Highly Symmetric $\sqrt{3}$-refinement Bi-frames for Surface Multiresolution Processing

Qingtang Jiang* and Dale K. Pounds

Abstract

Multiresolution techniques for (mesh-based) surface processing have been developed and successfully used in surface progressive transmission, compression and other applications. A triangular mesh allows $\sqrt{3}$, dyadic and $\sqrt{7}$ refinements. The $\sqrt{3}$-refinement is the most appealing one for multiresolution data processing since it has the slowest progression through scale and provides more resolution levels within a limited capacity. The $\sqrt{3}$ refinement has been used for surface subdivision and for discrete global grid systems

Recently lifting scheme-based biorthogonal bivariate wavelets with high symmetry have been constructed for surface multiresolution processing. If biorthogonal wavelets (with either dyadic or $\sqrt{3}$ refinement) have certain smoothness, they will have big supports. In other words, the corresponding multiscale algorithms have large templates; and this is undesirable for surface processing. On the other hand, frames provide a flexibility for the construction of system generators (called framelets) with high symmetry and smaller supports. In this paper we study highly symmetric $\sqrt{3}$-refinement wavelet bi-frames for surface processing. We design the frame algorithms based on the vanishing moments and smoothness of the framelets. The frame algorithms obtained in this paper are given by templates so that one can easily implement them. We also present interpolatory $\sqrt{3}$ subdivision-based frame algorithms. In addition, we provide frame ternary multiresolution algorithms for boundary vertices on an open surface.

Keywords: Biorthogonal wavelets; wavelet bi-frames; dual wavelet frames; $\sqrt{3}$-refinement; multiresolution algorithm templates; lifting scheme; surface multiresolution processing.

Mathematics Subject Classification (2000): 42C40, 65T60, 68U07, 65D17

1 Introduction

The subject of multiresolution (multiscale) analysis has been a popular area of research for more than two decades. Multiresolution techniques for (mesh-based) surface processing have been developed and successfully used in surface progressive transmission, compression and other applications [32, 44, 27, 45]. A triangular mesh allows $\sqrt{3}$, dyadic and $\sqrt{7}$ refinements [13, 3]. The dyadic refinement is shown in the middle of Fig. 1, where the nodes with circles $\bigcirc$ form the coarse mesh. The right part of Fig. 1 shows the $\sqrt{3}$ refinement with the nodes of circles $\bigcirc$ forming the coarse mesh of $\sqrt{3}$ refinement. The $\sqrt{3}$-refinement is the most appealing refinement for multiresolution

*Corresponding author: Qingtang Jiang, e-mail: jiangq@umsl.edu, phone: 1-314-516-6358, fax: 1-314-516-5400. The authors are with the Department of Mathematics and Computer Science, University of Missouri–St. Louis, St. Louis, MO 63121, USA.

data processing since it has the slowest progression through scale and provides more resolution levels within a limited capacity. In CAGD, the $\sqrt{3}$ subdivision, whose topological rule is the $\sqrt{3}$ refinement, has been studied by researchers, see e.g. [28, 29, 25, 26, 34, 16, 6, 7]. The $\sqrt{3}$ refinement has been used for discrete global grid systems [40] and for The PYXIS Digital Earth Reference Model [36].

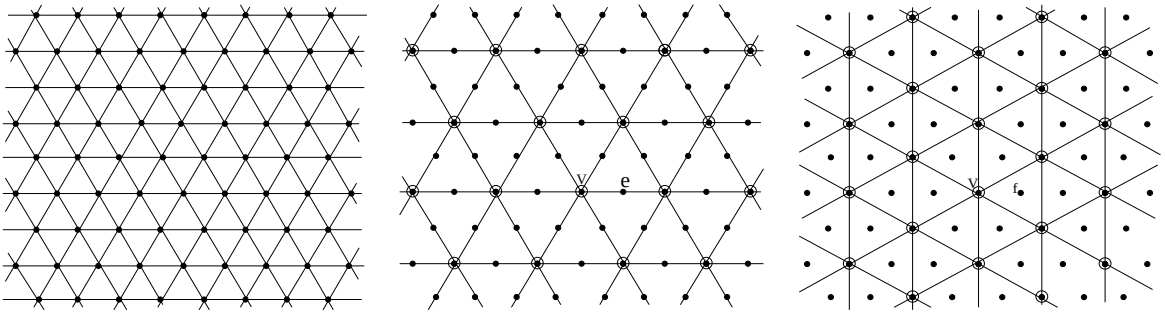

Figure 1: Left: *Triangular mesh;* Middle: *Dyadic refinement coarse mesh with nodes* ○ v; Right: $\sqrt{3}$-*refinement coarse lattice with nodes* ○ v

While some work on dyadic wavelets for surface processing has been carried out, see e.g. [31, 32, 42, 41, 1, 2, 46, 33, 23, 24], there is less work on $\sqrt{3}$-refinement wavelets. The authors of [4] construct $\sqrt{3}$-refinement complex pre-wavelets (semi-orthogonal wavelets) on the hexagonal lattice with the scaling functions being the elementary polyharmonic hexagonal B-splines. Since the wavelets in [4] have infinite support, they are not suitable for surface processing. The authors of [47] construct compactly supported biorthogonal $\sqrt{3}$-refinement wavelets. A "discrete inner product" related to the discrete filters is used in [47], which may result in the basis functions and wavelets that are not in $L^2(\mathbb{R}^2)$. Orthogonal and biorthogonal $\sqrt{3}$-refinement wavelets with the conventional L^2 inner product have been studied in [22]. [24] shows that biorthogonal $\sqrt{3}$-refinement wavelets in [22] have the symmetry required for surface processing and that the corresponding multiresolution algorithms can be represented as templates. If biorthogonal wavelets (with either dyadic or $\sqrt{3}$ refinement) have certain smoothness, they will have big supports. Namely, the multiresolution algorithms have large templates. This is not a desirable property for surface processing. On the other hand, the analysis and construction of wavelet (or affine) frames have been studied, see e.g. [5], [9], [18], [38], [39] and reference therein. A frame system provides a flexibility for the construction of framelets with high symmetry and smaller supports than biorthogonal wavelets. The main goal of this paper is to construct wavelet frames with high symmetry for surface $\sqrt{3}$ multiresolution processing.

Let A be a matrix (called a dilation matrix) that maps the triangular mesh onto its coarse mesh on the right of Fig. 1:

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}. \tag{1}$$

For a function g on $\mathbb{R}^2$, denote $g_{j,\mathbf{k}}(\mathbf{x}) = 3^{j/2} g(A^j \mathbf{x} - \mathbf{k})$. Functions $\psi^{(1)}, \psi^{(2)}, \cdots, \psi^{(L)}$ on $\mathbb{R}^2$, where $L \geq 2$, are called wavelet framelets (or wavelet frame generators), just called framelets for short in this paper, provided that $\{\psi^{(1)}_{j,\mathbf{k}}(\mathbf{x}), \psi^{(2)}_{j,\mathbf{k}}(\mathbf{x}), \cdots, \psi^{(L)}_{j,\mathbf{k}}(\mathbf{x})\}_{j\in\mathbb{Z},\mathbf{k}\in\mathbb{Z}^2}$ is a frame in the sense

that there are two positive constants B and C such that

$$B\|g\|_2^2 \le \sum_{\ell=1}^{L} \sum_{j\in\mathbb{Z},\mathbf{k}\in\mathbb{Z}^2} |\langle g, \psi_{j,\mathbf{k}}^{(\ell)}\rangle|^2 \le C\|g\|_2^2, \quad \forall g \in L^2(\mathbb{R}^2),$$

where $\langle\cdot,\cdot\rangle$ and $\|\cdot\|_2 := \langle\cdot,\cdot\rangle^{\frac{1}{2}}$ denote the inner product and the norm of $L^2(\mathbb{R}^2)$.

The construction of affine frames is related to frame filter banks and scaling functions. More precisely, for a sequence $\{p_\mathbf{k}\}_{\mathbf{k}\in\mathbb{Z}^2}$ of real numbers with finitely many $p_\mathbf{k}$ nonzero, let $p(\boldsymbol{\omega})$ denote the corresponding finite impulse response (FIR) filter (here a factor $1/3$ is multiplied):

$$p(\boldsymbol{\omega}) = \frac{1}{3}\sum_{\mathbf{k}\in\mathbb{Z}^2} p_\mathbf{k} e^{-i\mathbf{k}\omega}.$$

$p(\boldsymbol{\omega})$ is also called the symbol of $\{p_\mathbf{k}\}_{\mathbf{k}\in\mathbb{Z}^2}$.

For a pair of FIR frame filter banks $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$, let ϕ and $\widetilde{\phi}$ be the associated refinable (or scaling) functions (with dilation matrix A) satisfying the refinement equations

$$\phi(\mathbf{x}) = \sum_\mathbf{k} p_\mathbf{k}\phi(A\mathbf{x}-\mathbf{k}),\ \widetilde{\phi}(\mathbf{x}) = \sum_\mathbf{k} \widetilde{p}_\mathbf{k}\widetilde{\phi}(A\mathbf{x}-\mathbf{k}).$$

Let $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, be the functions defined by

$$\psi^{(\ell)}(\mathbf{x}) = \sum_\mathbf{k} q_\mathbf{k}^{(\ell)}\phi(A\mathbf{x}-\mathbf{k}),\ \widetilde{\psi}^{(\ell)}(\mathbf{x}) = \sum_\mathbf{k} \widetilde{q}_\mathbf{k}^{(\ell)}\widetilde{\phi}(A\mathbf{x}-\mathbf{k}).$$

We say that $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, generate **bi-frames** of $L^2(\mathbb{R}^2)$ or **dual wavelet frames** of $L^2(\mathbb{R}^2)$ if $\{\psi_{j,\mathbf{k}}^{(1)}(\mathbf{x}), \cdots, \psi_{j,\mathbf{k}}^{(L)}(\mathbf{x})\}_{j\in\mathbb{Z},\mathbf{k}\in\mathbb{Z}^2}$ and $\{\widetilde{\psi}_{j,\mathbf{k}}^{(1)}(\mathbf{x}), \cdots, \widetilde{\psi}_{j,\mathbf{k}}^{(L)}(\mathbf{x})\}_{j\in\mathbb{Z},\mathbf{k}\in\mathbb{Z}^2}$ are frames of $L^2(\mathbb{R}^2)$ and that for any $f \in L^2(\mathbb{R}^2)$, f can be written as (in L^2-norm)

$$f = \sum_{1\le\ell\le L}\sum_{j\in\mathbb{Z},\mathbf{k}\in\mathbb{Z}^2} \langle f, \widetilde{\psi}_{j,\mathbf{k}}^{(\ell)}\rangle \psi_{j,\mathbf{k}}^{(\ell)}.$$

In this case, $p, \widetilde{p}$ are called lowpass filters, and $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \le \ell \le L$, highpass filters.

Assume a pair of FIR frame filter banks $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ is **biorthogonal** (with dilation matrix A), namely,

$$\overline{p(\boldsymbol{\omega})}\widetilde{p}(\boldsymbol{\omega}+2\pi A^{-T}\boldsymbol{\eta}_j) + \sum_{\ell=1}^{L} \overline{q^{(\ell)}(\boldsymbol{\omega})}\widetilde{q}^{(\ell)}(\boldsymbol{\omega}+2\pi A^{-T}\boldsymbol{\eta}_j) = \begin{cases} 1, & j = 0, \\ 0, & j = 1, 2, \end{cases}$$

where $\boldsymbol{\eta}_j, j = 0, 1, 2$ are the representatives of the group $\mathbb{Z}^2/(A^T\mathbb{Z}^2)$:

$$\boldsymbol{\eta}_0 = (0,0),\ \boldsymbol{\eta}_1 = (1,0),\ \boldsymbol{\eta}_2 = (-1,0). \tag{2}$$

Then $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, generate bi-frames of $L^2(\mathbb{R}^2)$ provided that $\phi, \widetilde{\phi} \in L^2(\mathbb{R}^2)$ with $\widehat{\phi}(0,0)\widehat{\widetilde{\phi}}(0,0) \neq 0$, and that $p(0,0) = \widetilde{p}(0,0) = 1$, $p(2\pi A^{-T}\boldsymbol{\eta}_j) = \widetilde{p}(2\pi A^{-T}\boldsymbol{\eta}_j) = q^{(\ell)}(0,0) = \widetilde{q}^{(\ell)}(0,0) = 0$ for $j = 1, 2$ (see [39] and also [10, 12]).

A pair of frame filter banks $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ provides a frame multiresolution algorithm for regular meshes. More precisely, for an input regular mesh $\mathcal{C} = \{c_\mathbf{k}^0\}$

with regular vertices $c_{\mathbf{k}}^0$ (namely, the valence of each $c_{\mathbf{k}}^0$ is 6), the multiresolution decomposition (analysis) algorithm with dilation matrix A is

$$c_{\mathbf{n}}^j = \frac{1}{3}\sum_{\mathbf{k}\in\mathbb{Z}^2} p_{\mathbf{k}-A\mathbf{n}} c_{\mathbf{k}}^{j-1},\ d_{\mathbf{n}}^{(\ell,j)} = \frac{1}{3}\sum_{\mathbf{k}\in\mathbb{Z}^2} q_{\mathbf{k}-A\mathbf{n}}^{(\ell)} c_{\mathbf{k}}^{j-1}, \tag{3}$$

with $\ell = 1,\cdots,L, \mathbf{n}\in\mathbb{Z}^2$, where j is the level of decomposition with $j = 1,2,\cdots,J$ for some positive integer J. The multiresolution reconstruction (synthesis) algorithm is given by

$$\hat{c}_{\mathbf{k}}^{j-1} = \sum_{\mathbf{n}\in\mathbb{Z}^2} \widetilde{p}_{\mathbf{k}-A\mathbf{n}} \hat{c}_{\mathbf{n}}^j + \sum_{1\le\ell\le L}\sum_{\mathbf{n}\in\mathbb{Z}^2} \widetilde{q}_{\mathbf{k}-A\mathbf{n}}^{(\ell)} d_{\mathbf{n}}^{(\ell,j)}, \quad \mathbf{k}\in\mathbb{Z}^2 \tag{4}$$

for $j = J, J-2,\cdots,1$, where $\hat{c}_{\mathbf{n}}^J = c_{\mathbf{n}}^J$. When $\{p, q^{(1)},\cdots,q^{(L)}\}$ and $\{\widetilde{p},\widetilde{q}^{(1)},\cdots,\widetilde{q}^{(L)}\}$ are biorthogonal, then $\hat{c}_{\mathbf{k}}^j = c_{\mathbf{k}}^j$, $1\le j\le J$. $\{p, q^{(1)},\cdots,q^{(L)}\}$ ($\{\widetilde{p},\widetilde{q}^{(1)},\cdots,\widetilde{q}^{(L)}\}$ resp.) is called the analysis (synthesis resp.) frame filter bank; $\{c_{\mathbf{k}}^j\}$ and $\{d_{\mathbf{k}}^{(\ell,j)}\}$ are called the "approximation" and the "detail" of $\mathcal{C}$. In this paper, we will consider frames with 3 framelets. Recall that a $\sqrt{3}$-refinement biorthogonal system has 2 analysis or synthesis wavelets. Thus, compared with biorthogonal systems, our frames have only one more generator.

