Wavelet multiresolution representation of curves and surfaces

L-M Reissell
Department of Computer Science
University of British Columbia
e-mail: reissell@cs.ubc.ca
UBC Technical Report 93-17

May 1993. Revised February 1995.

Abstract

We develop wavelet methods for the multiresolution representation of parametric curves and surfaces. To support the representation, we construct a new family of compactly supported symmetric biorthogonal wavelets with interpolating scaling functions. The wavelets in these biorthogonal pairs have properties better suited for curves and surfaces than many commonly used filters. We also give examples of the applications of the wavelet approach: these include the derivation of compact hierarchical curve and surface representations using modified wavelet compression, identifying smooth sections of surfaces and a subdivision-like intersection algorithm for discrete plane curves.

1 Introduction

Many applications require the representation of complex multiscale digitized curves and surfaces in a geometric computing environment. Geographic information systems, medical imaging, computer vision, and graphics for scientific visualization are examples of such applications. A hierarchical data representation can be used to overcome the problems of dealing with the large amount of data inherent in such applications: apart from allowing computations at selected accuracy levels, a hierarchical representation has important applications such as rapid data classification, fast display, and surface design.

1.1 Our approach

In this work, we develop multiresolution wavelet methods for the representation of curves and surfaces, and construct a family of wavelets well suited for this purpose. Using these methods, we demonstrate some of the first applications of wavelets to curve and surface representation.

The wavelet coefficients of the surface are found by computing the wavelet transform of each coordinate function separately. The representation can then be compressed; in many cases, such modified wavelet coefficients are sparse. We also describe a different variant of wavelet compression, *spatially coherent compression*, which makes the results more easily refinable for other operations, such as display and interference detection. As a byproduct of this compression method, selected scaling coefficients of the surface yield a compact hierarchical surface representation using "natural" surface building blocks.

A new family of wavelets, *pseudocoiflets*, is constructed for use in these applications. Pseudocoiflets are symmetric, compactly supported biorthogonal [8] wavelets with an interpolating scaling function[1]. The scaling functions are functions of the type first described by Deslauriers–Dubuc ([13]). The interpolation property enables us to use curve or surface samples directly as initial coefficients in the wavelet representation. It also allows us to approximate curves well using the linearly interpolated scaling coefficients, which are simple and easily obtained from the wavelet representation.

Some of the applications of a wavelet representation of curves and surfaces are illustrated with examples of curve compression, finding smooth sections of surfaces, and the outline of a curve-curve intersection algorithm. The intersection algorithm is a variant of subdivision methods, but it relies on curve approximation, rather than subdivision. Our applications are all related – they are based on the modified form of wavelet compression, which leads to a compact and natural hierarchical representation of the object.

1.2 Why are wavelets useful in curve and surface representation?

Wavelets can be very useful in the *analysis* and *preparation* of the curve/surface for other operations. In other application areas, wavelets have been used for compression, for edge and discontinuity detection, and denoising [14], [22], [15]. We list below some of the main features of wavelets:

- Good approximation properties.

  The multiresolution approximations given by the orthonormal wavelet decomposition are the best possible L^2 approximations from the given multiresolution spaces, and simple wavelet thresholding gives L^2-optimal compression. While the L^2-optimality does not guarantee minimum distance approximations, it does give excellent curve and surface approximations, especially in the absence of sharp discontinuities. The good approximation properties of wavelets are the basis for their success in applications. We note that wavelet approximation works in spaces other than L^2 as well [14].

- The wavelet coefficients provide a precise measure of the approximation error.

  The behavior of the approximation error is well understood. The approximation properties of the wavelet representation are tied to the number of vanishing moments of the wavelet.

- Space-frequency localization.

  Wavelets identify frequency changes in small domains: the size of the wavelet coefficients gives information about the local "smoothness" of the underlying data.

[1]Pseudocoiflets derive their name from the observation that their construction can be obtained by building biorthogonal wavelets with "coiflet-like" moment properties – coiflets, in turn, are an orthonormal wavelet family specially constructed by Daubechies [11] to provide good approximation of function samples by scaling coefficients.

- Good numerical properties.

  Numerical calculations in the wavelet basis are fast and robust [3].

- Efficiency.

  The parametric wavelet representation is simple to compute ($O(n)$) and, as discussed in Section 3, it respects standard operations such as translation, rotation and scaling. The representation leads to easily parallellizable, efficient algorithms, and allows the use of different coefficient transformation techniques.

- Hierarchical surface representation and analysis tools.

  The wavelet representation we use is very closely related to hierarchical design schemes, such as hierachical splines [18]. Hierarchical design is also possible using the wavelet representation, although we concentrate more on the uses of the wavelet representation for curve and surface analysis. A compact hierarchical representation for a curve or surface can be obtained from a modified wavelet compression algorithm (Section 5). The analysis of wavelet coefficients can also be used to partition the curve or surface into areas of varying complexity for use in other operations or geometric algorithms.

1.3 Why are pseudocoiflets useful?

Pseudocoiflets, described in Section 4 are constructed to satisfy the following requirements:

- Smooth, symmetric, nonoscillating reconstructing scaling functions.
- Good space-frequency localization and regularity for both biorthogonal wavelets.
- Short filters.
- Interpolating reconstructing scaling function.

The first three of these are generally desirable properties in curve and surface representation. The interpolation requirement is more specific, but very useful. Interpolating scaling functions allow:

- Use of data samples as initial scaling coefficients.
- Fast, local schemes for curve and surface interpolation.
- Interchanging control points and curve points.
- Use of scaling coefficients in approximation.

Any wavelet application which uses the scaling coefficients themselves to approximate data needs an underlying wavelet, which produces scaling coefficients close to the data. Pseudocoiflets were designed to give scaling coefficients with these approximation properties.

For example, curve and surface data are usually provided as samples. Using these samples directly as starting coefficients for wavelet algorithms normally means allowing an initial approximation error. With interpolating scaling functions, the use of data samples is accurate.

Local, fast $O(n)$ interpolation using the scaling functions is useful for instance in design applications, where points on curves/surfaces are placed in desired locations. Interpolating scaling functions also allow one to go easily from control points to the actual corresponding curve segments. For very fast display using minimum storage, a curve or surface represented with an interpolating scaling function can be progressively "filled in" with refined data in simple filtering steps.

With interpolating scaling functions, we can also use the scaling coefficients directly as an easily obtained piecewise linear approximation to the original curve or surface. Scaling coefficients from different levels can be used in an adaptive approximation derived from wavelet compression. These applications again require wavelets which produce "good" scaling coefficients, such as pseudocoiflets. The examples in Section 5 illustrate these uses.

1.3.1 Comparison with other wavelets

A simple example in Figure 1 illustrates why the properties of the reconstructing scaling function are important, and why the easiest wavelets to use, the orthonormal Daubechies wavelets, are not ideal for curve and surface representation. The multiresolution approximation of simple data given by the Daubechies wavelet D_8 ([12]) oscillates much more than the same multiresolution approximation using the pseudocoiflet P_4.

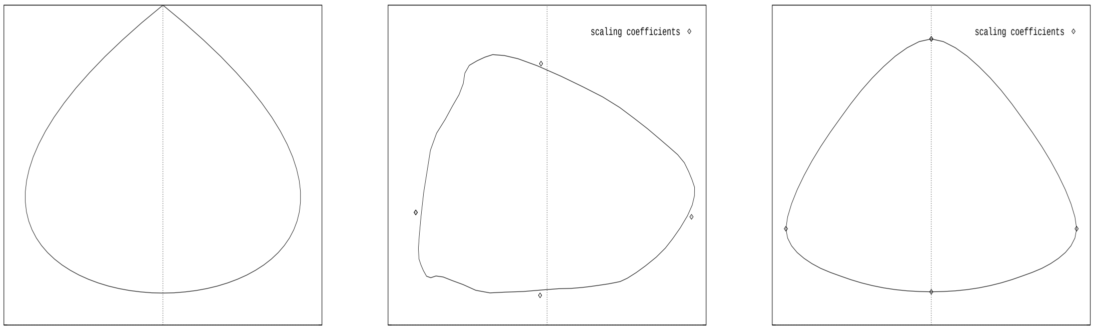


Figure 1: Original curve; approximations using Daubechies wavelets and pseudocoiflets

B-splines are in some respects ideal as scaling functions in curve representation. However, biorthogonal B-spline wavelet pairs with short supports tend to have very uneven properties [8] (see Figure 4). Semiorthogonal spline wavelets [6] have dual wavelets with *infinite* support. So to obtain wavelets with even properties and short filters, spline wavelets are not necessarily the best choice.

The following figures give an example of multiresolution approximation with scaling coefficients, using the pseudocoiflets constructed in this paper. The scaling coefficients, with linear interpolation, form good approximations of the original curve.

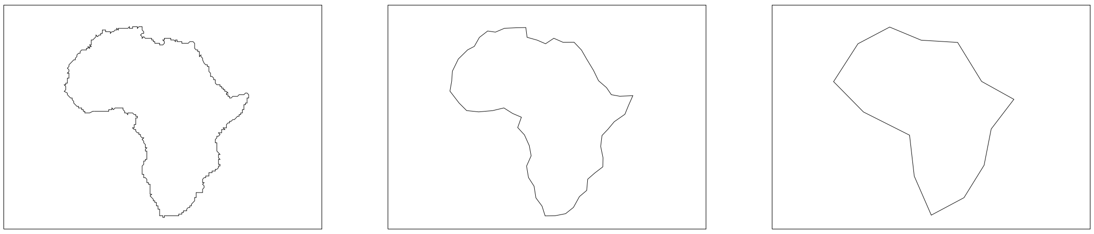


Figure 2: Original curve and scaling coefficient curves (1/8 and 1/32 of points)

1.4 Related work

In other application areas, such as image and signal processing, wavelets have been used for several years to provide multiresolution data representation ([22], [10]). Similarly, wavelet techniques for image and elevation surface compression have been studied in for instance [1], [14].

Methods of hierarchical curve and surface representation based on subdivision (e.g. strip trees, quadtrees, spline subdivision) have been used extensively in geometric modeling [2], [9], [19]. Hierarchical methods based on a scale space representation have been popular in computer vision [21]. Other hierarchical representations can be constructed for instance from variable knot spline approximation. The multiresolution representation of curves using the scaling coefficients from the wavelet representation is very closely related to hierarchical design schemes, such as hierachical splines [18], with the important difference that wavelets allow better control of the error in the approximation.

Many of the methods above use multiresolution smoothing, which convolves the data with an appropriately scaled kernel – wavelets are descended from these methods. One advantage of wavelets is that the approximation properties of wavelet multiresolution filters yield hierarchical curves and surfaces which lie close to the original data, which does not always happen with other filters [21]

Interpolating functions corresponding to multiresolution have been studied by Deslauriers–Dubuc ([13]) and [17]. These were not associated with wavelet spaces. Donoho [16] has constructed interpolating wavelets, where the interpolating function is used as a *wavelet*, not as a scaling function, as here. The use of an interpolating function as a wavelet is unorthodox, since its integral does not vanish, and this setup is used to prove theoretical approximation results, rather than in building practical wavelet tools.

