# Highly Symmetric Bi-frames for Triangle Surface Multiresolution Processing

Qingtang Jiang and Dale K. Pounds [*]

## Abstract

In this paper we investigate the construction of dyadic affine (wavelet) bi-frames for triangular-mesh surface multiresolution processing. We introduce 6-fold symmetric bi-frames with 4 framelets (frame generators). 6-fold symmetric bi-frames yield frame decomposition and reconstruction algorithms (for regular vertices) with high symmetry, which is required for the design of the corresponding frame multiresolution algorithms for extraordinary vertices on the triangular mesh. Compared with biorthogonal wavelets, the constructed bi-frames have better smoothness and smaller supports. In addition, we also provide frame multiresolution algorithms for extraordinary vertices. All the frame algorithms considered in this paper are given by templates (stencils) so that they are implementable. Furthermore, we present some preliminary experimental results on surface processing with frame algorithms constructed in this paper.

*AMS 2000 Math Subject Classification:* 42C40, 65T60, 68U07, 65D17

*Keywords:* Lifting scheme, 6-fold symmetric frame filter banks, symmetric wavelet bi-frames, frame multiresolution algorithm templates, Loop's scheme-based bi-frames, the butterfly scheme-based bi-frames, surface multiresolution decomposition/reconstruction

## 1 Introduction

This paper studies the biorthogonal affine (or wavelet) frames for triangular mesh-based surface multiresolution (multiscale) processing. When a filter bank is used for mesh-based surface multiresolution processing, several issues need to be considered. Firstly, the filters including both lowpass and highpass filters must have high symmetry. This is due to the fact that unlike an image, a set of 2-D data, a surface to be processed is an object in 3-D space and the algorithms for surface processing need to have a high symmetry. Secondly, the filters should have few nonzero coefficients. If the reconstruction (synthesis) filters have too many nonzero coefficients, undesired artifacts may be present in the reconstructed surfaces. Last but not least, the algorithms including those with highpass filters should be given by templates (stencils) so that the algorithms can be easily implemented.

Linear spline and butterfly scheme related semi-orthogonal wavelets for surface multiresolution processing have been studied in [38, 39], Doo's subdivision-scheme based wavelets for quadrilateral surfaces were considered in [45], and spherical wavelets were introduced in [46]. Recently

[*] The authors are with the Department of Mathematics and Computer Science, University of Missouri–St. Louis, St. Louis, MO 63121, USA.

with the idea of the lifting scheme [15, 48], biorthogonal wavelets with high symmetry for surface multiresolution processing have been constructed in [2, 3, 31, 32, 49, 50]. If the biorthogonal wavelets have certain smoothness, they will have big supports. In other words, the corresponding multiscale algorithms have large templates, and this is not desirable for surface processing. Loop's scheme-based biorthogonal wavelets have been considered in [34] with the biorthogonal dual wavelets constructed in [26]. However the corresponding highpass filters do not have desirable symmetry for surface processing with extraordinary vertices. This undesirable property will cause the problem to design the associated algorithms for extraordinary vertices (see [34] for detailed discussion).

Compared with (bi)orthogonal wavelet systems, the elements in a frame system may be linearly dependent; namely, frames can be redundant. The property of redundancy not only results in framelets with high symmetry and smaller supports than wavelets, but also provides high sparsity of frame transform coefficients. Such sparsity is a key property for many applications. In addition, frames work better in a noisy environment [13]. Thus, frames have been successfully used in noise removal [47], image recovery [9, 10], image inpainting/restoration [5, 6, 7], and signal classification [13], and medical image analysis [40].

The construction of multivariate wavelet frames has been studied in some papers, see e.g., [24, 29, 35, 43, 44]. Few of them address the symmetry of the framelets. When frames are applied for surface multiresolution processing, it is required that framelets have high symmetry and small supports. To the authors$'$ best knowledge, [11, 12, 20, 21] may be the only articles available in the literature which yield framelets with the desirable symmetry. The lowpass filter $p(\boldsymbol{\omega})$ for the framelets constructed in [20, 21] is the product of an interpolation filter and a trigonometric polynomial, which results in big supports of the corresponding scaling functions and framelets. While the method in [11, 12] does yield (tight) framelets with desirable symmetry and small supports, in general it leads to many numbers of framelets.

In this paper, we study bi-frames with 4 framelets for triangular (mesh-based) surface multiresolution processing. Recall that a biorthogonal system needs 3 analysis or synthesis wavelets. Thus, compared with biorthogonal systems, our frames have just one more generator. Our construction is based on symmetric templates of frame algorithms which lead to framelets with desirable symmetry. In addition, we will start with symmetric templates of small size (as small as possible) with the templates given by some parameters. Then we select the parameters such that the resulting framelets have optimal smoothness and/or vanishing moments. If the templates with a particular size cannot yield framelets with desirable smoothness and/or vanishing moments, then we consider templates with a bigger size. Thus the constructed symmetric bi-frames are optimal in the sense that with templates of particular (small) sizes, they achieve the highest smoothness and/or vanishing moment orders.

The rest of paper is outlined as follows. We give some background on wavelet frames and notations in §2. In §3, we show that frame multiresolution analysis and synthesis algorithms can be represented as templates by associating the outputs appropriately with the nodes of the original regular triangular mesh with which the input data is associated (sampled). Based on symmetric templates, we construct symmetric bi-frames with a 2-step algorithm and a 3-step algorithm in §4 and §5 respectively. In §6, we consider the butterfly scheme-based bi-frames, and in §7 we construct bi-frames based on a 4-step algorithm. In the last section, §8, we provide frame multiresolution algorithms for extraordinary vertices and present some preliminary experimental results with the framelets used for surface multiresolution decomposition and reconstruction.

## 2 Background and notations

A system $G \subset L^2(\mathbb{R}^2)$ is called a frame of $L^2(\mathbb{R}^2)$ if there are two positive constants $B$ and $C$ such that

$$B\|f\|_2^2 \le \sum_{g\in G} |\langle f, g\rangle|^2 \le C\|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^2),$$

where $\langle\cdot,\cdot\rangle$ and $\|\cdot\|_2 := \langle\cdot,\cdot\rangle^{\frac{1}{2}}$ denote the inner product and the norm of $L^2(\mathbb{R}^2)$. When $B = C$, $G$ is called a tight frame. The reader refers to [1], [8], [14], [16], [23], [27], [41], [42] for discussions on frames. In this paper, we consider affine (or wavelet) frames which are generated by the dilations and shifts of a set of functions. More precisely, for a function $f$ on $\mathbb{R}^2$, denote $f_{j,\mathbf{k}}(\mathbf{x}) = 2^j f(2^j\mathbf{x} - \mathbf{k})$. Functions $\psi^{(1)}, \psi^{(2)}, \cdots, \psi^{(L)}$ on $\mathbb{R}^2$, where $L \ge 3$, are called affine (or wavelet) framelets, or affine frame generators, just called framelets for short in this paper, if $G = \{\psi^{(1)}_{j,\mathbf{k}}, \psi^{(2)}_{j,\mathbf{k}}, \cdots, \psi^{(L)}_{j,\mathbf{k}}\}_{j\in\mathbf{Z},\mathbf{k}\in\mathbf{Z}^2}$ is a frame. In this case, $G$ is called an affine (or a wavelet) frame. We say that $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, generate a **wavelet bi-frame** (a **bi-frame** for short) of $L^2(\mathbb{R}^2)$ or a **dual wavelet frame** of $L^2(\mathbb{R}^2)$ if $\{\psi^{(1)}_{j,\mathbf{k}}, \cdots, \psi^{(L)}_{j,\mathbf{k}}\}_{j\in\mathbf{Z},\mathbf{k}\in\mathbf{Z}^2}$ and $\{\widetilde{\psi}^{(1)}_{j,\mathbf{k}}, \cdots, \widetilde{\psi}^{(L)}_{j,\mathbf{k}}\}_{j\in\mathbf{Z},\mathbf{k}\in\mathbf{Z}^2}$ 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)

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

For a sequence $\{p_\mathbf{k}\}_{\mathbf{k}\in\mathbf{Z}^2}$ of real numbers with finitely many $p_\mathbf{k}$ nonzero, let $p(\boldsymbol{\omega})$ denote the finite impulse response (FIR) filter with its impulse response coefficients $p_\mathbf{k}$ (here a factor 1/4 is multiplied):

$$p(\boldsymbol{\omega}) = \frac{1}{4}\sum_{\mathbf{k}\in\mathbf{Z}^2} p_\mathbf{k} e^{-i\mathbf{k}\boldsymbol{\omega}}.$$

$p(\boldsymbol{\omega})$ is also called the symbol of $\{p_\mathbf{k}\}_{\mathbf{k}\in\mathbf{Z}^2}$. A pair of FIR filter sets $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ is said to be **biorthogonal** or to form a perfect reconstruction (**PR**) frame filter bank if

$$\overline{p(\boldsymbol{\omega})}\widetilde{p}(\boldsymbol{\omega} + \pi\boldsymbol{\eta}_j) + \sum_{\ell=1}^{L} \overline{q^{(\ell)}(\boldsymbol{\omega})}\widetilde{q}^{(\ell)}(\boldsymbol{\omega} + \pi\boldsymbol{\eta}_j) = \begin{cases} 1, \ j = 0, \\ 0, \ 1 \le j \le 3, \end{cases}$$

where

$$\boldsymbol{\eta}_0 = (0,0), \ \boldsymbol{\eta}_1 = (-1,-1), \ \boldsymbol{\eta}_2 = (1,0), \ \boldsymbol{\eta}_3 = (0,1). \tag{1}$$

Let $\phi$ and $\widetilde{\phi}$ be the refinable (or scaling) functions satisfying the following refinement equations

$$\hat{\phi}(\boldsymbol{\omega}) = p(\frac{\boldsymbol{\omega}}{2})\hat{\phi}(\frac{\boldsymbol{\omega}}{2}), \ \hat{\widetilde{\phi}}(\boldsymbol{\omega}) = \widetilde{p}(\frac{\boldsymbol{\omega}}{2})\hat{\widetilde{\phi}}(\frac{\boldsymbol{\omega}}{2})$$

for some FIR filters $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ respectively. Throughout this paper, $\hat{f}$ denotes the Fourier transform of a function $f$ on $\mathbb{R}^2$: $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^2} f(\mathbf{x})e^{-i\mathbf{x}\cdot\boldsymbol{\omega}}d\mathbf{x}$, where $\mathbf{x}\cdot\boldsymbol{\omega} = x_1\omega_1 + x_2\omega_2$. Let $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, be the functions defined by

$$\hat{\psi}^{(\ell)}(\boldsymbol{\omega}) = q^{(\ell)}(\frac{\boldsymbol{\omega}}{2})\hat{\phi}(\frac{\boldsymbol{\omega}}{2}), \ \hat{\widetilde{\psi}}^{(\ell)}(\boldsymbol{\omega}) = \widetilde{q}^{(\ell)}\frac{\boldsymbol{\omega}}{2})\hat{\widetilde{\phi}}(\frac{\boldsymbol{\omega}}{2}),$$

for some FIR filters $q^{(\ell)}(\boldsymbol{\omega}), \widetilde{q}^{(\ell)}(\boldsymbol{\omega})$. The Mixed Unitary Extension Principle (**MUEP**) of [42] implies (see also [17, 20]) that if $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ form a PR FIR filter bank,

$\phi, \widetilde{\phi} \in L^2(\mathbb{R}^2)$ with $\widehat{\phi}(0,0)\widehat{\widetilde{\phi}}(0,0) \neq 0$, and that $p(0,0) = \widetilde{p}(0,0) = 1$, $q^{(\ell)}(0,0) = \widetilde{q}^{(\ell)}(0,0) = 0$, then $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, L$, generate a bi-frame of $L^2(\mathbb{R}^2)$.

Given a frame filter bank $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$, the multiresolution decomposition algorithm for input data $\mathcal{C} = \{c^0_{\mathbf{k}}\}$ is

$$c^{j+1}_{\mathbf{n}} = \frac{1}{4}\sum_{\mathbf{k}\in\mathbf{Z}^2} p_{\mathbf{k}-2\mathbf{n}} c^j_{\mathbf{k}}, \quad d^{(\ell,j+1)}_{\mathbf{n}} = \frac{1}{4}\sum_{\mathbf{k}\in\mathbf{Z}^2} q^{(\ell)}_{\mathbf{k}-2\mathbf{n}} c^j_{\mathbf{k}}, \tag{2}$$

with $\ell = 1, \cdots, L, \mathbf{n} \in \mathbf{Z}^2$ for $j = 0, 1, \cdots, J-1$, and the multiresolution reconstruction algorithm is given by

$$\widetilde{c}^j_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{p}_{\mathbf{k}-2\mathbf{n}} \widetilde{c}^{j+1}_{\mathbf{n}} + \sum_{1\le\ell\le L}\sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{q}^{(\ell)}_{\mathbf{k}-2\mathbf{n}} d^{(\ell,j+1)}_{\mathbf{n}} \tag{3}$$

with $\mathbf{k} \in \mathbf{Z}^2$ for $j = J-1, J-2, \cdots, 0$, where $\widetilde{c}^J_{\mathbf{n}} = c^J_{\mathbf{n}}$. $\{p, q^{(1)}, \cdots, q^{(L)}\}$ is called the **analysis frame filter set** and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ the **synthesis frame filter set**; and $p, \widetilde{p}$ are called lowpass filters and $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \le \ell \le L$ highpass (frame) filters. When $\{p, q^{(1)}, \cdots, q^{(L)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(L)}\}$ are biorthogonal, then $\widetilde{c}^j_{\mathbf{k}} = c^j_{\mathbf{k}}$, $0 \le j \le J-1$. $\{c^j_{\mathbf{k}}\}$, $\{d^{(\ell,j)}_{\mathbf{k}}\}$ are called the "approximation" (or "lowpass output") and the "details" (or "highpass outputs") of $\mathcal{C}$. When $d^{(\ell,j)}_{\mathbf{n}} = 0$ for $1 \le \ell \le L, 0 \le j \le J-1, \mathbf{n} \in \mathbf{Z}^2$, then (3) is reduced to $J$ steps of subdivision algorithm with subdivision mask $\{\widetilde{p}_{\mathbf{k}}\}_{\mathbf{k}}$ and initial control net $\{\widetilde{c}^J_{\mathbf{n}}\}_{\mathbf{n}}$:

$$\widetilde{c}^j_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{p}_{\mathbf{k}-2\mathbf{n}} \widetilde{c}^{j+1}_{\mathbf{n}}, j = J-1, J-2, \cdots, 0.$$

In this paper, we study bi-frames with 4 frame generators. For an FIR frame filter set $\{p, q^{(1)}, q^{(2)}, q^{(3)}, q^{(4)}\}$, with notation $q^{(0)}(\boldsymbol{\omega}) = p(\boldsymbol{\omega})$, write $q^{(\ell)}(\boldsymbol{\omega}), 0 \le \ell \le 4$ as

$$q^{(\ell)}(\boldsymbol{\omega}) = \frac{1}{2}\Big(q^{(\ell)}_0(2\boldsymbol{\omega}) + q^{(\ell)}_1(2\boldsymbol{\omega})e^{i(\omega_1+\omega_2)} + q^{(\ell)}_2(2\boldsymbol{\omega})e^{-i\omega_1} + q^{(\ell)}_3(2\boldsymbol{\omega})e^{-i\omega_2}\Big),$$

where $q^{(\ell)}_k(\boldsymbol{\omega}), 0 \le k \le 3$ are trigonometric polynomials. We define the **polyphase matrix** of the frame filter set $\{p, q^{(1)}, \cdots, q^{(4)}\}$ to be the $5 \times 4$ matrix $V(\boldsymbol{\omega})$ given by

$$V(\boldsymbol{\omega}) = \Big[q^{(\ell)}_k(\boldsymbol{\omega})\Big]_{0\le\ell\le4, 0\le k\le3}. \tag{4}$$

Clearly,

$$[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), \ \cdots, \ q^{(4)}(\boldsymbol{\omega})]^T = \frac{1}{2}V(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}),$$

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

$$I_0(\boldsymbol{\omega}) = [1, e^{i(\omega_1+\omega_2)}, e^{-i\omega_1}, e^{-i\omega_2}]^T. \tag{5}$$

Furthermore, we can easily find that two sets of frame filters $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(4)}\}$ are biorthogonal if and only if

$$V(\boldsymbol{\omega})^* \, \widetilde{V}(\boldsymbol{\omega}) = I_4, \quad \boldsymbol{\omega} \in \mathbb{R}^2,$$

where $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ are their polyphase matrices defined by (4). Throughout this paper, $M^*$ denotes the complex conjugate and transpose of a matrix $M$.

