L, \ldots, \mathbf{d}^{n-1}$. Note that the vector $\mathbf{d}^j$ is of size $D2^j$, $j = 0, \ldots, n-1$.

Using the matrices P^j and Q^j, also called *reconstruction (synthesis) filters*, the original signal $\mathbf{x}^n$ can be reconstructed recursively with

$$\mathbf{x}^j = P^j \mathbf{x}^{j-1} + Q^j \mathbf{d}^{j-1} \qquad \text{for } j = 1, \ldots, n.$$

This process is called **reconstruction**.

$$(\mathbf{x}^n) \ \longrightarrow \ (\mathbf{x}^{n-1}) \ \longrightarrow \ \cdots \quad (\mathbf{x}^1) \ \longrightarrow \ (\mathbf{x}^0)$$

$$\searrow \qquad\qquad \searrow \qquad\qquad\qquad \searrow$$

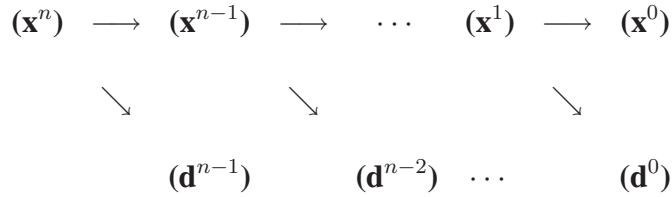$$(\mathbf{d}^{n-1}) \qquad (\mathbf{d}^{n-2}) \ \cdots \qquad (\mathbf{d}^0)$$

Figure 1: Filter bank.

Since the filter bank applies also to the scaling coefficients of a multiresolution curve (5), such a curve can be represented at any *level of resolution* $L \in \{0, \ldots, n\}$ by some coarse coefficients $\mathbf{x}^L$ that form approximations of the initial coefficients and by the detail coefficients $\mathbf{d}^L, \ldots, \mathbf{d}^{n-1}$ as follows:

$$\mathbf{c}(t) = (\mathbf{x}^L)^T(\varphi^L) + (\mathbf{d}^L)^T(\psi^L) + \cdots + (\mathbf{d}^{n-1})^T(\psi^{n-1}), \quad L = 0, \ldots, n. \qquad (6)$$

In the examples of this paper, we use a multiresolution representation of quadratic uniform parametric B-spline curves. In this setting the scaling coefficients are usually called *control points* defining the vertices of the so-called control polygon. The decomposition of a curve with $D2^n$ control points $x_0^n, \ldots, x_{D2^n-1}^n$ is given by the following coarse control points and detail coefficients:

$$
\begin{aligned}
x_i^{n-1} &= \tfrac{1}{4}(-x_{2i-2}^n + 3x_{2i-1}^n + 3x_{2i}^n - x_{2i+1}^n) \\
d_i^{n-1} &= \tfrac{1}{4}(x_{2i-2}^n - 3x_{2i-1}^n + 3x_{2i}^n - x_{2i+1}^n),
\end{aligned}
\qquad (7)
$$

where all indices are taken modulo the number of control points. The reconstruction formula is given by:

$$
\begin{aligned}
x_{2i}^n &= \tfrac{3}{4}(x_i^{n-1} + d_i^{n-1}) + \tfrac{1}{4}(x_{i+1}^{n-1} - d_{i+1}^{n-1}) \\
x_{2i+1}^n &= \tfrac{1}{4}(x_i^{n-1} + d_i^{n-1}) + \tfrac{3}{4}(x_{i+1}^{n-1} + d_{i+1}^{n-1}).
\end{aligned}
\qquad (8)
$$

Figure 2 illustrates the successive coarse approximations of a quadratic B-spline curve using the decomposition formulas (7).

Remark: If all detail coefficients d_i^n are zero then (8) corresponds to the Chaikin subdivision algorithm [4].
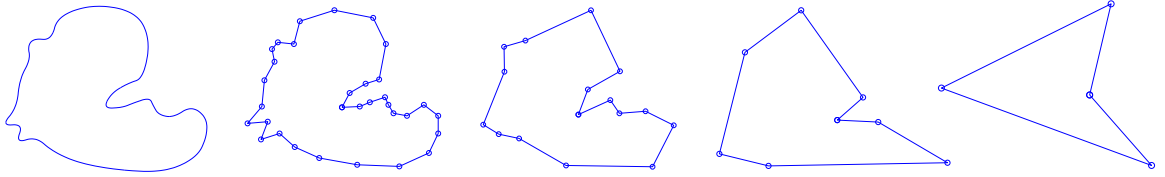
5

Figure 2: From left to right, a B-spline curve defined by 32 control points, its control polygon (level 5), the control polygons of its coarse approximations at levels 4, 3 and 2.

3 Computing the area of a MR-curve

The present paper deals in particular with periodic planar MR-curves $\mathbf{c}(t) = (x(t), y(t))$. Initially, at the highest level of resolution, a MR-curve is defined by $D2^n$ control points. After a complete decomposition it is represented by D coarse control points and $D(2^n-1)$ detail coefficients. In the following sections we want to develop a tool for area preserving deformation of these curves. For this aim we need to be able to evaluate the area of curves at any level of resolution. The present section derives the formulas for this computation (Section 3.1), explains how to do the computation efficiently (Section 3.2), and gives the actual formulas for the specific MR representation used in the examples (Section 3.3).

3.1 Area of a MR-curve

Let $\mathbf{c}(t) = (x(t), y(t))$ be a periodic planar freeform curve without self-intersections and which is sufficiently smooth so that all relevant derivatives may be defined. The (signed) area enclosed by $\mathbf{c}(t)$ is then given using Green's theorem by [10, 9]

$$\mathcal{A} = \frac{1}{2} \oint x(t)y'(t) - x'(t)y(t)dt. \tag{9}$$

For the following developments we need first some notations. Let us introduce the bilinear form

$$I(\varphi, \psi) := \left(\quad I(\varphi_i, \psi_j)_{ij} \quad \right), \tag{10}$$

where φ and ψ are two vectors of functions and $I(\varphi_i, \psi_j) = \oint \varphi_i(t)\psi_j'(t) - \varphi_i'(t)\psi_j(t)dt$.