The wavelet-based intersection algorithm is a variant of the basic subdivision algorithm [9], and is related to the methods described in [19]. However, the algorithm does not rely on curve subdivision, like the above schemes, but rather on hierarchical curve approximation. In practice, this results in smaller error boxes. In another departure from the standard method, the algorithm is designed to be supplemented by other, potentially faster ones on suitable curve sections, which can be identified using the underlying wavelet representation.

1.5 Organization of the paper

Section 2 contains selected preliminaries on wavelets. More details on general wavelet theory can be found for instance in [12]. Section 3 presents definitions, notation and some basic results on the parametric wavelet representation of curves/surfaces. Section 4 gives the construction of pseudocoiflets; it includes a brief comparison of pseudocoiflets to popular wavelet families.

Examples of the use of wavelets/pseudocoiflets in curve and surface representation are given in Section 5. This section contains three examples: compression of curve and surface data using a modified thresholding technique, analyzing surfaces for smooth sections, and an application to the intersection of plane curves. These examples are all related, and rely on the use of a compact, hierarchical, wavelet-based representation of the geometric object developed in this paper.

Proofs of the results are given in the Appendix.

2 Wavelet preliminaries

Wavelets are collections of functions in L^2 constructed from a *basic wavelet* ψ by dilations and translations. Wavelets are used for representing the local frequency content of functions; for this, the basic wavelet and its Fourier transform should both be reasonably well localized, and the wavelet should have zero mean. We will only consider discrete families of wavelets formed using dilations by powers of 2 and integer translations:

$$\psi_{i,j} = \sqrt{2^{-i}}\psi(2^{-i}x - j), \quad i, j \in \mathbb{Z}.$$

Multiresolution is an important general method for constructing orthonormal wavelet bases for L^2 consisting of these discrete wavelets ([22]). In multiresolution schemes, wavelets have corresponding *scaling functions* ϕ, whose analogously defined dilations and translations $\phi_{i,j}$ span a nested sequence of multiresolution spaces $V_i, i \in \mathbb{Z}$. Wavelets $(\psi_{i,j})_{ij}$ form orthonormal bases for the orthogonal complements $W_i = V_{i-1} - V_i$, and for all of L^2.

2.0.1 Biorthogonal wavelets

We will primarily use the more general biorthogonal families of wavelets of Cohen, Daubechies, and Feauveau ([8]). These wavelets $(\psi_{i,j}, \tilde{\psi}_{i,j})$ do not form orthogonal bases, but they are dual bases for L^2: $< \psi_{i,j}, \tilde{\psi}_{i',j'} > = \delta_{ii'}\delta_{jj'}$, and the following "analyzing" and "reconstructing" relations hold:

$$f = \sum_{ij} < f, \psi_{i,j} > \ \tilde{\psi}_{i,j} = \sum_{ij} < f, \tilde{\psi}_{i,j} > \ \psi_{i,j}.$$

In this case we have two dual multiresolution spaces V_i, $\tilde{V}_i$, which may coincide, and the complement wavelet spaces W_i, $\tilde{W}_i$. We do not necessarily have orthogonality between W_i and V_i, but have instead $W_i \perp \tilde{V}_i$ and $V_i \perp \tilde{W}_i$. The analyzing and reconstructing roles of the dual and primal wavelets can be interchanged, but we will stick to the above convention.

The *wavelet transform* or *decomposition* of a function f is the representation of f in the reconstructing wavelet basis: $f = \sum_{ij} c_{ij} \tilde{\psi}_{i,j}$; the coefficients $c_{ij} = < f, \psi_{i,j} >$ are called the *wavelet coefficients* of f. The *scaling coefficients* of f at level i are defined similarly as inner products $< f, \phi_{i,j} >$, or, equivalently, as the coefficients d_{ij} of the projection $P_i f = \sum_j d_{ij} \tilde{\phi}_{i,j}$ of f into the i^{th} multiresolution space $\tilde{V}_i$.

Biorthogonal bases have several advantages over orthonormal ones. First, it is much easier to construct biorthogonal wavelet bases than orthonormal ones ([8]). In addition, the different roles of the wavelets allow a more flexible construction of bases suited to the application.

2.0.2 Wavelet filters

In practice, the function f is usually given as an approximation in a multiresolution space using function samples. The scaling coefficients of this initial multiresolution approximation form the *initial coefficients* $\mathbf{s}$. The wavelet transform is then obtained by iterating Mallat's filtering scheme:

$$\begin{array}{ccccccc} & H & & H & & H & \\ \mathbf{s} & \longrightarrow & \mathbf{s}_1 & \longrightarrow & \mathbf{s}_2 & \longrightarrow & \dots \\ & G & & G & & G & \\ & \searrow & & \searrow & & \searrow & \\ & & \mathbf{w}_1 & & \mathbf{w}_2 & & \dots \end{array} \tag{1}$$

The filters $H = (h_j)$ and $G = (g_j)$ are the analyzing *scaling* and *wavelet* filters, respectively. Reconstruction is performed using the dual filters $(\tilde{h}_j)$ and $(\tilde{g}_j)$.

For compactly supported biorthogonal wavelets, only the scaling filters need to be specified. The wavelet filters are obtained from the scaling filters using the exact reconstruction condition. The (primal) wavelet filter is constructed from the *dual* scaling filter using the standard mirror filter construction, and similarly the dual wavelet from the primal scaling filter.

2.0.3 Approximation properties of wavelets

The approximation properties of the wavelet decomposition are determined by the number N of vanishing moments of the wavelet ψ. The number of vanishing moments is the largest number N for which

$$\int x^l \psi(x)\, dx = 0, \quad l = 0, \dots, N-1. \tag{2}$$

The number N determines the accuracy of approximating a function f by its projections in the multiresolution spaces V_i. The error decreases as 2^{iN} as i tends towards finer scales:

Approximation result (Strang and Fix [25]).

Suppose (V_i) is a multiresolution with a wavelet ψ. If ψ has N vanishing moments, the error of approximating a function f with at least N derivatives from the multiresolution space V_i is:

$$\| f - P_i f \| \leq C 2^{-iN} \| f \|_{W^N} .$$

(The norm of the function is the Sobolev space norm obtained from the derivative norms $\| f \|_{W^N}^2 = \sum_n \| f^{(n)} \|^2$.) The vanishing moment condition also means that polynomials of degree up to $N-1$ are spanned by the corresponding multiresolution spaces.

For biorthogonal wavelets, the vanishing moments of the *analyzing* (primal) wavelet determine the degree of approximation from the *reconstructing* (dual) multiresolution spaces.

Using the fact that the L^2-norm of a function f is $\sum_{ij} |w_{ij}|^2$, where w_{ij} are the wavelet coefficients of f, the above also implies that the wavelet coefficients of a sufficiently smooth function decay at least as a power of 2^N, if ψ has N vanishing moments:

$$\max_j |w_{ij}| \leq C 2^{-iN}.$$

So wavelets with a small number of vanishing moments, for example Haar wavelets, result in larger coefficients in smooth sections. As a consequence, these wavelets will not produce as marked a contrast in coefficient size between smooth and non-smooth sections of data as wavelets with more vanishing moments, and they will not approximate functions as rapidly.

2.0.4 Compression using wavelet coefficients

In wavelet compression, we approximate data from spaces Σ_n consisting of those functions with n nonzero wavelet coefficients. Unlike the multiresolution spaces, these are not linear spaces, since they are not closed under addition. In L^2, this approximation is based on the following: if A is the set of coefficients (i,j) chosen to be in the approximating function, the L^2 norm of the approximation error is

$$\| \text{error} \|^2 \simeq \sum_{(i,j) \neq A} |w_{ij}|^2.$$

This means that the L^2 error is smallest when the n largest wavelet coefficients are chosen for the approximation. This corresponds to simple threshold compression of the wavelet coefficients. From the above, it is clear that for smooth data, compression rates improve as the number of vanishing moments of the wavelet increases.

3 Parametric wavelet decomposition: basic definitions and results

In this discussion, we will focus on the wavelet decomposition of curves; the extension of the definitions to higher dimensions will be straightforward. We consider parametrically defined curves

$$C = \{ \mathbf{x} \in \mathbb{R}^N : \mathbf{x} = \mathbf{f}(t), t \in \mathbb{R} \},$$

where the component functions f^k are in L^2. Curves given on intervals are treated by using modified wavelets defined on intervals (e.g. [7]).

We also select a biorthogonal wavelet family determined by the analyzing scaling function ϕ and wavelet ψ, and the corresponding reconstructing functions $\tilde{\phi}$ and $\tilde{\psi}$. At this point, we are not limiting the wavelets to finitely supported ones.

3.1 Parametric wavelet decomposition

The *wavelet decomposition* of the curve C is the collection of wavelet decompositions of each coordinate function $f^k = \sum < f^k, \psi_{i,j} > \tilde{\psi}_{i,j}$. The *wavelet coefficients* of the curve are given by $c^k_{ij} = < f^k, \psi_{i,j} >$. Similarly, the *scaling coefficients* of the curve are given by the scaling coefficients for each coordinate: $d^k_{ij} = < f^k, \phi_{i,j} >$. The (normalized) *coefficient curve* $\mathrm{Coeffs}(C, i)$ on level i consists of the linearl interpolation of the points defined by the scaling coefficients multiplied by $2^{-i/2}$.

The *(multiresolution) approximation curve* $\mathrm{Approx}(C, i)$ at multiresolution level i is constructed from the scaling coefficients d^k_{ij} componentwise by $P_i f^k(t) = \sum_j d^k_{ij} \tilde{\phi}_{ij}(t)$. The *error* between two approximation curves is given by reconstruction from the wavelet coefficients using the wavelet basis functions.

The wavelet coefficients of a curve or surface can be modified to obtain better approximations, to compress surface data, and to eliminate noise. Methods for this include adaptive wavelet coefficient thresholding ([14]), quantization ([1]) and the shrinking of wavelet coefficients ([15]). Wavelet coefficients also provide a way to analyze the curve for discontinuities (see [12], [22]). The above wavelet coefficient transformation techniques will lead to *modified* wavelet and scaling coefficients and approximation curves. The modified data is obtained by first performing the standard decomposition of the original data $\mathbf{s}$ into scaling coefficients $\mathbf{s}_i$ and wavelet coefficients $\mathbf{w}_i$:

$$\begin{array}{ccccccc} \mathbf{s} & \longrightarrow & \mathbf{s}_1 & \longrightarrow & \mathbf{s}_2 & \longrightarrow & \dots \\ & \searrow & & \searrow & & \searrow & . \\ & & \mathbf{w}_1 & & \mathbf{w}_2 & & \dots . \end{array}$$

The wavelet coefficients are then transformed from $\mathbf{w}_i$ to $\mathbf{w'}_i$, as in for instance [14], [15], and reconstruction is performed on the new wavelet coefficients, leading to a new sequence of scaling coefficients $\mathbf{w'}_i$:

$$\begin{array}{ccccccc} \mathbf{s'} & \longleftarrow & \mathbf{s'}_1 & \longleftarrow & \mathbf{s'}_2 & \longleftarrow & \dots \\ & \nwarrow & & \nwarrow & & \nwarrow & \\ & & \mathbf{w'}_1 & & \mathbf{w'}_2 & & \dots \end{array}$$

Since the underlying wavelets have compact (or effectively compact) support, algorithms involving wavelet decomposition and reconstruction of parametric curves are fast and parallellizable. We should note that the wavelet transform is sensitive to the parametrization used, although the main features of the decomposition (rate of decay of coefficients, for example) will be invariant to changes such as parameter shifts.

