

# Multiresolutions Numerically from Subdivisions

Richard Bartels<sup>a</sup>, Faramarz Samavati<sup>b</sup>

<sup>a</sup>*Computer Graphics Laboratory, David R. Cheriton School of Computer Science,  
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

<sup>b</sup>*Computer Science Department, University of Calgary, Calgary, Alberta T2N 1N4, Canada*

---

## Abstract

In previous work we introduced a construction to produce multiresolutions from given subdivisions. A portion of that construction required solving bilinear equations using a symbolic algebra system. Here we replace the bilinear equations with a pair of linear equation systems, resulting in a completely numerical construction. Diagrammatic tools provide assistance in carrying this out. The construction is shown for an example of univariate subdivision. The results for a bivariate subdivision are given to illustrate the construction's ability to handle multivariate meshes, as well as special points, without requiring any modification of approach. The construction usually results in analysis and reconstruction filters that are finite, since it seeks each filter locally for the neighborhood of the mesh to which it applies. The use of a set of filters constructed in this way is compared with filters based on spline wavelets for image compression to show that the construction can yield satisfactory results.

*Keywords:* subdivision, refinement, multiresolution

---

## 1. Introduction

Multiresolution methods receive important mention in most reference works on graphics and computer-aided geometric design; e.g., [1, 2, 3]. This paper derives multiresolutions directly from subdivisions defined on mesh data. It improves upon [4], and uses the diagrammatic tools of [5]. The approach in [4] required solving a system of bilinear equations using symbolic algebra. The present work replaces this with a construction that is entirely numerical. This is done by solving two linear systems of the form

$$\mathbf{x}\mathbf{G} = \mathbf{i}_x$$

with given matrices  $\mathbf{G}$  for unknowns  $\mathbf{x}$ , where  $\mathbf{i}_x$  is an appropriate portion of a row of the identity, and obtaining a nontrivial solution,  $\mathbf{y}$ , of one linear system of the form

$$\mathbf{M}\mathbf{y} = \mathbf{0}$$

for a given matrix  $\mathbf{M}$ . These can be easily solved by appealing to the Singular Value Decomposition [6], whose implementation is included in most modern, numerical libraries for linear systems; e.g. [7].

To keep the presentation short by avoiding various complications, we shall be considering here only *finite meshes* and subdivisions consistent with *tessellation* of the mesh domain.

Section 2 places the construction in context. Section 3 reviews multiresolution matrices. Section 4 presents the construction's computations in outline. Section 5 reviews the use of diagrams in the construction. Section 6 fills in details, illustrated using a regular, univariate subdivision. Section 7 gives an example of bivariate subdivision on regular triangular meshes. In Section 8, as an indication that the construction has some merit, we make a few comparisons between the multiresolution derived in Section 6 and a multiresolution based on semiorthogonal B-spline wavelets. Section 9 will sum up our results.

## 2. Context and Literature

Using MathSciNet, the American Mathematical Society’s search engine, to locate the word *multiresolution* in titles yields, as of this writing, 518 hits. Using the Association for Computing Machinery’s search engine in a like manner currently yields 734 hits. A comprehensive review in the short space of a single paper has become impossible. Nevertheless, we can touch on two approaches for multiresolution: *function-based multiresolution* and *data-based multiresolution*. Our construction falls in the latter approach.

In the former approach to multiresolution, one function space  $\mathbb{F}$  is the direct sum of a subspace  $\mathbb{C}$  and its complement  $\mathbb{D}$ , and these give a single stage in a nested chain of spaces of the same genre (e.g. all in  $L^2$ ). The main goal is that of approximating functions of  $\mathbb{F}$  by functions of  $\mathbb{C}$  and expressing the approximation error using functions of  $\mathbb{D}$ . The approximation and the error are characterized by an inner product on the spaces, and the approximation and error expressions are facilitated by the judicious choice of bases  $\{\phi_{\mathbb{F}}\}$ ,  $\{\phi_{\mathbb{C}}\}$  and  $\{\psi_{\mathbb{D}}\}$  for the spaces of  $\mathbb{F}$ ,  $\mathbb{C}$ , and  $\mathbb{D}$  respectively. The goal is to have mappings  $p_{\mathbb{C}}$  and  $q_{\mathbb{D}}$  that will reconstruct any  $f \in \mathbb{F}$  from its approximation  $c \in \mathbb{C}$  together with its error complement  $d \in \mathbb{D}$

$$p_{\mathbb{C}} : c \oplus q_{\mathbb{D}} : d \rightarrow f \quad (1)$$

to have mappings  $a_{\mathbb{F}}$  and  $b_{\mathbb{F}}$  that will act in reverse

$$\begin{aligned} a_{\mathbb{F}} : f &\rightarrow c \\ b_{\mathbb{F}} : f &\rightarrow d \end{aligned} \quad (2)$$

and to have the form of these mappings be particularly elegant in terms of the chosen bases. An aspect of this elegance is that the mappings can be realized as simple *decomposition and reconstruction filters* that apply to the coefficients of the representations of  $f$ ,  $c$ , and  $d$  in terms of the bases. That elegance is achieved by the properties of orthogonality that can be engineered for these bases with respect to the genre of space they occupy. Obvious examples here would be provided by the work of Daubechies [8], Gori and Pitolli

[9], Dahmen and Micchelli [10], Cohen and Dyn [11], or Chui [12, 13].

In the latter approach to multiresolution, the data is paramount, and it is usually encountered in the form of  $k$ -dimensional meshes. One is confronted with a *coarse* mesh,  $\mathbf{c}$ , having some connectivity,  $\mathcal{C}$ , and a *subdivision* process,  $p_{\mathcal{C}}$ , that expands it into a *fine* mesh,  $\mathbf{f}$ , having a connectivity,  $\mathcal{F}$ , that is related to  $\mathcal{C}$  in some sense. In the literature that relationship is clarified with various definitions of *subdivision connectivity*. The construction of a multiresolution consists of reversing this process by finding a mapping  $a_{\mathcal{F}}$  that will deliver an approximation  $\mathbf{c}$  suitable for connectivity  $\mathcal{C}$  from any  $\mathbf{f}$  of connectivity  $\mathcal{F}$ , even if the fine mesh  $\mathbf{f}$  was not produced by subdivision. To cover the case in which the subdivision of the resulting  $\mathbf{c}$  only produces the mesh  $\mathbf{f}$  corrupted by a mesh of errors  $\mathbf{r}$  of  $\mathcal{F}$ -connectivity, another pair of processes are desired:  $b_{\mathcal{F}}$  that will extract error information  $\mathbf{d}$  that may be associated with the “difference connectivity”  $\mathcal{F} \setminus \mathcal{C}$ , and  $q_{\mathcal{F} \setminus \mathcal{C}}$  that will map that error information back to  $\mathbf{r}$ . In other words, (1) and (2) reappear as (3) and (4)

$$p_{\mathcal{C}} : \mathbf{c} \oplus q_{\mathcal{F} \setminus \mathcal{C}} : \mathbf{d} \rightarrow \mathbf{f} \text{ (}\mathcal{F}\text{-connected)} \quad (3)$$

$$\begin{aligned} a_{\mathcal{F}} : \mathbf{f} &\rightarrow \mathbf{c} \text{ (}\mathcal{C}\text{-connected)} \\ b_{\mathcal{F}} : \mathbf{f} &\rightarrow \mathbf{d} \text{ (}\mathcal{F} \setminus \mathcal{C}\text{-connected)} \end{aligned} \quad (4)$$

As in the function-based orientation, the goal is to express the mappings in the form of decomposition and reconstruction filters. But the focus is on data mappings. If any functions or function spaces underlie the meshes, they might be ignored, or they might be invoked secondarily as a means of constructing the mappings, or they might be replaced by spaces constructed for the single purpose of delivering convenient filters for the meshes in question.

Some of the work in data-based multiresolution has been carried out by purely procedural means; that is, the processes of subdivision and approximation are defined in terms of operations on the mesh (e.g. vertex insertion or edge collapse), while the error processes appear as information retained in data structures. Examples of

this approach would include the progressive mesh methods due to Hoppe [14], the mesh decimation of Schroeder, et. al. [15], and the kite trees of Xu, et. al. [16]. This type of data-based approach will not concern us here.

Some of the data-based work has been carried out by establishing ad-hoc function spaces to underlie the mesh data (e.g. piecewise linear functions on the facets of the mesh) so that the tools of the functional approach can be used to construct the mappings of (3) and (4). This might be done, even when pre-existing function spaces exist, if the ad-hoc spaces are easier to deal with. The landmark work of Lounsbery, DeRose, and Warren [17] would be a good example of the ad-hoc space approach.

Another data-based approach makes use of *lifting*, due to Sweldens [18, 19, 20, 21]. In lifting, the multiresolution processes of  $a_{\mathcal{F}}$ ,  $b_{\mathcal{F}}$  and  $q_{\mathcal{F}\setminus\mathcal{C}}$ , and even  $p_{\mathcal{C}}$ , can be bootstrapped directly from the data. This may be done without appealing to any function-space tools, if desired, or one may guide the bootstrap process towards filling some function-space requirement (e.g. zero moments for  $\{\psi_{\mathbb{D}}\}$ ). Lifting can also be used to modify an existing multiresolution into one having additional properties. Lifting appears as a tool in a number of works; e.g., Schröder and Sweldens [22], Qin [23, 24], and Bertram [25, 26, 27], and Olsen, et. al. [28].