The area (9) of a multiresolution curve (6) can now be evaluated at any level of resolution L. One obtains the bilinear equation

$$2\mathcal{A} = (\mathbf{X}^L) \left[M^L \right] (\mathbf{Y}^L)^T, \quad \forall L \in \{0, \dots, n\}, \tag{11}$$

6

where X^L and Y^L are the line vectors of the x- and y-coordinates resp. of all $D2^n$ coefficients (coarse and wavelet coefficients) of the MR representation of the curve, i.e.

$$\begin{pmatrix} \mathbf{X}^L \\ \mathbf{Y}^L \end{pmatrix} = (\mathbf{x}^L, \mathbf{d}^L, \mathbf{d}^{L+1}, \ldots, \mathbf{d}^{n-1}),$$

and

$$M^L = \begin{bmatrix} I(\varphi^L, \varphi^L) & I(\varphi^L, \psi^l)_{l=L}^{n-1} \\ I(\psi^k, \varphi^L)_{k=L}^{n-1} & I(\psi^k, \psi^l)_{k,l=L}^{n-1} \end{bmatrix} = \begin{bmatrix} A & B \\ -B^T & C \end{bmatrix}. \tag{12}$$

The area matrix M^L is a skew-symmetric matrix of size $D2^n \times D2^n$ consisting of 4 main blocks. Bloc A is of size $D2^L \times D2^L$, block B is of size $D2^L \times (D2^n - D2^L)$, and block C is of size $(D2^n - D2^L) \times (D2^n - D2^L)$. Figure 3 illustrates the composition of the four blocks which themselves contain several blocks each.
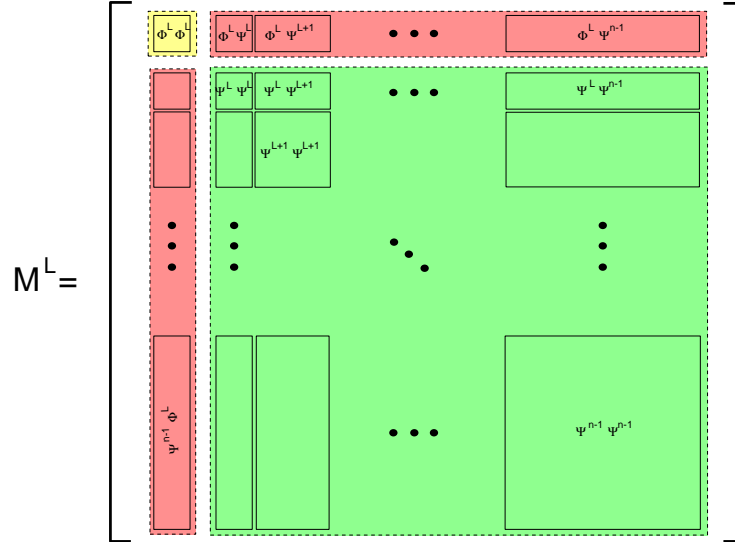

Figure 3: The matrix M^L that appears in the area formula (11) is composed of 4 main blocks. The yellow block which is labelled $(\varphi^L \varphi^L)$ meaning $I(\varphi^L, \varphi^L)$ corresponds to block A in formula (12) and involves only the scaling functions φ_i^L of level L. The red blocks labelled $(\varphi^L \psi^k)$ involve scaling functions at level L and wavelets of level k, with k varying from level L to $n-1$. The elements of the green blocks are calculated with (10) by using only the wavelets ψ^i of level $i \in \{L, \ldots, n-1\}$.

The size of the 4 blocks varies with the level of resolution L of the curve, see Figure 4. Note that in the case $L = n$ (highest level of resolution), the matrix is composed of only one block. This is the area matrix for the original curve that is expressed in terms of the highest resolution scaling functions φ^n.
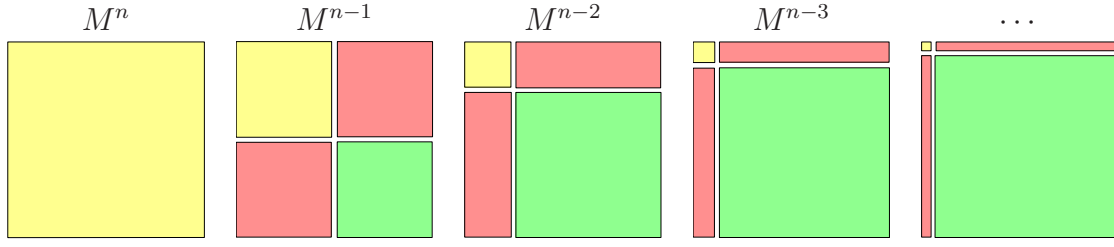
Figure 4: The area matrices M^L with $L \in \{0, \ldots, n\}$ have the same size $D2^n \times D2^n$ but their decomposition into the four blocks varies with level L.

3.2 Efficient computation of the area matrices

For all n levels of resolution a $D2^n \times D2^n$ area matrix has to be computed. An unnecessary high computational effort would be needed if one would calculate each matrix using formula (12). Indeed each element of the matrices needs an evaluation of an integral of type (9). If we set $N = 2^n$, the total number of integral evaluations would be $O(log(N) \cdot N^2)$. In the case of multiresolution curves based on uniform B-splines, the integrals can be evaluated explicitly, otherwise some quadrature formula must be used.

It turns out that explicit evaluation, or the evaluation using quadrature formula, of the elements given in (10) is computationally expensive. In the rest of this section, it is shown that only the integrals in the finest level matrix M^n have to be evaluated (either explicitly or using a quadrature formula). All other elements for the resolution levels $n - 1, \ldots, 0$ can thereafter be computed using simple recursive formulas from the values of the integrals at the finest level.

