$\sqrt{3}$ Multiresolution by Local Least Squares: The Diagrammatic Approach

Richard Bartels
Ali Mahdavi-Amiri
Faramarz Samavati

October 19, 2015

Abstract

In [2, 3, 20, 21] the authors explored a construction to produce multiresolutions from given subdivisions. Certain assumptions carried through that work, two of which we wish to challenge: (1) that multiresolutions for irregular meshes have to be constructed on the fly rather than being prepared beforehand and (2) that the connectivity graph of the coarse mesh would have to be a subgraph of the connectivity graph of the fine mesh. Kobbelt's $\sqrt{3}$ subdivision [11] lets us engage both of these assumptions. With respect to (2), the $\sqrt{3}$, post-subdivision connectivity graph shares no interior edges with the pre-subdivision connectivity graph. With respect to (1), we observe that subdivision does not produce an arbitrary connectivity graph. Rather, there are local regularities that subdivision imposes on the fine mesh that are exploitable to establish, in advance, the decomposition and reconstruction filters of a multiresolution for an irregular coarse mesh.

keywords: $\sqrt{3}$, subdivision, least squares, biorthogonal, multiresolution.

1 Introduction

Multiresolution frameworks have found many applications in science and technology including computer graphics, geospatial visualization, and large scale visualization. In these frameworks, objects can be represented, visualized, and analyzed at multiple level of details. This property provides the possibility of working with an overall shape of a very large model. In order to not lose any information in a multiresolution framework, it is desired that the framework provides the possibility of a perfect reconstruction. One way to provide such a multiresolution is to reverse the process of subdivision in which a mesh is smoothed by inserting new vertices and faces and averaging the vertices using a local smoothing mask.

Many subdivision schemes have been proposed for graphical objects including Doo-Sabin, Catmull-Clark, Loop, and $\sqrt{3}$ subdivision [6, 5, 14, 11]. $\sqrt{3}$ subdivision has the smallest factor of growth in the number of faces for triangular meshes [1]. As a result, $\sqrt{3}$ subdivision produces more resolution levels under a fixed maximum number of faces in comparison with other subdivision schemes proposed for triangular meshes. In this paper, we provide a new multiresolution framework for $\sqrt{3}$ subdivision based on diagrammatic approach initially proposed in [3]. However, in this paper, we have improved the method proposed in [3] by providing simple multiresolution filters that perform on vertices with arbitrary valence.

2 Related Work

Multiresolution provides a means to transition from high to low resolution and vice versa [22]. Many multiresolution frameworks have been proposed to reduce the resolution of an object with applications such as mesh morphing, compression, or view dependent rendering [13, 10, 9, 7, 22]. One way to make a multiresolution framework is to reverse the process of subdivision (i.e. reverse subdivision process) [17, 18, 4] or by considering a property of the coarse vertices like smoothness via the Laplacian [26]. Reverse subdivision is the process of finding the coarse approximation of a fine model. Olsen et al. [17] propose a method called constraining wavelets in which vertices resulting from subdivision methods with 1-to-4 refinements are categorized into *even* and *odd* vertices. Even vertices represent the old vertices that are repositioned using vertex-vertex masks and odd vertices are new vertices that are inserted to split faces or edges. The important fact about this type of multiresolution is that the details of even vertices are a linear combination of the details stored in odd vertices. Having such a categorization of vertices, they also mention the possibility of an efficient data structure in which odd vertices are replaced by their details and coarse approximations are placed at even vertices.

2.1 Constraining Wavelets

Reverses subdivision process in constraining wavelet approach can be divided into two parts of finding coarse approximations and smoothing. In this process, details (d_0s) and coarse approximations ($\tilde{c}_0$s) are initially determined. Afterwards, c_0 is smoothed to get better approximations using an optimization method that provides vector δ_0 to be added to ($c_0 = \tilde{c}_0 + \delta_0$). In [17, 18], the filters of Loop and Catmull-Clark are obtained using constraining wavelets. Mahdavi-Amiri and Samavati have used the same approach to find filters of $\sqrt{3}$, and $\sqrt{2}$ reverse subdivision [15, 16].

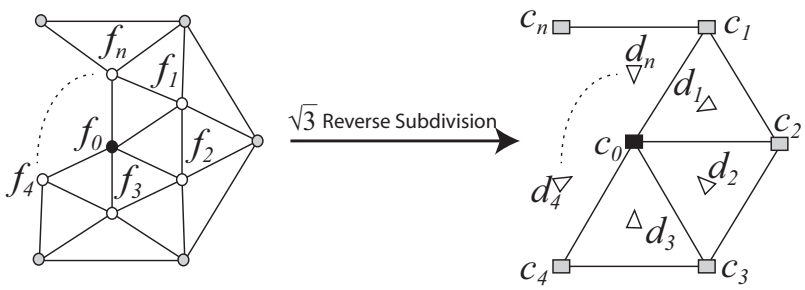

Figure 1: Left: f_0 is an old vertex and f_is are new vertices. Right: after one level of reverse subdivision, f_0 is replaced by its refined coarse approximation, c_0, and f_is are replaced by their details, d_is.

2.1.1 $\sqrt{3}$ Reverse Subdivision

Using the method proposed in [17], the following equations are obtained for $\sqrt{3}$ reverse subdivision. Similar to the Loop's case, the details of vertex-vertices (d_0) are determined by a linear combination of the details of face vertices (d_i) in their neighborhood (Figure 1):

$$d_0 = \frac{3}{2n}\alpha \sum_{i=1}^{n} d_i. \tag{2.1}$$

The coarse vertex of a face vertex can be obtained using the vertices in its neighborhood:

$$\tilde{c_0} = \frac{1}{1-\frac{3}{2}\alpha} f_0 - \frac{\alpha}{n(\frac{2}{3}-\alpha)} \sum_{i=1}^{n} f_i. \tag{2.2}$$

δ_0, which is used to reduce the magnifying effect of reverse subdivision is:

$$\delta_0 = \frac{\frac{3}{2n}(1-\alpha)+\frac{1}{3}}{(1-\alpha)^2+\frac{n}{9}} \sum_{i=1}^{n} d_i. \tag{2.3}$$

Note that $\alpha = \frac{4-2cos(\frac{2\pi}{n})}{9}$ is the parameter to find the position of vertex-vertices in $\sqrt{3}$ subdivision [12].

2.2 Smooth Reverse Subdivision

We discussed that δ_0 is used to reduce the magnifying effect of reverse subdivision. However, the resulting meshes of reverse subdivision can be still very coarse. As a result, smooth reverse subdivision is proposed to improve the results. The basic idea of smooth reverse

subdivision is to perturb coarse approximations by vector δ to minimize an energy function $E_{total}(\Delta) = \omega E_{sub}(\Delta) + (1-\omega)E_{smooth}(\Delta)$ in which E_{sub} is the euclidean distance between fine points and subdivided coarse approximations and E_{smooth} is the energy of coarse approximations in the local neighborhood. Perturbation vectors for Catmull-Clark and Loop subdivision are proposed in [19].

Using this method, Mahdavi-Amiri and Samavati [15, 16] provided filters for smooth $\sqrt{3}$ and $\sqrt{2}$ reverse subdivision methods and find perturbation vectors $\delta_{\sqrt{3}}$ as:

$$\delta_{\sqrt{3}} = \frac{(\omega(1-\alpha)\frac{3\alpha}{2n} + \frac{1}{3}\omega)\sum_{i=1}^{n} d_i}{\omega(\frac{n}{9} + (1-\alpha)^2) + (1-\omega)} + \frac{\frac{1-\omega}{n}\sum_{i=1}^{n} c_i - (1-\omega)c_0}{\omega(\frac{n}{9} + (1-\alpha)^2) + (1-\omega)}. \tag{2.4}$$

2.3 Biorthogonal Wavelets

The condition for having biorthogonal wavelets is to make $[P|Q]$ invertible [2, 3, 20, 21]. A common approach to make biorthogonal wavelets is lifting scheme proposed by Sweldens [24, 23]. In the lifting process, an initial set of P, Q, A, and B are constructed. Then, lifting matrix L modifies the original scheme as:$Q_{lift} = A + LB$ and $A_{lift} = Q - PL$. Matrices P and B are unchanged. In [25], lifting scheme has been used to derive biorthogonal wavelets for $\sqrt{3}$ subdivision. Two sets of filters have been proposed in [25] with considering one ring and two rings of neighborhood to set up the lifting equalities.

3 Overview

The approach presented in [2, 3, 20, 21] follows the outline for developing biorthogonal multiresolutions that one finds in the wavelet literature; e.g. as covered in [8]. As in much of the wavelet literature, the main attention in these publications was paid to constructing multiresolutions in regular settings. However, the publication that appeared latest [3] attempted to phrase the material in a way that would suggest how numerical constructions could be carried out, on the fly, in irregular settings.

The construction in [2, 3, 20, 21] is entirely data-oriented and intended for use in geometric subdivisions. No basis functions are part of the process, and no results appear as scale or wavelet functions or their duals. The authors assume only that a *subdivision* is defined, which may be expressed using a *subdivision matrix* $\mathbf{P}$, that is applied to a *coarse mesh* of points $\mathbf{c}$ to produce a *fine mesh* of points $\mathbf{f}$.

$$\mathbf{f} = \mathbf{P}\mathbf{c} \tag{3.1}$$

The construction first finds a *reversal matrix*, $\mathbf{A}$, that satisfies $\mathbf{AP} = \mathbf{I}$, and consequently

$$\mathbf{c} = \mathbf{A}\mathbf{f} \tag{3.2}$$

The matrices $\mathbf{P}$ for subdivisions are sparse and patterned. Specifically, each fine point is produced as an affine combination of coarse points lying in a small geometric neighborhood. The coefficients of the combination constitute the entries in the row of $\mathbf{P}$ that yields the fine point.

