Volume xxx, (1997) number yyy pp. 000–000

Multiresolution Curve and Surface Representation by Reversing Subdivision Rules

Faramarz F. Samavati† and Richard H. Bartels‡

†Sharif University of Technology
Mathematical Science Department
Tehran, Iran
‡University of Waterloo
Computer Science Department
Waterloo, Ontario N2L 3G1
Canada

Abstract

Subdivision curves and surfaces begin with a polygonal network of points and edges (and, for surfaces, faces) having a relatively coarse structure. Each type of curve and surface is defined by a set of subdivision rules that replace the coarse network by a finer network to which the rules could again be applied. The subdivision curve or surface is the limit of repeated application.

In this paper we explore the possibilities for reversing the subdivision process using least-squares techniques. We begin with a fine network that did not necessarily come from a set of subdivision rules. We wish to find an approximation to a coarser network that could have produced the finer network by the application of given subdivision rules. Since the finer network may not be produced exactly by the coarser network we find, we must also find the errors.

The approach taken borrows strongly from the framework of wavelets and multiresolution analysis with certain important differences. Firstly, we do not assume that we know either what scale functions or what wavelet functions might underlie our networks. We are allowed only to work with the coefficients of the scale functions, which we identify with the data points, and the coefficients of the wavelet functions, which we associate with the errors. Secondly, we assume that we have been given the data at the finest scale of resolution and we may only "coarsen" it. Our objective is to represent this data as a sequence of sets of error coefficients and a final network of points representing scale coefficients. The representation can subsequently be used for compression or for multiresolution display. Thirdly, since we begin and end in a finite setting, our use of the tools of wavelets will be centered around data fitting and discrete inner products rather than the usual focus of function approximation and continuous inner products. This results in new wavelets for B-splines, for example, that have smaller compact support and simpler representations than those previously used for multiresolution curves and surfaces.

Keywords: Subdivision rules; Multiresolution curves and surfaces; Wavelets; Least squares;

1. Introduction

Subdivision curves and surfaces begin with a polygonal network of points. In the case of curves, the network merely encodes the sequence of point-point edges. In the case of surfaces, the network provides the edges and faces of a polyhedral topology. In conventional usage a subdivision curve or surface begins with a coarse network structure consisting of relatively few points. To this is applied a set of rules that replace the coarse network by a finer network. The set of rules could again be applied to this finer network. In the limit of repeated application, the rules yield

curves or surfaces of some provable continuity (except at exceptional points). In practice, of course, the rules are repeated only a finite number of times to yield a network containing a large number of points that represent a fine approximate sampling of the limit.

In this paper we explore the possibilities for reversing the subdivision process. We begin with a fine network that did not necessarily come from a set of subdivision rules, and we wish to find an approximation to a coarser network that could have produced the finer network by the application of given subdivision rules. This implies that the coarse network would not necessarily return the fine network under the use of the subdivision rules. However, we can reconstruct the fine network with appropriate corrections derived from error terms. This will structure our approach along lines familiar to the users of wavelets. A fine set of data will be decomposed into a coarse approximation and error information. The fine data can subsequently be reconstructed from the coarse data and the error.

The framework of wavelets and multiresolution analysis is used; however, with certain important differences. We shall not assume that we know what scale functions underlie our data. Further, even if the scale functions were known, we would not necessarily want to find the conventional wavelets that are associated with them. We shall work only with the coefficients of the scale functions, which we identify with the data points, and the coefficients of the wavelet functions, which we associate with the errors.

We assume that we have been given the data at the finest scale of resolution; for example, from a scanning or ranging process. We are interested in decomposing this data into a multiresolution cascade of error coefficients and a final set of scale coefficients, and we are interested in doing this in the context of some given subdivision rule. Since we begin and end in a finite setting, our use of the tools of wavelets will be centered around data fitting and discrete inner products rather than the usual focus of function approximation and continuous inner products.

We shall give a general approach to achieving our goals for curves and tensor product surfaces for which the error terms represent the best least squares error. This approach will be particularly efficient when the subdivision rules are regular and repetitive.

In Section 2 we shall review basic matrix notation for subdivision rules. In Section 3 we shall review the connection of these matrices to wavelets. In Section 4 we introduce the least squares problem of interest, and in Section 5 we review the normal equations. In Section 6 we outline an approach to the construction of orthogonal complements for subdivision matrices. In Section 7 we use the special form of these orthogonal complements to solve for the error coefficients. Remarks on error are presented in Section 8. Several curve subdivision rules are presented in Section 9. Periodic rules are covered in Section 10. Tensor-product surfaces are covered in Section 11. We close by presenting some examples of our approach in Section 12.

We concentrate here on the mechanics for reversing subdivision rules using least-squares techniques. The end product of our work will be a multiresolution decomposition of given data into a sequence of error (detail) information and a base set of coarse data. We are presenting here only a technique that starts from suitable subdivision rules, costs linear time, and achieves the decomposition. Explorations of the use of the technique for compression or multi-level design and rendering are outside the intended scope of this work.

2. Subdivisions and Matrix Notation

The most compact introduction to wavelets and multiresolution analysis we can provide is [9]. Our notation will, however, be adapted from that of [5], a paper focusing upon the special case of cubic B-spline subdivision. For a more thorough introduction to B-spline wavelets, refer to [2].

We begin with *subdivision curves*. These start as a vector of *fine points*, c_j^k, that have been provided as data. To each of these is applied a *subdivision rule* to produce a new point

$$c_i^{k+1} = \sum_j p_{i,j}^{k+1} c_j^k \tag{1}$$

The totality of such rules yields a larger number of points. Thus, from a matrix point of view

$$\left[P^{k+1}\right]\left[C^k\right] = \left[C^{k+1}\right] \tag{2}$$

where the matrix P^{k+1} is $m \times n$ with $m > n$.

If we knew the matrix and the right hand side of equation (2), we would find the vector of *coarse points* C^k optimally by solving the equation in the best least squares sense. The result would satisfy the equation only if the fine points, C^{k+1}, were the result of subdividing the coarse points by the rules P^{k+1}. If the subdivision rules did not produce the fine points, the equation could be corrected as follows:

$$\begin{aligned}\left[P^{k+1}\ Q^{k+1}\right]\begin{bmatrix} C^k \\ E^k \end{bmatrix} &= \left[P^{k+1}\right]\left[C^k\right] \\ &+ \left[Q^{k+1}\right]\left[E^k\right] \\ &= \left[C^{k+1}\right]\end{aligned} \tag{3}$$

where the columns of the $m \times m-n$ matrix Q^{k+1} form

an *extension* of the column vectors of P^{k+1} to a basis for the m-dimensional vector space. Many extensions are possible; we need only adjoin linearly independent columns. However, the extension corresponding to the best least squares approximation requires the columns of Q^{k+1} to be orthogonal to the columns of P^{k+1}; that is, to be a basis for the null space of the transpose of P^{k+1}. For any extension Q^{k+1}, the vector $Q^{k+1}E^k$ will express the *error correction* $C^{k+1} - P^{k+1}C^k$. If Q^{k+1} is an orthogonal extension, the error correction expresses the best least squares error.

