

Single-knot Wavelets for Non-uniform B-splines

Martin Bertram

TU Kaiserslautern, Germany

Abstract

We propose a flexible and efficient wavelet construction for non-uniform B-spline curves and surfaces. The method allows to remove knots in arbitrary order minimizing the displacement of control points when a knot is re-inserted. Geometric detail subtracted from a shape by knot removal is represented by an associated wavelet coefficient replacing one of the control points at a coarser level of detail. From the hierarchy of wavelet coefficients, perfect reconstruction of the original shape is obtained. Both knot removal and insertion have local impact. Wavelet synthesis and analysis are both computed in linear time, based on the lifting scheme for biorthogonal wavelets. The method is perfectly suited for multiresolution surface editing, progressive transmission, and compression of spline curves and surfaces.

Keywords: Multiresolution Modeling, Hierarchical Splines, Wavelet Lifting.

1 Introduction

Discrete B-splines, introduced by Schumaker [21], are used in the *Oslo Algorithm* [7] defining the relation of B-splines on two different scales. The latter are associated with two nested knot sequences $\tau \subset \tau'$, defining k th-order B-Splines $N_{i,k}$ and $N'_{j,k}$, respectively. A coarse-scale

B-spline curve given by de Boor points \mathbf{d}_i can be represented on the finer scale as

$$\sum_j \mathbf{d}'_j N'_{j,k}(t) = \sum_i \mathbf{d}_i N_{i,k}(t), \quad (1)$$

where the control points \mathbf{d}'_j are computed with the aid of discrete B-splines $n_{i,k}(j)$, $j \in \mathbb{Z}$, as

$$\mathbf{d}'_j = \sum_i n_{i,k}(j) \mathbf{d}_i. \quad (2)$$

Hence, the discrete B-Splines correspond to the rows of a subdivision matrix satisfying

$$N_{i,k}(t) = \sum_j n_{i,k}(j) N'_{j,k}(t). \quad (3)$$

The Oslo Algorithm provides this subdivision matrix $s_{ji} := n_{i,k}(j)$, based on the knot sequences τ and τ' , and the B-spline order k .

In the scope of wavelets, the inverse transform (fine to coarse) is of interest, too. Here, a complementary space W of $V := \text{span}\{N_{i,k}\}$ with respect to $V' := \text{span}\{N'_{j,k}\}$ is constructed. A projection of V' onto $V \oplus W$ corresponds to a fitting operation, where the details removed from a B-spline curve are represented in the space W . Successive steps of fitting are denoted as *wavelet analysis*, whereas subdivision corresponds to *wavelet synthesis*.

Semi-orthogonal wavelet constructions require mutual orthogonality of the spaces V and W , implying that the fitting operation satisfies least-squares minimization. If multiple knots are removed in one pass, semi-orthogonal wavelets with finite support can be constructed [6, 15]. Similar constructions were used for hierarchical editing, progressive transmission, and signal processing of curves and surfaces [12, 16, 9]. A nice overview of non-uniform methods is given by Lyche et al. [17] and by Quak [18].

Bi-orthogonal wavelets do not require orthogonal spaces. In contrast to semi-orthogonal approaches where wavelet analysis is a global operation based on a banded system of equations, in the biorthogonal case all operations are local, increasing efficiency and flexibility. Biorthogonal approaches exist for uniform knots with dyadic refinement [23, 10] and for subdivision surfaces [14, 20, 19, 3]. To our knowledge, no such constructions are available for non-uniform B-Splines. More information on classical wavelet constructions can be found in the literature [8, 5, 22].

Adaptive representations other than wavelets include hierarchical B-splines [13], offering top-down refinement for the design of complex shapes. Due to the lacking construction of a complementary space W , however, geometric detail would be lost when coarsening the representation. When removing a knot from a B-spline curve, the fitted polygon can be obtained, for example, by averaging the C^{k-2} -continuous constructions from both sides [11].

An alternative to the orthogonalization of continuous functions in the L^2 -norm is the use of a discrete norm [10, 1]. This way, the fitting operation minimizes the displacement of control points rather than the error of the continuous shape. We note that in the scope of B-splines, both residuals are tied together, since B-splines define a Riesz basis [17, 18], i.e. there exists a constant D_k , such that for arbitrary knots τ it holds

$$D_k^{-1} \|\mathbf{d}\|_{l^2, \tau} \leq \left\| \sum_i \mathbf{d}_i N_{i,k}(t) \right\|_{L^2} \leq \|\mathbf{d}\|_{l^2, \tau}, \quad (4)$$

$$\|\mathbf{d}\|_{l^2, \tau} = \left(\frac{1}{k} \sum_i (t_{i+k} - t_i) \mathbf{d}_i^2 \right)^{\frac{1}{2}}.$$

Discrete orthogonalization has already been used for constructing subdivision-surface wavelets [19, 2]. Here, only few scaling functions were orthogonalized with a wavelet, to obtain local computation.

Single-knot wavelets removing only one knot on each scale were initially described for linear B-splines by Warren [22], p.102. This construction is based on local L^2 -orthogonalization such that a least-squares fit is approximated. In the present work, we found that discrete least-squares fitting is a local operation when combined with removing a single knot (or a local set of knots). We show how to obtain this optimization property and we factorize the corresponding wavelet construction into local *lifting operations* [23]. Our method is highly efficient and, in contrast to many other constructions, simple to implement.

These are the contents of our work: In section 2, we summarize the necessary steps to obtain our algorithm. We start with a summary Boehm's knot-insertion method [4] and provide an example for our wavelet construction for the cubic case. Then, we present our general algorithm

with possible extensions. In section 3, we provide examples for the wavelet shapes and show how the method is used in applications like progressive transmission and multiresolution editing.