The parametric wavelet decomposition of a curve behaves well with respect to the standard affine transformations:

Proposition 3.1

- *The normalized coefficient curve of a translated, rotated or scaled curve is obtained by translation, rotation or scaling from the original normalized coefficient curve, respectively.*
- *The approximation curve for a translated, rotated or scaled curve is also obtained by translation, rotation or scaling from the original approximation curve.*

The proofs are given in the Appendix. For example, for coefficient curves, these properties follow immediately from the observation that, for a function f in $L^2(R)$, the operation P_i giving the normalized scaling coefficients of f on level i is linear.

4 Pseudocoiflets: wavelets with interpolating scaling functions

In this section we construct a family of biorthogonal wavelets, *pseudocoiflets*, well suited for the representation of geometric objects. The pseudocoiflet construction was motivated by the requirements of curve and surface representation outlined in the introduction.

More specifically, we will construct compactly supported symmetric biorthogonal wavelets, for which one of the scaling functions is interpolating. The construction is based on the methods of Cohen, Daubechies, and Feauveau [8], and the interpolating scaling functions are Deslauriers–Dubuc functions ([13]). These wavelets are called pseudocoiflets, after the coiflet family of orthonormal wavelets constructed by Daubechies [11]. We observe that wavelets with interpolating scaling functions automatically have coiflet-like moment properties and so can be used in the same applications as coiflets.

4.1 Coiflet-like wavelets and interpolation: definitions

Coiflets are orthonormal wavelets whose scaling function ϕ also has higher vanishing moments. This approach has advantages for instance in the computation of scaling coefficients from function samples: the normalized scaling coefficients of f can be used at finer scales i to approximate the values of the function f, and vice versa. More specifically, if f is C^N and we have sampled f at intervals of length $h = 2^i$ we have ([12])

$$f(2^i k) = 2^{-i/2} < f, \phi_{i,k} > + O(h^N), \tag{3}$$

provided that the moments $l = 1, \ldots, N-1$ are zero for ϕ.

A biorthogonal family is called *coiflet-like* if there is an N such that both wavelets have the vanishing moment properties of (2) and one of the scaling functions satisfies (2) for $l = 1, \ldots, N$. A scaling function ϕ is *interpolating* when its scaling filter coefficients $(h_j)_j$ satisfy

$$h_{2j} = \frac{1}{\sqrt{2}}\delta_{j,0}.$$

In this case, we will also call the corresponding filter and the filter transfer function $m_0(\xi) = \frac{1}{\sqrt{2}}\sum h_j e^{-ij\xi}$ interpolating.

Iterating the reconstruction using an interpolating scaling filter (and zero wavelet coefficients) refines the values at each stage. We note that no compactly supported orthonormal wavelet basis can have interpolating scaling functions.

4.2 Construction outline

4.2.1 Construction requirements

We will construct a family P_{2N} of scaling functions $(\phi, \tilde{\phi})$ and corresponding wavelets satisfying the exact reconstruction condition (see [8]) and with the following properties:

- The scaling functions ϕ and $\tilde{\phi}$ are symmetric.
- The first 2N moments for ψ and $\tilde{\psi}$ vanish.
- $\tilde{\phi}$ satisfies the scaling function vanishing moment condition for 2N.
- ϕ, ψ, $\tilde{\phi}$, and $\tilde{\psi}$ are compactly supported.
- $\tilde{\phi}$ is interpolating.

We will outline the construction here; proofs are given in the Appendix. We follow the methods of [8]: for a given N, we find the appropriate trigonometric polynomials $m_0(\xi)$ and $\tilde{m}_0(\xi)$ corresponding to ϕ and $\tilde{\phi}$. We assume first that both $m_0(\xi)$ and $\tilde{m}_0(\xi)$ correspond to filters which are symmetric and consist of an odd number of elements. The moment conditions on the wavelet and the scaling function for such a filter transfer function m_0 can then be rewritten in the following way:

$$m_0(\xi) = (1 + \cos\xi)^N P_1(\cos\xi). \tag{4}$$

$$m_0(\xi) = 1 + (1 - \cos\xi)^N P_2(\cos\xi) \tag{5}$$

Here, both P_1 and P_2 are trigonometric polynomials of $\cos\xi$, and $2N$ is the number of vanishing moments required. (For details, see [8].) We will note that (4) is the only form the trigonometric polynomial $\tilde{m}_0$ can take if $\tilde{\phi}$ is to be interpolating.

4.2.2 The interpolation condition

We first observe that interpolating scaling functions can be obtained as a special case from the construction of coiflet-like scaling functions. For these scaling functions, both moment conditions (5) and (4) are satisfied.

The requirement that the wavelet is coiflet-like can be expressed as

$$(1+x)^N P_1(x) - (1-x)^N P_2(x) = 1, \tag{6}$$

where $x = \cos\xi$. This equation has the solution

$$P_1(x) = \frac{1}{2^N} \sum_0^{N-1} \binom{N-1+k}{k} \frac{1}{2^k}(1-x)^k \;+\; (1-x)^N F(x), \tag{7}$$

$$P_2(x) = -P_1(-x) \tag{8}$$

where F is an arbitrary odd polynomial. (See [12] for a proof of this in a slightly different form.) For $F = 0$ these P_1 correspond to functions studied by Deslauriers and Dubuc (see e.g. [13]).

In addition, interpolating scaling functions are also obtained this way by the following observation:

Proposition 4.1

- *Assume that the trigonometric polynomial $\tilde{m}_0$ is a solution of the coiflet equation (6) for $2N$ vanishing moments, that is, $\tilde{m}_0$ satisfies (5) and (4), where P_1 and P_2 are as in (7) and (8). Then $\tilde{m}_0$ is interpolating.*
- *Conversely, compactly supported symmetric interpolating scaling functions are coiflet-like, have an even number of vanishing moments, and the corresponding filter consists of an odd number of elements.*

4.2.3 The biorthogonality conditions

The interpolating trigonometric polynomials $\tilde{m}_0$ obtained in the above way are then inserted into the biorthogonality conditions of [8] to find the dual trigonometric polynomials m_0. We observe that the biorthogonality condition can always be satisfied if $\tilde{m}_0$ is a solution of the coiflet equation (6). The necessary condition for biorthogonality for m_0 and $\tilde{m}_0$ is

$$m_0(\xi)\overline{\tilde{m}_0(\xi)} + m_0(\xi+\pi)\overline{\tilde{m}_0(\xi+\pi)} = 1. \tag{9}$$

For m_0 and $\tilde{m}_0$ as in (5) and (4), with $2N$ and $2\tilde{N}$ giving the numbers of vanishing moments, the condition can be expressed as

$$(1+x)^{N+\tilde{N}}\tilde{P}(x)P(x)+(1-x)^{N+\tilde{N}}\tilde{P}(-x)P(-x)=1. \tag{10}$$

We have the following result:

Proposition 4.2 *If $\tilde{P}(x)$ is a solution (7) for P_1 to the coiflet equation (6), then there is a polynomial P such that P and $\tilde{P}$ solve the biorthogonality equation (10) with $N=\tilde{N}$. The unique minimum degree solution P corresponding to the minimum degree $\tilde{P}$ has degree $3N-2$.*

4.3 The pseudocoiflet family P_{2N}

The family of pseudocoiflets P_{2N}, a wavelet family (ϕ,ψ), $(\tilde{\phi},\tilde{\psi})$ satisfying the necessary biorthogonality condition (10), is now obtained by the following procedure.

Construction of pseudocoiflets P_{2N}

1. Let $\tilde{P}$ and P be the trigonometric polynomials $m_0(\xi)=(1+\cos\xi)^N P(\cos\xi)$ and $\tilde{m}_0(\xi)=(1+\cos\xi)^{N_1}\tilde{P}(\xi)$.
2. Find the minimal degree solution (7) for $\tilde{P}$ by letting $\tilde{P}=P_1$.
3. Find the minimal degree solution P for the given $\tilde{P}$ using the linear system in (10). This solution exists by the above proposition.
4. Evaluate the filter coefficients from P and $\tilde{P}$.

The above construction implies that there is an exact reconstruction filtering scheme corresponding to the functions (ϕ,ψ), $(\tilde{\phi},\tilde{\psi})$. It does not yet guarantee that the constructed functions (ϕ,ψ), $(\tilde{\phi},\tilde{\psi})$ are in L^2, or that the wavelets derived from $\psi,\tilde{\psi}$ form a dual basis. A necessary and sufficient condition for the functions (ϕ,ψ), $(\tilde{\phi},\tilde{\psi})$ to define a true biorthogonal L^2-wavelet family has been given by Cohen in [8]. This condition can be easily shown to hold for the first few members of the family P_{2N}, and so we have L^2-biorthogonal wavelet bases corresponding to at least these N. However, the exact reconstruction property for biorthogonality holds in all cases without further proof for the corresponding filters. The remaining biorthogonality issues will be discussed elsewhere. The properties of the interpolating scaling functions have been studied outside a wavelet context by Deslauriers and Dubuc.

The following properties of the pseudocoiflets ψ and $\tilde{\psi}$ follow immediately from the construction:

Properties of pseudocoiflets P_{2N}

- The pseudocoiflets ψ and $\tilde{\psi}$ have $2N$ vanishing moments, as does the scaling function $\tilde{\phi}$.
- The reconstructing scaling function $\tilde{\phi}$ is interpolating.
- The scaling functions are symmetric.

- The degrees of $\tilde{m}_0$ and m_0 are $N-1$ and $3N-2$, respectively.
- The lengths of the pseudocoiflet P_{2N} reconstructing and analyzing filters are $4N-1$ and $6N-1$, respectively.

4.3.1 Extensions

We note that it is possible to choose different values of $\tilde{N}$ and N in (10), leading to a construction of a family $P_{2\tilde{N},2N}$ of pseudocoiflets consisting of a family of analyzing functions, depending on N, for each reconstructing scaling function with moment properties given by $\tilde{N}$. Longer filters correspond to underlying functions with less oscillation; this is important if the function ϕ is used as a reconstructing function. Other variations of the construction can also be obtained, for instance, by considering longer than minimal length reconstructing filters.

4.4 Examples

The filter coefficients for the pseudocoiflets with $N=1$ and $N=2$ are listed below in Table 1. Note that the coefficients are exact. The pseudocoiflet for $N=1$ has the hat function as the reconstructing scaling function and the filter pair equals the corresponding spline-based biorthogonal filter of [8]. The pseudocoiflet scaling functions for $N=2$ are pictured in Figure 3.