An FIR filter $p(\boldsymbol{\omega}), \boldsymbol{\omega}=(\omega_1,\omega_2)$ is said to have **sum rule order** $K$ if it satisfies that $p(0,0)=1$ and

$$\frac{\partial^{\alpha_1+\alpha_2}p(\boldsymbol{\omega})}{\partial\omega_1^{\alpha_1}\partial\omega_2^{\alpha_2}}\Big|_{\boldsymbol{\omega}=(\pi,\pi)}=0,\quad \frac{\partial^{\alpha_1+\alpha_2}p(\boldsymbol{\omega})}{\partial\omega_1^{\alpha_1}\partial\omega_2^{\alpha_2}}\Big|_{\boldsymbol{\omega}=(\pi,0)}=0,\quad \frac{\partial^{\alpha_1+\alpha_2}p(\boldsymbol{\omega})}{\partial\omega_1^{\alpha_1}\partial\omega_2^{\alpha_2}}\Big|_{\boldsymbol{\omega}=(0,\pi)}=0, \tag{6}$$

for all $(\alpha_1,\alpha_2)\in\mathbf{Z}_+^2$ with $\alpha_1+\alpha_2<K$. Under some condition, sum rule order is equivalent to the approximation order of the associated scaling function $\phi$, see [28].

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

$$\frac{\partial^{\alpha_1+\alpha_2}q(\boldsymbol{\omega})}{\partial\omega_1^{\alpha_1}\partial\omega_2^{\alpha_2}}\Big|_{\boldsymbol{\omega}=(0,0)}=0,$$

for all $(\alpha_1,\alpha_2)\in\mathbf{Z}_+^2$ with $\alpha_1+\alpha_2<N$. Clearly, if $q(\boldsymbol{\omega})$ has vanishing moment order $N$ and $\psi$ is the compactly supported function defined by $\hat{\psi}(\boldsymbol{\omega})=q(\frac{\boldsymbol{\omega}}{2})\hat{\phi}(\frac{\boldsymbol{\omega}}{2})$, where $\phi$ is a compactly supported function in $L^2(\mathbb{R}^2)$, then $\psi$ has the vanishing moments of order $N$:

$$\int_{\mathbb{R}^2}\psi(x_1,x_2)x_1^{\alpha_1}x_2^{\alpha_2}dx_1dx_2=0,\quad \alpha_1+\alpha_2<N,\ \alpha_1,\alpha_2\in\mathbf{Z}_+.$$

Most importantly, one can show that if $q(\boldsymbol{\omega})$ has vanishing moment order $N$, then when it is used as an analysis highpass filter, it annihilates discrete polynomials of total degree less than $N$. In other words, if the input $c_{\mathbf{k}}=P(\mathbf{k})$, where $P$ is a polynomial with total degree $<N$, then the (highpass) output with filter $q(\boldsymbol{\omega})$ is zero:

$$d_{\mathbf{n}}=\frac{1}{4}\sum_{\mathbf{k}\in\mathbf{Z}^2}q_{\mathbf{k}-2\mathbf{n}}P(\mathbf{k})=0,\quad \mathbf{n}\in\mathbf{Z}^2.$$

It is important in many applications such as data sparse representation that highpass filters annihilate discrete polynomials. In addition, it has been shown in [17] that the approximation power of the frame truncation operator of a biorthogonal frame filter bank depends on not only the sum rule order of the synthesis lowpass filter $\widetilde{p}(\boldsymbol{\omega})$ but also the vanishing moment orders of the highpass filters.

When we construct bi-frames, we choose the parameters such that the synthesis scaling function $\widetilde{\phi}$ is smoother than the analysis scaling function $\phi$, 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 vanishing moments (if it is possible).

# 3 6-fold symmetry dyadic bi-frames and associated templates

In this section, we show how frame multiresolution analysis and synthesis algorithms for regular vertices can be represented as templates. This is the key step for our approach of template-based frame construction. We also discuss the symmetry of frame filter banks in this section.

To describe multiresolution analysis and synthesis algorithms by means of templates, we consider a regular infinite mesh $\mathcal{C}=\{c_{\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ that can be be represented as the regular triangular mesh, denote by $\mathcal{M}_0$, as shown on the left of Fig. 1. The indices for the nodes of $\mathcal{M}_0$ are also indicated in this picture. The middle of Fig. 1 shows the dyadic refinement, where the nodes with circles ◯ form the coarse mesh. The lowpass output after a one-level decomposition is a lower resolution of $\mathcal{C}$ that could be considered as a "subsampling" of $\mathcal{C}$ on the vertices of the coarse mesh.

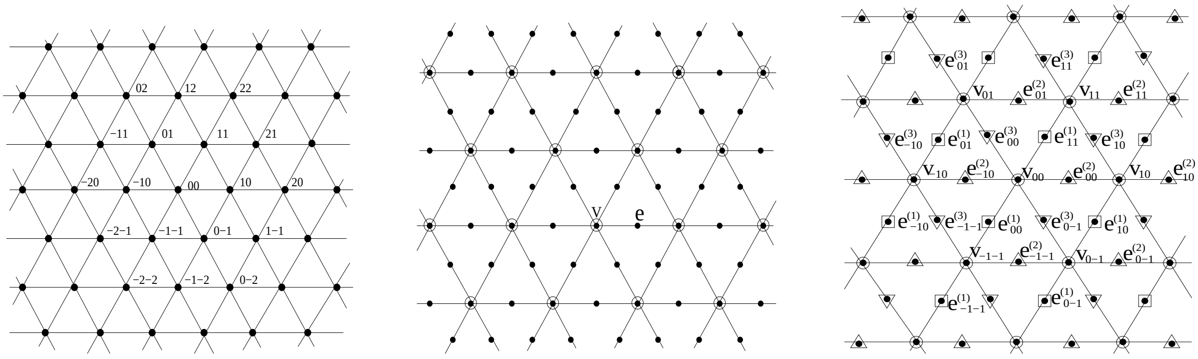

Figure 1: *Left: Indices for nodes and initial data; Middle: Coarse mesh; Right: Initial data separated into 4 groups:* $\{v_{\mathbf{k}}\}$, $\{e^{(1)}_{\mathbf{k}}\}$, $\{e^{(2)}_{\mathbf{k}}\}$, $\{e^{(3)}_{\mathbf{k}}\}$

The key to describe multiresolution algorithms by means of templates is to associate the outputs appropriately with the nodes of $\mathcal{M}_0$ with which the input data is associated (sampled). To this purpose, first, as in [32], we separate the nodes of $\mathcal{M}_0$ into different groups. More precisely, let $\mathbf{Z}^2$ denote the indices for the nodes of $\mathcal{M}_0$ as shown on the left of Fig. 1. The nodes are separated into two groups with one consisting of the nodes with indices $(2k_1, 2k_2)$ for the coarse mesh and the other group consisting of the remaining nodes with indices of $\mathbf{Z}^2 \backslash (2\mathbf{Z}^2)$. We call the nodes of the first group *type V* **nodes** (or **vertex nodes**), and those in the second group *type E* **nodes** (or **edge nodes**). We then further separate the *type E* nodes into three groups with labels in:

$$\{2\mathbf{k}-(1,1)\}_{\mathbf{k}\in\mathbf{Z}^2},\ \{2\mathbf{k}+(1,0)\}_{\mathbf{k}\in\mathbf{Z}^2},\ \{2\mathbf{k}+(0,1)\}_{\mathbf{k}\in\mathbf{Z}^2}.$$

See the right part of Fig. 1, where □, △ and ▽ denote these three groups of *type E* nodes respectively (recall that the big circles ◯ denote *type V* nodes).

For a regular mesh $\mathcal{C} = \{c_{\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ with vertices $c_{\mathbf{k}}$, since the multiresolution algorithm is applied to each component of $c_{\mathbf{k}}$, we assume $c_{\mathbf{k}}$ to be real numbers when we derive the corresponding templates for the algorithm. Let $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$ be the "approximation" and $\{d^{(1,\ell)}_{\mathbf{k}}\}_{\mathbf{k}}$, $1 \le \ell \le 4$ be the "details" after the decomposition algorithm (2) with $L = 4$ and an analysis frame filter set $\{p, q^{(1)}, \cdots, q^{(4)}\}$ is applied to $c^0_{\mathbf{k}} = c_{\mathbf{k}}$. Observe that $\{c_{2\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ is the set of data associated with *type V* nodes, $\{c_{2\mathbf{k}-(1,1)}\}_{\mathbf{k}\in\mathbf{Z}^2}$, $\{c_{2\mathbf{k}+(1,0)}\}_{\mathbf{k}\in\mathbf{Z}^2}$ and $\{c_{2\mathbf{k}+(0,1)}\}_{\mathbf{k}\in\mathbf{Z}^2}$ are three sets of the data associated with the above three groups of *type E* nodes. Denote

$$v_{\mathbf{k}} = c_{2\mathbf{k}},\ e^{(1)}_{\mathbf{k}} = c_{2\mathbf{k}-(1,1)},\ e^{(2)}_{\mathbf{k}} = c_{2\mathbf{k}+(1,0)},\ e^{(3)}_{\mathbf{k}} = c_{2\mathbf{k}+(0,1)},\ \mathbf{k}\in\mathbf{Z}^2. \tag{7}$$

Refer to the right picture of Fig. 1 for these four groups of data. We call *type V* vertices for $v_{\mathbf{k}}$ and *type E* vertices for any of $e^{(\ell)}_{\mathbf{k}}$, $\ell = 1, 2, 3$.

We also denote

$$\widetilde{v}_{\mathbf{k}} = c^1_{\mathbf{k}},\ \widetilde{g}_{\mathbf{k}} = d^{(1,1)}_{\mathbf{k}}, \widetilde{e}^{(1)}_{\mathbf{k}} = d^{(2,1)}_{\mathbf{k}},\ \widetilde{e}^{(2)}_{\mathbf{k}} = d^{(3,1)}_{\mathbf{k}},\ \widetilde{e}^{(3)}_{\mathbf{k}} = d^{(4,1)}_{\mathbf{k}}.$$

Then, the decomposition algorithm (2) can be written as

$$\left\{\begin{array}{l} \widetilde{v}_{\mathbf{k}} = \frac{1}{4}\sum_{\mathbf{k}'\in\mathbf{Z}^2} p_{\mathbf{k}'-2\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{g}_{\mathbf{k}} = \frac{1}{4}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(1)}_{\mathbf{k}'-2\mathbf{k}} c_{\mathbf{k}'}, \\ \widetilde{e}^{(1)}_{\mathbf{k}} = \frac{1}{4}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(2)}_{\mathbf{k}'-2\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{e}^{(2)}_{\mathbf{k}} = \frac{1}{4}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(3)}_{\mathbf{k}'-2\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{e}^{(3)}_{\mathbf{k}} = \frac{1}{4}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(4)}_{\mathbf{k}'-2\mathbf{k}} c_{\mathbf{k}'} \end{array}\right. \tag{8}$$

for $\mathbf{k} \in \mathbf{Z}^2$, and the reconstruction algorithm (3) is

$$c_{\mathbf{k}} = \sum_{\mathbf{k}' \in \mathbf{Z}^2} \{\widetilde{p}_{\mathbf{k}-2\mathbf{k}'} \widetilde{v}_{\mathbf{k}'} + \widetilde{q}^{(1)}_{\mathbf{k}-2\mathbf{k}'} \widetilde{g}_{\mathbf{k}'} + \widetilde{q}^{(2)}_{\mathbf{k}-2\mathbf{k}'} \widetilde{e}^{(1)}_{\mathbf{k}'} + \widetilde{q}^{(3)}_{\mathbf{k}-2\mathbf{k}'} \widetilde{e}^{(2)}_{\mathbf{k}'} + \widetilde{q}^{(4)}_{\mathbf{k}-2\mathbf{k}'} \widetilde{e}^{(3)}_{\mathbf{k}'}\}, \ \mathbf{k} \in \mathbf{Z}^2.$$

Considering $c_{\mathbf{k}}$ in the above equation with $\mathbf{k}$ in four different cases: $(2j_1, 2j_2)$, $(2j_1 - 1, 2j_2 - 1)$, $(2j_1 + 1, 2j_2)$, $(2j_1, 2j_2 + 1)$, and with the definitions of $v_{\mathbf{k}}, e^{(1)}_{\mathbf{k}}, e^{(2)}_{\mathbf{k}}, e^{(3)}_{\mathbf{k}}$ given in (7), one can further write the reconstruction algorithm (3) as

$$\begin{cases} v_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \{\widetilde{p}_{2\mathbf{n}} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{2\mathbf{n}} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{2\mathbf{n}} \widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{2\mathbf{n}} \widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(4)}_{2\mathbf{n}} \widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\}, \\ e^{(1)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \{\widetilde{p}_{2\mathbf{n}-(1,1)} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{2\mathbf{n}-(1,1)} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{2\mathbf{n}-(1,1)} \widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{2\mathbf{n}-(1,1)} \widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(4)}_{2\mathbf{n}-(1,1)} \widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\}, \\ e^{(2)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \{\widetilde{p}_{2\mathbf{n}+(1,0)} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{2\mathbf{n}+(1,0)} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{2\mathbf{n}+(1,0)} \widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{2\mathbf{n}+(1,0)} \widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(4)}_{2\mathbf{n}+(1,0)} \widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\}, \\ e^{(3)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \{\widetilde{p}_{2\mathbf{n}+(0,1)} \widetilde{v}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(1)}_{2\mathbf{n}+(0,1)} \widetilde{g}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(2)}_{2\mathbf{n}+(0,1)} \widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(3)}_{2\mathbf{n}+(0,1)} \widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}} + \widetilde{q}^{(4)}_{2\mathbf{n}+(0,1)} \widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\}. \end{cases} \tag{9}$$

Next, we associate the outputs $\widetilde{v}_{\mathbf{k}}, \widetilde{g}_{\mathbf{k}}, \widetilde{e}^{(\ell)}_{\mathbf{k}}, \ell = 1, 2, 3$ with the nodes of $\mathcal{M}_0$. We associate both the "approximation" $\{\widetilde{v}_{\mathbf{k}}\}_{\mathbf{k} \in \mathbf{Z}^2}$ and the first highpass output $\widetilde{g}_{\mathbf{k}}$ with *type V* nodes with labels $(2k_1, 2k_2)$, and associate the second, third and fourth highpass outputs $\widetilde{e}^{(1)}_{\mathbf{k}}, \widetilde{e}^{(2)}_{\mathbf{k}}, \widetilde{e}^{(3)}_{\mathbf{k}}$ with the *type E* nodes with labels $(k_1 - 1, k_2 - 1), (k_1 + 1, k_2)$ and $(k_1, k_2 + 1)$ respectively. In this way, both analysis and synthesis algorithms can be represented as templates.

Observe that when the "details" $\widetilde{g}_{\mathbf{k}}, \widetilde{e}^{(j)}_{\mathbf{k}}, \ell = 1, 2, 3, \mathbf{k} \in \mathbf{Z}^2$ are set to be zero, then (9) is reduced to the subdivision algorithm:

$$v_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \widetilde{p}_{2\mathbf{n}} \widetilde{v}_{\mathbf{k}-\mathbf{n}}, e^{(1)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \widetilde{p}_{2\mathbf{n}-(1,1)} \widetilde{v}_{\mathbf{k}-\mathbf{n}}, e^{(2)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \widetilde{p}_{2\mathbf{n}+(1,0)} \widetilde{v}_{\mathbf{k}-\mathbf{n}}, e^{(3)}_{\mathbf{k}} = \sum_{\mathbf{n} \in \mathbf{Z}^2} \widetilde{p}_{2\mathbf{n}+(0,1)} \widetilde{v}_{\mathbf{k}-\mathbf{n}},$$

from which the subdivision templates are derived.

Like subdivision templates, when analysis and synthesis algorithms templates are used for surface processing, they must have certain symmetry. First the templates to obtain $\widetilde{e}^{(1)}_{\mathbf{k}}, \widetilde{e}^{(2)}_{\mathbf{k}}$ and $\widetilde{e}^{(3)}_{\mathbf{k}}$ must be the same since all $\widetilde{e}^{(1)}_{\mathbf{k}}, \widetilde{e}^{(2)}_{\mathbf{k}}$ and $\widetilde{e}^{(3)}_{\mathbf{k}}$ are associated with *type E* vertices and they should be treated equally. For the same reason, the templates to recover $e^{(1)}_{\mathbf{k}}, e^{(2)}_{\mathbf{k}}$ and $e^{(3)}_{\mathbf{k}}$ by (9) should be identical. Furthermore, the templates to obtain $\widetilde{v}_{\mathbf{k}}$ and $\widetilde{g}_{\mathbf{k}}$ by (8), and that to recover $v_{\mathbf{k}}$ by (9) must be rotational and reflective invariant with respect to the coarse mesh. In addition, the template to obtain $\widetilde{e}^{(1)}_{\mathbf{k}}$ (now we know it is the same template to obtain $\widetilde{e}^{(2)}_{\mathbf{k}}$ and $\widetilde{e}^{(3)}_{\mathbf{k}}$) and the template to recover $e^{(1)}_{\mathbf{k}}$ (the same template to recover $e^{(2)}_{\mathbf{k}}$ and $e^{(3)}_{\mathbf{k}}$) have certain symmetry. Following the definition of the 6-fold line symmetry of biorthogonal wavelet filter banks in [32], we define below the 6-fold symmetry for a frame filter bank which results in templates with the desired symmetry mentioned above.