The refinement equations (2) of Section 2 state that the scaling functions and the wavelets at level $L - 1$ can be expressed in terms of the scaling functions at level L. By bilinearity of the functional (10) the elements of the area matrix M^{L-1} can be computed recursively from the area matrix of level L by using these refinement equations.

In the case where one argument of the bilinear form I is fixed one gets the following recursive relations (k is fixed). We call them (P) or (Q) filters depending on the refinement filter which is used:

$$\textbf{(P) - filter:} \qquad I(\varphi^{L-1}, \psi^k) = (P^L)^T \, I(\varphi^L, \psi^k) \qquad (13)$$

$$\textbf{(Q) - filter:} \qquad I(\psi^{L-1}, \psi^k) = (Q^L)^T \, I(\varphi^L, \psi^k). \qquad (14)$$

By symmetry one gets analogous formulas in the case where the first argument stays fixed. In the case where both arguments need to be computed recursively, the refinement equations result in the following recursive relations:

$$\textbf{(PP) - filter:} \qquad I(\varphi^{L-1}, \varphi^{L-1}) = (P^L)^T \, I(\varphi^L, \varphi^L) \, (P^L) \qquad (15)$$

$$\textbf{(PQ) - filter:} \qquad I(\varphi^{L-1}, \psi^{L-1}) = (P^L)^T \, I(\varphi^L, \varphi^L) \, (Q^L) \qquad (16)$$

(QQ) - filter:
$$I(\psi^{L-1}, \psi^{L-1}) = (Q^L)^T \, I(\varphi^L, \varphi^L) \, (Q^L) \qquad (17)$$

In order to compute now matrix M^L, one has just to compute M^n by using the finest scaling functions and then to apply the recurrence formulas (13) - (17) in order to compute successively the area matrices $M^{n-1}, M^{n-2}, \ldots, M^L$. This procedure is illustrated in Figure 5 where the 4 blocks are always colored in yellow, red and green corresponding to Figure 3.

More precisely, for each $L = n, \ldots, 1$ the matrix M^{L-1} is computed by applying the (PP), (PQ), (QQ)-filters to the submatrix $I(\varphi, \varphi)$ of M^L (yellow block in Figure 5), and by applying the (P) and (Q)-filters to submatrix $I(\varphi, \psi)$ (red block in Figure 5) of M^L. Note that, once the $I(\psi, \psi)$ submatrix (green block) is computed it will not be modified anymore during the recusive computations. This will significantly reduce the computational time. Hence the pre-computation and storage of all matrices can be avoided. Any matrix can be rapidly recomputed when the analysis level changes during the edition process. In addition the stability of this computation is ensured by the stability of the filter bank process, i.e. by the stability of the chosen wavelet basis.
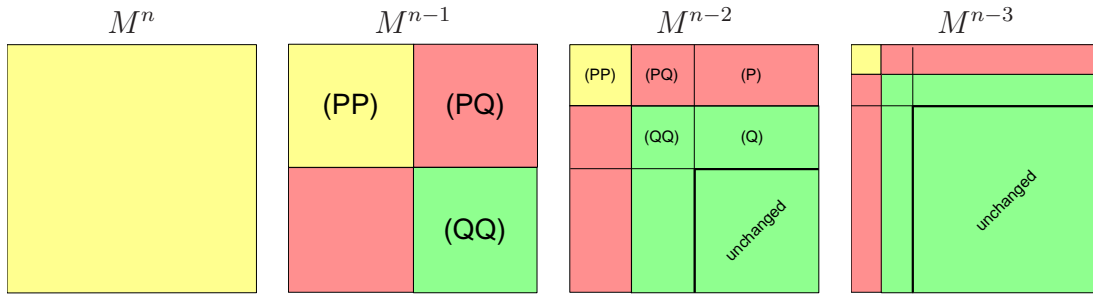

Figure 5: Recursive computation of area matrices by recursively computing the matrix M^{n-1} from M^n, M^{n-2} from M^{n-1} and so on.

3.3   Area matrices for the Chaikin MR-curves

Recall that the algorithms introduced in this paper work for all MR representations of curves following the framework of Section 2. Nevertheless for completeness we include here the area matrices for the specific MR representation based on the Chaikin subdivision curves (i.e. a wavelet representation where the scaling functions are quadratic uniform B-splines), used in the examples of the paper. If the MR representation is computed using

formulas (7) and (8), then the area matrix M^n equals:

$$M^n = \frac{1}{12}\begin{bmatrix} 0 & 10 & 1 & 0 & \cdots & & 0 & -1 & -10 \\ -10 & 0 & 10 & 1 & 0 & \cdots & & 0 & -1 \\ & \ddots & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & \ddots & & & 0 & \\ & 0 & & & \ddots & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & \ddots & \\ 1 & 0 & \cdots & & 0 & -1 & -10 & 0 & 10 \\ 10 & 1 & 0 & \cdots & & 0 & -1 & -10 & 0 \end{bmatrix}.$$

The coefficients of M^n are exact coefficients, obtained through an analytic integration of the piecewise polynomial functions. The reconstruction matrices P^n and Q^n are given by:

$$P^n = \frac{1}{4}\begin{bmatrix} 3 & 1 & 0 & \cdots & & 0 \\ 1 & 3 & 0 & \cdots & & 0 \\ 0 & 3 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 3 & 0 & \cdots & 0 \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ 1 & 0 & & \cdots & 0 & 3 \\ 3 & 0 & & \cdots & 0 & 1 \end{bmatrix} \qquad Q^n = \frac{1}{4}\begin{bmatrix} 3 & -1 & 0 & \cdots & & 0 \\ 1 & -3 & 0 & \cdots & & 0 \\ 0 & 3 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -3 & 0 & \cdots & 0 \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ -1 & 0 & & \cdots & 0 & 3 \\ -3 & 0 & & \cdots & 0 & 1 \end{bmatrix}.$$

4   Multiresolution area preserving deformation

Constraints can easily be incorporated into a freeform curve and surface interactive design system if they are linear, e.g. positional, tangential or orthogonality constraints. Non linear constraints, like area, volume, convexity or physically based constraints like energy, are more difficult to handle with, in particular because they require a high computational effort.