Tangentially, some further observations can be made about subdivision schemes from this matrix view. The matrix P^{k+1} defines the subdivision rules as transformations on the points C^k. In order to be geometrically meaningful [6], these transformations must produce *affine combinations* of the C^k, which implies that all row sums of P^{k+1} are 1. The matrix Q^{k+1} must define a *vector combination* of the E^k, since the second term in the sum appearing in equation 4 constitutes a displacement of the points represented by the first term.

Finally, note that the main requirements on Q^{k+1} is that it have full column rank and satisfy orthogonality, ${Q^{k+1}}^T P^{k+1} = 0$. These requirements can only specify Q^{k+1} up to an arbitrary multiple. Each column of Q^{k+1} can be scaled individually as we choose. This scaling, of course, influences the magnitude of the error coefficients, E^k. This, in turn, influences our judgment on whether to suppress an error coefficient for the purposes of compression. We shall come back to this point in Section 8.

3. Wavelet Connections

The points c_j^κ are interpreted as coefficients of a *basis* of functions ϕ_j^κ for a space V^κ ($\kappa = k, k+1$). Furthermore, $V^k \subset V^{k+1}$. The basis ϕ_j^k for V^k can be completed to a basis for V^{k+1} by adding independent functions ψ_ℓ^k. The functions ϕ are known as *scale functions*, and the functions ψ are called *wavelets* or *wavelet functions*. Taken all together, $V^\kappa, \phi^\kappa, \psi^k$ for $\kappa = k, k+1$, if repeated for a succession $k = 1, 2, 3, \ldots$, form a *multiresolution system*. With respect to an inner product, if the ϕ and the ψ are mutually orthogonal ($<\phi_i^\kappa, \phi_j^\kappa> = \delta_{i,j}$ ($\kappa = k, k+1$), $<\phi_i^k, \psi_\ell^k> = 0$, $<\psi_\ell^k, \psi_r^k> = \delta_{\ell,r}$) the multiresolution system is said to be *orthogonal*. If only $<\phi_i^k, \psi_\ell^k> = 0$ holds, the system is *semiorthogonal*. A third category of system is *biorthogonal* (refer to [9] for a definition). Biorthogonal systems will not concern us here.

Since the spaces are nested, we have

$$\phi_j^k = \sum_i p_{i,j}^{k+1} \phi_i^{k+1} \tag{4}$$

$$\psi_\ell^k = \sum_i q_{i,\ell}^{k+1} \phi_i^{k+1} \tag{5}$$

and conversely

$$\phi_i^{k+1} = \sum_j a_{j,i}^{k+1} \phi_j^k + \sum_\ell b_{\ell,i}^{k+1} \psi_\ell^k \tag{6}$$

for some coefficients $a_{j,i}^{k+1}, b_{\ell,i}^{k+1}, p_{i,j}^{k+1}, q_{i,\ell}^{k+1}$. Thus, for any element of V^{k+1} we can write

$$\begin{aligned} \sum_i c_i^{k+1} \phi_i^{k+1} &= \sum_i c_i^{k+1} \left(\sum_j a_{j,i}^{k+1} \phi_j^k + \sum_\ell b_{\ell,i}^{k+1} \psi_\ell^k \right) \\ &= \sum_j \left(\sum_i c_i^{k+1} a_{j,i}^{k+1} \right) \phi_j^k \\ &\quad + \sum_\ell \left(\sum_i c_i^{k+1} b_{\ell,i}^{k+1} \right) \psi_\ell^k \\ &= \sum_j c_j^k \phi_j^k + \sum_\ell e_\ell^k \psi_\ell^k \end{aligned} \tag{7}$$

This provides the *decomposition* process

$$c_j^k = \sum_i c_i^{k+1} a_{j,i}^{k+1} \tag{8}$$

$$e_\ell^k = \sum_i c_i^{k+1} b_{\ell,i}^{k+1} \tag{9}$$

whereby $\sum_j c_j^k \phi_j^k$ is the approximation of $\sum_i c_i^{k+1} \phi_i^{k+1}$ in V^k and $\sum_i c_i^{k+1} b_{\ell,i}^{k+1}$ is the corresponding error in $W^k = V^{k+1} \setminus V^k$. The c are the *scale coefficients*, and the e are the *error* (or *detail*) *coefficients*. In matrix terms,

$$\begin{bmatrix} A^{k+1} \\ B^{k+1} \end{bmatrix} \begin{bmatrix} C^{k+1} \end{bmatrix} = \begin{bmatrix} C^k \\ E^k \end{bmatrix} \tag{10}$$

Conversely,

$$\begin{aligned} &\sum_j c_j^k \phi_j^k + \sum_\ell e_\ell^k \psi_\ell^k \\ &\quad = \sum_j c_j^k \left(\sum_i p_{i,j}^{k+1} \phi_i^{k+1} \right) \\ &\quad + \sum_\ell e_\ell^k \left(\sum_i q_{i,\ell}^{k+1} \phi_i^{k+1} \right) \\ &\quad = \sum_i \left(\sum_j c_j^k p_{i,j}^{k+1} + \sum_\ell e_\ell^k q_{i,\ell}^{k+1} \right) \phi_i^{k+1} \\ &\quad = \sum_i c_i^{k+1} \phi_i^{k+1} \end{aligned} \tag{11}$$

This provides the *reconstruction* process

$$c_i^{k+1} = \sum_j c_j^k p_{i,j}^{k+1} + \sum_\ell e_\ell^k q_{i,\ell}^{k+1} \tag{12}$$

which in matrix format is simply equation (4).

The equations above lead to a matrix view of what constitutes a semiorthogonal system:

$$\begin{aligned} 0 &= \left\langle \phi_i^k, \psi_j^k \right\rangle = \left\langle \sum_r p_{r,i}^{k+1} \phi_r^{k+1}, \sum_s q_{s,j}^{k+1} \phi_s^{k+1} \right\rangle \\ &= \sum_r \sum_s p_{r,i}^{k+1} q_{s,j}^{k+1} \left\langle \phi_r^{k+1}, \phi_s^{k+1} \right\rangle \\ &= \left[P^{k+1\,T} \right] \left[\Phi^{k+1} \right] \left[Q^{k+1} \right] \end{aligned} \tag{13}$$

where Φ^{k+1} is the *Gram matrix* formed by the scale functions for V^{k+1} and the inner product $<\cdot,\cdot>$. Matrix equations for orthogonal multiresolution systems can be obtained similarly.

4. Least Squares

Assume that a subdivision rule, P^{k+1}, and a set of fine points (not necessarily generated by that rule), C^{k+1}, have been specified. Then the following hold:

1. C^k provides the best least squares solution to

$$\left[P^{k+1} \right] \left[C^k \right] = \left[C^{k+1} \right] \tag{14}$$

if and only if C^k satisfies the *normal equations*:

$$\left[P^{k+1\,T} P^{k+1} \right] \left[C^k \right] = \left[P^{k+1\,T} \right] \left[C^{k+1} \right] \tag{15}$$

2. If C^k is produced as in equation (15) and if the columns of Q^{k+1} form a basis for the nullspace of $P^{k+1\,T}$, then the equations

$$\left[Q^{k+1} \right] \left[E^k \right] = \left[C^{k+1} \right] - \left[P^{k+1} \right] \left[C^k \right] \tag{16}$$

are compatible.