Our previous work [4, 5], though firmly in the data-oriented camp, stands somewhat apart from any described above, although it has a spiritual affinity with the matrix approaches employed in lifting. Both can achieve their ends by purely matrix means without appealing to function spaces. However, lifting is capable of creating the subdivision,  $p_{\mathcal{C}}$ , along with the other mappings  $a_{\mathcal{F}}$ ,  $b_{\mathcal{F}}$ , and  $q_{\mathcal{F}\setminus\mathcal{C}}$ , whereas our construction never attempts to do this. We always start with a pre-existing  $p_{\mathcal{C}}$ . Lifting allows for a wide range of strategies in developing the multiresolution mappings, although the strategies are relatively general, flexible, even vague, in their description. Our construction, on the other hand, concentrates specifically on data approximation as its driving force; i.e., minimizing  $\|\mathbf{r}\| = \|\mathbf{f} - (p_{\mathcal{C}} : \mathbf{c})\|$  with

respect to some norm  $\|\cdot\|$ . Indeed, if the goal is to determine  $\mathbf{c}$  to be as close as possible, in some approximative sense, to a mesh that would produce  $\mathbf{f}$  under the given subdivision  $p_{\mathcal{C}}$ , then lifting is not a particularly convenient strategy. The goal in our construction is to begin with a given  $p_{\mathcal{C}}$  and its associated subdivision connectivity  $\{\mathcal{C}, \mathcal{F}\}$ , to completely ignore the use of any possible underlying function-space tools, to construct  $a_{\mathcal{F}}$  as a form of data approximation designed upon the given  $p_{\mathcal{C}}$ , and to construct  $q_{\mathcal{F}\setminus\mathcal{C}}$  and  $b_{\mathcal{F}}$  in strict consequence for representing the approximation error. The construction is carried out with the intention of producing finite decomposition and reconstruction filters, where possible, and satisfying biorthogonality relationships. It offers a constructive approach that may be carried out unchanged on meshes of any dimensionality. The intent is to start with works that introduce a particular subdivision, such as Stam and Loop [29], Stam [30], Loop [31], Doo [32], Dyn, et. al., [33, 34] or survey a selection of subdivisions, such as Ma [35], and to end with multiresolutions having finite filters that incorporate the given subdivisions.

Four important uses for multiresolutions have historically been: (1) as a means of displaying geometry at various scales and distances; e.g., as in Certain, et. al., [36]; (2) as a means of supporting multi-level editing; e.g., as in Kobbelt, et. al., [37], Zorin, et. al., [38], Finklestein and Salesin [39]; (3) as a means of analyzing signals to localize components by time and frequency; e.g., as is discussed in the book by Strang and Nguyen [40], and (4) to provide a means of compressing data, usually images or audio and video streams, an introduction for which can be found in the book by Stollnitz, et al. [2]

### 3. Underlying Matrices

We can revisit equations (3) and (4) in matrix terms. In a subdivision, the  $n$  points  $\mathbf{c}$  of a coarse mesh are mapped into the  $m > n$  points  $\mathbf{f}$  of a fine mesh, by the  $m \times n$  matrix-vector transformation

$$\mathbf{P}\mathbf{c} \rightarrow \mathbf{f} \tag{5}$$

If the fine points  $\mathbf{f}$  are given arbitrarily, not mapped from points  $\mathbf{c}$ , then it is to be expected that

$$\mathbf{P}\mathbf{c} + \mathbf{r} \rightarrow \mathbf{f} \quad (6)$$

for any coarse mesh  $\mathbf{c}$ , with a residual  $\mathbf{r}$  that would depend on  $\mathbf{f}$  and the choice of  $\mathbf{c}$ . A single stage of a multiresolution involving the subdivision (5) seeks to determine a good approximation  $\mathbf{c}$  from  $\mathbf{f}$  in the form

$$\mathbf{A}\mathbf{f} \rightarrow \mathbf{c} \quad (7)$$

where  $\mathbf{A}$  is  $n \times m$ . The measure of goodness is that that  $\|\mathbf{r}\|$  be small and  $\mathbf{r}$  can be derived from  $\mathbf{f}$  in the form

$$\mathbf{B}\mathbf{f} \rightarrow \mathbf{d} \quad (8)$$

so that

$$\mathbf{Q}\mathbf{d} \rightarrow \mathbf{r} \quad (9)$$

The matrix  $\mathbf{Q}$  is  $m \times (m - n)$  and the matrix  $\mathbf{B}$  is, correspondingly,  $(m - n) \times m$ .

As a result, (7) and (8) together provide a *decomposition* of  $\mathbf{f}$  while

$$\mathbf{P}\mathbf{c} + \mathbf{Q}\mathbf{d} \rightarrow \mathbf{f} \quad (10)$$

provides a *reconstruction*, and together the decomposition and reconstruction comprise a stage of the multiresolution. Equations (7), (8), and (10) represent (3) and (4) re-phrased, but without any connectivity information. The rows of  $\mathbf{P}$  and  $\mathbf{Q}$  act as the *reconstruction filters*, and the rows of  $\mathbf{A}$  and  $\mathbf{B}$  act as the *decomposition* (or *analysis*) *filters*, as they are usually presented in the literature.

Equations (7), (8), (9), and (10) imply that the matrices  $\mathbf{A}$  and  $\mathbf{B}$  form the inverse of  $\mathbf{P}$  and  $\mathbf{Q}$ , so that together they satisfy the biorthogonality relationships

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \quad (11)$$

and, consequently,

$$\begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = \mathbf{I} \quad (12)$$

Conversely, if (11) holds, we have one stage of a multiresolution: (7), (8), and (10).

#### 4. Construction Outline

Our earlier version of the construction, which used symbolic algebra, is covered in detail in [4], and we have used successfully it for a number of multiresolutions for 1D, 2D, and 3D data; see [41] for examples, applications, and further references.

In this paper we proceed as follows:

1. Obtain each row  $\mathbf{a}$  separately for the matrix  $\mathbf{A}$  to form a left inverse of  $\mathbf{P}$ ; that is,  $\mathbf{A}\mathbf{P} = \mathbf{I}$ , and  $\mathbf{a}$  is chosen so that the point  $c$  obtained as  $\mathbf{a}\mathbf{f} \rightarrow c$  is a least-squares approximation from the components of  $\mathbf{f}$  confined to a neighborhood.
2. Obtain each column  $\mathbf{q}$  separately for  $\mathbf{Q}$  to be in the nullspace of  $\mathbf{A}$ . Each  $\mathbf{q}$  will be chosen to have nonzeros confined to a neighborhood.
3. Obtain each row  $\mathbf{t}$  separately of a matrix  $\mathbf{T}$  to form a left inverse of  $\mathbf{Q}$ ; that is,  $\mathbf{T}\mathbf{Q} = \mathbf{I}$ . Each  $\mathbf{t}$  is chosen to have nonzeros confined to a neighborhood.
4. Obtain each column of  $\mathbf{B}$  by applying an appropriate submatrix of  $\mathbf{T}$  to the corresponding nonzero entries in the corresponding column of  $\mathbf{I} - \mathbf{P}\mathbf{A}$ .

All steps can be carried out numerically, and the justification comes from (12) by the following

$$\begin{aligned} \mathbf{Q}\mathbf{B} &= \mathbf{I} - \mathbf{P}\mathbf{A} \\ \mathbf{T}\mathbf{Q} &= \mathbf{I} \\ \Rightarrow \mathbf{B} &= \mathbf{T}(\mathbf{I} - \mathbf{P}\mathbf{A}) \end{aligned} \quad (13)$$

Since the entire construction is numerically based, this opens the possibility of implementing the construction in code for use dynamically, on the fly, for the input of a variety of subdivisions and meshes.

Each row  $\mathbf{a}$  of  $\mathbf{A}$  will be associated with one of the nodes of  $\mathcal{C}$ . Most of the elements of each row  $\mathbf{a}$  of  $\mathbf{A}$  will be zero, and the nonzero elements will be based on the structure of  $\mathbf{P}$ . They correspond to the components of  $\mathbf{f}$  in an  $\mathcal{F}$ -connectivity neighborhood, where the neighborhood takes into account the points of  $\mathbf{f}$  whose positions are influenced by the coarse point  $c$  associated with  $\mathbf{a}$ .

Everything is arranged to produce a least-squares approximation to that point  $c$  from those components of  $\mathbf{f}$ . It is only necessary to solve

$$\mathbf{a}_{subvec} \times \mathbf{P}_{submtx} = \mathbf{i}_{subvec} \quad (14)$$

for the unknown nonzeros in  $\mathbf{a}_{subvec}$ , where  $\mathbf{i}_{subvec}$  is the corresponding row portion of the corresponding columns of the identity matrix. The relationship between (14) and least-squares fitting is covered extensively in [4] and [42].

Each column,  $\mathbf{q}$  of  $\mathbf{Q}$  is found individually in Step 2 of our construction. Only a subvector of  $\mathbf{q}$  will have nonzero unknowns, corresponding a neighborhood in the  $\mathcal{F}$ -connectivity; the column  $\mathbf{q}$  will be associated with a fine point in the  $\mathcal{F} \setminus \mathcal{C}$  connectivity, and the unknowns are found by solving

$$\mathbf{A}_{submtx} \times \mathbf{q}_{subvec} = \mathbf{0} \quad (15)$$

In order that the solution be nontrivial, these equations are solved as a nullspace problem rather than as a system of linear equations

Step 3 of our construction mimics Step 1, using connectivity to find each row,  $\mathbf{t}$ , of  $\mathbf{T}$  by solving

$$\mathbf{t}_{subvec} \times \mathbf{Q}_{submtx} = \mathbf{i}_{subvec} \quad (16)$$

for the unknowns in  $\mathbf{t}_{subvec}$ . Unlike Step 1, where the rows were associated with points of  $\mathcal{C}$ , these rows are associated with points of  $\mathcal{F} \setminus \mathcal{C}$ .

Step 4 of our construction assembles the final matrix  $\mathbf{B}$  by multiplying each row of  $\mathbf{T}$  onto each column of  $\mathbf{I} - \mathbf{PA}$ . The structure of the rows and columns results in many of such products being trivially zero, for which no computation is needed.

The three linear problems (14), (15), and (16) can all be solved using the Singular Value Decompositions (SVD) [6] of the respective matrices in the problems. Any  $k \times \ell$  matrix  $\mathbf{M}$  can be factored into

$$\mathbf{M} = \mathbf{USV}^T$$

where  $\mathbf{U}$  is a  $k \times k$  orthogonal matrix ( $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ ),  $\mathbf{V}$  is an  $\ell \times \ell$  orthogonal matrix, and  $\mathbf{S}$  is a  $k \times \ell$  matrix whose diagonal entries ( $s_{i,i} = \sigma_i$ ,  $i = 1, \dots, \min(k, \ell)$ ) are the square roots of the eigenvalues of  $\mathbf{M}^T \mathbf{M}$  (known as the *singular values* of

$\mathbf{M}$ ) and all of whose other entries are zero. Further, if  $r$  is the rank of  $\mathbf{M}$ , then  $\sigma_{r+1}, \dots, \sigma_k = 0$ ; that is,  $\mathbf{M}$  can be put in the form

$$[\mathbf{U}_1 | \mathbf{U}_2] \left[ \begin{array}{c|c} \sigma_1 & \mathbf{O} \\ \vdots & \\ & \sigma_r & \mathbf{O} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

$\mathbf{U}_1$  represents the first  $r$  columns of  $\mathbf{U}$  and  $\mathbf{V}_1$  the first  $r$  columns of  $\mathbf{V}$ . This is the SVD of  $\mathbf{M}$ .

