

A Multiresolution Framework for Variational Subdivision

Leif Kobbelt*
University of Erlangen–Nürnberg

Peter Schröder†
Caltech

Abstract

Subdivision is a powerful paradigm for the generation of curves and surfaces. It is easy to implement, computationally efficient, and useful in a variety of applications because of its intimate connection with multiresolution analysis. An important task in computer graphics and geometric modeling is the construction of curves that interpolate a given set of points and minimize a fairness functional (variational design). In the context of subdivision, fairing leads to special schemes requiring the solution of a banded linear system at every subdivision step. We present several examples of such schemes including one that reproduces non-uniform interpolating cubic splines. Expressing the construction in terms of certain elementary operations we are able to embed variational subdivision in the lifting framework, a powerful technique to construct wavelet filter banks given a subdivision scheme. This allows us to extend the traditional lifting scheme for FIR filters to a certain class of IIR filters. Consequently we show how to build variationally optimal curves *and* associated, stable wavelets in a straightforward fashion. The algorithms to perform the corresponding decomposition and reconstruction transformations are easy to implement and efficient enough for interactive applications.

Introduction

A basic component of many algorithms in shape design is the efficient construction of interpolating curves, given some set of data points. Many techniques have been developed for this purpose [18, 13], the most widely used ones being based on piecewise polynomials. However, to obtain high quality curves differentiability is usually not sufficient but optimality with respect to some fairness measure is required as well [3, 26].

Applications such as level of detail rendering [25], data compression, progressive transmission, hierarchical editing [14], and adaptive numerical solvers [16, 24] require widely varying levels of detail. This makes it desirable to have multiresolution representations and associated wavelets for parametric curves. In the particular case of splines, this can be realized through iterative knot insertion and the use of bi-orthogonal spline wavelets, for example. However, we are interested in more general subdivision schemes which do not necessarily lead to piecewise polynomial curves. Specifically, we consider subdivision schemes which minimize a fairness functional under point interpolation constraints [19].

In this paper we describe a framework which unifies a number of earlier, independent approaches. The resulting algorithms are based entirely on combining a small set of primitive operations. These can be expressed conveniently in *Wiring Diagrams*, which posses immediate and straightforward implementations leading to very efficient computations. Specifically we construct interpolating curves with minimal second and third divided difference energy using a simple subdivision scheme. This extends earlier work [19] from the uniform to the non-uniform parameter setting. In special cases of our construction, interpolating cubic splines and Butterworth half-band filters are generated. Since the traditional Fourier approach does not apply in the non-uniform parameter setting we require a general banded system solver. A particularly suitable choice for this purpose is cyclic reduction. A novel aspect of our approach is the interpretation of cyclic reduction as the repeated splitting of the high-band in a lifting factorization of the subdivision algorithm. This observation effectively extends the classical Lifting Scheme from the FIR filter setting to a class of filters whose inverse is FIR. We exploit this connection to derive and implement a novel set of stable, bi-orthogonal wavelet bases for variationally optimal subdivision schemes.

*Leif.Kobbelt@informatik.uni-erlangen.de

†ps@cs.caltech.edu

Our emphasis in this paper is on the framework and fundamental algorithms for variationally optimal, non-uniform, subdivision curve construction in a multiresolution setting. The importance of ideas such as variational modeling [26, 33, 21], non-uniform parameterizations [13], hierarchical curves [14], and wavelets [16] in computer graphics applications, has been amply demonstrated by others. So far the previous work has considered these aspects individually. Our goal in this paper is to exhibit the simplicity and uniformity of the underlying structures in all these previous approaches. We argue that they all fit neatly into a common framework in which they can be exploited synergistically.

The paper is structured in 2 parts. In the first part we begin by fixing our notation and review interpolatory subdivision schemes both in the uniform and non-uniform setting. Section 1.1 reviews the variational subdivision setting introduced earlier in [19] and gives a set of examples. We then *extend* these schemes to the non-uniform setting and conclude the first part with some examples comparing different variational curves with and without uniform parameterizations.

The goal for the second part of the paper is to build a multiresolution framework for variational subdivision, i.e., we introduce detail spaces between subdivision levels. We proceed in a number of stages. First we define a convenient formalism, *Wiring Diagrams*, to speak about the Lifting Scheme [28] and manipulate lifting steps. Next we discuss *Cyclic Reduction* as the most appropriate technique for solving the linear systems which appear in variational subdivision. We show how CR can be expressed as a sequence of subband splits and merges applied to the high-band of a high-pass/low-pass wavelet filter. Using this observation we are able to use the Lifting Scheme, a well known approach to the construction of finitely supported bi-orthogonal wavelet transforms, in the globally supported setting demanded by variational subdivision. Thus we can apply additional lifting steps to stabilize, or improve the condition, of the variational wavelet bases. The result is a multiresolution setting for variational subdivision schemes in the non-uniform parameterization setting. Because everything is expressed in terms of the small set of primitives needed in the Lifting Scheme simple and efficient implementation is straightforward. We conclude with non-uniform Butterworth filtering as an application example and an outlook towards future research.

Part 1: Interpolatory Subdivision Schemes

In this section we review some basic examples of interpolatory subdivision schemes and fix our notation before moving on to the setting of interpolatory *variational* subdivision.

Definition: An *interpolatory subdivision scheme* is formally described by an operator Φ which maps a given sequence of points (a polygon) $P_m = [\mathbf{p}_i^m]$ to a refined sequence $P_{m+1} = [\mathbf{p}_i^{m+1}]$ with $\mathbf{p}_{2i}^{m+1} = \mathbf{p}_i^m$, i.e., the points \mathbf{p}_i^m at level m are the even numbered points at level $m + 1$, while the odd numbered points, \mathbf{p}_{2i+1}^{m+1} , are newly inserted inbetween old points (see Figure 1). Iterating Φ , the vertices become more and more dense. In the limit, a curve P_∞ is generated which interpolates the initially given control vertices $P_0 = [\mathbf{p}_i^0]$.

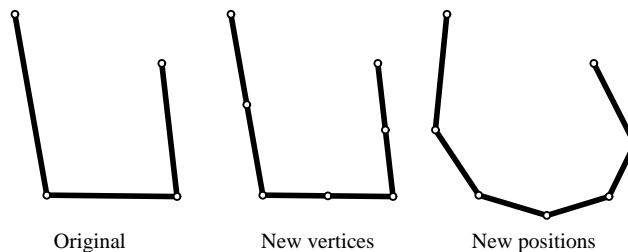


Figure 1: Interpolatory refinement: Inserting new vertices and smoothing the original shape.

Example The simplest interpolatory scheme is *piecewise linear subdivision*. In this case the new points are the midpoints of the old polygon’s edges. This corresponds to the construction of a linear interpolating polynomial

between two successive vertices \mathbf{p}_i^m and \mathbf{p}_{i+1}^m , i.e., the edge itself, and sampling this interpolant halfway between the old vertices. The resulting subdivision rule is

$$\mathbf{p}_{2i+1}^{m+1} := \frac{1}{2}(\mathbf{p}_i^m + \mathbf{p}_{i+1}^m) = \frac{1}{2}(\mathbf{p}_{2i}^{m+1} + \mathbf{p}_{2i+2}^{m+1}).$$

The middle of Figure 1 shows one step of this subdivision scheme. Repeating linear subdivision ad infinitum simply reproduces the original polygon.

Piecewise linear subdivision is merely a split operation. In general, we want to place the new vertices such that P_{m+1} is smoother than P_m (right of Figure 1). One way to do this is to increase the order of the polynomial interpolant in the above example. Consider cubic interpolation which is defined by four interpolatory constraints. To compute the new vertex \mathbf{p}_{2i+1}^{m+1} we construct the cubic polynomial interpolating the points $\mathbf{p}_{i-1}^m, \dots, \mathbf{p}_{i+2}^m$ and sample it halfway between \mathbf{p}_i^m and \mathbf{p}_{i+1}^m . Assuming a uniform parameterization, the resulting subdivision rule is

$$\mathbf{p}_{2i+1}^{m+1} := \frac{1}{16}(-\mathbf{p}_{i-1}^m + 9\mathbf{p}_i^m + 9\mathbf{p}_{i+1}^m - \mathbf{p}_{i+2}^m). \quad (1)$$

The coefficients $\frac{1}{16}(-1, 9, 9, -1)$ are well known as the 4-point subdivision mask for the generation of smooth interpolating curves [9, 11] (see Figure 2 for several steps in this procedure). The boundaries can be handled by a simple extrapolation procedure. Initially the left boundary, for example, is extended with two extra points \mathbf{p}_{-1}^0 and \mathbf{p}_{-2}^0 based on the cubic polynomial which interpolates $\mathbf{p}_0^0, \dots, \mathbf{p}_3^0$. Then, when going from P_m to P_{m+1} the vertex \mathbf{p}_{-1}^{m+1} can be computed by (1), $\mathbf{p}_{-2}^{m+1} = \mathbf{p}_{-1}^m$, and \mathbf{p}_{-2}^m is simply chopped off the sequence.