The construction of the reversal in [2, 3, 20, 21] uses a technique called *local least squares fitting* to find $\mathbf{A}$ one row at a time. The κ^{th} row $\mathbf{a}_\kappa$ of $\mathbf{A}$ provides an approximation of the position of the coarse point c_κ whose displaced-point image in the fine mesh is f_k. To construct $\mathbf{a}_\kappa$, a geometric neighborhood of fine points is taken that includes most or all of the fine points around f_k whose positions are defined by affine combinations involving c_κ. The neighborhood chosen establishes a submatrix $\mathbf{P}_S$ of $\mathbf{P}$ and corresponding subvectors $\mathbf{f}_S$ and $\mathbf{c}_S$ of $\mathbf{f}$ and $\mathbf{c}$. The point c_κ is least-squares estimated from $\mathbf{f}_S$ and $\mathbf{P}_S$. Other components of $\mathbf{c}_S$ are also estimated, of course, but their estimates are discarded. Each of the other components will be estimated in establishing their corresponding rows of $\mathbf{A}$. The resulting $\mathbf{A}$ is sparse and patterned.

If a mesh $\mathbf{g}$ is encountered with a connectivity graph consistent with the subdivision $\mathbf{P}$, yet whose points were not produced according to (3.1) from any $\mathbf{c}$, then for the particular $\mathbf{c}$ we would obtain according to (3.2), we would have

$$\mathbf{g} + \mathbf{e} = \mathbf{P}\mathbf{c} \tag{3.3}$$

To correct for the error $\mathbf{e}$, the construction finds two more matrices, the *detail matrix* $\mathbf{B}$ and the *correction matrix* $\mathbf{Q}$, such that

$$\mathbf{d} = \mathbf{B}\mathbf{g} \tag{3.4}$$

and

$$\mathbf{e} = \mathbf{Q}\mathbf{d} \tag{3.5}$$

The rows of the matrices $\mathbf{A}$ and $\mathbf{B}$ provide *decomposition filters* that may be applied to a fine mesh $\mathbf{g}$ to obtain, respectively, a *coarse approximation* $\mathbf{c}$ to $\mathbf{g}$ and *detail information* $\mathbf{d}$ for $\mathbf{g}$. The rows of the matrices $\mathbf{P}$ and $\mathbf{Q}$ provide *reconstruction filters* that yield $\mathbf{g}$ as

$$\mathbf{g} = \mathbf{P}\mathbf{c} + \mathbf{Q}\mathbf{d} \tag{3.6}$$

The matrix $\mathbf{Q}$ is generated from $\mathbf{A}$ one column at a time. Each column of $\mathbf{Q}$ starts with a geometric neighborhood, which determines a submatrix of $\mathbf{A}$, and the submatrix, in turn, defines a nullspace of $\mathbf{A}$ in which the column of $\mathbf{Q}$ is required to lie. The $\mathbf{Q}$ resulting from determining all columns in this way is sparse and patterned.

The matrix $\mathbf{B}$ is generated in two steps. A matrix $\mathbf{T}$ is constructed from $\mathbf{Q}$, row by row, as a left inverse, $\mathbf{T}\mathbf{Q} = \mathbf{I}$ in the same manner as $\mathbf{A}$ was constructed from $\mathbf{P}$. Then $\mathbf{B} = \mathbf{T}(\mathbf{I} - \mathbf{P}\mathbf{A})$. The $\mathbf{B}$ that results is sparse and patterned.

Let $\#c$ be the number of coarse points and $\#f$ be the number of fine points. Then $\mathbf{P}$ has dimensions $\#f \times \#c$ and $\mathbf{A}$ has $\#c \times \#f$. $\mathbf{Q}$ has dimensions $\#f \times (\#f - \#c)$ and $\mathbf{B}$

has dimensions $(\#f - \#c) \times \#f$. These dimensions reflect the fact that the subdivision maps the positions of the $\#c$ coarse points onto the $\#c$ positions of fine-mesh *displaced points* and the fact that the subdivision creates $(\#f - \#c)$ new *introduced points* into the fine mesh.

Each column of $\mathbf{P}$ is one-to-one associated with a coarse point c_κ, yet we will refer to the κ-indexed column by the index k, the index of the fine point f_k to which the subdivision displaces c_κ. This is done to relate all work in the construction to the connectivity diagrams of the fine mesh. Likewise, each row of $\mathbf{A}$ is associated with a coarse point c_κ, but we will refer to the κ-indexed row by the index k in this same manner. Each row of $\mathbf{B}$ and each column of $\mathbf{Q}$ are associated with a unique introduced fine point f_i, and each such row or column will be referred to with this index i.

Part of the advantage of a multiresolution is that the vector $\mathbf{e}$, which must have $\#f$ components, is constructible from the vector $\mathbf{d}$ that has only $(\#f - \#c)$ components, and so the fine mesh $\mathbf{g}$ can be encoded as a coarse approximation $\mathbf{c}$ plus error information $\mathbf{d}$ in only the $\#f$ locations that would be needed to store $\mathbf{g}$ itself. If the connectivity of $\mathbf{c}$ is also consistent with the subdivision, then the decomposition can be repeated on $\mathbf{c}$, potentially yielding a nesting of ever coarser meshes plus associated details. For some applications, the coarse-plus-detail representation has advantages; among them, applications such as compression, multi-level editing and edge detection.

The matrices $\mathbf{P}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{Q}$ by themselves define nothing more than linear transformations (in fact, $\mathbf{P}$ and $\mathbf{A}$ are affine transformations); as such, they take no account of the connectivity of the mesh points. However, the constructions in [2, 3, 20, 21] make use of the connectivity graph of the fine points to locate the geometric neighborhoods that have been mentioned above. The connectivity graphs in those references visually distinguish the *displaced points*, shown as circle nodes, from the *introduced points.* shown as square nodes. The diagrams we will use in this work, for example Figure 2, also follow this practice.

The construction involves columns of $\mathbf{P}$ and $\mathbf{Q}$ and involves rows of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{T}$. Correspondingly, *column diagrams* for $\mathbf{P}$ and $\mathbf{Q}$, and *row diagrams* for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{T}$ are employed. Each diagram is formed for the chosen row or column of a chosen matrix by writing the nonzero elements of that row or column onto the corresponding nodes of the fine mesh's connectivity graph. The *interaction* of any two diagrams is taken as consistent with the interactions between rows and columns in matrix-matrix multiplication.

The column diagram for column κ of $\mathbf{P}$ will, as said above, be referred to by the index k, where $c_\kappa \rightarrow f_k$, and the node (circle) associated with the point f_k will serve as the *home node* of that column diagram; e.g., as in Figure 2. The row diagram for row κ of $\mathbf{A}$, which serves to locate c_κ from the fine points in a chosen geometric neighborhood of f_k, will also be referred to by the index k and the node associated with f_k will serve as the row diagram's home node; e.g., Figure 3. The column diagram of column i of $\mathbf{Q}$ will have introduced-point node i as its home node; e.g., Figure 6. Similarly, the column diagram of column i of $\mathbf{B}$ will have home node i; e.g., Figure 16.

4 $\sqrt{3}$ Subdivision: P

$\sqrt{3}$ subdivision, as defined by Kobbelt, is broken down into the computations needed to produce the introduced points followed by the computations needed to displace the coarse points. These operations define the rows of **P**. For our development we reformulate the subdivision according to the columns of **P**, which leads to the diagram shown in Figure 2. The key to understanding this diagram is that each n represents the valence of a node, and

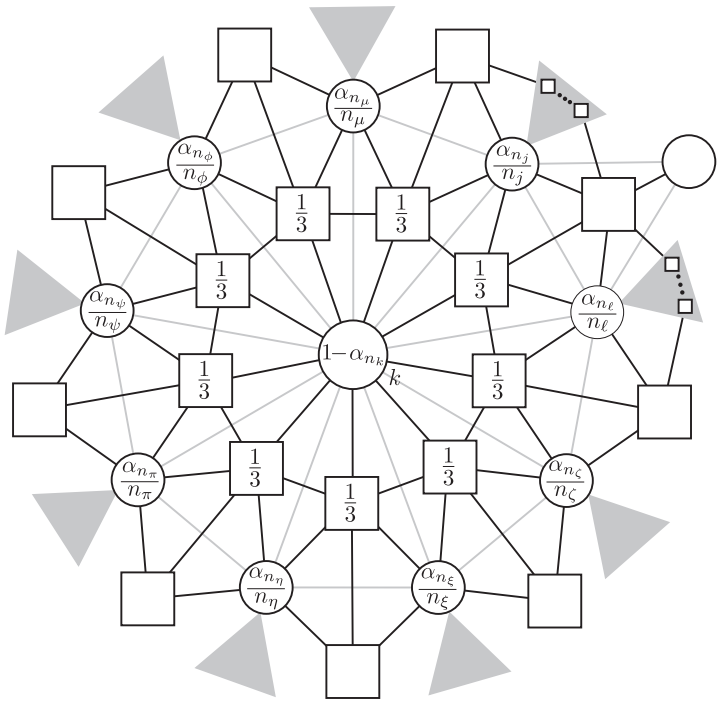

Figure 2: **P** Column Diagram $\mathbf{p}_k$ for $c_\kappa \to f_k$

the α quantities are as Kobbelt has defined them, for example as in (4.1), where λ can be any of the circle-node indices.

$$\alpha_{n_\lambda} = \frac{4 - 2\cos\left(\frac{2\pi}{n_\lambda}\right)}{9} \tag{4.1}$$

The home node of the column diagram in Figure 2 is 9-connected only as an example. Any node representing a displaced point (circle) may have any number of connections. It will become clear that, whatever we achieve with this connectivity, will be true in general. The circle nodes surrounding the home node may each have different connectivities; and so, the grey triangles hide one or more edges. The fine mesh's connectivity graph consists of the black edges, the grey triangles, and the circle and square nodes. Grey edges are also

shown. They are not part of the fine mesh's connectivity graph but are included to reveal the connectivities between the original coarse nodes. Every other circle node in the fine mesh serves as the home node for a similar **P** column diagram, and what we establish with respect to Figure 2 will hold throughout.