2 Wavelet Construction

2.1 Knot Insertion

A B-spline curve of order k is represented by $m + 1$ de Boor points \mathbf{d}_i ($i = 0, \dots, m$) and knots t_i ($i = 0, \dots, m + k$). The curve is defined as

$$\mathbf{f}(t) = \sum_{i=0}^m \mathbf{d}_i N_i^k(t), \quad t \in [t_{k-1}, t_{m+1}], \quad (5)$$

where $N_i^k(t)$ are the B-splines of order k .

To increment the number of control points, a new inner knot $u \in [t_{k-1}, t_{m+1}]$ can be inserted using Böhm's algorithm [4], as follows. Let r denote the index of the knot left of u , i.e. $u \in [t_r, t_{r+1})$. We compute the subdivision ratios α_i^1 according to de Boor's algorithm:

$$\alpha_i^1 = \frac{u - t_i}{t_{i+k-1} - t_i} \quad (i = r - k + 2, \dots, r) \quad (6)$$

The subsequence $\mathbf{d}_{r-k+2}, \dots, \mathbf{d}_{r-1}$ of de Boor points is now replaced by

$$\mathbf{d}'_i := (1 - \alpha_i^1)\mathbf{d}_{i-1} + \alpha_i^1\mathbf{d}_i \quad (i = r - k + 2, \dots, r). \quad (7)$$

All other control points remain unchanged:

$$\begin{aligned} \mathbf{d}'_i &:= \mathbf{d}_i & (i = 0, \dots, r - k + 1); \\ \mathbf{d}'_{i+1} &:= \mathbf{d}_i & (i = r, \dots, m). \end{aligned} \quad (8)$$

The refined curve is composed of $m + 2$ control points \mathbf{d}'_i ($i = 0, \dots, m + 1$) and the knot vector $\tau' = \{t_0, \dots, t_r, u, t_{r+1}, \dots, t_{m+k}\}$. If the knot u already exists in the coarser knot vector $\tau = \{t_0, \dots, t_{m+k}\}$, then its multiplicity is incremented by the insertion.

Knot insertion does not change the geometry of a curve. Only the control polygon and the knot vector are locally changed. Repeated bisection of all inner knot intervals corresponds to a

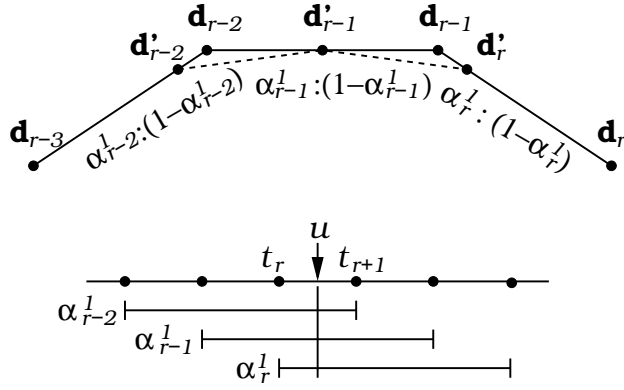


Figure 1: Knot insertion for $k = 4$.

subdivision scheme for multiresolution curves [12]. If multiple knots are inserted simultaneously, like in the dyadic case where every knot interval is subdivided, the Oslo Algorithm [7] may be used. A single knot insertion for the cubic case is depicted in figure 1.

2.2 Lazy Cubic Wavelet Transform

Before we introduce a general scheme, we consider the cubic case $k = 4$ as an example. To add further detail to a geometric shape when inserting a new knot, a *wavelet coefficient* \mathbf{w}' is introduced on the coarse level. In the case of the *lazy wavelet transform*, this coefficient simply defines a displacement of \mathbf{d}'_{r-1} after knot insertion. Hence, the lazy wavelet synthesis is defined

as

$$\begin{pmatrix} \mathbf{d}'_{r-3} \\ \mathbf{d}'_{r-2} \\ \mathbf{d}'_{r-1} \\ \mathbf{d}'_r \\ \mathbf{d}'_{r+1} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ (1-a_1) & a_1 & & & \\ & (1-a_2) & 1 & a_2 & \\ & & & (1-a_3) & a_3 \\ & & & & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{r-3} \\ \mathbf{d}_{r-2} \\ \mathbf{w}' \\ \mathbf{d}_{r-1} \\ \mathbf{d}_r \end{pmatrix}, \quad (9)$$

where $a_i = \alpha_{r-3+i}^1$. We denote this this by $\mathbf{d}' = \mathbf{S} \mathbf{d}$, where \mathbf{S} is our subdivision matrix.

To construct the inverse of equation (9), called *lazy wavelet analysis*, we factorize the sub-

division matrix into three *lifting operations* [23], as follows:

$$\mathbf{S} = \begin{pmatrix} 1 & & & & & \\ (1-a_1) & a_1 & & & & \\ & & 1 & & & \\ & & & (1-a_3) & a_3 & \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & (1-a_2) & 1 & a_2 & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}. \quad (10)$$

The lifting operations manipulate the coefficients \mathbf{w}' (right matrix), as well as \mathbf{d}_{r-2} and \mathbf{d}_{r-1} .

The latter two operations are independent and can be computed simultaneously.

Under the assumption that all inner knots have a multiplicity of no more than three, we have $a_1 \neq 0$ and $a_3 \neq 1$ and the inverse of \mathbf{S} exists:

$$\mathbf{S}^{-1} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ (a_2-1) & 1 & -a_2 & & & \\ & & & 1 & & \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & & \\ \frac{a_1-1}{a_1} & \frac{1}{a_1} & & & & \\ & & 1 & & & \\ & & & \frac{1}{1-a_3} & \frac{a_3}{a_3-1} & \\ & & & & & 1 \end{pmatrix}. \quad (11)$$

In the algorithm, the coefficients are first manipulated by the individual lifting operations and then re-named. The lazy cubic synthesis can be implemented as follows:

$$\begin{aligned} \mathbf{w}' &\longleftarrow \mathbf{w}' + (1-a_2)\mathbf{d}_{r-2} + a_2\mathbf{d}_{r-1}; \\ \mathbf{d}_{r-2} &\longleftarrow a_1\mathbf{d}_{r-2} + (1-a_1)\mathbf{d}_{r-3}; \\ \mathbf{d}_{r-1} &\longleftarrow (1-a_3)\mathbf{d}_{r-1} + a_3\mathbf{d}_r. \end{aligned} \quad (12)$$

The coefficients \mathbf{d}_{r-2} , \mathbf{w}' , and \mathbf{d}_{r-1} are then renamed to \mathbf{d}'_{r-2} , \mathbf{d}'_{r-1} , and \mathbf{d}'_r , respectively, sharing the same memory in the implementation of the algorithm.