3. The overdetermined system (16) may be solved in the least squares sense via the normal equations

$$\begin{aligned} &\left[Q^{k+1} \right]^T \left[Q^{k+1} \right] \left[E^k \right] \\ &\quad = \left[Q^{k+1} \right]^T \left(\left[C^{k+1} \right] - \left[P^{k+1} \right] \left[C^k \right] \right) \end{aligned} \tag{17}$$

In view of the compatibility of (16), however, the solution will exactly satisfy the equation system, and this solution can be obtained equivalently from any subselection of the rows of Q^{k+1} that forms a nonsingular matrix, $\hat{Q}^{k+1}$, with a corresponding selection of the elements of the residual vector $C^{k+1} - P^{k+1} C^k$.

Another observation is also important:

4. Subdivision matrices P^{k+1} are usually banded and characterized by columns that are shifted versions of a single column (except for a few initial and final columns).

cubic B-spline subdivision gives a good example [5]:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \frac{3}{4} & \frac{1}{4} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \frac{3}{16} & \frac{11}{16} & \frac{1}{8} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{8} & \frac{11}{16} & \frac{3}{16} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{4} & \frac{3}{4} & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{18}$$

5. Normal Equations

The *normal-equation matrix* $P^{k+1\,T} P^{k+1}$, see equation (15), is symmetric and positive definite. This, together with the form of typical subdivision matrices (simple entries, row sums equal to one, largest absolute values at the center of each column's nonzero entries with values decreasing strongly in the rows above and below), implies that the matrix can be found explicitly, that it is banded, diagonally dominant, highly repetitive, has simple entries, and may be solved efficiently and accurately via simple *Gaussian elimination*, via the corresponding (LU) factorization, via a *Cholesky* factorization, or via an LDL^T factorization. No row/column rearrangement is needed in any of these approaches [7].

Other methods of solving the least squares problem (14) that avoid the normal equations entirely are by way of the QR factorization of P^{k+1} or via the corresponding elimination process, which is based upon elementary orthogonal transformations such as those due to *Householder*. Any of these factorizations or elimination processes will cost an amount of work that is linear in the size of C^{k+1}. There are various trade-offs. In the context of curves and tensor-product surfaces, the use of the LDL^T factorization is advised for reasons to be given below.

Again, cubic B-spline subdivision provides a conve-

nient example for a normal-equation matrix:

$$\begin{bmatrix}
\frac{5}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\frac{1}{4} & \frac{217}{256} & \frac{81}{256} & \frac{3}{128} & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & \frac{81}{256} & \frac{205}{256} & \frac{55}{128} & \frac{1}{64} & 0 & 0 & \cdots & 0 & 0 \\
0 & \frac{3}{128} & \frac{55}{128} & \frac{35}{32} & \frac{7}{16} & \frac{1}{64} & 0 & \cdots & 0 & 0 \\
0 & 0 & \frac{1}{64} & \frac{7}{16} & \frac{35}{32} & \frac{7}{16} & \frac{1}{64} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \frac{1}{64} & \frac{7}{16} & \frac{35}{32} & \frac{7}{16} & \frac{1}{64} & 0 & 0 \\
0 & 0 & \cdots & 0 & \frac{1}{64} & \frac{7}{16} & \frac{35}{32} & \frac{55}{128} & \frac{3}{128} & 0 \\
0 & 0 & \cdots & 0 & 0 & \frac{1}{64} & \frac{55}{128} & \frac{205}{256} & \frac{81}{256} & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \frac{3}{128} & \frac{81}{256} & \frac{217}{256} & \frac{1}{4} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{5}{4}
\end{bmatrix} \qquad (19)$$

More specifically, the subdivision rules of interest to us are those which have most columns in a standard form containing μ nonzero elements that begin at row λ for the first occurrence of these standard columns and are shifted by τ rows for each successive column:

$$\pi_{\lambda+i\tau}, \ldots, \pi_{\lambda+i\tau+\mu-1} \text{ for i} = 0, 1, \ldots \qquad (20)$$

The matrix is allowed to have a few initial and final columns with μ or less nonzero elements each, where the elements have different values from and/or are not shifted by the same amounts as the standard columns. For the matrix of equation (18), for instance, $\lambda = 4, \tau = 2, \mu = 5$, and all but the first three and last three columns are of standard form.

Since the (i, j) element of the normal-equation matrix is given by the inner product of columns i and j of the matrix P^{k+1}, equation (20) implies that most such elements are

$$\sum_{\gamma=0}^{\mu-1} \pi_{\lambda+i\tau+\gamma} \pi_{\lambda+j\tau+\gamma} \qquad (21)$$

And in turn, for a standard row i of the normal-equation matrix, the element is zero unless

$$j = i - \left\lfloor \frac{\mu-1}{\tau} \right\rfloor, \ldots, i + \left\lfloor \frac{\mu-1}{\tau} \right\rfloor \qquad (22)$$

This means that, except for a certain number of initial and final rows, each row of the normal-equation matrix will consist of exactly the same $2\left\lfloor \frac{\mu-1}{\tau} \right\rfloor + 1$ numbers given by equation (21).

All entries of the normal-equation matrix can be obtained in advance for the subdivision rules under consideration, so we do not have to count their cost. The process of computing C^k will require applying ${P^{k+1}}^T$ to C^{k+1} to produce an intermediate vector G^{k+1}. This process looks like the application of a finite filter and involves μ multiplications and additions for each element of G^{k+1} (except a few at the beginning and end):

$$g_i^{k+1} = \sum_{\gamma=\lambda+i\tau}^{\lambda+i\tau+\mu-1} \pi_\gamma c_\gamma^{k+1} \text{ for i} = 0, 1, \ldots \qquad (23)$$

The total effort will be $O(2m\mu)$ elementary floating point operations. Since μ is typically small compared to m, at least in the initial stages of recursive decompositions of C^{k+1} into C^k and E^k, the bandwidth of the normal-equation matrix will be small compared to its size, which is exactly the situation for which an LDL^T factorization is advised [7]. The use of this factorization to solve a banded system, with right-hand side given by G^{k+1}, band width given by $\left\lfloor \frac{\mu-1}{\tau} \right\rfloor$, and band elements given by equation (21), will cost $O\left(n(\left\lfloor \frac{\mu-1}{\tau} \right\rfloor^2 + 8\left\lfloor \frac{\mu-1}{\tau} \right\rfloor + 1)\right)$ elementary floating point operations. Thus, overall, finding C^k will incur a cost that is linear in m; that is, linear in the size of C^{k+1}. (This counts both the cost of producing the factors L and D as well as the forward- and back-solution processes. One could save a considerable portion of this, at the cost of $n \times \left\lfloor \frac{\mu-1}{\tau} \right\rfloor$ storage, by computing the factors in advance.)

6. Orthogonal Complements

The second part of the process is to find the elements of E^k whereby the error $C^{k+1} - P^{k+1}C^k$ may be expressed. For this we must extend the matrix P^{k+1} by adding linearly-independent columns Q^{k+1} such that ${Q^{k+1}}^T P^{k+1} = 0$. As will be seen, there is an additional bonus in constructing the columns of Q^{k+1} to have a regular pattern of shifted columns with as few nonzero entries as possible, imitating the structure of P^{k+1} as well as that of wavelets with shifted structure and minimal support.