The $\sqrt{3}$ subdivision generates introduced points with a regular connectivity. Each introduced node (square) is 6-connected, and its adjacent neighbors alternate between introduced-point nodes (squares) and displaced-point nodes (circles). Something like this property is reflected in every other triangular subdivision that we know of, because practically every paper on such a subdivision includes an argument that, as subdivision is repeated, points with special connectivity become isolated in regions of points with a regular connectivity. As a consequence of this, for $\sqrt{3}$ subdivision, the three square nodes at the upper right edge of Figure 2, in the 1 o'clock, 2 o'clock and 4 o'clock positions will be joined through a chain of square nodes, and no circle nodes, as indicated by the two mini-chains of square nodes within the two upper-right grey triangles. This subdivision-induced regularity, even after just one step of subdivision, it turns out, will permit us to develop a multiresolution for generally-connected coarse meshes.

5 $\sqrt{3}$ Reversal: A

To reverse the $\sqrt{3}$ subdivision we find an **A** filter for each coarse node. In particular, **A** will be found a row at a time, where the κ^{th} row, $\mathbf{a}_\kappa$, will be the filter that determines the pre-subdivision position c_κ of the coarse node that is displaced into its final position associated with circle node f_k. Figure 3 shows how such a filter would appear, shown as a *row diagram*. This filter has been chosen to be of *width 1*; that is, all of the nodes it occupies lie one edge-step or less from the home node. This width has proven to be the most satisfactory of several that were tried.

The nodes in Figure 3 having an a value define a subvector $\mathbf{f}_S$ of the vector of fine points **f**, and they define a corresponding submatrix $\mathbf{P}_S$ of **P**, which can be determined from all the **P** column diagrams (e.g. Figure 2) associated with the circle nodes in this neighborhood whose interaction with the **A** row diagram is nonzero. Figure 4 shows one such example. The subproblem to be solved is

$$\mathbf{P}_S\mathbf{c}_S \approx \mathbf{f}_S \tag{5.1}$$

This will yield estimates of coarse-point positions $\mathbf{c}_S$, including the coarse point c_κ associated with the home node k of the **A** row diagram. Equivalently to forming $(\mathbf{P_S}^T\mathbf{P_S})^{-1}\mathbf{P_S}^T$ to solve (5.1), we may solve (5.2)

$$\mathbf{a}_{\kappa S}\mathbf{P}_S = \mathbf{u} \tag{5.2}$$

for the row subvector $\mathbf{a}_{\kappa S}$ of minimum Euclidian norm, where **u** is the vector of zeros except for a 1 in the position corresponding to the home node of the **A** row diagram. This

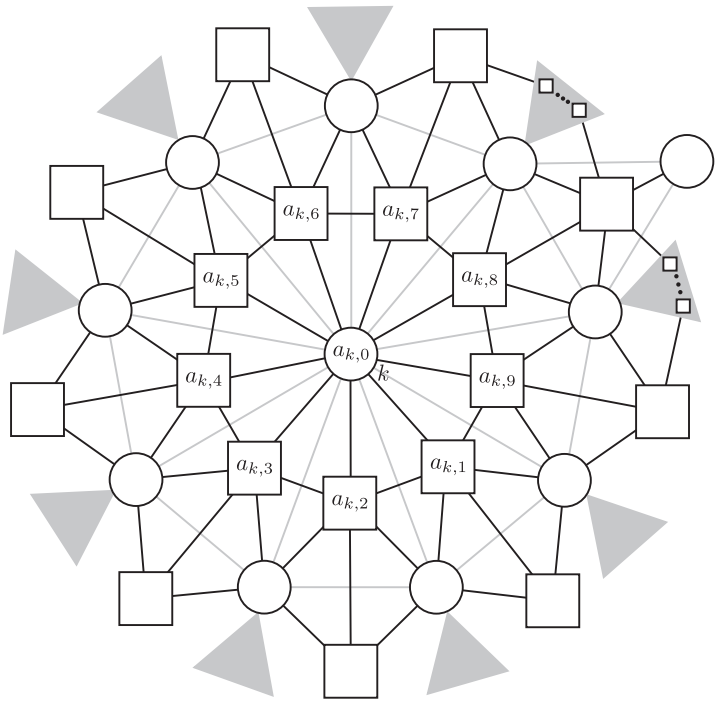

Figure 3: **A** Row Diagram $\mathbf{a}_k$ to Obtain c_κ

leads more directly to the composition of **A** as a left inverse of **P**, and that is the way we proceed here.

The row-column interaction between the a values of Figure 3 and the column values p values represented by Figure 2 must result in 1

$$(1 - \alpha_{n_k})a_{k,0} + \frac{1}{3}a_{k,1} + \cdots + \frac{1}{3}a_{k,9} = 1$$

The row-column interaction between the a values of Figure 3 and the p values of any of the peripheral coarse nodes; e.g., those in Figure 4 must result in zero

$$\frac{\alpha_{n_k}}{n_k}a_{k,0} + \cdots + 0a_{k,\lambda-1} + \frac{1}{3}a_{k,\lambda} + \frac{1}{3}a_{k,\lambda+1} + 0a_{k,\lambda+2} + \cdots + 0a_{k,9} = 0$$

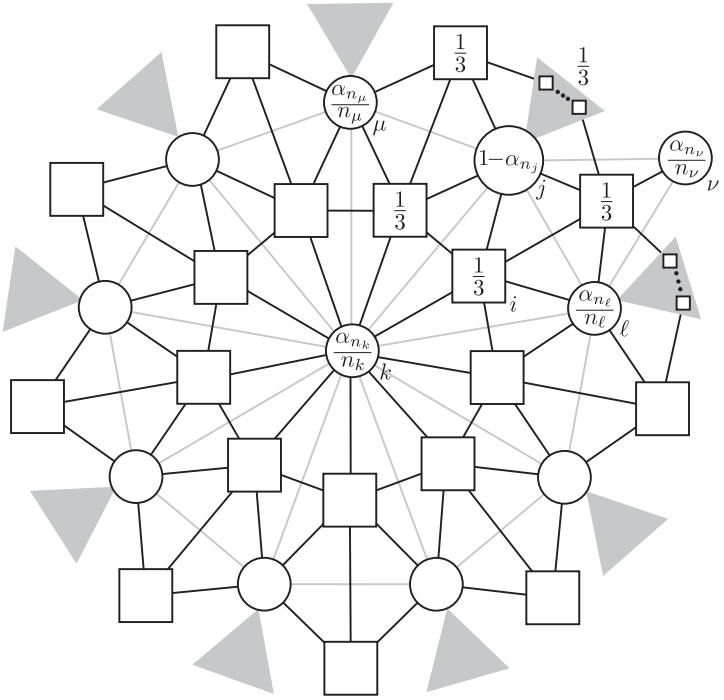

Figure 4: **P** Column on the Periphery

It is more informative to put this in terms of linear equations

$$[a_{k,0}, a_{k,1}, \ldots, a_{k,9}] \begin{bmatrix} 1-\alpha_{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} & \frac{\alpha_{n_k}}{n_k} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = [1, 0, \ldots, 0] \tag{5.3}$$

The matrix in equation (5.3) is the version for $n_k = 7$, which has 8 columns. Generally, for a node with n_k-connectivity, the matrix will have $n_k + 1$ columns, and the pattern of the contents in each column follows what is shown in (5.3).

The matrix corresponding to (5.3) has rank n_k for even n_k; that is, the matrix is deficient in rank by 1, and the matrix has rank $n_k + 1$ for odd n_k; that is, it has full rank.

The solutions in all cases follow a simple pattern. If the central node has n_k-connectivity as in Figure 2, then

$$a_{k,0} = -\frac{2}{3\alpha_{n_k} - 2} \tag{5.4}$$

and

$$a_{k,\lambda} = \frac{3}{n_k}\,\frac{\alpha_{n_k}}{3\alpha_{n_k} - 2} \tag{5.5}$$

for $i > 2$. ($i \leq 2$ is possible only for boundary nodes, and they are discussed in Section 8.)