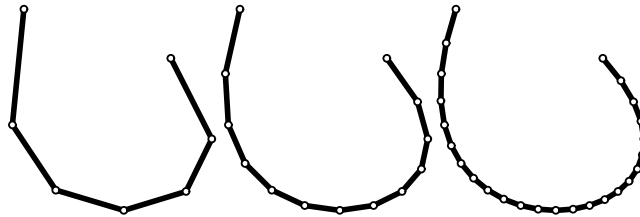


Figure 2: Refinement through cubic interpolatory subdivision, the so-called “4-point rule.”

General Setting The above schemes are instances of a general construction introduced by Deslauriers and Dubuc [8]. The $2n + 2$ vertices $\mathbf{p}_{i-n}^m, \dots, \mathbf{p}_{i+n+1}^m$ define a polynomial of degree $2n + 1$ and the new vertex \mathbf{p}_{2i+1}^{m+1} is found by evaluating this polynomial halfway between \mathbf{p}_i^m and \mathbf{p}_{i+1}^m . These schemes are *stationary* since the same refinement rule is used at every level (the rules do not depend on m) and they are *local* since the influence of one initial vertex \mathbf{p}_i^0 on the shape of P_∞ is restricted to some local neighborhood. More formally, these schemes are defined by using a finite sequence of weight coefficients $[\alpha_j]_{j=-r}^{r+1}$ with $\sum_j \alpha_j = 1$ (affine invariance) to compute new vertices as

$$\mathbf{p}_{2i+1}^{m+1} := \sum_{j=-r}^{r+1} \alpha_j \mathbf{p}_{i+j}^m, \quad (2)$$

Note that such schemes are also *uniform* since the α_j do not depend on i . Neglecting the special cases at the boundaries, both schemes above are uniform. Uniform stationary schemes are thoroughly investigated in [2, 10].

Non-Uniform Subdivision While the curves resulting from uniform stationary schemes are smooth, they are not as fair as one might wish. The left of Figure 3 shows the curve resulting from applying the 4-point scheme (1) to a set of points in the plane. It suffers from severe overshoot. Remember that this scheme is derived from local *uniform* interpolation. It is well known that the extra degrees of freedom afforded by allowing non-uniform parameterizations can be used to improve the shape of interpolating curves. Schemes which are derived from a local polynomial

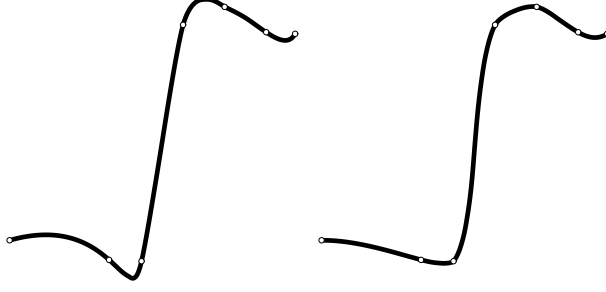


Figure 3: Interpolating curves for the same initial data (hollow dots) using the uniform (left) and a non-uniform (right) 4-point scheme.

interpolant can easily be generalized to non-uniform parameterizations. Simply associate a parameter value t_i^m with every point \mathbf{p}_i^m and solve the interpolation problem.

The weights of the *non-uniform* 4-point scheme can be found by evaluating the cubic Lagrange polynomials L_{xxxx} which are defined over the knot sequence $[t_{i-1}^m, \dots, t_{i+2}^m]$

$$\begin{aligned} t_{2i+1}^{m+1} &:= \frac{1}{2}(t_i^m + t_{i+1}^m) \\ \mathbf{p}_{2i+1}^{m+1} &:= \mathbf{p}_{i-1}^m L_{1000}(t_{2i+1}^{m+1}) + \mathbf{p}_i^m L_{0100}(t_{2i+1}^{m+1}) \\ &\quad + \mathbf{p}_{i+1}^m L_{0010}(t_{2i+1}^{m+1}) + \mathbf{p}_{i+2}^m L_{0001}(t_{2i+1}^{m+1}). \end{aligned} \quad (3)$$

Here $L_{u_0 u_1 u_2 u_3}$ is the unique cubic polynomial which takes on the value u_j at t_{i+j-1}^m for $j = 0, \dots, 3$. As in the uniform case, the new vertex \mathbf{p}_{2i+1}^{m+1} depends on four neighboring old vertices. In the special case $t_i^m = i2^{-m}$ the weights $\frac{1}{16}(-1, 9, 9, -1)$ are reproduced.

A curve generated by (3) is shown on the right of Figure 3 where we used the *centripetal* parameterization [23], i.e., $t_{i+1}^0 - t_i^0 := \sqrt{\|\mathbf{p}_{i+1}^0 - \mathbf{p}_i^0\|}$. This technique remedies the overshoot problem. The resulting curve however, while being analytically smooth (C^1) [31], still lacks fairness. To generate curves which are not only differentiable up to some order but also fair, we now turn our attention to variational subdivision schemes.

1.1 Variational Subdivision Schemes

In many applications continuity of some order is not sufficient to obtain high quality curves. Additionally one also wants the curve to be “pleasing.” This is often expressed in terms minimizing some physical measure such as strain energy. In the context of subdivision schemes this approach was considered by Kobbelt [19], who proposed the new class of *variational subdivision schemes*. The goal of these schemes is to place the new vertices \mathbf{p}_{2i+1}^{m+1} such that a fairness functional is minimized. As a consequence, the new vertices are not given by explicit rules of the type described in Equation (2), but instead as the solution of a linear system.

Definition: [19] Variational approaches start with a quadratic fairness (energy) functional

$$E(P_{m+1}) := \sum_i K(\mathbf{p}_i^{m+1})^2$$

where

$$K(\mathbf{p}_i^{m+1}) := \sum_j \beta_j \mathbf{p}_{i+j}^{m+1}$$

is an affinely invariant ($\sum_j \beta_j = 0$) measure of local strain energy. Here, the indices i and j are understood to range over all integers for which the associated variables \mathbf{p}^{m+1} are defined. Since K is a local measure, only finitely many

β_j do not vanish. The requirement to place new (odd) vertices such that this functional takes on a minimum, leads to the corresponding discrete Euler-Lagrange equation

$$\forall i: \quad \sum_j \gamma_j \mathbf{p}_{2i+1+j}^{m+1} = 0 \quad (4)$$

where $\gamma_j = \gamma_{-j}$ are the coefficients of the Laurent polynomial $\gamma(z) = \beta(z)\beta(z^{-1})$ with $\beta(z) = \sum_j \beta_j z^j$. Defining the matrix $C := [\gamma_{j-2i-1}]_{i,j}$, we rewrite Equation (4) as

$$C[\mathbf{p}_i^{m+1}] = 0. \quad (5)$$

Using the even and odd upsampling operators

$$U_e := \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \\ \vdots & & & \ddots \end{pmatrix} \quad U_o := \begin{pmatrix} 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 0 & \\ 0 & 0 & 1 & \\ \vdots & & & \ddots \end{pmatrix}, \quad (6)$$

we split a given vector into even and odd components $[\mathbf{p}_i^{m+1}] = U_e[\mathbf{p}_{2i}^{m+1}] + U_o[\mathbf{p}_{2i+1}^{m+1}]$, i.e, into the known and unknown vertices at level $m+1$. Substituting this expression into Equation (5) the matrix C is decomposed into $A := CU_o$ and $B := CU_e$ such that

$$A[\mathbf{p}_{2i+1}^{m+1}] = -B[\mathbf{p}_i^m]. \quad (7)$$

Notice that A is always square and symmetric. Its regularity follows from mild sufficient conditions on the characteristic Laurent polynomial $\beta(z)$ of K . In [19] it is shown that A has full rank if $\beta(z)$ does not have diametric roots $\beta(\pm\omega) = 0$ on the complex unit circle $\omega = e^{2\pi ix}$.

As shown above, a variational refinement scheme is related to its corresponding Euler-Lagrange-equation by $\gamma(z) = \beta(z)\beta(z^{-1})$. On the other hand, using the characteristic Laurent polynomials each (symmetric) linear system of the type (4) can be factored to find a corresponding variational scheme [19].