**Definition 1.** *A (dyadic) frame filter set* $\{p, q^{(1)}, \cdots, q^{(4)}\}$ *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* $S_0, \cdots, S_5$ *on the left of Fig. 2; (ii) the coefficients* $q^{(2)}_{\mathbf{k}}$ *of its second highpass filter is symmetric around the axes* $S_0, S_3''$ *on the right of Fig. 2; and (iii)* $q^{(3)}_{\mathbf{k}}, q^{(4)}_{\mathbf{k}}$ *of other two highpass filters* $q^{(3)}(\boldsymbol{\omega})$ *and* $q^{(4)}(\boldsymbol{\omega})$ *are the* $\frac{2\pi}{3}$ *and* $\frac{4\pi}{3}$ *rotations of* $q^{(2)}_{\mathbf{k}}$*.*

In Fig. 2, the multi-indices $\mathbf{k} = (k_1, k_2)$ are the indices for the coefficients $p_{\mathbf{k}}, q^{(\ell)}_{\mathbf{k}}, 1 \leq \ell \leq 4$. On the right of Fig. 2, *type V* nodes with labels $2\mathbf{k}$ are indicated by ○. From the definition

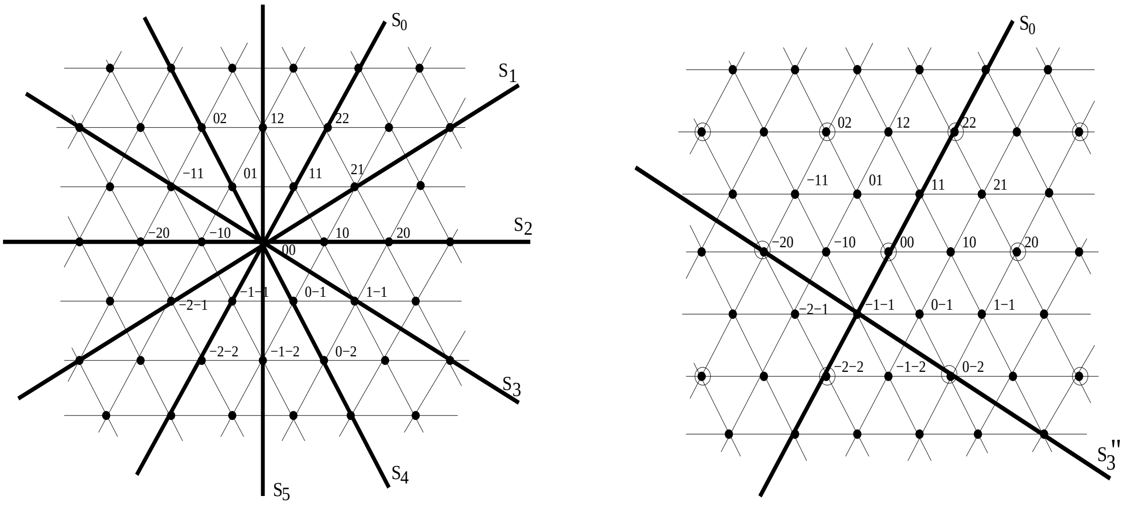

Figure 2: *Left: Symmetry lines for lowpass filter and 1st frame highpass filter; Right: Symmetry lines for 2nd frame highpass filter*

of 6-fold symmetry, we know $q^{(3)}_{\mathbf{k}}$ and $q^{(4)}_{\mathbf{k}}$ also have two symmetric axes with symmetric centers $(1,0)$ and $(0,1)$ respectively.

The 6-fold symmetry of a frame filter set $\{p, q^{(1)}, \cdots, q^{(4)}\}$ can be characterized by the symmetry of its polyphase matrix $V(\boldsymbol{\omega})$.

**Proposition 1.** *A frame filter set $\{p, q^{(1)}, \cdots, q^{(4)}\}$ has 6-fold axial symmetry if and only if its polyphase matrix $V(\boldsymbol{\omega})$ defined in (4) satisfies*

$$V(L_0\boldsymbol{\omega}) = S_{01}V(\boldsymbol{\omega})S_{02},\ V(R_1^{-T}\boldsymbol{\omega}) = S_1(\boldsymbol{\omega})V(\boldsymbol{\omega})S_2(\boldsymbol{\omega}), \tag{10}$$

*where*

$$R_1 = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix},\ L_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ S_{01} = \begin{bmatrix} I_3 & \mathbf{0} \\ \mathbf{0} & L_0 \end{bmatrix},\ S_{02} = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & L_0 \end{bmatrix},$$

$$S_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\omega_2} \\ 0 & 0 & e^{-i(\omega_1+\omega_2)} & 0 & 0 \\ 0 & 0 & 0 & e^{i\omega_1} & 0 \end{bmatrix},\quad S_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & e^{i(\omega_1+\omega_2)} & 0 \\ 0 & 0 & 0 & e^{-i\omega_1} \\ 0 & e^{-i\omega_2} & 0 & 0 \end{bmatrix}.$$

The proof of Proposition 1 is similar to that for the characterization of a 6-fold symmetry of dyadic wavelet filter set given in [32]. Compared with a 6-fold symmetric wavelet filter set in [32], our 6-fold symmetric frame filter set has one extra highpass filter $q^{(1)}(\boldsymbol{\omega})$ which has the same symmetry as the lowpass filter $p(\boldsymbol{\omega})$.

Since the algorithms to obtain $\widetilde{e}^{(1)}_{\mathbf{k}}, \widetilde{e}^{(2)}_{\mathbf{k}}, \widetilde{e}^{(3)}_{\mathbf{k}}$ are the same, and those to recover *type E* vertices $e^{(1)}_{\mathbf{k}}, e^{(2)}_{\mathbf{k}}, e^{(3)}_{\mathbf{k}}$ are also identical, we may simply let $e$ denote *type E* vertices, and use $\widetilde{e}$ to denote the second to fourth outputs after the decomposition algorithm. Thus, the decomposition algorithm is to decompose the original data $\{v\} \cup \{e\}$ into $\{\widetilde{v}\}$, $\{\widetilde{g}\}$ and $\{\widetilde{e}\}$, and the reconstruction algorithm recovers $\{v\} \cup \{e\}$ from $\{\widetilde{v}\}$, $\{\widetilde{g}\}$ and $\{\widetilde{e}\}$, see Fig. 3. In the following sections when we describe multiresolution algorithms, we simply use $v, e$ and $\widetilde{v}, \widetilde{g}, \widetilde{e}$.

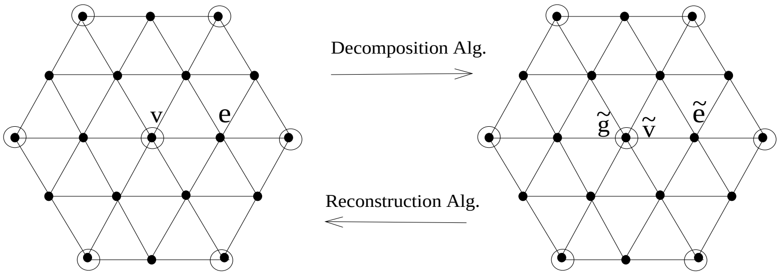

Figure 3: *Decomposition and reconstruction algorithms*

To construct 6-fold symmetric bi-frames, we start with symmetric templates of decomposition and reconstruction algorithms. Using the lifting scheme idea [15, 48], the algorithm templates are given by several iterative steps with each step given by a template. In the next 4 sections we consider the algorithms given by 2, 3 and 4 steps of iterations. With the templates and decomposition and reconstruction algorithms (8)(9), we then obtain the corresponding bi-frame filter bank which is given by some parameters. Finally, we select the parameters based upon the smoothness and vanishing moments of the framelets.

# 4 2-step bi-frame multiresolution algorithm

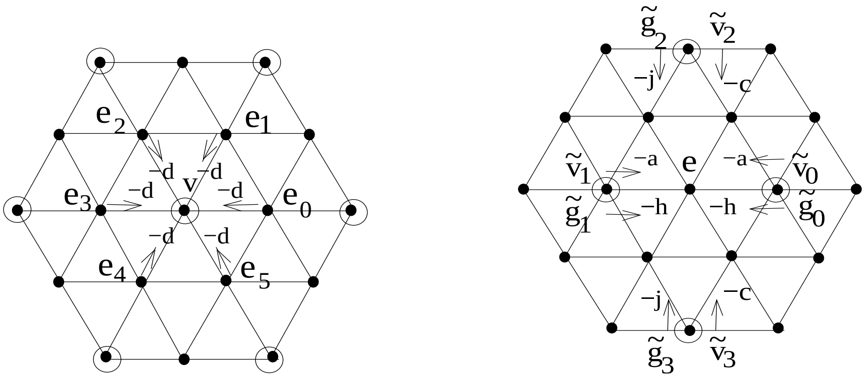

Figure 4: *Left: Template to obtain lowpass output $\widetilde{v}$ in Decomposition Alg. Step 1 (template to obtain first highpass output $\widetilde{g}$ is similar with $-d$ replaced by $-n$); Right: Decomposition Alg. Step 2*

In this section we consider a 2-step frame algorithm. For given triangular mesh $\mathcal{C}$ (or equivalently, for given $\{v\}$ and $\{e\}$), the multiresolution decomposition algorithm is given by (11) and (12) and shown in Fig. 4, where $b, d, n, a, c, h, j$ are constants to be determined. Namely, first we obtain lowpass output $\widetilde{v}$ and the first highpass output $\widetilde{g}$, both associated with *type V* nodes of $\mathcal{M}_0$, with $\widetilde{v}, \widetilde{g}$ given by formulas in (11). Then, based on $\widetilde{v}, \widetilde{g}$ obtained, we obtain other three highpass outputs $\{\widetilde{e}\}$ $(= \{\widetilde{e}^{(1)}_{\mathbf{k}}\} \cup \{\widetilde{e}^{(2)}_{\mathbf{k}}\} \cup \{\widetilde{e}^{(3)}_{\mathbf{k}}\})$ associated with *type E* nodes of $\mathcal{M}_0$ with $\widetilde{e}$ given in (12). The algorithm is very simple and efficient.

**2-step Decomposition Algorithm:**

$$\text{Step 1.} \begin{cases} \widetilde{v} = \frac{1}{b}\Big\{v - d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}, \\ \widetilde{g} = v - n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5) \end{cases} \tag{11}$$

$$\text{Step 2. } \widetilde{e} = e - a(\widetilde{v}_0 + \widetilde{v}_1) - c(\widetilde{v}_2 + \widetilde{v}_3) - h(\widetilde{g}_0 + \widetilde{g}_1) - j(\widetilde{g}_2 + \widetilde{g}_3). \tag{12}$$

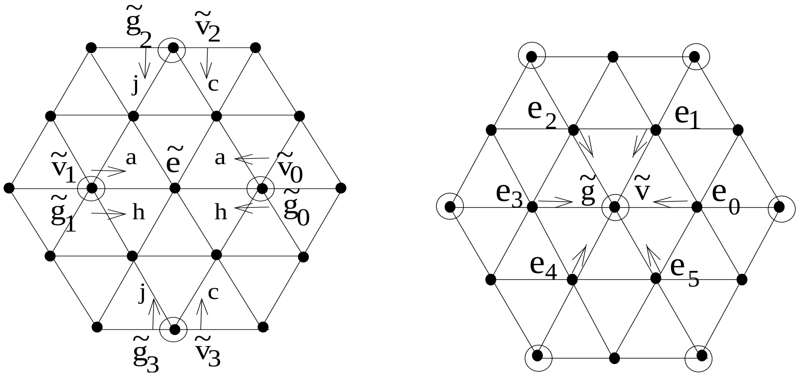

Figure 5: *Left: Reconstruction Alg. Step 1; Right: Reconstruction Alg. Step 2*

The multiresolution reconstruction algorithm is given by (13) and (14) and shown in Fig. 5, where $b, d, n, a, c, h, j$ are the same constants in the decomposition algorithm and $t \in \mathbb{R}$. The reconstruction algorithm is the reverse algorithm of the decomposition algorithm. More precisely, first we replace $\widetilde{e}$ by $e$ given in (13). After that, based on $e$ obtained in Step 1, $\widetilde{v}, \widetilde{g}$ are replaced by $v$ given by formula (14). Again, the reconstruction algorithm from $\widetilde{v}, \widetilde{g}, \widetilde{e}$ to $v, e$ is very simple and efficient.

**2-step Reconstruction Algorithm:**

$$\text{Step 1. } e = \widetilde{e} + a(\widetilde{v}_0 + \widetilde{v}_1) + c(\widetilde{v}_2 + \widetilde{v}_3) + h(\widetilde{g}_0 + \widetilde{g}_1) + j(\widetilde{g}_2 + \widetilde{g}_3) \tag{13}$$

$$\text{Step 2. } \begin{array}{c} v = t\Big\{b\widetilde{v} + d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\} + \\ (1-t)\Big\{\widetilde{g} + n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}. \end{array} \tag{14}$$

With the formulas in (8) and (9), and by careful calculations (the reader refers to [30] for some detailed discussion on how to obtain 1-D filters associated with some specific given templates), one can obtain the filter bank $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(4)}\}$ corresponding to this (2-step) algorithm to be

$$\Big[p(\boldsymbol{\omega}),\ q^{(1)}(\boldsymbol{\omega}),\ \cdots,\ q^{(4)}(\boldsymbol{\omega})\Big]^T = D_1(2\boldsymbol{\omega})D_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}),$$

$$\Big[\widetilde{p}(\boldsymbol{\omega}),\ \widetilde{q}^{(1)}(\boldsymbol{\omega}),\ \cdots,\ \widetilde{q}^{(4)}(\boldsymbol{\omega})\Big]^T = \frac{1}{4}\widetilde{D}_1(2\boldsymbol{\omega})\widetilde{D}_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}),$$

where $I_0(\boldsymbol{\omega})$ is defined by (5), and

$$D_0(\boldsymbol{\omega}) = \begin{bmatrix} \frac{1}{b} & -\frac{d}{b}(1+xy) & -\frac{d}{b}(1+\frac{1}{x}) & -\frac{d}{b}(1+\frac{1}{y}) \\ 1 & -n(1+xy) & -n(1+\frac{1}{x}) & -n(1+\frac{1}{y}) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{15}$$

$$D_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -a(1+\frac{1}{xy}) - c(\frac{1}{x}+\frac{1}{y}) & -h(1+\frac{1}{xy}) - j(\frac{1}{x}+\frac{1}{y}) & 1 & 0 & 0 \\ -a(1+x) - c(xy+\frac{1}{y}) & -h(1+x) - j(xy+\frac{1}{y}) & 0 & 1 & 0 \\ -a(1+y) - c(xy+\frac{1}{x}) & -h(1+y) - j(xy+\frac{1}{x}) & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

$$\widetilde{D}_0(\boldsymbol{\omega}) = \begin{bmatrix} tb & 0 & 0 & 0 \\ 1-t & 0 & 0 & 0 \\ (td+(1-t)n)(1+\frac{1}{xy}) & 1 & 0 & 0 \\ td+(1-t)n)(1+x) & 0 & 1 & 0 \\ td+(1-t)n)(1+y) & 0 & 0 & 1 \end{bmatrix}, \tag{17}$$

$$\widetilde{D}_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & a(1+xy)+c(x+y) & a(1+\frac{1}{x})+c(\frac{1}{xy}+y) & a(1+\frac{1}{y})+c(\frac{1}{xy}+x) \\ 0 & 1 & h(1+xy)+j(x+y) & h(1+\frac{1}{x})+j(\frac{1}{xy}+y) & h(1+\frac{1}{y})+j(\frac{1}{xy}+x) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

Here and throughout this paper, we use $x, y$ to denote $e^{-i\omega_1}, e^{-i\omega_2}$ respectively:

$$x = e^{-i\omega_1},\ y = e^{-i\omega_2}. \tag{19}$$

Observe that the polyphase matrices $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ of $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(4)}\}$ are $2D_1(\boldsymbol{\omega})D_0(\boldsymbol{\omega})$ and $\frac{1}{2}\widetilde{D}_1(\boldsymbol{\omega})\widetilde{D}_0(\boldsymbol{\omega})$ respectively. One can easily show that $D_0(\boldsymbol{\omega})^*\ \widetilde{D}_0(\boldsymbol{\omega}) = I_4, D_1(\boldsymbol{\omega})^*\ \widetilde{D}_1(\boldsymbol{\omega}) = I_5, \boldsymbol{\omega} \in \mathbb{R}^2$. Thus, $V(\boldsymbol{\omega})^*\ \widetilde{V}(\boldsymbol{\omega}) = I_4, \boldsymbol{\omega} \in \mathbb{R}^2$, and hence, $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(4)}\}$ are indeed biorthogonal. Furthermore, one can also easily show that $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ satisfy (10). Thus both $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(4)}\}$ are 6-fold symmetric.