We let a_k stand for the value of $a_{k,0}$ of (5.4) and h_k stand for the common value of all other $a_{k,\lambda}$ of (5.5), and we arrive at the row diagram for the **A** filter associated with the home node (circle node k) as shown in Figure 5. Every other circle node in the fine mesh

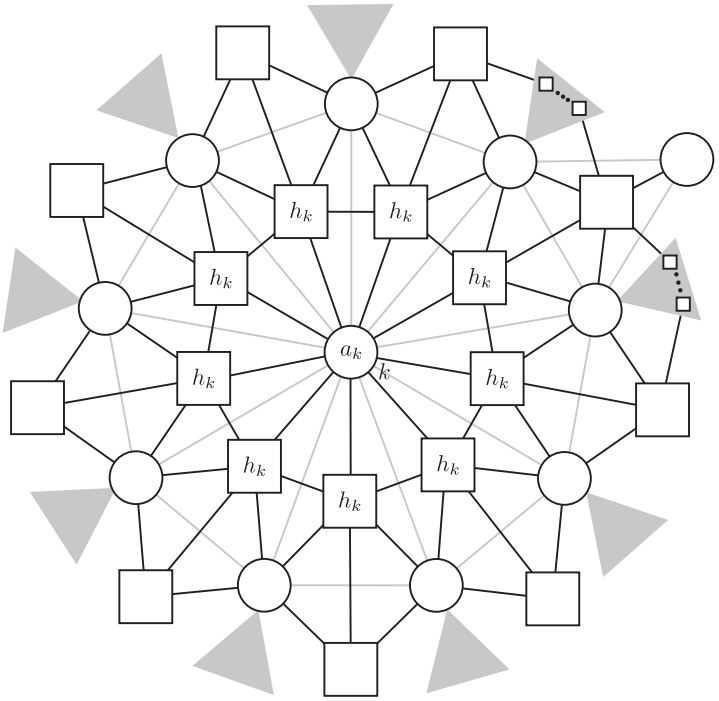

Figure 5: **A** Row Diagram

is the *home node* of an **A** row diagram formed in a corresponding way.

In summary, if f_k is a fine point that is derived from a coarse point c_κ by subdivision displacement, and if $f_{k+1}, \ldots, f_{k+m}$ are the immediately surrounding introduced fine points, and if a_k and h_k are the **A** filter values associated with the f_k node, then the approximated position of c_κ is

$$c_\kappa = a_k f_k + h_k \sum_{\lambda=1}^{m} f_{k+\lambda} \tag{5.6}$$

where a_k and h_k are as in (5.4) and (5.5). If the fine points **f** are derived by subdivision from coarse points **c**, then c_κ will be in exactly the correct position. That is, subdividing the **c** that are obtained in this way will exactly yield the **f** again. If the fine points are, instead, points **g** obtained in another way, then subdividing the **c** will yield **g** only within an error **e**.

6 $\sqrt{3}$ Correction: Q

Finding corrections **e** (3.5) from detail information **d** (3.4) is done using the matrix **Q**, which is found one column at a time. The i^{th} column of **Q** governs the contribution to **e** made by the detail d_i on the introduced (square) node i, and there is one column of **Q** uniquely associated with each introduced node; e.g., see Figure 6. That node, and the figure's surrounding nodes, are labeled with symbols for a **Q** column diagram of width 1. The construction that has been discussed in [3] determines a **Q** column such that all of its

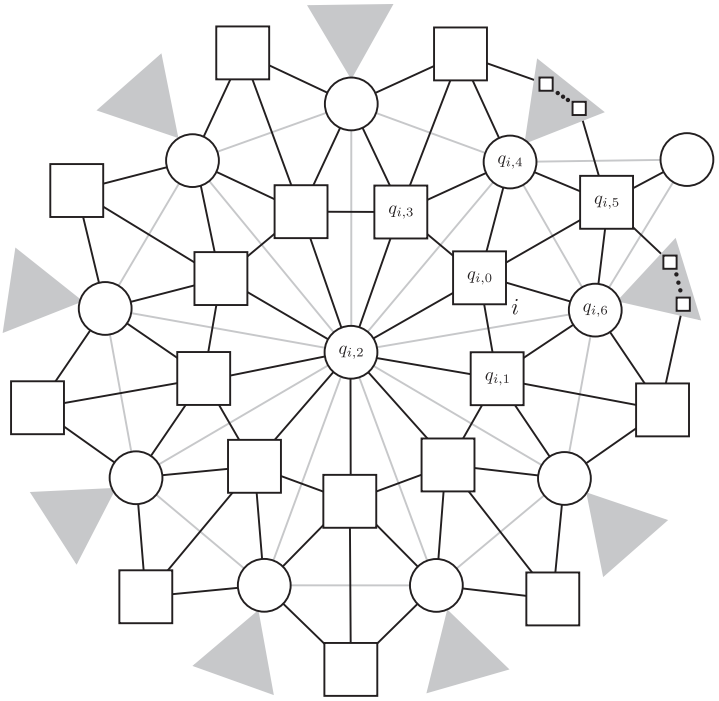

Figure 6: The **Q** Column Diagram for Node i

possible interactions with the rows of **A** will be zero. Recall that we have found **A** row diagrams to consist of a displaced coarse (circular) node as a home node together with all its immediately surrounding introduced (square) nodes. Remembering this, we can easily

see that Figure 7 shows the home nodes of all the **A** row diagrams that will interact with the **Q** column diagram proposed in Figure 6.

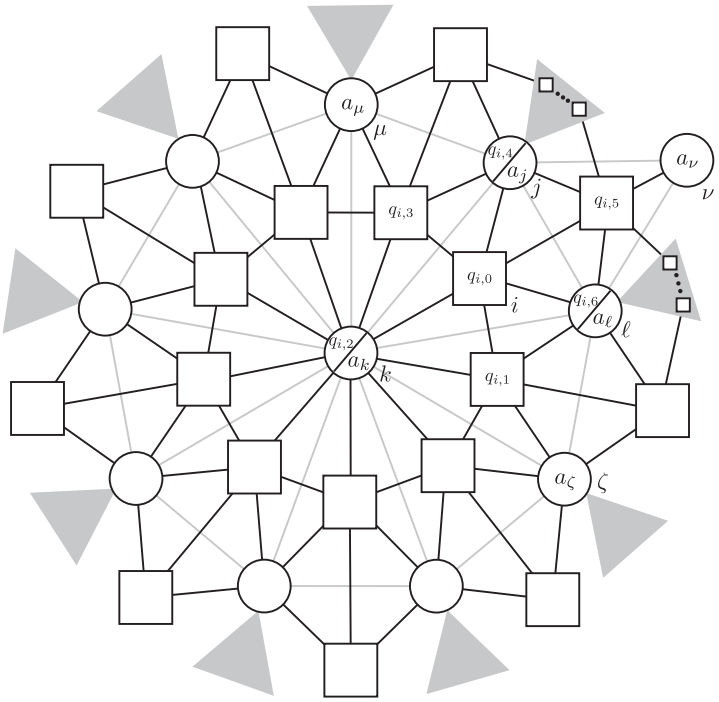

Figure 7: Centers for Interacting **A** Row Diagrams

Figure 8 shows the interaction of the **A** row diagram having home node k with the proposed **Q** column diagram having home node i. The interaction of Figure 8, which we wish to equal zero, can be set down as in (6.1).

$$\begin{bmatrix} h_k & h_k & a_k & h_k & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_{i,0} \\ q_{i,1} \\ q_{i,2} \\ q_{i,3} \\ q_{i,4} \\ q_{i,5} \\ q_{i,6} \end{bmatrix} = 0 \tag{6.1}$$

If we continue in this manner with the other **A** row diagrams indicated in Figure 7, and

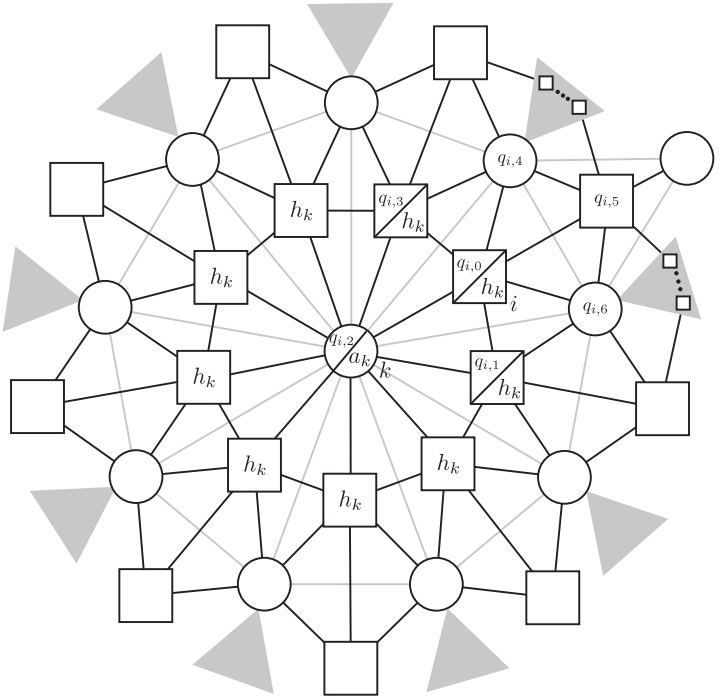

Figure 8: Example **A**-**Q** Interaction

discard any duplicate columns in the matrix, the result is (6.2).

$$\begin{bmatrix} h_k & h_k & a_k & h_k & 0 & 0 & 0 \\ 0 & h_\zeta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_\mu & 0 & 0 & 0 \\ h_j & 0 & 0 & h_j & a_j & h_j & 0 \\ 0 & 0 & 0 & 0 & 0 & h_\nu & 0 \\ h_\ell & h_\ell & 0 & 0 & 0 & h_\ell & a_\ell \end{bmatrix} \begin{bmatrix} q_{i,0} \\ q_{i,1} \\ q_{i,2} \\ q_{i,3} \\ q_{i,4} \\ q_{i,5} \\ q_{i,6} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.2}$$

The requirement that the **Q** column diagram produce a zero in every interaction with an **A** row diagram is equivalent to demanding that the $\mathbf{q}_i$ vector of (6.2) lie in the nullspace of the matrix in (6.2). The nullspace of this matrix is one-dimensional, and a basis vector

is shown in (6.3).

$$\begin{bmatrix} -\frac{a_\ell}{h_\ell} \\ 0 \\ \frac{a_\ell}{h_\ell}\frac{h_k}{a_k} \\ 0 \\ \frac{a_\ell}{h_\ell}\frac{h_j}{a_j} \\ 0 \\ 1 \end{bmatrix} \tag{6.3}$$

The 1 value can be chosen to be on any of the four nontrivial nodes. A change of position is achieved by a re-scaling of the **Q** vector, and any rescaling of a nullspace vector does not move the vector from the nullspace. For example, if every element of (6.3) is multiplied by

$$-\frac{h_\ell}{a_\ell}$$

then the resulting **Q** column diagram is shown in Figure 9, and it is this version of **Q** that we shall use.