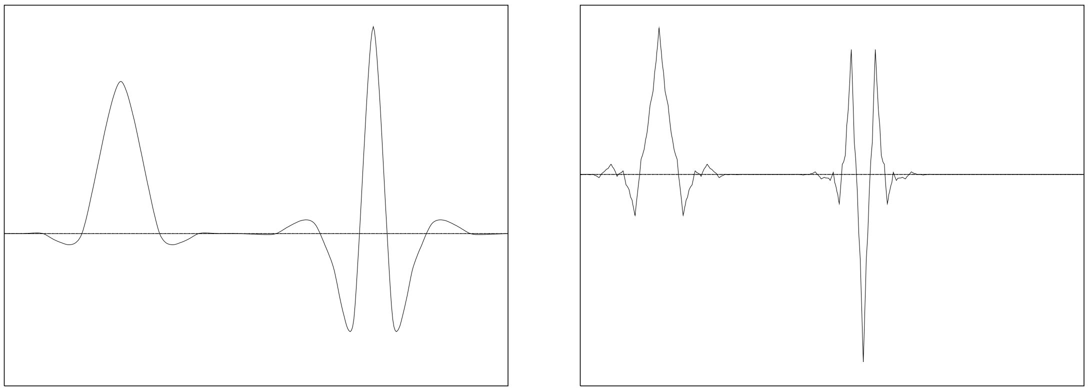


Figure 3: Pseudocoiflet P_2 scaling function, wavelet and the duals

analyzing filter N = 1 multiply by $\frac{1}{\sqrt{2}}$	reconstructing filter N = 1 multiply by $\sqrt{2}$	analyzing filter N = 2 multiply by $\frac{1}{\sqrt{2}}$	reconstructing filter N = 2 multiply by $\sqrt{2}$
		-0.00390625	
		0	
		0.0703125	
		-0.0625	-0.03125
-0.25		-0.24609375	0
0.5	0.25	0.5625	0.28125
1.5	0.5	1.359375	0.5
0.5	0.25	0.5625	0.28125
-0.25		-0.24609375	0
		-0.0625	-0.03125
		0.0703125	
		0	
		-0.00390625	

Table 1: Scaling filter coefficients for pseudocoiflets with $N = 1, 2$.

The wavelet filter coefficients are obtained by the mirror filter construction from the scaling filters. The analyzing wavelet is obtained from the reconstructing scaling function, and vice versa. The wavelet coefficients for $N = 2$ are, for the analyzing filter,

$$(0.03125, \quad 0, \quad -0.28125, \quad 0.5, \quad -0.28125, \quad 0, \quad 0.03125)$$

multiplied by $\sqrt{2}$, and, for the reconstructing wavelet filter,

$$(-0.00390625, \quad 0, \quad 0.0703125, \quad 0.0625, \quad -0.24609375, \quad -0.5625, \quad 1.359375,$$

$$-0.5625, \quad -0.24609375, \quad 0.0625, \quad 0.0703125, \quad 0, \quad -0.00390625)$$

multiplied by $\frac{1}{\sqrt{2}}$. Note that the application of the analyzing wavelet filter to data has to be shifted by one step from the application of the scaling filter to achieve exact reconstruction.

4.5 Comparison with other wavelets

In this section we compare pseudocoiflets P_4 to other wavelets with similar numbers of vanishing moments. We look at the Fourier transforms of the scaling functions and their duals (the wavelets themselves have similar properties, since they are obtained directly from the scaling functions).

We compare the pseudocoiflets to Daubechies wavelets D_8, and biorthogonal spline wavelet pairs $S_{3,3}$ and $S_{3,7}$ with 3 and 3, and 3 and 7 vanishing moments respectively [12]. Aside from the specific uses of interpolation, pseudocoiflets compare well with these wavelets if the requirements of the introduction are considered:

- Smooth, symmetric, nonoscillating reconstructing scaling functions.
- Good space-frequency localization and reasonable regularity for both wavelets.
- Short filters.

B-splines are in some respects ideal as scaling functions. However, biorthogonal B-spline wavelet pairs with short supports tend to have very uneven properties and filter lengths [8]. This is illustrated in Figure 4, where the dual wavelet can be seen to be very irregular. Semiorthogonal spline wavelets [6] have dual wavelets with *infinite* support. So if we wish to obtain wavelet pairs with even properties and short filters, we need to look at other wavelets.

To obtain more even spline-based wavelets, Daubechies has constructed modified wavelets in [8]. These are the wavelets that were used with success in [1]. However, in this case, neither scaling function is a cardinal B-spline, and some of the advantages of using spline-based wavelets are now lost.

Figure 5 compares the Fourier transform magnitudes of the primal scaling function for the filters D_8, P_4, and $S_{3,7}$. (In the spline case, this is a quadratic spline.) Figure 5 compares the Fourier transform magnitudes of the duals for D_8, P_4, $S_{3,3}$, and $S_{3,7}$.

The plots of the Fourier transform of the Daubechies wavelets show subsidiary "lobes". This affects the regularity of the wavelet and its frequency localization. The Fourier transform of the B-spline is the "best possible" in the biorthogonal setup. However, the duals of the biorthogonal spline wavelets are not as well behaved – in fact, the dual wavelets are very irregular. This is very noticeable for the spline wavelets with 3 vanishing moments for both wavelets (the large "bumps" in the Fourier transform in Figure 5) and can also be seen from the shape of the duals in Figure 4.

Pseudocoiflets are almost as good as splines on the primal side, and on the dual side, they are much better. Pseudocoiflets have better regularity properties than the orthonormal Daubechies filters D_8 (and the orthonormal coiflets with 4 vanishing moments, which are very similar to Daubechies wavelets).

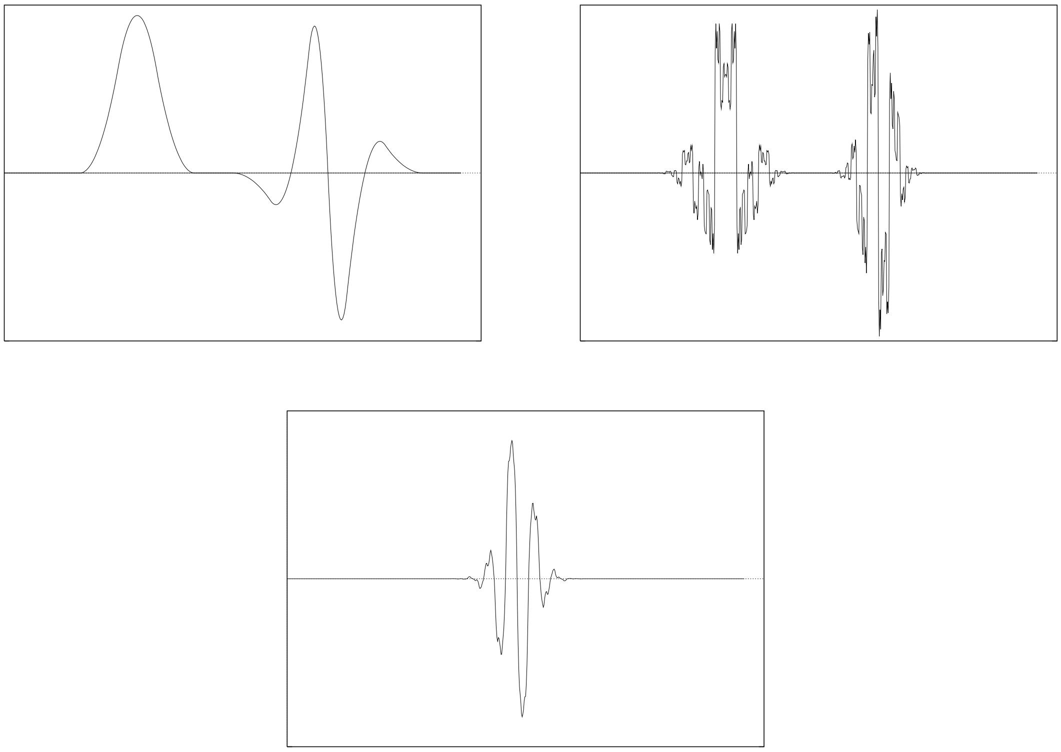


Figure 4: Spline wavelets $S_{3,3}$ and $S_{3,7}$: Biorthogonal pair corresponding to 3 vanishing moments each; dual wavelet corresponding to 7 vanishing moments.

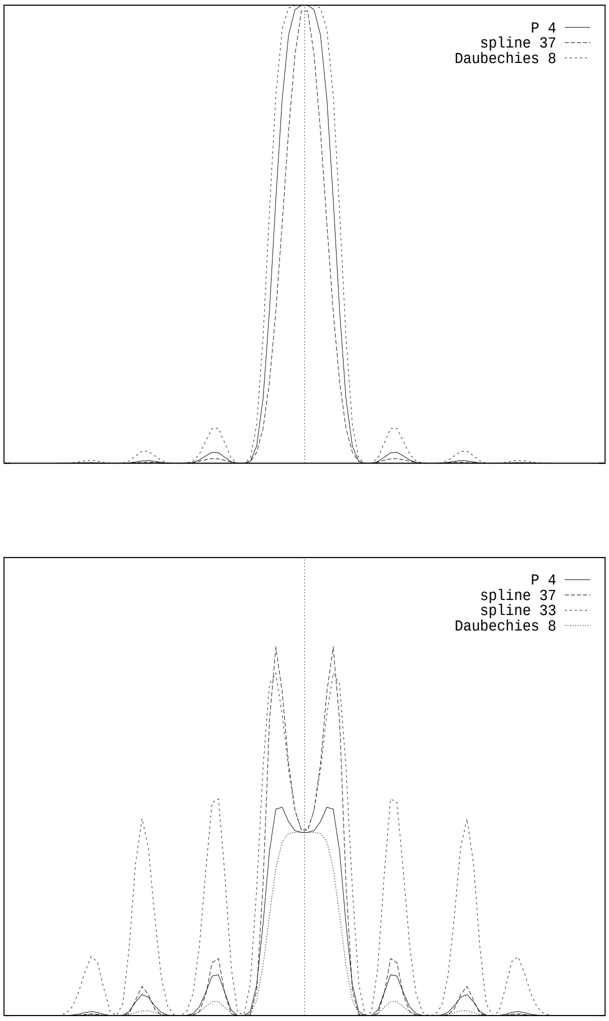


Figure 5: Fourier transform magnitude for spline-based scaling functions. Top: reconstructing (primal) scaling functions. Bottom: analyzing (dual) scaling functions.