The essential trick in finding such a matrix column is to concentrate on the general, shifted column of P^{k+1}. If the order of the nonzero elements of such a column is reversed and the signs of these elements are alternated, then any column containing the resulting nonzero entries will be orthogonal to every column of P^{k+1} with which it overlaps in an even number of row positions (i.e. $0, 2, 4, \ldots$). The cancellation pattern that results can easily be seen from the following

example:

$$\begin{array}{cc} \underline{P^{k+1}} & \underline{Q^{k+1}} \\ \vdots & \vdots \\ 0 & \vdots \\ \pi_{\lambda+i\tau} & \pi_{\lambda+i\tau+4} \\ \pi_{\lambda+i\tau+1} & -\pi_{\lambda+i\tau+3} \\ \pi_{\lambda+i\tau+2} & \pi_{\lambda+i\tau+2} \\ \pi_{\lambda+i\tau+3} & -\pi_{\lambda+i\tau+1} \\ \pi_{\lambda+i\tau+4} & \pi_{\lambda+i\tau} \\ \vdots & 0 \\ \vdots & \vdots \end{array} \tag{24}$$

Since most common subdivision rules essentially double the number of points, reflecting an underlying *two-scale relationship* between the scale functions at level k and those at level $k+1$, the usual shift amount, τ, is 2. This means that a column of Q^{k+1} that overlaps one of the general columns of P^{k+1} by an even number of row positions will overlap all the other general columns by an even number of row positions. The few special columns at the beginning and end of P^{k+1} must be handled separately. We have used Maple [8] successfully to solve orthogonality equations for these columns in the subdivision rules that we have investigated.

Using these observations, we arrive at the following extension for the cubic B-spline subdivision matrix of equation (18):

$$\begin{bmatrix} -\frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{3}{4} & \frac{1}{8} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{3} & -\frac{1}{2} & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{12} & \frac{3}{4} & \frac{1}{8} & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & \cdots & 0 & 0 & \frac{1}{8} & \frac{3}{4} & -\frac{1}{12} \\ 0 & 0 & \cdots & 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{3} \\ 0 & 0 & \cdots & 0 & 0 & 0 & \frac{1}{8} & -\frac{3}{4} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & -\frac{1}{2} \end{bmatrix} \tag{25}$$

Note that the subdivision matrix of equation (18) is the same one used by Finkelstein and Salesin in [5]. However, our matrix Q^{k+1} is significantly simpler than the one presented in that paper. More surprisingly, since the columns of Q^{k+1} provide the representation of the level-k wavelets, ψ_ℓ^k, in terms of the scale functions at level $k+1$, ϕ_i^{k+1}, and since the scale functions in question for this example are cubic B-splines, we see that (25) defines wavelets that (except for the special ones at the extremes of the domain) have the same support as the level-k scale functions. Since Chui [2] proves that the minimal-support B-spline wavelet must have support that is essentially *twice* that of the B-splines at the corresponding level, we have some explaining to do.

The reason for the unusually compact wavelets we are defining lies in the form of semiorthogonality we are using. Since we do not assume that we know the underlying scale functions, we have constructed Q^{k+1} to satisfy

$$\left[{P^{k+1}}^T\right]\left[Q^{k+1}\right] = 0 \tag{26}$$

which agrees with equation (14) only if the Gram matrix Φ^{k+1} is the identity. This, in turn, implies that we must be using a different inner product than the usual one. In [2], as in most literature on wavelets, the inner product used is

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(z)\bar{g}(z)dz \tag{27}$$

In our case, the inner product is defined implicitly to yield the following relationships:

$$\left\langle \phi_i^{k+1}, \phi_j^{k+1}\right\rangle_{k+1} = \delta_{i,j} \tag{28}$$

which defines the inner product on V^{k+1} by

$$\begin{aligned} \langle f, g\rangle_{k+1} &= \sum_\gamma f_\gamma g_\gamma \\ \text{where} & \\ f &= \sum_\gamma f_\gamma \phi_\gamma^{k+1} \\ g &= \sum_\gamma g_\gamma \phi_\gamma^{k+1} \end{aligned} \tag{29}$$

The inner product is flagged with "$k+1$" because it is a different inner product for each space V^{k+1} in the nested spaces of the multiresolution analysis.

This inner product is, in fact, the inner product that is conventionally used for discrete, finite-dimensional data fitting. This is certainly a reasonable inner product to use in our case. The alternative of using (27) is consistent with the view that the components of C^{k+1} are *coefficients* for the basis elements $\left\{\phi_i^{k+1}\right\}$ of a function space. Our view, however, must be that the components of C^{k+1} are *points*; that is, the *sampled values* of some unknown function, since they are assumed to be provided by some form of measuring process.

7. Solving for the Error Coefficients

The system (16) is an overdetermined, yet consistent, system of equations. This means that any selection of the rows of system (16) that produces a nonsingular submatrix of Q^{k+1} may be solved for E^k. The result will not depend on which selection of rows is taken.

The subdivision rules of primary interest to us are those for which the matrix Q^{k+1} has a selection of rows that yield a triangular submatrix, for then the vector E^k can be obtained without any significant further matrix factorization. So far, this has been true of *all* subdivision rules we have looked at.

The selection of rows of Q^{k+1}, just as the construction of Q^{k+1} and the construction of ${P^{k+1}}^T P^{k+1}$, can be made in advance of having any sets of data, and so accounts for no computational cost.

An appropriate row-selection from the matrix of (25), for example, would be the matrix $\hat{Q}^{k+1}$ given by:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{3} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

This matrix represents the selection of the second row of Q^{k+1}, followed by rows $6, 8, 10, \ldots$ and ending with the next to last row. Correspondingly, elements $2, 6, 8, 10, \ldots, m-1$ of the residual vector must be chosen to constitute the equations that must be solved for E^k.

Solving for the vector E^k can be done in linear time, and the back-solution computation looks essentially like the application of a finite filter of length 2 to the vector subselected from $C^{k+1} - P^{k+1}C^k$. This is a fitting place to remark that, while both the decomposition and reconstruction processes we are describing have linear cost, the decomposition process is the more expensive. Reconstruction involves only the processing of the vector C^k by the rows of P^{k+1}; that is, by applying a filter of at most length 3 in our example, and processing the vector E^k by the rows of Q^{k+1}, again a filter of at most length 3 in our example. In common applications (e.g. compression, multi-level rendering), fast reconstruction is desirable.

8. Remarks on Error

A frequent use of multiresolution representations is for the purpose of compression. In this use, the original data C^{k+1} is represented as a telescoping series

$$\begin{aligned} C^{k+1} &= P^{k+1}C^k + Q^{k+1}E^k \\ &= P^{k+1}\left(P^kC^{k-1} + Q^kE^{k-1}\right) + Q^{k+1}E^k \\ &\cdots \end{aligned} \tag{31}$$

The information (for some N) represented by C^{k-N} and $E^{k-N}, \ldots, E^k$ is stored instead of C^{k+1}, and compression is achieved by discarding bits from the sequence $\left\{E^{k-\lambda}\right\}$.