The above analysis and synthesis algorithms are for regular vertices only. However, an input triangular mesh to be processed has in general an arbitrary topology, namely, it consists of not only regular vertices but also extraordinary vertices with valences $\neq 6$. Thus we need to design corresponding algorithms for extraordinary vertices. On the other hand, the multiresolution algorithms should be given by templates so that they are easy to implement. Our procedure to design multiresolution algorithms for surfaces with an arbitrary topology will be as follows. (i) Firstly, we construct frame algorithms for regular vertices with the algorithms given by symmetric templates. (ii) After that, we design algorithm templates for extraordinary vertices with the algorithms in (i) applied to regular vertices.

The algorithm templates in Step (i) are given by some parameters. We should choose the parameters such that the resulting framelets have certain nice properties such as high approximation order, smoothness, and vanishing moments. These properties are determined by filters. Thus we need to address the issue of how to find the filter banks corresponding to algorithms given by templates.

The rest of paper is outlined as follows. In §2, we show how to find the filter banks corresponding to algorithms given by templates. In §3, we construct symmetric bi-frames with a 3-step algorithm based on symmetric templates. Interpolatory $\sqrt{3}$ subdivision-based frame algorithms and framelets are constructed in §4. In §5, we address the treatment of boundary vertices. In the last section, §6, one experimental result with the framelets used for surface denoising is provided and the future work in this research direction is presented.

2 6-fold line symmetric $\sqrt{3}$ bi-frames and associated templates

As mentioned above it is desirable that we can represent multiresolution algorithms as templates so that we can easily implement them. When "detail" $d_{\mathbf{k}}^{(\ell,j)}$ is set to zero, (4) is reduced to

$$\hat{c}_{\mathbf{k}}^{j-1} = \sum_{\mathbf{n}\in\mathbb{Z}^2} \widetilde{p}_{\mathbf{k}-A\mathbf{n}} \hat{c}_{\mathbf{n}}^j, \quad j = J, J-2,\cdots.$$

This is a $\sqrt{3}$-subdivision algorithm with subdivision mask $\{\widetilde{p}_{\mathbf{k}}\}_{\mathbf{k}}$ starting with the control net with vertices $\hat{c}^J_{\mathbf{k}}$. (See [43, 48] and references therein about surface subdivision.) It is hard to implement this subdivision algorithm if it is given by the above formula, and we need to represent it as templates. For example, when nonzero $p_{\mathbf{k}}$ are

$$p_{0,0} = \tfrac{2}{3}, p_{1,0} = p_{1,1} = p_{0,1} = p_{-1,0} = p_{-1,-1} = p_{0,-1} = \tfrac{1}{3}$$
$$p_{2,1} = p_{1,2} = p_{1,-1} = p_{-2,-1} = p_{-1,-2} = p_{-1,1} = \tfrac{1}{18},$$

we have Kobbelt's $\sqrt{3}$-subdivision scheme (for regular vertices)[28], which can be represented as templates in Fig. 2, where $\widetilde{v}, \widetilde{v}_j$ denotes old vertices in the coarse mesh, f is the inserted new vertex in the finer mesh, and v is the updated vertex (in the finer mesh) of $\widetilde{v}$ with

$$f = \frac{1}{3}(\widetilde{v}_0 + \widetilde{v}_1 + \widetilde{v}_2),\ v = \frac{2}{3}\widetilde{v} + \frac{1}{18}\sum_{j=0}^{5} \widetilde{v}_j.$$

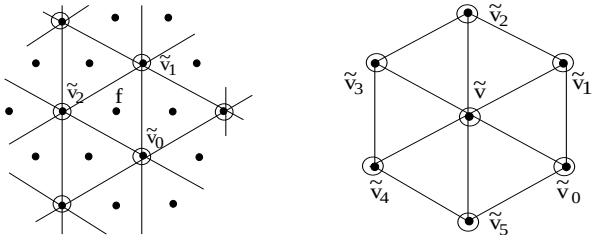

Figure 2: *Kobbelt's subdivision scheme: Templates for new inserted vertex f* (left) *and for updating old regular vertex $\widetilde{v}$* (right)

Similarly, for a given pair of filter banks, we cannot implement its multiresolution algorithm given by (3) and (4). Thus we need to represent the algorithm (3) and (4) as templates. Vice versa, if the algorithm is given by templates in terms of some parameters, we need to find the corresponding filter banks (equivalently, $p_{\mathbf{k}}, q^{(\ell)}_{\mathbf{k}}, \widetilde{p}_{\mathbf{k}}, \widetilde{q}^{(\ell)}_{\mathbf{k}}$ in (3) and (4)) so that we can select suitable parameters based on the property of the framelets.

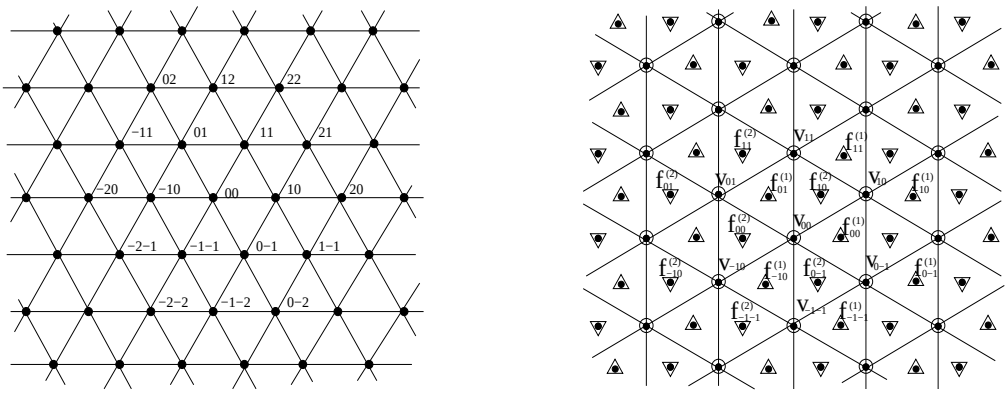

Figure 3: Left: *Indices for nodes of $\mathcal{M}_0$;* Right: *type V nodes with $\bigcirc$, type F nodes with $\triangle$ and ∇*

In this section we show how to find a pair of frame filter banks corresponding to decomposition and reconstruction algorithms for regular vertices given by templates. The key is to associate

both lowpass and highpass outputs appropriately with the nodes of $\mathcal{M}_0$, where $\mathcal{M}_0$ is the infinite regular triangular mesh on the left of Fig. 1 with which an input regular mesh $\mathcal{C} = \{c_{\mathbf{k}}\}_{\mathbf{k}\in\mathbb{Z}^2}$ is represented. To this regard, we first label the nodes of $\mathcal{M}_0$ with indices in $\mathbb{Z}^2$ shown on the left of Fig. 3. Next, we separate the nodes of $\mathcal{M}_0$ into different groups. More precisely, let A be the dilation matrix defined by (1). Denote $A\mathbf{k} = (2k_1 - k_2, k_1 + k_2)$ for $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$. Then $A\mathbf{k}$ with $\mathbf{k} \in \mathbb{Z}^2$ are the indices for the nodes of the coarse mesh on the right of Fig. 1. We call the nodes with indices $A\mathbf{k}$ *type V* nodes, and the others *type F* nodes. Furthermore, we then separate *type F* nodes into two groups with indices $A\mathbf{k} + (1, 0)$ and $A\mathbf{k} - (1, 0)$ respectively, where $\mathbf{k} \in \mathbb{Z}^2$. Denote

$$v_{\mathbf{k}} = c_{A\mathbf{k}},\ f^{(1)}_{\mathbf{k}} = c_{A\mathbf{k}+(1,0)},\ f^{(2)}_{\mathbf{k}} = c_{A\mathbf{k}-(1,0)},\ \mathbf{k} \in \mathbb{Z}^2. \tag{5}$$

We call $v_{\mathbf{k}}, \mathbf{k} \in \mathbb{Z}^2$ *type V* vertices, and both $f^{(1)}_{\mathbf{k}}$ and $f^{(2)}_{\mathbf{k}}$ *type F* vertices. See the right of Fig. 3 for the nodes with which these vertices associated, where the big circles $\bigcirc$ denote *type V* nodes, $\triangle$ and ∇ denote two groups of *type F* nodes. Next we rewrite the $\sqrt{3}$ frame decomposition algorithm.

Since the same $p_{\mathbf{k}}, q^{(\ell)}_{\mathbf{k}}, \widetilde{p}_{\mathbf{k}}, \widetilde{q}^{(\ell)}_{\mathbf{k}}$ are used in all levels of decomposition and reconstruction, we just need to consider 1-level of decomposition and reconstruction when we derive the corresponding templates. Let $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$ and $\{d^{(1,1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{d^{(2,1)}_{\mathbf{k}}\}_{\mathbf{k}}, \{d^{(3,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ be the "approximation" and "detail" after the 1-level decomposition algorithm given by (3) with $L = 3$. Denote

$$\widetilde{v}_{\mathbf{k}} = c^1_{\mathbf{k}},\ \widetilde{g}_{\mathbf{k}} = d^{(1,1)}_{\mathbf{k}},\ \widetilde{f}^{(1)}_{\mathbf{k}} = d^{(2,1)}_{\mathbf{k}},\ \widetilde{f}^{(2)}_{\mathbf{k}} = d^{(3,1)}_{\mathbf{k}}.$$

Then the 1-level decomposition algorithm can be formulated as

$$\begin{cases} \widetilde{v}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbb{Z}^2} p_{\mathbf{k}'-A\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{g}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbb{Z}^2} q^{(1)}_{\mathbf{k}'-A\mathbf{k}} c_{\mathbf{k}'}, \\ \widetilde{f}^{(1)}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbb{Z}^2} q^{(2)}_{\mathbf{k}'-A\mathbf{k}} c_{\mathbf{k}'}, \widetilde{f}^{(2)}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbb{Z}^2} q^{(3)}_{\mathbf{k}'-A\mathbf{k}} c_{\mathbf{k}'} \end{cases} \tag{6}$$

for $\mathbf{k} \in \mathbb{Z}^2$; and the reconstruction algorithm (after 1-level decomposition) is

$$c_{\mathbf{k}} = \sum_{\mathbf{k}'} \left\{ \widetilde{p}_{\mathbf{k}-A\mathbf{k}'} \widetilde{v}_{\mathbf{k}'} + \widetilde{q}^{(1)}_{\mathbf{k}-A\mathbf{k}'} \widetilde{g}_{\mathbf{k}'} + \widetilde{q}^{(2)}_{\mathbf{k}-A\mathbf{k}'} \widetilde{f}^{(1)}_{\mathbf{k}'} + \widetilde{q}^{(3)}_{\mathbf{k}-A\mathbf{k}'} \widetilde{f}^{(2)}_{\mathbf{k}'} \right\}. \tag{7}$$

Considering $c_{\mathbf{k}}$ in (7) with $\mathbf{k}$ in three different cases: $A\mathbf{j}$, $A\mathbf{j} + (1, 0)$, $A\mathbf{j} - (1, 0)$, and using the notations for $v_{\mathbf{k}}, f^{(1)}_{\mathbf{k}}, f^{(2)}_{\mathbf{k}}$ in (5), we can write the reconstruction algorithm (7) as

$$\begin{cases} v_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbb{Z}^2} \left\{ \widetilde{p}_{A\mathbf{n}} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{A\mathbf{n}} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{A\mathbf{n}} \widetilde{f}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{A\mathbf{n}} \widetilde{f}^{(2)}_{\mathbf{k}-\mathbf{n}} \right\}, \\ f^{(1)}_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbb{Z}^2} \left\{ \widetilde{p}_{A\mathbf{n}+(1,0)} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{A\mathbf{n}+(1,0)} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{A\mathbf{n}+(1,0)} \widetilde{f}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{A\mathbf{n}+(1,0)} \widetilde{f}^{(2)}_{\mathbf{k}-\mathbf{n}} \right\}, \\ f^{(2)}_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbb{Z}^2} \left\{ \widetilde{p}_{A\mathbf{n}-(1,0)} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{A\mathbf{n}-(1,0)} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{A\mathbf{n}-(1,0)} \widetilde{f}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{A\mathbf{n}-(1,0)} \widetilde{f}^{(2)}_{\mathbf{k}-\mathbf{n}} \right\}. \end{cases} \tag{8}$$

If we associate both the "approximation" $\widetilde{v}_{\mathbf{k}}$ and the first highpass output $\widetilde{g}_{\mathbf{k}}$ with *type V* nodes with labels $A\mathbf{k}$, and associate the second and third highpass outputs $\widetilde{f}^{(1)}_{\mathbf{k}}, \widetilde{f}^{(2)}_{\mathbf{k}}$ with *type F* nodes with labels $A\mathbf{k} + (1, 0)$ and $A\mathbf{k} - (1, 0)$ respectively, then the analysis algorithm (6) and synthesis algorithm (8) can be represented as templates. Vice versa, with such association of outputs with nodes, for given templates of algorithms, we can find corresponding $p_{\mathbf{k}}, q^{(\ell)}_{\mathbf{k}}$ in (6) and $\widetilde{p}_{\mathbf{k}}, \widetilde{q}^{(\ell)}_{\mathbf{k}}$ in (8), namely we can obtain the corresponding filter banks.