Equations of the form

$$\mathbf{xM} = \mathbf{y}$$

are solved for  $\mathbf{x}$  as

$$\mathbf{y} [\mathbf{V}_1 | \mathbf{V}_2] \left[ \begin{array}{c|c} \frac{1}{\sigma_1} & \mathbf{O} \\ \vdots & \\ & \frac{1}{\sigma_r} & \mathbf{O} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right]^T \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix}$$

Moreover, for nullspace problems,

$$\mathbf{Mv} = \mathbf{0}$$

the columns of the submatrix of  $\mathbf{V}_2$  provide a full, orthonormal basis for the nullspace of  $\mathbf{M}$ .

The SVD is convenient in being the only single matrix decomposition that can solve both equations and nullspace problems and do so in a numerically accurate and stable manner. For extensive details on the theory and use of the SVD, see [6]. For an example of its implementation in a numerical subroutine library, see [7]. For its use in analyzing the approximations  $c$  produced in Step 1, see [42].

## 5. Structure and Connectivity

The structure of the subdivision matrices and the connectivities of the meshes play an important role in the construction. This is easiest to discuss when the *subdivision connectivity* corresponds to *tessellation* of the mesh domain. Figure 1 shows a coarse tessellation of a portion of a domain, and

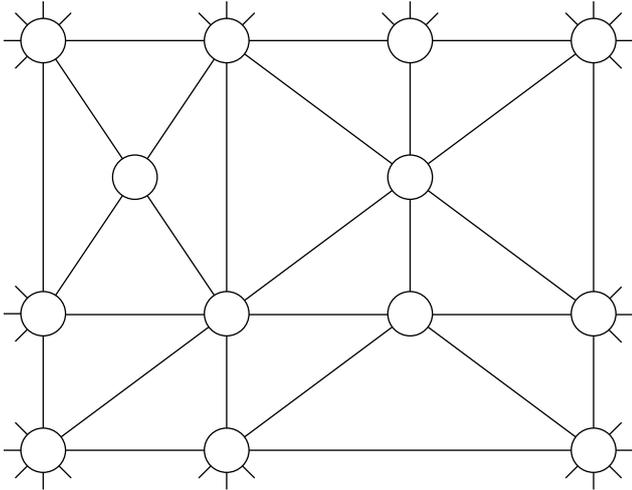


Figure 1: Coarse Tessellation Mesh

Figure 2 shows that portion divided further into a fine tessellation. Many, but not all, multiresolutions conform to tessellation connectivity. Our construction has been used on other forms of subdivision; e.g. [43], but we confine ourselves to tessellation here for simplicity.

The geometric position of the points in the mesh are not revealed in such figures, but to each circle-node in Figure 1 there is associated one of the points  $c$  of  $\mathbf{c}$ , and to each circle-node or square-node in Figure 2 there is associated one of the points  $f$  of  $\mathbf{f}$ . We can connect these figures with the matrix  $\mathbf{P}$  in the following way. Each

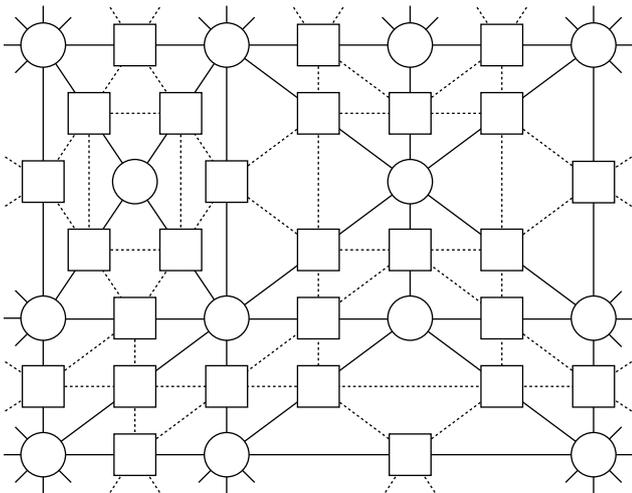


Figure 2: Fine Tessellation Mesh

node of Figure 1, being associated with one geo-

metric point  $c_j$ , can be connected to the column  $\mathbf{p}_j$  of  $\mathbf{P}$  whose elements multiply  $c_j$

$$\mathbf{P}\mathbf{c} = \cdots + \mathbf{p}_j c_j + \cdots \rightarrow \mathbf{f}$$

Each node of Figure 2, being associated with one geometric point  $f_i$ , can be connected to the row  $\mathbf{p}^i$  of  $\mathbf{P}$  whose elements produce  $f_i$

$$\mathbf{p}^i \cdot \mathbf{c} = \sum_j p_{i,j} c_j \rightarrow f_i$$

There is one node of Figure 1 for each column of  $\mathbf{P}$ , and consequently all nodes of Figure 1 together can be put in one-to-one correspondence with the entries in any chosen row of  $\mathbf{P}$ . There is one node of Figure 2 for each row of  $\mathbf{P}$ , and consequently all nodes of Figure 2 together can be put in one-to-one correspondence with the entries in any chosen column of  $\mathbf{P}$ .

To be complete, the diagrams we use should employ both coarse connectivity figures such as Figure 1 and the corresponding fine figures such as Figure 2, and for more general subdivision connectivities this is done. However, with connectivities coming from tessellation, it is possible to economize. Figure 2 clearly has Figure 1 embedded within it, and we can make do with just Figure 2; e.g., writing entries of a row of  $\mathbf{P}$  simply on the circle-nodes of Figure 2 or writing the entries of a column of  $\mathbf{P}$  on both the square-nodes and circle-nodes of that figure.

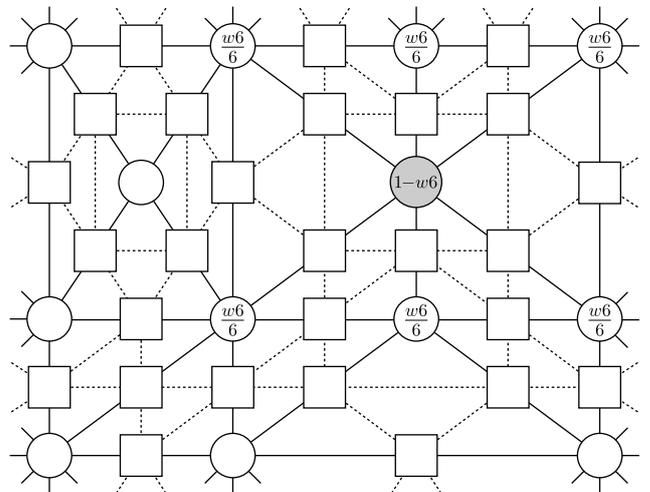


Figure 3: Loop  $\mathbf{P}$  Row Diagram

To illustrate, we will use the  $\mathbf{P}$  for the subdivision due to Loop [31], which is based upon tessellation connectivity. Figure 3, for example,

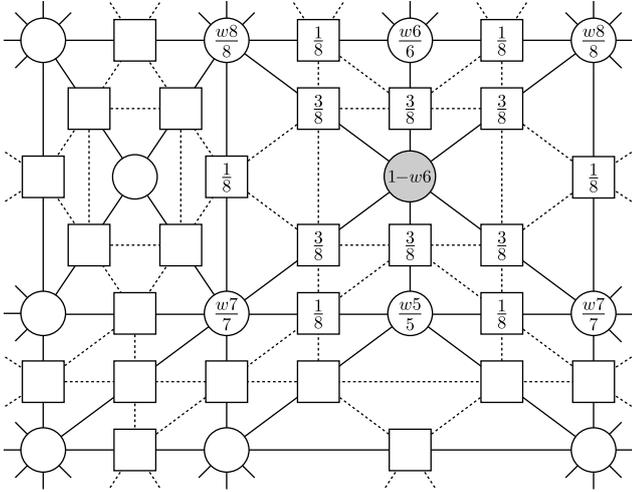


Figure 4: Loop  $\mathbf{P}$  Column Diagram

shows the entries of the *row* of  $\mathbf{P}$  that is connected to the shaded circle-node, which is acting as a fine-mesh node associated with a corresponding point  $f_\bullet$  of  $\mathbf{f}$ . Figure 4 shows the entries of the *column* of  $\mathbf{P}$  that is connected to the shaded circle-node, which is now acting as a coarse-mesh

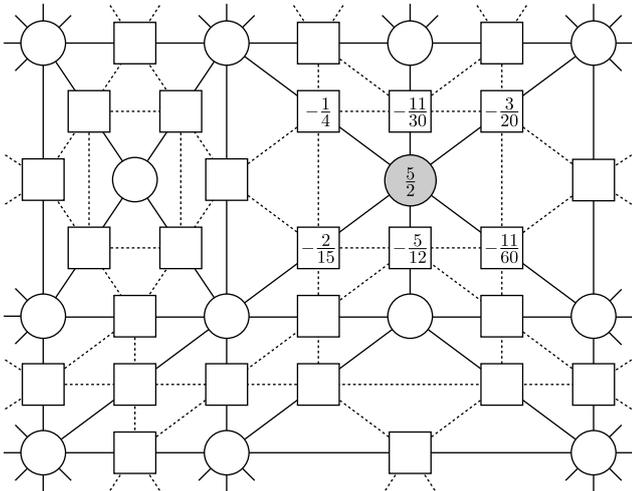


Figure 5: Loop  $\mathbf{A}$  Row Diagram

node associated with a corresponding point  $c_\bullet$  of  $\mathbf{c}$ . In these figures  $wn$ , for any positive integer  $n$ , stands for  $\frac{5}{8} - (\frac{3}{8} + \frac{1}{4} \cos(\frac{2\pi}{n}))^2$ . Figure 3 is, in fact, an example of the kind of *mask* that is generally found in references to Loop subdivision,

and the appearance of such masks, having nodes representing associations with coarse or fine mesh points depending on context, is quite common in the subdivision literature.