The inverse operation, removing a knot, is denoted as *lazy wavelet analysis*. Therefore, the

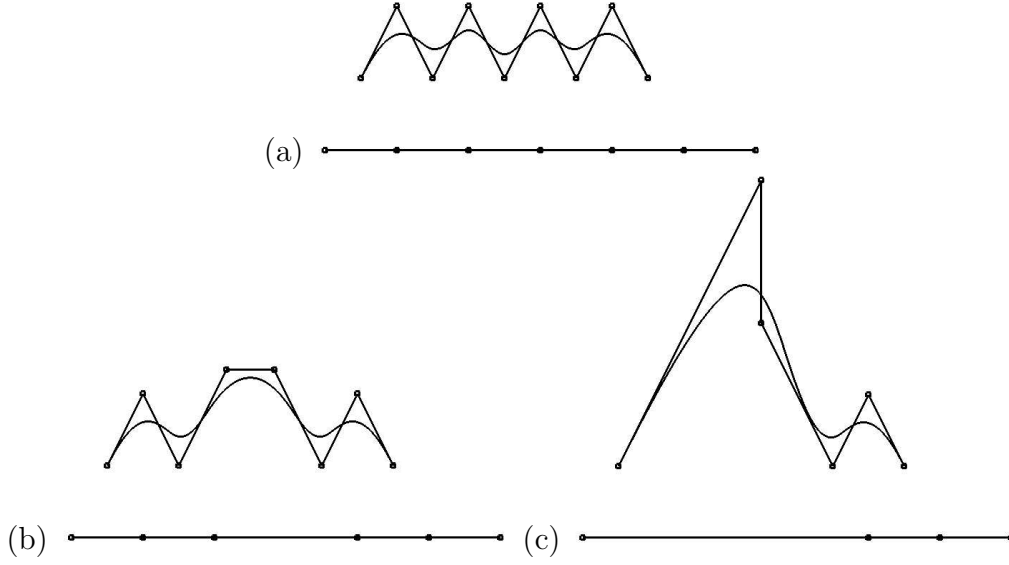


Figure 2: Repeated knot removal based on the lazy wavelet analysis leads to undesirable shapes.

(a) Original B-spline curve; (b) one knot removed; (c) three knots removed.

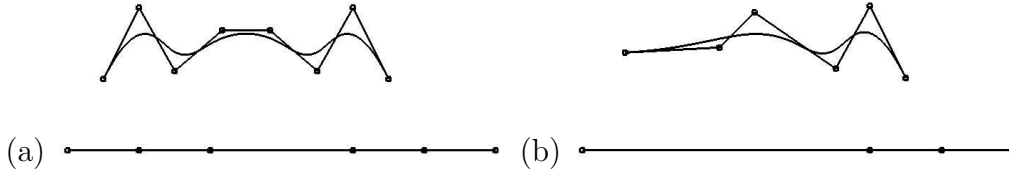


Figure 3: Knot removal minimizing displacement of control points for the curve in figure 2a.

(a) One knot removed; (b) three knots removed.

inverse lifting operations are computed in reverse order:

$$\begin{aligned}
 \mathbf{d}_{r-1} &\leftarrow \frac{1}{1-a_3} (\mathbf{d}_{r-1} - a_3 \mathbf{d}_r); \\
 \mathbf{d}_{r-2} &\leftarrow \frac{1}{a_1} (\mathbf{d}_{r-2} - (1-a_1) \mathbf{d}_{r-3}); \\
 \mathbf{w}' &\leftarrow \mathbf{w}' - (1-a_2) \mathbf{d}_{r-2} - a_2 \mathbf{d}_{r-1}.
 \end{aligned} \tag{13}$$

2.3 Optimized Fitting for Cubic Wavelet

When using the lazy wavelet analysis for repeated knot removal, the effects on the corresponding curve are unpredictable, see figure 2. The problem is that this transform provides unfavorable fitting, indicating that the basis defined by the underlying wavelets may be unstable. To

improve the fitting, we combine the inverse subdivision with an additional lifting operation, such that the wavelet analysis becomes

$$\mathbf{d} = \mathbf{F} \mathbf{S}^{-1} \mathbf{d}', \quad (14)$$

where $\mathbf{d} = (\mathbf{d}_{r-3}, \mathbf{d}_{r-2}, \mathbf{w}, \mathbf{d}_{r-1}, \mathbf{d}_r)^T$ and $\mathbf{d}' = (\mathbf{d}'_{r-3}, \dots, \mathbf{d}'_{r+1})^T$.

A classical approach for the construction of \mathbf{F} is the orthogonalization of the lazy wavelet with respect to the coarse-level B-splines (scaling functions), resulting in a semi-orthogonal wavelet construction [12, 22]. Due to orthogonality of both subspaces, the coarse-level curve is a least-squares fit with respect to the original shape. When using the L^2 -norm, however, this fit is a global operation, which is not desirable when removing single knots.

In contrast to L^2 -orthogonalization, we minimize the displacement of control points obtained from knot removal followed by re-insertion with a zero wavelet coefficient. A similar approach was used for the construction of dyadic splines [10]. For an arbitrary control polygon \mathbf{d}' , our optimization will satisfy

$$\|\mathbf{d}' - \mathbf{S}\mathbf{F}^{-1}\mathbf{X}\mathbf{F}\mathbf{S}^{-1}\mathbf{d}'\|_{l^2, \tau'} \longrightarrow \min, \quad (15)$$

where the matrix \mathbf{X} simply cancels out the wavelet coefficient \mathbf{w} . This optimization has local impact and it provides a well-shaped fit, since the displacement of the curve is bounded by the displacement of its control points, according to equation (4). An example for this optimization is depicted in figure 3.