5 Applications

In this section we develop some applications of wavelets to curve and surface manipulation. We demonstrate wavelet-based compression of parametric curves and surfaces using a new variant of the standard technique. This compression method is well suited for curves and surfaces which must be sampled or approximated adaptively for other operations. It also allows us to obtain an adaptive approximation of the curve or surface, using a smaller number of natural components. (If the underlying wavelets are spline-based, the result is similar to the hierarchical B-spline representation of [18].) We also illustrate a related application: the use of the wavelet representation to find smooth sections of surfaces. We discuss error box estimation, and compare the wavelet multiresolution approximations to a common hierarchical curve representation technique, the strip tree [2], [19]. Finally, we outline an intersection algorithm, which is applied to curves preprocessed by the above techniques.

The primary wavelet used is the pseudocoiflet P_4 with 4 vanishing moments. Pseudocoiflets are useful in these applications partly because they allow us to approximate well with scaling coefficients. Scaling coefficients are easy to calculate and can be used in adaptive piecewise linear approximations to the original data; however, their use requires the underlying wavelet to produce coefficients close to the approximation curves, a property which pseudocoiflets were designed to satisfy. The interpolation property of pseudocoiflet scaling functions allows us to take given data samples as initial coefficients without penalty. Pseudocoiflets also have good localization properties.

Using wavelets with higher vanishing moments gives better approximations – for scaling coefficients, this translates to coefficients which lie closer to the original curve.

5.1 Example

The example in Figure 6 shows the finer resolution wavelet coefficients of the y-coordinate of a curve with 1,536 points. The coefficients are arranged with coarser resolution on top, finest level at the bottom, and shown dilated, so that they correspond to the spatial locations where they affect on the curve. The curve parameter space consists of the indices given on the horizontal axis. Regions near indices 400-600 and 1100-1300 are of particular interest, since the small wavelet coefficients there indicate that these curve sections are relatively smooth. (The finer level coefficients are not zero although they appear so on the scale given.) In the next section we look at the compression of this curve for display and other operations.

5.2 Spatially coherent compression

A key advantage of using a wavelet representation is the ease of compression of curves and surfaces as preparation for other operations, such as display, interference detection, and so on. For these operations, it is important that the approximation given by the compression is quickly refinable when more accurate results are needed. We develop a new variant of the standard L^p-optimal compression method [14] to obtain fast refinability. This procedure gives a compact hierarchical representation of the surface in terms of its scaling coefficients.

A difference between the compression performed here and the standard algorithm needs to be noted.

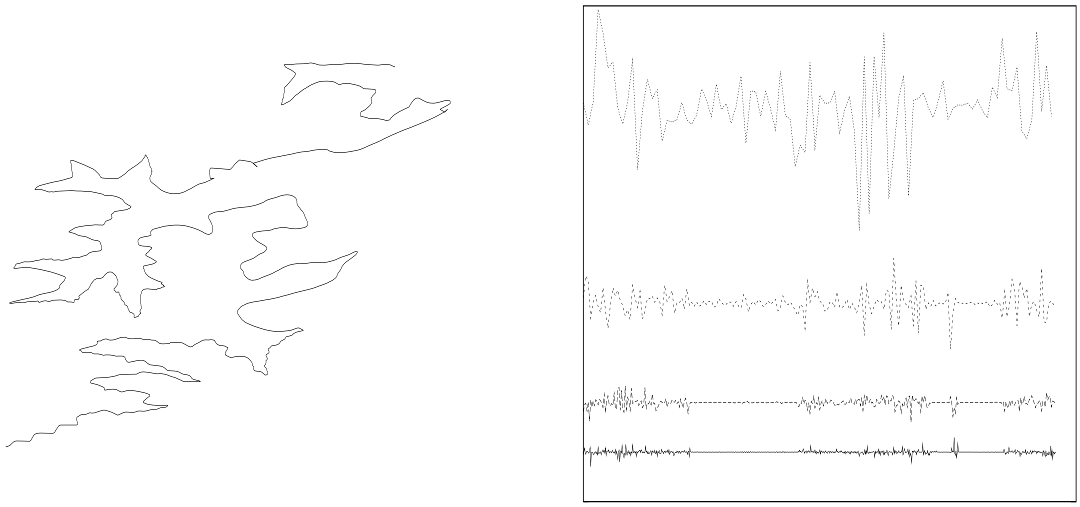


Figure 6: Original curve and wavelet coefficients

Lossy compression of curves and surfaces is usually performed in order to minimize storage – the objective is to specify an approximation of the curve using the least amount of data. A standard procedure is to set all wavelet coefficients below a given threshold value to zero. This constitutes an optimal L^2 compression method, where optimality means a minimal number of nonzero wavelet coefficients for a given error. However, in this approach, wavelet coefficients can be set to zero in areas where large coefficients are dense. Dropping these small wavelet coefficients is not useful for operations such as display, since, for accurate results, the area with the otherwise large coefficients has to be sampled at a high density, regardless. Further, when the dropped coarse level coefficients are added during refinement, they affect a wide region in the data. This is why it is useful to include these small coefficients in the compressed data even when strict thresholding would require them to be dropped.

In our variant *Compress*, wavelet coefficients below a given threshold are set to zero only if they occur in blocks with other small coefficients on the same level. The minimum block length is specified separately[2]. Curve regions where all the higher level wavelet coefficients are small can then be represented with the corresponding coarse resolution scaling coefficients. If the curve has smooth sections, the number of these scaling coefficients is much smaller than the number of curve samples, and the curve is approximated, *in this region*, by the corresponding portion of the multiresolution approximating curve. If the wavelet is well chosen, the piecewise linear scaling coefficient curve is also a suitable approximation.

The underlying data structure for this approximation is a truncated binary tree, the *segment tree*, where the segments are associated with scaling coefficients. The tree corresponds to an adaptive subdivision of the parameter space. Each segment on a given level corresponds to a unique section of the underlying curve or surface; these sections are nested across the scales.

[2]For surfaces, blocks are defined as the index sets corresponding to a square in the parameter space.

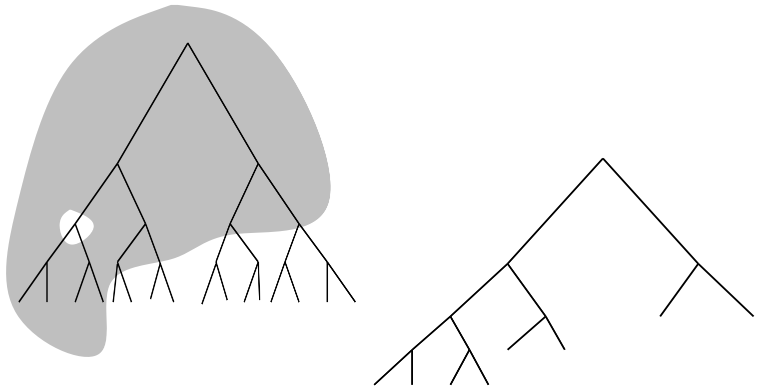


Figure 7: Wavelet coefficients with large coefficients shaded; corresponding truncated scaling coefficient tree

The leaves of the truncated scaling coefficient tree represent the underlying compressed surface. The surface can be recovered at the original sampling density by extending the tree to its full binary form (for instance, by carrying out the complete reconstruction algorithm).

The input to the following procedures is a curve or surface wavelet decomposition $\mathbf{W}$, wavelet coefficient threshold ϵ [3], and minimum blocksizes $B(L)$. The blocksize varies with the level L, since typically, we require a larger block of zero coefficients on coarser levels. Here, the coarsest level considered is indexed with $L = 0$.

Compress ($\mathbf{W}$, B, ϵ)
for $L = 0$ **to** $L =$ finest level **do**:

- Set wavelet coefficients $w_{Lj} < \epsilon$ on level L to 0, if they occur in blocks of size at least $B(L)$. **end do**

Compression results in a modified surface wavelet decomposition. This can now be selectively reconstructed. The result of the following procedure is a hierarchical representation $\mathbf{S}$ of the curve or surface using approximating segments defined by the scaling coefficients. $\mathbf{S}$ also contains a list of index regions R, each of which is marked `final` at a level L_R.

[3] We could allow the thresholds ϵ to be level-dependent. This is analogous to optimal compression in different norms ([14]).

SelectiveReconstruct($\mathbf{W}$, B, ϵ)
for $L = 0$ **to** $L =$ finest level **do**:

- Mark a connected parameter space index region `final` at level L, if it has not previously been marked `final`, and all wavelet coefficients corresponding to that region have been set to 0, at all levels l finer than L.
- If an index region has not been marked `final`, compute the scaling coefficients on level L for that region. **end do**

The compression and the hierarchical representation can be refined by the following procedure, which takes as input the previous wavelet coefficients $\mathbf{W}$ and their selective reconstruction $\mathbf{S}$ for the old threshold ϵ_1, and a new threshold, $\epsilon_2 < \epsilon_1$:

RefineCompression($\mathbf{W}$, $\mathbf{S}$, B, ϵ_2)
Compress $\mathbf{W}$ selectively:
for $L = 0$ **to** $L =$ finest level **do**:

- If wavelet coefficients w_{Lj} belong to a region marked `final` at a level L_1 finer than L, do not process them.
- Else set wavelet coefficients $w_{Lj} < \epsilon_2$ on level L to 0, if they occur in blocks of size at least $B(L)$.

Update reconstruction $\mathbf{S}$ selectively only in regions previously marked `final`:
for $L = 0$ **to** $L =$ finest level **do**:

- If a connected parameter space index region has not been marked `final` by level L for previous ϵ_1, do not process it.
- Else:
  Mark appropriate regions as `final` at level L as before, but for the new threshold.
  If an index region has not been marked `final`, compute the scaling coefficients on level L for that region, as before.
  end do

The advantage of this compression variant is the following: as the approximation is improved, there will be minimal updating of coarser resolution data – and so any recalculation that has to be done to add new wavelet coefficients will involve fewer decomposition levels, and large curve regions will remain fixed. (Note that all "dense" region coefficients have been kept on every level. As refinement increases, formerly `final` regions can become dense, and dense regions will remain so.)

In the following example we threshold the curve of Figure 6 with a threshold corresponding to 300 nonzero points in the optimal compression. The number of nonzero points is 366 in our variant, or about 24 % compression. Blocksizes vary from 1 to 5. This is an example to illustrate the procedure; note that when using small compression ratios, the large majority of curve regions will in fact be truncated at decomposition levels well below the original curve sampling density.