Examples of Variational Subdivision Four important examples of variational schemes are the minimization of

- Second forward differences

$$K_2(\mathbf{p}_i^{m+1}) := \Delta^2 \mathbf{p}_i^{m+1} = \mathbf{p}_i^{m+1} - 2\mathbf{p}_{i+1}^{m+1} + \mathbf{p}_{i+2}^{m+1}$$

with the Euler-Lagrange equation

$$\forall i: \quad \mathbf{p}_{2i+1}^{m+1} - 4\mathbf{p}_{2i+2}^{m+1} + 6\mathbf{p}_{2i+3}^{m+1} - 4\mathbf{p}_{2i+4}^{m+1} + \mathbf{p}_{2i+5}^{m+1} = 0$$

$$\iff$$

$$\forall i: \quad \Delta^4 \mathbf{p}_{2i+1}^{m+1} = 0.$$

Physically speaking this generates curves which minimize the change in stretch or velocity. Geometrically the curves have minimal total curvature. In terms of digital filters this scheme is equivalent to the 4th order Butterworth halfband filter. Butterworth filters are well known in signal processing for their optimal filter shape [17].

- Third forward differences

$$K_3(\mathbf{p}_i^{m+1}) := \Delta^3 \mathbf{p}_i^{m+1}$$

with the Euler-Lagrange equation

$$\forall i: \quad \Delta^6 \mathbf{p}_{2i}^{m+1} = 0.$$

These curves minimize the change in bending or acceleration. From the geometric point of view this means minimal variation of curvature or equal distribution of curvature. This scheme implements the discrete 6th order Butterworth halfband filter.

- The weighted average

$$K_{\text{spl}}(\mathbf{p}_i^{m+1}) := (1 + \sqrt{3})K_2(\mathbf{p}_i^{m+1}) + (2 + \sqrt{3})K_3(\mathbf{p}_i^{m+1}),$$

with the Euler-Lagrange equation

$$\forall i: \quad 2\Delta^4 \mathbf{p}_{2i+1}^{m+1} - \Delta^6 \mathbf{p}_{2i}^{m+1} = 0. \quad (8)$$

This scheme reproduces the uniformly interpolating cubic splines, i.e., the resulting curves are piecewise cubic polynomials. See the Appendix for a proof of (8).

- The weighted average

$$K_{4\text{pt}}(\mathbf{p}_i^{m+1}) := (3 + \sqrt{3})K_2(\mathbf{p}_i^{m+1}) + K_3(\mathbf{p}_i^{m+1})$$

with the Euler-Lagrange equation

$$\begin{aligned} \forall i: \quad -16\mathbf{p}_{2i+3}^{m+1} &= \mathbf{p}_{2i}^{m+1} - 9\mathbf{p}_{2i+2}^{m+1} - 9\mathbf{p}_{2i+4}^{m+1} + \mathbf{p}_{2i+6}^{m+1} \\ &\iff \\ \forall i: \quad 6\Delta^4 \mathbf{p}_{2i+1}^{m+1} + \Delta^6 \mathbf{p}_{2i}^{m+1} &= 0. \end{aligned}$$

This scheme coincides with the 4-point rule (Equation (1)), exemplifying the fact that interpolatory stationary subdivision schemes are a proper sub-class of variational schemes.

In the first three cases, the new vertices are obtained by solving a tridiagonal system A (see Equation (7)). In the last case, the scheme turns out to be stationary, i.e., $A = I$. Using the formalism developed in [20] one can derive a lower bound for the smoothness of the limit curves P_∞ generated by these schemes. They are C^2 and C^4 for the minimization of K_2 and K_3 respectively. The cubic splines resulting from the minimization of K_{spl} are C^2 . Since the minimization of $K_{4\text{pt}}$ reduces to the 4-point scheme, the resulting limit curves are C^1 [12].

Non-Uniform Subdivision In the above definition of $E(P_{m+1})$, a uniform parameterization of bi-infinite or closed polygons is assumed (no boundaries). Generalizing to non-uniform parameterizations of finite open polygons leads to local fairness measures $K_i(\mathbf{p}_i^{m+1})$ depending on the index i . Similarly the associated non-uniform Euler-Lagrange equations

$$C[\mathbf{p}_i^{m+1}] = 0 \quad (9)$$

reflect the parameterization. Each row of the banded but no longer Toeplitz matrix C corresponds to the partial derivative of $E(P_{m+1})$ with respect to a newly inserted vertex \mathbf{p}_{2i+1}^{m+1} . Hence, the variational refinement schemes automatically treat the special cases at the boundaries of open polygons since partial derivatives are well-defined for all odd-indexed vertices \mathbf{p}_{2i+1}^{m+1} (natural boundary conditions).

The schemes based on the minimization of K_2 (see next paragraph) and K_3 are easily generalized to non-uniform parameterizations by going from forward differences to *divided* differences. The generalization of $K_{4\text{pt}}$ was already given by Equation (3). Generalizing K_{spl} to the reproduction of non-uniform cubic splines is slightly more involved since a piecewise polynomial basis has to be constructed explicitly. The matrix C_{spl} is derived in Appendix A.

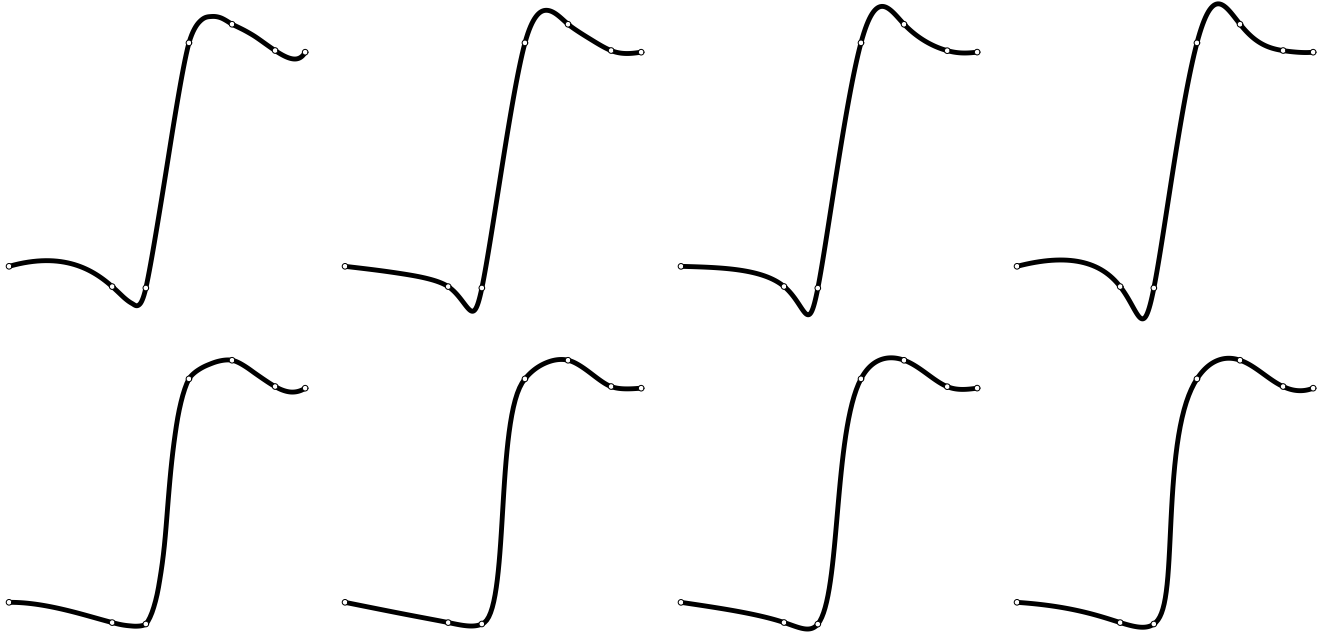


Figure 4: Comparison of the curves which minimize K_{4pt} , K_2 , K_{spl} , and K_3 (left to right) with uniform (top) and centripetal (bottom) parameterizations.

Here we limit ourselves to the case of the second divided difference energy. The third divided difference case is entirely similarly albeit more tedious. Let us assume that each vertex \mathbf{p}_i^{m+1} is associated with the parameter value t_i^{m+1} . The divided difference operators are recursively defined by

$$[t_i^{m+1}, \dots, t_{i+k}^{m+1}] \mathbf{p}_i^{m+1} = \frac{[t_{i+1}^{m+1}, \dots, t_{i+k}^{m+1}] \mathbf{p}_{i+1}^{m+1} - [t_i^{m+1}, \dots, t_{i-1+k}^{m+1}] \mathbf{p}_i^{m+1}}{t_{i+k}^{m+1} - t_i^{m+1}}$$

and the second divided difference energy is

$$K_2(\mathbf{p}_i^{m+1}) = \frac{\mathbf{p}_{i+2}^{m+1}}{\Delta t_{i+1}^{m+1} (\Delta t_{i+1}^{m+1} + \Delta t_i^{m+1})} - \frac{\mathbf{p}_{i+1}^{m+1}}{\Delta t_{i+1}^{m+1} \Delta t_i^{m+1}} + \frac{\mathbf{p}_i^{m+1}}{(\Delta t_{i+1}^{m+1} + \Delta t_i^{m+1}) \Delta t_i^{m+1}}.$$

The partial derivative of $K_2(\mathbf{p}_j^{m+1})^2$ with respect to \mathbf{p}_{2i+1}^{m+1} is $h_{2i+1} K_2(\mathbf{p}_j^{m+1})$ where h_{2i+1} is the linear coefficient of \mathbf{p}_{2i+1}^{m+1} in $K_2(\mathbf{p}_j^{m+1})^2$. Summing the $h_{2i+1} K_2(\mathbf{p}_j^{m+1})$ for $j = 2i - 1, 2i, 2i + 1$ gives the i -th row of the Euler-Lagrange matrix C . The matrix is penta-diagonal and separating the known \mathbf{p}_{2i}^{m+1} from the unknown \mathbf{p}_{2i+1}^{m+1} yields a tridiagonal system with non-constant entries along the diagonals (main, upper, and lower).