Common practice uses rows of **Q** as *correction filters*. To apply **Q**, we must have detail vectors d assigned to each introduced (square) node, as is suggested in Figure 10. The elements of **Q** in that figure will be used to compute the errors **e** according to the information in Figure 9 as follows: each of the diagram's node values in Figure 9 specifies what multiple of the d_i associated with the home node i is added onto the diagram node's accumulating e value. For instance, $-\frac{h_k}{a_k} \times d_i$ is added to node k as node i's contribution to e_k. When every square node has been visited, and all multiples have been accumulated, we have that the error on the i^{th} square node is

$$e_i = d_i \tag{6.4}$$

and the error on the k^{th} circle node is

$$e_k = -\frac{h_k}{a_k} \sum_{\lambda=1}^{m} d_{k+\lambda} \tag{6.5}$$

where a_k and h_k are the **A** values associated with the k^{th} circle node, and the $d_{k+\lambda}$ are the details associated with all the m square nodes immediately surrounding that circle node.

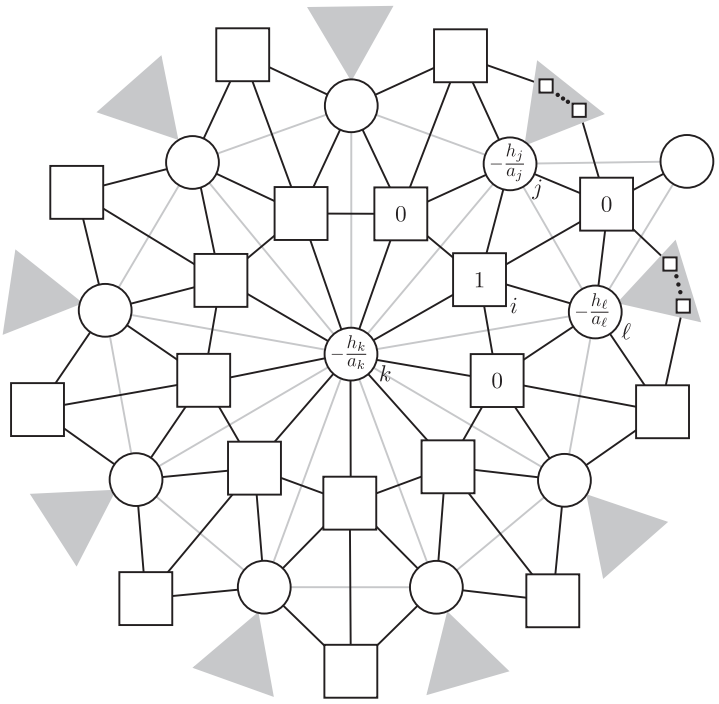

Figure 9: **Q** Diagram Re-Scaled

7 $\sqrt{3}$ Detail: B

In [3] it was established that **B** can be found in two steps. First a left inverse **T** is found for **Q**:

$$\mathbf{TQ} = \mathbf{I}$$

This is done via *local least squares* just as **A** was found from **P**. Then **B** is obtained as:

$$\mathbf{B} = \mathbf{T}\left(\mathbf{I} - \mathbf{PA}\right) \tag{7.1}$$

In particular, **B** will be found a row at a time, where the i^{th} row, $\mathbf{b}_i$, will be the filter that determines the detail, d_i, on square node i. From (7.1) we see that the row, $\mathbf{b}_i$, will be obtained by subtracting the corresponding row, $\mathbf{t}_i$ of **T**, from the result of applying row $\mathbf{t}_i$ to the matrix **P**, to obtain a row, $\mathbf{r}_i$, and applying $\mathbf{r}_i$ to the matrix **A** to obtain a row, $\mathbf{s}_i$.

$$\begin{array}{rcl} \mathbf{r}_i & = & \mathbf{t}_i\mathbf{P} \\ \mathbf{s}_i & = & \mathbf{r}_i\mathbf{A} \\ \mathbf{b}_i & = & \mathbf{t}_i - \mathbf{s}_i \end{array} \tag{7.2}$$

The construction of **T** turns out to be trivial for the **Q** we have determined in Section 6. Figure 11 shows the interaction of the **T** row diagram of width 1 for node i and the **Q**

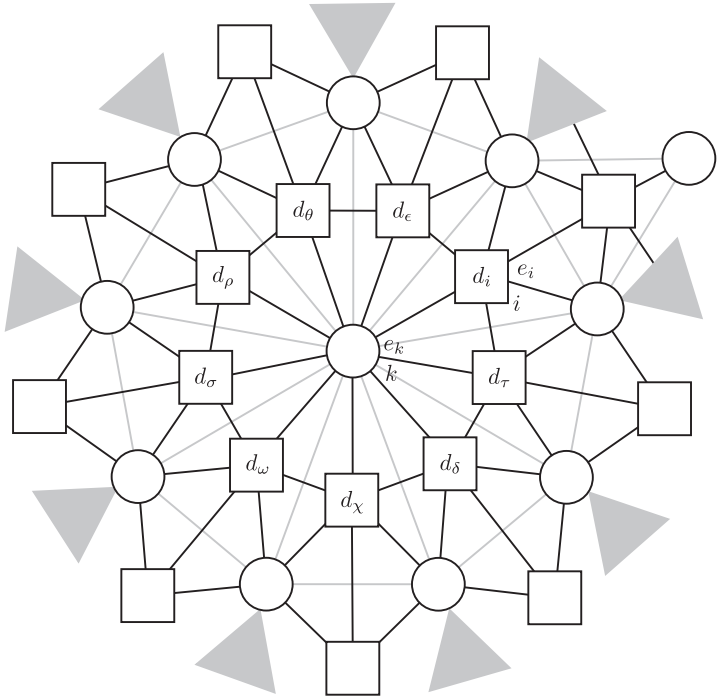

Figure 10: Computing the Errors

column diagram for the same node. The sum of the products of the node entries in Figure 11 must add up to 1, and the sum of the products of the quantities $t_{i,0}, \ldots, t_{i,6}$ with any other **Q** column diagram must be 0. This leads to the (7.3) in which duplicate columns have been removed.

$$[\ t_{i,0}\ t_{i,1}\ t_{i,2}\ t_{i,3}\ t_{i,4}\ t_{l,5}\ t_{i,6}\ ] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{h_k}{a_k} & -\frac{h_k}{a_k} & -\frac{h_k}{a_k} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{h_j}{a_j} & 0 & 0 & -\frac{h_j}{a_j} & -\frac{h_j}{a_j} & -\frac{h_j}{a_j} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{h_\ell}{a_\ell} & -\frac{h_\ell}{a_\ell} & 0 & 0 & 0 & -\frac{h_\ell}{a_\ell} & -\frac{h_\ell}{a_\ell} \end{bmatrix} = [\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ ] \tag{7.3}$$

If there is no solution, we proceed to a wider **T** diagram. If there is a unique solution, we use it. If there are many solutions, we use the one of least Euclidian norm. There is, in fact, one unique solution given by $t_{i,0} = 1$ and $t_{i,1} = \cdots = t_{i,6} = 0$. The diagram for this solution is shown in Figure 12. As we see, the **T** row diagram actually has width 0, and we may take the $\mathbf{t}_i$ vector to be $\mathbf{t}_i = [1]$.

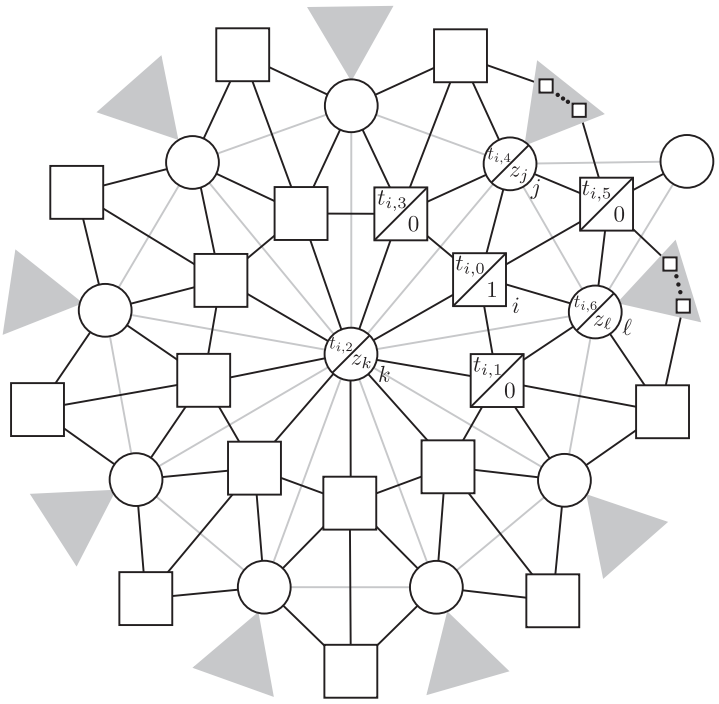

Figure 11: **TQ** Interaction

Referring to (7.2), our next task is to find $\mathbf{r}_i$. Expressed as diagrams, we are interested in all the **P** column diagrams that have a nontrivial overlap with the **T** row diagram of Figure 12. Figure 13 shows one such interaction. The result of the row-column multiplication depicted in Figure 13 will be the value $\frac{1}{3}$, and this will appear as one element of the $\mathbf{r}_i$ row diagram, referring to the equations (7.2). The proper node on which to write this value is the home circle node associated with the column of **P** that participated in this interaction. Continuing, we arrive at the row diagram for $\mathbf{r}_i$ shown in Figure 14.