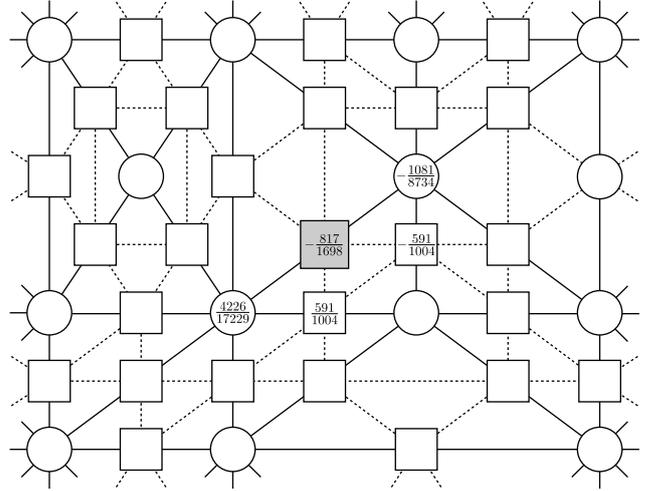


Figure 6: Loop  $\mathbf{Q}$  Column Diagram

In [5] we introduced the terms *row diagram* and *column diagram* on the one hand to be able to name diagrams more informatively as to their role in the construction and on the other hand to avoid confusion the common term *mask*. In the literature a mask is typically a depiction of the row of a matrix, and we need depictions of columns as well.

Similar considerations, presented in [5], provide the connection between any row of  $\mathbf{A}$  and the circle-nodes, any row of  $\mathbf{B}$  and the square nodes, or any column of  $\mathbf{Q}$  and the square-nodes. The diagrams of Figure 5, Figure 6, and Figure 7 show examples that are consistent with Figure 4 in the sense of (11) and (12).

In short: Figure 2 can present all the important connectivities:  $\mathcal{C}$ ,  $\mathcal{F}$ , and  $\mathcal{F} \setminus \mathcal{C}$ . The circles alone, together with the solid edges, provide  $\mathcal{C}$ . The circles and squares, together with the solid and dashed edges combined, give  $\mathcal{F}$ . The squares without the circles, and all edges necessary to join up the squares, show  $\mathcal{F} \setminus \mathcal{C}$ .

This economization comes in particularly handy in formulating the computations needed for the construction, all of which are based on matrix-row times matrix-column products. With

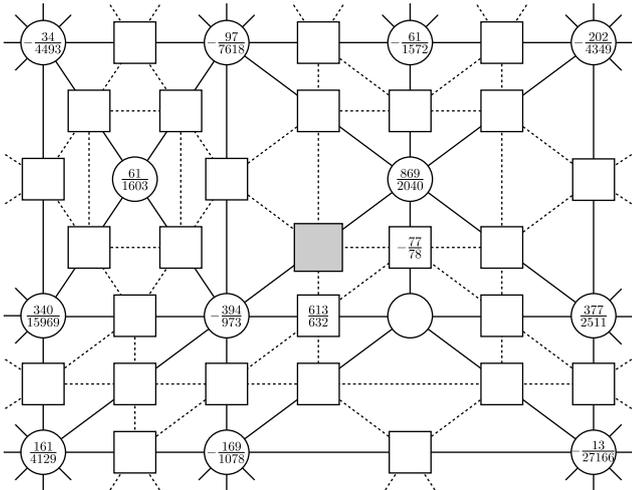


Figure 7: Loop **B** Row Diagram

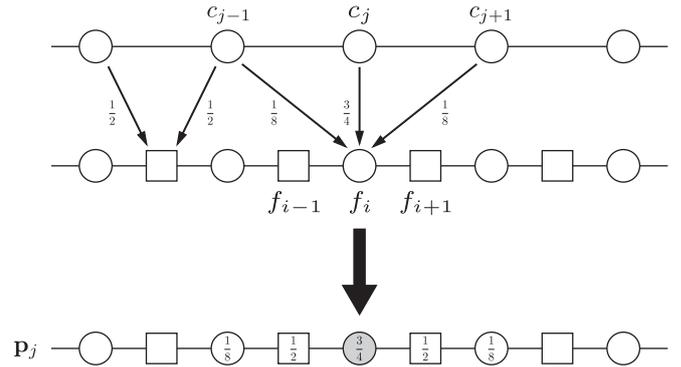


Figure 8: A 1D Subdivision

two appropriate diagrams juxtaposed on a single figure, the row correspondence of the left-hand matrix in a product and the column correspondence of the right-hand matrix in a product, it is possible to determine easily what the outcome of the row-column multiplication will be. Simply multiply the corresponding left-hand and right-hand entries appearing in common on any node, and sum all resulting products. The economization also governs the steps of the construction: assign one  $\mathbf{P}$  column and one  $\mathbf{A}$  row to each circle-node; assign one  $\mathbf{Q}$  column and one  $\mathbf{B}$  row to each square-node; equations (11) and (12) dictate that the  $\mathbf{A}$ - $\mathbf{P}$  and  $\mathbf{B}$ - $\mathbf{Q}$  interactions on common nodes produce 1, and all other interactions implied by those equations produce 0. See [5] for more details. We shall be using these conventions in Section 6.

## 6. Multiresolution Construction

Here we will present our construction by carrying it completely through using the 1D (univariate) subdivision displayed in Figure 8 as an illustrative example.  $\mathcal{C}$  is shown at the top, with the associated geometric points  $c$  indicated alongside.  $\mathcal{F}$  is in the middle, with associated  $f$  indicated, shown with the affine combinations that produce the points associated with  $\mathcal{F} \setminus \mathcal{C}$  (the squares:  $[\frac{1}{2}, \frac{1}{2}]$ ) and the remaining points (the circles:  $[\frac{1}{8}, \frac{3}{4}, \frac{1}{8}]$ ). The corresponding  $\mathbf{P}$  column

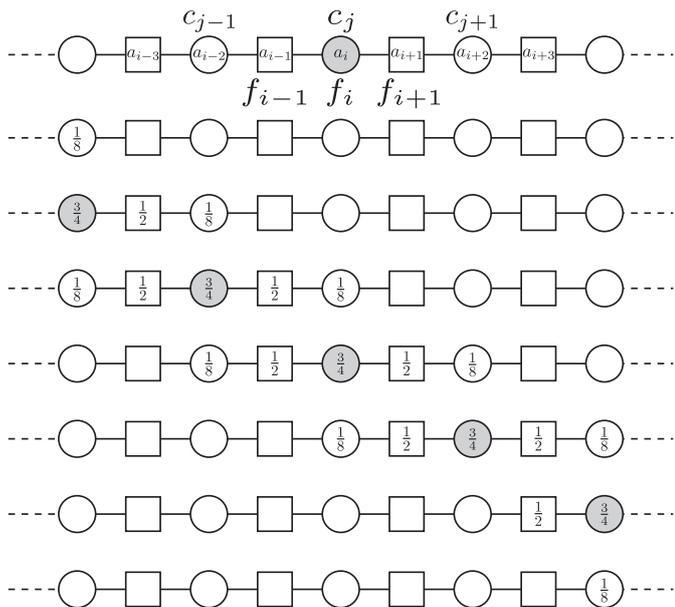
diagram for  $c_j$  as its connected “home” node is shown on a separate copy of  $\mathcal{F}$  at the bottom.

While the use of diagrams for a simple, 1-dimensional subdivision is over-kill, and while the construction in this case is almost trivial, and while 1-dimensional meshes may not seem very interesting, we are hoping that the details to this simple example make our construction clear in general.

### 6.1. The Reversal Approximation $\mathbf{A}$

Figure 9 expands Figure 8 to what is necessary for deriving (14), the problem to find the nonzero elements in the row of  $\mathbf{A}$  represented by the diagram on the copy of  $\mathcal{F}$  at the top of the figure. The row of  $\mathbf{A}$  is the one whose connected “home” node is the darkened point corresponding to  $c_j$ , and its (as yet unknown) nonzero column elements are shown on their corresponding nodes of  $\mathcal{F}$ . The  $\mathbf{A}$  row diagram of Figure 8 has width 3. (The *width* of a diagram is the maximum number of edges in any shortest path from the diagram’s home node to any of its labeled nodes.) Each successive row of  $\mathbf{A}$  will be similarly connected to each successive circle-node and will have some chosen width; the width of any  $\mathbf{A}$  row diagram can be chosen independently of the width of any of the other  $\mathbf{A}$  row diagrams, though in a regular setting such as this there is no reason nor need to do that.

Every nearby  $\mathbf{P}$  column diagram is shown on the repetitions of  $\mathcal{F}$  drawn below the  $\mathbf{A}$  row diagram. The row-diagram  $\times$  column-diagram interactions represented by Figure 9 are the sums



Multiple Copies of  $\mathcal{F}$

Figure 9:  $\mathbf{A}$  Row Interaction with  $\mathbf{P}$  Columns

of the products of the numeric labels on corresponding mesh points. These amount to a vector  $\times$  matrix product of the form mentioned in (14). The results for Figure 9 are shown in (17).

$$\begin{aligned}
 & [a_{i-3} \ a_{i-2} \ a_{i-1} \ a_i \ a_{i+1} \ a_{i+2} \ a_{i+3}] \times \\
 & \times \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (17)
 \end{aligned}$$

To obtain the corresponding row of the identity, the row-diagram  $\times$  column-diagram interaction involving the diagrams with the same home node must be 1 and all other interactions must be zero; that is, the vector  $\mathbf{i}_{subvec}$  in (14) corresponds to  $[0 \ 0 \ 1 \ 0 \ 0]$  for (17).

This is how the entire construction proceeds: diagram interactions, with one diagram having unknown labels, lead to matrix problems from which the unknowns are determined. Solving (17)

for the  $a$ -values using  $[0 \ 0 \ 1 \ 0 \ 0]$  as a right-hand side yields

$$\left[ \frac{23}{196}, -\frac{23}{49}, \frac{9}{28}, \frac{52}{49}, \frac{9}{28}, -\frac{23}{49}, \frac{23}{196} \right]$$

It is usual to select connectivity neighborhoods incrementally by width. For  $\mathbf{A}$  we would begin with a neighborhood of width 0 (the point labeled  $a_i$  by itself) and look at the resulting  $\mathbf{P}_{submtx}$  that interacts with this lone nonzero. Then we repeat for a neighborhood of width 1. We may stop at the first width for which the corresponding  $\mathbf{P}_{submtx}$  has full rank and at least as many rows as columns. We may also continue past this width to make a wider approximation to the coarse point being sought.