Visual Evaluation of the Fairness Figure 4 shows a comparison of the curves resulting from variational subdivision. Going from left to right the curves minimize K_{4pt} , K_2 , K_{spl} , and K_3 . The top row uses a uniform parameterization while the bottom uses the centripetal parameterization. Note that the computational cost of minimizing K_{spl} and K_3 is the same, yet the latter appears noticeably fairer. This is verified by looking at the curvature plots of both curves in Figure 5.

1.2 Summary

In this section we reviewed classic interpolatory subdivision and the recently introduced variational subdivision setting. We extended the latter to the non-uniform parameterization setting. It is well known in the CAGD community

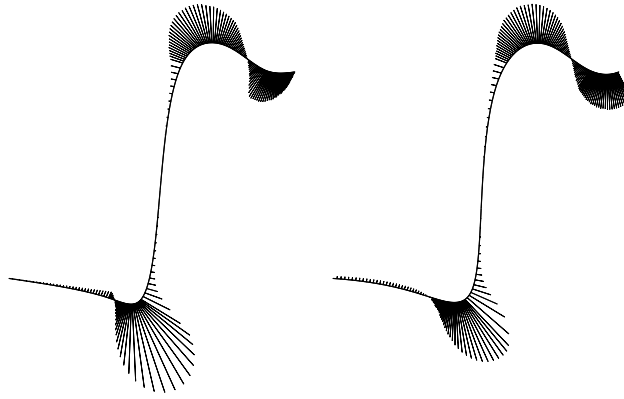


Figure 5: Curvature distribution of the cubic splines (left) compared to $\min \sum K_3(\mathbf{p}_i)^2$ – curves (right).

(see for example [18]) that both variational settings and non-uniform settings are very important in practice, giving a much richer foundation on which to build smooth curves. We gave a simple example hinting at some of the differences between the different variational energy measures and the effects of non-uniform parameterization.

Clearly the design of particular variational schemes together with the various ways of introducing non-uniform parameterization is a whole subject worthy of investigation all by itself. Instead of pursuing this direction we continue with the construction of a multiresolution setting, i.e., the introduction of details into the subdivision process. Such multiresolution representations have already proven themselves to be important for applications (e.g. [14]) and we will show how to build them in a simple and straightforward way in the case of non-uniform variational subdivision schemes.

Part 2: Wavelets for Variational Subdivision

Overview At this point all we need is a banded linear system solver, for example LU decomposition, and we are ready to construct variationally optimal interpolating curves. However, we also aim to construct wavelets which are appropriate for these subdivision curves, i.e., for the underlying scaling functions. Since Fourier techniques cannot be used in the non-uniform setting we need to use an alternative construction technique which is situated in the spatial rather than frequency domain. This can be done either through direct inversion of large, sparse linear systems or, equivalently, but much more elegantly and with more insight through the use of the Lifting Scheme [28].

To have a convenient means to discuss the lifting scheme and exhibit its essential simplicity we begin by describing the *wiring diagram* formalism, which is commonly used to describe the manipulation of FIR filter based bi-orthogonal wavelet transforms [29]. In the lifting scheme each block of the wiring diagram corresponds to an individual lifting step. These steps can be composed into *sequences* to obtain more complex filters. In the notation of linear algebra each building block is represented by a very simple, special matrix and the composition corresponds to matrix multiplication. An added benefit of the use of lifting steps and wiring diagrams is that they can be transcribed literally into very simple and efficient C++ class objects.

Finally, for the banded linear system solver we choose *Cyclic Reduction*, rather than, for example, LU decomposition. Cyclic Reduction has the decisive advantage of being expressible entirely in terms of lifting steps as we will show in Section 2.4. This observation seamlessly integrates variational subdivision into the classical lifting framework, *extending* it from the FIR setting to certain IIR settings. Because of this connection between lifting and cyclic reduction, we are able to give a straightforward construction of an *initial completion*, i.e., functions which span the space of details between two scales of a multiresolution analysis based on a variational subdivision scheme. Applying additional lifting steps we show in Section 2.5 how the resulting decompositions can be stabilized.

By way of motivation we begin with a very simple example of a wiring diagram [29] and its inverse:

Example Figure 6 shows a wiring diagram implementing an interpolatory refinement scheme through lifting. The points at level m enter the diagram on the left side. After upsampling, i.e., inserting zeros between the elements, they are split into even (upper wire) and odd (lower wire). This is done by a splitting T-node. We think of this part as the input stage to the subdivision scheme. Note that the even wire carries all the vertices from level m ($\mathbf{p}_{2i}^{m+1} = \mathbf{p}_i^m$) and the odd wire only carries zeros so far. The even values flow unmodified through the subdivision diagram to the T-node at the right end. This node implements a merge operation, the inverse of splitting. The odd vertices result from the even values flowing through the *Predict Odd* box, P_o . It represents the action of computing odd elements as a weighted average of even elements (using α_i of (2)). The term predict is chosen since interpolation “predicts” intermediate values between given samples.

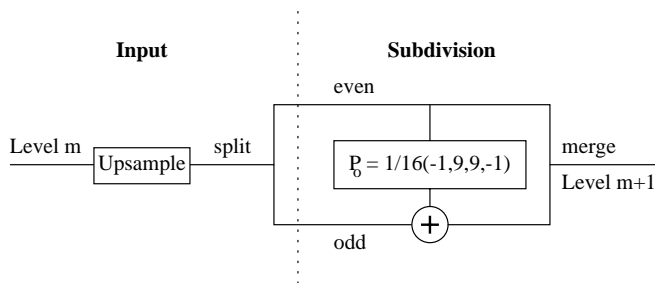


Figure 6: Example of a wiring diagram. Points at level $m + 1$ are generated by upsampling level m followed by a split into even and odd. The odd locations are then filled in through the action of a P_o (“Predict Odd”) box using the weights α_i (for example, the weights of the 4-point scheme). Finally, even and odd locations are merged.

Wavelet Transforms Given a subdivision wiring diagram there is an immediate way to describe a “wavelet-like” transform. Recall that a wavelet transform starts at a finer level, say $m + 1$, and computes a coarser approximation at level m , as well as a set of detail, or *wavelet*, coefficients. These coefficients encode the difference between the signal at the finer level $m + 1$ and its approximation at the coarser level m . The wiring diagram on the left of Figure 7 has this structure. It is obtained by inverting the diagram on the right which is the subdivision diagram of Figure 6.

In the case of the inverse diagram, coefficients at level $m + 1$ enter on the left and are split into even and odd. Now the P_o box uses the even values to predict a value at the odd location. Subtracting this prediction we get a wavelet coefficient, i.e., the difference of the approximation at level m (even wire) and the original data from level $m + 1$. Using the output of the wavelet diagram as input to the subdivision diagram, the original sequence is rebuilt. This time the odd wire does not carry the zeros of upsampling, but rather the wavelet coefficients.

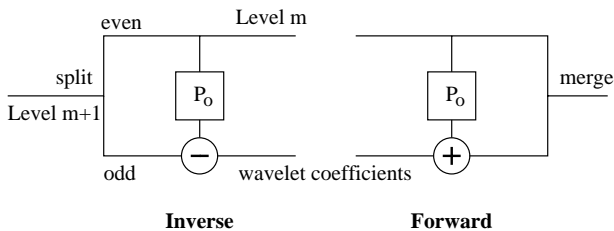


Figure 7: Wiring diagrams are easy to invert. Simply flip left to right and exchange signs. On the left is the inverse (wavelet) diagram of the (subdivision) wiring diagram on the right.

Discussion As simple as the above example is, it illustrates the basic ideas for the rest of the paper well. Build the subdivision scheme as a wiring diagram consisting of lifting steps, then invert the diagram to get a corresponding

wavelet transform. Merely flipping the subdivision diagram does not yield a particularly good wavelet transform,¹ but it has the correct formal structure. We will see later how to improve the numerical properties of these transforms (see Section 2.5).

In the above example we could have also used Fourier analysis, or generating functions, to make the same argument. However, these techniques only work in the uniform, bi-infinite setting, which is much too restrictive for the settings that appear in graphics applications.