When algorithm templates are used for surface processing, they must have certain symmetry so that we can design the corresponding algorithms for extraordinary vertices. The templates

to obtain $\widetilde{f}^{(1)}_{\mathbf{k}}, \widetilde{f}^{(2)}_{\mathbf{k}}$ must be the same and the templates to recover $f^{(1)}_{\mathbf{k}}, f^{(2)}_{\mathbf{k}}$ should be identical. In addition, the templates to obtain $\widetilde{v}_{\mathbf{k}}$ and $\widetilde{g}_{\mathbf{k}}$ by (6), and that to recover $v_{\mathbf{k}}$ by (4) have to be rotational and reflective invariant with respect to the coarse mesh. Furthermore, the template to obtain $\widetilde{f}^{(1)}_{\mathbf{k}}$ and that to recover $f^{(1)}_{\mathbf{k}}$ are also rotational and reflective invariant with respect to the coarse mesh. $\sqrt{3}$ frame filter banks with the 6-fold (axial) line symmetry (defined below) will result in templates with such desired symmetry.

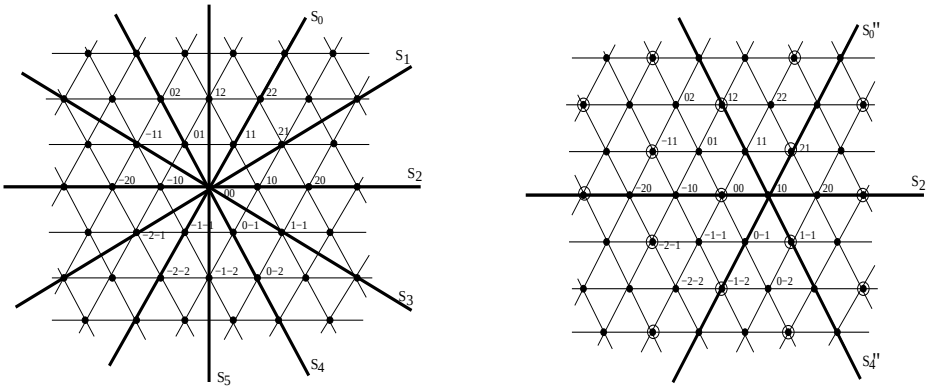

Figure 4: Left: *Symmetry lines for lowpass filter and 1st frame highpass filter;* Right: *Symmetry lines for 2nd frame highpass filter*

Definition 1. *Let $S_j, 0 \le j \le 5$ be the axes in the left part of Fig. 4. A $\sqrt{3}$ frame filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ is said to have* 6**-fold axial (line) symmetry** *or a* **full set of symmetries** *if* (i) *coefficients $p_{\mathbf{k}}$ and $q^{(1)}_{\mathbf{k}}$ of its lowpass filter $p(\boldsymbol{\omega})$ and first highpass filter $q^{(1)}(\boldsymbol{\omega})$ are symmetric around axes $S_0, \cdots, S_5$,* (ii) *the coefficients $q^{(2)}_{\mathbf{k}}$ of its second highpass filter are symmetric around the axes S_0'', S_2, S_4'' on the right of Fig. 4, and* (iii) *$q^{(3)}_{\mathbf{k}}$ is the π rotation around $(0,0)$ of $q^{(2)}_{\mathbf{k}}$, i.e., $q^{(3)}_{\mathbf{k}} = q^{(2)}_{-\mathbf{k}}$.*

The 6-fold symmetry of a $\sqrt{3}$ frame filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ can be characterized by the symmetry of its polyphase matrix $V(\boldsymbol{\omega})$ which is a 4×3 matrix defined as

$$V(\boldsymbol{\omega}) = \left[q^{(\ell)}_k(\boldsymbol{\omega})\right]_{0\le\ell\le 3, 0\le k\le 2}, \tag{9}$$

where, with $q^{(0)}(\boldsymbol{\omega}) = p(\boldsymbol{\omega})$, $q^{(\ell)}_k(\boldsymbol{\omega}), 0 \le \ell \le 3, 0 \le k \le 3$ are trigonometric polynomials defined by

$$q^{(\ell)}(\boldsymbol{\omega}) = \frac{1}{\sqrt{3}}\Big(q^{(\ell)}_0(A^T\boldsymbol{\omega}) + q^{(\ell)}_1(A^T\boldsymbol{\omega})e^{-i\omega_1} + q^{(\ell)}_2(A^T\boldsymbol{\omega})e^{i\omega_1}\Big).$$

Proposition 1. *A $\sqrt{3}$ frame filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if its polyphase matrix $V(\boldsymbol{\omega})$ (with dilation matrix A) satisfies*

$$V(L_0\boldsymbol{\omega}) = J_{01}V(\boldsymbol{\omega})J_{02}, \quad V(R_1^{-T}\boldsymbol{\omega}) = N_1(\boldsymbol{\omega})V(\boldsymbol{\omega})N_2(\boldsymbol{\omega}), \tag{10}$$

where

$$L_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ R_1 = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, J_{01} = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & L_0 \end{bmatrix},\ J_{02} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & L_0 \end{bmatrix},$$

$$N_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & e^{-i\omega_1} \\ 0 & 0 & e^{i\omega_1} & 0 \end{bmatrix}, N_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & e^{-i\omega_1} \\ 0 & e^{i\omega_1} & 0 \end{bmatrix}.$$

One can give the proof of Proposition 1 by following the similar proof in [24] for the characterization of six-fold symmetry of $\sqrt{3}$ wavelet filter banks and by using the fact that

$$[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T = \frac{1}{\sqrt{3}} V(A^T\boldsymbol{\omega}) I_0(\boldsymbol{\omega}),$$

where $I_0(\boldsymbol{\omega})$ is defined by

$$I_0(\boldsymbol{\omega}) = [1,\ e^{-i\omega_1},\ e^{i\omega_1}]^T,\ \boldsymbol{\omega} = (\omega_1, \omega_2) \in \mathbb{R}^2. \tag{11}$$

Since the pair $\widetilde{f}_{\mathbf{k}}^{(1)}$, $\widetilde{f}_{\mathbf{k}}^{(2)}$, and the pair $f_{\mathbf{k}}^{(1)}$, $f_{\mathbf{k}}^{(2)}$ are treated equally, and the templates to obtain $\widetilde{f}_{\mathbf{k}}^{(1)}, \widetilde{f}_{\mathbf{k}}^{(2)}$ are the same, and those to recover $f_{\mathbf{k}}^{(1)}, f_{\mathbf{k}}^{(2)}$ are identical, we may use v, f and $\widetilde{v}, \widetilde{g}, \widetilde{f}$ to describe the multiresolution algorithms. Therefore, the decomposition algorithm decomposes the original data $\{v\} \cup \{f\}$ into $\{\widetilde{v}\}$, $\{\widetilde{g}\}$ and $\{\widetilde{f}\}$, and the reconstruction algorithm recovers $\{v\} \cup \{f\}$ from $\{\widetilde{v}\}$, $\{\widetilde{g}\}$ and $\{\widetilde{f}\}$, see Fig. 5.

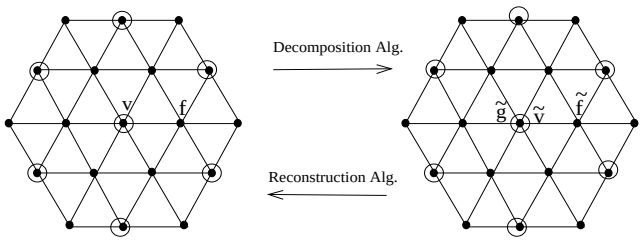

Figure 5: $\sqrt{3}$ *frame decomposition and reconstruction algorithms*

For an input (fine) triangular surface, we assume it has a $\sqrt{3}$-refinement connectivity. Namely, vertices of the surface can be separated into two groups, with one group consisting of *type V* vertices and the other consisting of *type F* such that the *type V* vertices form a coarse triangular mesh and each coarse triangle "contains" one *type F* vertex. Methods to resample a triangular mesh with the resulting mesh having the 1-to-4 subdivision connectivity have been developed in [11, 15, 30, 35, 14]. For $\sqrt{3}$-refinement, the software (called *TriReme*) developed by Guskov with the modified method in [14] can produce a mesh with $\sqrt{3}$-refinement connectivity with guaranteed errors. The resulting mesh may have extraordinary vertices. When a mesh with an arbitrary topology and $\sqrt{3}$-connectivity is used as the input mesh for multiresolution processing, we also use v, f and $\widetilde{v}, \widetilde{g}, \widetilde{f}$ to describe the multiresolution algorithms, where $\{v\}$ is the set of *type V* vertices forming the coarse mesh, $\{f\}$ is the set of *type F* vertices, $\{\widetilde{v}\}$ is the "approximation", and $\{\widetilde{g}\}, \{\widetilde{f}\}$ are the "details" with $\widetilde{v}, \widetilde{g}$ attached to *type V* vertex v, and $\widetilde{f}$ attached to *type F* vertex f.

3 $\sqrt{3}$-refinement frame multiresolution algorithms

In this section we study a 3-step frame multiresolution algorithm. The 3-step algorithm is given by (12)-(17) with templates shown in Figs. 6 and 7, where k is the valence of a *type V* vertex v, $b(k), d(k), n(k), a, h, w(k), n_1(k), t$ are constants to be determined with $b(k), d(k), n(k), w(k), n_1(k)$ depending on k. More precisely, for the decomposition algorithm, first we replace each *type V* vertex v by v'', g'' given by (12). Then, based on v'', g'' obtained, we replace all *type F* vertices f by $\widetilde{f}$ given in formula (13). Finally, based on $\widetilde{f}$ obtained, all v'', g'' in the first step are updated by $\widetilde{v}$ and $\widetilde{g}$ given in formula (14). The reconstruction algorithm is the reverse algorithm of the decomposition algorithm. Namely, firstly, we replace the lowpass output $\widetilde{v}$ and "detail" $\widetilde{g}$ both associated with a *type V* vertex by v'' and g'' respectively given by formula (15). Secondly, based on v'', g'' obtained, we replace other "detail" $\widetilde{f}$ by f given in (16). Finally, based on f obtained in Step 2, all v'', g'' in Step 1 are replaced by v with the formula in (17). The decomposition algorithm to obtain "approximation" $\{\widetilde{v}\}$ and "detail" $\{\widetilde{g}\}$ and $\{\widetilde{f}\}$, and the reconstruction algorithm to recover $\{v, f\}$ from $\{\widetilde{v}\}, \{\widetilde{g}\}$ and $\{\widetilde{f}\}$ are simple and efficient.

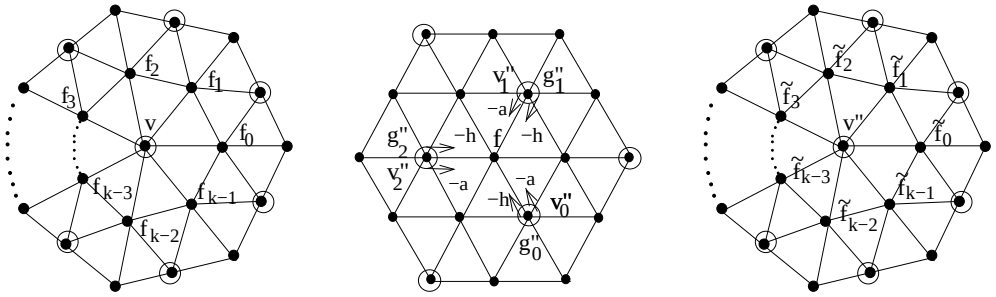

Figure 6: Left: *Template to obtain v'' or g'' in Decomposition Alg. Step 1;* Middle: *Decomposition Alg. Step 2;* Right: *Template to obtain lowpass output $\widetilde{v}$ in Decomposition Alg. Step 3 (Template to obtain 1st highpass output $\widetilde{g}$ is similar with v'' replaced by g'')*

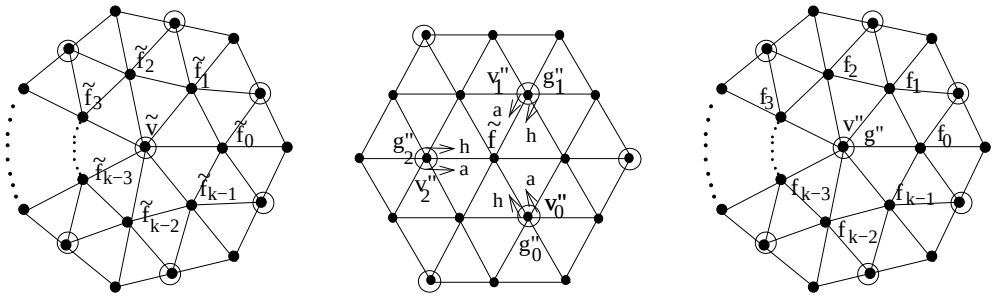

Figure 7: Left: *Template to obtain v'' in Reconstruction Alg. Step 1 (Template to obtain g'' is similar with $\widetilde{v}$ replaced by $\widetilde{g}$);* Middle: *Template to obtain type F vertices f in Reconstruction Alg. Step 2;* Right: *Template to obtain type V vertices v in Reconstruction Alg. Step 3*

$\sqrt{3}$-refinement Decomposition Algorithm:

$$\textbf{Step 1.} \quad v'' = \frac{1}{b(k)}\big\{v - d(k)\sum_{j=0}^{k-1} f_j\big\},\ g'' = v - n(k)\sum_{j=0}^{k-1} f_j, \tag{12}$$

$$\textbf{Step 2.} \quad \widetilde{f} = f - a(v_0'' + v_1'' + v_2'') - h(g_0'' + g_1'' + g_2'') \tag{13}$$

$$\textbf{Step 3.} \quad \widetilde{v} = v'' - w(k)\sum_{j=0}^{k-1} \widetilde{f}_j,\ \widetilde{g} = g'' - n_1(k)\sum_{j=0}^{k-1} \widetilde{f}_j. \tag{14}$$

$\sqrt{3}$-refinement Reconstruction Algorithm:

$$\textbf{Step 1.} \quad v'' = \widetilde{v} + w(k)\textstyle\sum_{j=0}^{k-1} \widetilde{f}_j,\ g'' = \widetilde{g} + n_1(k)\sum_{j=0}^{k-1} \widetilde{f}_j \tag{15}$$

$$\textbf{Step 2.} \quad f = \widetilde{f} + a(v_0'' + v_1'' + v_2'') + h(g_0'' + g_1'' + g_2'') \tag{16}$$

$$\textbf{Step 3.} \quad v = t\big\{b(k)v'' + d(k)\textstyle\sum_{j=0}^{k-1} f_j\big\} + (1-t)\big\{g'' + n(k)\sum_{j=0}^{k-1} f_j\big\}. \tag{17}$$

The above templates are for 1-level decomposition and reconstruction. For more than 1 level decomposition and reconstruction, one merely applies the decomposition templates to the "approximation" to get further "approximation" and more "details", and then uses the templates of reconstruction on the further "approximation" and all "details" for reconstruction.