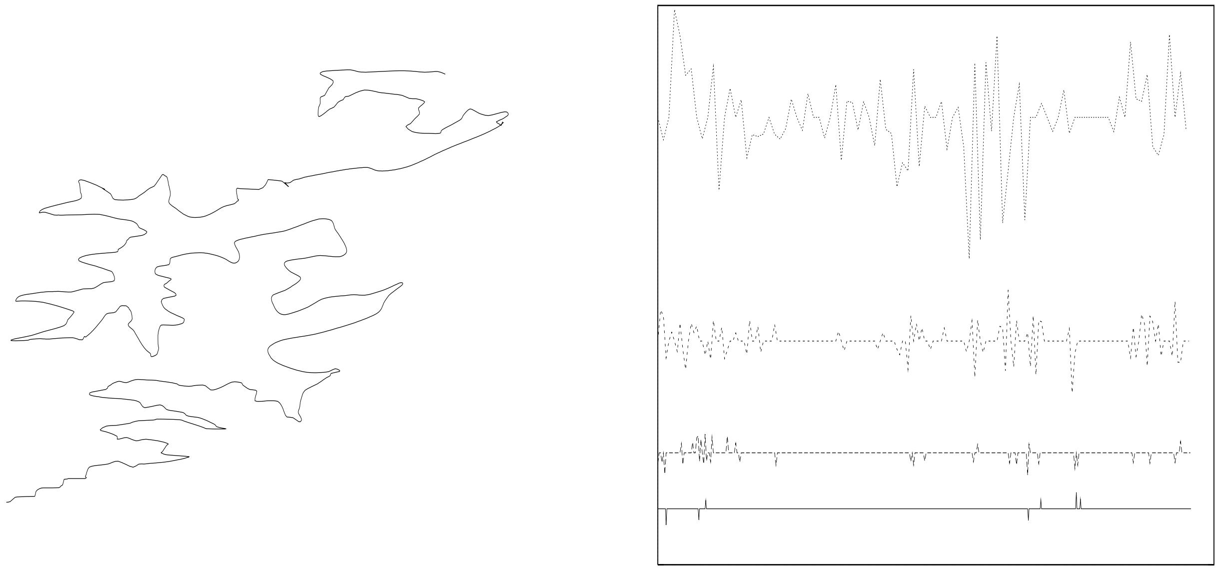


Figure 8: Compressed curve. Compressed wavelet coefficients

5.3 Adaptive approximation using scaling coefficients

The above compression method effectively sections the curve or surface into regions of different complexity. This allows the curve to be approximated by piecewise linear scaling coefficient curve segments at different levels of refinement.

Instead of calculating the true compressed surface from the scaling coefficients, the coefficients can be used by themselves in a piecewise linear approximation. This approximation consists of linear pieces – at the boundaries between regions corresponding to different levels, the end points can be equated, since the error of doing this is within the error bound for the approximation).

Figures **??**, 9 give an example of an adaptive, piecewise linear scaling coefficient approximation to a $\approx$ 5000-point curve obtained from brain scan data[4], using the pseudocoiflet P_4.

Figure **??** is an illustration of the method. For clarity, the error allowed is large, and only two levels of wavelet decomposition are used. To show the areas from different levels better, the regions have not been connected to one piecewise linear curve. Wavelet coefficients for both coordinates are used to determine the appropriate scaling coefficient blocks. Most of the curve can be adequately

[4]Data courtesy of Peter Cahoon, UBC.

approximated using the lower resolution level, but some sharper corners require the use of higher resolution scaling coefficients.

Figure 9 shows a scaling coefficient approximation is superimposed on the original curve. The number of points in the scaling coefficient approximation is 245, representing compression to less than 5 % of the original data.

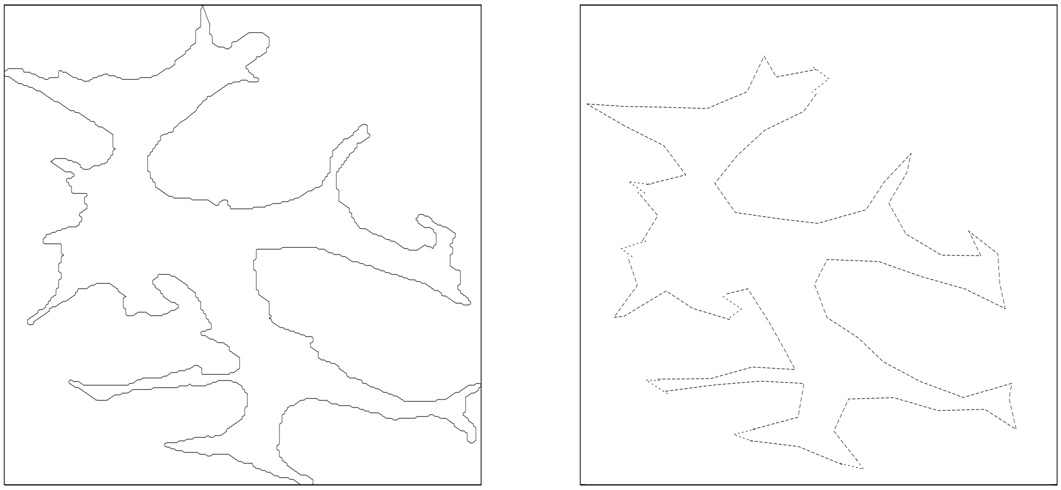


Figure 9: Original curve and a simplified, 2-level adaptive scaling coefficient approximation. Some sections of the curve are approximated with the sparser low resolution scaling coefficients, sharper corners need higher resolution coefficients.

Approximating surfaces adaptively using wavelet compression differs from approximating from the linear multiresolution spaces by the multiresolution surfaces. In the latter case the representation is always stopped on a given level and no higher resolution wavelet coefficients are included. Wavelet compression allows faster approximation of the original surface. Using the above compression variant, the recalculation needed for refining the approximation is minimized.

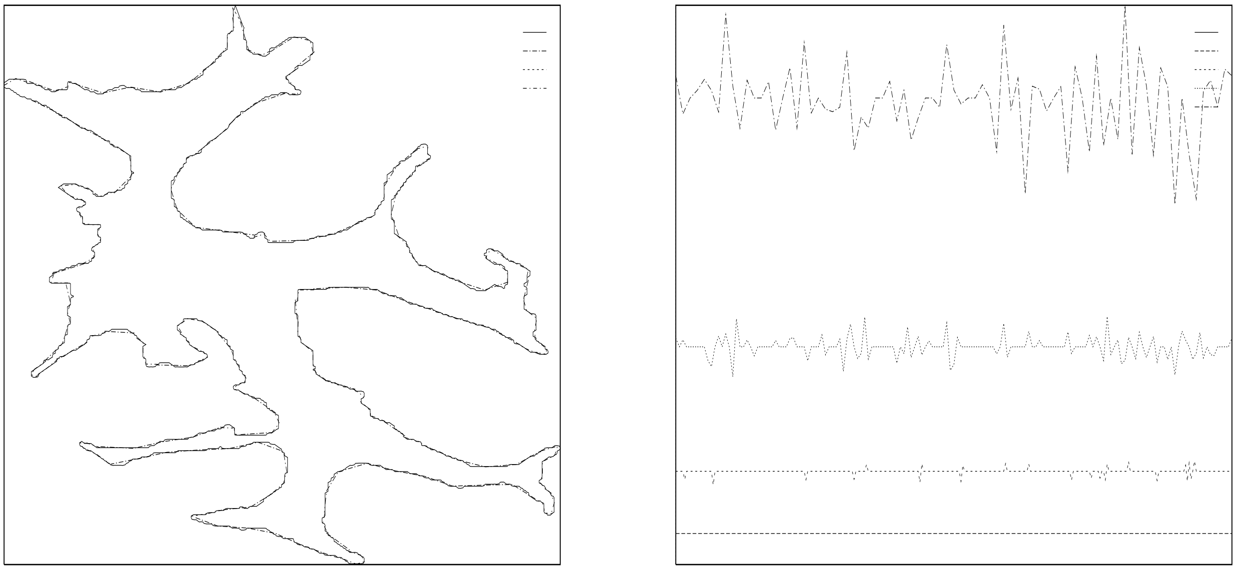


Figure 10: The original curve superimposed on the adaptive scaling coefficient representation for 5 % compression. Selected levels of the corresponding compressed wavelet coefficients.

5.4 Analyzing curves and surfaces for smooth sections

The size of the wavelet coefficients can be used to analyze the curve or surface for relative "smoothness". Smooth sections are the ones for which higher level wavelet coefficients are small, that is, the sections which are approximated well by coarser level scaling coefficients. This fact is used in [23] to hierarchically plan mobile robot paths through natural terrain.

Figure 10 shows an example of identifying the "smooth" sections of a surface. The sections are found by determining where the finer level coefficients are small. The smooth sections are shown by the marked squares on the original surface.

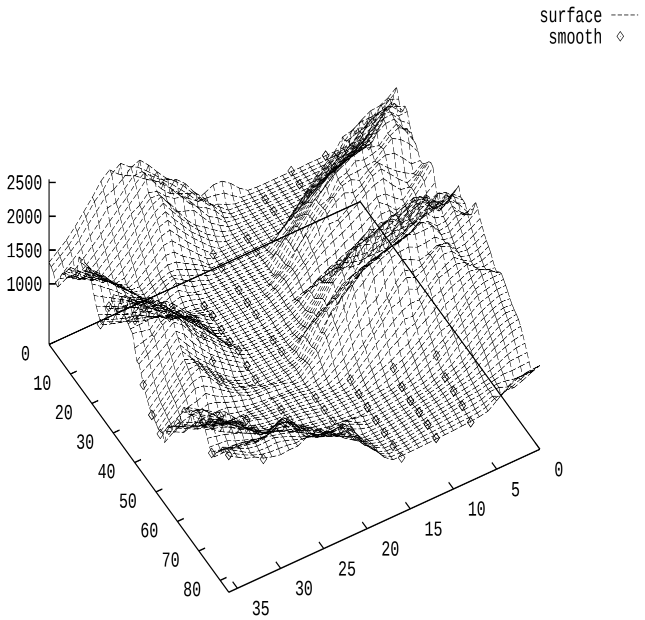


Figure 11: Identifying smooth surface sections - original surface and smoother sections on coarse level scaling coefficients. Note that in the figure the two axes are not drawn to the same scale.

5.5 Subdivision and wavelet approximation – comparison with strip trees

Subdivision methods are in general use in geometric operations such as curve and surface intersection. There are several choices for obtaining the hierarchical subdivision, for instance: basic binary subdivision [9], strip tree subdivision (subdivision points are chosen to minimize maximum distance) [2], arc length subdivision [19]. The latter methods attempt to subdivide so that the resulting piecewise linear approximations are good.

By contrast, we do not subdivide at all. Instead, we choose a discrete set of points, not necessarily on the curve, but guaranteed to approximate the curve well by a given measure. It must be possible to naturally refine the approximation by using twice as many points, just as in subdivision. We choose the approximating points as the scaling coefficients of the wavelet multiresolution approximations to the curve.

The example below compares wavelet approximations with strip tree subdivision. The wavelet approximations are the linearly interpolated coefficient curves obtained using the pseudocoiflets P_4. The scaling coefficient curves "average" the original data, rather than subdivide it, and this results in more uniform errors than the strip tree method. Figure 12 depics a refined approximation, Figure 11 a coarser one. Note also that the number of strip tree segments is 50 % larger than the number of scaling coefficient curve segments in each example.