Alternatively, everything in the above example could have been written as the application of a matrix and its inverse respectively. However, writing it in terms of wiring diagrams it becomes easy to see that in the above example both the subdivision matrix and its inverse are banded. That the inverse is banded is far from obvious when considering subdivision as the application of a matrix. In the non-uniform setting this will be even less obvious. Using wiring diagrams and the lifting scheme, which is the formal basis for their use, finding inverses for example will be a trivial exercise and properties such as bandedness will be evident. In the next section we put these observations on a formal basis.

2.3 A Complete Wiring Diagram Toolbox

So far we have only used P_o boxes and split or merge T-nodes in our diagrams. To complete the set of tools we need to introduce “Predict Even” and “Scale” boxes.

Example Leaving our framework of interpolatory subdivision schemes for a moment, we use cubic splines as an example of subdivision requiring the additional elements. Figure 8 shows a diagram which implements the well known algorithm for uniform subdivision of cubic splines [22]

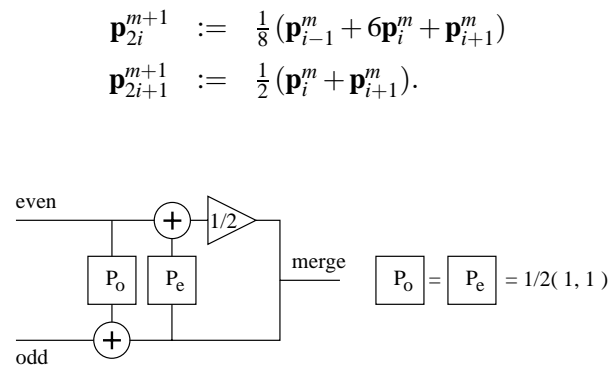


Figure 8: In this example a predict odd box P_o followed by a predict even box P_e , and finally a scale on the even wire generates (non-interpolating) cubic splines. Due to the Predict Even box, the scheme is no longer interpolatory.

The Toolbox Figure 9 shows the complete set of wiring diagram elements corresponding to elementary lifting steps. The merging T-node is an implementation of $U := (U_e | U_o)$ with the upsampling matrices of (6). Accordingly, the splitting T-node implements its inverse $U^{-1} = U^T$. Note that the split operator U^T effectively puts all even elements into the top half and all odd elements into the bottom half of its output vector (lazy wavelet transform [28]).

¹The condition number of the associated transform is not uniformly bounded independent of the number of levels over which we perform it.

With this convention, all other operators correspond to matrices in an obvious way

$$\begin{aligned}
 P_o &= \begin{pmatrix} I & 0 \\ A & I \end{pmatrix} & P_o^{-1} &= \begin{pmatrix} I & 0 \\ -A & I \end{pmatrix} \\
 P_e &= \begin{pmatrix} I & A \\ 0 & I \end{pmatrix} & P_e^{-1} &= \begin{pmatrix} I & -A \\ 0 & I \end{pmatrix} \\
 S_e &= \begin{pmatrix} D & 0 \\ 0 & I \end{pmatrix} & S_e^{-1} &= \begin{pmatrix} D^{-1} & 0 \\ 0 & I \end{pmatrix} \\
 S_o &= \begin{pmatrix} I & 0 \\ 0 & D \end{pmatrix} & S_o^{-1} &= \begin{pmatrix} I & 0 \\ 0 & D^{-1} \end{pmatrix}
 \end{aligned}$$

Each of these operators is a 2×2 block matrix, with A being banded and D diagonal. Putting several wiring diagram elements together corresponds to multiplying the associated matrices in the same order. Since each wiring diagram element, or matrix, is trivial to invert or transpose it is easy to invert or transpose whole diagrams: Simply reverse the order of stages and invert or transpose each stage.

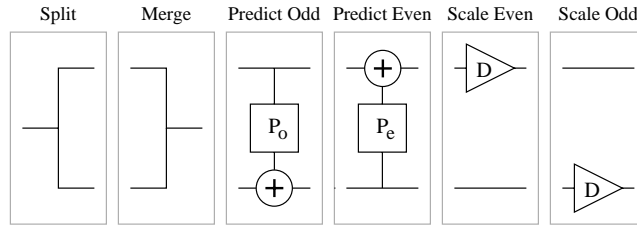


Figure 9: All the elements needed to build wiring diagrams for subdivision schemes and wavelet transforms.

As an example, we can write the diagrams of Figures 6 and 8 in matrix notation

$$P_{m+1} = U \begin{pmatrix} I & 0 \\ A & I \end{pmatrix} \begin{pmatrix} P_m \\ Q_m \end{pmatrix} \quad \text{and}$$

$$P_{m+1} = U \begin{pmatrix} D & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & A \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ A & I \end{pmatrix} \begin{pmatrix} P_m \\ Q_m \end{pmatrix}$$

respectively. Here Q_m denotes the coefficients entering the diagram on the odd wire. In pure subdivision these are all zero. In wavelet transforms they denote the wavelet coefficients. The matrix A has coefficients $\frac{1}{2}$ on the main and upper diagonal and $D = \frac{1}{2}I$. The inverses of these expressions are easy to write down as well

$$\begin{pmatrix} P_m \\ Q_m \end{pmatrix} = \begin{pmatrix} I & 0 \\ -A & I \end{pmatrix} U^T P_{m+1} \quad \text{and}$$

$$\begin{pmatrix} P_m \\ Q_m \end{pmatrix} = \begin{pmatrix} I & 0 \\ -A & I \end{pmatrix} \begin{pmatrix} I & -A \\ 0 & I \end{pmatrix} \begin{pmatrix} D^{-1} & 0 \\ 0 & I \end{pmatrix} U^T P_{m+1}.$$

This scheme of factoring subdivision [5] as well as wavelet transforms [1] is quite general and it can be shown that *any* finitely supported wavelet transform can be written as a sequence of lifting steps [7]. Consequently writing a given subdivision scheme as a wiring diagram immediately gives us access to *all* associated bi-orthogonal wavelet transforms [30]. In Section 2.4 we will show that these ideas also apply to the variational schemes of Section 1.1. In this way we extend the traditional lifting scheme setting from FIR filters, or banded matrices with banded inverses, to the setting of IIR filters, which are dense inverses of banded matrices.

Its solution

$$x_e = (RAU_e)^{-1} Rb \quad (15)$$

is computed by applying CR recursively. Since R is a projection, the solutions of (13) are a subset of the solutions of (14). Hence, if RAU_e is regular, Equation (15) uniquely determines the even components of the solution x of (13). Back-substitution into (13),

$$AU_o x_o = b - AU_e x_e$$

and multiplication on the left by U_o^T yields

$$x_o = (U_o^T AU_o)^{-1} (U_o^T b - U_o^T AU_e x_e). \quad (16)$$

Wiring Diagram Implementation Formulæ (15) and (16) are implemented as a wiring diagram. The matrix R is of type (11) but with arbitrary weights and we decompose $Rb = RU_e b_e + RU_o b_o$ where the matrix RU_e is diagonal. Similarly, $U_o^T AU_o$ is diagonal since A is tridiagonal. Figure 11 shows the corresponding diagram. The $(RAU_e)^{-1}$ box is replaced recursively by another copy of the basic CR diagram with a projection matrix R' so that the odd columns of RAU_e are eliminated, etc.

Expanding the recursion, we obtain a wiring diagram *network* with several branches consisting of only the building blocks of Section 1.2. The property that inversion and transposition of such diagrams is trivial continues to hold. Further, since the lifting scheme is also expressed in terms of wiring diagrams, we can integrate the whole CR implementation as a composite component into a larger diagram and use additional lifting steps to obtain stable bi-orthogonal wavelet bases (see Section 2.5).

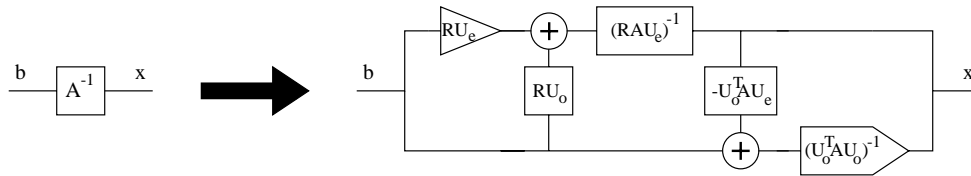


Figure 11: A wiring diagram implementation of a single step of the general CR algorithm. The box $(RAU_e)^{-1}$ at the top contains recursive CR wiring diagrams. Each scaling box requires one multiplication per value and the predict boxes require two multiplications and one addition.

Finite Precision Arithmetic If all arithmetic operations are performed without rounding error, the CR recursion will stop when the linear system is reduced to a scalar equation. However, implementations in finite precision arithmetic can stop the recursion much earlier without affecting the result.