In this section we will present a method of deformation of planar periodic (i.e. closed) curves at any scale (level of resolution) that preserves the enclosed area. In an elegant and simple way it will be possible to

- preserve the details of the curves while globally deforming its shape, or

- deform the curve at different scales,

while preserving the area enclosed by the curve.

4.1 Overview

The deformation method mainly consists of three steps:

1. **decomposition:** First a decomposition of the curve down to a user specified level of resolution is done. The curve is now represented by a coarse polygon and a set of detail coefficients (see formula (6)).

2. **deformation:** Second, a deformation is defined. This deformation applies to the control points of the coarse polygon by moving one or several of them.

3. **area preservation:** Last, a new curve is computed that respects the defined deformation and that has the same enclosed area as the original curve. To this end, a slight correction of some coarse control points (and possibly of some detail coefficients), of the deformed polygon is made such that after reconstruction the new curve satisfies the constant area constraint. This step is important, because after deformation of the coarse polygon in the previous step, the reconstruction generally does not produce a curve with the same enclosed area.

Remark: Note that the area computation and area correction of the deformed curve is applied to its multiresolution representation at level L and not to its original representation. This implies the need of a multiresolution representation of the area, as we have developed before in Section 3.2

For a given MR-curve $\mathbf{c}(t)$ and a fixed level of resolution L at which the deformation will be applied, let us make the following notations by leaving out the indices L and n:

- $\mathcal{A}_{ref}$ is the reference area we want to preserve, i.e. the area enclosed by the given parametric curve $\mathbf{c}(t)$.

- $\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$ are the coordinates after deformation of the coarse polygon and before area preserving corrections.

- $\mathcal{A}_0 = \frac{1}{2} X_0 M Y_0^T$ is the corresponding new area that has to corrected.

- $\begin{pmatrix} X \\ Y \end{pmatrix}$ are the coordinates after correction.

- $\mathcal{A} = \frac{1}{2} X M Y^T$ is the new corresponding area.

In terms of these notations, an area preserving deformation method of an MR-curve can now be stated as follows:

- express curve $\mathbf{c}(t)$ in a multiresolution basis at level L, see formula (6).

- define deformation by bending the coarse polygon to get X_0 and Y_0.

- compute X and Y "close to X_0 and Y_0" such that $\mathcal{A} = \mathcal{A}_{ref}$.

Let us notice that we have only one scalar constraint and at least 2^n vector degrees of freedom. Hence a lot of ideas for defining a deformation satisfying this constraint can be realized. We are not going to talk about methods decreasing artificially the degrees of freedom with particular choices of deformations. This would however be an interesting issue for future work.

We rather focus on a broader method using optimization, i.e. minimization of a cost function that integrates a smoothness term and a distance term subject to the area constraint. The smoothness term prevents the resulting curve to have unwanted wiggles. The distance term is needed in order to respect the defined deformation as much as possible. Two further features of the present method are the possibility to choose between a local or a global deformation of the curve, and the possibility to modify the details additionally to the coarse coefficients.

4.2 Smoothness criteria

In variational design a physical model is used for the description of a "smooth" or "fair" curve or surface. It is based on the minimization of the bending energy. This most widely used fairness criteria originates from the observation that the shape of a thin elastic beam or plate under deformation which minimizes the bending energy is always smooth, i.e. has a visual pleasing shape.

Since the bending energy for a planar parametric curve, $E = \int \kappa^2(t)dt$ is a non-linear functional, it is common to use instead the linearized version of the bending energy

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

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

In the particular multiresolution setting of this paper, the smoothness criterion needs to be evaluated for a MR-curve (6) at any level of resolution $L \in \{0, \ldots, n\}$. Similar to the area formula (11), the energy of a curve can be expressed as the following bilinear form:

$$\mathcal{E}(X, Y) = \frac{1}{2}(X^L H^L (X^L)^T + Y^L H^L (Y^L)^T)$$

where $H^L = \oint \varphi^{L''}(t)^T \varphi^{L''}(t)dt$ is a symmetric matrix, for all $L \in \{0, \ldots, n\}$. $\varphi^{L''}(t) = [\varphi_0^{L''}, \ldots, \varphi_{D2^L-1}^{L}{}'']^T$ denotes the line vector of the second derivatives of the basis functions of V^L.

Similar to the developments of Section 3.2 concerning the area matrix, it can be shown that the same recurrence relations hold between H^{L-1} and H^L. The computation of all n energy matrices H^L, $L = 0, \ldots, n-1$ can therefore recursively be done by computing the initial matrix H^n and by applying the refinement filters (P) and (Q) (13) - (17) in order to obtain $H^{n-1} \rightarrow H^{n-2} \rightarrow \cdots \rightarrow H^0$. The elements of H^n are computed using only the finest scaling functions φ^n.

$O(N^2)$ is the theoretical worst case cost for this (pre-)computation. In practice it turns out that the matrices H^L are very sparse since the basis functions have a very small support. All computations (e.g. evaluations, matrix-vector multiplications) involving H^L are therefore of $O(N)$ cost.

4.3   Area preserving deformation

As mentioned in Section 4.1 the deformation of the curve is defined by a deformation of its control polygon at some level of resolution. The new control points define a new set of scaling coefficients. The details have not been modified. The vectors of the x and y coordinates of these new coefficients together with all details are denoted by X_0 and Y_0 (without index "L"). The problem of area preserving deformation we aim to solve is now the following optimization problem:

$$\min_{X,Y} \quad (1-\alpha)\mathcal{E}(X,Y) + \alpha\mathcal{D}(X,Y) \qquad \text{subject to} \qquad \mathcal{A} - \mathcal{A}_{ref} = 0. \qquad (18)$$

where $\mathcal{D}(X,Y) = (\|X - X_0\|^2 + \|Y - Y_0\|^2)$ mesures the distance between the original and the area prserving deformed curve. The scalar value α, $0 \leq \alpha \leq 1$, therefore balances between a smoother curve (small value of α) and a curve which better respects the defined deformation (big value of α). The area constraint is a quadratic expression. Remember that one of the main objectives of this method is to provide a very fast algorithm for interactive deformation. Thus, instead of minimizing a quadratic cost function subject to a quadratic constraint, we linearize the constraint for speed.