If $E^{k-\lambda}$ is replaced by $E^{k-\lambda} + \Delta^{k-\lambda}$, it is easily seen that C^{k+1} is changed by

$$P^{k+1} \ldots P^{k-\lambda+2} Q^{k-\lambda+1} \Delta^{k-\lambda} \tag{32}$$

This change is bounded by

$$\| P^{k+1} \| \cdots \| P^{k-\lambda+2} \| \| Q^{k-\lambda+1} \| \| \Delta^{k-\lambda} \| \tag{33}$$

In particular, if the infinity matrix and vector norms are used (maximum absolute row sum of a matrix and maximum absolute element of a vector [7]), and if we arrange to scale the columns of the Q matrices so that $\| Q^{k-i} \| = 1$, then magnitude of the change to any component of C^{k+1} will be no larger than the maximum magnitude of change made to any component of the E vectors, in view of the fact that the infinity norm of P^{k+1} is 1 (the rows of P^{k+1} represent affine combinations).

9. Curve Subdivision Rules

We have experimented with four common curve subdivision rules: the cubic B-spline rule that we have used as an example above, *Faber subdivision* [4], *Chaikin subdivision* [1], and *Dyn-Levin-Gregory subdivision* [3] (considered in Section 10).

The subdivision matrix, P^{k+1}, of the Chaikin rule is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{34}$$

the Chaikin normal-equation matrix is:

$$\begin{bmatrix} \frac{14}{9} & \frac{4}{9} & 0 & \cdots & 0 & 0 & 0 \\ \frac{4}{9} & \frac{10}{9} & \frac{4}{9} & \cdots & 0 & 0 & 0 \\ 0 & \frac{4}{9} & \frac{10}{9} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{10}{9} & \frac{4}{9} & 0 \\ 0 & 0 & 0 & \cdots & \frac{4}{9} & \frac{10}{9} & \frac{4}{9} \\ 0 & 0 & 0 & \cdots & 0 & \frac{4}{9} & \frac{14}{9} \end{bmatrix} \tag{35}$$

a suitable extension, Q^{k+1}, is:

$$\begin{bmatrix} \frac{1}{3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\frac{2}{3} & \frac{1}{3} & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{3} & -\frac{2}{3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & \cdots & 0 & 0 & 0 \\ 0 & -\frac{1}{3} & -\frac{2}{3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{1}{3} & -\frac{2}{3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \cdots & 0 & -\frac{1}{3} & -\frac{2}{3} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{3} \end{bmatrix} \tag{36}$$

and a possible submatrix, $\hat{Q}^{k+1}$, is:

$$\begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{3} \end{bmatrix} \tag{37}$$

formed by selecting row 1 and every even row beginning with 4.

The subdivision matrix of the Faber rule is:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{38}$$

the normal-equation matrix of the Faber rule is:

$$\begin{bmatrix} \frac{5}{4} & \frac{1}{4} & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{4} & \frac{3}{2} & \frac{1}{4} & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{3}{2} & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & \cdots & \frac{3}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \cdots & \frac{1}{4} & \frac{3}{2} & \frac{1}{4} \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{4} & \frac{5}{4} \end{bmatrix} \tag{39}$$

a suitable orthogonal extension is:

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{2} \end{bmatrix} \tag{40}$$

and a possible submatrix, $\hat{Q}^{k+1}$, is simply the negative of the identity, corresponding to selecting the even rows of (40) starting at the second.

10. Periodic Subdivision Rules

The discussion so far has concentrated on *open* curves. Dyn-Levin-Gregory subdivision is given as a *periodic* rule to be applied to *closed* curves. Such curves are usually provided by subdivision rules whose P matrices have rows/columns that wrap around *periodically*; for example, the Dyn-Levin-Gregory matrix (with the

authors' parameter w set to $\frac{1}{16}$) is:

$$\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
-\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\
-\frac{1}{16} & 0 & 0 & 0 & 0 & \cdots & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\
\frac{9}{16} & -\frac{1}{16} & 0 & 0 & 0 & \cdots & 0 & -\frac{1}{16} & \frac{9}{16} \\
1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & \cdots & 0 & 0 & -\frac{1}{16}
\end{bmatrix} \tag{41}$$

and the closed-curve version of the Chaiken rule is:

$$\begin{bmatrix}
\frac{2}{3} & \frac{1}{3} & 0 & 0 & \cdots & 0 & 0 \\
\frac{1}{3} & \frac{2}{3} & 0 & 0 & \cdots & 0 & 0 \\
0 & \frac{2}{3} & \frac{1}{3} & 0 & \cdots & 0 & 0 \\
0 & \frac{1}{3} & \frac{2}{3} & 0 & \cdots & 0 & 0 \\
0 & 0 & \frac{2}{3} & \frac{1}{3} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & \frac{2}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \cdots & \frac{1}{3} & \frac{2}{3} \\
\frac{1}{3} & 0 & 0 & 0 & \cdots & 0 & \frac{2}{3} \\
\frac{2}{3} & 0 & 0 & 0 & \cdots & 0 & \frac{1}{3}
\end{bmatrix} \tag{42}$$

The normal-equation matrix is also cyclic. For example, the Dyn-Levin-Gregory normal-equation matrix is:

$$\begin{bmatrix}
\delta & \gamma & -\beta & \alpha & 0 & 0 & 0 & 0 & \alpha & -\beta & \gamma \\
\gamma & \delta & \gamma & -\beta & \alpha & 0 & 0 & 0 & 0 & \alpha & -\beta \\
-\beta & \gamma & \delta & \gamma & -\beta & \alpha & 0 & 0 & 0 & 0 & \alpha \\
\alpha & -\beta & \gamma & \delta & \gamma & -\beta & \alpha & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \alpha & -\beta & \gamma & \delta & \gamma & -\beta & \alpha \\
\alpha & 0 & 0 & 0 & 0 & \alpha & -\beta & \gamma & \delta & \gamma & -\beta \\
-\beta & \alpha & 0 & 0 & 0 & 0 & \alpha & -\beta & \gamma & \delta & \gamma \\
\gamma & -\beta & \alpha & 0 & 0 & 0 & 0 & \alpha & -\beta & \gamma & \delta
\end{bmatrix} \tag{43}$$

where $\alpha = \frac{1}{256}$, $\beta = \frac{9}{128}$, $\gamma = \frac{63}{256}$, and $\delta = \frac{105}{64}$. Such matrices can be broken into LDL^T factors as in the case of open curves. The factors do not have the banded structure of the open-curve normal-equation matrices, but their structure does contain only a small, fixed number of nonzero entries for each row, in regular positions, which is just as good as a purely banded structure. The profile of the upper triangular factor of (43), for example, is:

$$\begin{bmatrix}
X & X & X & X & 0 & 0 & 0 & \cdots & 0 & X & X & X \\
0 & X & X & X & X & 0 & 0 & \cdots & 0 & X & X & X \\
0 & 0 & X & X & X & X & 0 & \cdots & 0 & X & X & X \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & X & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & X & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & X & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & X & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & X & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & X & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & X
\end{bmatrix} \tag{44}$$

The general row in (44) has four entries in a band along the diagonal and three entries in the last three columns. Consequently, the generation of these numbers and the backsolution process costs linear time. This is characteristic of the case with periodic P matrices: their corresponding normal-equation matrices are periodic and essentially banded in the style of (43), while their LDL^T factors are banded with an additional number of final columns that are full.