The final step is to form the row diagram for $\mathbf{s}_i$ given in (7.2). To see how this diagram should be formed, we note that it represents the linear combination of rows,

$$r_{i,1}\mathbf{a}_1 + \cdots + r_{i,\#c}\mathbf{a}_{\#c}$$

or, for our purposes, the corresponding linear combination of the row diagrams for **A**. There will be one term in the sum for each circle node occupied by one of the nonzero **r** values, and the term itself will consist of the row diagram of **A** corresponding to that circle node, multiplied by the **r** value. Figure 15 shows the result of $\mathbf{r}_i$ times $\mathbf{a}_k$ to produce the $\mathbf{s}_k$ diagram. There will be three such diagrams: $\mathbf{s}_k$, $\mathbf{s}_j$ and $\mathbf{s}_\ell$.

The final step in (7.2) consists merely of subtracting the row diagram of Figure 15 from

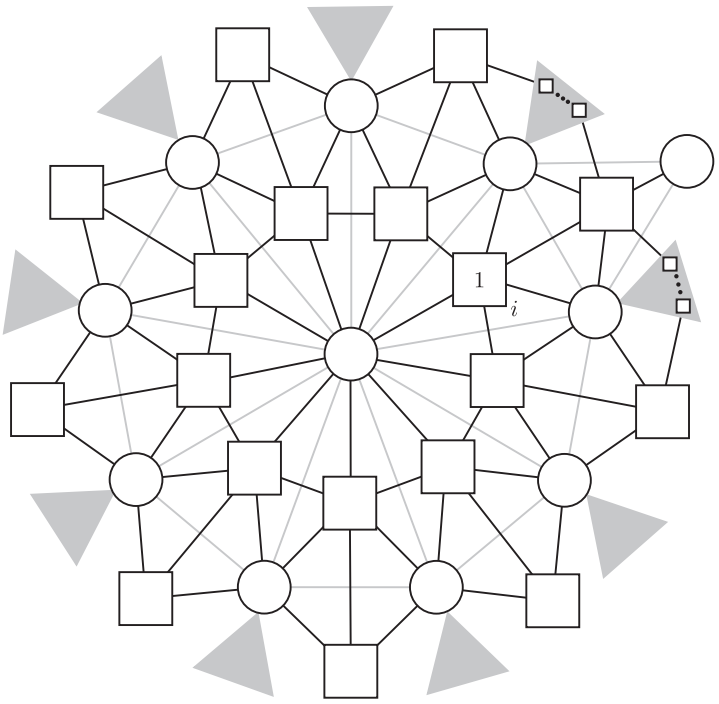

Figure 12: The Row Diagram for $\mathbf{t}_i$

the row diagram of Figure 12. The result is shown in Figure 16, where

$$\begin{aligned} w_{kj} &= -\tfrac{h_k+h_j}{3} \\ w_{k\ell} &= -\tfrac{h_k+h_\ell}{3} \\ w_{j\ell} &= -\tfrac{h_j+h_\ell}{3} \\ w_{kj\ell} &= 1-\tfrac{h_k+h_j+h_\ell}{3} \end{aligned} \tag{7.4}$$

The diagram of Figure 16 looks more complicated than it really is. Node i will be at the centroid of a triangle joining 3 circle nodes. Each of those circle nodes will be the home node for an **A** diagram. To obtain the **B** diagram, simply put a 1 on the i square node and then subtract $\frac{1}{3}$ times each of the 3 circle-node **A** diagrams. The **B** diagram is the resulting row diagram. To use the diagram of Figure 16 to compute the detail coordinates on the i square node, take each value written on a node in the figure, multiply that value times the fine point coordinates associated with that node, and sum up the results.

It is an interesting special property of $\sqrt{3}$ subdivision that the detail coordinates d for any square node are actually the error coordinates e for that node. This can be inferred from the fact that the form of the **Q** diagram for any square node has a 1 for that node

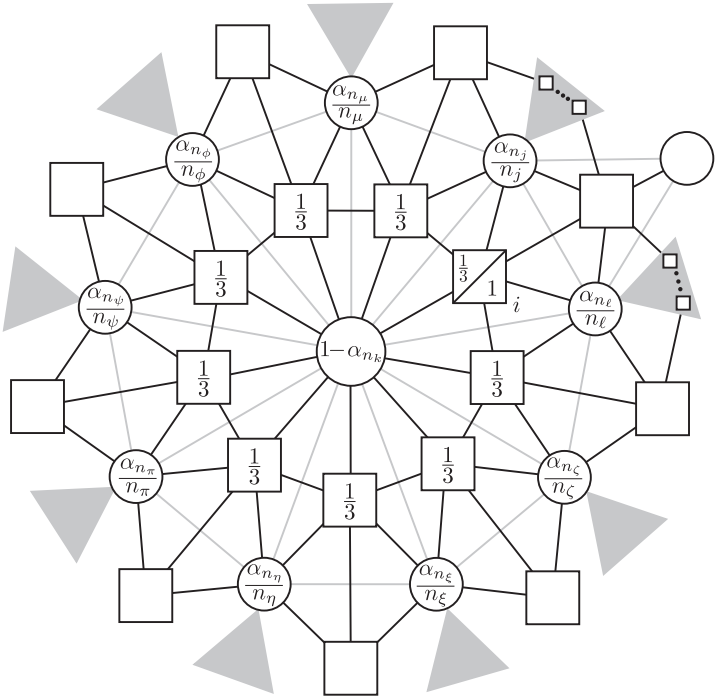

Figure 13: The $\mathbf{t}_i$ Row Diagram Interacting with the $\mathbf{p}_k$ Column Diagram

and a zero for every other square node. So applying the **Q** filter to convert the detail for a square node into the error for that square node trivializes into simply taking the detail as the error. Also, the action of the $\mathbf{b}_i$ row diagram of **B** can be read as the process of reversing the subdivision to estimate the coarse position of each of the three circle nodes adjoining a square node, followed by applying the subdivision to these estimated coarse circle positions ($\frac{1}{3}$ and $\frac{1}{3}$ and $\frac{1}{3}$) to create the "subdivision version" of the introduced square node, and finally subtracting that from the actual position of the square node. And that is, in words, the description of how the error associated with the square node would be produced. When implementing the $\sqrt{3}$ multiresolution, if one has a more convenient way of determining the errors e on the square nodes, then one is free to use these errors as the d values and to ignore **B**.

8 Boundary Curves

Boundary points are handled separately by Kobbelt [11]. He treats coarse points on the boundary as if they were the coefficients of a cubic spline with uniform knots, let's say

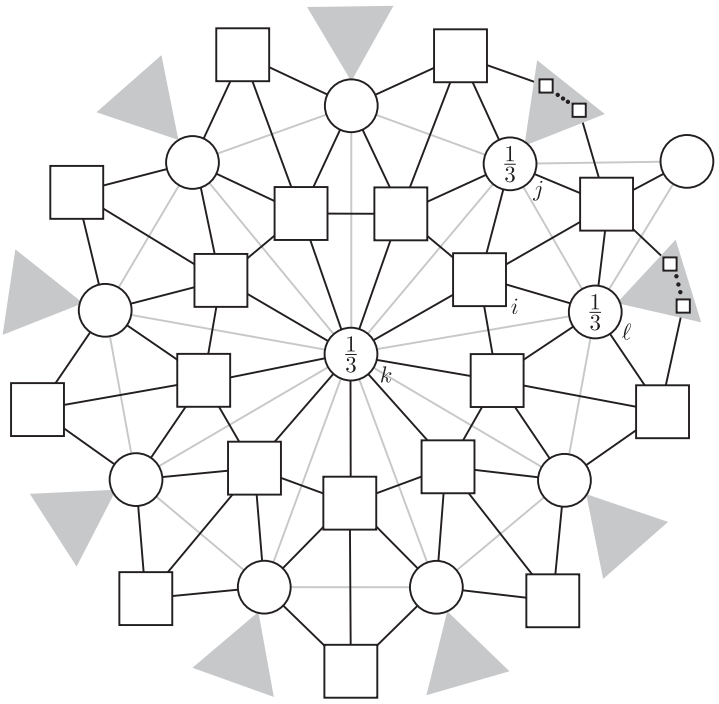

Figure 14: The Row Diagram for $\mathbf{r}_i$

with 3 units of spacing per knot:

$$\ldots \; 10 \quad 13 \quad 16 \quad 19 \; \ldots$$

and he employs the uniform insertion of two knots per existing knot interval; that is,

$$\ldots \; 10 \; 11 \; 12 \; 13 \; 14 \; 15 \; 16 \; 17 \; 18 \; 19 \; \ldots$$

to transform the coarse boundary points into the fine ones. This corresponds, in our terms, to the **P** diagram shown in Figure 17. Diagrams for B-splines have been developed in, for example [20, 2]. It is straightforward to establish corresponding **A**, **B** and **Q** diagrams for the **P** of Figure 17.