4.4   Linearizing the area constraint

The constrained minimzation problem (18) can be converted into an unconstrained problem using the technique of Lagrange multiplyers [5] defined by

$$\max_{\lambda} \min_{X,Y} \quad (1-\alpha)\mathcal{E}(X,Y) + \alpha\mathcal{D}(X,Y) + \lambda(XMY^T - 2\mathcal{A}_{ref}) \qquad (19)$$

where λ is the Langrangien multiplyer for the constraint $XMY^T - 2\mathcal{A}_{ref} = 0$.

However, the constraint is quadratic and not linear. A solution of (19) can be found by linearizing the area constraint as follows: The quadratic area constraint can be approx-

imated by the following symmetric linear constraint:

$$X_0 MY^T + XMY_0^T = 2(\mathcal{A}_0 + \mathcal{A}_{ref}). \tag{20}$$

For $X = X_0 + dX$ and $Y = Y_0 + dY$ satisfying (20) one gets $2\mathcal{A} = 2\mathcal{A}_{ref} + dXMdY^T$. Thus, if $dXMdY^T$ is close to 0 the area will be well approximated: $\mathcal{A} \approx \mathcal{A}_{ref}$.

The linearized area constraint replaces now the quadratic one in (19) and leads to the following Lagrangien function $L(X, Y, \lambda)$:

$$(1 - \alpha)\mathcal{E}(X, Y) + \alpha\mathcal{D}(X, Y) + \lambda(X_0 MY^T + XMY_0^T - 2(\mathcal{A}_0 + \mathcal{A}_{ref})). \tag{21}$$

Instead of solving the constrained minimization problem (18) we now solve the unconstrained min-max problem

$$\max_{\lambda} \min_{X,Y} \quad L(X, Y, \lambda). \tag{22}$$

A necessary condition for the triplet (X, Y, λ) to be a solution of (22) is the vanishing of all partial derivatives of L with respect to the unknowns x_i, y_i and λ. This is equivalent to solve the following system of linear equations:

$$\begin{bmatrix} (1-\alpha)H + 2\alpha\mathrm{Id} & 0 & MY_0^T \\ 0 & (1-\alpha)H + 2\alpha\mathrm{Id} & M^T X_0^T \\ Y_0 M^T & X_0 M & 0 \end{bmatrix} \begin{bmatrix} X^T \\ Y^T \\ \lambda \end{bmatrix} = \begin{bmatrix} 2\alpha X_0^T \\ 2\alpha Y_0^T \\ 2(\mathcal{A}_{ref} + \mathcal{A}_0) \end{bmatrix} \tag{23}$$

Note that the solution X, Y doesn't preserve exactly the area due to the linearization (20). In practice, some further iterations like

> **while** $|\mathcal{A}_0 - \mathcal{A}_{ref}| > threshold * |\mathcal{A}_{ref}|$ **loop**
>
> compute X and Y with approximate method
> $(X_0, Y_0) \leftarrow (X, Y)$
>
> **end loop**,

can be used in order to increase the precision up to an arbitrary small user pre-defined value. For example, a threshold of 10^{-4} can generally be reached in no more than 1 or 2 iterations. This method has the advantage to treat X and Y symmetrically, and it can be shown that it is invariant under isometries: if X_0 and Y_0 are modified by an isometry, the solution X, Y follows the same isometry.

4.5 Influence of the weigth α

In all following illustrations we will use color in order to identify the original curve (blue), the deformed curve (green), and the final area preserving curve (red). The control polygons that are shown in most of the figures correspond to the coarse polygon. They are formed by the scaling coefficients at the resolution level L, at which the deformation is defined.

Figure 6 illustrates the three steps of the algorithm and shows the influence of the weight α in formula (21). The original curve (blue) is a Chaikin subdivision curve with 32 control points ($n = 5$). After 2 analysis steps one gets 8 scaling coefficients and 24 detail coefficients representing the curve in the basis of $V^5 = V^3 \oplus W^3 \oplus W^4$. The scaling coefficients form the coarse control polygon at resolution level $L = 3$.

In the example of Figure 6 a deformation is defined by displacing one of the coarse control points (green control polygon). Thanks to the multiresolution representation it is possible to deform the curve at a large scale by displacing only a few number of coarse coefficients. Reconstruction of the deformed control polygon would result in a deformed curve (green) that is not area preserving. An area preserving curve that respects the defined deformation is now computed by solving (22). Note that all computations are done at resolution level $L = 3$, i.e. the vectors X_0 and Y_0 contain the coefficients (scaling and detail) of the deformed curve (green), and X and Y contain the coefficients of the area preserving deformed curve at level $L = 3$ (red polygon). The last step consists of reconstructing the curve from these coefficients (red).