In analogy to the semi-orthogonal approach, we obtain the minimum of equation (15) by orthogonalizing the lazy wavelet with the coarse scaling functions. We use a discrete norm induced by the following scalar product

$$\langle \mathbf{p}, \mathbf{q} \rangle_{l^2, \tau} = \sum_i c_i \mathbf{p}_i \cdot \mathbf{q}_i, \quad c_i = (t_{i+k} - t_i)/k, \quad (16)$$

where \mathbf{p} and \mathbf{q} are control polygons based on the same knot vector τ . We use this norm on the finer level corresponding to $\tau' = \{t_0, \dots, t_r, u, t_{r+1}, \dots, t_{m+k}\}$, from which the weighting c_i for the

control points is determined¹.

We associate with each control point \mathbf{d}_i a *discrete scaling function* ϕ'_i . In the semi-orthogonal approach, these scaling functions are simply the fine-level B-splines. For our construction minimizing displacement of control points, however, the ϕ'_i must form an orthogonal basis with proper weighting, i.e.,

$$\langle \phi'_i, \phi'_j \rangle_{l^2, \tau'} = c_i \delta_{ij}. \quad (17)$$

This way, we obtain superposition of fine-level control points,

$$\left\| \sum_i \mathbf{d}_i \phi'_i \right\|_{l^2, \tau'} = \left(\sum_i c_i \mathbf{d}_i^2 \right)^{\frac{1}{2}}. \quad (18)$$

We do not need to know anything about the shape of these scaling functions, except for their mutual inner products. The ϕ'_i are introduced to show the analogy to semi-orthogonal constructions.

The coarse-level basis functions can now be defined as linear combinations of the ϕ'_i . Therefore, we consider the lazy wavelet analysis transforming a control polygon from $span\{\phi'_i\}_{i=0}^{m+1}$ into the coarse-level basis,

$$\mathbf{w}' \psi' + \sum_{i=0}^m \mathbf{d}_i \phi_i = \sum_{i=0}^{m+1} \mathbf{d}'_i \phi'_i, \quad (19)$$

where ψ' is the discrete lazy wavelet and ϕ_i are the coarse discrete scaling functions. These are obtained from

$$(\phi_{r-3} \quad \phi_{r-2} \quad \psi' \quad \phi_{r-1} \quad \phi_r) = (\phi'_{r-3} \quad \dots \quad \phi'_{r+1}) \mathbf{S}. \quad (20)$$

The fitting matrix \mathbf{F} is now constructed to orthogonalize ψ' with respect to ϕ_i ($i = 0, \dots, m$) based on the discrete inner product. The optimized wavelet ψ is then obtained by

$$\psi = \psi' + \sum_{i=0}^3 b_i \phi_{r-3+i}, \quad (21)$$

with constants b_i satisfying

$$\langle \psi, \phi_i \rangle_{l^2, \tau'} = 0 \quad (i = 0, \dots, m). \quad (22)$$

¹In a preliminary implementation, uniform weighting $c_i = 1$ was used. It was suggested by the referees that stability can be improved by a non-uniform weighting, according to equation (4).

Inserting equation (21) into these orthogonality constraints provides the system for the b_i :

$$\sum_{i=0}^3 b_i \langle \phi_{r-3+i}, \phi_{r-3+j} \rangle_{l^2, \tau'} = - \langle \psi', \phi_{r-3+j} \rangle_{l^2, \tau'} \quad (j = 0, \dots, 3). \quad (23)$$

In matrix notation, this system is

$$\begin{pmatrix} c_0 + c_1(1 - a_1)^2 & c_1(1 - a_1)a_1 & & & \\ c_1(1 - a_1)a_1 & c_1a_1^2 + c_2(1 - a_2)^2 & c_2(1 - a_2)a_2 & & \\ & c_2(1 - a_2)a_2 & c_2a_2^2 + c_3(1 - a_3)^2 & c_3(1 - a_3)a_3 & \\ & & c_3(1 - a_3)a_3 & c_3a_3^2 + c_4 & \\ & & & & \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0 \\ c_2(a_2 - 1) \\ -c_2a_2 \\ 0 \end{pmatrix}, \quad (24)$$

where

$$\begin{aligned} c_0 &= (u - t_{r-3})/4; \\ c_i &= (t_{r+i} - t_{r-3+i})/4, \quad (i = 1, \dots, 3); \\ c_4 &= (t_{r+4} - u)/4. \end{aligned}$$

The matrix is tridiagonal and non-singular, since the discrete scaling functions are linearly independent (provided that multiplicity of knots does not exceed $k = 4$). The discrete scaling functions not involved in the subdivision remain mutually orthogonal and do not require consideration. Consequently, the optimization problem of equation (15) has a local solution.

From the basis transform in equation (21), the inverse fitting matrix \mathbf{F}^{-1} is already determined. Hence, the optimized wavelet synthesis $\mathbf{d}' = \mathbf{S} \mathbf{F}^{-1} \mathbf{d}$ is defined as

$$\begin{pmatrix} \mathbf{d}'_{r-3} \\ \mathbf{d}'_{r-2} \\ \mathbf{d}'_{r-1} \\ \mathbf{d}'_r \\ \mathbf{d}'_{r+1} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ (1 - a_1) & a_1 & & & \\ & (1 - a_2) & 1 & a_2 & \\ & & & (1 - a_3) & a_3 \\ & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & b_0 \\ 1 & b_1 \\ 1 & \\ b_2 & 1 \\ b_3 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_{r-3} \\ \mathbf{d}_{r-2} \\ \mathbf{w} \\ \mathbf{d}_{r-1} \\ \mathbf{d}_r \end{pmatrix}. \quad (25)$$

The corresponding lifting operations are illustrated in figure 4. Wavelet analysis is computed by the inverse operations, applied in reverse order.