Solving the system of equations for sum rule order 3 of $\widetilde{p}$ and sum rule order 1 of $p$, we have

$$d = -\frac{1}{2},\ b = 4,\ n = \frac{1}{6},\ a = \frac{3}{8},\ c = \frac{1}{8},\ t = \frac{1}{10}. \tag{20}$$

The resulting $p$ and $\widetilde{p}$, which actually have sum rule orders 4 and 2 respectively because of their symmetry, are

$$\begin{aligned} p(\boldsymbol{\omega}) &= \tfrac{1}{8}e^{i(\omega_1+\omega_2)}(1+e^{-i\omega_1})(1+e^{-i\omega_2})(1+e^{-i(\omega_1+\omega_2)}), \\ \widetilde{p}(\boldsymbol{\omega}) &= \tfrac{1}{64}e^{i2(\omega_1+\omega_2)}(1+e^{-i\omega_1})^2(1+e^{-i\omega_2})^2(1+e^{-i(\omega_1+\omega_2)})^2. \end{aligned} \tag{21}$$

The scaling functions $\phi$ and $\widetilde{\phi}$ corresponding to these two lowpass filters are respectively the continuous linear box-spline $B_{111}$ and $C^2$ box-spline $B_{222}$ associated with the direction sets (refer to [4] for box-splines)

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & -1 & -1 \end{bmatrix}.$$

$\widetilde{p}$ in (21) is the filter (also called mask) for Loop's scheme [37], one of the most commonly used subdivision schemes. Thus we are particularly interested in the choices of the parameters given in (20). However, we cannot choose other parameters such that the resulting $\widetilde{q}^{(\ell)}, 1 \leq \ell \leq 4$ have 1 vanishing moment. The corresponding $q^{(1)}, q^{(2)}, q^{(3)}, q^{(4)}$ automatically have vanishing moment order 2 with $q^{(2)}, q^{(3)}, q^{(4)}$ depending on $h, j$. Furthermore, if $h = \frac{9}{8} - j$, then $q^{(2)}, q^{(3)}, q^{(4)}$ have vanishing moment order 4. Though for such choices of parameters, the vanishing moment condition of $\widetilde{q}^{(\ell)}$ for MUEP is not satisfied, in the following we provide the corresponding filters because of the simplicity of the algorithm. For the value of $j$, we choose $j = \frac{3}{32}$ so that $q^{(2)}, q^{(3)}, q^{(4)}$ have fewer nonzero coefficients. The resulting highpass filters (with $h = 33/32, j = 3/32$) are

$$\begin{aligned}
q^{(1)}(\boldsymbol{\omega}) &= 1 - \tfrac{1}{6}(x + y + xy + \tfrac{1}{x} + \tfrac{1}{y} + \tfrac{1}{xy}),\\
q^{(2)}(\boldsymbol{\omega}) &= \tfrac{1}{8}\big\{\tfrac{10}{xy} + \tfrac{1}{y} + \tfrac{1}{x} + \tfrac{1}{x^2y} + \tfrac{1}{xy^2} - 9 - \tfrac{9}{x^2y^2} - \tfrac{1}{x^2} - \tfrac{1}{y^2} + x + y + xy + \tfrac{1}{x^3y^2} + \tfrac{1}{x^2y^3} + \tfrac{1}{x^3y^3}\big\},\\
q^{(3)}(\boldsymbol{\omega}) &= \tfrac{1}{8}\big\{10x + \tfrac{1}{y} + xy + x^2y + \tfrac{x}{y} - 9 - 9x^2 - \tfrac{1}{y^2} - x^2y^2 + \tfrac{1}{xy} + y + \tfrac{1}{x} + x^3y + \tfrac{x^2}{y} + x^3\big\},\\
q^{(4)}(\boldsymbol{\omega}) &= \tfrac{1}{8}\big\{10y + \tfrac{1}{x} + xy + xy^2 + \tfrac{y}{x} - 9 - 9y^2 - \tfrac{1}{x^2} - x^2y^2 + \tfrac{1}{xy} + x + \tfrac{1}{y} + xy^3 + \tfrac{y^2}{x} + y^3\big\},\\
\widetilde{q}^{(1)}(\boldsymbol{\omega}) &= \tfrac{3}{128}\Big\{\tfrac{81}{5} + 11(x + y + xy + \tfrac{1}{x} + \tfrac{1}{y} + \tfrac{1}{xy}) + x^2y + xy^2 + \tfrac{1}{x^2y} + \tfrac{1}{xy^2} + \tfrac{x}{y} + \tfrac{y}{x}\\
&\qquad + \tfrac{13}{10}(x^2 + y^2 + x^2y^2 + \tfrac{1}{x^2} + \tfrac{1}{y^2} + \tfrac{1}{x^2y^2})\Big\},\\
\widetilde{q}^{(2)}(\boldsymbol{\omega}) &= \tfrac{1}{4xy} + \tfrac{1}{40}(1 + \tfrac{1}{x^2y^2}),\ \widetilde{q}^{(3)}(\boldsymbol{\omega}) = \tfrac{x}{4} + \tfrac{1}{40}(1 + x^2),\ \widetilde{q}^{(4)}(\boldsymbol{\omega}) = \tfrac{y}{4} + \tfrac{1}{40}(1 + y^2).
\end{aligned}$$

(Again, $x, y$ are given by (19).) Observe that the lowpass filters $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ in the above PR frame filter bank are supported on $[-1, 1]^2$ and $[-2, 2]^2$ respectively with $\widetilde{p}(\boldsymbol{\omega})$ being the filter for Loop's scheme. We use Loop-$\mathrm{F}_{1,2}$ to denote this PR frame filter bank.

Next we consider other choices of parameters such that each of the highpass filters has at least one vanishing moment. Solving the system of equations for sum rule order 1 of $p$ and $\widetilde{p}$, and for vanishing moment order 1 of $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \leq \ell \leq 4$, we have

$$b = 4,\ d = -\frac{1}{2},\ n = \frac{1}{6},\ a = \frac{1}{2} - c,\ t = 1.$$

In this case the resulting $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \leq \ell \leq 4$ have 2 vanishing moments, and $p$ and $\widetilde{p}$ have sum rule order 2 with $p$ given by (21) and $\widetilde{p}$ depending on $c$. If $c = -\frac{1}{4}$, we have the Sobolev (numerically) smoothest $\widetilde{\phi}$ that is in $W^{0.44076}$. In this case if $j = h = 0$, then $\widetilde{q}^{(1)}(\boldsymbol{\omega}) = 0$. Thus the frame filter bank is reduced to be a biorthogonal wavelet filter bank. In this paper, $W^s$ $(s > 0)$ denotes the Sobolev space consisting of $f$ on $\mathbb{R}^2$ with $\int_{\mathbb{R}^2}(1 + |\boldsymbol{\omega}|^2)^s |\hat{f}(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} < \infty$. The reader refers to [33] for computing the Sobolev smoothness of a refinable function $\phi(\mathbf{x})$.

From the above discussion, we know that to obtain framelets with a higher smoothness order and/or higher vanishing moment orders, we need to consider algorithms with more iterative steps. In the next section we consider a 3-step algorithm.

## 5 3-step bi-frame multiresolution algorithm

In this section we consider a 3-step bi-frame multiresolution algorithm. The decomposition algorithm is given by (22)-(24) and shown in Fig. 6, where $b, d, n, a, c, h, j, w, n_1$ are constants to be determined. Namely, first we replace all $v$ associated with *type V* nodes of $\mathcal{M}_0$ by $v'', g''$ given by formula (22). Then, based on $v'', g''$ obtained, we replace all $e$ associated with *type E* nodes

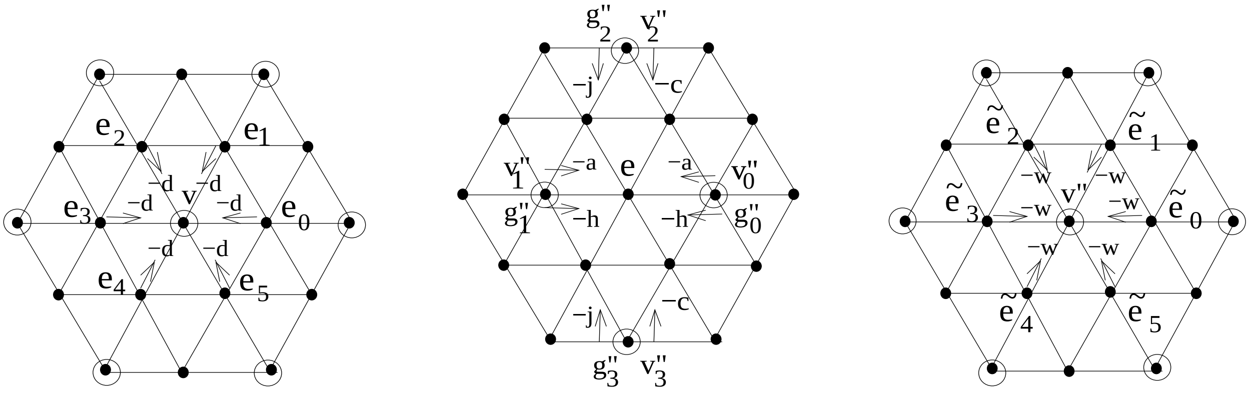

Figure 6: *Left: Template to obtain $v''$ in Decomposition Alg. Step 1 (template to obtain $g''$ is similar with $-d$ replaced by $-n$); Middle: Decomposition Alg. Step 2; Right: Template to obtain lowpass output $\widetilde{v}$ in Decomposition Alg. Step 3 (template to obtain first highpass output $\widetilde{g}$ is similar with $v''$ and $-w$ replaced by $g''$ and $-n_1$ resp.)*

of $\mathcal{M}_0$ by $\widetilde{e}$ given in formula (23). After that, based on $\widetilde{e}$ obtained in Step 2, all $v'', g''$ in Step 1 are updated by $\widetilde{v}$ and $\widetilde{g}$ given in formula (24).

**3-step Dyadic Frame Decomposition Algorithm:**

$$\text{Step 1.} \left\{ \begin{array}{l} v'' = \frac{1}{b}\Big\{v - d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}, \\ g'' = v - n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5) \end{array} \right. \tag{22}$$

$$\text{Step 2. } \widetilde{e} = e - a(v''_0 + v''_1) - c(v''_2 + v''_3) - h(g''_0 + g''_1) - j(g''_2 + g''_3) \tag{23}$$

$$\text{Step 3.} \left\{ \begin{array}{l} \widetilde{v} = v'' - w(\widetilde{e}_0 + \widetilde{e}_1 + \widetilde{e}_2 + \widetilde{e}_3 + \widetilde{e}_4 + \widetilde{e}_5), \\ \widetilde{g} = g'' - n_1(\widetilde{e}_0 + \widetilde{e}_1 + \widetilde{e}_2 + \widetilde{e}_3 + \widetilde{e}_4 + \widetilde{e}_5). \end{array} \right. \tag{24}$$

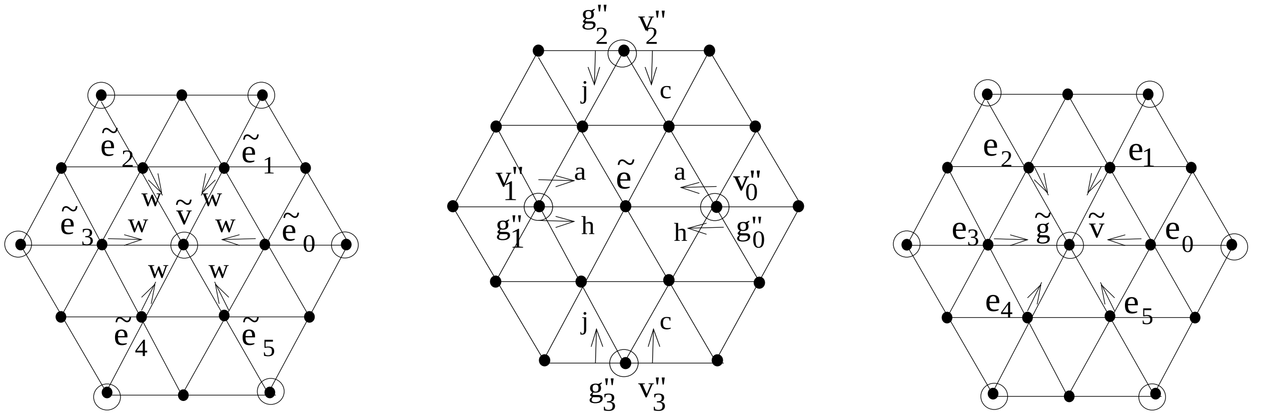

Figure 7: *Left: Template to obtain $v''$ in Reconstruction Alg. Step 1 (template to obtain $g''$ is similar with $\widetilde{v}$ and $w$ replaced by $\widetilde{g}$ and $n_1$ resp.); Middle: Reconstruction Alg. Step 2; Right: Reconstruction Alg. Step 3*

The multiresolution reconstruction algorithm is given by (25)-(27) and shown in Fig. 7, where $b, d, n, a, c, h, j, w, n_1$ are the same constants in the multiresolution decomposition algorithm and

$t \in \mathbb{R}$. More precisely, first we replace the lowpass output $\widetilde{v}$ and the first highpass output $\widetilde{g}$ both associated with *type V* nodes of $\mathcal{M}_0$ by $v''$ and $g''$ respectively given by formula (25). After that, based on $v'', g''$ obtained, we obtain $e$ with formula in (26). Finally, based on $e$ obtained in Step 2, all $v'', g''$ in Step 1 are replaced by $v$ with formula (27).

**3-step Dyadic Frame Reconstruction Algorithm:**

$$\text{Step 1.} \begin{cases} v'' = \widetilde{v} + w(\widetilde{e}_0 + \widetilde{e}_1 + \widetilde{e}_2 + \widetilde{e}_3 + \widetilde{e}_4 + \widetilde{e}_5), \\ g'' = \widetilde{g} + n_1(\widetilde{e}_0 + \widetilde{e}_1 + \widetilde{e}_2 + \widetilde{e}_3 + \widetilde{e}_4 + \widetilde{e}_5) \end{cases} \tag{25}$$

$$\text{Step 2. } e = \widetilde{e} + a(v''_0 + v''_1) + c(v''_2 + v''_3) + h(g''_0 + g''_1) + j(g''_2 + g''_3) \tag{26}$$

$$\text{Step 3. } \begin{aligned} v = t\Big\{bv'' + d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\} + \\ (1-t)\Big\{g'' + n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}. \end{aligned} \tag{27}$$

For the 3-step algorithm (22)-(27), with the formulas in (8) and (9), and the filters for the 2-step algorithm given above, one can obtain its corresponding filter bank, also denoted by $\{p, q^{(1)}, \cdots, q^{(4)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(3)}\}$, to be

$$\Big[p(\boldsymbol{\omega}),\ q^{(1)}(\boldsymbol{\omega}),\ \cdots,\ q^{(4)}(\boldsymbol{\omega})\Big]^T = D_2(2\boldsymbol{\omega})D_1(2\boldsymbol{\omega})D_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \tag{28}$$

$$\Big[\widetilde{p}(\boldsymbol{\omega}),\ \widetilde{q}^{(1)}(\boldsymbol{\omega}),\ \cdots,\ \widetilde{q}^{(4)}(\boldsymbol{\omega})\Big]^T = \frac{1}{4}\widetilde{D}_2(2\boldsymbol{\omega})\widetilde{D}_1(2\boldsymbol{\omega})\widetilde{D}_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \tag{29}$$

where $I_0(\boldsymbol{\omega})$ is defined by (5), $D_0(\boldsymbol{\omega})$, $D_1(\boldsymbol{\omega}), \widetilde{D}_0(\boldsymbol{\omega})$ and $\widetilde{D}_1(\boldsymbol{\omega})$ are defined by (15), (16), (17) and (18) respectively, and

$$D_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & -w(1+xy) & -w(1+\frac{1}{x}) & -w(1+\frac{1}{y}) \\ 0 & 1 & -n_1(1+xy) & -n_1(1+\frac{1}{x}) & -n_1(1+\frac{1}{y}) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{30}$$

and $\widetilde{D}_2(\boldsymbol{\omega}) = \Big(D_2(\boldsymbol{\omega})^{-1}\Big)^*$:

$$\widetilde{D}_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ w(1+\frac{1}{xy}) & n_1(1+\frac{1}{xy}) & 1 & 0 & 0 \\ w(1+x) & n_1(1+x) & 0 & 1 & 0 \\ w(1+y) & n_1(1+y) & 0 & 0 & 1 \end{bmatrix}, \tag{31}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2)$, $x = e^{-i\omega_1}, y = e^{-i\omega_2}$, as defined in (19).