The choice of the weight α in (21) and (23) is left to the user. A small value gives preference to a smooth curve in the sense of minimal bending energy, see Figure 6(c) with $\alpha = 0.5$. A higher value of α keeps the distance as small as possible, see Figure 6(d), with $\alpha = 0.99$.
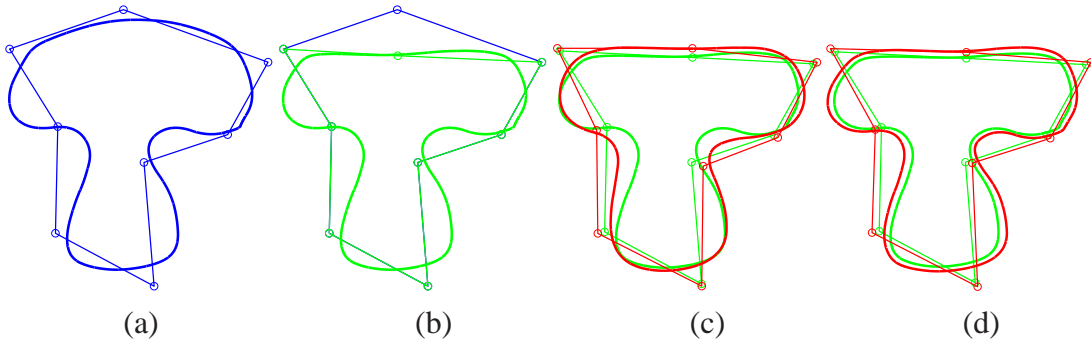

(a)   (b)   (c)   (d)

Figure 6: Influence of the coefficient α. (a) original curve and its coarse control polygon (blue), n=5, L=3. (b) deformed coarse polygon and reconstructed curve without area correction (green), L=3. (c) corrected coarse polygon and area preserving curve (red), $\alpha = 0.5$. (d) $\alpha = 0.99$.

Remarks:

1. **Global method**. The area preserving deformation method presented in this section

is a global method. This means that all coefficients of the curve are involved in the computation of the area preserving curve, see (21). This can be of advantage as shown in Figure 6(d) where the area correction (red) is a kind of symmetric growing of the deformed curve (green) until the prescribed area is reached. However, a better control is generally obtained using local optimization, as explained in the next section.

2. **Unwanted shift**. It can happen that the area preserving curve shifts. This is a quite natural side-effect of the global method in the case of Figure 7. The original curve (blue) with $n = 7$ is decomposed until level $L = 2$. One of the four coarse control points is displaced resulting in the green curve after reconstruction. The global area preserving method in this case leads now to a symmetrically shrinked version of the deformed (green) curve. At the same time the resulting curve shifts away from the original and the deformed curve. Section 5 will treat this feature.

3. **Loss of details**. When reconstructing a curve, more and more details are added. These details can get lost if the detail coefficients, contained in X and Y, are modified during the area preserving step (23), in particular if their absolute value decreases, see Figure 7(d). The next section will treat this problem.
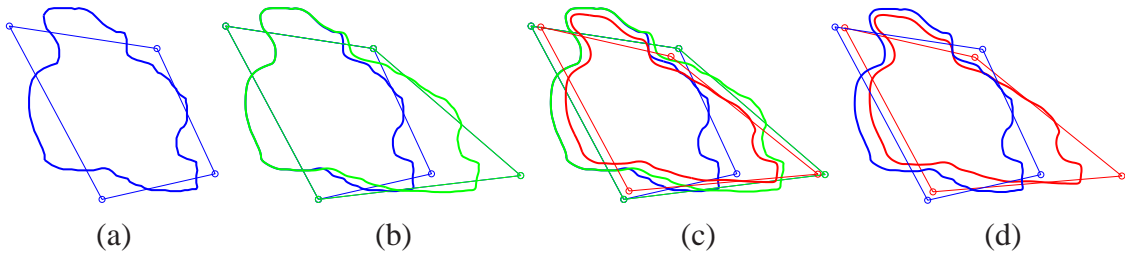

Figure 7: Shifting and loss of details. (a) original curve and coarse control polygon, $n = 7, L = 2$. (b) deformed curve without area constraint (green). (c) deformed curve with area constraint (red). (d) same as (c), without the green curve.

5   Localized MR-deformation under constraints

The basic idea of multiresolution deformation of curves under constraints has been introduced in the previous sections. It is possible to perform an area preserving deformation of a curve at different scales by displacing only a few coarse coefficients (control points) at a low resolution level. Nevertheless the whole power behind the multiresolution formulation of this deformation method has not entirely been exploited until now.

In the present section we will focus on two further features of the present method, namely its ability to

- localize the deformation scheme, and to

- make use / or not of the detail coefficients.

These new features will result from the selection of some subset of the unknowns, that will be fixed during the optimization. Before explaining in Sections 5.1, 5.2, and 5.3 what subsets can be fixed - and what are the corresponding effects - we begin by describing how any given subset of the unknowns can be fixed during the optimization.

Selection of index subset At any level of resolution the curve is always defined by a constant number of $D2^n$ coefficients. Let us split the index set $\{1, 2, \ldots, D2^n\} = I \cup J$ where

- I is the index set of coefficients that can be modified

- J is the index set of coefficients that stay unchanged.

The solution of (22) with respect to $X(I)$, $Y(I)$ and λ results in the following linear system of equations:

$$
\begin{bmatrix}
(1-\alpha)H(I,I) & 0 & M(I,.)Y_0^T \\
+2\alpha\,\mathrm{Id} & & \\
& & \\
0 & (1-\alpha)H(I,I) & M(.,I)^T X_0^T \\
& +2\alpha\,\mathrm{Id} & \\
& & \\
Y_0 M(I,.)^T & X_0 M(.,I) & 0
\end{bmatrix}
\begin{bmatrix}
X(I)^T \\
\\
Y(I)^T \\
\\
\lambda
\end{bmatrix}
=
\begin{bmatrix}
2\alpha X_0^T - \\
(1-\alpha)H(I,J)X_0(J)^T \\
\\
2\alpha Y_0^T - \\
(1-\alpha)H(I,J)Y_0(J)^T \\
\\
2(\mathcal{A}_{ref} + \mathcal{A}_0) \\
-X_0(J)H(J,.)Y_0^T \\
-X_0 H(.,J)Y_0(J)^T
\end{bmatrix}
\tag{24}
$$

where $X(I)$ is a vector of size $|I|$ containing the elements x_i of X with $i \in I$. Analogous are the matrix notations: $H(I, J)$ for example is a $(|I| \times |J|)$-submatrix of H containing only the lines of index I and the columns of index J.