The Q-extension of a periodic matrix is also easy to find. There are no special columns, so the sign-alternation and shifting for alignment of even numbers of elements, as explained in Section 6, can be carried out easily (and periodically). The orthogonal extension of (41), for example, is:

$$\begin{bmatrix}
\frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & \cdots & 0 & 0 & -\frac{1}{16} \\
0 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
-\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \\
-\frac{1}{16} & 0 & 0 & 0 & 0 & \cdots & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \\
\frac{9}{16} & -\frac{1}{16} & 0 & 0 & 0 & \cdots & 0 & -\frac{1}{16} & \frac{9}{16} \\
-1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0
\end{bmatrix} \tag{45}$$

The best subselection of rows to take from this matrix, of course, consists of all rows having only the entry -1.

Not all subselections are quite so convenient. The periodic, cubic B-spline P matrix is:

$$\begin{bmatrix} \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{8} & \frac{3}{4} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{8} & 0 & 0 & 0 & \cdots & 0 & \frac{1}{8} & \frac{3}{4} \\ \frac{1}{2} & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{8} & 0 & 0 & \cdots & 0 & 0 & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \tag{46}$$

The Q-extension is:

$$\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & \cdots & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \cdots & 0 & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{8} & 0 & 0 & 0 & \cdots & 0 & \frac{1}{8} & \frac{3}{4} \\ -\frac{1}{2} & 0 & 0 & 0 & \cdots & 0 & 0 & -\frac{1}{2} \\ \frac{3}{4} & \frac{1}{8} & 0 & 0 & \cdots & 0 & 0 & \frac{1}{8} \end{bmatrix} \tag{47}$$

No subselected matrix $\hat{Q}$ is triangular, However, the following matrix is almost triangular (only two Gaussian elimination steps are needed to triangularize it):

$$\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -\frac{1}{2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & 0 & 0 & \cdots & 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \tag{48}$$

11. Tensor-Product Surfaces

For a tensor-product surface, the subdivision process given by equation (2) becomes

$$\left[P_L^{k+1}\right]\left[C^k\right]\left[P_R^{k+1\,T}\right] = \left[C^{k+1}\right] \tag{49}$$

where P_L^{k+1} is $m_L \times n_L$, P_R^{k+1} is $m_R \times n_R$, C^k is $n_L \times n_R$, and C^{k+1} is $m_L \times m_R$. Completing each P matrix by a corresponding Q matrix yields

$$\left[P_L^{k+1} Q_L^{k+1}\right] \begin{bmatrix} C^k & F^k \\ G^k & H^k \end{bmatrix} \begin{bmatrix} P_R^{k+1\,T} \\ Q_R^{k+1\,T} \end{bmatrix} \tag{50}$$

where Q_L^{k+1} is $m_L \times m_L - n_L$ and Q_R^{k+1} is $m_R \times m_R - n_R$. Accordingly, the relationship between coarse and fine levels is composed of four terms, corresponding to the terms found in the literature for tensor-product wavelets so that:

$$\begin{aligned} \left[C^{k+1}\right] = & \left[P_L^{k+1}\right]\left[C^k\right]\left[P_R^{k+1\,T}\right] \\ & + \left[Q_L^{k+1}\right]\left[G^k\right]\left[P_R^{k+1\,T}\right] \\ & + \left[P_L^{k+1}\right]\left[F^k\right]\left[Q_R^{k+1\,T}\right] \\ & + \left[Q_L^{k+1}\right]\left[H^k\right]\left[Q_R^{k+1\,T}\right] \end{aligned} \tag{51}$$

Finding coarse versions of C^{k+1} may proceed with some flexibility. A *left coarse version* is produced by solving

$$\left(\left[P_L^{k+1}\right]^T\left[P_L^{k+1}\right]\right)\left[C_L^k\right] = \left[P_L^{k+1}\right]^T\left[C^{k+1}\right] \tag{52}$$

column by column for C_L^k. The corresponding error is given by:

$$\begin{aligned} & \left(\left[Q_L^{k+1}\right]^T\left[Q_L^{k+1}\right]\right)\left[E_L^k\right] \\ & \quad = \left[Q_L^{k+1}\right]^T\left(\left[C^{k+1}\right] - \left[P_L^{k+1}\right]\left[C_L^k\right]\right) \end{aligned} \tag{53}$$

which may be found by solving

$$\left[\hat{Q}_L^{k+1}\right]\left[E_L^k\right] = \left[C^{k+1}\right] - \left[P_L^{k+1}\right]\left[C_L^k\right] \tag{54}$$

where $\hat{Q}_L^{k+1}$ is the submatrix selected from Q_L^{k+1} as described in Section 7. In terms of the four tensor-product wavelet terms

$$\begin{aligned} C_L^k &= \left[C^k\right]\left[P_R^{k+1\,T}\right] + \left[F^k\right]\left[Q_R^{k+1\,T}\right] \\ E_L^k &= \left[G^k\right]\left[P_R^{k+1\,T}\right] + \left[H^k\right]\left[Q_R^{k+1\,T}\right] \end{aligned} \tag{55}$$

Similarly a *right coarse version* of C^{k+1} may be found by solving

$$\left[C_R^k\right]\left(\left[P_R^{k+1}\right]^T\left[P_R^{k+1}\right]\right) = \left[C^{k+1}\right]\left[P_R^{k+1}\right] \tag{56}$$

row by row. The corresponding error, E_R^k, may be found in a way corresponding to E_L^k. The *fully coarse version* of C^{k+1}, C^k, is obtained by finding the left coarse version of C_R^k or the right coarse version of C_L^k.

12. Examples

Several examples are offered here. Figure 1 shows a free-hand curve that is made coarse through several

levels by the reverse of cubic B-spline subdivision and then reconstructed. The reconstruction "without wavelets" was made by ignoring all error terms, $Q^{k+1}E^k$, and using only scale terms, $P^{k+1}C^k$.

Figure 2 shows a grey-scale image of a fox. Figure 3 shows the same image after two levels of being made coarse through the reverse of tensor-product, cubic B-spline subdivision. Figure 4 is the full reconstruction of the image, and Figure 5 is the reconstruction without wavelets (i.e., without including the error terms).

Figure 6 shows data taken from range data of a bust of Victor Hugo. Since we have been taking subdivision rules in as simple a form as possible, we have had to reduce this data somewhat. A typical subdivision rule will transform m_k points into m_{k+1} points. If we are given data that is not exactly m_{k+1} points, we must adjust the number somehow. This is a typical problem familiar from wavelet and FFT decompositions. We have simply reduced the Victor Hugo data by removing all rows from the top of the head downward as needed, and by deleting every other column from the back of the head (where detail is slight and so much data unnecessary.) Thus the data covers the front of the bust with twice the density of that in the back. Figure 7 shows the result of coarse approximation in one direction only via tensor-product, cubic B-spline subdivision. Figure 8 shows coarse approximation for two levels in two directions. Figure 9 is the reconstructed original data fully reconstructed from the surface of Figure 8, and Figure 10 shows a reconstruction that did not include the error (wavelet) terms.

Acknowledgments

The authors are indebted to Guenther Greiner of the University of Erlangen in Germany for very enlightening discussions at the inception of this work. The second author thanks him and Hans-Peter Seidel for their kind hospitality in Erlangen during these discussions. We also wish to thank Alexander Nicolaou for insightful comments and questions during the later stages of development. The collaboration on this work took place while the first author was visiting the University of Waterloo. His visit was enabled through arrangements made by Prof. Nezam Mahdavi-Amiri of Sharif University of Technology. We are especially grateful to Prof. Mahdavi-Amiri for this. This work has been supported by NATO, by the National Science and Engineering Research Council of Canada, and by the Sharif University of Technology.