One can easily verify that this pair of frame filter sets are biorthogonal and that they have 6-fold symmetry by looking at their polyphase matrices.

After 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 4 for $q^{(\ell)}, \ell = 2, 3, 4$, we have that

$$c = \frac{1}{8},\ a = \frac{3}{8},\ n = \frac{1}{6},\ d = \frac{1+20w}{2(60w-1)},\ b = \frac{4}{1-60w},\ h = \frac{15}{2}w + \frac{9}{8} - j,\ t = \frac{1}{10} - 6w.$$

The resulting $\widetilde{p}$ is the filter given in (21). Thus the corresponding scaling $\widetilde{\phi}$ is the $C^2$ cubic box spline $B_{222}$.

If $w=-\frac{3}{20}$, then $\widetilde{q}^{(\ell)}, 1\le \ell\le 4$ have vanishing moment order 2. In this case we can choose $j$ such that $\phi$ has certain smoothness. For example, if $j=-\frac{23}{16}$, then $\phi\in W^{1.07362}$. For the parameter $n_1$, we may just set $n_1=0$. In the following we provide the corresponding filters with the choice of $w=-\frac{3}{20}, j=-\frac{23}{16}, n_1=0$ (hence, $d=\frac{1}{10}, b=\frac{2}{5}, h=\frac{23}{16}, t=1$):

$$\begin{aligned}
p(\boldsymbol{\omega}) = {} & \tfrac{19}{320}\Big\{\tfrac{944}{19}-\tfrac{55}{19}(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy})+2(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2}+\tfrac{x}{y}+\tfrac{y}{x})\\
& -\tfrac{144}{19}(x^2+y^2+x^2y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+\tfrac{1}{x^2y^2})+x^3+y^3+\tfrac{1}{x^3}+\tfrac{1}{y^3}\\
& +xy^3+x^3y+x^2y^3+x^3y^2+x^3y^3+\tfrac{1}{x^3y^3}+\tfrac{1}{x^3y^2}+\tfrac{1}{x^2y^3}+\tfrac{1}{x^3y}+\tfrac{1}{xy^3}+\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2}\Big\},\\
q^{(1)}(\boldsymbol{\omega}) = {} & 1-\tfrac{1}{6}(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy}),\\
q^{(2)}(\boldsymbol{\omega}) = {} & \tfrac{1}{48xy}\Big\{34+6(x+y+\tfrac{1}{x}+\tfrac{1}{y})+24(xy+\tfrac{1}{xy})-84(\tfrac{x}{y}+\tfrac{y}{x})\\
& -7(x^2y^2+x^2y+xy^2+\tfrac{1}{x^2y^2}+\tfrac{1}{x^2y}+\tfrac{1}{xy^2})+13(x^2+y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2})\Big\},\\
q^{(3)}(\boldsymbol{\omega}) = {} & q^{(2)}(-\omega_1-\omega_2,\omega_2),\ q^{(4)}(\boldsymbol{\omega}) = q^{(2)}(-\omega_1-\omega_2,\omega_1),
\end{aligned}$$

$$\begin{aligned}
\widetilde{p}(\boldsymbol{\omega}) = {} & \tfrac{1}{64x^2y^2}(1+x)^2(1+y)^2(1+xy)^2,\\
\widetilde{q}^{(1)}(\boldsymbol{\omega}) = {} & \tfrac{23}{64}\Big\{-\tfrac{3}{5}-(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy})+(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2}+\tfrac{x}{y}+\tfrac{y}{x})\\
& +\tfrac{1}{10}(x^2+y^2+x^2y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+\tfrac{1}{x^2y^2})\Big\},\\
\widetilde{q}^{(2)}(\boldsymbol{\omega}) = {} & \tfrac{3}{640xy}\Big\{\tfrac{142}{3}-4(x+y+\tfrac{1}{x}+\tfrac{1}{y})-\tfrac{1}{6}(xy+\tfrac{1}{xy})-(\tfrac{x}{y}+\tfrac{y}{x})\\
& -3(x^2y^2+x^2y+xy^2+\tfrac{1}{x^2y^2}+\tfrac{1}{x^2y}+\tfrac{1}{xy^2})-(x^2+y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+x^3y^2+x^2y^3+\tfrac{1}{x^3y^2}+\tfrac{1}{x^2y^3})\\
& -\tfrac{1}{2}(x^3y^3+x^3y+xy^3+\tfrac{1}{x^3y^3}+\tfrac{1}{x^3y}+\tfrac{1}{xy^3})\Big\},\\
\widetilde{q}^{(3)}(\boldsymbol{\omega}) = {} & \widetilde{q}^{(2)}(-\omega_1-\omega_2,\omega_2),\ \widetilde{q}^{(4)}(\boldsymbol{\omega}) = \widetilde{q}^{(2)}(-\omega_1-\omega_2,\omega_1),
\end{aligned}$$

where $x,y$ are defined by (19). Observe that $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ in the above bi-frame filter bank are supported on $[-3,3]^2$ and $[-2,2]^2$ respectively with $\widetilde{p}(\boldsymbol{\omega})$ being the filter for Loop's scheme. We call the resulting framelets **Loop's scheme-based bi-framelets**, and we use Loop-F$_{3,2}$ to denote this frame filter bank.

**Remark 1.** *The resulting $\phi\in W^{1.07362}$ of Loop-F$_{3,2}$ is not quite smooth. It seems that the frame system does not result in compactly supported framelets with nice smoothness and small supports. Actually, compared with the biorthogonal wavelet system, it does. It is indicated in [32] that it is impossible to construct such biorthogonal wavelets that they have the same supports as these resulting framelets and that their associated synthesis scaling function $\widetilde{\phi}$ is the $C^2$ cubic spline $B_{222}$ and the analysis scaling function $\phi$ is in $L^2(\mathbb{R}^2)$. In particular, $\phi$ of the biorthogonal wavelets constructed in [2] is not in $L^2(\mathbb{R}^2)$.*

Finally, in this section, we also mention that if $w=\frac{1}{5}, j=-\frac{3}{4}$ (hence, the corresponding $d=\frac{5}{22}, b=-\frac{4}{11}, h=\frac{27}{8}, t=-\frac{11}{10}$), then the resulting $p$ is

$$p(\boldsymbol{\omega})=\frac{1}{64}(1+e^{-i\omega_1})(1+e^{-i\omega_2})(1+e^{i(\omega_1+\omega_2)})(1+e^{-i2\omega_1})(1+e^{-i2\omega_2})(1+e^{i2(\omega_1+\omega_2)}). \qquad (32)$$

Thus, the corresponding $\phi$ is the $C^2$ box-spline associated with the direction set

$$\Theta_1 = \begin{bmatrix} 1 & 0 & -1 & 2 & 0 & -2 \\ 0 & 1 & -1 & 0 & 2 & -2 \end{bmatrix}. \tag{33}$$

In this case, $q^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}, \widetilde{q}^{(4)}$ depend on $n_1$. If $n_1 = -\frac{1}{15}$, $\widetilde{q}^{(\ell)}, \ell = 2, 3, 4$ have vanishing moment order 2. However $\widetilde{q}^{(1)}$, which is independent of $n_1$, has no vanishing moment.

To obtain a smoother $\phi$ or higher vanishing moment order $q^{(\ell)}$, we need to consider algorithms with more iterative steps.

## 6 Butterfly scheme-based bi-frames

In this section, we consider bi-frames with the synthesis lowpass filter being the symbol of the butterfly interpolatory scheme [19]. The decomposition algorithm of the butterfly scheme-based bi-frames is given by (22)(34)(24), and the reconstruction is presented by (25)(35)(27), where $b, d, n, a, c, r, h, j, w, n_1, t$ are some constants. Observe that the difference between the 3-step frame algorithm and the butterfly scheme-based frame algorithm is that the latter has bigger templates in Step 2 of its decomposition and reconstruction algorithms. Refer to Fig. 8 for Step 2 of the decomposition and reconstruction algorithms.

**Step 2 of Butterfly Scheme-based Frame Decomposition Algorithm:**

Step 2. $$\begin{aligned}\widetilde{e} = e - a(v_0'' + v_1'') - c(v_2'' + v_3'') - r(v_4'' + v_5'' + v_6'' + v_7'') \\ -h(g_0'' + g_1'') - j(g_2'' + g_3'') - s(g_4'' + g_5'' + g_6'' + g_7''),\end{aligned} \tag{34}$$

**Step 2 of Butterfly Scheme-based Frame Reconstruction Algorithm:**

Step 2. $$\begin{aligned}e = \widetilde{e} + a(v_0'' + v_1'') + c(v_2'' + v_3'') + r(v_4'' + v_5'' + v_6'' + v_7'') \\ +h(g_0'' + g_1'') + j(g_2'' + g_3'') + s(g_4'' + g_5'' + g_6'' + g_7'').\end{aligned} \tag{35}$$

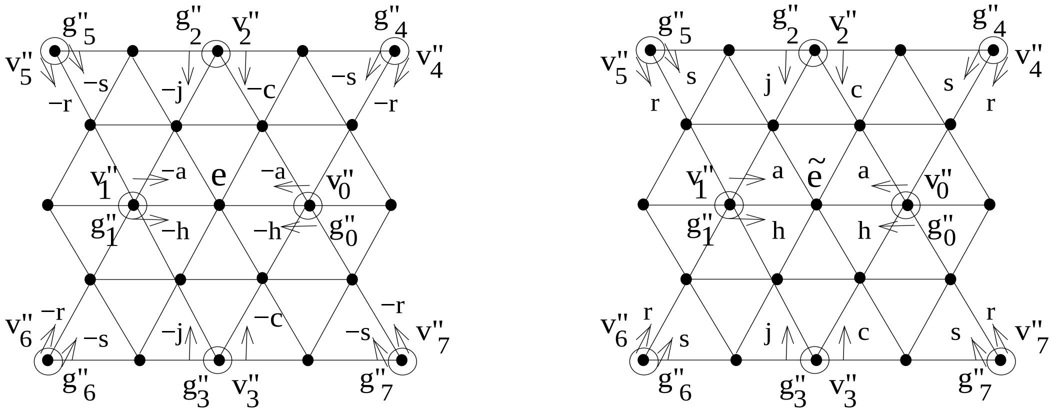

Figure 8: *Left: Decomposition Alg. Step 2 of butterfly scheme-based frame algorithm; Right: Reconstruction Alg. Step 2 of butterfly scheme-based frame algorithm*

One can get the corresponding filters to be given by (28) and (29) with $D_1(\boldsymbol{\omega})$ and $\widetilde{D}_1(\boldsymbol{\omega})$ replaced accordingly by

$$D_1(\boldsymbol{\omega}) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -a(1+\frac{1}{xy}) - c(\frac{1}{x}+\frac{1}{y}) - r(x+y+\frac{1}{x^2y}+\frac{1}{xy^2}) & -h(1+\frac{1}{xy}) - j(\frac{1}{x}+\frac{1}{y}) - s(x+y+\frac{1}{x^2y}+\frac{1}{xy^2}) & 1 & 0 & 0 \\ -a(1+x) - c(xy+\frac{1}{y}) - r(\frac{1}{xy}+y+x^2y+\frac{x}{y}) & -h(1+x) - j(xy+\frac{1}{y}) - s(\frac{1}{xy}+y+x^2y+\frac{x}{y}) & 0 & 1 & 0 \\ -a(1+y) - c(xy+\frac{1}{x}) - r(\frac{1}{xy}+x+xy^2+\frac{y}{x}) & -h(1+y) - j(xy+\frac{1}{x}) - s(\frac{1}{xy}+x+xy^2+\frac{y}{x}) & 0 & 0 & 1 \end{bmatrix},$$

$$\widetilde{D}_1(\boldsymbol{\omega}) = \big(D_1(\boldsymbol{\omega})^{-1}\big)^* =$$

$$\left[\begin{matrix} 1 & 0 & a(1+xy) + c(x+y) + r(\frac{1}{x}+\frac{1}{y}+x^2y+xy^2) \\ 0 & 1 & h(1+xy) + j(x+y) + s(\frac{1}{x}+\frac{1}{y}+x^2y+xy^2) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}\right.$$

$$\left.\begin{matrix} a(1+\frac{1}{x}) + c(\frac{1}{xy}+y) + r(xy+\frac{1}{y}+\frac{1}{x^2y}+\frac{y}{x}) & a(1+\frac{1}{y}) + c(\frac{1}{xy}+x) + r(xy+\frac{1}{x}+\frac{1}{xy^2}+\frac{x}{y}) \\ h(1+\frac{1}{x}) + j(\frac{1}{xy}+y) + s(xy+\frac{1}{y}+\frac{1}{x^2y}+\frac{y}{x}) & h(1+\frac{1}{y}) + j(\frac{1}{xy}+x) + s(xy+\frac{1}{x}+\frac{1}{xy^2}+\frac{x}{y}) \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{matrix}\right].$$

If $t = \frac{1}{b},\ d = (1-b)n$, then the subdivision scheme corresponding to the resulting $\widetilde{p}(\boldsymbol{\omega})$ is interpolatory; namely $\widetilde{p}_{0,0} = 1, \widetilde{p}_{2\mathbf{k}} = 0$ for $\mathbf{k} \in \mathbf{Z}^2\backslash\{\mathbf{0}\}$. In addition, if

$$a = \frac{1}{2},\ c = \frac{1}{8},\ r = -\frac{1}{16},$$

then the corresponding subdivision scheme is the butterfly interpolatory scheme in [19] with $\widetilde{\phi} \in W^{2.44076}$. Furthermore, if

$$b = 1,\ n = \frac{1}{6},\ w = -\frac{1}{8},\ h = -j - 2s,$$

then the resulting $p(\boldsymbol{\omega})$ has sum rule order 2, $q^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(\ell)}(\boldsymbol{\omega}), 1 \le \ell \le 4$ have vanishing moment order 2, and $q^{(\ell)}(\boldsymbol{\omega}), \ell = 2, 3, 4$ have vanishing moment order 4. We can choose $j, s$ such that $p(\boldsymbol{\omega})$ has certain smoothness. For example, if $j = -0.45519128680281, s = 0.03436808229118$, then the resulting $\phi$ is in $W^{1.27077}$; if $j = -\frac{7}{16},\ s = \frac{1}{32}$, then $\phi \in W^{1.26809}$; and if $j = -\frac{21}{64}, s = 0$, then $\phi \in W^{1.18774}$. In the following, we provide the corresponding filters with $j = -\frac{21}{64}, s = 0, n_1 = 0$

and other parameters given above:

$$
\begin{aligned}
p(\boldsymbol{\omega}) &= \tfrac{1}{256}\Big\{97 + \tfrac{163}{4}(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy}) - \tfrac{7}{2}(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2}+\tfrac{x}{y}+\tfrac{y}{x})\\
&\quad -\tfrac{19}{2}(x^2+y^2+x^2y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+\tfrac{1}{x^2y^2}) - \tfrac{7}{4}(x^3y^3+\tfrac{1}{x^3}+\tfrac{1}{y^3}+xy^3+x^3y+x^2y^3\\
&\quad +x^3y^2+y^3+x^3+\tfrac{1}{x^3y^3}+\tfrac{1}{x^3y^2}+\tfrac{1}{x^2y^3}+\tfrac{1}{x^3y}+\tfrac{1}{xy^3}+\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2})\\
&\quad +4(x^2y^4+x^4y^2+\tfrac{1}{x^2y^4}+\tfrac{1}{x^4y^2}+\tfrac{y^2}{x^2}+\tfrac{x^2}{y^2})\Big\},\\
q^{(1)}(\boldsymbol{\omega}) &= 1-\tfrac{1}{6}(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy}),\\
q^{(2)}(\boldsymbol{\omega}) &= \tfrac{1}{128xy}\Big\{142-106(xy+\tfrac{1}{xy})+26(\tfrac{x}{y}+\tfrac{y}{x})+7(x^2y^2+x^2y+xy^2+\tfrac{1}{x^2y^2}+\tfrac{1}{x^2y}+\tfrac{1}{xy^2})\\
&\quad -7(x^2+y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2}+\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2})+8(x^3y+xy^3+\tfrac{1}{xy^3}+\tfrac{1}{x^3y})\Big\},\\
q^{(3)}(\boldsymbol{\omega}) &= q^{(2)}(-\omega_1-\omega_2,\omega_2),\ q^{(4)}(\boldsymbol{\omega}) = q^{(2)}(-\omega_1-\omega_2,\omega_1),
\end{aligned}
$$

