Biorthogonal Wavelets with 6-fold Axial Symmetry for Hexagonal Data and Triangle Surface Multiresolution Processing

Qingtang Jiang *

Abstract

This paper discusses the construction of highly symmetric compactly supported wavelets for hexagonal data/image and triangle surface multiresolution processing. Recently hexagonal image processing has attracted attention. Compared with the conventional square lattice, the hexagonal lattice has several advantages, including that it has higher symmetry. It is desirable that the filter banks for hexagonal data also have high symmetry which is pertinent to the symmetric structure of the hexagonal lattice. The high symmetry of filter banks and wavelets not only leads to simpler algorithms and efficient computations, it also has the potential application for the texture segmentation of hexagonal data. While in the field of CAGD, when the filter banks are used for surface multiresolution processing, it is required that the corresponding decomposition and reconstruction algorithms for regular vertices have high symmetry, which make it possible to design the corresponding multiresolution algorithms for extraordinary vertices.

In this paper we study the construction of 6-fold symmetric biorthogonal filter banks and the associated wavelets, with both the dyadic and $\sqrt{3}$ refinements. The constructed filter banks have the desirable symmetry for hexagonal data processing. By associating the outputs (after 1-level multiresolution decomposition) appropriately with the nodes of the regular triangular mesh with which the input data is associated (sampled), we represent multiresolution analysis and synthesis algorithms as templates. The 6-fold symmetric filter banks constructed in this paper result in algorithm templates with desirable symmetry for triangle surface processing.

Key words and phrases: Hexagonal lattice, hexagonal data, 6-fold symmetric filter bank, biorthogonal hexagonal filter bank, biorthogonal dyadic refinement wavelet, biorthogonal $\sqrt{3}$-refinement wavelet, surface multiresolution decomposition/reconstruction.

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

1 Introduction

This paper is motivated by the desire of highly symmetric compactly supported wavelets for hexagonal data/image processing and the requirement of such highly symmetric wavelets for triangle surface multiresolution processing. Recently hexagonal data/image processing has attracted attention. Compared with the square lattice, the conventionally used lattice for image sampling and processing, the hexagonal lattice has several advantages, and hence it has been used in many areas, see e.g. [32, 28, 14, 42, 35] and the references therein. One of the advantages the hexagonal lattice possesses is that it has 6-fold line symmetry while a square lattice possesses 4-fold line symmetry. See a square and a hexagonal lattices in the left and middle pictures of Fig. 1, and the 6 symmetric lines (axes) S_j in the right picture of Fig. 1. Since the hexagonal lattice has high symmetry, it is desirable that the filter banks along it also have 6-fold symmetry. The lowpass filters considered in [38, 43] have 6-fold

*Q. Jiang is with the Department of Mathematics and Computer Science, University of Missouri–St. Louis, St. Louis, MO 63121, USA, e-mail: jiangq@umsl.edu , web: http://www.math.umsl.edu/∼jiang .

symmetry, but the filter banks constructed in these papers are not perfect reconstruction (biorthogonal) filter banks. The filter banks constructed in [9, 1, 2, 19] are orthogonal or biorthogonal filter banks, but they have only 3-fold symmetry. In this paper we discuss the construction of compactly supported biorthogonal wavelets with 6-fold axial symmetry. The high symmetry of filter banks and wavelets not only leads to simpler algorithms and efficient computations, it also has the potential application for the texture segmentation of hexagonal data. The reader refers to [11, 31, 34] for the application of isotropic wavelets which are rotation invariant to medical data texture segmentation.

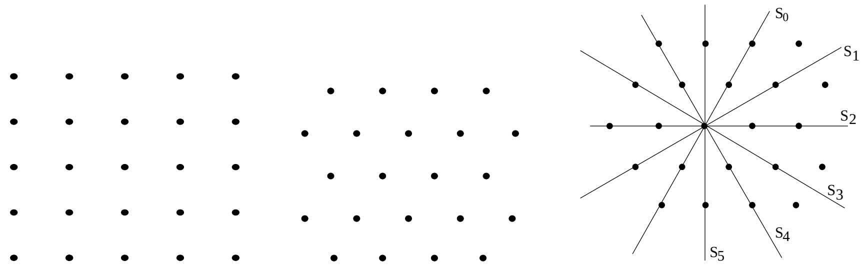

Figure 1: *Left: Square lattice; Middle: Hexagonal lattice; Right: 6 symmetric axes (lines)*

In the field of CAGD, the construction of wavelets for surface multiresolution processing has been proposed for more than one decade [26, 27], where linear spline and butterfly-scheme related semi-orthogonal wavelets are considered. Since then researchers have made efforts to construct other wavelets. For example, Doo's subdivision scheme based wavelets for quadrilateral surfaces are constructed in [36], Loop's scheme based wavelets for triangle surfaces are presented in [3].