The construction starts with a width that delivers a least-squares approximation of the point  $c_j$  from fine points in the smallest neighborhood about  $f_i$  for which the system (14) has a matrix (17) that is overdetermined and of full rank, and it runs incrementally up to the classical least-squares approximation of the point  $c_j$  from the entire vector of data  $\mathbf{f}$ , for which the matrix of (17) is the entire matrix  $\mathbf{P}$ .

The best width for a subdivision will depend not only on the subdivision, but also on the application to which the multiresolution will be applied. The use of the multiresolution for data compression, for example, might require a different width from a multiresolution used for multi-level model editing. The width sets the amount of data from  $f$  used in making a *local least-squares approximation* to  $c_j$ . How much is optimal for any application is an open research question; however, our paper [42] may be of use, since it offers a way of comparing how the approximations vary with width and how they compare to classical least-squares. Another approach to analyzing the issue is to note that local least-squares approximations amount to discrete-data versions of quasi-interpolation; e.g. see [44].

For the subdivision in question, we could stop successfully at width 1, but we have chosen to go on to width 3 for this example of our construction. In Section 8 we also use results for width 5 (the points labeled  $a_{i-5}, \dots, a_{i+5}$ ).



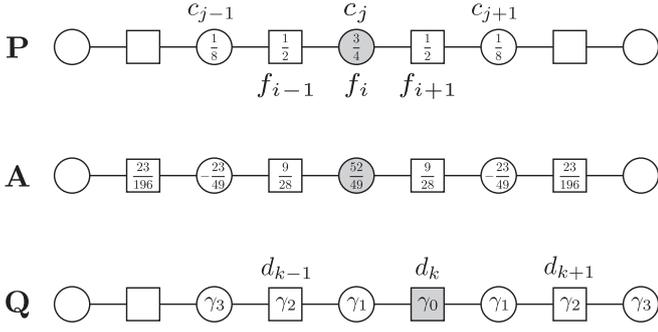


Figure 12: Multiresolution: Adding  $\mathbf{Q}$

completed by looking at the nonzero structure of the  $\mathbf{q}$  vectors alone. If dependence is found, the width of the  $\mathbf{Q}$  diagram must be increased.

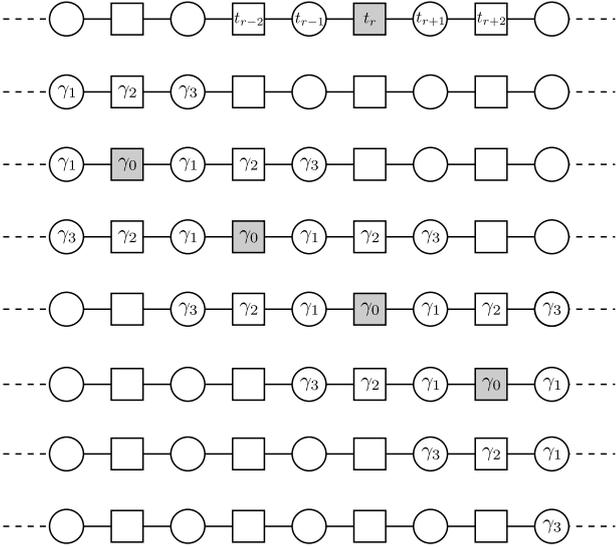


Figure 13: Interactions for  $\mathbf{T}$  and  $\mathbf{Q}$

For this example, however, the only one basis vector is found, and it is a suitable solution. It is shown in Figure 12, where

$$\begin{aligned}\gamma_0 &= -\frac{104}{2471}\sqrt{353} \\ \gamma_1 &= \frac{9}{706}\sqrt{353} \\ \gamma_2 &= \frac{46}{2471}\sqrt{353} \\ \gamma_3 &= \frac{23}{4942}\sqrt{353}\end{aligned}$$

### 6.3. The Residual Analysis $\mathbf{B}$

Finally, each row of  $\mathbf{B}$  is to be found. This is done via the intermediate step of finding a left inverse  $\mathbf{T}$  for  $\mathbf{Q}$ , and then obtaining  $\mathbf{B}$  from

$\mathbf{T}(\mathbf{I} - \mathbf{P}\mathbf{A})$ . Determining  $\mathbf{T}$  from  $\mathbf{Q}$  mimics the determination of  $\mathbf{A}$  from  $\mathbf{P}$ . Both determine left inverses of a known matrix row-by-row.

The appropriate diagrams for rows of  $\mathbf{T}$  having width 2 is shown in Figure 13. The interaction of the column diagram of  $\mathbf{Q}$  and the row diagram of  $\mathbf{T}$  with the same home node will be 1, and the interaction of all other column diagrams of  $\mathbf{Q}$  with the row diagram of  $\mathbf{T}$  will be zero. That is, the right-hand vector for (19) will be  $[0\ 0\ 1\ 0\ 0]$ .

We may proceed by incrementally investigating successive widths for  $\mathbf{T}$  row diagrams. The first width that yields a full-rank  $\mathbf{Q}_{submtx}$  as in (19) may be chosen.

$$\begin{bmatrix} t_{r-2} & t_{r-1} & t_r & t_{r+1} & t_{r+2} \end{bmatrix} \times \begin{bmatrix} \gamma_2 & \gamma_0 & \gamma_2 & 0 & 0 \\ \gamma_3 & \gamma_1 & \gamma_1 & \gamma_3 & 0 \\ 0 & \gamma_2 & \gamma_0 & \gamma_2 & 0 \\ 0 & \gamma_3 & \gamma_1 & \gamma_1 & \gamma_3 \\ 0 & 0 & \gamma_2 & \gamma_0 & \gamma_2 \end{bmatrix} \quad (19)$$

Again, it may be that the incremental process terminates only when the full mesh is involved, in which case  $\mathbf{T} = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T$ .

The solution to (19) is given in (20).

$$\begin{aligned}t_{r-2} &= t_{r+2} = -\frac{1}{112}\sqrt{353} \\ t_{r-1} &= t_{r+1} = \frac{1}{28}\sqrt{353} \\ t_r &= -\frac{3}{56}\sqrt{353}\end{aligned} \quad (20)$$

A local portion of  $\mathbf{I} - \mathbf{P}\mathbf{A}$  is shown in (21).

$$\begin{bmatrix} \frac{23}{98} & -\frac{23}{392} & 0 & 0 & 0 \\ \frac{43}{196} & -\frac{201}{1568} & \frac{23}{392} & -\frac{23}{1568} & 0 \\ -\frac{29}{98} & -\frac{43}{196} & \frac{23}{98} & -\frac{23}{392} & 0 \\ \frac{9}{28} & -\frac{29}{98} & \frac{43}{196} & -\frac{201}{1568} & \frac{23}{392} \\ -\frac{29}{98} & \frac{19}{28} & -\frac{29}{98} & -\frac{43}{196} & \frac{23}{98} \\ \frac{43}{196} & -\frac{29}{98} & \frac{9}{28} & -\frac{29}{98} & \frac{43}{196} \\ \frac{23}{98} & -\frac{43}{196} & -\frac{29}{98} & \frac{19}{28} & -\frac{29}{98} \\ \frac{23}{392} & -\frac{201}{1568} & \frac{43}{196} & -\frac{29}{98} & \frac{9}{28} \\ 0 & -\frac{23}{392} & \frac{23}{98} & -\frac{43}{196} & -\frac{29}{98} \\ 0 & -\frac{23}{1568} & \frac{23}{392} & -\frac{201}{1568} & \frac{43}{196} \\ 0 & 0 & 0 & -\frac{23}{392} & \frac{23}{98} \end{bmatrix} \quad (21)$$

The portion of  $\mathbf{B}$  in (22) corresponds to the portion of  $\mathbf{I} - \mathbf{PA}$  shown in (21).

$$\begin{bmatrix} \frac{1}{28} & -\frac{1}{112} & 0 & 0 & 0 \\ \frac{1}{28} & -\frac{3}{56} & \frac{1}{28} & -\frac{1}{112} & 0 \\ 0 & -\frac{1}{112} & \frac{1}{28} & -\frac{3}{56} & \frac{1}{28} \\ 0 & 0 & 0 & -\frac{1}{112} & \frac{1}{28} \end{bmatrix} \quad (22)$$

(**Note:** each element in (22) has been **scaled** by being divided by  $\sqrt{353}$ .) This example is slightly unusual in that  $\mathbf{B} = \mathbf{T}$ , which is not often the case.)

The final results of the construction make up the diagrams of Figure 14 for which

$$\begin{aligned} \beta_0 &= -\frac{3}{56}\sqrt{353} \\ \beta_1 &= \frac{1}{28}\sqrt{353} \\ \beta_2 &= -\frac{1}{112}\sqrt{353} \end{aligned}$$

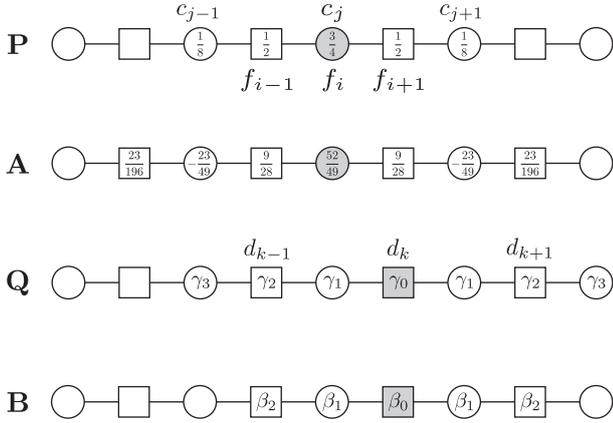


Figure 14: Multiresolution: Adding  $\mathbf{B}$

#### 6.4. Diagrams vs. Filters

When the diagrams of  $\mathbf{A}$ ,  $\mathbf{Q}$  and  $\mathbf{B}$  have been found, together with the diagrams of  $\mathbf{P}$ , they may be used directly as the decomposition and reconstruction filters of the constructed multiresolution, provided they are properly interpreted.