$$
\begin{aligned}
\widetilde{p}(\boldsymbol{\omega}) &= \tfrac{1}{4}\Big\{1+\tfrac{1}{2}(x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy})+\tfrac{1}{8}(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2}+\tfrac{x}{y}+\tfrac{y}{x})\\
&\quad -\tfrac{1}{16}(xy^3+x^3y+x^2y^3+x^3y^2+\tfrac{1}{x^3y^2}+\tfrac{1}{x^2y^3}+\tfrac{1}{x^3y}+\tfrac{1}{xy^3}+\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2})\Big\},\\
\widetilde{q}^{(1)}(\boldsymbol{\omega}) &= \tfrac{21}{256}\Big\{x+y+xy+\tfrac{1}{x}+\tfrac{1}{y}+\tfrac{1}{xy}-(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2}+\tfrac{x}{y}+\tfrac{y}{x})\Big\},\\
\widetilde{q}^{(2)}(\boldsymbol{\omega}) &= \tfrac{1}{256xy}\Big\{56-5(x+y+\tfrac{1}{x}+\tfrac{1}{y})-8(xy+\tfrac{1}{xy})-\tfrac{1}{2}(x^2+y^2+\tfrac{1}{x^2}+\tfrac{1}{y^2})-4(x^2y^2+\tfrac{1}{x^2y^2})\\
&\quad -\tfrac{7}{2}(x^2y+xy^2+\tfrac{1}{x^2y}+\tfrac{1}{xy^2})+\tfrac{1}{2}(\tfrac{x^2}{y}+\tfrac{x}{y^2}+\tfrac{y^2}{x}+\tfrac{y}{x^2})-\tfrac{1}{x^2y^3}-\tfrac{1}{x^3y^2}-x^3y^2-x^2y^3\\
&\quad +\tfrac{1}{2}(\tfrac{1}{x^4y^3}+\tfrac{1}{x^3y^4}+\tfrac{1}{x^2y^4}+\tfrac{1}{x^4y^2}+\tfrac{1}{y^3}+\tfrac{1}{x^3}+x^3y^4+x^4y^3+x^4y^2+x^2y^4+y^3+x^3)\Big\},\\
\widetilde{q}^{(3)}(\boldsymbol{\omega}) &= \widetilde{q}^{(2)}(-\omega_1-\omega_2,\omega_2),\ \widetilde{q}^{(4)}(\boldsymbol{\omega}) = \widetilde{q}^{(2)}(-\omega_1-\omega_2,\omega_1),
\end{aligned}
$$

where $x, y$ are defined by (19).

**Remark 2.** *The butterfly scheme-based spherical wavelets were constructed in [46]. These wavelets will results in butterfly scheme-based multiresolution algorithms for surface processing. However, they can be used for a surface with regular vertices and extraordinary vertices of valence* 5 *only. In addition, those multiresolution algorithms are non-stationary; namely the algorithms are different from one scale to the next finer one. If we set $h, j, s$ to zero in the above algorithms (22)(34)(24), and (25)(35)(27), then we will obtain butterfly scheme-based wavelets. More precisely, the wavelet decomposition and reconstruction algorithms are given by (36)(37) and (38)(39) respectively:*

**2-step Butterfly Scheme-based Wavelet Decomposition Algorithm:**

*Step* 1. $\widetilde{e} = e - \frac{1}{2}(v_0+v_1) - \frac{1}{8}(v_2+v_3) + \frac{1}{16}(v_4+v_5+v_6+v_7),$ (36)

*Step 2.* $\widetilde{v} = v - w(\widetilde{e}_0+\widetilde{e}_1+\widetilde{e}_2+\widetilde{e}_3+\widetilde{e}_4+\widetilde{e}_5);$ (37)

**2-step Butterfly Scheme-based Wavelet Reconstruction Algorithm:**

*Step 1.* $v = \widetilde{v} + w(\widetilde{e}_0+\widetilde{e}_1+\widetilde{e}_2+\widetilde{e}_3+\widetilde{e}_4+\widetilde{e}_5),$ (38)

*Step* 2. $e = \widetilde{e} + \frac{1}{2}(v_0+v_1) + \frac{1}{8}(v_2+v_3) - \frac{1}{16}(v_4+v_5+v_6+v_7).$ (39)

*Clearly, when all details $\widetilde{e}$ in (38)(39) are set to zero, the reconstruction algorithm (38)(39) is reduced to the butterfly subdivision scheme:*

$$v = \widetilde{v}, \quad e = \frac{1}{2}(\widetilde{v}_0 + \widetilde{v}_1) + \frac{1}{8}(\widetilde{v}_2 + \widetilde{v}_3) - \frac{1}{16}(\widetilde{v}_4 + \widetilde{v}_5 + \widetilde{v}_6 + \widetilde{v}_7).$$

*When $w = -\frac{1}{8}$, the resulting $\phi$ is in $W^{0.03512}$, $p$ has sum order 2 while $\widetilde{p}$ is the mask for the butterfly scheme. Thus we have the butterfly scheme-based biorthogonal wavelets.*

## 7 4-step bi-frame multiresolution algorithm

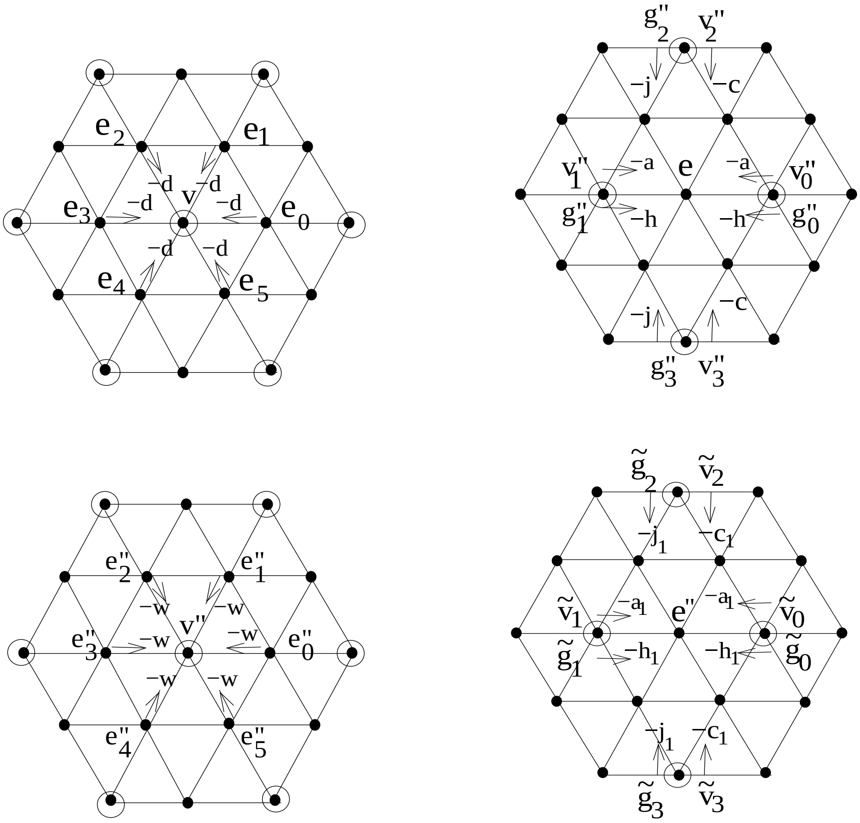

Figure 9: *Top-left: Template to obtain $v''$ in Decomposition Alg. Step 1 (template to obtain $g''$ is similar with $-d$ replaced by $-n$); Top-right: Decomposition Alg. Step 2; Bottom-left: Template to obtain lowpass output $\widetilde{v}$ in Decomposition Alg. Step 3 (template to obtain first highpass output $\widetilde{g}$ is similar with $v''$ and $-w$ replaced by $g''$ and $-n_1$ resp.); Bottom-right: Decomposition Alg. Step 4*

In this section, we discuss a 4-step frame algorithm. The decomposition algorithm and reconstruction algorithm are given by (40)-(43) and (44)-(47), and shown in Figs. 9 and 10, where $b, d, n, a, c, h, j, w, n_1,\ a_1, c_1, h_1, j_1, t$ are constants to be determined.

**4-step Dyadic Frame Decomposition Algorithm:**

$$\text{Step 1. } \begin{cases} v'' = \frac{1}{b}\Big\{v - d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}, \\ g'' = v - n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5) \end{cases} \tag{40}$$

$$\text{Step 2. } e'' = e - a(v''_0 + v''_1) - c(v''_2 + v''_3) - h(g''_0 + g''_1) - j(g''_2 + g''_3) \tag{41}$$

$$\text{Step 3. } \begin{cases} \widetilde{v} = v'' - w(e''_0 + e''_1 + e''_2 + e''_3 + e''_4 + e''_5), \\ \widetilde{g} = g'' - n_1(e''_0 + e''_1 + e''_2 + e''_3 + e''_4 + e''_5) \end{cases} \tag{42}$$

$$\text{Step 4. } \widetilde{e} = e'' - a_1(\widetilde{v}_0 + \widetilde{v}_1) - c_1(\widetilde{v}_2 + \widetilde{v}_3) - h_1(\widetilde{g}_0 + \widetilde{g}_1) - j_1(\widetilde{g}_2 + \widetilde{g}_3). \tag{43}$$

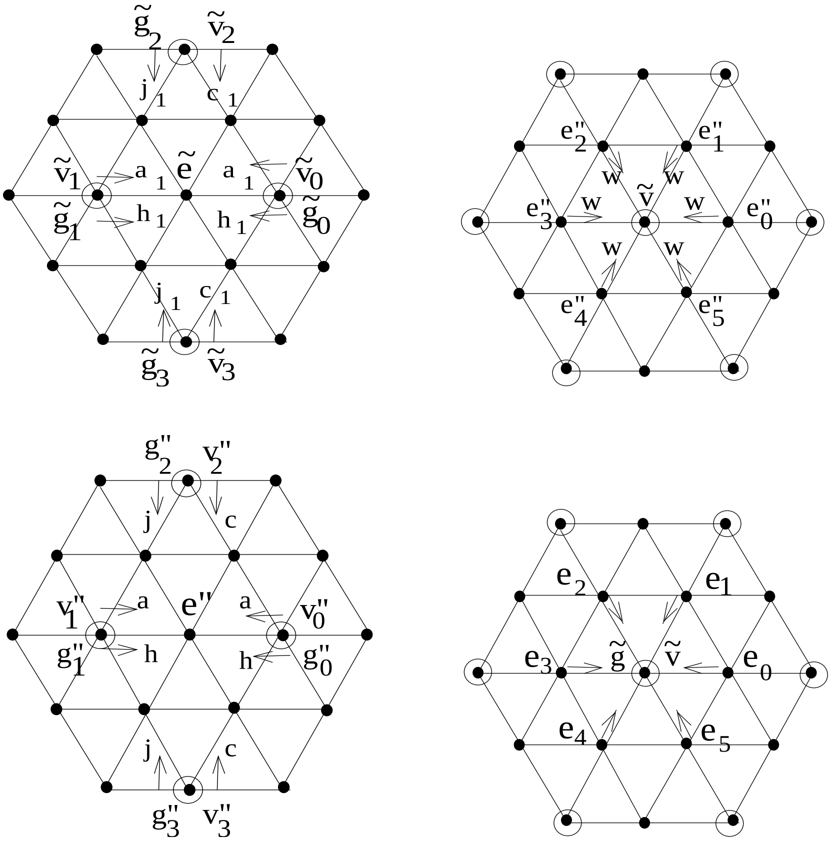

Figure 10: *Top-left: Reconstruction Alg. Step 1; Top-right: Template to obtain $v''$ in Reconstruction Alg. Step 2 (template to obtain $g''$ is similar with $\widetilde{v}$ and $w$ replaced by $\widetilde{g}$ and $n_1$ resp.); Bottom-left: Reconstruction Alg. Step 3; Bottom-right: Reconstruction Alg. Step 4*

**4-step Dyadic Frame Reconstruction Algorithm:**

$$\text{Step 1. } e'' = \widetilde{e} + a_1(\widetilde{v}_0 + \widetilde{v}_1) + c_1(\widetilde{v}_2 + \widetilde{v}_3) + h_1(\widetilde{g}_0 + \widetilde{g}_1) + j_1(\widetilde{g}_2 + \widetilde{g}_3) \tag{44}$$

$$\text{Step 2. } \begin{cases} v'' = \widetilde{v} + w(e''_0 + e''_1 + e''_2 + e''_3 + e''_4 + e''_5), \\ g'' = \widetilde{g} + n_1(e''_0 + e''_1 + e''_2 + e''_3 + e''_4 + e''_5) \end{cases} \tag{45}$$

$$\text{Step 3. } e = e'' + a(v''_0 + v''_1) + c(v''_2 + v''_3) + h(g''_0 + g''_1) + j(g''_2 + g''_3) \tag{46}$$

$$\text{Step 4. } \begin{array}{c} v = t\Big\{bv'' + d(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\} + \\ (1-t)\Big\{g'' + n(e_0 + e_1 + e_2 + e_3 + e_4 + e_5)\Big\}. \end{array} \tag{47}$$

With the formulas in (8) and (9), and using the filter bank for the 3-step algorithm in §5, one can obtain the filter bank $\{p, q^{(1)}, \cdots, q^{(4)}\}$, $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(3)}\}$ corresponding to the algorithms (40)-(47):

$$\begin{aligned} \Big[p(\boldsymbol{\omega}),\ q^{(1)}(\boldsymbol{\omega}),\ \cdots,\ q^{(4)}(\boldsymbol{\omega})\Big]^T &= D_3(2\boldsymbol{\omega})D_2(2\boldsymbol{\omega})D_1(2\boldsymbol{\omega})D_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \\ \Big[\widetilde{p}(\boldsymbol{\omega}),\ \widetilde{q}^{(1)}(\boldsymbol{\omega}),\ \cdots,\ \widetilde{q}^{(4)}(\boldsymbol{\omega})\Big]^T &= \frac{1}{4}\widetilde{D}_3(2\boldsymbol{\omega})\widetilde{D}_2(2\boldsymbol{\omega})\widetilde{D}_1(2\boldsymbol{\omega})\widetilde{D}_0(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \end{aligned}$$

where $I_0(\boldsymbol{\omega})$ is defined by (5), $D_0(\boldsymbol{\omega})$, $D_1(\boldsymbol{\omega}), \widetilde{D}_0(\boldsymbol{\omega})$, $\widetilde{D}_1(\boldsymbol{\omega})$, $D_2(\boldsymbol{\omega})$, and $\widetilde{D}_2(\boldsymbol{\omega})$ are defined by (15), (16), (17), (18), (30) and (31) respectively, and

$$D_3(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -a_1(1+\frac{1}{xy}) - c_1(\frac{1}{x}+\frac{1}{y}) & -h_1(1+\frac{1}{xy}) - j_1(\frac{1}{x}+\frac{1}{y}) & 1 & 0 & 0 \\ -a_1(1+x) - c_1(xy+\frac{1}{y}) & -h_1(1+x) - j_1(xy+\frac{1}{y}) & 0 & 1 & 0 \\ -a_1(1+y) - c_1(xy+\frac{1}{x}) & -h_1(1+y) - j_1(xy+\frac{1}{x}) & 0 & 0 & 1 \end{bmatrix}, \tag{48}$$

and $\widetilde{D}_3(\boldsymbol{\omega}) = \Big(D_3(\boldsymbol{\omega})^{-1}\Big)^*$:

$$\widetilde{D}_3(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & a_1(1+xy)+c_1(x+y) & a_1(1+\frac{1}{x})+c_1(\frac{1}{xy}+y) & a_1(1+\frac{1}{y})+c_1(\frac{1}{xy}+x) \\ 0 & 1 & h_1(1+xy)+j_1(x+y) & h_1(1+\frac{1}{x})+j_1(\frac{1}{xy}+y) & a_1(1+\frac{1}{y})+c_1(\frac{1}{xy}+x) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{49}$$

For this 4-step algorithm, we can choose parameters such that $q^{(\ell)}, \widetilde{q}^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment order 2 with $\phi, \widetilde{\phi}$ in $C^2$. With

$$t = \frac{132w^3n + 24w^2n - 18n_1w^2 - 120w^3n_1 - 22w^3 + n_1^2 - 5n_1^2w - 4w^2}{n_1^2(1-5w)}, a_1 = -\frac{24wn - 3n_1 + 15n_1w - 4w}{24n_1w},$$

$$c_1 = \frac{12wn - 3n_1 + 15n_1w - 2w}{24n_1w},\ d = n + \frac{40w^2 - 3w - 1 - 2n + 10wn}{2t(1-5w)},\ b = \frac{n_1}{w} + \frac{22w^2 + 4w - n_1 + 5wn_1}{tw(1-5w)},$$

$$h = \frac{2d + 32w + 1}{192w(d-n)},\ c = \frac{16wb - b - 16n + 16d - 2nb}{192w(d-n)},\ a = \frac{32n - 32d - 32wb - b - 2nb}{192w(d-n)},\ j = \frac{1 + 2d - 16w}{192w(d-n)},$$

the resulting $p(\boldsymbol{\omega})$ is given by (32), $\widetilde{p}(\boldsymbol{\omega})$ has sum rule order 4, and $q^{(\ell)}(\boldsymbol{\omega}), \widetilde{q}^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment order 2. Thus the corresponding scaling $\phi$ is the $C^2$ box-spline associated with the direction set $\Theta_1$ in (33). $\widetilde{p}(\boldsymbol{\omega})$ depends on $w, n, n_1$. If we choose

$$w = -0.15407716460528,\ n = -0.56146830232430,\ n_1 = 0.54008429124822,$$

then the resulting $\widetilde{\phi} \in W^{3.19626}$, while if

$$w = -\frac{5}{32},\ n = -\frac{9}{16},\ n_1 = \frac{71}{128}, \tag{50}$$

then $\widetilde{\phi} \in W^{3.05267}$. Below we also provide other corresponding parameters when $w, n, n_1$ are given by (50):

$$[b, d, a, c, h, j, a_1, c_1, t] = [\frac{244879}{348216}, \frac{24553}{126624}, \frac{269429}{296780}, -\frac{7181}{15620}, \frac{45739}{287337}, -\frac{2591}{15123}, -\frac{941}{8520}, \frac{6541}{8520}, \frac{87054}{95779}].$$

For $h_1$ and $j_1$, one may just choose $h_1 = j_1 = 0$.