For surface multiresolution reconstruction, the input is some "details" and an "approximation", a coarse-resolution triangle or quadrilateral mesh with 3D vertices, while for surface multiresolution decomposition, the input is a fine-resolution triangle or quadrilateral mesh also with 3D vertices. In either case, the coarse- or fine-resolution mesh in general contains extraordinary vertices whose valences are not 6 for triangle mesh and are not 4 for quadrilateral mesh. This requires the decomposition and reconstruction algorithms have highly symmetry.

When we set the "details" to be zero, the multiresolution reconstruction (or called wavelet synthesis) is reduced to be the subdivision algorithm, which is an efficient method to generate smooth surfaces with an arbitrary topology and has been successfully used in animation movie production and other fields [37, 41]. The symmetry of a subdivision algorithm is required. For example, the triangle subdivision mask for the regular vertex, which is the lowpass filter of the synthesis filter bank, must be symmetric around S_j, $0 \le j \le 5$, or equivalently, the corresponding scaling function (also called the basis function) is symmetric around S_j, $0 \le j \le 5$ (the 3-direction box-splines with such a symmetry are called to have the full set of symmetries in [4]).

For surface multiresolution processing, which also involves the analysis lowpass filter, highpass filters and wavelets, there is not much work on the symmetry the wavelets and highpass filters should have. In this paper we introduce the 6-fold axial symmetry of biorthogonal filter banks, including both lowpass and highpass filters. The filter banks with such a symmetry not only are desirable for hexagonal image processing, but also can be used for triangle surface multiresolution processing.

Using the idea of lifting scheme [39, 10], paper [3] introduces a novel surface multiresolution algorithms which work for both regular and extraordinary vertices. When considering the biorthogonality, [3] does not use the conventional $L^2(\mathbb{R}^2)$ inner product. Instead, it uses a "discrete inner product" related to the discrete filters. Thus, [3] does not consider (and it is not unnecessary to consider) the lowpass filter, highpass filters, scaling function and wavelets, and hence it does not discuss their symmetry. However, this method may result in scaling functions and wavelets which are not in $L^2(\mathbb{R}^2)$. Indeed, the analysis scaling function and wavelets (associated with regular vertices) constructed in [3] are not in $L^2(\mathbb{R}^2)$, and hence they cannot generate biorthogonal (Riesz) bases for

$L^2(\mathbb{R}^2)$. In this paper, we consider $L^2(\mathbb{R}^2)$ inner product and the wavelets constructed here generate biorthogonal (Riesz) bases.

The dyadic (4-size) refinement (or 1-to-4 split subdivision) is the most commonly used refinement for multiresolution image processing and for surface subdivision. The hexagonal lattice also allows the $\sqrt{3}$ (3-size) refinement [12, 5], and in GAGD $\sqrt{3}$ (triangle) subdivision has been studied by researchers [13, 23, 24, 21, 22, 30]. Compared with the dyadic refinement, the $\sqrt{3}$-refinement generates more resolutions and, hence, gives applications more resolutions from which to choose. This fact is the main motivation for the study of $\sqrt{3}$-subdivision and for the recommendation of the $\sqrt{3}$-refinement for discrete global grid systems in [35]. The $\sqrt{3}$-refinement has also been used by engineers and scientists of the PYXIS innovation Inc. to develop The PYXIS Digital Earth Reference Model [33]. The method in [3] for the construction of dyadic wavelets was adopted in [40] to construct $\sqrt{3}$-refinement biorthogonal wavelets. As in [3], a "discrete inner product" related to the discrete filters is used in [40] to define the biorthogonality of the wavelets. Again, that discrete inner product may result in scaling functions and wavelets not in $L^2(\mathbb{R}^2)$. In [20], $\sqrt{3}$-refinement wavelets and filter banks with 6-fold symmetry are studied. In this paper we show that 6-fold symmetric $\sqrt{3}$ filter banks result in very simple decomposition and reconstruction algorithms given by templates. These algorithms include those (for regular vertices) constructed in [40].