The column diagrams for  $\mathbf{P}$  and  $\mathbf{Q}$  act as *update filters*; that is, referring to Figure 9, for example, the fourth  $\mathbf{P}$  column diagram, whose identified

node is  $c_j$ , serves to incrementally update points  $f_{i-2}$  through  $f_{i+2}$ :

$$\begin{bmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix} + c_j \begin{bmatrix} \frac{1}{8} \\ \frac{1}{2} \\ \frac{3}{4} \\ \frac{1}{2} \\ \frac{1}{8} \end{bmatrix} \rightarrow \begin{bmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix}$$

The final position of  $f_i$  will be the result of starting with zero and applying this update associated with  $c_j$  as well as the those updates given by the  $\mathbf{P}$  column diagrams associated with  $c_{j-1}$  and  $c_{j+1}$ , since it is these three updates, and only these three, that have any influence on  $f_i$ .

Similarly, referring to Figure 12, the column diagram shown for  $\mathbf{Q}$  expresses an update for the residuals for  $f_{i-2}$  through  $f_{i+4}$ :

$$\begin{bmatrix} r_{i-2} \\ r_{i-1} \\ r_i \\ r_{i+1} \\ r_{i+2} \\ r_{i+3} \\ r_{i+4} \end{bmatrix} + d_k \begin{bmatrix} \gamma_3 \\ \gamma_2 \\ \gamma_1 \\ \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \rightarrow \begin{bmatrix} r_{i-2} \\ r_{i-1} \\ r_i \\ r_{i+1} \\ r_{i+2} \\ r_{i+3} \\ r_{i+4} \end{bmatrix}$$

The row diagrams for  $\mathbf{A}$  and  $\mathbf{B}$ , on the other hand, act as *convolution filters*; that is, referring to Figure 14, the row diagram shown for  $\mathbf{A}$  provides an approximation to the coarse point  $c_j$  as:

$$\begin{aligned} &\frac{23}{196}f_{i-3} - \frac{23}{49}f_{i-2} + \frac{9}{28}f_{i-1} \\ &+ \frac{52}{49}f_i + \frac{9}{28}f_{i+1} - \frac{23}{49}f_{i+2} \\ &+ \frac{23}{196}f_{i+3} \rightarrow c_j \end{aligned}$$

Similarly, referring to the same figure, the value of  $d_k$  is found as:

$$\beta_2 f_{i-1} + \beta_1 f_i + \beta_0 f_{i+1} + \beta_1 f_{i+2} + \beta_2 f_{i+3} \rightarrow d_k$$

## 7. Higher Dimensions: Loop in 2D

Section 5 used Loop subdivision [31] on a mesh of points of mixed degree as a way of reviewing the material on diagrams in [5]. However, it is more common to see Loop subdivision, along with many other subdivisions, used on meshes that are mostly regular. In Loop's case, regions of reg-

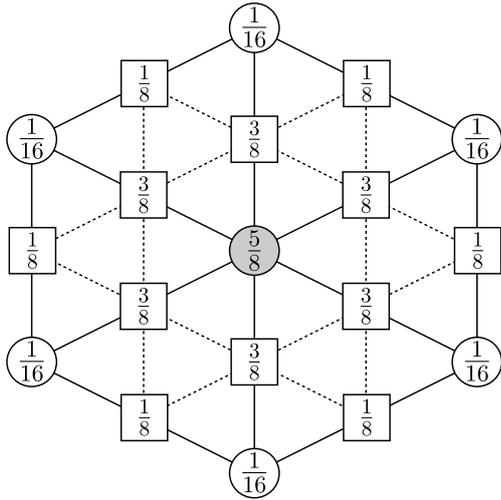


Figure 15: Loop **P** Column Diagram

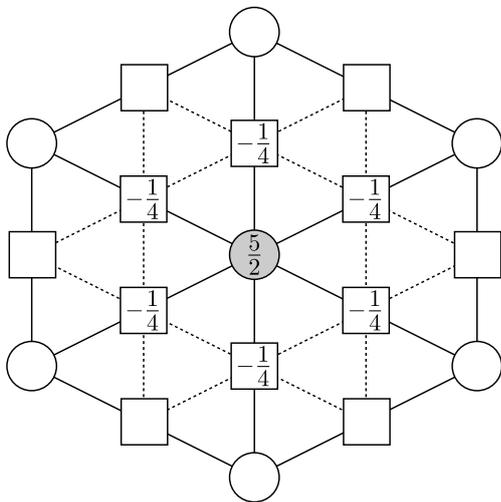


Figure 16: Loop **A** Row Diagram for Width 1

ularity correspond to 2D meshes whose domain is tessellated uniformly into triangles so that all points have degree six. Exceptional points to this are small in number; for example, points along the boundary will fall into a second class of points all of degree four, those at the corners into a third

and possibly a fourth, all of degree three or possibly of two, and there may be a few isolated points in the interior of degrees different from six. For such meshes, it is only necessary to carry out one construction for the generic situation in the interior of the region of regularity, and a few more times for exceptional points and their neighborhoods. The **A**, **Q**, and **B** that result contain only

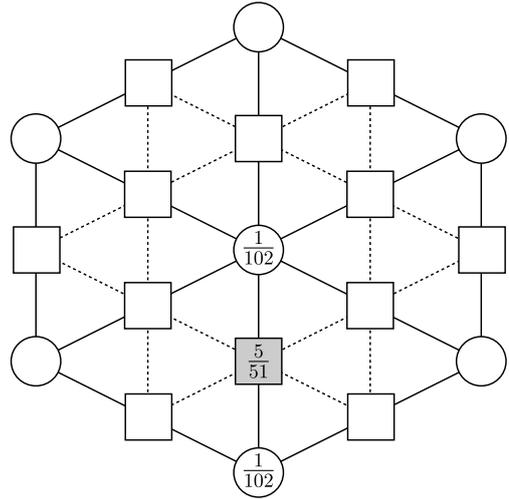


Figure 17: Loop **Q** Column Diagram for **A** in Figure 16

a few different kinds of rows/columns, which can be pre-computed and stored, and these may be re-used throughout an entire multiresolution on any appropriate mesh; for example, for infinite 1D meshes involving the subdivision of Section 6, all decompositions and reconstructions of all points at all scales of fineness/coarseness may simply use the results that were constructed, one each for **A**, **Q**, and **B** together with the given **P**. The early literature associated with function-based multiresolution addressed only the regular case, and much of the multiresolution literature for both functions and data still expects that only a few points or functions, if any, will be exceptional.

As an example, then, of the regular case in 2D and the results of our construction, we present diagrams in Figures 15 through 18 for Loop subdivision on a regular triangular mesh of infinite extent. (**Note:** The diagrams of Figures 17 through 18 have all been **scaled**. All numeric entries in those figures have been divided by  $\sqrt{102}$  for simplicity of display. **Note also:** the home positions of the **Q** and **B** diagrams correspond to node po-

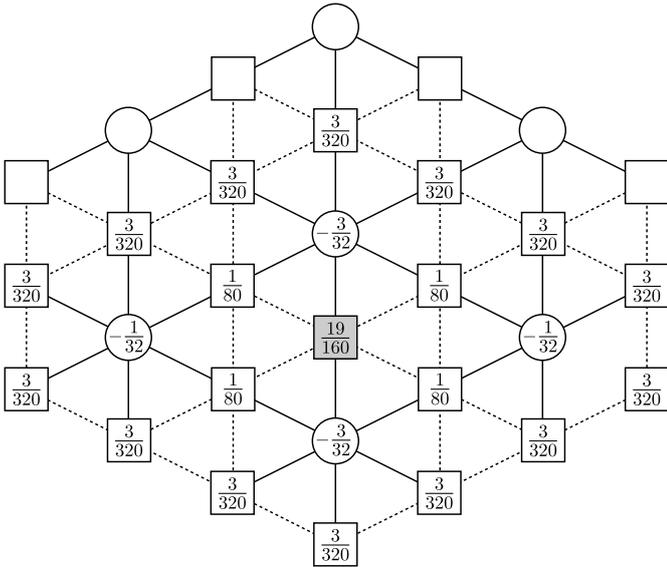


Figure 18: Loop **B** Column Diagram for **Q** in Figure 17

sitions on vertical lines. The diagrams for node positions on slanted lines, for example, would appear tilted by  $\pm 60$  degrees from the vertical.)

Further examples of 2D meshes, with attention to special points, are given in [5], where points on the boundary of a hexagonally connected mesh are studied.

## 8. A Comparison

We have established a method of finding matrices **A**, **Q**, and **B** for a given **P** so that equations (11) and (12) hold. The approach works directly with the given matrix **P** without any consideration of an underlying function space, or even whether such a function space exists. To emphasize this, we will now reveal that the subdivision **P** that we used as an example to present the construction in Section 6 is the two-scale relation of midpoint knot insertion for uniform cubic B-splines. The reader should note that *absolutely no properties of splines*, “B” or otherwise, “cubic” or otherwise, were used or needed in our example construction beyond the 2-scale relation.

In this section we compare the results of our example construction with the multiresolution due to Quak and Weyrich [45]. It is a convenient comparison to make, because both are based on the same subdivision system. The multiresolution due to Quak and Weyrich carries the

semiorthogonal spline wavelets from the book by Chui [12] over to closed, bounded intervals, and the wavelet multiresolution is deeply dependent on the function-space properties of B-splines. Our multiresolution uses as **P** only the 2-scale relation for cubic B-splines, and it ignores all other properties of splines. If our method can compare in any modest way, perhaps it has merit.

As for the comparison application, spline wavelets have been used for graphics most notably in image processing and image compression, so we chose a brief comparison in this area of application, namely the one given in [45], as a sensible one to offer. This choice has the added advantage of indicating that applications for our multiresolutions to pixel data are as possible as applications to geometric point data, even though most of our discussion here, and our previous published applications; e.g., [4, 5, 41, 43], was motivated by geometric data.

We make the comparison in three ways: (1) visual differences in a processed image, (2) quality of approximating a coarse image **c** from a given image **f** in terms of the size of  $\mathbf{r} = \mathbf{P}\mathbf{c} - \mathbf{f}$ , and (3) quality of producing an approximate fine image  $\hat{\mathbf{f}}$  from compressed **d** in terms of the size of  $\hat{\mathbf{f}} - \mathbf{f}$ .