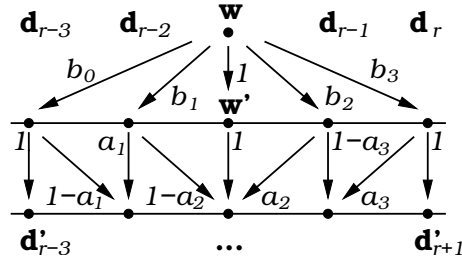


Figure 4: Vertex-manipulating lifting operations for cubic wavelet synthesis.

The approach described above works analogously for every order k . The values a_i are determined from the knot vector according to Böhm's algorithm and the values b_i are computed from a tridiagonal system whose rank is the order k . The general algorithm is summarized in the following.

2.4 The Algorithm

We describe the general wavelet synthesis inserting a single knot $u \in [t_r, t_{r+1})$ into the knot vector t_0, \dots, t_{m+k} of a B-spline curve of order $k \geq 2$. The control points \mathbf{d}_i ($i = 0, \dots, m$) and a wavelet coefficient \mathbf{w} are transformed into a refined control polygon \mathbf{d}'_i ($i = 0, \dots, m + 1$).

On the refined polygon, only the points $\mathbf{d}'_{\mathbf{q}}, \dots, \mathbf{d}'_{\mathbf{q}+k}$ with $q = r - k + 1$ need to be computed. All other control points are identical to the coarser-level points. When constructing a one-to-one correspondence between coefficients on both levels, the wavelet coefficient is placed at $\mathbf{d}'_{\mathbf{q}+s}$, $s \in \{0, \dots, k\}$. Generally, the middle $s := \lfloor k/2 \rfloor$ is a good choice, but in special cases, for example when t_r has a great multiplicity, a different choice may be necessary (we consider this later in equation (32)). Our wavelet synthesis is now defined as

$$(\mathbf{d}'_{\mathbf{q}} \quad \dots \quad \mathbf{d}'_{\mathbf{q}+k})^T = \mathbf{S} \mathbf{F}^{-1} (\mathbf{d}_{\mathbf{q}} \quad \dots \quad \mathbf{d}_{\mathbf{q}+s-1} \quad \mathbf{w} \quad \mathbf{d}_{\mathbf{q}+s} \quad \dots \quad \mathbf{d}_{\mathbf{q}+k-1})^T \quad (26)$$

representation in the space Φ is minimal, which is equivalent to equation (15).

□

If the new knot u does not exist in the knot vector $\{t_i\}_{i=0}^{m+k}$, then the requirements of theorem 1 are satisfied, independently of the choice of s . In the case $u = t_r$, this choice may be restricted. For example, if $t_{r-k+2} = t_r = u$, then we have $a_i = 0$ ($i = 1, \dots, k-1$) and s must be either 0 or 1. In any case, however, a valid choice of s exists. Considering numerical stability of the method, the numbers a_{s-1} and $(1 - a_{s+1})$ may not become too small. The other numbers a_i are less crucial, due to the monotony of $a_i \geq a_{i+1}$ ($i = 1, \dots, k-2$). For optimal stability, we choose $s \in 1, \dots, k-1$ such that

$$\min\{a_{s-1}, (1 - a_{s+1})\} \longrightarrow \max, \quad (32)$$

where $a_0 := 1$ and $a_k := 0$. If this maximum is not unique, we simply select the smallest among the optimal indices. We note that the choice of s does not impact the shape of our wavelet, except for scaling.

Algorithm: Wavelet synthesis is implemented by the following lifting scheme presented in pseudo code:

$$\begin{aligned} \text{for } (i = 0, \dots, k-1) \quad & \mathbf{d}_{q+i} \longleftarrow \mathbf{d}_{q+i} + b_i \mathbf{w}; \\ \mathbf{w} \longleftarrow & \mathbf{w} + (1 - a_s) \mathbf{d}_{q+s-1} + a_s \mathbf{d}_{q+s}; \\ \text{for } (i = s-1, \dots, 1) \quad & \mathbf{d}_{q+i} \longleftarrow a_i \mathbf{d}_{q+i} + (1 - a_i) \mathbf{d}_{q+i-1}; \\ \text{for } (i = s, \dots, k-2) \quad & \mathbf{d}_{q+i} \longleftarrow (1 - a_{i+1}) \mathbf{d}_{q+i} + a_{i+1} \mathbf{d}_{q+i+1}; \end{aligned}$$

For subsequent operations, the following symbolic substitutions are necessary:

$$\begin{aligned} \mathbf{d}'_i &:= \mathbf{d}_i \quad (i = 0, \dots, q+s-1), \\ \mathbf{d}'_{q+s} &:= \mathbf{w}, \\ \mathbf{d}'_{i+1} &:= \mathbf{d}_i \quad (i = q+s, \dots, m). \end{aligned} \quad (33)$$

The analysis scheme corresponds to the inverse operations in reverse order, where above

substitutions are already withdrawn:

$$\begin{aligned}
\text{for } (i = k - 2, \dots, s) \quad \mathbf{d}_{\mathbf{q}+i} &\longleftarrow \frac{1}{1 - a_{i+1}} (\mathbf{d}_{\mathbf{q}+i} - a_{i+1} \mathbf{d}_{\mathbf{q}+i+1}); \\
\text{for } (i = 1, \dots, s - 1) \quad \mathbf{d}_{\mathbf{q}+i} &\longleftarrow \frac{1}{a_i} (\mathbf{d}_{\mathbf{q}+i} - (1 - a_i) \mathbf{d}_{\mathbf{q}+i-1}); \\
\mathbf{w} &\longleftarrow \mathbf{w} - (1 - a_s) \mathbf{d}_{\mathbf{q}+s-1} - a_s \mathbf{d}_{\mathbf{q}+s}; \\
\text{for } (i = k - 1, \dots, 0) \quad \mathbf{d}_{\mathbf{q}+i} &\longleftarrow \mathbf{d}_{\mathbf{q}+i} - b_i \mathbf{w};
\end{aligned}$$

For a given B-spline curve or surface, our wavelet transform is computed by removing all inner knots in arbitrary order, based on the analysis formula. The result is a single polynomial segment and a (sparse) set of wavelet coefficients representing all finer geometric details. The original shape can be reconstructed by re-inserting the knots in reverse order, based on the synthesis formula.