In the example of minimization of second forward differences, CR starts with the tridiagonal matrix $A = \text{tridiag}(1, 6, 1)$ and generates a new matrix $A' = RAU_e = \text{tridiag}(-1, 34, -1)$. The next CR step generates $A'' = \text{tridiag}(-1, 1154, -1)$ and, in general, the k th step transforms $A^{(k)} = \text{tridiag}(-1, v_k, -1)$ into $A^{(k+1)} = \text{tridiag}(-1, v_k^2 - 2, -1)$. Consequently the number of significant digits of the ratio between diagonal and off-diagonal elements is doubled in each CR step. After a few iterations the matrix $A^{(k)}$ can no longer be distinguished from a diagonal matrix in finite precision arithmetic. At that point an approximate inverse to within machine epsilon of $A^{(k)}$ can be computed by simply neglecting the off-diagonal elements. If we use a floating point representation with r mantissa bits $\lceil \log_2 r \rceil$ CR steps are sufficient to compute the answer. A similar, but more involved argument, can be made for the general, non-Toeplitz setting of (12).

Computational Cost CR not only mimics the binary multi-scale characteristics of subdivision and wavelet transforms, but it is also very efficient. As seen in Figure 11, the computational complexity of a single CR step is $[\otimes 3n \oplus 2n]$ multiplications and additions per scalar component for n vertices. Accounting for all recursive calls leads

to an overall operations count of $[\otimes 6n \oplus 4n]$, which compares favorably with LU decomposition, for example. In the uniform case this can be reduced to $[\otimes 2n \oplus 4n]$ operations by exploiting symmetry.

To obtain the complete wiring diagram for a given variational scheme, we need an initial predict odd box to compute the right hand side of Equation (7), which is the input for the CR diagram (see Figure 12). The additional cost is at most $[\otimes 4n \oplus 3n]$ ($[\otimes 2n \oplus 3n]$ for uniform schemes). Thus the total FLOP-count adds up to $[\otimes 10 \oplus 7]$ per output value ($[\otimes 4 \oplus 7]$ for uniform schemes). By way of comparison we note that the de Boor algorithm for the evaluation of non-uniform splines requires $[\otimes 12 \oplus 6]$ per value in the cubic case.

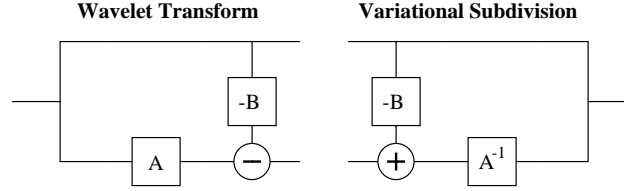


Figure 12: Complete implementation of a variational subdivision scheme as a wiring diagram, together with the associated wavelet transform. The boxes A and A^{-1} hide recursive CR subdiagrams.

Discussion In this section we have shown how CR is implemented using only the basic building blocks of the wiring diagram formalism. In this way we can build wavelet filter banks with the primitives of the lifting scheme not only in the FIR filter setting, but also in the setting of IIR filters if these arise as inverses of banded matrices. The latter covers all variational schemes with finitely supported energy measures. Expressing the banded linear system solver in this way we can immediately leverage the advantages of the lifting scheme for variational multiresolution settings: (1) easy invertibility in terms of simple elementary steps even in the non-uniform setting; (2) efficient and straightforward implementation. What we don't yet have are numerically well behaved wavelet transforms. However, additional lifting steps can address this issue and are easily incorporated since we are in the wiring diagram framework. Stabilizing, i.e., improving the condition, of the multiresolution transforms is the subject of the next section.

2.5 Stable Wavelet Bases

In order to understand the numerical condition of our transforms better we need to examine more closely the underlying linear systems. According to Equation (7), the solution of a variational subdivision scheme $C[\mathbf{p}_i^{m+1}] = 0$ can be computed as

$$[\mathbf{p}_{2i+1}^{m+1}] = -A^{-1}B[\mathbf{p}_{2i}^{m+1}] = -A^{-1}B[\mathbf{p}_i^m] \quad (17)$$

where $A = CU_o$ and $B = CU_e$. Hence, the subdivision matrix Φ mapping $[\mathbf{p}_i^m]$ to $[\mathbf{p}_i^{m+1}]$ is given by

$$\Phi := U_e - U_o A^{-1} B.$$

In the context of wavelet decompositions this is the basis transform from the m -th scale space to the $(m+1)$ -st scale space. The wavelets enter as a set of functions which describe the additional degrees of freedom when going from level m to $(m+1)$. In our subdivision wiring diagrams the wavelets “live” on the odd wires. There are many possible choices of wavelets for a given set of scaling functions. In the lifting framework *all* bi-orthogonal choices can be described as long as one initial choice is given. This is often referred to as the initial completion [32, 1, 28]. In terms of linear algebra, finding an initial completion is equivalent to asking for a matrix Ψ such that the matrix $(\Phi|\Psi)$ is invertible. Posed as a linear algebra question it is not immediately obvious how to find such a Ψ . In the lifting framework it is trivial: simply consider the odd wire of the wiring diagram implementation of the subdivision scheme. In our case this amounts to

$$\Psi := U_o A^{-1}$$

(see Figure 12): The detail coefficients entering the diagram on the odd wire flow through the CR block, but not through the initial predict odd box which computes $-B[\mathbf{p}_i^m]$. The matrices Φ and Ψ together build a reconstruction, or synthesis, matrix

$$S = (\Phi | \Psi).$$

The columns of this matrix give the two-scale relations for the primal scaling functions and wavelets respectively. The inverse is surprisingly simple. It is

$$S^{-1} = \begin{pmatrix} U_e^T \\ C \end{pmatrix}. \quad (18)$$

This follows immediately by inverting the subdivision diagram (Figure 12) and transcribing it into a matrix equation.

In the wavelet setting the matrix $2S^{-T} =: (\tilde{\Phi} | \tilde{\Psi})$ plays an important role. It is the dual reconstruction matrix and its columns give the two-scale relation coefficients for the dual scaling functions and dual wavelets respectively. It is interesting to notice that although the variational schemes are globally supported (S is a full matrix in general), their dual scaling functions and wavelets are generated by stationary subdivision schemes.

Just as S controls the analytic properties of the subdivision limit functions and associated wavelets, so does $2S^{-T}$ control the properties of the dual scaling functions and wavelets respectively. For example, if both subdivision schemes Φ and $\tilde{\Phi}$ are convergent we have a complete bi-orthogonal wavelet basis setting. Even though in applications we may only appear to be working with the primal scaling functions and wavelets, the dual functions enter in the way we go from a fine function to a coarse function. In this way the properties of the dual functions control the numerical condition of the transform. Recall that the condition of a linear transform encodes the relationship between the magnitude of a vector and its transform. If this relationship is not bounded by a constant independent of the number of levels over which the wavelet transform is applied we say that the transform is not stable. Practically this means that small errors can get magnified quickly, rendering the transform useless.³ A sufficient condition for stability of the transform is the continuity of the basic solutions of the corresponding two-scale relations [4]. On the primal side this is satisfied since Φ does generate at least continuous functions for all the examples we considered. The dual subdivision operator $\tilde{\Phi} = 2U_e$ has to be modified to bring its condition under control. If it can be achieved at all, it can be achieved through additional lifting steps, since all bi-orthogonal transforms for a given subdivision scheme can be found through repeated lifting.

The most natural approach to consider then is to find the lifting with the least support, which ensures a continuous dual scaling function.

Stabilizing the Basis In the bi-orthogonal setting, additional lifting steps can be used to improve the smoothness properties of $\tilde{\Phi}$ [28]. The primal synthesis matrix S is replaced by

$$T := S \begin{pmatrix} I & -V \\ 0 & I \end{pmatrix} = (\Phi | \Psi - \Phi V), \quad (19)$$

that is, we add a linear combination of scaling functions to the primal wavelets. Typically one chooses V to increase the number of vanishing moments of the primal wavelets. This is motivated by the fact that at least 1 vanishing moment is a necessary condition for stability [6]. Additional vanishing moments on the primal wavelet increase the approximation order of the dual scaling functions eventually leading to stability. Lifting then changes S^{-T} to

$$T^{-T} := S^{-T} \begin{pmatrix} I & -V \\ 0 & I \end{pmatrix}^{-T} = S^{-T} \begin{pmatrix} I & 0 \\ V^T & I \end{pmatrix},$$

that is, the dual scaling functions are modified by adding the same linear combination of dual wavelets.

Translated into wiring diagrams, stabilizing the scheme S through an additional lifting step is equivalent to including a predict even box $P_e(V)$ as the first element in the diagram (see Equation (19)). Note that the subdivision scheme itself is not affected by this modification since for pure subdivision the odd wire carries only zeroes (see Figure 13).

³Although in some cases the instability is so weak that for all practical purposes no ill effects appear.