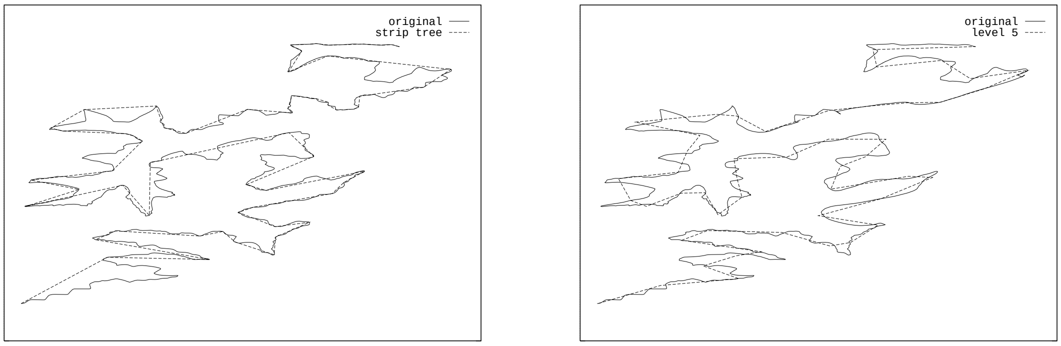


Figure 12: Strip tree subdivision (64 segments), P_4 level 5 scaling coefficients (48 segments).

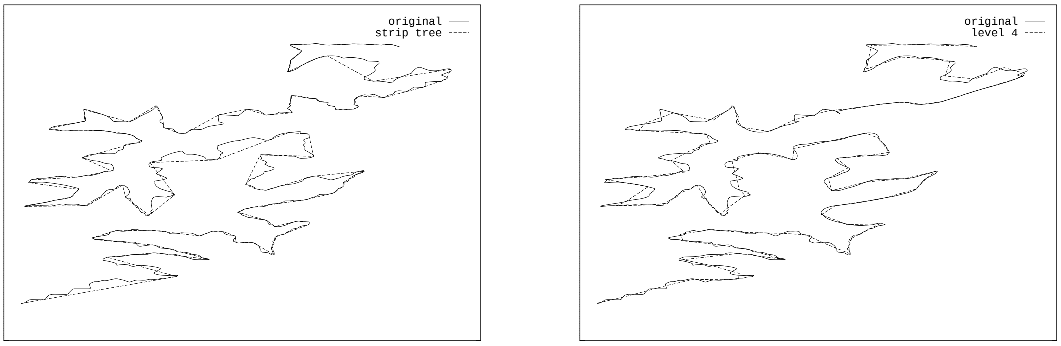


Figure 13: Strip tree subdivision (128 segments), P_4 level 4 scaling coefficients (96 segments).

5.6 Error estimation

In general wavelet multiresolution gives optimal L^2 approximation from the underlying spaces. However, L^2 optimality does not necessarily mean optimal maximum norm (L^∞) approximation, or the ideal, optimal approximation using distance. But in practice, it does provide excellent approximations. We can estimate an upper bound on the distance error from the wavelet coefficients, or we can directly precompute a smaller error box aligned with the scaling coefficients. We discuss this briefly below.

5.6.1 Error bounds from wavelet coefficients

Conservative error bounds for a replacing a given curve segment s by an approximating curve on level i can be obtained from the L^∞ errors of the coordinate functions, so we need only look at the approximation errors for a function f of one variable. An upper bound for the L^∞ error of approximating a function f by its multiresolution approximation is obtained very easily from the wavelet coefficients:

Suppose that the component function is f and its wavelet coefficients corresponding to the region s on level i are denoted by $w_i(s) = (w_{ij} : j \in A(s,i))$. These are the coefficients entering into calculations about s. This set is restricted to a set of indices $A(s,i)$, which is determined by the filter length used.

The coefficients of the error in terms of the next level scaling functions, are $w_i^*(s) = \tilde{G}w_i(s)$, where $\tilde{G}$ is the reconstructing wavelet filter. The new coefficients are similarly restricted to a subset of all the coefficients w^*. Let $f_i(s)$ denote the approximation f_i on the segment s, and let $\tilde{\phi}$ be the reconstructing scaling function. Then the error between a segment and its refinement by one level is given by

$$\begin{aligned}
\mathrm{error}_i(s) &= \| f_{i-1}(s) - f_i(s) \|_\infty \\
&= \max_j | \sum_{j \in A(s,i)} w_{ij}\tilde{\phi}_{ij} | \\
&\le (\max_j |w^*_{ij}(s)|) \sum_j |\tilde{\phi}_{ij}|.
\end{aligned}$$

The quantity $U(\tilde{\phi}) = \sum_j |\tilde{\phi}_{ij}|$ can be estimated or calculated for general scaling functions and we have

$$\mathrm{error}_i(s) \le U(\tilde{\phi}) \ \max_j |(w^*_{ij}(s))|.$$

The total error at level i is now bounded simply as

$$\mathrm{TotError}_i(s) = \sum_{i' \le i} \mathrm{error}_{i'}(s).$$

For positive scaling functions, such as B-splines, $\sum_j |\tilde{\phi}_{ij}| = 1$, by the partition of unity property, and $\mathrm{error}_i \le \max_j |(w^*_{ij}(s))|$, that is, the error is obtained by adding the maximum reconstructed wavelet coefficient norms on each level.

For pseudocoiflets, the maximum real errors on each level are almost the same as the maximum reconstructed coefficient norms: as can be seen from an example in Figure 13, the wavelet coefficients, with one step of the reconstruction algorithm performed, give a good approximation of the real error.

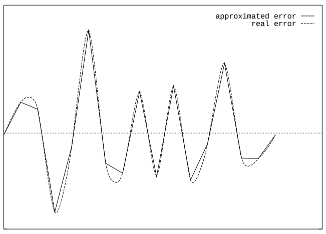


Figure 14: Real error compared to wavelet coefficients with 1 step reconstruction

The maximum reconstructed coefficient norm $a = \max_j \left| w^*_{ij}(s) \right|$ can also be estimated from the wavelet coefficient maximum $b = \max_j |w_{ij}(s)|$ directly: in the worst case, $a = \sqrt{2}\ b$. This worst case is usually not attained. This procedure gives reasonable (but not minimal) error bounds, especially for smoother sections of curves.

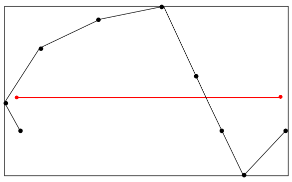
5.6.2 Linearization error

The previous error estimate was valid for approximation curves. For the piecewise linear scaling coefficient curves, the effect of linearizing the approximation curve has to be estimated as well. This linearization error is usually not as large as the error from the wavelet coefficients.

The linearization error can also be computed from the wavelet transform by looking at the difference functions between the real approximation curve and the piecewise linear scaling coefficient curve. The scaling coefficient curve can be formally obtained by applying the hat function as a reconstructing function to the scaling coefficients.

So, the difference functions are obtained by looking at the difference "basis" function $\tilde{\phi} - \phi_{\text{hat}}$, where $\tilde{\phi}$ is the reconstructing scaling function, and ϕ_{hat} the hat function. Estimating the max norm of this "basis" function, and applying this to the scaling coefficients, gives a bound for the linearization error.

In cases where the above wavelet coefficient estimates do not give sufficiently tight bounds, minimal error regions for replacing a curve section with scaling coefficients can be computed as follows. For two consecutive scaling coefficients on level L, find the 2^L points on the original curve, corresponding to these scaling coefficients, and compute a minimal box aligned with the line segment between the scaling coefficients (the line in the figure), and containing the points ("X"):



5.7 Curve-curve intersection

In our final example, we use the wavelet representation in a subdivision-like curve-curve intersection algorithm. This algorithm is a natural consequence of the hierarchical wavelet representation. The quality of the approximation will determine the speed of a subdivision-like algorithm – since wavelets are well known to have good approximation properties, they are a reasonable choice for a hierarchical representation.

The algorithm does not rely on a subdivision of the curve, instead it approximates curves with the refinable piecewise linear ones determined by the scaling coefficients.

The input to the algorithm is a segment tree, that is, a selectively reconstructed, compressed curve obtained by the procedures of Section 5.2. The compression is performed to half the intersection tolerance. Error boxes can be precomputed. The algorithm proceeds level by level exactly as a subdivision algorithm, and identifies potentially intersecting segment pairs. Segment pairs found not to intersect are discarded. The refinement of a segment is done by replacing each scaling coefficient by two scaling coefficients on a finer level (not by subdividing) in the truncated scaling coefficient tree of Figure 7.

The main part of intersection algorithm for a given level i is described below:

```
levelIntersection(i)
begin
for     all potentially intersecting segment pairs (l_1, l_2) on level i
    if   (not lastLevel(i, l_1, l_2))
        if   (errorRegionsIntersect(Error(l_1), Error(l_2))
             /* refine both segments ... */
             l_{C_1}(0) = RefinedSegm(C_1, l_1, 0)
             l_{C_1}(1) = RefinedSegm(C_1, l_1, 1)
             l_{C_2}(0) = RefinedSegm(C_2, l_2, 0)
             l_{C_2}(1) = RefinedSegm(C_2, l_2, 1)
                  /* ...add the new segment pairs to the list for the refined level: */
             for     (j = 0, 1 and k = 0, 1)  addToPotentialIntersections(l_{C_1}(j), l_{C_2}(k))
        /* if error regions do not intersect, discard the pair: */
        else    break
    else   if    (lastLevel(i, l_1, l_2)) intersectSegments(l_1, l_2)
end
```

This is exactly like a subdivision algorithm, except for the refinement of a segment, which is not done using subdivision. This binary refinement process is continued until a final refinement level in the segment tree hierarchy has been reached (the function *lastlevel* $(i,\ l_1,\ l_2)$ returns true). When this occurs, the potentially intersecting curve segments are used to find the final intersection points.

This can be done in two ways. The first is by continuing with binary refinement until the original point density is reached, as in subdivision. This is guaranteed to succeed, but in certain cases may take longer than necessary (for instance, when intersecting two nearly tangential curves).

The second intersection method relies on the use of the properties of the scaling coefficients defining the whole contiguous curve section. As this is the final refinement level, the scaling coefficients describe the underlying curve in this whole region. The intersection of these longer sections can be performed by another method, when this is more appropriate. As a simple example, the intersection of two long, nearly linear sections can be found as a simple line-line intersection. Other fast methods for nonoscillating, long, smooth sections include Newton method variants or secant methods adapted to parametric curves.

The error boxes for the algorithm are computed so that the underlying curve segment corresponding to the scaling coefficients on a given level of the segment tree is always contained in the error box. (See section 5.6 for the computation of the error.)

The algorithm can be applied to the case of discrete plane curves defined by n line segments. The worst case performance of this, and other subdivision-type algorithms, is n^2. However, the algorithm performs well if the number of intersections K is small relative to the length n of the curves, a common occurrence in practice; in these situations the binary refinement approach will often be much faster than the commonly used output-sensitive $n \log n + K$ algorithms (e.g. [5]). Since wavelet-based approximation produces small error boxes, the algorithm is fast.

6 Conclusions

The main contributions of this paper are the construction of a new family of wavelets, pseudocoiflets, well suited for curve and surface representation, and new applications to the hierarchical, adaptive representation of curves and surfaces using a parametric wavelet decomposition.