The appropriate **A** diagram is shown in Figure 18. This diagram corresponds to approximating the location of the coarse point c_κ from its displaced image $f_k = f_{3\kappa}$ as

$$c_\kappa = \sum_{\lambda=k-4}^{k+4} a_\lambda f_\lambda$$

with the coefficient a_λ standing for the fraction shown on node f_λ.

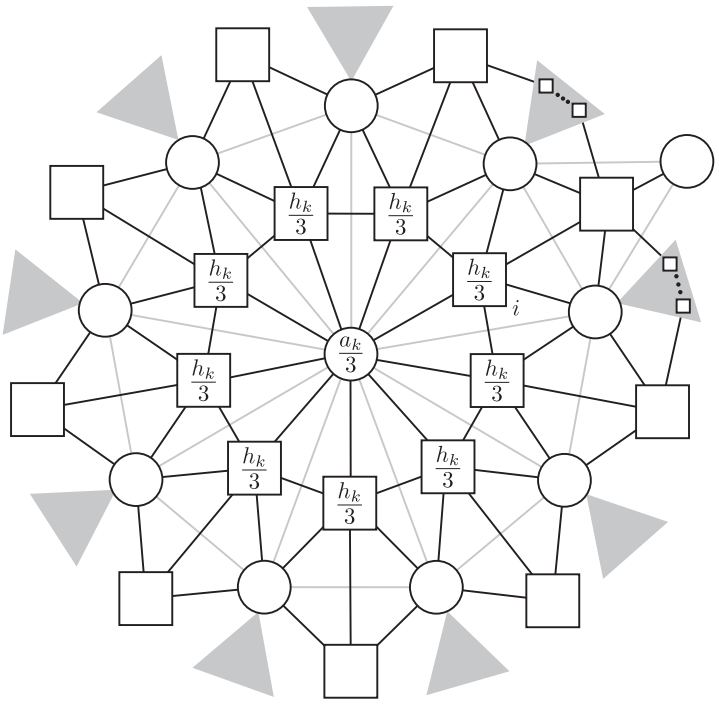

Figure 15: Interaction of $\mathbf{r}_{i,k}$ and $\mathbf{a}_k$

The appropriate **B** diagrams (there are two) are shown in Figure 19. The b values are as in (8.1)

$$\begin{aligned}
b_1 &= -\frac{360908824401952}{6124761750733245} \\
b_2 &= \frac{821150352767744}{6124761750733245} \\
b_3 &= \frac{324486832948544}{6124761750733245} \\
b_4 &= -\frac{227763072235964}{680529083414805} \\
b_5 &= \frac{276752921024}{1311231374595} \\
b_6 &= \frac{34302231721304}{680529083414805} \\
b_7 &= -\frac{149920452798584}{6124761750733245} \\
b_8 &= -\frac{412135978326626}{6124761750733245} \\
b_9 &= \frac{276003571289944}{6124761750733245} \\
b_{10} &= -\frac{1740361028}{513090537885} \\
b_{11} &= -\frac{4404178928}{513090537885} \\
b_{12} &= \frac{1935707674}{513090537885}
\end{aligned} \tag{8.1}$$

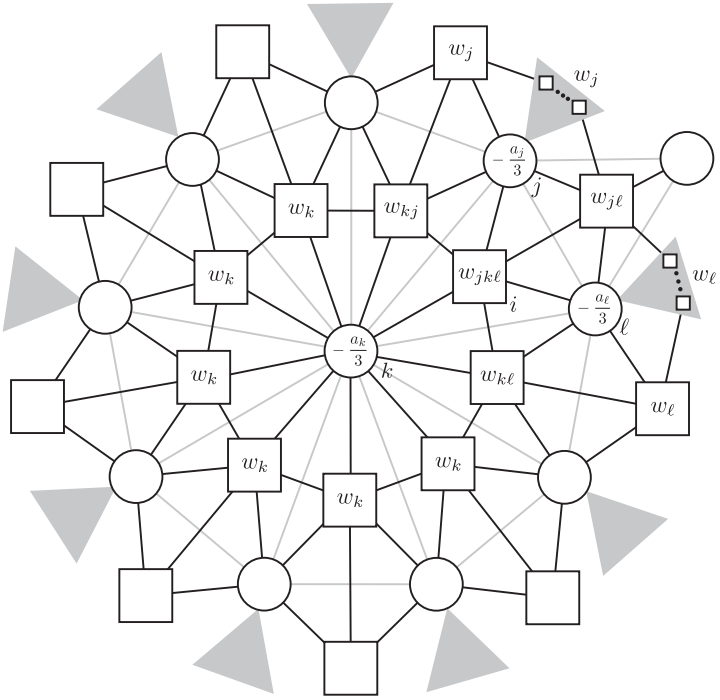

Figure 16: The Row Diagram for **B**

The first diagram corresponds to computing the detail coordinates associated with $f_{3\kappa-1} = f_{k-1}$ as

$$d_{k-1} = \sum_{\lambda=-7}^{+4} b_{5-\lambda} f_{k+\lambda}$$

The second diagram corresponds to computing the detail coordinates associated with $f_{3\kappa+1} = f_{k+1}$ as

$$d_{k+1} = \sum_{\lambda=-4}^{+7} b_{5+\lambda} f_{k+\lambda}$$

Finally, the appropriate **Q** diagrams (there are two) are shown in Figure 20. The q values are as in (8.2)

$$\begin{aligned} q_1 &= \frac{11710197}{17758786} \\ q_2 &= \frac{13321692}{8879393} \\ q_3 &= 1 \\ q_4 &= \frac{11774114}{8879393} \\ q_5 &= -\frac{39839315}{17758786} \end{aligned} \tag{8.2}$$

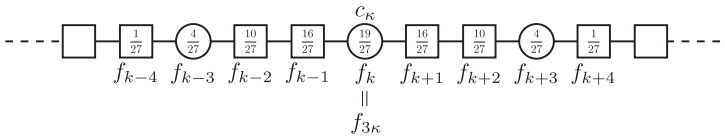

Figure 17: The Boundary **P** Diagram

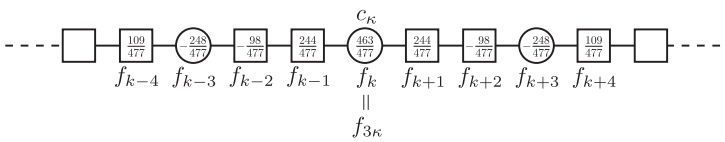

Figure 18: Boundary **A** Diagram

These **Q** diagrams are column diagrams, and they translate into three distinct ways of computing the error of a fine point as follows:

1. For a *displaced point*, which corresponds to a circle node in Figure 20 we have, for example for $f_{3\kappa} = f_k$,

$$e_k = q_1 d_{k-2} + q_4 d_{k-1} + q_4 d_{k+1} + q_1 d_{k+2}$$

   where the d are the coordinates of the details on the nodes indicated by the subscripts;

2. For an *introduced point* preceding a displaced point, which corresponds to a square node to the left of a circle node in Figure 20 we have, for example for $f_{3\kappa-1} = f_{k-1}$,

$$e_{k-1} = q_2 d_{k-2} + q_3 d_{k-1} + q_5 d_{k+1}$$

3. For an introduced point following a displaced point, which corresponds to a square node the the right of a circle node Figure 20 we have, for example for $f_{3\kappa+1} = f_{k+1}$,

$$e_{k+1} = q_5 d_{k-1} + q_3 d_{k+1} + q_2 d_{k+2}$$

If a boundary curve is not closed and cyclic (or doubly infinite), we must be prepared to deal with end points. Kobbelt ignores this possibility, but we have taken the trouble to provide **P**, **A**, **B** and **Q** for all cases from 4 up to 18 coarse points, at which juncture the interior boundary points may be treated as in Figures 17 though 20. A cubic boundary curve, of course, must have at least 4 control vertices, hence the starting point of our work.

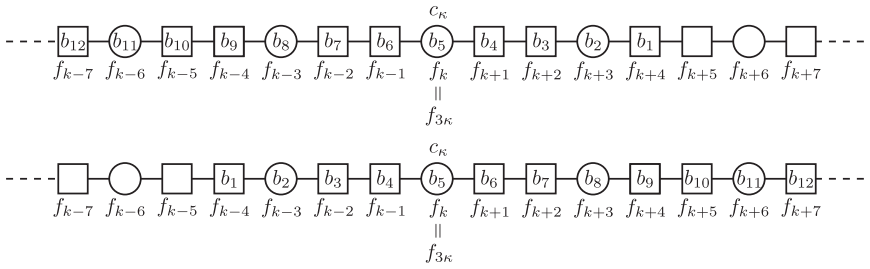

Figure 19: Boundary **B** Diagrams

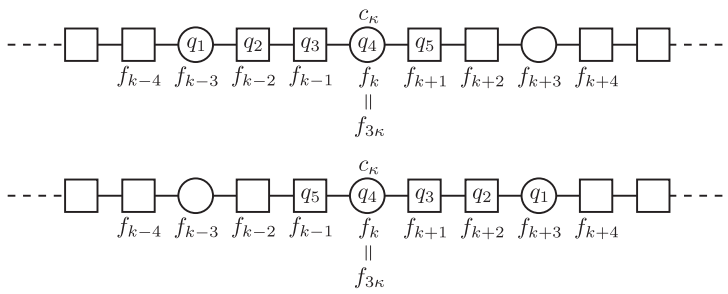

Figure 20: Boundary **Q** Diagrams

We use, as does Kobbelt, cubic, B-spline knot structure, and take

$$0, 0, 0, 0, 5, 8, 11, 14, \ldots \quad \text{knots every 3 integer positions onward}$$

for the beginning of the coarse boundary curve (and symmetrically at the end). We introduce new knots to obtain the knot structure

$$0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, \ldots \quad \text{knots every integer position onward}$$

for the fine boundary curve. The use of a quadruple knot at the beginning (and end) ensures that the curve's endpoints do not move in their spatial positions under the subdivision. Our choice of 5 as the first coarse knot thereafter, and of $1, 2, 3, 4$ as the knots introduced to transition to the sequence of fine knots, is based on trial an error to settle on what generally gave the most satisfying results.