Scheme	columns of V
K_2	$\frac{1}{64}[-3, 15, 15, -3]^T$
K_3	$\frac{1}{512}[-25, 105, 105, -25]^T$
K_{spl}	$\frac{1}{144}[-7, 31, 31, -7]^T$
$K_{4\text{pt}}$	$\frac{1}{32}[-1, 9, 9, -1]^T$

Table 1: Lifting weights for variational subdivision schemes leading to stable transforms in the uniform setting. These weights were computed by considering the first 4 moments of the scaling functions and wavelets and then combining them in such a way as to ensure 4 vanishing moments on the primal wavelet. Details of this computation can be found in [27].

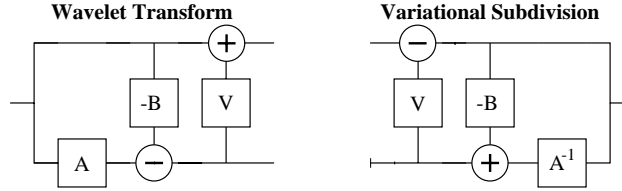


Figure 13: Implementation of a variational subdivision scheme with an additional lifting step to obtain stability of the corresponding wavelet decomposition.

The matrix V is usually chosen to be a banded matrix with a minimal number of non-vanishing diagonals. The coefficients in one column of V are the weights to linearly combine a primal wavelet with neighboring scaling functions such that their moments cancel one another. In general this requires the solution of a linear system. Table 1 shows the resulting weights for our variational schemes. In the non-uniform setting moments are computed recursively and the weights in $P_e(V)$ are based on the solution of a small linear system for each i (details of these linear systems are described in [27]).

Remark: The classical theory regarding the stability of the wavelet basis and its connection to vanishing moments of the wavelets was developed for the uniform setting only [6]. However, numerical experiments performed by the authors with the correspondingly stabilized non-uniform schemes presented in this paper indicate that the condition number of the transforms remains bounded just as in the uniform case.

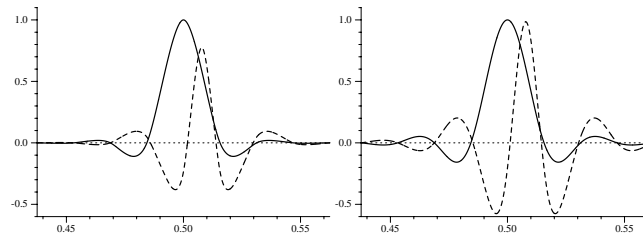


Figure 14: Scaling functions (solid line) and wavelets (dashed line) in the uniform setting (K_2 on the left, K_3 on the right). In both cases the wavelets have 4 vanishing moments.

Examples: Figure 14 shows the wavelets resulting from our construction for K_2 and K_3 (uniform setting). In both cases they have 4 vanishing moments. The scaling functions are the impulse response of a 4th and 6th order Butterworth filter respectively. In contrast to the classical, Fourier based filter setting our construction continues to work in the non-uniform setting. Figure 15 shows an application of this. From left to right we see successively

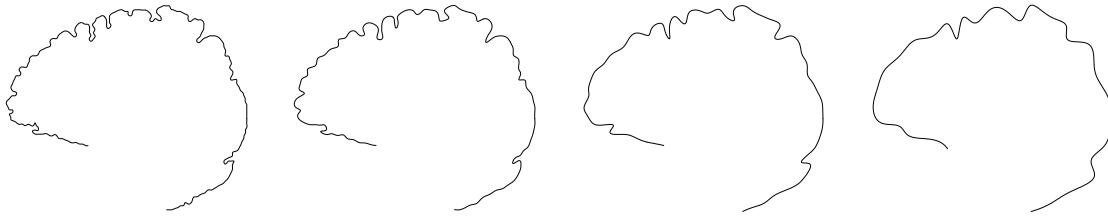


Figure 15: Result of applying a wavelet transform to an original finest level curve. Left to right: original curve, and successively coarser approximations. (Using K_3 with centripetal parameterization and lifting.)

coarser levels of approximation to an initial data set. We are effectively using a generalization of a Butterworth, bi-orthogonal filter bank in a non-uniform setting.

Conclusion

In this paper we showed how the construction of variationally optimal, interpolating curves can be combined with the subdivision paradigm. The implementation of variational schemes as a network of lifting steps and the implied matrix decomposition of the corresponding operator allowed us to derive associated wavelets. We used wiring diagrams as a formalism to reason about these transforms and showed how the CR algorithm provides the necessary element to include certain IIR filters in a lifting framework. Applying further lifting steps to these initial transforms we were able to construct stable wavelet bases in the uniform setting. The subdivision rules are finitely supported in finite precision arithmetic and can be computed with a low FLOP count. Generalization to the non-uniform setting is straightforward and leads to curves of high quality. The result is a family of curves which includes interpolating cubic splines as a special case, and is a proper superset of interpolatory, stationary schemes. Our constructions make variationally optimal curves and associated wavelets available to any algorithm which exploits multiresolution representations, e.g., hierarchical editing and compression.

Some questions are particularly interesting for future work:

- So far we can only prove the stability of the variationally optimal wavelet transforms in the uniform setting. Numerical experiments suggest that the transforms remain stable in the non-uniform parameter setting as well. Since adaptive and non-uniform constructions are important in applications, a stronger theoretical foundation would be desirable.
- The application of these ideas to the surface setting is the most exciting avenue. Unfortunately, except for the trivial tensor-product setting, no 2D generalization of CR is known and it appears unlikely. Other techniques will have to be employed to create fast solvers for the variational surface setting.

Acknowledgments

We are greatly indebted to Wim Sweldens for many discussions which helped refine the wiring diagram ideas and who also pointed out the connection to cyclic reduction.

This research was supported in part through grants from the Intel Corporation, the Charles Lee Powell Foundation, the Sloan Foundation, and an NSF CAREER award (ASC-9624957) to the second author.

References

- [1] CARNICER, J. M., DAHMEN, W., AND PEÑA, J. M. Local decompositions of refinable spaces. *Appl. Comput. Harmon. Anal.* 3 (1996), 127–153.

- [2] CAVARETTA, A. S., DAHMEN, W., AND MICCHELLI, C. A. Stationary Subdivision. *Memoirs Amer. Math. Soc.* 93, 453 (1991).
- [3] CELNIKER, G., AND GOSSARD, D. Deformable Curve and Surface Finite Elements for Free-Form Shape Design. *Computer Graphics* 25 (1991), 257–265.
- [4] DAHMEN, W. Stability of multiscale transformations. Tech. rep., Institut für Geometrie und Praktische Mathematik, RWTH Aachen, 1994.
- [5] DAHMEN, W., AND MICCHELLI, C. A. Banded Matrices with Banded inverses II: Locally Finite decompositions of spline spaces. *Constr. Approx.* 9, 2-3 (1993), 263–281.
- [6] DAUBECHIES, I. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
- [7] DAUBECHIES, I., AND SWELDENS, W. Factoring wavelet and subband transforms into lifting steps. Preprint, 1996.
- [8] DESLAURIERS, G., AND DUBUC, S. Interpolation Dyadique. In *Fractals, dimensions non entières et applications*. Masson, Paris, 1987, pp. 44–55.
- [9] DUBUC, S. Interpolation Through an Iterative Scheme. *Jour. of Mathem. Anal. and Appl.* 114 (1986), 185–204.
- [10] DYN, N. Subdivision Schemes in Computer Aided Geometric Design. In *Advances in Numerical Analysis II, Wavelets, Subdivision and Radial Functions*, W. Light, Ed. Oxford University Press, 1991, pp. 36–104.
- [11] DYN, N., GREGORY, J., AND LEVIN, D. A 4-point Interpolatory Subdivision Scheme for Curve Design. *Comput. Aided Geom. Des.* 4 (1987), 257–268.
- [12] DYN, N., LEVIN, D., AND MICCHELLI, C. A. Using Parameters to Increase Smoothness of Curves and Surfaces Generated by Subdivision. *Comput. Aided Geom. Des.* 7 (1990), 129–140.
- [13] FARIN, G. *Curves and Surfaces for CAGD*, 3rd ed. Academic Press, 1993.
- [14] FINKELSTEIN, A., AND SALESIN, D. H. Multiresolution Curves. *Computer Graphics Proceedings, Annual Conference Series*, 261–268, July 1994.
- [15] GOLUB, G. H., AND LOAN, C. F. V. *Matrix Computations*, 2nd ed. The John Hopkins University Press, Baltimore, 1983.
- [16] GORTLER, S. J., AND COHEN, M. F. Hierarchical and Variational Geometric Modeling with Wavelets. In *Proceedings Symposium on Interactive 3D Graphics*, May 1995.
- [17] HERLEY, C., AND VETTERLI, M. Wavelets and Recursive Filter Banks. *IEEE Trans. Sig. Proc.* 41 (1993), 2536–2556.
- [18] HOSCHEK, J., AND LASSER, D. *Fundamentals of Computer Aided Geometric Design*. AK Peters, 1993.
- [19] KOBBELT, L. A Variational approach to subdivision. *Comput. Aided Geom. Des.* 13 (1996), 743–761.
- [20] KOBBELT, L. Using the Discrete Fourier-Transform to Analyze the Convergence of Subdivision Schemes. *Appl. Comp. Harmonic Anal.* 5 (1998), 68–91.
- [21] KOBBELT, L. *Variational Design with Parametric Meshes of Arbitrary Topology*. Teubner, 1998.
- [22] LANE, J., AND RIESENFELD, R. A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces. *IEEE Trans. on Pattern Anal. and Mach. Int.* 2 (1980), 35–46.