Next, we study how to select the parameters. To this regard, we first consider the algorithms for regular vertices. Denote

$$b = b(6), d = d(6), n = n(6), w = w(6), n_1 = n_1(6).$$

With the formulas in (6) and (8), one can obtain that the filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ corresponding to the algorithms (12)-(17) with $k = 6$. The filter banks are provided in Appendix A. With the filter banks, we can easily obtain their polyphase matrices $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ to be $\sqrt{3}B_2(\boldsymbol{\omega})B_1(\boldsymbol{\omega})B_0(\boldsymbol{\omega})$ and $\frac{1}{\sqrt{3}}\widetilde{B}_2(\boldsymbol{\omega})\widetilde{B}_1(\boldsymbol{\omega})\widetilde{B}_0(\boldsymbol{\omega})$ respectively (see Appendix A for $B_j, \widetilde{B}_j, j = 0, 1, 2$). It is easy to verify $B_0(\boldsymbol{\omega})^*\widetilde{B}_0(\boldsymbol{\omega}) = I_3$, $B_j(\boldsymbol{\omega})^*\widetilde{B}_j(\boldsymbol{\omega}) = I_4, \boldsymbol{\omega} \in \mathbb{R}^2$ for $j = 1, 2$. Thus, $V(\boldsymbol{\omega})^*\widetilde{V}(\boldsymbol{\omega}) = I_3, \boldsymbol{\omega} \in \mathbb{R}^2$, which implies that $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are biorthogonal. In addition, one can show that $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ satisfy (10). Hence, $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are 6-fold axial symmetric.

With filter banks available, we then select the parameters such that the resulting framelets have some nice properties such as high sum rule orders, smoothness, and vanishing moments. An FIR filter $p(\boldsymbol{\omega})$ is said to have **sum rule order** K (with a dilation matrix A) if it satisfies that $p(0, 0) = 1$ and

$$D_1^{\alpha_1} D_2^{\alpha_2} p(\boldsymbol{\omega})|_{\boldsymbol{\omega}=(\frac{2\pi i}{3}, \frac{2\pi i}{3})} = 0,\ D_1^{\alpha_1} D_2^{\alpha_2} p(\boldsymbol{\omega})|_{\boldsymbol{\omega}=(\frac{4\pi i}{3}, \frac{4\pi i}{3})} = 0, \tag{18}$$

for all $(\alpha_1, \alpha_2) \in \mathbb{Z}_+^2$ with $\alpha_1 + \alpha_2 < K$, where D_1 and D_2 denote the partial derivatives with the first and second variables of $p(\boldsymbol{\omega})$ respectively. Sum rule order implies the approximation order of the associated scaling function ϕ [19].

For an FIR (highpass) filter $q(\boldsymbol{\omega})$, we say it has the **vanishing moments of order** J if

$$D_1^{\alpha_1} D_2^{\alpha_2} q(\boldsymbol{\omega})|_{\boldsymbol{\omega}=(0,0)} = 0,$$

for all $(\alpha_1, \alpha_2) \in \mathbb{Z}_+^2$ with $\alpha_1 + \alpha_2 < J$. One can prove that if $q(\boldsymbol{\omega})$ has vanishing moment order J, then when it is used as an analysis highpass filter, it annihilates discrete polynomials of total degree less than J. Annihilation of discrete polynomials is an important property of highpass filters in many applications such as image/surface sparse representation.

The smoothness of the synthesis framelets is determined by the scaling function $\widetilde{\phi}$ (also called the subdivision basis function). The visual quality of the reconstructed surface depends on the smoothness of $\widetilde{\phi}$. In this paper we use Sobolev smoothness. We say $g(\mathbf{x})$ on $\mathbb{R}^2$ to be in the Sobolev space W^s for some $s > 0$ provided that $\int_{\mathbb{R}^2}(1 + |\boldsymbol{\omega}|^2)^s |\widehat{g}(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} < \infty$, where $\widehat{g}$ is the Fourier transform of g. The reader is referred to [21, 20, 25] for the estimate and computation of the Sobolev smoothness of a refinable function.

In the following, when we construct bi-frames, we choose the parameters such that the synthesis scaling function $\widetilde{\phi}$ is smoother than the analysis scaling function ϕ, the synthesis lowpass filter $\widetilde{p}(\boldsymbol{\omega})$ has a higher sum rule order than the analysis lowpass filter $p(\boldsymbol{\omega})$, and that the analysis highpass filters $q^{(\ell)}(\boldsymbol{\omega})$ have higher orders of vanishing moments.

Now let us return back to the filter banks for algorithms (12)-(17) with $k = 6$. Solving the system of equations for sum rule order 3 of $\widetilde{p}$, sum rule order 2 of p, and for vanishing moment order 2 of $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \le \ell \le 3$, we have

$$a = \tfrac{1}{3},\ n = \tfrac{1}{6},\ w = -\tfrac{5}{36},\ d = \tfrac{1}{6} - \tfrac{b}{6},\ h = \tfrac{2}{15} - \tfrac{1}{15b},\ t = \tfrac{1}{2b}.$$

With $x = e^{-i\omega_1}, y = e^{-i\omega_2}$, the resulting $\widetilde{p}$ is

$$\widetilde{p}(\boldsymbol{\omega}) = \tfrac{1}{3}\Big\{\tfrac{2}{3} + \tfrac{1}{3}(x + y + \tfrac{1}{xy} + \tfrac{1}{x} + \tfrac{1}{y} + xy) + \tfrac{1}{18}(x^2y + xy^2 + \tfrac{x}{y} + \tfrac{y}{x} + \tfrac{1}{x^2y} + \tfrac{1}{xy^2})\Big\}. \tag{19}$$

The resulting $\widetilde{p}$ is the Kobbelt's $\sqrt{3}$-subdivision scheme in [28] with the corresponding $\widetilde{\phi} \in W^{2.93604}$. $p(\boldsymbol{\omega})$ depends on b. If $b = 2$, then $p(\boldsymbol{\omega})$ has sum rule order 3 with $\phi \in W^{1.80016}$. If we choose $b = \frac{55}{27}$, then the resulting ϕ is in $W^{1.85267}$. For n_1, we choose $n_1 = \frac{5b}{36(2b-1)}$ so that the resulting $q^{(1)}$ has vanishing moment order 4. In this paper we choose $b = 2$. In the following we list the other corresponding parameters:

$$a = \tfrac{1}{3},\ n = \tfrac{1}{6},\ w = -\tfrac{5}{36},\ d = -\tfrac{1}{6},\ h = \tfrac{1}{10},\ t = \tfrac{1}{4}, n_1 = \tfrac{5}{54}. \tag{20}$$

We also provide the corresponding filters in Appendix B. The lowpass filters $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ in the above pair of bi-frame filter banks are supported on $[-2, 2]^2$ with $\widetilde{p}(\boldsymbol{\omega})$ being the filter for Kobbelt's scheme. Thus, we call the resulting framelets **Kobbelt's scheme-based bi-framelets**, and we use Kobbelt-$\mathrm{F}_{2,2}$ to denote this pair of bi-frame filter banks.

After we determine the parameters for regular vertices, we consider algorithms for extraordinary vertices. Here we provide a Kobbelt's scheme-based algorithm. More precisely, Kobbelt-$\mathrm{F}_{2,2}$ constructed above will be applied to regular vertices and the reconstruction algorithm designed below for extraordinary vertices will be reduced to Kobbelt's scheme when the "detail" is set to be zero. To this regard, we first rewrite Kobbelt's scheme for extraordinary vertices.

Let $\{\widetilde{v}\}$ denote the set of vertices in a coarse mesh. Let $\{f\}$ be the set of new inserted vertices in the finer mesh after one $\sqrt{3}$ subdivision iteration, and $\{v\}$ be the set of vertices replacing old vertices $\widetilde{v}$. Then Kobbelt's scheme can be written as (refer to [47])

$$\begin{cases} f = \frac{1}{3}(\widetilde{v}_0 + \widetilde{v}_1 + \widetilde{v}_2), (\forall f) \\ v = \theta(k)\widetilde{v} + \frac{1}{k}\big(1 - \theta(k)\big)\sum_{j=0}^{k-1} f_j, \end{cases} \tag{21}$$

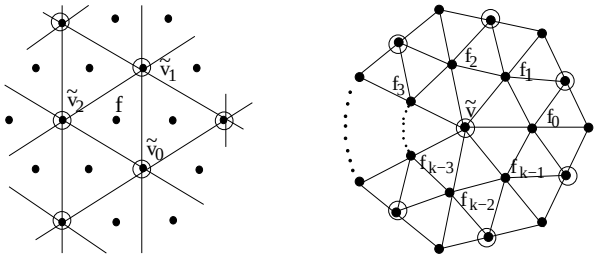

Figure 8: *Kobbelt's subdivision scheme*

where f denotes a new vertex inserted inside a triangle of the coarse mesh with vertices $\widetilde{v}_s, s = 0, 1, 2$, v is the vertex in the finer mesh which replaces $\widetilde{v}$ (see Fig. 8), k is the valence of v, and

$$\theta(k) = \frac{1}{3}(1 + \cos\frac{2\pi}{k}). \tag{22}$$

When the "detail" is set to zero: $\widetilde{g} = 0, \widetilde{f} = 0$, the reconstruction algorithm (15)-(17) is a subdivision algorithm:

$$\begin{cases} f = a(\widetilde{v}_0 + \widetilde{v}_1 + \widetilde{v}_2), (\forall f) \\ v = tb(k)\widetilde{v} + \big\{td(k) + (1-t)n(k)\big\}\sum_{j=0}^{k-1} f_j, \end{cases} \tag{23}$$

Comparing (23) with (21), $a = \frac{1}{3}$ (note that we already selected $a = \frac{1}{3}$ in (20)), and $tb(k) = \theta(k), td(k)+(1-t)n(k) = (1-\theta(k))/k$. In addition, we should choose the parameters such that the "detail" of constant input with decomposition algorithm (12)-(14) are annihilated: $\widetilde{g} = 0, \widetilde{f} = 0$. If $n(k) = 1/k$ and $d(k) = (1 - b(k))/k$, the "detail"-annihilation property is satisfied. With $t = \frac{1}{4}$ given in (20), we reach $b(k) = 4\theta(k), n(k) = 1/k$ and $d(k) = \big(1 - b(k)\big)/k$. For $w(k), n_1(k)$, we may simply choose $w(k) = -5/(6k)$, $n_1(k) = 5/(9k)$. With such choices of $w(k), n_1(k)$, the above algorithm with $k = 6$ coincides with Kobbelt-$\mathrm{F}_{2,2}$. To summarize, we choose

$$\begin{cases} a = \frac{1}{3}, h = \frac{1}{10}, b(k) = \frac{4}{3} + \frac{4}{3}\cos(\frac{2\pi}{k}), d(k) = \frac{1}{k}\big(1 - b(k)\big) \\ n(k) = \frac{1}{k}, w(k) = -\frac{5}{6k}, n_1(k) = \frac{5}{9k}, t = \frac{1}{4}. \end{cases} \tag{24}$$

3.1 2-step $\sqrt{3}$-refinement frame multiresolution algorithms

When $w(k) = n_1(k) = 0$ in (14) and (15), the above 3-step algorithm is reduced to a 2-step algorithm with the analysis algorithm given by (12) (13) (with $\widetilde{v} = v'', \widetilde{g} = g''$) and the synthesis algorithm given by (16) (17) (with $v'' = \widetilde{v}, g'' = \widetilde{g}$). For the regular vertices, with $b = b(6), d = d(6), n = n(6)$, one can find the corresponding filter banks, also denoted by $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$, are given by (48) in Appendix A with $B_2(\boldsymbol{\omega}) = \widetilde{B}_2(\boldsymbol{\omega}) = I_4$.

Solving the system of equations for sum rule order 3 of $\widetilde{p}$, sum rule order 2 of p, and for vanishing moment order 2 of $q^{(1)}$ and vanishing moment order 3 of $q^{(2)}$ and $q^{(3)}$, we have

$$b = 3,\ a = \frac{1}{3},\ n = \frac{1}{6},\ d = -\frac{1}{3},\ h = \frac{5}{9},\ t = \frac{1}{6}. \tag{25}$$

The resulting lowpass filter $\widetilde{p}$ is given by (19) and

$$p(\boldsymbol{\omega}) = \frac{1}{3}\Big\{1 + \frac{1}{3}(x + y + \frac{1}{xy} + \frac{1}{x} + \frac{1}{y} + xy)\Big\}, \tag{26}$$

with the corresponding ϕ in $W^{1.65713}$. Observe that the lowpass filter $p(\boldsymbol{\omega})$ is supported on $[-1,1]^2$ with $\widetilde{p}(\boldsymbol{\omega})$ being the filter for Kobbelt's scheme. We use Kobbelt-$\mathrm{F}_{1,2}$ to denote this pair of bi-frame filter banks. The corresponding highpass filters are provided in Appendix C.