References

1. G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.
2. C. K. Chui. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications*. Academic Press, Inc., 1992.
3. N. Dyn, D. Levin, and J. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4(4):257–268, 1987.
4. G. Faber. Über stetige functionen. *Mathematische Annalen*, 66:81–94, 1909.
5. Adam Finkelstein and David H. Salesin. Multiresolution curves. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 261–268. ACM SIGGRAPH, ACM Press, July 1994.
6. R. Goldman. Illicit expressions in vector algebra. *ACM Transactions on Graphics*, 4(3):223–243, 1985.
7. G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1989.
8. K. M. Heal, M. L. Hansen, and K. M. Rickard. *Maple V Learning Guide*. Springer Verlag, 1996.
9. Peter Schröder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 161–172. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

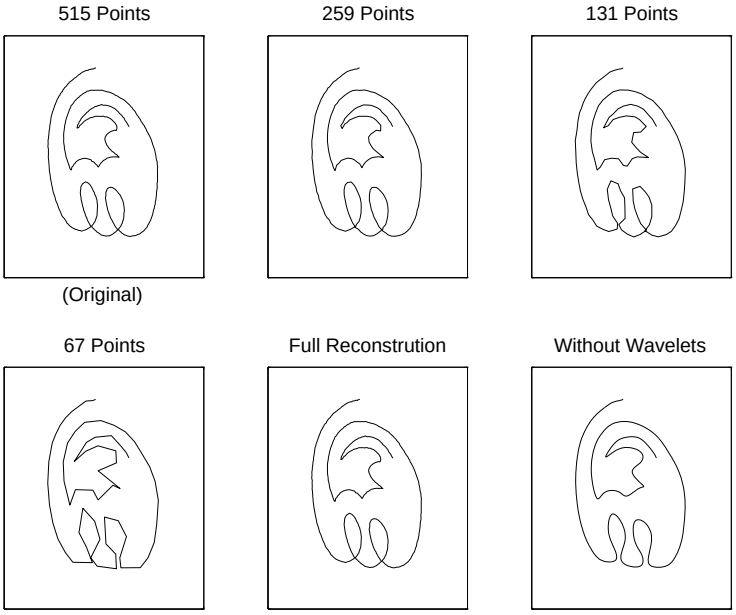

Figure 1: *Free-hand curve*

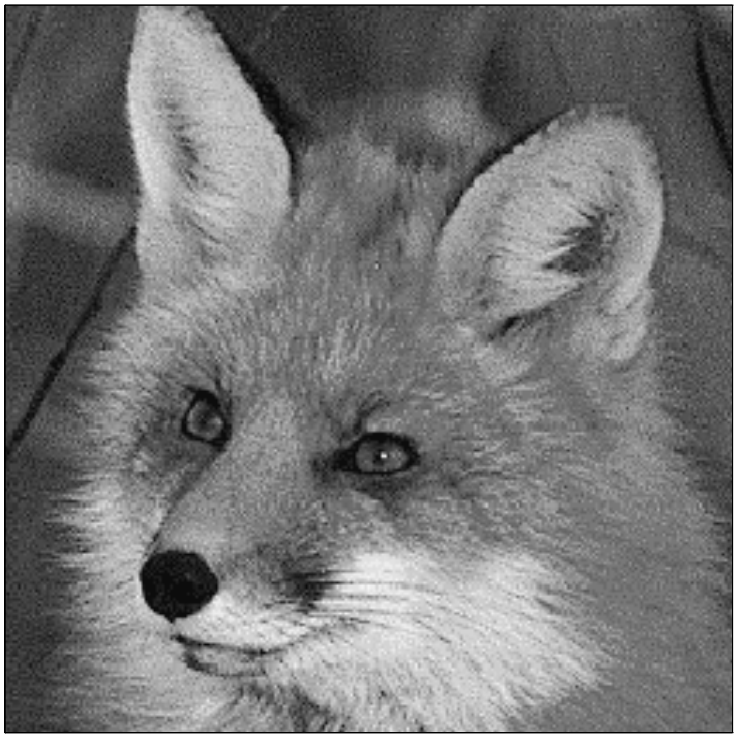

Figure 2: *Original fox image* (259×259)

Figure 3: *Fox image after two levels of approximation* (67×67)

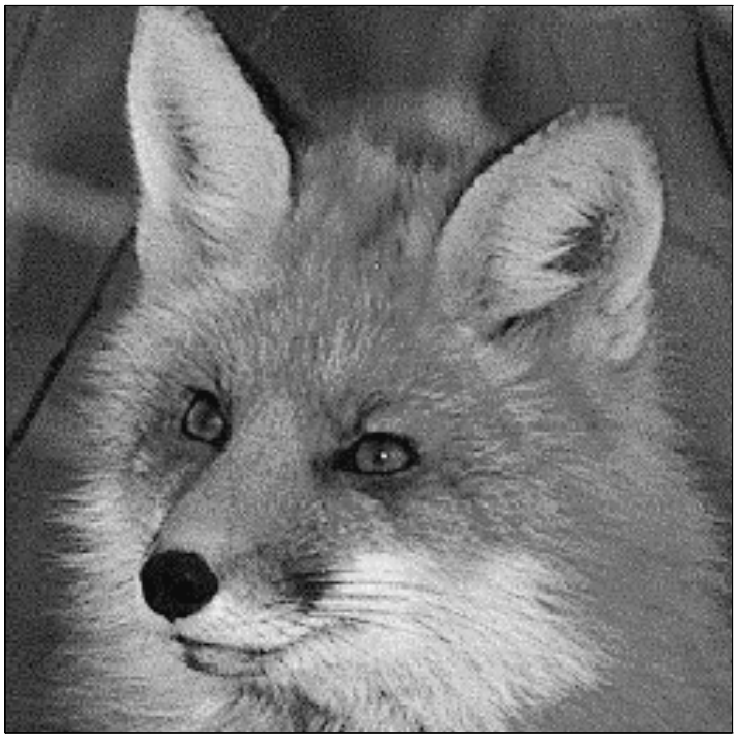

Figure 4: *Fully reconstructed fox image* (259×259)

Figure 5: *Reconstructed fox image without error terms* (259×259)

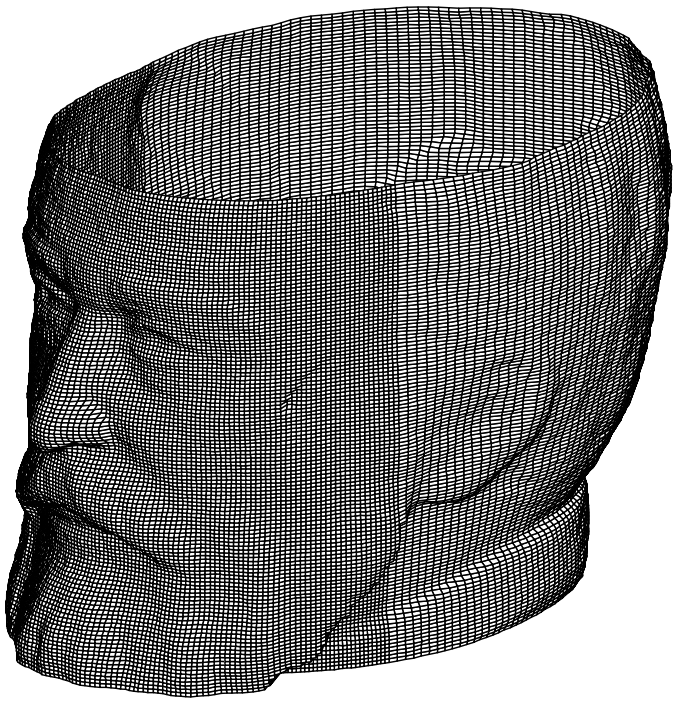

Figure 6: *Original Hugo data* (259×131)