We can also select other numbers for the parameters such that $q^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment order 4 ($p$ is not the filter given by (32)), but the resulting $\phi, \widetilde{\phi}$ cannot have nice smoothness. For example, we can select the parameters such that $p$ and $\widetilde{p}$ have sum rule orders 2 and 4, $q^{(\ell)}, 1 \le \ell \le 4$ and $\widetilde{q}^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment orders 4 and 2, respectively, and $\phi \in W^{0.00049}, \widetilde{\phi} \in W^{1.84163}$. We are unable to obtain $\widetilde{\phi} \in W^2$ with $\phi \in L^2(\mathbb{R}^2)$, $q^{(\ell)}, 1 \le \ell \le 4$ and $\widetilde{q}^{(\ell)}, 1 \le \ell \le 4$ having vanishing moment orders 4 and 2, respectively.

If we drop the conditions for the vanishing moment of $\widetilde{q}^{(\ell)}, 1 \le \ell \le 4$, then we can choose the parameters such that $q^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment order 4 and both $\phi$ and $\widetilde{\phi}$ are $C^2$. For example, if

$$[b, d, n, a, c, h, j, w, n_1, a_1, c_1, t] = [\frac{11}{8}, \frac{71}{616}, \frac{1601}{7854}, \frac{3297}{1564}, \frac{385}{782}, -\frac{10269}{4048}, \frac{105}{368}, \frac{17}{154}, \frac{2}{63}, -\frac{993}{1496}, \frac{117}{1496}, 1],$$

then $q^{(\ell)}, 1 \le \ell \le 4$ have vanishing moment order 4, $\widetilde{\phi} \in W^{3.19573}$ and $\phi$ is the $C^2$ box-spline associated with the direction set $\Theta_1$ in (33).

# 8 Multiresolution algorithms for extraordinary vertices and boundary vertices

To apply the above constructed PR FIR frame filter banks to a (high-resolution) triangular mesh with extraordinary vertices, we need to design the algorithms for extraordinary vertices. A vertex in a triangular mesh is called an extraordinary vertex if its valence is not 6. (The valence of a vertex $v$ is the number of the edges that meet at $v$.) In this section we mainly consider Loop's scheme-based frame filter banks. First, we recall that Loop's scheme for an extraordinary vertex $\widetilde{v}$ with valence $k$ can be written as (refer to [2])

$$\begin{array}{l} e = \frac{3}{8}(\widetilde{v}_0 + \widetilde{v}_1) + \frac{1}{8}(\widetilde{v}_2 + \widetilde{v}_3), (\forall e) \\ v = \beta(k)\widetilde{v} + \gamma(k) \sum_{j=0}^{k-1} e_j, \end{array}$$

where $e$ denotes an edge vertex (new inserted vertex) in the finer mesh, $\widetilde{v}_s, s = 0, 1, 2, 3$ are four old vertices in the coarse mesh surrounding $e$, and $v$ is the vertex in the finer mesh which replaces $\widetilde{v}$, and

$$\beta(k) = \frac{8}{5}\Big(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\Big)^2, \quad \gamma(k) = \frac{1}{k}\big(1 - \beta(k)\big). \tag{51}$$

Next we provide a 2-step algorithms for extraordinary vertices with Loop-$\mathrm{F}_{1,2}$ used for regular vertices. The decomposition and reconstruction algorithm are given in (52)(53) and (54)(55), where $k$ is the valence of the extraordinary vertex $v$, and $e_0, e_1, \cdots, e_{k-1}$ are $k$ (edge) vertices

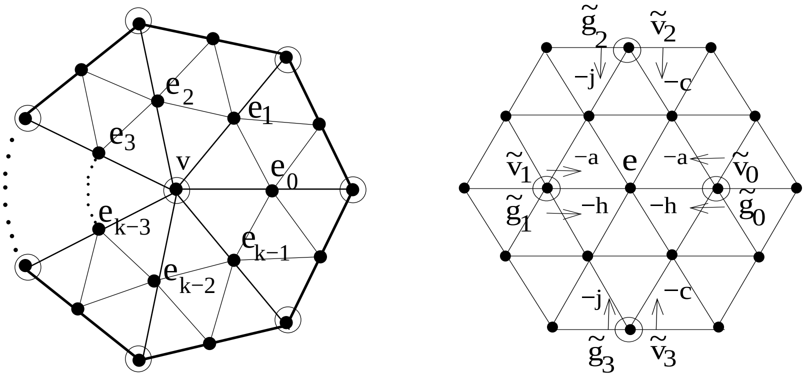

Figure 11: *Left: Template to obtain $\widetilde{v}$ or $\widetilde{g}$ in 2-step Decomposition Alg. Step 1; Right: Decomposition Alg. Step 2*

surrounding $v$. See Fig. 11 for the templates of the decomposition algorithm. The parameters $a, c, h, j, t$ are the same numbers in Loop-F$_{1,2}$, and $b(k), d(k), n(k)$ are numbers to be determined.

**2-step Dyadic Frame Decomposition Algorithm for Extraordinary Vertices:**

Step 1.$\widetilde{v} = \dfrac{1}{b(k)}\Big\{v - d(k)\displaystyle\sum_{j=0}^{k-1} e_j\Big\}, \quad \widetilde{g} = v - n(k)\displaystyle\sum_{j=0}^{k-1} e_j$ (52)

Step 2. $\widetilde{e} = e - a(\widetilde{v}_0 + \widetilde{v}_1) - c(\widetilde{v}_2 + \widetilde{v}_3) - h(\widetilde{g}_0 + \widetilde{g}_1) - j(\widetilde{g}_2 + \widetilde{g}_3).$ (53)

**2-step Dyadic Frame Reconstruction Algorithm for Extraordinary Vertices:**

Step 1. $e = \widetilde{e} + a(\widetilde{v}_0 + \widetilde{v}_1) + c(\widetilde{v}_2 + \widetilde{v}_3) + h(\widetilde{g}_0 + \widetilde{g}_1) + j(\widetilde{g}_2 + \widetilde{g}_3)$ (54)

Step 2. $v = t\Big\{b(k)\widetilde{v} + d(k)\sum_{j=0}^{k-1} e_j\Big\} + (1-t)\Big\{\widetilde{g} + n(k)\sum_{j=0}^{k-1} e_j\Big\}.$ (55)

Setting the "details" $\widetilde{g} = 0, \widetilde{e} = 0$, we reduce the reconstruction algorithm (54)(55) to the subdivision algorithm:

$$e = a(\widetilde{v}_0 + \widetilde{v}_1) + c(\widetilde{v}_2 + \widetilde{v}_3),\ v = tb(k)\widetilde{v} + \Big\{td(k) + (1-t)n(k)\Big\}\sum_{j=0}^{k-1} e_j. \tag{56}$$

When $a = 3/8, c = 1/8, tb(k) = \beta(k), td(k) + (1-t)n(k) = \gamma(k)$, this subdivision scheme is Loop's scheme. For $n(k)$, we choose $n(k) = \frac{1}{k}$ so that highpass outputs are zero if the input $v \equiv 1, e \equiv 1$. To summarize, the parameters selected are

$$a = \frac{3}{8},\ c = \frac{1}{8},\ h = \frac{33}{32},\ j = \frac{3}{32},\ t = \frac{1}{10}, b(k) = 10\beta(k), d(k) = \frac{1}{k}\big(1 - b(k)\big),\ n(k) = \frac{1}{k}. \tag{57}$$

Clearly, when $k = 6$, the above algorithm is the 2-step algorithm with Loop-F$_{1,2}$.

Corresponding to the 3-step algorithm (22)-(27) in §5, the algorithm for extraordinary vertices is given by (58)-(63) (refer to Fig. 12 for decomposition algorithm templates), where $a, c, h, j, t$ are the same numbers in (22)-(27).

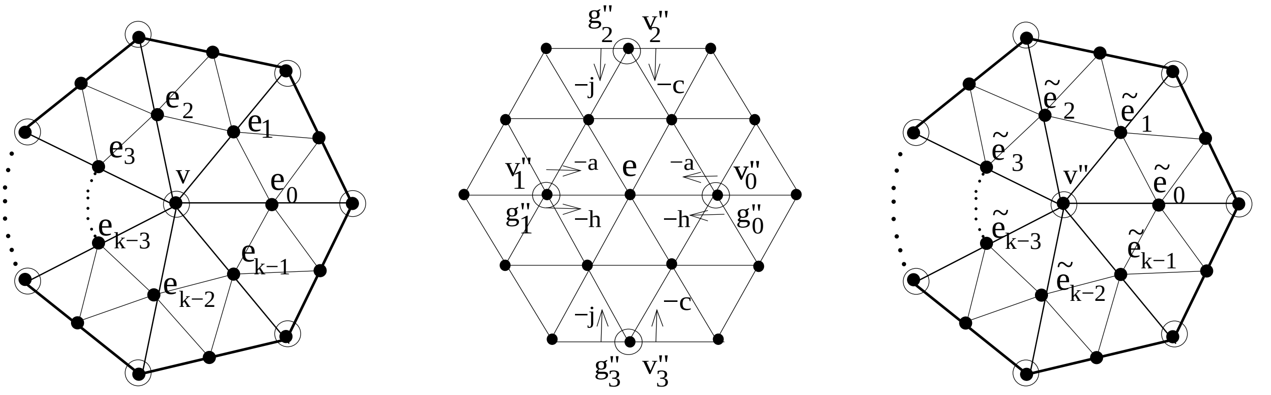

Figure 12: *Left: Template to obtain $v''$ or $g''$ in Decomposition Alg. Step 1; Middle: Decomposition Alg. Step 2; Right: Template to obtain lowpass output $\widetilde{v}$ in Decomposition Alg. Step 3*

**3-step Dyadic Frame Decomposition Algorithm for Extraordinary Vertices:**

$$\text{Step 1.} v'' = \frac{1}{b(k)}\Big\{v - d(k)\sum_{j=0}^{k-1} e_j\Big\}, \quad g'' = v - n(k)\sum_{j=0}^{k-1} e_j \tag{58}$$

$$\text{Step 2. } \widetilde{e} = e - a(v_0'' + v_1'') - c(v_2'' + v_3'') - h(g_0'' + g_1'') - j(g_2'' + g_3'') \tag{59}$$

$$\text{Step 3.} \widetilde{v} = v'' - w(k)\sum_{j=0}^{k-1} \widetilde{e}_j, \ \widetilde{g} = g'' - n_1(k)\sum_{j=0}^{k-1} \widetilde{e}_j. \tag{60}$$

**3-step Dyadic Frame Reconstruction Algorithm for Extraordinary Vertices:**

$$\text{Step 1. } v'' = \widetilde{v} + w(k)\textstyle\sum_{j=0}^{k-1} \widetilde{e}_j, \ g'' = \widetilde{g} + n_1(k)\sum_{j=0}^{k-1} \widetilde{e}_j \tag{61}$$

$$\text{Step 2. } e = \widetilde{e} + a(v_0'' + v_1'') + c(v_2'' + v_3'') + h(g_0'' + g_1'') + j(g_2'' + g_3'') \tag{62}$$

$$\text{Step 3. } v = t\Big\{b(k)v'' + d(k)\sum_{j=0}^{k-1} e_j\Big\} + (1-t)\Big\{g'' + n(k)\sum_{j=0}^{k-1} e_j\Big\}. \tag{63}$$

When "details" $\widetilde{e}, \widetilde{g}$ in (61)(62) are set to zero, the above reconstruction (61)-(63) is reduced to the subdivision algorithm (56). Again, when $a = 3/8, c = 1/8, tb(k) = \beta(k), td(k) + (1-t)n(k) = \gamma(k)$, this subdivision scheme is Loop's scheme. For $n(k)$, we choose $n(k) = \frac{1}{k}$ so that the resulting analysis highpass frame filters annihilate the constant. For $w(k), n_1(k)$, we may simply select $w(k) = \frac{6w}{k}, n_1(k) = \frac{6n_1}{k}$, where $w, n_1$ are the numbers for the regular vertices. In the following we list the selected parameters with Loop-F$_{3,2}$ used for regular vertices:

$$\begin{array}{l} a = \frac{3}{8}, \ c = \frac{1}{8}, \ , h = \frac{23}{16}, \ j = -\frac{23}{16}, n_1 = 0, \ t = 1, \\ b(k) = \beta(k), \ d(k) = \frac{1}{k}\big(1 - b(k)\big), \ n(k) = \frac{1}{k}, \ w(k) = -\frac{9}{10k}, \ n_1(k) = 0. \end{array} \tag{64}$$

When the input (high-resolution) surface is an open mesh, we also need multiresolution algorithms for (interior) boundary vertices. Symmetric 1-D bi-frames are considered in [30] with the corresponding frame multiresolution algorithms also given by iterative templates. Those 1-D frame algorithms can be used as boundary algorithms. Here we are going to use the following

1-D frame algorithms from [30] for boundary vertices.

**3-step Frame Decomposition Algorithm for Boundary Vertices:**

$$\text{Step 1. } v'' = \tfrac{1}{b}\Big\{v - d(e_{-1}+e_0)\Big\},\ f'' = v - n(e_{-1}+e_0) \tag{65}$$

$$\text{Step 2. } \widetilde{e} = e - u(v_0''+v_1'') - w(f_0''+f_1'') \tag{66}$$

$$\text{Step 3. } \widetilde{v} = v'' - d_1(\widetilde{e}_{-1}+\widetilde{e}_0),\ \widetilde{f} = f'' - n_1(\widetilde{e}_{-1}+\widetilde{e}_0), \tag{67}$$

**3-step Frame Reconstruction Algorithm for Boundary Vertices:**

$$\text{Step 1. } v'' = \widetilde{v} + d_1(\widetilde{e}_{-1}+\widetilde{e}_0),\ f'' = \widetilde{f} + n_1(\widetilde{e}_{-1}+\widetilde{e}_0) \tag{68}$$

$$\text{Step 2. } e = \widetilde{e} + u(v_0''+v_1'') + w(f_0''+f_1'') \tag{69}$$

$$\text{Step 3. } v = t\Big\{bv'' + d(e_{-1}+e_0)\Big\} + (1-t)\Big\{v'' + n(e_{-1}+e_0)\Big\}, \tag{70}$$

where

$$[b,\ d,\ n,\ u,\ w,\ d_1,\ n_1,\ t] = [\frac{4}{3},\ -\frac{1}{6},\ \frac{1}{2},\ \frac{1}{2},\ -\frac{5}{24},\ -\frac{3}{8},\ \frac{3}{10},\ \frac{3}{8}].$$

Refer to Fig. 13 and Fig. 14 for these boundary algorithms. The corresponding $\widetilde{\phi}(x)$ is the $C^2$ cubic B-spline supported on [-2, 2], and $\phi(x) \in W^{1.82037}(\mathbb{R})$.