The multiresolution proposed by Quak and Weyrich uses cubic B-splines on the unit interval. The knot structure and its refinement pattern are as follows: at stage  $k$ , the spline knots are at locations  $\frac{i}{2^k}$  for  $i = 1, \dots, 2^k - 1$ , but with the first, the last, and the middle of these knots removed. Additionally, 0 and 1 always appear as quadruple knots. Such a knot structure supports  $2^k$  B-splines for each positive integer  $k \geq 3$ , and the B-splines for  $k$  form a basis for a space containing the space generated by the B-splines for  $k - 1$  as a proper subspace. The corresponding



Figure 19: Tessellation: 4 to 8 Nodes

mesh-domain tessellation is exemplified by Figure 19. That is; in words, each interval except the middle one is split into two, and the middle interval is split into three.

The  $2^k \times 2^{k-1}$  matrix that expresses the B-splines,  $\phi_j^{k-1}$ , for  $k-1$  as linear combinations of the B-splines,  $\phi_i^k$ , for  $k$  is the transpose of the matrix  $\mathbf{P}^k$  for the Quak-Weyrich multiresolution. The columns of this matrix are described in Table 2.3 of [45] (though there is one error: the entry for  $k=4$  should have the value  $l=6$ ). The transpose of the  $\mathbf{Q}^k$  matrix corresponding to this  $\mathbf{P}^k$  is the matrix that expresses the semiorthogonal, cubic B-spline wavelets,  $\psi_j^{k-1}$ , for  $k-1$ , in terms of those for  $k$ ,  $\psi_j^k$ . The elements of  $\mathbf{Q}^k$  are given in Table 2.4 of [45]. We have chosen to scale the columns of the Quak-Weyrich matrix  $\mathbf{Q}^k$  so that each column has Euclidian norm 1. This induces a corresponding scaling on the rows of  $\mathbf{B}^k$ , but the scalings are inessential to any of the properties of the multiresolution. Since our construction, if implemented using the Singular Value Decomposition [6], conveniently produces a  $\mathbf{Q}$  with unit-length columns, this will make the resulting detail coefficients for the two methods more comparable.

The Quak-Weyrich matrix  $\mathbf{A}^k$  can be expressed as  $(\Phi^k)^{-1} \mathbf{P}^{kT} \Phi^k$ , where  $\Phi^k$  is the Gram matrix of the B-splines with the special knot pattern described; that is, the matrix whose  $r, s$  element is  $\int \phi_r^k \phi_s^k$ . The final matrix  $\mathbf{B}^k$  can be expressed as  $(\Psi^k)^{-1} \mathbf{Q}^{kT} \Phi^k$ , where  $\Psi^k$  is the Gram matrix of the wavelets. Neither the  $\mathbf{A}$  nor the  $\mathbf{B}$  filters in this system are finite; that is, all elements of the  $\mathbf{A}^k$  and  $\mathbf{B}^k$  matrices are nonzero, but their values decay in magnitude quite quickly off the “diagonal” positions  $(i, 2i)$ . For efficiency of implementation, the matrices  $\mathbf{A}^k$  and  $\mathbf{B}^k$  are not formed. Rather the factors  $(\Phi^k)^{-1}$  and  $(\Psi^k)^{-1}$  are applied through backsolving via their  $LU$  factorizations. This converts the application of  $\mathbf{A}^k$  and  $\mathbf{B}^k$  to any set of data as a sequence of multiplications/backsolves involving banded matrices. Nevertheless, each application involves processing every item of the fine data, unless the filters are truncated, in which case the multiresolution reconstruction will involve some error. Many multiresolutions constructed from function-space tools do require truncation to be practical; e.g., as in [12, 17]. For our multiresolution, by contrast,

no truncation is required; all filters are finite and even relatively short.

For our multiresolution we apply the construction beginning with  $\mathbf{P}^k$  and use the connectivity induced by the Quak-Weyrich system of knots. Their version of  $\mathbf{P}$  corresponds to the one we have used in our example construction throughout Section 6, except in the first four, the last four, and the middle four columns in order to reflect their special knot structure. Choosing the  $\mathbf{A}$  filters to be of width 3, most of the elements of our  $\mathbf{A}^k$  will be the same as those in Subsection 6.1. The only exceptions are for the first three, last three, and middle four rows, since these must reflect the special knot structure in these positions. Similar statements hold for the remaining filters. We shall also show some results for a version of our multiresolution for which the width of the  $\mathbf{A}$  filters is chosen to be 5.



Figure 20:  $512 \times 512$  Image of Lena

The example results all use tensor-product data, so a single stage of the decomposition process is as indicated in (23).

$$\begin{aligned}
 \mathbf{G} &\leftarrow \mathbf{A}\mathbf{F} \\
 \mathbf{C} &\leftarrow \mathbf{G}\mathbf{A}^T \\
 \mathbf{H} &\leftarrow \mathbf{B}\mathbf{F} \\
 \mathbf{D} &\leftarrow \mathbf{G}\mathbf{B}^T
 \end{aligned}
 \tag{23}$$

and a single stage of the reconstruction process is

as indicated in (24).

$$\begin{aligned} \mathbf{M} &\leftarrow \mathbf{C}\mathbf{P}^T + \mathbf{D}\mathbf{Q}^T \\ \mathbf{F} &\leftarrow \mathbf{P}\mathbf{M} + \mathbf{Q}\mathbf{H} \end{aligned} \quad (24)$$

The fine geometric data is  $\mathbf{F}$ , the coarse geometric data is  $\mathbf{C}$ , and  $\mathbf{D}$  and  $\mathbf{H}$  represent the detail information.  $\mathbf{G}$  and  $\mathbf{M}$  are intermediate quantities that may be discarded.



Figure 21: Quak-Weyrich Compression

B-spline wavelet multiresolutions are important for image compression. An image is decomposed one or more times to provide a coarse image,  $\mathbf{C}$ , and several stages of detail information,  $\mathbf{D}$  and  $\mathbf{H}$ . As many of the less important components of the detail information are then set to zero as can be done without visually impacting the version of the full-sized image that can be reconstructed from  $\mathbf{C}$  and the adjusted detail information. The more detail information that can be set to zero, the higher the compression and the better the multiresolution is regarded to be. The current “gold standard” in this regard, at least according to Unser [46], is represented by the multiresolutions based upon spline wavelets, so we should expect the Quak-Weyrich multiresolution to be a very good test for our multiresolution.

Figure 20 shows the image used by Quak and Weyrich, the well-known  $512 \times 512$  black and white version of “Lena” that has been used in a wide number of other papers as well.



Figure 22: Our Compression with Width-3  $\mathbf{A}$

Figure 21 shows the result of decomposing this image by Quak-Weyrich through two stages to produce a  $128 \times 128$  image together with  $128 \times 128$  and  $256 \times 256$  detail information,  $\mathbf{D}_{128}$  and  $\mathbf{D}_{256}$  respectively, and  $128 \times 256$  and  $256 \times 512$  detail information  $\mathbf{H}_{128}$  and  $\mathbf{H}_{256}$  respectively. All elements of  $\mathbf{D}_{256}$  and  $\mathbf{H}_{256}$  were set to zero (this had no visible effect on the reconstruction), and as many of the smaller elements of  $\mathbf{D}_{128}$  and  $\mathbf{H}_{128}$  were set to zero as possible without causing noticeable degradation in the reconstruction. Figure 21 represents a reconstructed image using only the  $128 \times 128$  coarse image, 8243 elements of  $\mathbf{D}_{128}$  and 13663 elements of  $\mathbf{H}_{128}$  for a total of 38290 elements out of a possible  $512 \times 512 = 262144$ ; that is, only 14.6 per cent of the amount of the original image data. Removing any more detail information leads to some faint but noticeable artifacts. Figure 22 shows what is achievable with our multiresolution using a width-3  $\mathbf{A}$ ; that is, using the results of the construction in Section 6. We set detail information to zero in the style described above for Quak-Weyrich compression to achieve an image with only 14.6 per cent of the amount of the original data, which is the same level of compression as that of Figure 21. The quality of our result is slightly lower than that for Quak-Weyrich.

Figure 23 shows a compression with 14.6 per cent of the amount of the original data achieved



Figure 23: Our Compression with Width-5  $\mathbf{A}$

with our multiresolution using 5 as the width of the  $\mathbf{A}$  filter. There has been some slight improvement in the quality of the image, suggesting that longer  $\mathbf{A}$  filters might yield even better results. Indeed, the width of the Quak-Weyrich  $\mathbf{A}$  filter is 256 for Figure 21 (as is also the case for the  $\mathbf{B}$  filter), and truncated versions of the Quak-Weyrich filters, which might be used in practical applications, reduce their compression quality. The length of the  $\mathbf{A}$  filter does correlate well with the quality of compression. We will maintain the full filter length in the Quak-Weyrich examples below, even though this may reflect better results for their multiresolution than would be achieved in practice, where truncated filters might be used.

Several numeric comparisons show that Quak-Weyrich multiresolution and our multiresolution are not that far different. To measure how well each multiresolution produces a coarse  $\mathbf{c}$  approximation from a fine  $\mathbf{f}$  in the sense that  $\|\mathbf{f} - \mathbf{P}\mathbf{c}\|$  is small, we produced a simple reduction to  $256 \times 256$  and subsequent subdivision to  $512 \times 512$  with both multiresolutions; that is,

$$\mathbf{C} \leftarrow \mathbf{A}\mathbf{F}\mathbf{A}^T \quad \text{and} \quad \mathbf{P}\mathbf{C}\mathbf{P}^T \rightarrow \hat{\mathbf{F}}$$

The root-squared error; that is,

$$\sqrt{\sum_{i=1}^{512} \sum_{j=1}^{512} |\mathbf{F}_{i,j} - \hat{\mathbf{F}}_{i,j}|^2}$$

for Quak-Weyrich multiresolution was 2,773.2, while for our multiresolution with an  $\mathbf{A}$  filter of width 5 it was 2,816.4. Since there are 262,144 terms in the sum of squares, this represents an average per-pixel error difference of 0.000165 between the two compressions. That is, there is virtually no difference in the quality of approximation  $\mathbf{C}$  provided from  $\mathbf{F}$  by the Quak-Weyrich  $\mathbf{A}$  filter of width 256 and our  $\mathbf{A}$  filter of width 5.