The time-complexity for a single analysis or synthesis step is $O(k)$, since the numbers a_i and b_i are computed in linear time (due to the tridiagonal matrix used for orthogonalization) and the number of vertex updates is also linear in the B-spline order k . Consequently, the complexity for building the entire hierarchy based on $m + k + 1$ initial knots is $O(mk)$, i.e. linear in m . In contrast to most semi-orthogonal approaches [22], analysis and synthesis require the same number of operations and are both computed locally.

2.5 Discussion

Some applications, like compression, require normalization of the discrete wavelet ψ . The absolute value of the corresponding wavelet coefficient is then equal to the perturbation of the control polygon in the discrete norm when detail is removed. A quantization error introduced to a wavelet coefficient would exactly match the error introduced to the control polygon at the level of detail where the corresponding knot is inserted.

From the basis transform for discrete wavelet and scaling functions,

$$(\phi_q \dots \phi_{q+s-1} \quad \psi \quad \phi_{q+s} \dots \phi_r) = (\phi'_q \quad \dots \quad \phi'_{r+1}) \mathbf{S} \mathbf{F}^{-1}, \quad (34)$$

we obtain the discrete norm for the wavelet:

$$\|\psi\|_{l^2, \tau'}^2 = c_0 b_0^2 + c_k b_{k-1}^2 + \sum_{i=1}^{k-1} c_i (b_{i-1}(1 - a_i) + \delta_{is} + b_i a_i)^2. \quad (35)$$

Unfortunately, our discrete norm depends on the level of detail. When removing multiple knots subsequently, the discrete error in each step is minimized. However, this does not imply that the error introduced by the combined operations (with respect to the finest level) is minimized. In particular, the results even differ with respect to the order in which the knots are removed.

When working with hierarchical representations in computer-aided geometric design, symmetry is a desired property. We can conserve this property, including the least-squares minimization, by a simple modification of our algorithm, allowing to insert a set of knots u_1, \dots, u_n in a single step. The corresponding subdivision matrix $\mathbf{S} = \mathbf{S}_n \dots \mathbf{S}_1$ shapes the discrete scaling functions ϕ_q, \dots, ϕ_r , where the indices q and r are properly chosen to include all subdivision masks. The fitting matrices $\mathbf{F}_1, \dots, \mathbf{F}_n$ are now independently computed from the orthogonalization of each ψ_i with the above scaling functions.

Due to the orthogonality of the spaces $\text{span}\{\phi_i\}_{i=q}^r$ and $\text{span}\{\psi_i\}_{i=1}^n$, our least-squares optimization is still satisfied, providing a symmetric fit in case of a symmetric knot vector. The lifting operations implementing the orthogonalization are much wider now, since every wavelet coefficient \mathbf{w}_i manipulates every control point $\mathbf{d}_q, \dots, \mathbf{d}_r$. In the worst case $r - q = O(n)$ (with overlapping subdivision masks), the entire operation has time-complexity $O(n^2)$. A more efficient approach would orthogonalize every wavelet only with a fixed number of scaling functions in its neighborhood, dropping the optimization property, but maintaining symmetry and linear computation time.

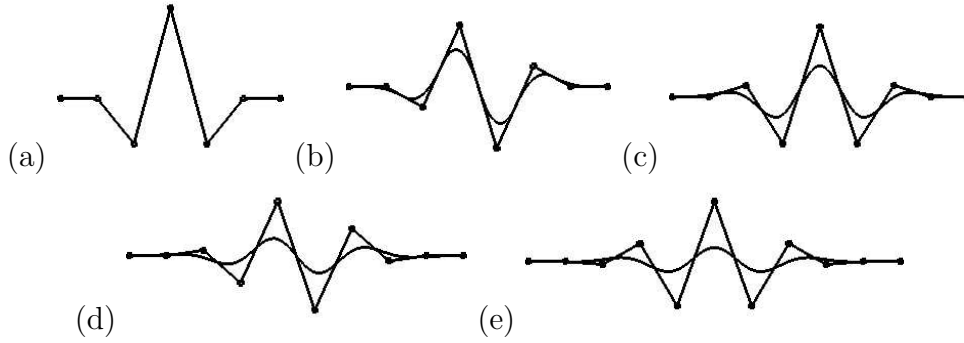


Figure 5: Discrete and continuous wavelets for uniform knots and orders $k = 2, \dots, 6$.

3 Results and Examples

3.1 Wavelet Shapes

In the following, we provide examples for the geometric shape of single-knot wavelets. Unfortunately, nothing is known about mathematical *stability* of our wavelet basis. Stability is satisfied, if the wavelets obtained after inserting an arbitrary number of knots form a *Riesz basis*, i.e. the norm of any representable function is bounded by a discrete norm of its wavelet coefficients. Most likely, the stability criterion depends on the choice of inserted knots, which has been observed for other non-uniform wavelet constructions [18]. To leave no doubt about usability of our construction, we have tested our method in different applications.

The discrete and continuous shape of wavelets of different orders ($k = 2, \dots, 6$) are provided in figure 5. To obtain these examples, we remove a single knot from a uniform knot sequence, introduce a vertical unit vector as wavelet coefficient, and re-insert the knot, again. The shape of the discrete wavelet is then offsetted to the initial (linear) shape of the control polygon, displacing $k + 1$ points. Since the support of every B-spline N_i^k is composed of k knot intervals, the continuous shape of the wavelet has a compact support of $2k$ intervals.