5.1   Local deformation

Suppose the curve has been deformed at level L by displacing one of its coarse control points. By selecting k neighbour points on both sides of this point which are free to move during the area correction step, the deformation can be controlled locally.

Let us illustrate this principle with the 'potato' example of Figure 8. The curve has 128 ($n = 7$) initial control points, and the deformation resolution level is $L = 4$. One of

the 16 coarse control points has been moved. Figure 8 from left to right shows the area preserving curves obtained by leaving $k = 1, 2$ and 5 neighbouring control points on each side free for the area constraint deformation.
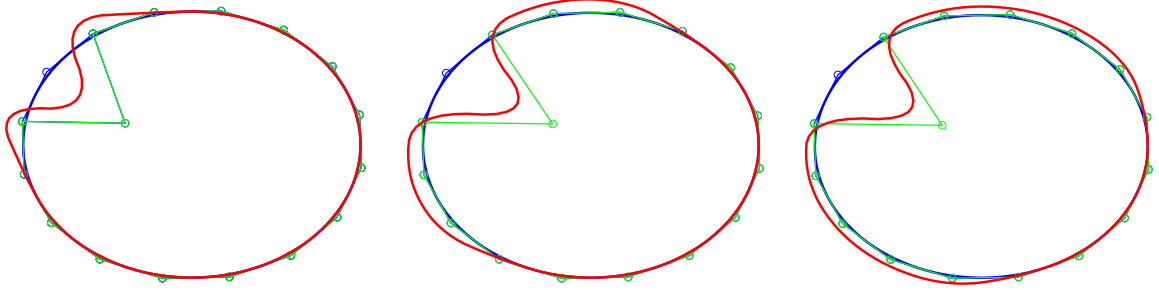

Figure 8: Local control. $n = 7$ and $L = 4$. From left to right: $k = 1$, $k = 3$, and $k = 5$.

5.2 Upholding the moved point

For a better/closer control of the deformation it may be useful to keep the moved point in position. Lets take again a potato-example ($n = 7$, $L = 3$). Its multiresolution decomposition consists of 8 scaling (coarse) coefficients $x^3_{k(1 \leq k \leq 8)}$. Now suppose we move the point x^3_5 (see Figure 9) and we want to correct the curve in the extent of 2 control points in order to keep area. We can make two different choices:

- $I = \{3, 4, 5, 6, 7\}$: x^3_5 moves during the correction (Figure 9, left).

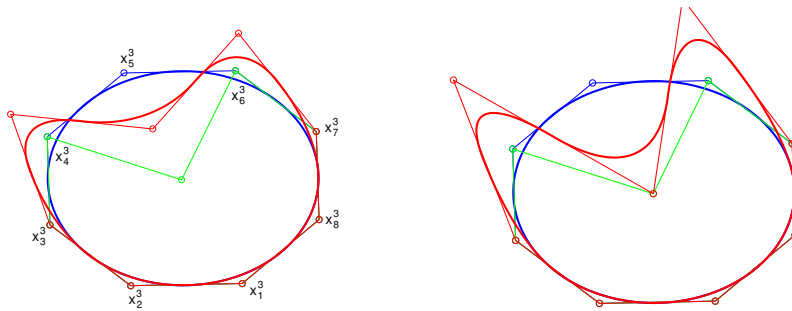- $I = \{3, 4, 6, 7\}$: x^3_5 stays in position during the correction (Figure 9, right).

Figure 9: Left: the moved point is not kept in position; Right: it is.

5.3 Modification of detail coefficients

The technique of index subset selection gives the possibility to preserve the detail coefficients $(d^k_j)_{j,k}$ or to modify them by choosing the set I. Different shape effects result:

- $I \cap [D2^L; D2^n - 1] \neq \emptyset$: some detail coefficient will be modified in order to satisfy the area constraint. The left part in Figure 10 shows that the final curve lost most of the fine details. It is smoother.

- $I \cap [D2^L; D2^n - 1] = \emptyset \iff I \subseteq [0; D2^L - 1]$: only coarse coefficients will be modified. The right part in Figure 10 shows that the global shape is modified but all the fine details are kept.
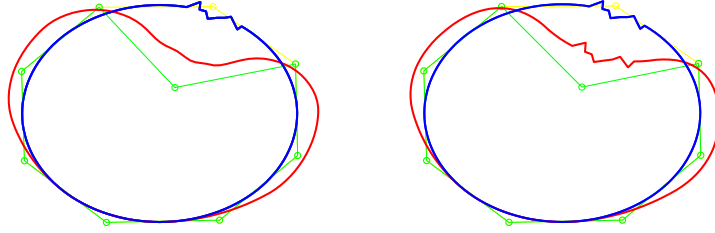

Figure 10: Two curves with similar coarse polygons. left: area preservation with details changed; right: area preservation with details preserved.

Note that the extensions presented in the last three subsections can be formulated equivalently by adding linear interpolation constraints into the minimization (18). But the resulting system would be bigger instead of being reduced as in (24).

5.4   Results

We implemented this method in an interactive 2D graphics modeling system. The curve can be decomposed and edited at any level of resolution. The user selects and drags a control point of the coarse approximation of the curve with the mouse. The corresponding area preserving deformation of the initial curve follows immediately. All computations are done in real-time, i.e. any continuous displacement of a coarse control point by the user leads to an immediate continuous deformation of the curve. Notice that for each small displacement of a control point the system (24) has to be solved. The matrices are coded as sparse matrices and solved iteratively with the bi-conjugate gradient method using a library for sparce matrix computations. In worst case, when using global optimization (the matrix (24) has maximal size), we reach a frame rate of more than 150 Hz with 500 control points, between 20 and 70 Hz with 2000 points and between 10 and 30 Hz with 4000 points. The frame rate mostly depends on the editing level L (decomposition fills the matrices) and on the the displacement speed (the faster you move the more the area needs to be corrected).