- [23] LEE, E. T. Y. Choosing nodes in parametric curve interpolation. *Computer Aided Design* 21, 6 (August 1989), 363–370.
- [24] LIU, Z., GORTLER, S. J., AND COHEN, M. F. Hierarchical Spacetime Control. *Computer Graphics Proceedings*, Annual Conference Series, 35–42, July 1994.
- [25] REISSELL, L.-M. Wavelet Multiresolution Representation of Curves and Surfaces. *CVGIP: Graphical Models and Image Processing* 58, 2 (1996), 198–217.
- [26] SAPIDIS, N. *Designing Fair Curves and Surfaces*. SIAM, Philadelphia, 1994.
- [27] SCHRÖDER, P., AND SWELDENS, W. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Computer Graphics Proceedings*, Annual Conference Series, 161–172, 1995.
- [28] SWELDENS, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* 3, 2 (1996), 186–200.
- [29] SWELDENS, W., AND SCHRÖDER, P. Building your own Wavelets at Home. In *Wavelets in Computer Graphics*. ACM SIGGRAPH Course notes, 1996, pp. 15–87.
- [30] VETTERLI, M., AND HERLEY, C. Wavelets and filter banks: Theory and design. *IEEE Trans. Acoust. Speech Signal Process.* 40, 9 (1992), 2207–2232.
- [31] WARREN, J. Binary Subdivision Schemes for Functions over Irregular Knot Sequences. In *Mathematical Methods for Curves and Surfaces*, M. Daehlen, T. Lyche, and L. Schumaker, Eds. 1995, pp. 543–562.
- [32] WARREN, J. Subdivision Methods for Geometric Design. Unpublished manuscript, November 1995.
- [33] WELCH, W., AND WITKIN, A. Free-Form Shape Design Using Triangulated Surfaces. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, A. Glassner, Ed., Computer Graphics Proceedings, Annual Conference Series, 247–256, July 1994.

A Euler-Lagrange Equations for non-uniform Interpolating Cubic Splines

We derive the entries of the matrix C_{spl} for the case of non-uniform interpolating cubic splines. As before we define the parameter values of new intermediate points to be $t_{2i+1}^{m+1} := \frac{1}{2}(t_i^m + t_{i+1}^m)$. The resulting parameterization is *piecewise uniform*.

To go from P_m to P_{m+1} , we have to compute the vertices \mathbf{p}_{2i+1}^{m+1} while $\mathbf{p}_{2i}^{m+1} = \mathbf{p}_i^m$ are already given by the interpolation condition. Let s_i be the polynomial segments of the cubic spline over the knot sequence $[t_i^m] = [t_{2i}^{m+1}]$ with $s_{i-1}(t_i^m) = \mathbf{p}_i^m = s_i(t_i^m)$. We represent each segment with respect to a special basis

$$s_i(t) = \sum_{j=0}^3 \alpha_{i,j} b_{i,j}(t)$$

where

$$\begin{aligned} b_{i,0}(t) &= \frac{2}{(\Delta t_i^m)^2} (t - t_{2i+1}^{m+1})(t - t_{2i+2}^{m+1}) \\ b_{i,1}(t) &= \frac{4}{(\Delta t_i^m)^2} (t - t_{2i}^{m+1})(t - t_{2i+2}^{m+1}) \\ b_{i,2}(t) &= \frac{2}{(\Delta t_i^m)^2} (t - t_{2i}^{m+1})(t - t_{2i+1}^{m+1}) \\ b_{i,3}(t) &= \frac{2}{(\Delta t_i^m)^3} (t - t_{2i}^{m+1})(t - t_{2i+1}^{m+1})(t - t_{2i+2}^{m+1}) \end{aligned}$$

and hence

$$\alpha_{i,0} = s_i(t_{2i}^{m+1}) = \mathbf{p}_i^m = \mathbf{p}_{2i}^{m+1} \quad \alpha_{i,1} = s_i(t_{2i+1}^{m+1}) = \mathbf{p}_{2i+1}^{m+1} \quad \alpha_{i,2} = s_i(t_{2i+2}^{m+1}) = \mathbf{p}_{i+1}^m = \mathbf{p}_{2i+2}^{m+1}.$$

The linear system (9) which defines the new vertices \mathbf{p}_{2i+1}^{m+1} , is set up by transforming the C^2 -condition between successive polynomial segments s_{i-1} , s_i , and s_{i+1} , i.e., the conditions

$$\begin{aligned} s'_{i-1}(t_i^m) &= s'_i(t_i^m), & s''_{i-1}(t_i^m) &= s''_i(t_i^m), \\ s'_i(t_{i+1}^m) &= s'_{i+1}(t_{i+1}^m), & s''_i(t_{i+1}^m) &= s''_{i+1}(t_{i+1}^m) \end{aligned}$$

can be used to eliminate the coefficient $\alpha_{i,3}$ and we obtain the resulting system

$$\begin{aligned} & \left[\begin{aligned} & \frac{-\Delta t_i^m}{(\Delta t_{i-1}^m + \Delta t_i^m) \Delta t_{i-1}^m}, \\ & \frac{8 \Delta t_i^m}{(\Delta t_{i-1}^m + \Delta t_i^m) \Delta t_{i-1}^m}, \\ & \frac{\Delta t_{i+1}^m}{(\Delta t_i^m + \Delta t_{i+1}^m) \Delta t_i^m} - \frac{7}{\Delta t_{i-1}^m} - \frac{5}{\Delta t_i^m}, \\ & \frac{8}{\Delta t_i^m} \left(3 - \frac{\Delta t_{i-1}^m}{\Delta t_{i-1}^m + \Delta t_i^m} - \frac{\Delta t_{i+1}^m}{\Delta t_i^m + \Delta t_{i+1}^m} \right), \\ & \frac{\Delta t_{i-1}^m}{(\Delta t_{i-1}^m + \Delta t_i^m) \Delta t_i^m} - \frac{7}{\Delta t_{i+1}^m} - \frac{5}{\Delta t_i^m}, \\ & \frac{8 \Delta t_i^m}{(\Delta t_i^m + \Delta t_{i+1}^m) \Delta t_{i+1}^m}, \\ & \frac{-\Delta t_i^m}{(\Delta t_i^m + \Delta t_{i+1}^m) \Delta t_{i+1}^m} \end{aligned} \right] [\mathbf{p}_{2i-2}^{m+1}, \dots, \mathbf{p}_{2i+4}^{m+1}]^T = 0 \end{aligned} \quad (20)$$

which we have to solve for the unknown \mathbf{p}_{2i+1}^{m+1} . In the first and the last polynomial segment, we replace the C^2 -conditions by appropriate boundary conditions. A natural choice is to require vanishing second order derivatives, $s''_0(t_0^m) = 0$. Again by elimination of $\alpha_{0,3}$ the first row of the linear system (9) becomes

$$\begin{aligned} & \left[\begin{aligned} & \frac{4}{(\Delta t_0^m)^2} + \frac{1}{\Delta t_0^m (\Delta t_0^m + \Delta t_1^m)}, \\ & - \left(\frac{8}{(\Delta t_0^m)^2} + \frac{8}{\Delta t_0^m (\Delta t_0^m + \Delta t_1^m)} \right), \\ & \frac{4}{(\Delta t_0^m)^2} + \frac{7}{\Delta t_0^m \Delta t_1^m}, \\ & \frac{-8}{(\Delta t_0^m + \Delta t_1^m) \Delta t_1^m}, \\ & \frac{1}{(\Delta t_0^m + \Delta t_1^m) \Delta t_1^m} \end{aligned} \right] [\mathbf{p}_0^{m+1}, \dots, \mathbf{p}_4^{m+1}]^T = 0. \end{aligned} \quad (21)$$

and the last row analogously.

Since the piecewise polynomial splines build a refinable space of functions, equations (20) and (21) hold for every m : By construction, the cubic spline which interpolates the values \mathbf{p}_i^{m+1} over the knot sequence $[t_i^{m+1}]$ is identical to the spline which interpolates \mathbf{p}_i^m over $[t_i^m]$. Consequently, the resulting limit curve will be the cubic spline. If we consider the special case of a uniform knot sequence $t_i^0 = i$, equation (20) reduces to (8).