Of course, the wavelets are (point-) symmetric only in special cases, like uniform knots. When removing multiple knots, the shape of the corresponding wavelets is altered, resulting in mostly non-symmetric functions. In the case of a multiple knot, however, the construction is

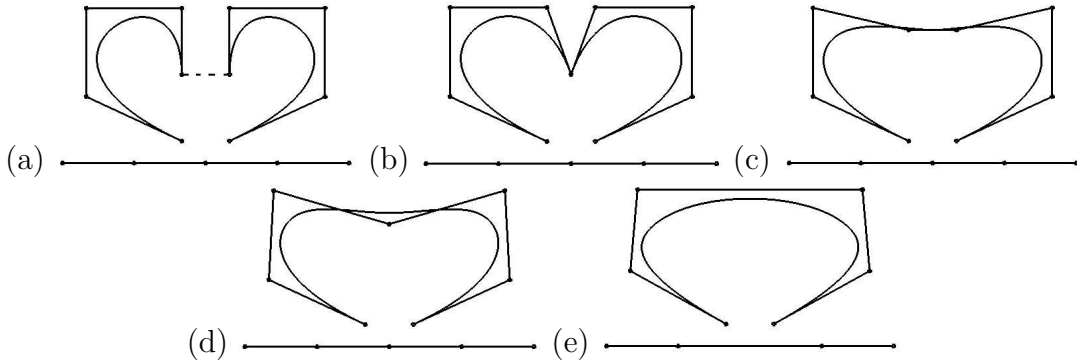


Figure 6: Removing a multiple knot. (a) The middle knot has multiplicity $k = 4$, implying discontinuity of the curve. (b-e) Removing instances of the multiple knot increases continuity of the curve.

still symmetric, as depicted in figure 6.

3.2 Applications

In the literature [12, 10, 22], many applications for wavelets can be found, including editing and sculpting of curves and surfaces, compression, progressive transmission, wavelet radiosity, and multigrid methods for solving partial differential equations (PDE's). We have tested our method in two applications:

Progressive transmission. Internet applications transmitting large data sets often suffer from low network bandwidth. Progressive transmission allows a user to explore an initial thumbnail of a data set while more detail is transmitted. The geometric model is constantly updated and displayed.

Our method is particularly suited for progressive transmission of curves and surfaces, since both analysis and synthesis are locally computed, requiring no more processing time than drawing or rendering the geometry. On the server side, all inner knots in a region of interest (ROI) are removed, for example by error-based knot selection, or simply by dyadic coarsening. The coarsest representation, followed by the removed knots with associated wavelet coefficients (in reverse order), are then sent to the client. An example for assembling a curve defined by 60

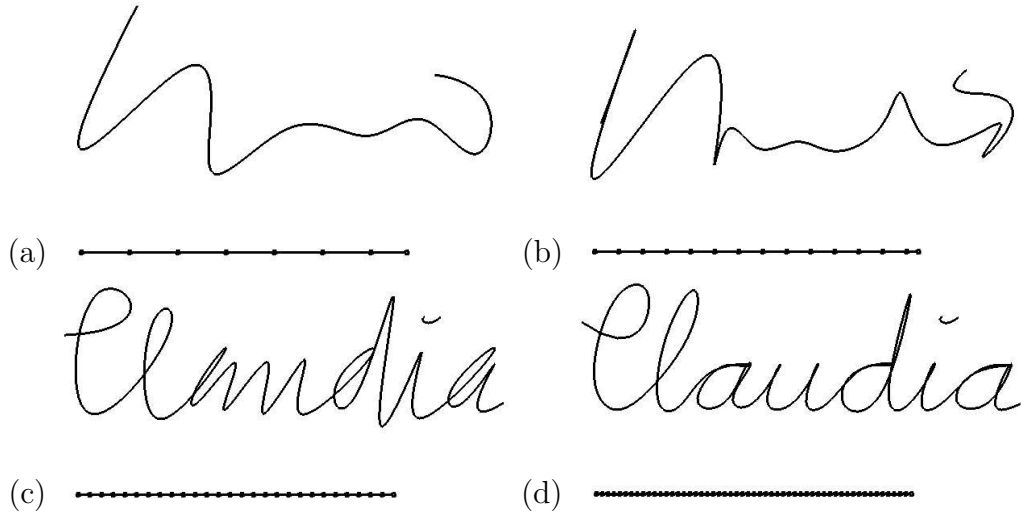


Figure 7: Progressive transmission of a curve by inserting knots and introducing geometric detail from wavelet coefficients.

control points from this information is provided in figure 7.

In the bivariate case, each knot is associated with a row or column of wavelet coefficients. If compression is desired, these coefficients may be quantized and processed by an entropy coder. Particularly in the surface case, it makes sense to transmit the complete knot vectors first, such that the final resolution of the geometry is known by the client. For rendering, all knots are inserted, where the missing wavelet coefficients are replaced by zeros.

Multiresolution editing. For the process of designing complex geometric shapes, multiple tools operating on different levels of detail are required. If the coarse shape of a geometric model needs to be altered, it is desired to keep the details of finer scale in the representation. An example is shown in figure 8, where the overall shape of a B-spline curve is altered by moving only two control points on a coarse level. After this simple edit, only few corrections on finer levels would be sufficient to obtain a pleasing result.