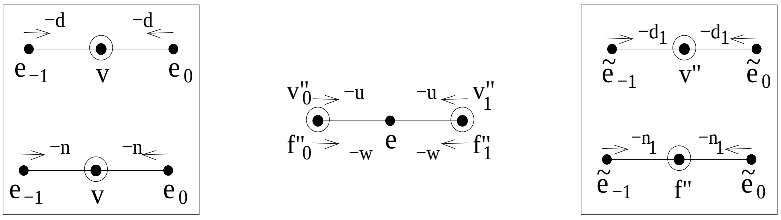

Figure 13: *Left: Decomposition Step 1; Middle: Decomposition Step 2; Right: Decomposition Step 3*

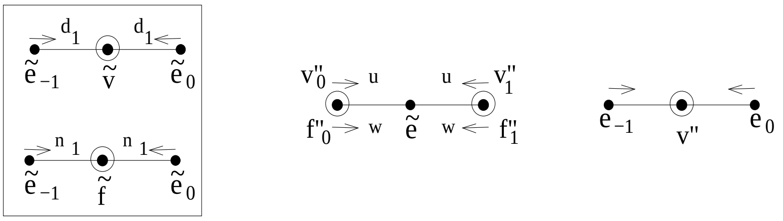

Figure 14: *Left: Reconstruction Step 1; Middle: Reconstruction Step 2; Right: Reconstruction Step 3*

If $d_1 = n_1 = 0$, then the above 1-D frame algorithm is a 2-step algorithm. If we choose

$$[b,\ d,\ n,\ u,\ w,\ d_1,\ n_1,\ t] = [2,\ -\frac{1}{2},\ \frac{1}{2},\ \frac{1}{2},\ \frac{3}{4},\ 0,\ 0,\ \frac{1}{4}],$$

then the corresponding $\widetilde{\phi}(x)$ is the $C^2$ cubic B-spline supported on [-2, 2], and $\phi(x)$ is the continuous linear B-spline supported on [-1, 1].

Our highly symmetric frame algorithms can be applied immediately for some applications such as mesh-based surface sparse representation, noise removal, compression, progressive transmission,

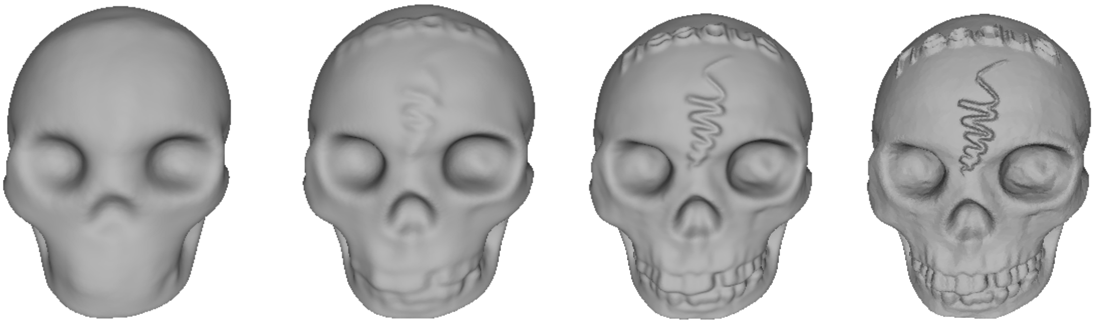

Figure 15: *From left to right: Low resolution to high resolution surfaces*

etc. Here we just show some preliminary results. To apply multiresolution algorithms to a mesh-based surface, it is required that the input mesh should have a semi-regular structure (a subdivision connectivity). One could use MAPs or other methods (see for example [25, 36]) to get a surface that has a subdivision connectivity and closely approximates (with guaranteed errors) to the original (non semi-regular) surface. Lowpass outputs ("approximations") with different decomposition levels of a high resolution surface can be used for surface progressive transmission. In Fig. 15 we show 3- 2-, and 1-level approximations (from the left) of a high resolution surface (on the right) with the above Loop-$F_{1,2}$ frame algorithm. The frame algorithms can also be used for surface sparse representation. On top-left of Fig. 16 is a cortical surface. The top-middle and top-right are the reconstructed surfaces when we use 10% and 5% lowpass and highpass coefficients after the original surface is decomposed by the above Loop-$F_{1,2}$ frame algorithm. (We use all coefficients of 3-level lowpass output and 8.44% and 3.44% coefficients of highpass outputs respectively.) The bottom (from left) of Fig. 16 are the approximations after 1-, 2- and 3-level decompositions.

The frame algorithms could be used for surface noise removal. We show denoised surfaces in the left column of Fig. 17, where the original surfaces and the surfaces with noise are in the right and middle columns respectively. We have applied a few iterated steps of soft threshold denoising (see [18]) using the above Loop-$F_{1,2}$ frame algorithm. The de-speckle procedure (refer to [51]) was also applied. Here we remark that the total variation-based surface denoising model has been developed in [22] (the reader refers to [22] for other methods for surface denoising). Surface denoising is significantly different than 2-D image denoising. For surface denoising, [22] applied the important work of geometers who were interested in extending certain classical theorems from smooth to polyhedral manifolds. When frame algorithms are applied for surface denoising, and other applications, the wide variety of methods and techniques of 2-D image multiscale processing could be used. In addition, the frame multiscale algorithm is fast.

**Acknowledgment.** The authors thank two anonymous referees for their valuable suggestions and comments. The models used in Fig. 15 and Fig. 17 were provided by Igor Guskov (originally from Headus and Cyberware respectively) and the cortical surface courtesy INRIA Gamma team Research Database.

# References

[1] J.J. Benedetto and S.D. Li, The theory of multiresolution analysis frames and applications to filter banks, *Appl. Comput. Harmon. Anal.*, 5 (1998), 389–427.

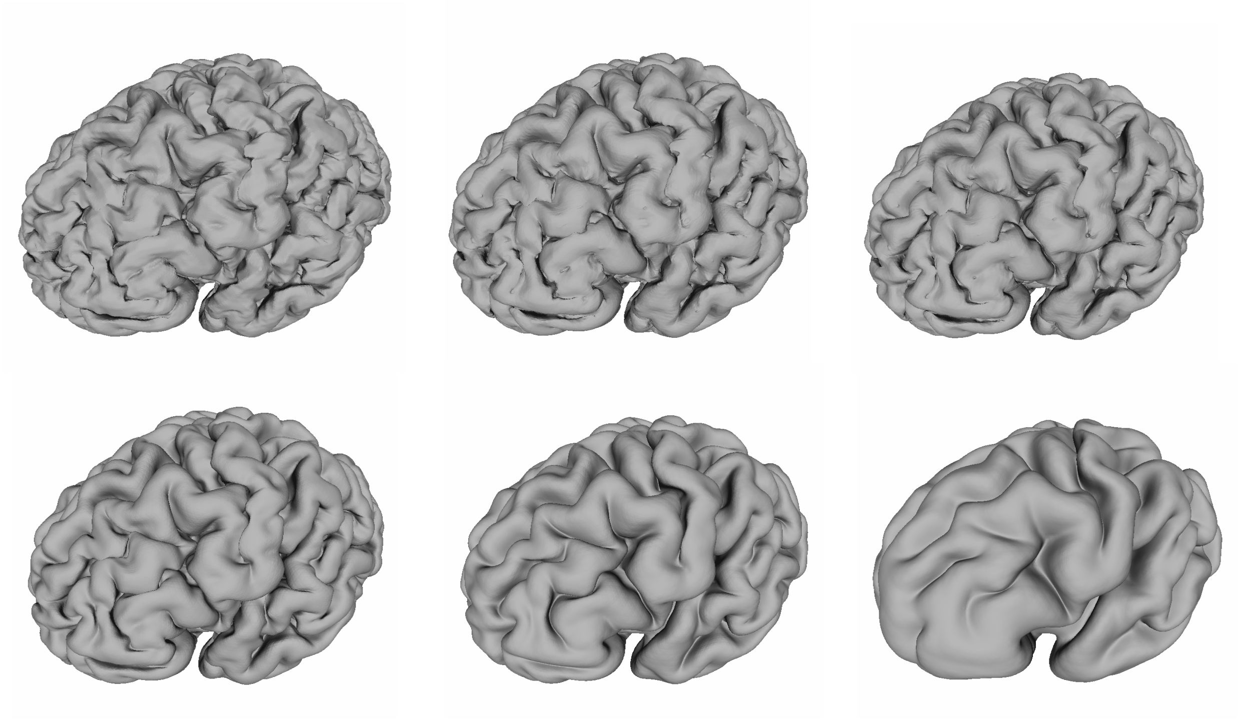

Figure 16: *Top-left: Original cortical surface, Top-middle: reconstructed surface with* 10% *lowpass and highpass coefficients after 3-level decompositions, Top-right: reconstructed surface with* 5% *lowpass and highpass coefficients after 3-level decompositions; Bottom-row (from left): 1-level, 2-level and 3-level approximations*

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

[3] 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.

[4] C. de Boor, K. Höllig, and S.D. Riemenschneider, *Box Splines*, Springer-Verlag, New York, 1993.

[5] J.F. Cai, R.H. Chan, L.X. Shen, and Z.W. Shen, Restoration of chopped and nodded images by framelets, *SIAM J. Sci. Comput.*, 24 (2008), 1205–1227.

[6] J.F. Cai, R.H. Chan, and Z.W. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmon. Anal.*, 25 (2008), 131–149.

[7] J.F. Cai, S. Osher, and Z.W. Shen, Split Bregman methods and frame based image restoration, *Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal*, 8 (2009), 337-369.

[8] P.G. Casazza, The art of frame theory, *Taiwanese J. Math.*, 4 (2000), 129–201.

[9] R.H. Chan, T.F. Chan, L.X. Shen, and Z.W. Shen, Wavelet algorithms for high-resolution image reconstruction, *SIAM J. Sci. Comput.*, 24 (2003), 1408–1432.

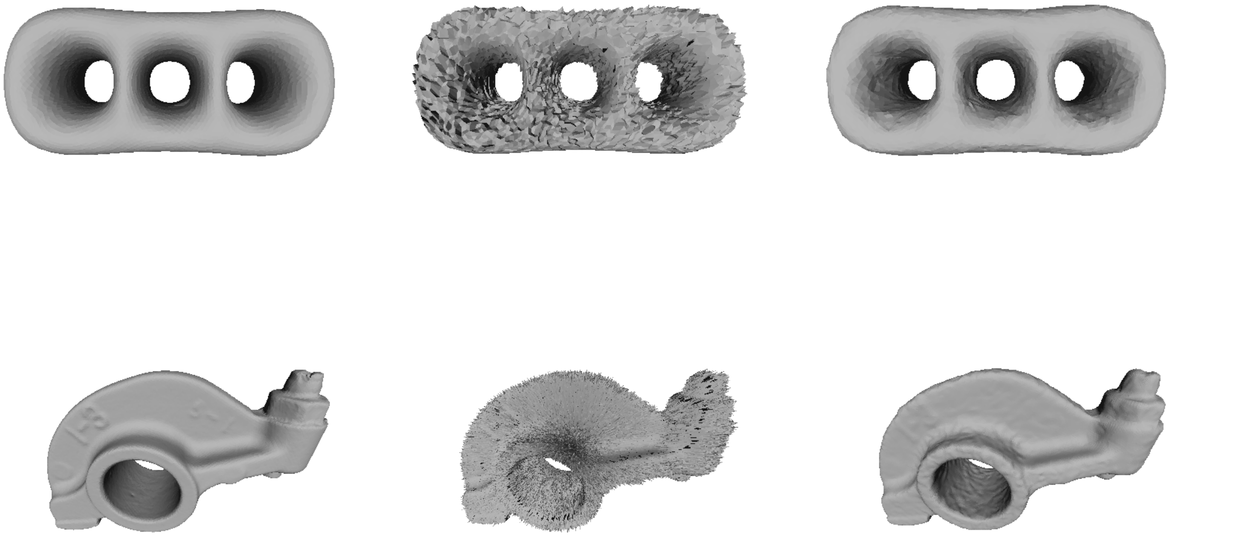

Figure 17: *Left column: Original surfaces; Middle column: Surfaces with noise; Right column: Denoised surfaces*

[10] R.H. Chan, S.D. Riemenschneider, L.X. Shen, and Z.W. Shen, Tight frame: An efficient way for high-resolution image reconstruction, *Appl. Comput. Harmon. Anal.*, 17 (2004), 91–115.

[11] M. Charina and J. Stöckler, Tight wavelet frames for subdivision, *J. Comput. Appl. Math.*, 221 (2008), 293–301.

[12] M. Charina and J. Stöckler, Tight wavelet frames for irregular multiresolution analysis, *Appl. Comput. Harmon. Anal.*, 25 (2008), 98–113.

[13] A. Chebira, *Adaptive Multiresolution Frame Classification of Biomedical Images*, Ph.D. Dissertation, Carnegie Mellon University, 2008.

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

[15] W. Dahmen, Decomposition of refinable spaces and applications to operator equations, *Numer. Algor.*, 5 (1993), 229–245.

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

[17] 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.

[18] D. Donoho, Denoising by soft thresholding, *IEEE Trans. Inform. Theory*, 41 (1995), 613–627.

[19] N. Dyn, J.A. Gregory, and D. Levin, A butterfly subdivision scheme for surface interpolation with tension control, *ACM Trans. Graphics*, 2 (1990), 160–169.

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

[21] M. Ehler and B. Han, Wavelet bi-frames with few generators from multivariate refinable functions, *Appl. Comput. Harmon. Anal.*, 25 (2008), 407–414.

[22] M. Elsey and S. Esedoglu, Analogue of the total variation denoising model in the context of geometry processing, *SIAM J. Multiscale Modeling and Simulation*, 7 (2009), 1549–1573.

[23] K. Gröchenig, *Foundations of Time-Frequency Analysis*, Birkhäuser, Boston, 2001.

[24] K. Gröchenig and A. Ron, Tight compactly supported wavelet frames of arbitrary high smoothness, *Proc. Amer. Math. Soc.*, 126 (1998), 1101–1107.

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

[26] B. Han and Z.W. Shen, Wavelets from the Loop scheme, *J. Fourier Anal. Appl.*, 11 (2005), 615–637.

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

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

[29] Q.T. Jiang, Hexagonal tight frame filter banks with idealized high-pass filters, *Advances in Comput. Math.*, 31 (2009), 215–236.

[30] Q.T. Jiang, Wavelet bi-frames with uniform symmetry for curve multiresolution processing, *J. Comput. Appl. Math.*, 235 (2011), 1653–1675.

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

[32] Q.T. Jiang, Biorthogonal wavelets with 6-fold axial symmetry for hexagonal data and triangle surface multiresolution processing, *International Journal of Wavelets, Multiresolution and Information Processing*, 9 (2011).

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

[34] A. Khodakovsky, P. Schröder, and W. Sweldens, Progressive geometry compression, *Proceedings of SIGGRAPH 2000.*

[35] M.-J. Lai and J. Stöckler, Construction of multivariate compactly supported tight wavelet frames, *Appl. Comput. Harmon. Anal.*, 21 (2006), 324-348.

[36] 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.

[37] C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, Master's Thesis, University of Utah, Department of Mathematics, Salt Lake City, 1987.

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

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

[40] J. Romero, S. Alexander, S. Baid, S. Jain and M. Papadakis, The geometry and the analytic properties of isotropic multiresolution analysis, *Advances in Comput. Math.*, 31 (2009), 283–328.

[41] 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.

[42] 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.

[43] A. Ron and Z.W. Shen, Compactly supported tight affine spline frames in $L_2(\mathbb{R}^d)$, *Math. Comput.*, 67 (1998), 191–207.

[44] A. Ron and Z.W. Shen, Construction of compactly supported affine spline frames in $L_2(\mathbb{R}^d)$, in *Advances in Wavelets*, (K. S. Lau ed.), Springer-Verlag, Singapore, 1998, pp. 27–49.

[45] 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.

[46] P. Schröder and W. Sweldens, Spherical wavelets: Efficiently representing functions on the sphere, *Proc. SIGGRAPH 95*, pp. 161–172, 1995.

[47] L.X. Shen, M. Papadakis, I.A. Kakadiaris, I. Konstantinidis, I. Kouri, and D. Hoffman, Image denoising using a tight frame, *IEEE Trans. Image Proc.*, 15 (2006), 1254–1263.

[48] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.*, 3 (1996), 186–200.

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

[50] 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.

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