The rest of this paper is organized as follows. In Section 2, we recall multiresolution decomposition and reconstruction algorithms and present some basic results on the biorthogonality of filter banks. Section 3 and Section 4 are about the dyadic and $\sqrt{3}$ refinement wavelets respectively. The characterization of 6-fold symmetric filter banks, the symmetry of the associated scaling functions and wavelets, and the construction of biorthogonal filter banks with 6-fold symmetry and the associated wavelets are studied in theses two sections. We also discuss how to represent decomposition and reconstruction algorithms as templates for surface processing in this two sections. The relationship between the algorithms for regular vertices in [3, 40] and the 6-fold symmetric filter banks in this paper is also presented.

In this paper we use bold-faced letters such as $\mathbf{k}, \mathbf{x}, \boldsymbol{\omega}$ to denote elements of $\mathbf{Z}^2$ and $\mathbb{R}^2$. A multi-index $\mathbf{k}$ of $\mathbf{Z}^2$ and a point $\mathbf{x}$ in $\mathbb{R}^2$ will be written as row vectors

$$\mathbf{k} = (k_1, k_2), \ \mathbf{x} = (x_1, x_2).$$

However, $\mathbf{k}$ and $\mathbf{x}$ should be understood as column vectors $[k_1, k_2]^T$ and $[x_1, x_2]^T$ when we consider $A\mathbf{k}$ and $A\mathbf{x}$, where A is a 2×2 matrix. For a matrix M, we use M^* to denote its conjugate transpose $\overline{M^T}$, and for a nonsingular matrix M, M^{-T} denotes $(M^{-1})^T$.

2 Multiresolution Algorithms

In this section, we recall multiresolution decomposition and reconstruction algorithms and present some basic results on the biorthogonality of filter banks.

Let $\mathcal{G}$ denote the regular unit hexagonal lattice defined by

$$\mathcal{G} = \{k_1\mathbf{v}_1 + k_2\mathbf{v}_2 : \quad k_1, k_2 \in \mathbf{Z}\}, \tag{2.1}$$

where

$$\mathbf{v}_1 = [1, 0]^T, \ \mathbf{v}_2 = [-\frac{1}{2}, \frac{\sqrt{3}}{2}]^T.$$

A regular hexagonal lattice has several appealing refinements, including the dyadic and $\sqrt{3}$ refinements (see [12, 5]).

In the left part of Fig. 2, the nodes with circles $\bigcirc$ form a new coarse hexagonal lattice, which is called the 4-size (4-branch, or 4-aperture) sublattice of $\mathcal{G}$, and is denoted by $\mathcal{G}_4$ here. From $\mathcal{G}$ to $\mathcal{G}_4$, the nodes are reduced by a factor $\frac{1}{4}$. Repeating this process, we have a set of regular lattices which forms a "pyramid". Connecting each node of circle $\bigcirc$ in $\mathcal{G}_4$ to its 6 adjacent nodes of $\bigcirc$, we have a mesh (grid) consisting of triangles, see the left part of Fig. 2. Each of other nodes of $\mathcal{G}$ is the middle of an edge of the triangle with 3 vertices of nodes $\bigcirc$ in $\mathcal{G}_4$. We use v to denote a node $\bigcirc$ in $\mathcal{G}_4$ and let e denote a node on the edge of a triangle, namely, e is a node in $\mathcal{G}$ but not in $\mathcal{G}_4$, see the left part of Fig. 2. v is called a *type V* **node** or a **vertex node** and e is called a *type E* **node** or an **edge node**.

In the right part of Fig. 2, the nodes with circles $\bigcirc$ form another new coarse lattice, which are called the 3-size (3-branch, or 3-aperture) sublattice of $\mathcal{G}$ here, and it is denoted by $\mathcal{G}_3$ here. From $\mathcal{G}$ to $\mathcal{G}_3$, the nodes are reduced by a factor $\frac{1}{3}$. Again, repeating this process, we have a set of regular lattices which forms a "pyramid". Connecting each node of circle $\bigcirc$ in $\mathcal{G}_3$ to its 6 adjacent nodes of $\bigcirc$, we have a mesh consisting of triangles, see the right part of Fig. 2. Each of other nodes of $\mathcal{G}$ is the centroid of the triangle with 3 vertices of nodes $\bigcirc$ in $\mathcal{G}_3$. In this case, we also use v to denote a node $\bigcirc$ in $\mathcal{G}_3$, but we use f to denote a node on the centroid of a triangle, namely, f is a node in $\mathcal{G}$ but not in $\mathcal{G}_3$, see the right of Fig. 2. v is also called a *type V* **node** or **vertex node** and f a *type F* **node** or a **face node**.