A severe restriction of our algorithm is that knots must be re-inserted in reverse order of removal. With some computational effort, however, this restriction can be overcome. If the designer wants to insert a certain knot s , then all intermediate knots (these that were removed after s), including s are re-inserted. Then, the intermediate knots are removed, again, such

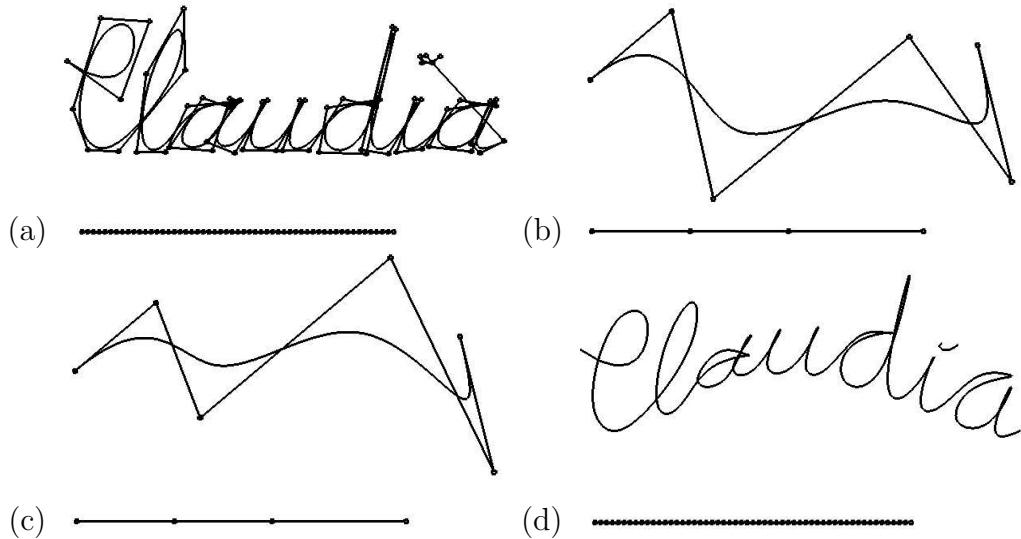


Figure 8: Multiresolution editing. (a) Initial B-spline curve with control polygon; (b) removing almost all inner knots; (c) editing the coarse-level control polygon; (d) resulting shape after re-inserting knots.

that from the user’s point of view only one knot was inserted. This way, it is also feasible to insert new knots at any level of detail.

We note that multiresolution editing is not restricted to the manipulation of control points. High-level tools like sculpting, offsetting, blending, and boolean operations on solids do also benefit from a wavelet representation. Additional knots are adaptively inserted when a prescribed error bound is not satisfied, and knots are removed, whenever sufficient accuracy is maintained without them.

3.3 Conclusions

We have introduced a simple and highly efficient method for the adaptive representation of non-uniform B-spline curves and surfaces. Based on a discrete norm, our orthogonalization process used for wavelet construction has a local solution. We found a way of factorizing the entire construction into lifting operations, providing all benefits of biorthogonal wavelets, like local computation and linear time complexity. We anticipate that our method will be very

useful in computer-aided geometric design and related applications.

References

- [1] A. Aldroubi, M. Eden, and M. Unser. Discrete spline filters for multiresolution and wavelets of l_2 . *SIAM Journal on Mathematical Analysis*, 25(5):1412–1432, September 1994.
- [2] M. Bertram. Biorthogonal loop-subdivision wavelets. *Computing*, 72(1-2):29–39, 2004.
- [3] M. Bertram, M.A. Duchaineau, B. Hamann, and K.I. Joy. Generalized b-spline subdivision-surface wavelets for geometry compression. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):326–338, 2004.
- [4] W. Böhm. Inserting new knots into b-spline curves. *Computer Aided Design*, 12(4):199–201, 1980.
- [5] C.K. Chui. *An Introduction to Wavelets*. Academic Press, 1992.
- [6] C.K. Chui and E. Quak. Wavelets on a bounded interval. In *Numerical Methods in Approximation Theory*, pages 53–75. Birkhäuser, 1992.
- [7] E. Cohen, T. Lyche, and R.F. Riesenfeld. Discrete b-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14(2):87–111, 1980.
- [8] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
- [9] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens. Wavelets on irregular point sets. *Philosophical Transactions A, Royal Society London*, 357(1760):2397–2413, 1999.
- [10] M. Duchaineau. Dyadic splines. *Ph.D. thesis, Department of Computer Science, University of California, Davis*, 1996.

- [11] G. Farin, G. Rein, N. Sapidis, and A.J. Worsey. Fairing cubic b-spline curves. *Computer Aided Geometric Design*, 4(1-2):91–103, 1987.
- [12] A. Finkelstein and D.H. Salesin. Multiresolution curves. *ACM Siggraph'94*, pages 261–268, 1994.
- [13] D. Forsey and R. Bartels. Hierarchical b-spline refinement. *ACM Siggraph'88*, pages 205–212, 1988.
- [14] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.
- [15] T. Lyche and K. Morken. Spline-wavelets of minimal support. In *Numerical Methods in Approximation Theory*, pages 177–194. Birkhäuser, 1992.
- [16] T. Lyche and K. Morken. Orthogonal decomposition of non-uniform b-spline spaces using wavelets. *Computer Graphics Forum*, 16(3):27–38, 1997.
- [17] T. Lyche, K. Morken, and E. Quak. Theory and algorithms for non-uniform spline wavelets. In *Multivariate Approximation and Applications*, pages 152–187. Cambridge University Press, 2001.
- [18] E. Quak. Non-uniform b-splines and b-wavelets. In *Tutorials on Multiresolution in Geometric Modelling*, pages 101–146. Springer, 2002.
- [19] F.F. Samavati, N. Mahdavi-Amiri, and R.H. Bartels. Multiresolution representation of surface with arbitrary topology by reversing doo subdivision. *Computer Graphics Forum*, 21(2):121–136, 2002.
- [20] P. Schröder and W. Sweldens. Spherical wavelets: efficiently representing functions on the sphere. *ACM Siggraph'95*, pages 161–172, 1995.

- [21] L.L. Schumaker. Constructive aspects of discrete polynomial spline functions. In *Approximation Theory (C.C. Lorentz, Ed.)*, pages 469–476. Academic Press, 1973.
- [22] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics—Theory and Applications*. Morgan Kaufmann, 1996.
- [23] W. Sweldens. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.