For the parameters $b(k), d(k), n(k)$ for extraordinary vertices, as above, when the "details" in the 2-step reconstruction algorithm are set to 0, then the reconstruction algorithm is reduced to (23). If we choose $a = \frac{1}{3}$ (observe that we already selected $a = \frac{1}{3}$ in (25)), $tb(k) = \theta(k), td(k) + (1-t)n(k) = (1-\theta(k))/k$, then this has reached Kobbelt's subdivision scheme again. To assure that decomposition algorithm (12)-(13) annihilates constant input, we choose $n(k) = 1/k$ and $d(k) = (1-b(k))/k$. To summarize, we choose

$$\begin{cases} a = \frac{1}{3},\ h = \frac{5}{9},\ t = \frac{1}{6},\ b(k) = 2 + 2\cos(\frac{2\pi}{k}), \\ d(k) = \frac{1}{k}(1-b(k)),\ n(k) = \frac{1}{k}. \end{cases} \tag{27}$$

4 Interpolatory $\sqrt{3}$-subdivision-based frame multiresolution algorithms

In this section we study interpolatory $\sqrt{3}$-subdivision-based frame multiresolution algorithms such that the reconstruction algorithm is reduced to an interpolatory subdivision algorithm when the "detail" is set to zero. Thus the vertices of "approximation" lie on the reconstructed mesh after the synthesis algorithm even though during this process, some "detail" coefficients may be discarded.

The interpolatory scheme-based algorithm studied here is similar to the 3-step algorithm in §3 except that in Step 2, we need a larger template for f surrounded by regular *type V* vertices and a special template f near an extraordinary *type V* vertex. Here we assume that any two extraordinary vertices in the mesh to be decomposed are not adjacent. The decomposition algorithm is given by (28)-(30), and shown in Fig. 9, where $a, h, r, s, b(k), d(k), n(k)$, $\alpha(k), \beta(k), \gamma(k), \alpha_1(k), \beta_1(k), \gamma_1(k), w(k), n_1(k)$ are constants to be determined. Namely, firstly, each *type V* vertex v is replaced by v'', g'' with the formulas in (28). Secondly, based on v'', g'' obtained, we replace all *type F* vertices f by $\widetilde{f}$. If f is surrounded by three regular *type V* vertices (see the top-right of Fig. 9), then $\widetilde{f}$ is given by (29)(i). Otherwise, if f is adjacent an extraordinary vertex v with valence k, then $\widetilde{f}$ is given by the formula in (29)(ii). Finally, based on $\widetilde{f}$ obtained, all v'', g'' in the first step are updated by $\widetilde{v}$ and $\widetilde{g}$ given by the formula in (30). The reconstruction algorithm given by (31)-(33) is the reverse algorithm of the decomposition algorithm.

Interpolatory Scheme-based Decomposition Algorithm:

$$\textbf{Step 1.} v'' = \frac{1}{b(k)}\big\{v - d(k)\sum_{j=0}^{k-1} f_j\big\},\ g'' = v - n(k)\sum_{j=0}^{k-1} f_j \tag{28}$$

$$\textbf{Step 2.} \begin{cases} \text{(i) if } f \text{ is surrounded by 3 regular } \textit{type V} \text{ vertices:} \\ \quad \widetilde{f} = f - a\sum_{j=0}^{2} v_j'' - r\sum_{j=3}^{5} v_j'' - h\sum_{j=0}^{2} g_j'' - s\sum_{j=3}^{5} g_j'', \\ \text{(ii) if } f \text{ is adjacent an extraordinary } \textit{type V} \text{ vertex:} \\ \quad \widetilde{f} = f + a(v_0'' + v_1'' + v_2'') + h(g_0'' + g_1'' + g_2'') \end{cases} \tag{29}$$

$$\textbf{Step 3.} \begin{array}{l} \widetilde{f} = f - \alpha(k)v'' - \beta(k)(v_0'' + v_1'') - \gamma(k)(v_2'' + v_3'') \\ \qquad -\alpha_1(k)g'' - \beta_1(k)(g_0'' + g_1'') - \gamma_1(k)(g_2'' + g_3'') \end{array} \tag{30}$$

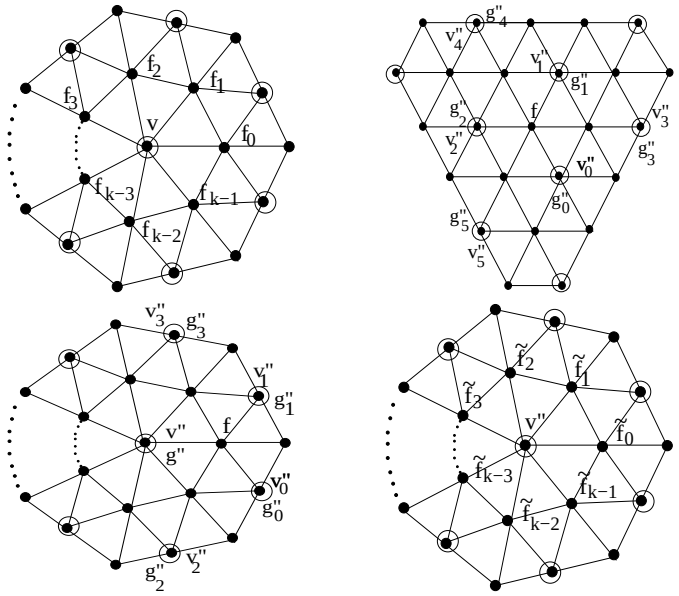

Figure 9: Top-left: *Template to obtain v'' and g'' in Decomposition Alg. Step 1;* Top-right: *Template in Decomposition Alg. Step 2 to obtain $\widetilde{f}$ when f is surrounded by three regular vertices v_0, v_1, v_2;* Bottom-left: *Template in Decomposition Alg. Step 2 to obtain $\widetilde{f}$ when f is adjacent an extraordinary vertex v;* Bottom-right: *Template in Decomposition Alg. Step 3 to obtain "approximation" $\widetilde{v}$* (*Template to obtain "detail" $\widetilde{g}$ is similar with v'' replaced by g''*)

Interpolatory Scheme-based Reconstruction Algorithm:

$$\textbf{Step 1.}\ \ v'' = \widetilde{v} + w(k)\sum_{j=0}^{k-1} f_j,\ \ g'' = \widetilde{g} + n_1(k)\sum_{j=0}^{k-1} f_j \tag{31}$$

$$\textbf{Step 2.}\ \begin{cases} \text{(i) if } \widetilde{f} \text{ is surrounded by 3 regular } \textit{type } V \text{ vertices:} \\ f = \widetilde{f} + a\sum_{j=0}^{2} v_j'' + r\sum_{j=3}^{5} v_j'' + h\sum_{j=0}^{2} g_j'' + s\sum_{j=3}^{5} g_j'', \\ \text{(ii) if } \widetilde{f} \text{ is adjancent an extraordinary } \textit{type } V \text{ vertex:} \\ f = \widetilde{f} + \alpha(k)v'' + \beta(k)(v_0'' + v_1'') + \gamma(k)(v_2'' + v_3'') \\ \quad +\alpha_1(k)g'' + \beta_1(k)(g_0'' + g_1'') + \gamma_1(k)(g_2'' + g_3'') \end{cases} \tag{32}$$

$$\textbf{Step 3.}\quad v = t\big\{b(k)v'' + d(k)\textstyle\sum_{j=0}^{k-1} f_j\big\} + (1-t)\big\{g'' + n(k)\textstyle\sum_{j=0}^{k-1} f_j\big\}. \tag{33}$$

Again, when $\widetilde{g} = 0, \widetilde{f} = 0$, then the above reconstruction algorithm is a subdivision algorithm:

$$\begin{cases} f = a\sum_{j=0}^{2}\widetilde{v}_j + r\sum_{j=3}^{5}\widetilde{v}_j \text{ (if } f \text{ is surrounded by 3 regular vertices } \widetilde{v}_0, \widetilde{v}_1, \widetilde{v}_2 \\ \text{or } f = \alpha(k)\widetilde{v} + \beta(k)(\widetilde{v}_0 + \widetilde{v}_1) + \gamma(k)(\widetilde{v}_2 + \widetilde{v}_3) \text{ (if } f \text{ is adjacent an extraordinary vertex } \widetilde{v}) \\ v = tb(k)\widetilde{v} + \big\{td(k) + (1-t)n(k)\big\}\sum_{j=0}^{k-1} f_j. \end{cases}$$

Furthermore, if $tb(k) = 1, td(k) + (1-t)n(k) = 0$, namely,

$$t = 1/b(k),\ d(k) = \big(b(k) - 1\big)n(k), \tag{34}$$

then we have an interpolatory scheme.

To choose the parameters, again, we first consider the regular case. If we choose $a = 4/9, r = -1/9$, then we reach an interpolatory scheme for regular vertices studied in [25] with resulting $\widetilde{p}$ having sum rule order 3 and $\widetilde{\phi} \in W^{1.8959}$. It was shown numerically in [25] that $\widetilde{\phi}$ is in $C^1(\mathbb{R}^2)$. To simplify the presentation of the paper, we call this scheme JO's interpolatory scheme. Next we choose other parameters such that ϕ has certain smoothness (at least it is in $L^2(\mathbb{R}^2)$), and the framelets have some vanishing moments. To do this, we first obtain the corresponding filter banks. With the notation, $b = b(6), d = d(6), n = n(6), w = w(6), n_1 = n_1(6)$, the corresponding frame filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are provided in Appendix D. With filter banks available and the interpolating condition (34), we then select the parameters based on the smoothness and the vanishing moments of framelets. Here we choose

$$[b,\ d,\ n,\ a,\ r,\ h,\ s,\ w,\ n_1,\ t] = [4,\ -\tfrac{1}{2},\ \tfrac{1}{6},\ \tfrac{4}{9}, -\tfrac{1}{9}, -\tfrac{13}{72},\ \tfrac{1}{18},\ -\tfrac{1}{9},\ \tfrac{4}{27},\ \tfrac{1}{4}]. \tag{35}$$

The resulting $\widetilde{p}$ is the lowpass filter for JO's scheme, p has sum rule order 3 with $\phi \in W^{2.62904}$, $\widetilde{q}^{(\ell)}, 1 \le \ell \le 3, q^{(2)}, q^{(3)}$ have vanishing moment order 2 and $q^{(1)}$ has vanishing moment order 3. In this case the coefficients $p_{\mathbf{k}}$ and $\widetilde{p}_{\mathbf{k}'}$ of $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ are zero when $\mathbf{k} \notin [-4, 4]^2$ and $\mathbf{k}' \notin [-2, 2]^2$ respectively. We use JO-$\mathrm{F}_{4,2}$ to denote this pair of frame filter banks.

After determining the parameters for regular vertices, we consider those for extraordinary vertices. The key is to select suitable $\alpha(k), \beta(k), \gamma(k)$ for the subdivision algorithm for extraordinary vertices. In general $\alpha(k), \beta(k), \gamma(k)$ are chosen to assure C^1-continuity of the subdivision limiting surface. [37] provides a sufficient condition for C^1-continuity based on the characteristic map introduced there. More precisely, if the eigenvalues $\lambda_0, \lambda_1, \lambda_2, \cdots$ of the subdivision matrix satisfy, $\lambda_0 = 1$, $1 > |\lambda_1| = |\lambda_2| > |\lambda_j|, j = 3, 4, \cdots$, and that the characteristic map is regular and injective, then the limiting surface is C^1. If the scaling function $\widetilde{\phi}$ is not a spline function, it is hard to verify the regularity and injectiveness of the characteristic map. In this paper, we choose $\alpha(k), \beta(k), \gamma(k)$ to be the values given in (36) below such that eigenvalues of the subdivision matrix satisfy the above conditions. The eigenvalue analysis of the subdivision matrix is provided in Appendix E.

$$\alpha(k) = \frac{11}{12} - \frac{1}{2}\cos\frac{2\pi}{k},\ \beta(k) = \frac{1}{12} + \frac{1}{4}\cos\frac{2\pi}{k},\ \gamma(k) = -\frac{1}{24}. \tag{36}$$

For $\alpha_1(k), \beta_1(k), \gamma_1(k)$, we may simply set them to be zero or be $\alpha(k), \beta(k), \gamma(k)$ respectively. After we choose $\alpha(k), \beta(k), \gamma(k)$ and $\alpha_1(k), \beta_1(k), \gamma_1(k)$, then based on the parameters for regular vertices and the annihilation of the "detail" with constant input, we choose other parameters as

$$b(k) = 4,\ d(k) = -\frac{3}{k},\ n(k) = \frac{1}{k},\ w(k) = -\frac{3}{2k},\ n_1(k) = \frac{8}{9k}. \tag{37}$$

Again, the above templates are for 1-level decomposition and reconstruction. For more than 1 level decomposition and reconstruction, we apply the decomposition templates repeatedly to get further "approximation" and more "details", and then use the reconstruction templates for reconstruction. If we apply the decomposition algorithm J times, the input mesh is required to have J-level $\sqrt{3}$ connectivity. Any two *type V* extraordinary vertices in the input mesh and in the j-th level "approximation" mesh are not adjacent for $0 \le j \le J-2$. Thus we can apply the decomposition algorithm (28)-(30) to these meshes. However, extraordinary vertices in the $(J-1)$-th level "approximation" mesh may be adjacent. In this case, we simply use (29) to define $\widetilde{f}$ in Step 2 of the J-th decomposition whether f is surrounded by 3 regular *type V* vertices or not (accordingly, (32) is used in Step 2 of J-level reconstruction).