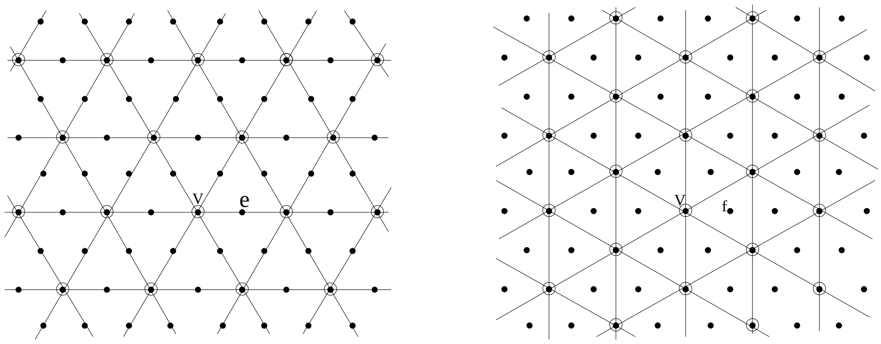

Figure 2: *Left: Coarse hexagonal lattice $\mathcal{G}_4$ with nodes v; Right: Coarse hexagonal lattice $\mathcal{G}_3$ with nodes v*

To a node $\mathbf{g} = k_1\mathbf{v}_1 + k_2\mathbf{v}_2$ of $\mathcal{G}$ defined by (2.1), we use (k_1, k_2) to indicate $\mathbf{g}$, see the left part of Fig. 3 for the labelling of $\mathcal{G}$. Thus, for hexagonal data $\mathcal{C}$ sampled on $\mathcal{G}$, instead of using $c_\mathbf{g}$, we use c_{k_1,k_2} to denote the values of $\mathcal{C}$ at $\mathbf{g} = k_1\mathbf{v}_1 + k_2\mathbf{v}_2$. (Here and below, for triangle surface multiresolution processing, $\mathcal{C}$ is a given triangle mesh and c_{k_1,k_2} are the 3-D vertices on $\mathcal{C}$ or one components of the 3-D vertices on $\mathcal{C}$.) Therefore, we write $\mathcal{C}$, data hexagonally sampled on $\mathcal{G}$, as $\mathcal{C} = \{c_{k_1,k_2}\}_{k_1,k_2\in\mathbf{Z}}$, see the right part of Fig. 3 for c_{k_1,k_2}.

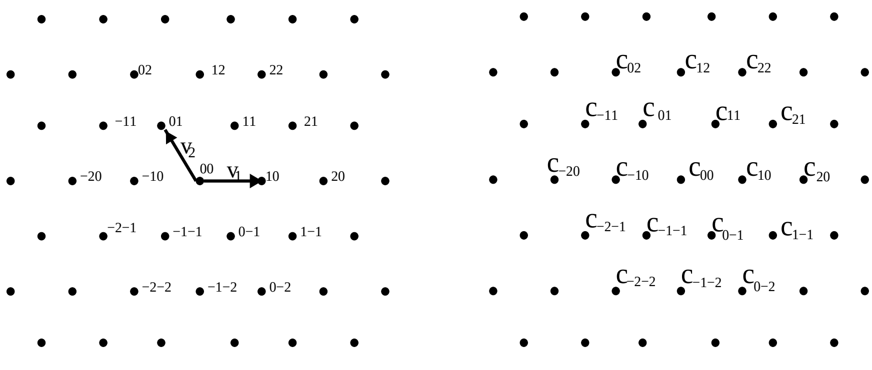

Figure 3: *Left: Indices for hexagonal nodes; Right: Indices for hexagonally sampled data $\mathcal{C}$*

To provide the multiresolution algorithms, first we need to choose a dilation matrix which maps $\mathcal{G}$ onto its sublattice $\mathcal{G}_4$ ($\mathcal{G}_3$ for $\sqrt{3}$-refinement). The labels for $\mathcal{G}_4$ are $(2k_1, 2k_2)$, while those for $\mathcal{G}_3$

are $(2k_1-k_2, k_1+k_2)$. Thus, for the dyadic refinement, $M = 2I_2$. For $\sqrt{3}$-refinement, we may choose M to be the following matrix (refer to [7, 20] for other choices of M):

$$M_1 = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}. \tag{2.2}$$

Denote

$$m = |\det(M)|.$$

Namely, for the dyadic refinement, $m = 4$, and for the $\sqrt{3}$-refinement, $m = 3$.