If  $\hat{\mathbf{F}}$  is replaced by the images of Figures 21 and 23, we have a root-squared error comparison of the images for comparable levels of compression. In this case, for our compression the root-squared error was 4,577.5, which was a smaller error than the value of 5,560.0 due to Quak-Weyrich. A further comparison of the degree of similarity between the two compressions is given by a histogram of the per-pixel errors in the images of Figures 21 and 23 shown by Figure 24. The his-

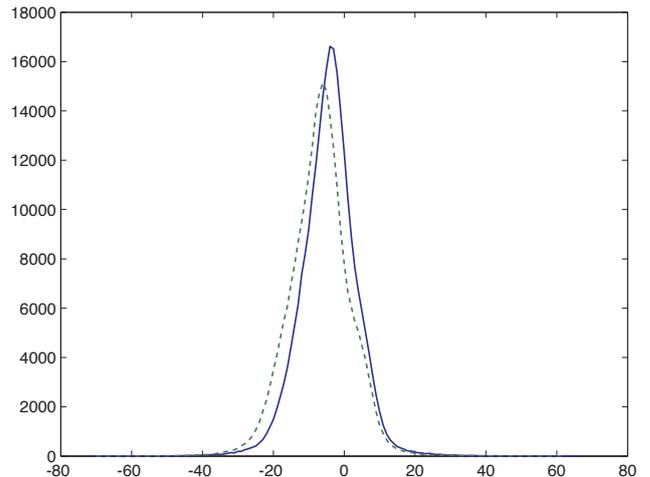


Figure 24: Pixel-Error Histograms

together shows on the  $y$  axis the number of pixels that are at a difference shown on the  $x$  axis from the pixels of the original image. The solid histogram line is from our compression, and the dashed histogram line is from the Quak-Weyrich compression. Pixel values range from 0 to 255, and both methods have almost all pixel differences within  $[-40, +20]$ ; that is, a less than 16% error in almost all pixels for either method. Our compression has delivered noticeably more pixels with error closer to zero, and with a smaller bias towards negative pixel errors.

Ideally, readers should be able to generate for themselves the filters corresponding to our width-7  $\mathbf{A}$  for the example of this section. It is a good way to test understanding of the construction. However, a MATLAB script [47] for assembling  $\mathbf{P}$ ,  $\mathbf{A}$ ,  $\mathbf{Q}$ , and  $\mathbf{B}$  in full matrix format can be obtained at

<http://www.cgl.uwaterloo.ca/~rhhbartel/QW/QW.m>

No special properties of MATLAB have been used, so the script can also be regarded as simple pseudo-code for easy transcription into other languages.

## 9. Summary

We have provided a numerical construction for multiresolutions based upon [4, 5], producing multiresolutions from given scale relationships  $\mathbf{P}$  without requiring any knowledge of an underlying function space. It employs only numerical computations. Use is made of the connectivities in the fine and course mesh. This approach allows for the ready application on different mesh geometries, even at special points, and we have given a comparison with a well-known, function-based multiresolution to show that the quality of the multiresolutions resulting from our construction can be quite acceptable.

## Acknowledgments

This work was carried out under funding by the Natural Sciences and Engineering Research Council of Canada. We wish to thank the referees for a number of helpful, constructive comments.

## References

- [1] Farin G. *Curves and Surfaces for GAGD, A Practical Guide*. Morgan Kaufmann; 5 ed.; 2002.
- [2] Stollnitz EJ, DeRose TD, Salesin DH. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers; 1996. ISBN 1-55860-375-1.
- [3] Warren J, Weimer H. *Subdivision Methods for Geometric Design : A Constructive Approach*. Morgan Kaufmann series in computer graphics and geometric modeling; San Francisco: Morgan Kaufmann; 2002.

- [4] Bartels RH, Samavati FF. Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics* 2000;119(1–2):29–67.
- [5] Samavati FF, Bartels RH. Diagrammatic tools for generating biorthogonal multiresolutions. *International Journal of Shape Modeling* 2006;12(1):47–73.
- [6] Golub GH, Loan CFV. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences; The Johns Hopkins University Press; 1996. ISBN 0801854148.
- [7] Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J, Du Croz J, et al. *LAPACK's user's guide*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics; 1992.
- [8] Daubechies I. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics: 61; SIAM; 1992. ISBN 0-89871-274-2.
- [9] Gori L, Pitolli F. Multiresolution analyses originated from nonstationary subdivision schemes. *Journal of Computational and Applied Mathematics* 2008;221(2):406–15.
- [10] Dahmen W, Micchelli CA. Biorthogonal wavelet expansions. *Constructive Approximation* 1997;13(3):293–328.
- [11] Cohen A, Dyn N. Nonstationary subdivision schemes and multiresolution analysis. *SIAM Journal on Mathematical Analysis* 1996;27(6):1745–69.
- [12] Chui CK. *An Introduction to Wavelets; vol. 1 of Wavelet Analysis and its Applications*. Academic Press, Inc.; 1992. ISBN 0-12-174584-8.
- [13] Chui CK, Quack E. Wavelets on a bounded interval. In: Braess D, Schumaker LL, editors. *Numerical methods in approximation theory; vol. 9 of International Series of Numerical Mathematics, 105*. Birkhäuser, Basel. ISBN 3-7643-2746-4; 1992, p. 53–75.
- [14] Hoppe H. Progressive meshes. In: *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM; 1996, p. 99–108.
- [15] Schroeder WJ, Zarge JA, Lorensen WE. Decimation of triangle meshes. In: *Computer Graphics (SIGGRAPH '92 Proceedings); vol. 26. ACM SIGGRAPH; 1992, p. 65–70*.
- [16] Xu W, Hammersley R, Lu K. Lossless subdivision-based multiresolution representation of arbitrary triangle meshes using kite trees. *International Journal of Foundations of Computer Science* 2002;13(2):243–60.
- [17] Lounsbery M, DeRose TD, Warren J. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics* 1997;16(1):34–73.
- [18] Sweldens W. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In: *Proc. of SPIE. SPIE; 1995, p. 68–79*.
- [19] Sweldens W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and*

- Computational Harmonic Analysis 1996;3(2):186–200.
- [20] Sweldens W. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis* 1998;29(2):511–46.
- [21] Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps. *J Fourier Anal Appl* 1998;4(3):247–69.
- [22] Schröder P, Sweldens W. Spherical wavelets: efficiently representing functions on the sphere. In: *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM; 1995, p. 161–72.
- [23] Li D, Qin KH, Sun H. Unlifted loop subdivision wavelets. In: *12th Pacific Conference on Computer Graphics and Applications*. 2004, p. 25–33.
- [24] Wang H, Qin KH, Tang K. Efficient wavelet construction with catmull-clark subdivision. *The Visual Computer* 2006;22:874–84.
- [25] Bertram M, Duchaineau M, Hamann B, Joy K. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In: *VIS '00: Proceedings of the conference on Visualization '00*. IEEE Computer Society Press; 2000, p. 389–96.
- [26] Bertram M. Biorthogonal wavelets for subdivision volumes. In: *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications*. New York, NY, USA: ACM; 2002, p. 72–82.
- [27] Bertram M. Biorthogonal loop-subdivision wavelets. *Computing* 2004;72(1-2):29–39.
- [28] Olsen L, Samavati F, Bartels R. Multiresolution for curves and surfaces based on constraining wavelets. *Computers & Graphics* 2007;31(3):449–62.
- [29] Stam J, Loop C. Quad/triangle subdivision. *Computer Graphics Forum* 2003;22(1):1–7.
- [30] Stam J. On subdivision schemes generalizing uniform b-spline surfaces of arbitrary degree. *Computer Aided Geometric Design* 2001;18(5):383–96.
- [31] Loop C. Smooth subdivision surfaces based on triangles. Master's thesis; Department of Mathematics, University of Utah; 1987.
- [32] Doo D. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In: *Proced. Int'l Conf. Interactive Techniques in Computer Aided Design*. 1978, p. 157–65. Bologna, Italy, IEEE Computer Soc.
- [33] Dyn N, Levin D, Gregory J. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design* 1987;4(4):257–68.
- [34] Dyn N, Levin D, Gregory JA. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics* 1990;9(2):160–9.
- [35] Ma W. Subdivision surfaces for cad: an overview. *Comput Aided Des* 2005;37(7):693–709.
- [36] Certain A, Popović J, DeRose T, Duchamp T, Salesin D, Stuetzle W. Interactive multiresolution surface viewing. In: Rushmeier H, editor. *SIGGRAPH 96 Conference Proceedings*. Annual Conference Series; ACM SIGGRAPH; Addison Wesley; 1996, p. 91–8. Held in New Orleans, Louisiana, 04-09 August 1996.
- [37] Kobbelt L, Campagna S, Vorsatz J, Seidel HP. Interactive multi-resolution modeling on arbitrary meshes. In: Cohen M, editor. *SIGGRAPH '98 Conference Proceedings*. Annual Conference Series; ACM SIGGRAPH; Addison Wesley. ISBN 0-89791-999-8; 1998, p. 105–14.
- [38] Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology. In: *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM; 1996, p. 189–92.
- [39] Finkelstein A, Salesin DH. Multiresolution curves. In: Glassner A, editor. *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*. Computer Graphics Proceedings, Annual Conference Series; ACM SIGGRAPH; ACM Press. ISBN ISBN 0-89791-667-0; 1994, p. 261–8.
- [40] Strang G, Nguyen T. *Wavelets and Filter Banks*. Wellesley-Cambridge Press; 1996. ISBN 0-9614088-7-1.
- [41] Samavati F, Bartels R, Olsen L. *Synthesis and Analysis in Biometrics*; vol. 67; chap. Local B-spline Multiresolution with Examples in Iris Synthesis and Volumetric Rendering. World Scientific Publishing; 2006, p. 65–101.
- [42] Bartels R, Golub G, Samavati F. Some observations on local least squares. *BIT* 2006;46(3):455–77.
- [43] Samavati F, Mahdavi-Amiri N, Bartels R. Multiresolution surfaces having arbitrary topologies by a reverse doo subdivision method. *Computer Graphics Forum* 2002;21(2):121–36.
- [44] Schumaker L, Lai MJ. *Spline Functions on Triangulations*. Cambridge University Press; 2007.
- [45] Quak E, Weyrich N. Algorithms for spline wavelet packets on an interval. *BIT* 1997;37(1):76–95.
- [46] Unser M. Splines : A perfect fit for signal/image processing. *IEEE Transactions on Signal Processing* 1999;16:22–38.
- [47] Higham DJ, Higham NJ. *The MATLAB Guide*. SIAM; 2000.