5 Multiresolution algorithms for boundary vertices

For open surfaces, we also need to consider the algorithms for boundary vertices. The treatment of boundary vertices for $\sqrt{3}$-refinement is slightly different from that with the dyadic refinement. For $\sqrt{3}$ subdivision, [28, 29] keep boundary vertices unchanged in odd-level subdivision iterations and apply 1-D ternary (3-dilation) subdivision schemes in even-level subdivision iterations. This idea was used in [47] to design decomposition/reconstruction algorithms for boundary vertices. Here as in [47], boundary vertices are unchanged in odd-level surface decompositions, while a 1-D ternary (3-dilation) frame decomposition algorithm constructed below is applied to boundary vertices in odd-level surface decompositions. More precisely, assume the boundary of an open surface is represented locally as in Fig. 10 (here we represent the case that all vertices are regular), where the boundary consists of vertices $\{h, W\}$, $\{V, U, W\}$ forms a 1-level $\sqrt{3}$-refinement coarse mesh (with dashed lines), and $\{U, W\}$ forms the 2-level $\sqrt{3}$-refinement (equivalent to the ternary or 1-to-9 split) coarse mesh (with solid lines). The operations of a 2-level multiresolution algorithm are carried out as in (38) and (39), where the arrow "$\rightarrow$" means "is replaced by"; $\{\widetilde{V}, \widetilde{U}, \widetilde{W}\}$ and $\{\widetilde{h}, \widetilde{f}, \widetilde{h}\}$ are the "approximation" and "detail" after 1-level $\sqrt{3}$ decomposition; $\{\widetilde{\widetilde{U}}, \widetilde{\widetilde{W}}\}$ (with $\widetilde{\widetilde{W}} = \widetilde{W}$) and $\{\widetilde{\widetilde{V}}, \widetilde{\widetilde{g}}\}$ are the "approximation" and "detail" after 2-level $\sqrt{3}$ decomposition.

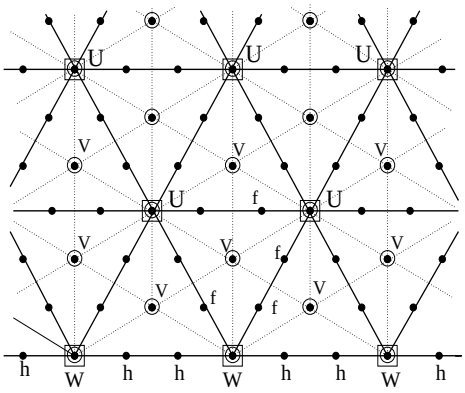

Figure 10: *1-level $\sqrt{3}$-refinement coarse mesh (consisting of dashed lines) with nodes $\bigcirc V, U, W$ and 2-level $\sqrt{3}$-refinement coarse mesh (consisting of thick solid lines) with nodes big squares U, W*

1-level decomposition:

$$\begin{array}{ll} \text{interior vertices:} & V, U \rightarrow \widetilde{V}, \widetilde{U}, \widetilde{g}; \quad f \rightarrow \widetilde{f}, \\ \text{boundary vertices:} & W \rightarrow \widetilde{W}, \widetilde{g}; \quad h \rightarrow \widetilde{h}. \end{array} \tag{38}$$

2-level decomposition:

$$\begin{array}{ll} \text{interior vertices:} & \widetilde{U} \rightarrow \widetilde{\widetilde{U}}, \widetilde{\widetilde{g}}; \quad \widetilde{V} \rightarrow \widetilde{\widetilde{V}}, \\ \text{boundary vertices:} & \text{no operation} \quad \widetilde{\widetilde{W}} = \widetilde{W}. \end{array} \tag{39}$$

For boundary vertices, a 1-D ternary (3-dilation) frame decomposition algorithm is operated in the 1-level decomposition, and no operation is carried in the 2-level decomposition. Observe that in the 2-level decomposition for interior vertices, $\widetilde{U}$ are considered as *type* V vertices and $\widetilde{V}$ are considered as *type* F vertices. The output $\widetilde{\widetilde{U}}$ associated with $\widetilde{U}$ (actually associated with U) form

the 2-level "approximation". For more than a two-level decomposition, we repeat the above procedure with boundary vertices operated on by a 1-D ternary frame decomposition algorithm in the odd-level decomposition and no decomposition operation in the even-level decomposition. The reconstruction is the reverse operation of the decomposition. For example, for 2-level reconstruction, there is no reconstruction operation to boundary vertices in the first level reconstruction (namely just set $\widetilde{W} = \widetilde{\widetilde{W}}$), and a 1-D ternary frame reconstruction algorithm is applied in the second level reconstruction (namely, to recover W, h from $\widetilde{W}, \widetilde{g}, \widetilde{h}$). Next, we design 1-D ternary frame multiresolution algorithms for boundary vertices.

The 1-D ternary frame decomposition algorithm and reconstruction algorithm are given in (40)-(45) and shown in Figs. 11 and 12 respectively. Here we still use v, f to denote the vertices on the boundary, and use the same letters for the parameters of 1-D frame algorithms.

1-D Ternary Frame Decomposition Algorithm for Boundary Vertices:

$$\text{Step 1. } v'' = \tfrac{1}{b}\big\{v - d(f_0 + f_1)\big\},\ g'' = v - n(f_0 + f_1) \tag{40}$$

$$\text{Step 2. } \widetilde{f} = f - av_0'' - a_1 v_1'' - rv_2'' - hg_0'' - h_1 g_1'' - sg_2'' \tag{41}$$

$$\text{Step 3. } \widetilde{v} = v'' - d_1(\widetilde{f}_0 + \widetilde{f}_1),\ \widetilde{g} = g'' - n_1(\widetilde{f}_0 + \widetilde{f}_1). \tag{42}$$

1-D Ternary Frame Reconstruction Algorithm for Boundary Vertices:

$$\text{Step 1. } v'' = \widetilde{v} + d_1(\widetilde{f}_0 + \widetilde{f}_1),\ g'' = \widetilde{f} + n_1(\widetilde{f}_0 + \widetilde{f}_1) \tag{43}$$

$$\text{Step 2. } f = \widetilde{f} + av_0'' + a_1 v_1'' + rv_2'' + hg_0'' + h_1 g_1'' + sg_2'' \tag{44}$$

$$\text{Step 3. } v = t\big\{bv'' + d(f_0 + f_1)\big\} + (1-t)\big\{v'' + n(f_0 + f_1)\big\}. \tag{45}$$

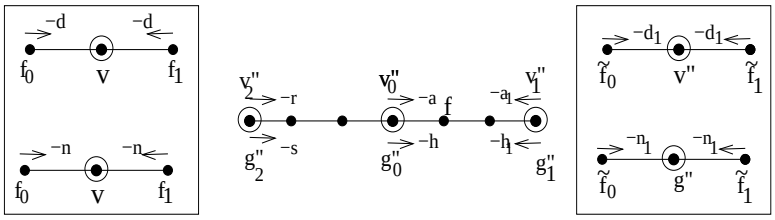

Figure 11: Left: *Decomposition Step 1;* Middle: *Decomposition Step 2;* Right: *Decomposition Step 3*

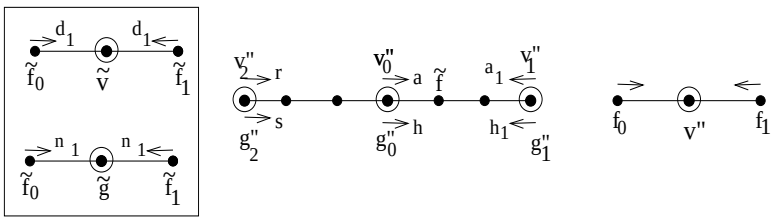

Figure 12: Left: *Reconstruction Step 1;* Middle: *Reconstruction Step 2;* Right: *Reconstruction Step 3*

We can find as in §3 and §4 the corresponding ternary (3-dilation) bi-frame analysis and synthesis filter banks, which are also denoted by $\{p(\omega), q^{(1)}(\omega), q^{(2)}(\omega), q^{(3)}(\omega)\}$ and $\{\widetilde{p}(\omega), \widetilde{q}^{(1)}(\omega),$

$\widetilde{q}^{(2)}(\omega), \widetilde{q}^{(3)}(\omega)\}$, $\omega \in \mathbb{R}$. The filter banks are provided in Appendix F. Then based on the smoothness and the vanishing moments of the framelets, we select suitable parameters. Here we choose

$$[b,\ d,\ n,\ a,\ a_1,\ r,\ h,\ h_1,\ s,\ d_1,\ n_1, t] = [3, -1, \tfrac{1}{2}, \tfrac{16}{27}, \tfrac{10}{27}, \tfrac{1}{27}, -\tfrac{4}{9}, \tfrac{1}{9}, 0, -\tfrac{5}{11}, \tfrac{3}{22}, \tfrac{1}{11}]. \tag{46}$$

The corresponding $\widetilde{\phi}(x)$ is the C^2 cubic spline supported on [-2, 2], $\phi(x) \in W^{0.30438}(\mathbb{R})$, $q^{(1)}$ has vanishing moment order 4, $q^{(2)}$, $q^{(3)}$ and $\widetilde{q}^{(1)}$ have vanishing moment order 2, and $\widetilde{q}^{(2)}, \widetilde{q}^{(3)}$ have vanishing moment order 1. The resulting lowpass synthesis filter $\widetilde{p}(\omega)$ is the 1-D ternary scheme used in [28]. The above 1-D ternary algorithm can be used for boundary vertices when Kobbelt's scheme-based frame algorithm in §3 is applied for interior vertices.

For interpolatory scheme-based algorithms in §4, we need to design 1-D interpolatory ternary scheme-based multiresolution algorithms for boundary vertices. The 1-D algorithm is still given by (40)-(45), but we choose different parameters such that the reconstruction algorithm is reduced to an interpolatory ternary subdivision scheme when the "detail" is set to zero. The parameters are chosen to be

$$[b,\ d,\ n,\ a,\ a_1,\ r,\ h,\ h_1,\ s,\ w,\ n_1,\ t] = [3, -1, \tfrac{1}{2}, \tfrac{4}{5}, \tfrac{4}{15}, -\tfrac{1}{15}, -\tfrac{4}{9}, \tfrac{1}{9}, 0, -\tfrac{1}{3}, \tfrac{5}{18}, \tfrac{1}{3}]. \tag{47}$$

The resulting $\widetilde{p}$ and p have sum rule order 2 with $\widetilde{\phi} \in W^{1.73248}(\mathbb{R})$ and $\phi \in W^{0.44662}(\mathbb{R})$, $q^{(1)}$ has vanishing moment order 4, $q^{(2)}$, $q^{(3)}$ and $\widetilde{q}^{(1)}$ have vanishing moment order 2, and $\widetilde{q}^{(2)}, \widetilde{q}^{(3)}$ have vanishing moment order 1.

When the "detail" is set to zero, then the above reconstruction algorithm is an interpolatory 3-point ternary C^1 scheme:

$$f = \frac{4}{5}\widetilde{v}_0 + \frac{4}{15}\widetilde{v}_1 - \frac{1}{15}\widetilde{v}_2, \quad v = \widetilde{v},$$

where $\widetilde{v}_j$ are the vertices in the coarse polygon. An interpolatory 4-point C^2 ternary subdivision scheme is provided in [17]. Here we construct an interpolatory scheme with a smaller template.

When JO's scheme-based algorithm in §4 is applied to a *type F* vertex f near the boundary, we will use (29)(ii) to define $\widetilde{f}$ (accordingly, (32)(ii) is used in Step 2 to reconstruct f).

6 One experimental result and future work

Our highly symmetric frame algorithms can be applied immediately for some applications such as surface sparse representation, noise removal, compression, progressive transmission, etc. To this regard, it is required that the input surfaces have a semi-regular structure (a subdivision connectivity). One could use MAPS or other methods (see for example [30, 14]) to get a surface that has a subdivision connectivity and closely approximates (with guaranteed errors) the original (non semi-regular) surface. Here we use *TriReme* developed by Guskov.

Here we show the result of applying the 3-step $\sqrt{3}$ algorithm of Kobbelt-$\mathrm{F}_{2,2}$ (with parameters given by (24)) with two levels of decomposition of a semi-regular surface mesh to noise removal. The original, noisy and denoised images of the fandisk and threehole surfaces are shown in Figs. 13 and 14. The noisy surfaces were produced by adding Gaussian noise normal to the original semi-regular surface at each vertices. The added Gaussian noise is modeled as a measurement noise with a mean of zero and a standard deviation equal to a known measurement error. The measurement error was calculated as a percentage of the diagonal length of the box that contains the surface. For our application we chose a measurement error of 0.06% for the fandisk and threehole semi-regular surface meshes. For each surface, multiplying the number of vertices times

the square of the measurement error gives us the best estimate of the total noise energy added to the surface. Note that since our algorithm uses a frame decomposition and reconstruction the amount of noise energy that needs to be removed from the highpass coefficients will be more than what was added to the surface mesh. Our 3-step $\sqrt{3}$ algorithm with two levels of decomposition and reconstruction requires that an amount of noise energy to be removed is equal to approximately 150% of the total noise energy added to the semi-regular surface mesh.

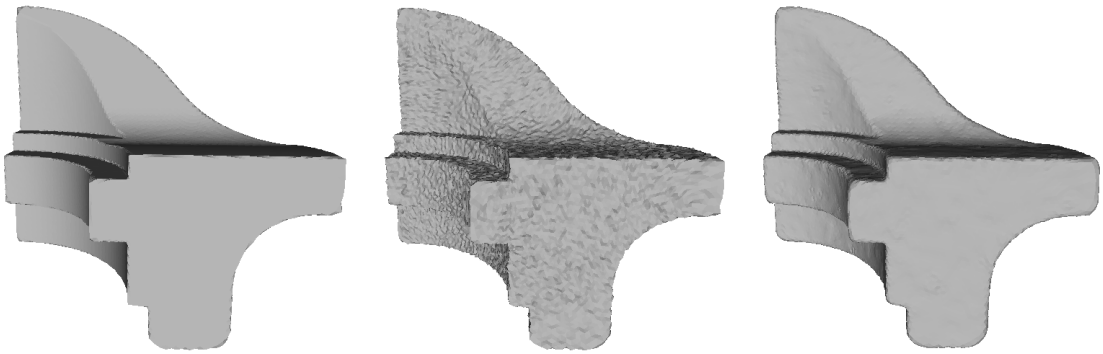

Figure 13: Left*: Original fandisk mesh;* Middle: *Noisy mesh;* Right: *Denoised mesh*

Our procedure of denoising consists of two parts. First we decompose the surface and set to zero those lowest valued highpass coefficients whose energies sum to 150% of the total added noise energy. Second, we use a de-speckle routine to remove the remaining speckle type noise (refer to [49]). For the first part of the procedure this means we need to remove specific percentages of the total noise energy from the high pass coefficients at each level of decomposition. We were able to determine these percentages by calculating the highpass coefficient energies for the original noiseless and noisy surface meshes. We found that these percentages were very closely the same for different surfaces and with different measurement errors.