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 $1/m$ is multiplied):

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

When $\mathbf{k}, \mathbf{k} \in \mathbf{Z}^2$, are considered as indices for nodes $\mathbf{g} = k_1\mathbf{v}_1 + k_2\mathbf{v}_2$ of $\mathcal{G}$, $p(\boldsymbol{\omega})$ is a hexagonal filter, see Fig. 4 for the coefficients p_{k_1,k_2}. In this paper, a filter means a hexagonal filter though the indices of its coefficients are given by $\mathbf{k}$ with $\mathbf{k}$ in the square lattice $\mathbf{Z}^2$.

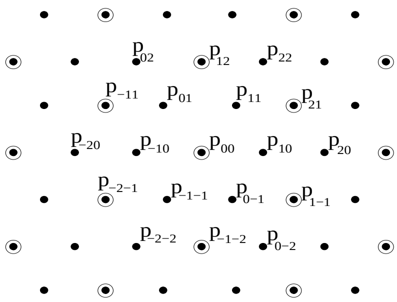

Figure 4: *Indices for impulse response coefficients* p_{k_1,k_2}

For a pair of filter banks $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$, the multiresolution decomposition algorithm with a dilation matrix M for an input hexagonally sampled image/data $\mathcal{C} = \{c^0_{\mathbf{k}}\}$ is

$$c^{j+1}_{\mathbf{n}} = \frac{1}{m}\sum_{\mathbf{k}\in\mathbf{Z}^2} p_{\mathbf{k}-M\mathbf{n}} c^j_{\mathbf{k}},\ d^{(\ell,j+1)}_{\mathbf{n}} = \frac{1}{m}\sum_{\mathbf{k}\in\mathbf{Z}^2} q^{(\ell)}_{\mathbf{k}-M\mathbf{n}} c^j_{\mathbf{k}}, \tag{2.3}$$

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

$$\hat{c}^j_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{p}_{\mathbf{k}-M\mathbf{n}} \hat{c}^{j+1}_{\mathbf{n}} + \sum_{1\le\ell\le m-1}\sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{q}^{(\ell)}_{\mathbf{k}-M\mathbf{n}} d^{(\ell,j+1)}_{\mathbf{n}} \tag{2.4}$$

with $\mathbf{k} \in \mathbf{Z}^2$ for $j = J-1, J-2, \cdots, 0$, where $\hat{c}_{\mathbf{n},J} = c_{\mathbf{n},J}$. We say hexagonal filter banks $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$ to be the **perfect reconstruction (PR) filter banks** if $\hat{c}^j_{\mathbf{k}} = c^j_{\mathbf{k}}$, $0 \le j \le J-1$ for any input hexagonally sampled image $c^0_{\mathbf{k}}$. $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ is called the **analysis filter bank** and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$ the **synthesis filter bank**. $\{c^j_{\mathbf{k}}\}$, $\{d^{(\ell,j)}_{\mathbf{k}}\}$ are called the "approximation" and the "details" of $\mathcal{C}$. When $d^{(\ell,j)}_{\mathbf{k}} = 0$, (2.4) is reduced to $\hat{c}^j_{\mathbf{k}} = \sum_{\mathbf{n}\in\mathbf{Z}^2} \widetilde{p}_{\mathbf{k}-M\mathbf{n}} \hat{c}^{j+1}_{\mathbf{n}}, j = J-1, J-2, , \cdots$. This is the subdivision algorithm with subdivision mask $\{\widetilde{p}_{\mathbf{k}}\}_{\mathbf{k}}$.