9 Results

In this Section, we provide some of the results of the $\sqrt{3}$ multiresolution framework provided in this paper. To obtain these results, we used ACM data structure in which details and coarse vertices can be easily distinguished from each other. Figure 21 illustrates the

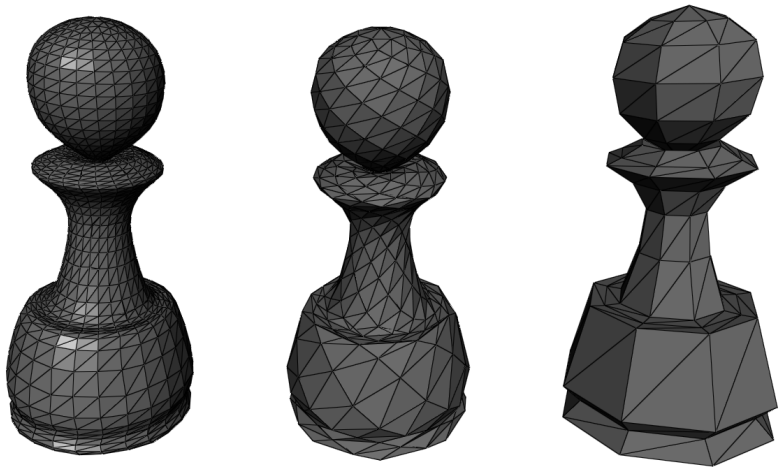

Figure 21: The result of the proposed $\sqrt{3}$ multiresolution framework on a pawn that has been subdivided twice.

result of our multiresolution framework that has been subdivided two times using $\sqrt{3}$ subdivision. It is apparent that the original mesh is obtained. In Figure 22 after two times of $\sqrt{3}$ subdivision, we have randomly displaced some of the vertices to observe how stable out multiresolution performs when the geometry of the mesh is not obtained from the subdivision.

To better observe the performance of our $\sqrt{3}$ multiresolution framework, we have applied its filters on a toroidal polyhedron whose vertices are obtained from sampling the parametric equation of a torus. Figure 23 illustrates the results of four iterations of our multiresolution framework. It is apparant that the proposed $\sqrt{3}$ multiresolution framework is quite stable.

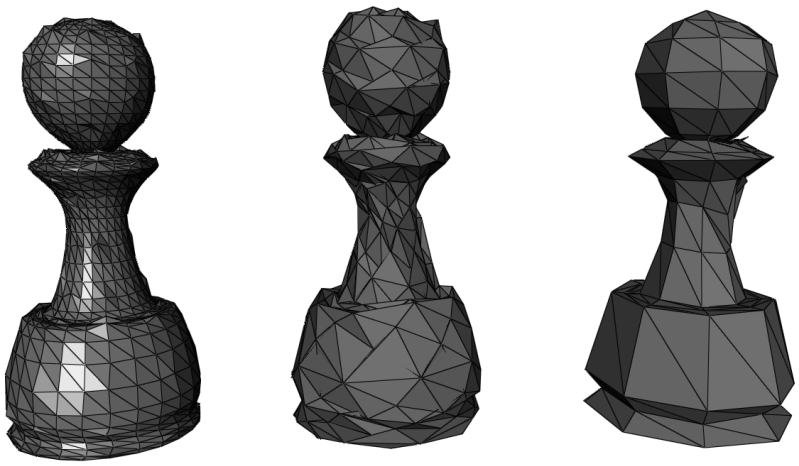

Figure 22: The result of the proposed $\sqrt{3}$ multiresolution framework on a pawn that has been subdivided twice and its vertices are randomly perturbed.

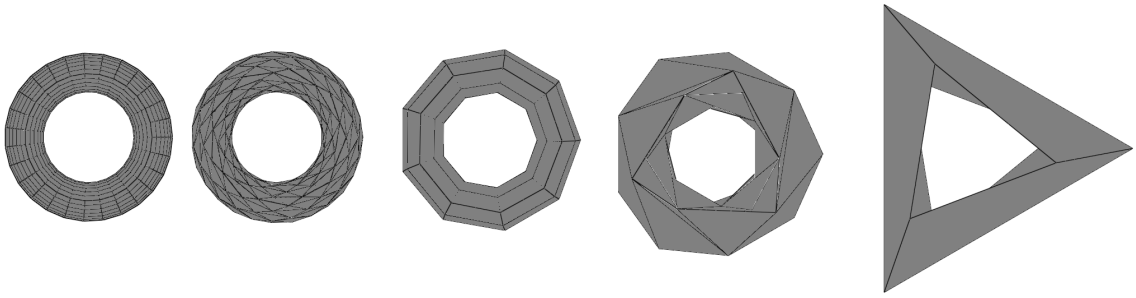

Figure 23: Three iterations of our $\sqrt{3}$ multiresolution framework applied on a toroidal polyhedron obtained from sampling the parametric equation of a torus.

References

[1] Marc Alexa. Refinement operators for triangle meshes. *Comput. Aided Geom. Des.*, 19(3):169–172, 2002.

[2] R. H. Bartels and F. F. Samavati. Reversing subdivision rules: Local linear conditions and observations on inner products. *Journal of Computational and Applied Mathematics*, 119(1–2):29–67, July 2000.

[3] R. H. Bartels and F. F. Samavati. Multiresolutions numerically from subdivisions. *Computers and Graphics*, 35(2):185–197, 2011.

[4] M. Bertram. Biorthogonal loop-subdivision wavelets. *Computing*, 72(1-2):29–39, April 2004.

[5] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Des.*, 10(6):350–355, November 1978.

[6] D. Doo and M. Sabin. Seminal graphics. pages 177–181, 1998.

[7] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 173–182. ACM, 1995.

[8] J.C. Goswami and A.K. Chan. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. Wiley Series in Microwave and Optical Engineering. Wiley, 2011.

[9] Hugues Hoppe. View-dependent refinement of progressive meshes. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 189–198, 1997.

[10] Hugues Hoppe and Emil Praun. Shape compression using spherical geometry images. In NeilA. Dodgson, MichaelS. Floater, and MalcolmA. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 27–46. Springer Berlin Heidelberg, 2005.

[11] L. P. Kobbelt. Root-three subdivision. In *SIGGRAPH'00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000.

[12] Leif Kobbelt. $\sqrt{3}$ subdivision. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH 2000, pages 103–112. ACM, 2000.

[13] Aaron W. F. Lee, David Dobkin, Wim Sweldens, and Peter Schröder. Multiresolution mesh morphing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 343–350, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[14] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah, 1987.

[15] Ali Mahdavi-Amiri and Faramarz Samavati. ACM: Atlas of connectivity maps for semiregular models. In *Proceedings of Graphics Interface 2013*, GI '13, pages 99–107, 2013.

[16] Ali Mahdavi-Amiri and Faramarz Samavati. Atlas of connectivity maps. *Computers & Graphics*, 39(0):1 – 11, 2014.

[17] L. Olsen, F.F. Samavati, and R.H. Bartels. Multiresolution for curves and surfaces based on constraining wavelets. *Computers and Graphics*, 31(3):449 – 462, 2007.

[18] Luke J. Olsen and Faramarz F. Samavati. A discrete approach to multiresolution curves and surfaces. In *Proceedings of the 2008 International Conference on Computational Sciences and Its Applications*, pages 468–477. IEEE Computer Society, 2008.

[19] Javad Sadeghi and Faramarz F. Samavati. Smooth reverse loop and catmull-clark subdivision. *Graphical Models*, 73(5):202–217, Sep 2011.

[20] F. F. Samavati and R. H. Bartels. Multiresolution curve and surface representation by reversing subdivision rules. *Computer Graphics Forum*, 18(2):97–119, June 1999.

[21] F. F. Samavati and R. H. Bartels. Diagrammatic tools for generating biorthogonal multiresolutions. *International Journal of Shape Modeling*, 12(1):47–73, 2006.

[22] Eric J. Stollnitz, Tony D. Derose, and David H. Salesin. *Wavelets for computer graphics: theory and applications*. Morgan Kaufmann Publishers Inc., 1996.

[23] Wim Sweldens. Wavelets and the lifting scheme: A 5 minute tour.

[24] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[25] Huawei Wang, Kaihuai Qin, and Hanqiu Sun. 3-subdivision-based biorthogonal wavelets. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5):914–925, 2007.

[26] Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive multiresolution mesh editing. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 259–268, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.