Upon completion of the first part of our procedure the reconstructed surface exhibits a noise with a highly speckled nature. After the second part of our procedure, application of a despeckle routine, the only remaining noise is faint one-ring neighborhood plateau artifacts.

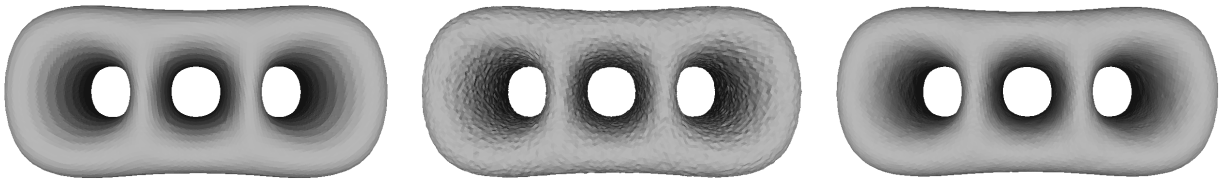

Figure 14: Left*: Original threehole mesh;* Middle: *Noisy mesh;* Right: *Denoised mesh*

In this paper we mainly show how to construct framelets with stencils for the implementation by the idea of lifting scheme and how the coefficients can be determined from standard requirements such as vanishing moments, smoothness and sum rule orders. Here we just show the surface denoising result with one bi-frame algorithm. In our future work, we will consider the selection of the coefficients based on the condition number of the frame transform. We will incorporate more advanced image denoising techniques with our bi-frame algorithms for surface denoising. We will also explore other surface multiresolution applications and study the issue that which algorithm among different algorithms Kobbelt-$F_{1,2}$, Kobbelt-$F_{2,2}$ and JO-$F_{4,2}$ should be used to a particular

Bi-frame filters	Approx. or interpolatory	Symmetry	Sum rule order	Smoothness	Vanishing moment order
Kobbelt-$F_{1,2}$	Approx. $\widetilde{p}$ is Kobbelt's filter in [28]	$p, \widetilde{p}, q^{(1)}, \widetilde{q}^{(1)}$: 6-fold line sym. $q^{(2)}, q^{(3)}, \widetilde{q}^{(2)}\widetilde{q}^{(3)}$: 3-fold line sym.	p: order 2 $\widetilde{p}$: order 3	$\phi \in W^{1.6571}$ $\widetilde{\phi} \in W^{2.9360}$	$q^{(1)}$: order 2 $q^{(2)}$, $q^{(3)}$: order 3 $\widetilde{q}^{(1)}$, $\widetilde{q}^{(2)}$, $\widetilde{q}^{(3)}$: order 0
Kobbelt-$F_{2,2}$	Approx. $\widetilde{p}$ is Kobbelt's filter in [28]	$p, \widetilde{p}, q^{(1)}, \widetilde{q}^{(1)}$: 6-fold line sym. $q^{(2)}, q^{(3)}, \widetilde{q}^{(2)}\widetilde{q}^{(3)}$: 3-fold line sym.	p: order 3 $\widetilde{p}$: order 3	$\phi \in W^{1.8001}$ $\widetilde{\phi} \in W^{2.9360}$	$q^{(1)}$: order 4 $q^{(2)}$, $q^{(3)}$: order 2 $\widetilde{q}^{(1)}$, $\widetilde{q}^{(2)}$, $\widetilde{q}^{(3)}$: order 2
JO-$F_{4,2}$	Interpolatory $\widetilde{p}$ is from [25]	$p, \widetilde{p}, q^{(1)}, \widetilde{q}^{(1)}$: 6-fold line sym. $q^{(2)}, q^{(3)}, \widetilde{q}^{(2)}\widetilde{q}^{(3)}$: 3-fold line sym.	p: order 2 $\widetilde{p}$: order 3	$\phi \in W^{2.6290}$ $\widetilde{\phi} \in W^{1.8959}$	$q^{(1)}$: order 3 $q^{(2)}$, $q^{(3)}$: order 2 $\widetilde{q}^{(1)}$, $\widetilde{q}^{(2)}$, $\widetilde{q}^{(3)}$: order 2

Table 1: Properties of Kobbelt-$F_{1,2}$, Kobbelt-$F_{2,2}$ and JO-$F_{4,2}$ bi-frame filter banks

application. For the convenience to the reader and for our future study, we summarize in Table 1 the properties of the bi-frame filters of these algorithms. Note that Kobbelt-$F_{1,2}$ and Kobbelt-$F_{2,2}$ have the same $\widetilde{\phi}$ for reconstruction with $\widetilde{\phi}$ having a high smooth order. Compared with Kobbelt-$F_{2,2}$, Kobbelt-$F_{1,2}$ has simpler algorithms and smaller templates, but its corresponding $\widetilde{q}^{(1)}$, $\widetilde{q}^{(2)}$, $\widetilde{q}^{(3)}$ for reconstruction have no vanishing moments. JO-$F_{4,2}$ is interpolatory, but it has bigger templates and a lower smoothness order $\widetilde{\phi}$ for reconstruction.

Appendices

In the following appendices, $x = e^{-i\omega_1},\ y = e^{-i\omega_2}$.

Appendix A

With $b = b(6), d = d(6), n = n(6), w = w(6), n_1 = n_1(6)$, the $\sqrt{3}$ frame filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ corresponding to the algorithms (12)-(17) with $k = 6$ are

$$\begin{cases} \left[p(\boldsymbol{\omega}),\ q^{(1)}(\boldsymbol{\omega}),\ q^{(2)}(\boldsymbol{\omega}),\ q^{(3)}(\boldsymbol{\omega})\right]^T = B_2(A^T\boldsymbol{\omega})B_1(A^T\boldsymbol{\omega})B_0(A^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \\ \left[\widetilde{p}(\boldsymbol{\omega}),\ \widetilde{q}^{(1)}(\boldsymbol{\omega}),\ \widetilde{q}^{(2)}(\boldsymbol{\omega}),\ \widetilde{q}^{(3)}(\boldsymbol{\omega})\right]^T = \frac{1}{3}\widetilde{B}_2(A^T\boldsymbol{\omega})\widetilde{B}_1(A^T\boldsymbol{\omega})\widetilde{B}_0(A^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \end{cases} \tag{48}$$

where $I_0(\boldsymbol{\omega})$ is define by (11), and

$$B_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & -wY_2 & -wY_1 \\ 0 & 1 & -n_1Y_2 & -n_1Y_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{49}$$

$$B_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -aY_1 & -hY_1 & 1 & 0 \\ -aY_2 & -hY_2 & 0 & 1 \end{bmatrix}, \tag{50}$$

$$B_0(\boldsymbol{\omega}) = \begin{bmatrix} \frac{1}{b} & -\frac{d}{b}Y_2 & -\frac{d}{b}Y_1 \\ 1 & -nY_2 & -nY_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{51}$$

$$\widetilde{B}_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ wY_1 & n_1Y_1 & 1 & 0 \\ wY_2 & n_1Y_2 & 0 & 1 \end{bmatrix}, \tag{52}$$

$$\widetilde{B}_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & aY_2 & aY_1 \\ 0 & 1 & hY_2 & hY_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{53}$$

$$\widetilde{B}_0(\boldsymbol{\omega}) = \begin{bmatrix} tb & 0 & 0 \\ 1-t & 0 & 0 \\ (td+(1-t)n)Y_1 & 1 & 0 \\ (td+(1-t)n)Y_3 & 0 & 1 \end{bmatrix}, \tag{54}$$

where

$$Y_1 = 1 + x + 1/y,\ Y_2 = 1 + 1/x + y. \tag{55}$$

Appendix B

The filters of Kobbelt-$\mathrm{F}_{2,2}$ with $b = 2$ and other parameters given by (20) are

$$\begin{aligned} p(\boldsymbol{\omega}) = \frac{1}{81}\Big\{ & 36 + 13(x + y + xy + \frac{1}{x} + \frac{1}{y} + \frac{1}{xy}) - \frac{3}{2}(x^2y + xy^2 + \frac{1}{x^2y} + \frac{1}{xy^2} + \frac{x}{y} + \frac{y}{x}) \\ & -2(x^2 + y^2 + x^2y^2 + \frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{x^2y^2}) \\ & -(x^3y^2 + x^3y + x^2y^3 + xy^3 + \frac{1}{x^3y^2} + \frac{1}{x^3y} + \frac{1}{x^2y^3} + \frac{1}{xy^3} + \frac{x^2}{y} + \frac{x}{y^2} + \frac{y^2}{x} + \frac{y}{x^2})\Big\}, \\ q^{(1)}(\boldsymbol{\omega}) = \frac{1}{243}\Big\{ & 252 - 53(x + y + xy + \frac{1}{x} + \frac{1}{y} + \frac{1}{xy}) + 3(x^2y + xy^2 + \frac{1}{x^2y} + \frac{1}{xy^2} + \frac{x}{y} + \frac{y}{x}) \\ & +4(x^2 + y^2 + x^2y^2 + \frac{1}{x^2} + \frac{1}{y^2} + \frac{1}{x^2y^2}) \\ & +2(x^3y^2 + x^3y + x^2y^3 + xy^3 + \frac{1}{x^3y^2} + \frac{1}{x^3y} + \frac{1}{x^2y^3} + \frac{1}{xy^3} + \frac{x^2}{y} + \frac{x}{y^2} + \frac{y^2}{x} + \frac{y}{x^2})\Big\}, \end{aligned}$$

$$q^{(2)}(\boldsymbol{\omega}) = \frac{1}{45}\Big\{39x - 3(1+x^2y+\frac{x}{y}) - 4(\frac{1}{y}+xy+x^2)$$
$$-2(\frac{1}{x}+\frac{x}{y^2}+x^3y^2+y+\frac{1}{xy}+\frac{1}{y^2}+\frac{x^2}{y}+x^2y^2+x^3y)\Big\},$$
$$\widetilde{q}^{(1)}(\boldsymbol{\omega}) = \frac{1}{30}\Big\{7-(x+y+xy+\frac{1}{x}+\frac{1}{y}+\frac{1}{xy}) - \frac{1}{6}(x^2y+xy^2+\frac{1}{x^2y}+\frac{1}{xy^2}+\frac{x}{y}+\frac{y}{x})\Big\},$$
$$\widetilde{q}^{(2)}(\boldsymbol{\omega}) = \frac{1}{54}\Big\{15x+\frac{2}{3}(1+x^2y+\frac{x}{y}) - 2(\frac{1}{y}+xy+x^2) - (\frac{1}{x}+\frac{x}{y^2}+x^3y^2+y+\frac{1}{xy}+\frac{1}{y^2}+\frac{x^2}{y}+x^2y^2+x^3y)$$
$$-\frac{1}{3}(x^3+\frac{1}{xy^2}+xy^2) - \frac{1}{6}(x^4y^2+x^3y^3+\frac{x^2}{y^2}+\frac{y}{x}+\frac{1}{x^2y}+\frac{1}{y^3})\Big\},$$

and $\widetilde{p}(\boldsymbol{\omega})$ is given by (19), $q^{(3)}(\boldsymbol{\omega}) = q^{(2)}(-\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega}) = \widetilde{q}^{(2)}(-\boldsymbol{\omega})$.

Appendix C

The highpass filters of Kobbelt-$\mathrm{F}_{1,2}$ with lowpass filters $\widetilde{p}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ given by (19) and (26) respectively are

$$q^{(1)}(\boldsymbol{\omega}) = \frac{1}{6}\Big\{6-(x+y+xy+\frac{1}{x}+\frac{1}{y}+\frac{1}{xy})\Big\},$$
$$q^{(2)}(\boldsymbol{\omega}) = \frac{1}{18}\Big\{21x - 12(1+x^2y+\frac{x}{y}) + 2(\frac{1}{y}+xy+x^2)$$
$$+(\frac{1}{x}+\frac{x}{y^2}+x^3y^2+y+\frac{1}{xy}+\frac{1}{y^2}+\frac{x^2}{y}+x^2y^2+x^3y)\Big\},$$
$$\widetilde{q}^{(1)}(\boldsymbol{\omega}) = \frac{5}{3}\widetilde{p}(\omega), \quad \widetilde{q}^{(2)}(\boldsymbol{\omega}) = \frac{1}{36}\Big\{12x+(1+x^2y+\frac{x}{y})\Big\},$$

and $q^{(3)}(\boldsymbol{\omega}) = q^{(2)}(-\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega}) = \widetilde{q}^{(2)}(-\boldsymbol{\omega})$.