From (2.3) and (2.4), we know when the indices of hexagonally sampled data are labelled by $(k_1, k_2) \in \mathbf{Z}^2$ as in Fig. 3, the decomposition and reconstruction algorithms for hexagonal data with hexagonal filter banks are the conventional multiresolution decomposition and reconstruction algorithms for squarely sampled images. Thus, the integer-shift invariant multiresolution analysis theory implies that $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$ are PR filter banks if and only if

$$\sum_{0\le j\le m-1} p(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)\overline{\widetilde{p}(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)}=1, \tag{2.5}$$

$$\sum_{0\le j\le m-1} p(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)\overline{\widetilde{q}^{(\ell)}(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)}=0, \tag{2.6}$$

$$\sum_{0\le j\le m-1} q^{(\ell')}(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)\overline{\widetilde{q}^{(\ell)}(\boldsymbol{\omega}+2\pi M^{-T}\boldsymbol{\eta}_j)}=\delta_{\ell'-\ell}, \tag{2.7}$$

for $1 \le \ell, \ell' \le m-1$, $\boldsymbol{\omega} \in \mathbb{R}^2$, where $\boldsymbol{\eta}_j, 0 \le j \le m-1$ are the representatives of the group $\mathbf{Z}^2/(M^T\mathbf{Z}^2)$, δ_k is the kronecker-delta sequence: $\delta_k = 1$ if $k = 0$, and $\delta_k = 0$ if $k \neq 0$. When $M = 2I_2$, we may choose $\boldsymbol{\eta}_j, 0 \le j \le 3$ to be

$$\boldsymbol{\eta}_0=(0,0), \boldsymbol{\eta}_1=(-1,-1), \boldsymbol{\eta}_2=(1,0), \boldsymbol{\eta}_3=(-1,0), \tag{2.8}$$

while for $M = M_1$ in (2.2), we may use

$$\boldsymbol{\eta}_0=(0,0), \boldsymbol{\eta}_1=(1,0), \boldsymbol{\eta}_2=(-1,0). \tag{2.9}$$

Filter banks $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$ are also said to be **biorthogonal** if they satisfy (2.5)-(2.7).

Let $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$ be a pair of FIR filter banks. Let ϕ and $\widetilde{\phi}$ be the analysis and synthesis scaling functions (with dilation matrix M) associated with lowpass filters $p(\boldsymbol{\omega})$ and $\widetilde{p}(\boldsymbol{\omega})$ respectively, namely, $\phi, \widetilde{\phi}$ satisfy the refinement equations:

$$\phi(\mathbf{x})=\sum_{\mathbf{k}\in\mathbf{Z}^2} p_{\mathbf{k}}\phi(M\mathbf{x}-\mathbf{k}),\ \widetilde{\phi}(\mathbf{x})=\sum_{\mathbf{k}\in\mathbf{Z}^2} \widetilde{p}_{\mathbf{k}}\widetilde{\phi}(M\mathbf{x}-\mathbf{k}), \tag{2.10}$$

and $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, 1 \le \ell \le m-1$ are given by

$$\psi^{(\ell)}(\mathbf{x})=\textstyle\sum_{\mathbf{k}\in\mathbf{Z}^2} q^{(\ell)}_{\mathbf{k}}\phi(M\mathbf{x}-\mathbf{k}),\ \widetilde{\psi}^{(\ell)}(\mathbf{x})=\sum_{\mathbf{k}\in\mathbf{Z}^2} \widetilde{q}^{(\ell)}_{\mathbf{k}}\widetilde{\phi}(M\mathbf{x}-\mathbf{k}), \tag{2.11}$$

where $p_{\mathbf{k}}, \widetilde{p}_{\mathbf{k}}, q^{(\ell)}_{\mathbf{k}}, \widetilde{q}^{(\ell)}_{\mathbf{k}}$ are the coefficients of $p(\boldsymbol{\omega}), \widetilde{p}(\boldsymbol{\omega}), q^{(\ell)}(\boldsymbol{\omega}), \widetilde{q}^{(\ell)}(\boldsymbol{\omega})$, respectively

For an FIR lowpass filter $p(\boldsymbol{\omega}) = \frac{1}{m}\sum_{\mathbf{k}\in\mathbf{Z}^2} p_{\mathbf{k}}e^{-i\mathbf{k}\cdot\boldsymbol{\omega}}$, let T_p denote its transition operator matrix (with dilation matrix M):

$$T_p=[A_{M\mathbf{k}-\mathbf{j}}]_{\mathbf{k},\mathbf{j}\in[-K,K]^2}, \tag{2.12}$$

where $A_{\mathbf{j}} = (1/m)\sum_{\mathbf{n}\in\mathbf{Z}^2} p_{\mathbf{n}-\mathbf{j}}p_{\mathbf{n}}$ and K is a suitable positive integer depending on the filter length of p and the dilation matrix M. We say that T_p satisfies Condition E if 1 is its simple eigenvalue and all other eigenvalues λ of T_p satisfy $|\lambda| < 1$.