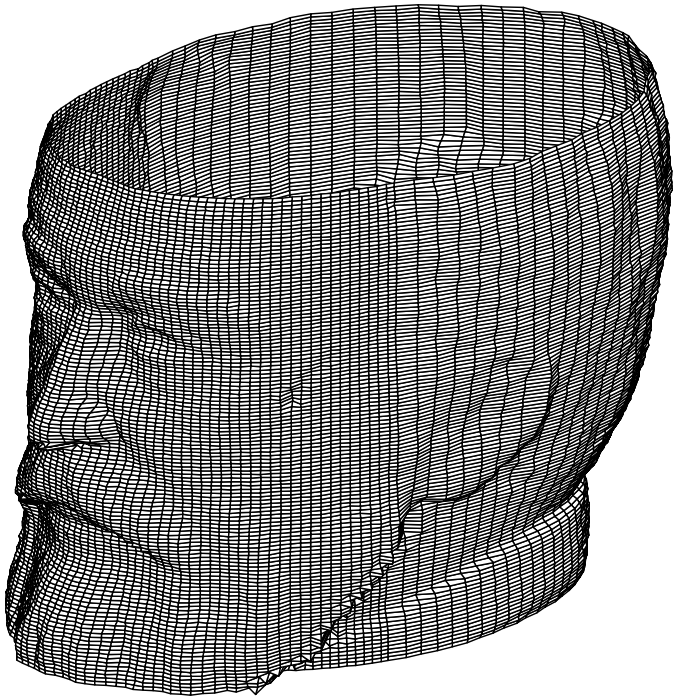

Figure 7: *Hugo data approximated one level in one direction (*131×131*)*

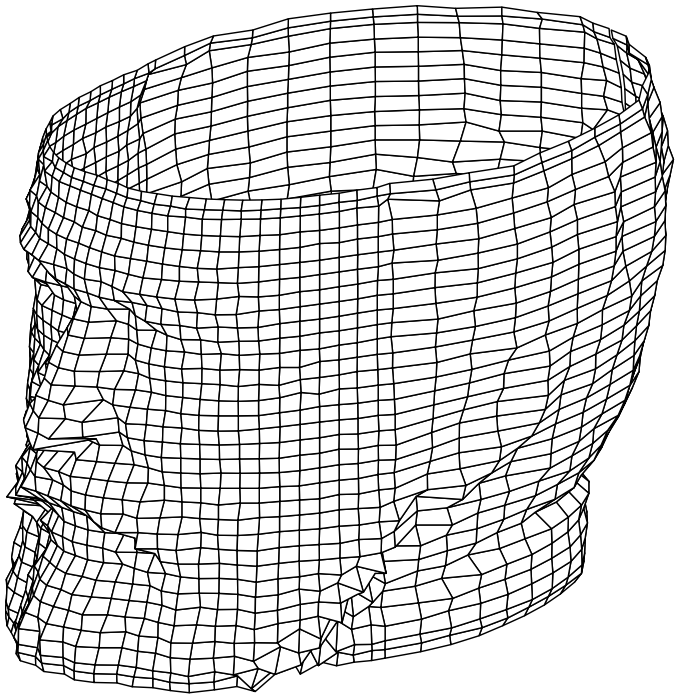

Figure 8: *Hugo data approximated two levels in both directions (*67×35*)*

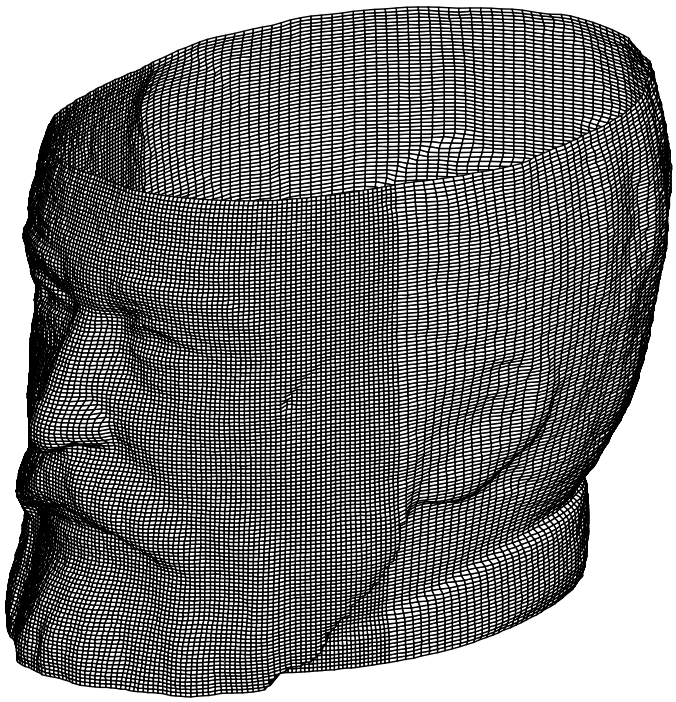

Figure 9: *Fully reconstructed Hugo from two levels (*259×131*)*

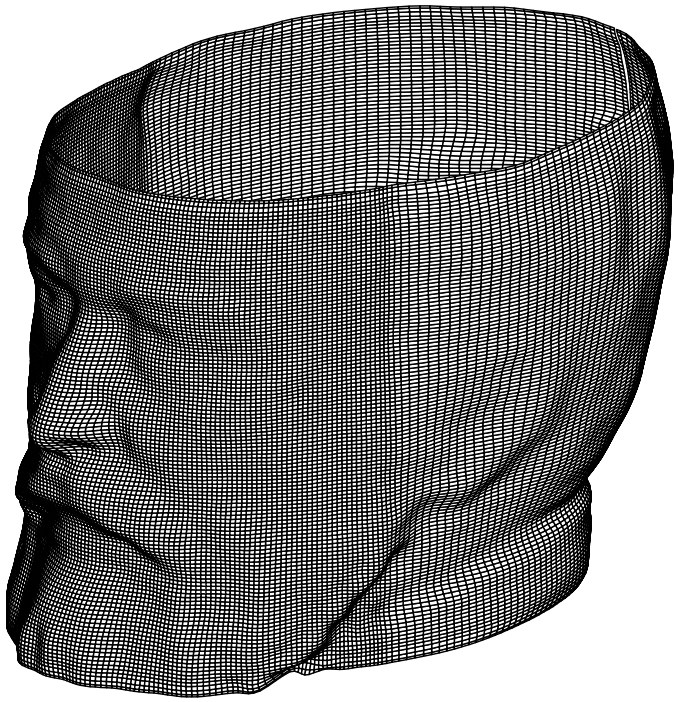

Figure 10: *Reconstructed Hugo from two levels without error terms (*259×131*)*