Appendix D

The $\sqrt{3}$ frame filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ corresponding to the algorithms (28)-(33) with $k=6$ are given by (48) with $B_2(\boldsymbol{\omega})$, $B_0(\boldsymbol{\omega})$, $\widetilde{B}_2(\boldsymbol{\omega})$, $\widetilde{B}_0(\boldsymbol{\omega})$ defined by (49), (51), (52), (54) respectively, and $B_1(\boldsymbol{\omega})$ and $\widetilde{B}_1(\boldsymbol{\omega})$ given by

$$B_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -aY_1 - rY_3 & -hY_1 - sY_3 & 1 & 0 \\ -aY_2 - rY_4 & -hY_2 - sY_4 & 0 & 1 \end{bmatrix},$$
$$\widetilde{B}_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & aY_2 + rY_4 & aY_1 + rY_3 \\ 0 & 1 & hY_2 + sY_4 & hY_1 + sY_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where Y_1, Y_2 are defined by (55) and

$$Y_3 = xy + 1/(xy) + x/y,\ Y_4 = xy + 1/(xy) + y/x.$$

Appendix E. Eigenvalue analysis of subdivision matrix S in §4

Denote $\alpha = \alpha(k), \beta = \beta(k), \gamma = \gamma(k)$. Using suitable labels of the vertices (refer to [50, 8]), one can obtain that the subdivision matrix $\mathbf{S}$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \alpha & \beta & \beta & \gamma & 0 & \cdots & 0 & \gamma \\ \alpha & \gamma & \beta & \beta & \gamma & \cdots & 0 & 0 \\ \alpha & 0 & \gamma & \beta & \beta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \alpha & \beta & \gamma & 0 & 0 & \cdots & \gamma & \beta \end{bmatrix}.$$

Applying the discrete Fourier transform to the $k \times k$ sub-matrix of $\mathbf{S}$ (resulted from the removal of the first row and column of $\mathbf{S}$), one can obtain that the eigenvalues of $\mathbf{S}$ are

$$1,\ e^{-\frac{m\pi}{k}i} h(\frac{2\pi m}{k}), \quad m = 0, 1, 2, \cdots, k-1,$$

where

$$h(t) = 2\beta \cos\frac{t}{2} + 2\gamma \cos\frac{3t}{2}.$$

Next we choose β, α such that $|h(t)|$ obtains the maxima on $[0, 2\pi]$ at $t = \frac{2\pi}{k}, \frac{2\pi(k-1)}{k}$. Refer to [34] for similar discussion. Observe that $h(2\pi - t) = -h(t)$. Thus let us focus $h(t)$ on $[0, \pi]$. One can obtain

$$h'(t) = -\beta \sin\frac{t}{2} - 3\gamma \sin\frac{3t}{2} = -\sin\frac{t}{2}(\beta + 3\gamma + 6\gamma\cos t).$$

Thus if we choose β, γ such that $\beta = -3\gamma - 6\gamma\cos\frac{2\pi}{k}$, then $h'(\frac{2\pi}{k}) = 0$. Furthermore, if $\gamma < 0$, then $h(t)$ is strictly increasing and decreasing on $[0, \frac{2\pi}{k}]$ and $[\frac{2\pi}{k}, \pi]$ respectively. In this paper we choose $\gamma = -\frac{1}{24}$. The other parameter α is chosen as $\alpha = 1 - 2\beta - 2\gamma$ so that $[1, 1, \cdots, 1]^T$ is a right 1-eigenvector of $\mathbf{S}$. With such choices of α, β, γ, the leading eigenvalues $\lambda_0,\ \lambda_1,\ \lambda_2,\ \lambda_3, \cdots$ of $\mathbf{S}$ satisfy $\lambda_0 = 1$, $1 > |\lambda_1| = |\lambda_2|$ $(= h(\frac{2\pi}{k}))$, and $|\lambda_3| < |\lambda_1|$.

Appendix F

The 1-D ternary frame filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ corresponding to multiresolution algorithm (40)-(45) are given by

$$\begin{aligned} \left[p(\omega),\ q^{(1)}(\omega),\ q^{(2)}(\omega),\ q^{(3)}(\omega)\right]^T &= D_2(3\omega)D_1(3\omega)D_0(3\omega)[1,\ z,\ \tfrac{1}{z}]^T, \\ \left[\widetilde{p}(\omega),\ \widetilde{q}^{(1)}(\omega),\ \widetilde{q}^{(2)}(\omega), \widetilde{q}^{(3)}(\omega)\right]^T &= \tfrac{1}{3}\widetilde{D}_2(3\omega)\widetilde{D}_1(3\omega)\widetilde{D}_0(3\omega)[1,\ z,\ \tfrac{1}{z}]^T, \end{aligned}$$

where $z = e^{-i\omega}$, and

$$D_2(\omega) = \begin{bmatrix} 1 & 0 & -d_1 & -d_1 \\ 0 & 1 & -n_1 & -n_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ D_1(\omega) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a - a_1 z - \frac{r}{z} & -h - h_1 z - \frac{s}{z} & 1 & 0 \\ -a - \frac{a_1}{z} - rz & -h - \frac{h_1}{z} - sz & 0 & 1 \end{bmatrix},$$

$$D_0(\omega) = \begin{bmatrix} \frac{1}{b} & -\frac{d}{b} & -\frac{d}{b} \\ 1 & -n & -n \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \widetilde{D}_2(\omega) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ d_1 & n_1 & 1 & 0 \\ d_1 & n_1 & 0 & 1 \end{bmatrix},$$

$$\widetilde{D}_1(\omega) = \begin{bmatrix} 1 & 0 & a + \frac{a_1}{z} + rz & a + a_1 z + \frac{r}{z} \\ 0 & 1 & h + \frac{h_1}{z} + sz & h + h_1 z + \frac{s}{z} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ \widetilde{D}_0(\omega) = \begin{bmatrix} tb & 0 & 0 \\ 1 - t & 0 & 0 \\ td + (1-t)n & 1 & 0 \\ td + (1-t)n & 0 & 1 \end{bmatrix}.$$

Acknowledgments. The authors would like to thank Dr. Igor Guskov for redeveloping his surface remeshing software, *TriReme*, to include the $\sqrt{3}$-refinement surface remeshing. The most recent version allows us to apply our algorithms to semi-regular meshes with $\sqrt{3}$-subdivision connectivity. The authors also thank two anonymous referees for their valuable comments.

References

[1] M. Bertram, "Biorthogonal Loop-subdivision wavelets", *Computing*, 72 (2004), 29–39.

[2] M. Bertram, M.A. Duchaineau, B. Hamann, and K.I. Joy, "Generalized B-spline subdivision-surface wavelets for geometry compression", *IEEE Trans. Visualization and Computer Graphics*, 10 (2004), 326–338.

[3] P.J. Burt, "Tree and pyramid structures for coding hexagonally sampled binary images", *Computer Graphics and Image Proc.*, 14 (1980), 271–280.

[4] L. Condat, B. Forster-Heinlein, and D. Van De Ville, "A new family of rotation-covariant wavelets on the hexagonal lattice", In: *Proc. of the SPIE Optics and Photonics 2007 Conference on Mathematical Methods: Wavelet XII*, San Diego CA, USA, August, 2007, vol. 6701, pp. 67010B-1/67010B-9.

[5] O. Christensen, *An Introduction to Frames and Riesz Bases*, Birkhäuser, Boston, 2002.

[6] C.K. Chui and Q.T. Jiang, "Surface subdivision schemes generated by refinable bivariate spline function vectors", *Appl. Comput. Harmonic Anal.*, 15 (2003), 147–162.

[7] C.K. Chui and Q.T. Jiang, "Matrix-valued symmetric templates for interpolatory surface subdivisions, I. Regular vertices", *Appl. Comput. Harmonic Anal.*, 19 (2005), 303–339.

[8] C.K. Chui and Q.T. Jiang, "From extension of Loop's approximation scheme to interpolatory subdivisions", *Comput. Aided Geom. Design*, 25 (2008), 96–115.

[9] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 61, SIAM, Philadelphia, PA, 1992.

[10] I. Daubechies, B. Han, A. Ron, and Z.W. Shen, "Framelets: MRA-based construction of wavelet frames", *Appl. Comput. Harmon. Anal.*, 14 (2003), 1–46.

[11] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes", In: *Computer Graphics (SIGGRAPH 95 Proceedings)*, pp. 173–182, 1995.

[12] M. Ehler, "On multivariate compactly supported bi-frames", *J. Fourier Anal. Appl.*, 13 (2007), 511–532.

[13] M.J.E. Golay, "Hexagonal parallel pattern transformations", *IEEE Trans. Computers*, 18 (1969), 733–740.

[14] I. Guskov, "Manifold-based approach to semi-regular remeshing", *Graphical Models*, 69 (2007), 1–18.

[15] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder, "Normal meshes", In: *Proceedings of SIGGRAPH 00*, pp. 95–102, 2000.

[16] B. Han, T. Yu, and B. Piper, "Multivariate refinable Hermite interpolants", *Math. of Computation*, 73 (2004), 1913–1935.

[17] M.F. Hassan, I.P. Ivrissimitzis, and N.A. Dodgson, "An interpolatory 4-point C^2 ternary stationary subdivision scheme", *Computer Aided Geom. Design*, 19 (2002), 1–18.

[18] C. Heil and D. Walnut, "Continuous and discrete wavelet transforms", *SIAM Rev.*, 31 (1989), 628–666.

[19] R.Q. Jia, "Approximation properties of multivariate wavelets", *Math. Comp.*, 67 (1998), 647–665.

[20] R.Q. Jia and Q.T. Jiang, "Spectral analysis of transition operators and its applications to smoothness analysis of wavelets", *SIAM J. Matrix Anal. Appl.*, 24 (2003), 1071–1109.

[21] R.Q. Jia and S.R. Zhang, "Spectral properties of the transition operator associated to a multivariate refinement equation", *Linear Algebra Appl.*, 292 (1999), 155–178.

[22] Q.T. Jiang, "Orthogonal and biorthogonal $\sqrt{3}$-refinement wavelets for hexagonal data processing", *IEEE Trans. Signal Proc.*, 57 (2009), 14304–4313.

[23] Q.T. Jiang, "Biorthogonal wavelets with 4-fold axial symmetry for quadrilateral surface multiresolution processing", *Advances in Comput. Math.*, 34 (2011), 127–165.

[24] Q.T. Jiang, "Biorthogonal wavelets with 6-fold axial symmetry for hexagonal data and triangle surface multiresolution processing", Int'l J. Wavelets, Multiresolution and Info.. Proc., 9 (2011), 773–812.

[25] Q.T. Jiang and P. Oswald, "Triangular $\sqrt{3}$-subdivision schemes: The regular case", *J. Comput. Appl. Math.*, 156 (2003), 47–75.

[26] Q.T. Jiang, P. Oswald, and S.D. Riemenschneider, "$\sqrt{3}$-subdivision schemes: Maximal sum rules orders", *Constr. Approx.*, 19 (2003), 437–463.

[27] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression", In: *Proceedings of SIGGRAPH 00*, pp. 271–278, 2000.

[28] L. Kobbelt, "$\sqrt{3}$-subdivision", In: *SIGGRAPH Computer Graphics Proceedings*, pp. 103–112, 2000.

[29] U. Labsik and G. Greiner, "Interpolatory $\sqrt{3}$-subdivision", *Computer Graphics Forum*, 19 (2000), 131–138.

[30] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution adaptive parameterization of surfaces", In: *Proceedings of SIGGRAPH 98*, pp. 95–104, 1998.

[31] J.M. Lounsbery, *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*, Ph.D. Dissertation, University of Washington, Department of Mathematics, 1994.

[32] J.M. Lounsbery, T.D. Derose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type", *ACM Trans. Graphics*, 16 (1997), 34–73.

[33] J. Maes and A. Bultheel, "Stability analysis of biorthogonal multiwavelets whose duals are not in L2 and its application to local semiorthogonal lifting", *Appl. Numer. Math.*, 58 (2008), 1186–1211.

[34] P. Oswald and P. Schröder, "Composite primal/dual $\sqrt{3}$-subdivision schemes", *Comput. Aided Geom. Design*, 20 (2003), 135–164.

[35] E. Praun, W. Sweldens, and P. Schröder, "Consistent mesh parameterizations", In: *Proceedings of SIGGRAPH 01*, pp. 179–184, 2001.

[36] "Pyxis Innovation Inc. Documents", `www.pyxisinnovation.com` .

[37] U. Reif, "A unified approach to subdivision algorithms near extraordinary vertices", *Comput. Aided Geom. Design*, 21 (1995), 153–174.

[38] A. Ron and Z.W. Shen, "Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operators", *J. Funct. Anal.*, 148 (1997), 408–447.

[39] A. Ron and Z.W. Shen, "Affine systems in $L_2(\mathbb{R}^d)$ II: Dual systems", *J. Fourier Anal. Appl.*, 3 (1997), 617–637.

[40] K. Sahr, D. White, and A.J. Kimerling, "Geodesic discrete global grid systems", *Cartography and Geographic Information Science*, 30 (2003), 121–134.

[41] F.F. Samavati, N. Mahdavi-Amiri, and R.H. Bartels, "Multiresolution representation of surface with arbitrary topology by reversing Doo subdivision", *Computer Graphic Forum*, 21 (2002), 121–136.

[42] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere", In: *Proceedings of SIGGRAPH 95*, pp. 161–172, 1995.

[43] P. Schröder and D. Zorin, *Subdivision for Modeling and Animation*, *SIGGRAPH Course Notes*, 1999.

[44] E. Stollnitz, T. DeRose, and H. Salesin, *Wavelets for Computer Graphics*, Morgan Kaufmann Publishers, San Francisco, 1996.

[45] S. Valette and R. Prost, "Wavelet-based progressive compression scheme for triangle meshes: Wavemesh", *IEEE Trans. Visualization and Computer Graphics*, 10 (2004), 123–129.

[46] H.W. Wang, K.H. Qin, and K. Tang, "Efficient wavelet construction with Catmull-Clark subdivision", *The Visual Computer*, 22 (2006), 874–884.

[47] H.W. Wang, K.H. Qin, and H.Q. Sun, "$\sqrt{3}$-subdivision-based biorthogonal wavelets", *IEEE Trans. Visualization and Computer Graphics*, 13 (2007), 914–925.

[48] J. Warren and H. Weimer, *Subdivision Methods For Geometric Gesign: A Constructive Approach*, Morgan Kaufmann Publ., San Francisco, 2002.

[49] Y.L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal", *IEEE Trans. on Image Processing*, 9 (2000), 1723–1730.

[50] D. Zorin, "A method for analysis of C^1-continuity of subdivision surfaces", *SIAM J. Numer. Anal.*, 37 (2000), 1677–1708.