The construction of pseudocoiflets was motivated by curve representation requirements, but these wavelets are also of general interest. The interpolation property of pseudocoiflets is important in many applications, as it allows the direct use of surface samples as control points, for instance. In addition, pseudocoiflets have good space-frequency localization properties for both wavelets, which other biorthogonal wavelets often lack. Pseudocoiflets are now in use in wavelet libraries ([20], [4]).

We illustrate some of the advantages of using a wavelet representation of curves and surfaces. We give examples of adaptive approximation of curves using scaling coefficients, an example of the analysis of surfaces for smoothness, and an intersection algorithm. The wavelet-based intersection algorithm is a form of the basic subdivision algorithm, but it is more flexible and does not rely on curve subdivision, but rather, on hierarhical curve approximation. All of these examples rely on a new variant of wavelet compression, and use the pseudocoiflets developed here.

Acknowledgements. I would like to thank Dr. Dinesh Pai and the anonymous reviewers for their many valuable comments and suggestions. I also wish to thank Drs. Alan Mackworth, Jack Snoeyink and Robert Woodham for providing the data used in the applications.

References

[1] M. Antonini, M. Barlaud, and P. Mathieu, Image coding using lattice vector quantization of

wavelet coefficients, *Proc. IEEE Int. Conf. ASSP, 1991*, pp. 2273–2276.

[2] D. Ballard, Strip trees: a hierarchical representation for curves, *ACM Communications*, 24, pp. 310–321, 1981.

[3] G. Beylkin, R. Coifman, and V. Rokhlin, Fast Wavelet Transforms and Numerical Algorithms I, *Yale University preprint*, 1990.

[4] M. Bourges-Sévenier, Réalisation d'une bibliothéque C de fonctions ondelettes, IRISA, 1994.

[5] B. Chazelle and H. Edelsbrunner, An optimal algorithm for intersecting line segments in the plane, *Proc. 29th IEEE Symposium on Foundations of Computer Science, 1988*, pp. 590–600.

[6] C.K. Chui, *An Introduction to Wavelets*, Academic Press, 1992.

[7] A. Cohen, I. Daubechies, and P. Vial, Wavelets and fast wavelet transform on the interval, *AT&T Bell Laboratories preprint*, 1992.

[8] A. Cohen, I. Daubechies, and J.C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure and Appl. Math* **45**, pp. 485–560, 1992.

[9] E. Cohen, T. Lyche, and R. Riesenfeld, Discrete B–splines and subdivision techniques in computer–aided geometric design and computer graphics. *Computer Graphics and Image Processing* **14**, pp. 87–111, 1980.

[10] J.M. Combes, A. Grossman, and Ph. Tchamitchian, Eds., *Wavelets – Time Frequency Methods and Phase Space*, Proceedings of the Int. Conf., Marseille, 1987, Springer-Verlag, Berlin.

[11] I. Daubechies, Orthonormal bases of compactly supported wavelets II. Variations on a theme, *AT&T Bell Laboratories preprint*, 1990.

[12] I. Daubechies, *Ten Lectures on Wavelets*, *CBMS–NSF Regional Conference Series in Applied Mathematics* **61**, SIAM, 1992.

[13] G. Deslauriers and S. Dubuc *Symmetric Iterative Interpolation Processes*, *CONAP* **5**, 1, pp 49–68, 1989.

[14] R.A. DeVore and B.J. Lucier, Wavelets, *Acta Numerica*, pp. 1–56, 1991.

[15] D.L. Donoho and I.M. Johnstone, Ideal Spatial Adaptation by Wavelet Shrinkage, *Stanford University preprint*, 1992.

[16] D. L. Donoho, Interpolating wavelet transforms, *Stanford University preprint*, 1992.

[17] N. Dyn, A. Gregory, and D. Levin, A 4-point interpolatory subdivision scheme for curve design, *CAGD* **4**, pp 257–268, 1987.

[18] D. Forsey and R.H. Bartels, Hierarchical B–spline refinement, *Computer Graphics* **22**, pp. 205–212.

[19] O. Gunther and S. Dominguez, Hierarchical schemes for curve representation, *IEEE Computer Graphics and Applications* 13(3), May 1993, 55–63.

[20] R. Lewis, C wavelet library, Department of Computer Science, University of British Columbia, `http://www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html`, 1994.

[21] A.K. Mackworth and F. Mokhtarian, The renormalized curvature scale space and the evolution properties of planar curves, *Proc. IEEE CVPR*, pp. 318–326, 1988.

[22] S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. PAMI* **11**, pp. 674–693, 1989.

[23] D.K. Pai and L.-M. Reissell, Multiresolution rough terrain motion planning, Department of Computer Science, UBC, Technical Report TR 94–33, 1994.

[24] L.-M. Reissell, Wavelet Representation and Geometric Algorithms for Multiscale Curves and Surfaces in Image Processing, *Proceedings of the International Workshop on Image Analysis and Synthesis, Graz*, June 2–4, 1993.

[25] G. Strang and G. Fix, A Fourier analysis of the finite element variational method, CIME II Ciclo 1971, Constructive Aspects of Functional Analysis (Geymonat, ed.), pp. 793–840, 1973.

7 Appendix: Proofs

Proof of Proposition 3.1.

For coefficient curves, these properties follow from the observation that, for a function f in $L^2(R)$, the operation P giving the normalized scaling coefficients of f on level i, d^*_{ij}, is linear. Slightly more specifically, we show that

$$P(\alpha f + \beta) = \alpha P(f) + \beta. \tag{11}$$

(Restrict all functions to compact sets when necessary.) This follows immediately for instance from the inner product definition of scaling coefficients and their normalization:

$$P(f)(j) = d^*_{ij} = 2^{-i/2} < f, \phi_{ij} >,$$

where ϕ is the analyzing scaling function. This implies

$$P(\alpha f + \beta)(j) = 2^{-i/2} < \alpha f + \beta, \phi_{ij} > = \alpha d^*_{ij} + \beta 2^{-i/2} \int \phi_{ij} = \alpha d^*_{ij} + \beta.$$

The analogous properties for the approximating curves also follow directly: a point $p(t)$ on the approximating curve Approx(C, i) has the coordinate component $f_i(t) = \sum_j d(f)_{ij}\phi_{ij}(t)$, where the $d(f)_{ij}$ are the (unnormalized) scaling coefficients for the appropriate coordinate function f, and ϕ is the reconstructing scaling function. Again, we have the analogue of (11): this follows from the fact that $2^{i/2} \sum_j \phi_{ij}(t) = \sum_j \phi(2^{-i}t - j) = 1$ and

$$\sum_j d(\alpha f+\beta)_{ij}\phi_{ij}(t) = \sum_j (\alpha d(f)_{ij} + 2^{i/2}\beta)\phi_{ij}(t) = \alpha \sum_j d(f)_{ij}\phi_{ij}(t) + 2^{i/2}\beta \sum_j \phi_{ij}(t).$$

Since translation, rotation and scaling are linear on the coordinate functions, the approximation curve respects these operations.

□

Proof of Proposition 4.1. We note first that interpolating scaling functions with an even number of vanishing moments and an odd number of filter elements are a special case of coiflet-like scaling functions: that is, if they satisfy the first moment condition (5) for $2N$ they also satisfy the second moment condition (4) for $2N$.

An equivalent condition for interpolation is requiring that the filter transfer function

$$m_0(\xi) = \frac{1}{\sqrt{2}} \sum_j h_j e^{-ij\xi}$$

has the property

$$m_0(\xi) + m_0(\xi + \pi) = 1. \tag{12}$$

The interpolation condition (12) for a filter transfer function satisfying the moment condition (5) can now be written as:

$$(1-x)^N P_1(-x) + (1+x)^N P_1(x) = 1, \tag{13}$$

where $x = \cos\xi$.

On the other hand, the solutions P_1 and P_2 to the coiflet equation for $2N$ vanishing moments, (6), satisfy

$$P_2(x) = -P_1(-x). \tag{14}$$

This can be seen by substituting $-x$ for x in the equation, and using the uniqueness of the minimum degree solutions. The solutions (7) and (8) are obtained in this way. We then have the following equation for such P_1:

$$(1+x)^N P_1(x) + (1-x)^N P_1(-x) = 1. \tag{15}$$

This is a form of the "coiflet equation" and is exactly the interpolation condition (13) obtained above. This means that the solutions to the coiflet equation obtained in the above way from (7) are automatically interpolating.

For the converse, we note that symmetric interpolating $\tilde{m}_0$ can only take the form

$$m_0(\xi) = (1+\cos\xi)^{N_1} P_1(\cos\xi), \tag{16}$$

where $2N_1$ is the number of vanishing moments for the scaling function.

This follows immediately from the observation that the filter corresponding to a symmetric interpolating $\tilde{m}_0$ must consist of an odd number of filter elements by the interpolation condition (12). Thus, $\tilde{m}_0(\xi) = |\tilde{m}_0(\xi)|$, the number of vanishing moments is even, and the function can be expressed in the form (16). The fact that $\tilde{m}_0$ is coiflet-like then follows from the equivalence of the interpolation condition (13) and the coiflet-condition (15) as before. □

Proof of Proposition 4.2.

If the polynomials $\tilde{P}(x)$ and $\tilde{P}(-x)$ have no common zeros, the biorthogonality condition (10) has a unique minimum degree solution $(P(x), P(-x))$, with P of degree $N + \tilde{N} + \text{degree}(\tilde{P}) - 1$ (as well as other solutions of higher degrees). This follows from a theorem by Bezout (see e.g. [12], Chapter 6). Therefore, to find the analyzing polynomial P corresponding to the reconstructing polynomial $\tilde{P}$ constructed above, it only remains to show that if $\tilde{P}(x)$ is a solution (7) to the coiflet equation (6), then $\tilde{P}(x)$ and $\tilde{P}(-x)$ have no common zeros:

Lemma 7.1 *If $P_1(x)$ is a solution of the coiflet equation (15), then $P_1(x)$ and $P_1(-x)$ have no common zeros.*

Proof. Since $R(x)$ is a solution to the coiflet equation (15), $R(x) - 1/2$ is an odd polynomial, and so divisible by x. Therefore, $R(x)$ does not have a zero at $x = 0$.

Suppose $R(x)$ and $R(-x)$ have a common zero at $x = a$, $a \neq 0$. Then

$$R(x) = (x^2 - a^2) Q(x)$$

for some polynomial Q, and the coiflet equation becomes

$$(x^2 - a^2)[(1+x)^N Q(x) + (1-x)^N Q(-x)] = 1.$$

But it is easy to see (for instance by a direct power series solution method) that there are no polynomial solutions to this equation. So the polynomials $R(x)$ and $R(-x)$ have no common zeros.
□

The existence of a unique minimum degree solution P follows from Bezout's theorem and the previous lemma. Since $\tilde{N} = N$ and $\text{degree}(\tilde{P}) = N - 1$ this degree is

$$\text{degree}(P) = N + \tilde{N} + \text{degree}(\tilde{P}) - 1 = N + N + (N-1) - 1 = 3N - 2.$$

□