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

$$D_1^{\alpha_1}D_2^{\alpha_2}p(2\pi M^{-T}\boldsymbol{\eta}_j)=0,\ 1\le j\le m-1, \tag{2.13}$$

for all $(\alpha_1, \alpha_2) \in \mathbf{Z}^2_+$ with $\alpha_1 + \alpha_2 < K$, where $\boldsymbol{\eta}_j, 1 \le j \le m-1$, together with $\boldsymbol{\eta}_0 = (0,0)$, are the representatives of $\mathbf{Z}^2/(M^T\mathbf{Z}^2)$, D_1 and D_2 denote the partial derivatives with the first and second variables of $p(\boldsymbol{\omega})$ respectively. Under some conditions, sum rule order is equivalent to the approximation order of ϕ, see [15].

For a pair of biorthogonal FIR filter banks $\{p, q^{(1)}, \cdots, q^{(m-1)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \cdots, \widetilde{q}^{(m-1)}\}$, the associated scaling functions ϕ and $\widetilde{\phi}$ of $L^2(\mathbb{R}^2)$ are biorthogonal duals: $\int_{\mathbb{R}^2} \phi(\mathbf{x})\overline{\widetilde{\phi}(\mathbf{x}-\mathbf{k})}\, d\mathbf{x} = \delta_{k_1}\delta_{k_2}$, $\mathbf{k} = (k_1, k_2) \in \mathbf{Z}^2$, if and only if p and $\widetilde{p}$ have sum rule order 1, and the transition operator matrices T_p and $T_{\widetilde{p}}$ associated with p and $\widetilde{p}$ satisfy Condition E (see e.g. [8, 16]). In this case, $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, m-1$, are biorthogonal wavelets, namely, they generate biorthogonal (Riesz) bases $\{m^{\frac{j}{2}}\psi^{(\ell)}(M^j\mathbf{x}-\mathbf{k}) : 1 \leq \ell \leq m-1, j \in \mathbf{Z}, \mathbf{k} \in \mathbf{Z}^2\}$ and $\{m^{\frac{j}{2}}\widetilde{\psi}^{(\ell)}(M^j\mathbf{x}-\mathbf{k}) : \ 1 \leq \ell \leq m-1, j \in \mathbf{Z}, \mathbf{k} \in \mathbf{Z}^2\}$ for $L^2(\mathbb{R}^2)$. Thus to construct biorthogonal wavelets, we need to construct birthogonal FIR filter banks with the lowpass filters p and $\widetilde{p}$ to have sum rule of order at least 1 and the associated T_p and $T_{\widetilde{p}}$ to satisfy Condition E.

The scaling functions and wavelets $\phi, \widetilde{\phi}$ and $\psi^{(\ell)}, \widetilde{\psi}^{(\ell)}, \ell = 1, \cdots, m-1$ are the conventional scaling functions and wavelets: ϕ and $\widetilde{\phi}$ are refinable functions along $\mathbf{Z}^2$, and $\psi^{(\ell)}(\frac{\mathbf{x}}{2}), \widetilde{\psi}^{(\ell)}(\frac{\mathbf{x}}{2})$ are finite linear combinations of the shifts of ϕ and $\widetilde{\phi}$ along $\mathbf{Z}^2$. Let U be the matrix defined by

$$U = \begin{bmatrix} 1 & \frac{\sqrt{3}}{3} \\ 0 & \frac{2\sqrt{3}}{3} \end{bmatrix}.$$

Then U transforms the regular unit hexagonal lattice $\mathcal{G}$ onto the square lattice $\mathbf{Z}^2$. Define

$$\begin{aligned} &\Phi(\mathbf{x}) = \phi(U\mathbf{x}), \ \Psi^{(\ell)}(\mathbf{x}) = \psi^{(\ell)}(U\mathbf{x}), \\ &\widetilde{\Phi}(\mathbf{x}) = \widetilde{\phi}(U\mathbf{x}), \ \widetilde{\Psi}^{(\ell)}(\mathbf{x}) = \widetilde{\psi}^{(\ell)}(U\mathbf{x}), \ell = 1, \cdots, m-1. \end{aligned} \tag{2.14}$$