Some complex real-world example presenting a lot of details in its shape is shown in Figure 11. The seahorse has 512 control points. The two series of deformations are obtained by moving one coarse control point of the abdomen to the left and to the right. The upper row curves keep the enclosed area constant producing more realistic looking deformations. The lower row curves don't. Lasseter's principle of animations [15] can certainly be confirmed.
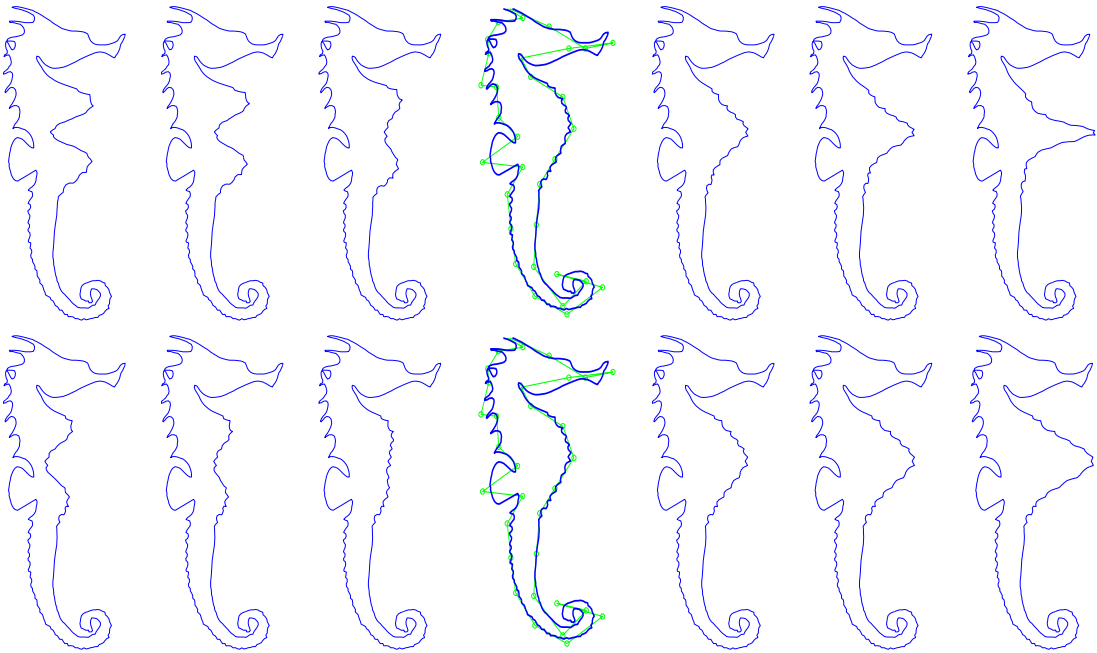
Figure 11: Multiresolution deformation of a seahorse with 512 control points. The original curve (in the middle) has been deformed by moving only one coarse control points inside and outside the abdomen. Upper row: with area preservation. Lower row: without area preservation.

6   Conclusion and future work

A method for area preserving multiresolution deformation of curves at any level of detail has been presented. The curves under consideration are planar and periodic. The area constraint, which is a non-linear constraint, is maintained through an optimization process. The constraint is linearized for speed, but it can be approached as exactly as needed. For wavelet based multiresolution curves, we have developed a multiresolution expression of the area constraint, which results in the computation of a sequence of area matrices. We developed a recurrence relation for these matrices, which allows for an efficient linear time (pre-) computation of these matrices. Real-time deformation is possible. All these developments apply to any kind of multiresolution curves defined as in Section 2.

This paper presented a general method of multiresolution area preserving deformation. It has not been our intention to work out particular deformation schemes. Since the coarse control polygons and detail coefficients of the curve at any scale are directly accessible and since they are free parameters, the method offers a maximum number of degrees of freedom for any particular deformation simulating any particular behaviour.

Currrent and future research concerns the integration of other non-linear constraints into a multiresolution curve editing environment. Length preserving deformation for example would allow to work with open and with 3D curves. The generalization to surfaces is also under investigation.

References

[1] D. Baraff and A. Witkin. Large steps in cloth simulation. *Computer Graphics Proceedings (SIGGRAPH 98)*, pages 43–54, 1998.

[2] G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991.

[3] G.-P. Bonneau, S. Hahmann, and G.M. Nielson. Blac-wavelets: a multiresolution analysis with non-nested spaces. In *VIS '96 Proceedings*, pages 43–48. IEEE Computer Society Press, 1996.

[4] G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.

[5] P.G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimization*. Cambridge University Press, 1988.

[6] A. Cohen and B. Matei. Compact representations of images by edge adapted multiscale transforms. In *IEEConference ICIP conference, Tessaloniki*, 2001.

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

[9] D. Eberly and J. Lancaster. On gray scale image measurements i. arc length and area. *GMIP*, 53(6):538–549, 1991.

[10] G. Elber. Linearizing the area and volume constraints. *Technical Report CIS-2000-04*, 2000.

[11] 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.

[12] A. Finkelstein and D.H. Salesin. Multiresolution curves. *Computer Graphics Proceedings (SIGGRAPH 94)*, pages 261–268, 1994.

[13] S.J. Gortler, P. Schröder, M.F. Cohen, and Pat Hanrahan. Wavelet radiosity. *Computer Graphics Proceedings (SIGGRAPH 93)*, pages 221–230, 1993.

[14] M.H. Gross, L. Lippert, A. Dittrich, and S. Häring. Two methods for wavelet-based volume rendering. *Computers & Graphics*, 21(2):237–252, 1997.

[15] J. Lasseter. Principles of traditional animation applied to 3d computer animation. *Computer Graphics Proceedings (SIGGRAPH 87)*, pages 35–44, 1987.

[16] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Transaction on Image Processing*, 2004. to appear.

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

[18] A. Mertins. *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*. John Wiley & Sons, Chichester, 1999.

[19] 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.

[20] G. Sapiro and A. Tannenbaum. Area and length preserving geometric invariant scale-spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):67–72, 1995.

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

[22] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics Proceedings (SIGGRAPH 87)*, pages 205–214, 1987.