Then Φ and $\widetilde{\Phi}$ are refinable along $\mathcal{G}$ with the same coefficients $p_{\mathbf{k}}$ and $\widetilde{p}_{\mathbf{k}}$ for ϕ and $\widetilde{\phi}$, and $\Psi^{(\ell)}, \ell = 1, \cdots, m-1$ and $\widetilde{\Psi}^{(\ell)}, \ell = 1, \cdots, m-1$ are hexagonal biorthogonal wavelets (along the hexagonal lattice $\mathcal{G}$). The reader refers to [6] for refinable functions along a general lattice.

To end this section, we give the definitions of the symmetries of filter banks considered in this paper.

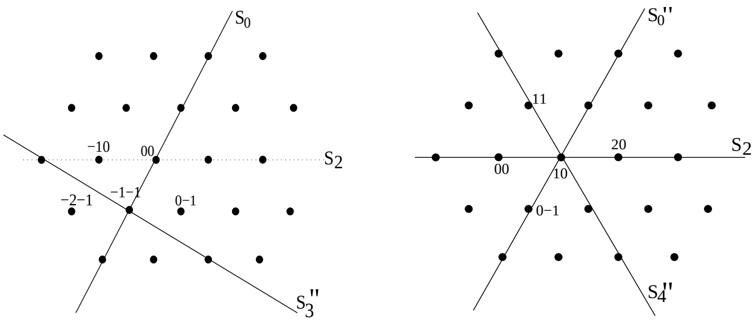

Figure 5: *Left: 2 axes (lines) of symmetry for the dyadic refinement highpass filter $q^{(1)}$; Right: 3 axes (lines) of symmetry for $\sqrt{3}$-refinement highpass filter $q^{(1)}$*

Definition 1. *Let $S_j, 0 \leq j \leq 5$ be the axes in the left part of Fig. 1. A (dyadic refinement) hexagonal filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ is said to have* **6-fold axial (line) symmetry** *or a* **full set of symmetries** *if (i) its lowpass filter $p(\boldsymbol{\omega})$ is symmetric around $S_0, \cdots, S_5$, (ii) its highpass filter satisfies that $e^{-i(\omega_1+\omega_2)}q^{(1)}(\boldsymbol{\omega})$ is symmetric around the axes S_0, S_3, and (iii) the other highpass filters $q^{(2)}$ and $q^{(3)}$ are the $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ rotations of highpass filter $q^{(1)}$ respectively.*

The left of Fig. 5 shows the symmetry of $q^{(1)}$, namely, $q^{(1)}$ is symmetric around the axes S_0, S_3'', where S_3'' is the 1-unit left and down shifts of S_3.

Definition 2. *Let $S_j, 0 \le j \le 5$ be the axes in the left part of Fig. 1. A ($\sqrt{3}$-refinement) hexagonal filter bank $\{p, q^{(1)}, q^{(2)}\}$ is said to have* **6-fold axial (line) symmetry** *or a* **full set of symmetries** *if (i) its lowpass filter $p(\boldsymbol{\omega})$ is symmetric around $S_0, \cdots, S_5$, (ii) its highpass filter satisfies that $e^{-i\omega_1} q^{(1)}(\boldsymbol{\omega})$ is symmetric around the axes S_0, S_2, S_4, and (iii) the other highpass filter $q^{(2)}$ is the π rotation of highpass filter $q^{(1)}$.*

The right of Fig. 5 shows the symmetry of $q^{(1)}$, namely, $q^{(1)}$ is symmetric around the axes S_0'', S_2, S_4'', where S_0'' (S_4'' resp.) is the 1-unit right shift of S_0 (S_4 resp.)

3 Dyadic refinement wavelets with 6-fold axial symmetry

In this section, we study the dyadic refinement biorthogonal wavelets with 6-fold axial symmetry. In §3.1, we provide some results on the 6-fold symmetry of filter banks and the associated scaling functions and wavelets. After that, in §3.2, we present families of FIR biorthogonal filter banks with 6-fold symmetry given by blocks. Also in §3.2, we construct biorthogonal wavelets associated with these filter banks. Finally, in §3.3, we first discuss how to represent decomposition and reconstruction algorithms as templates for surface processing. Then we show that some filter banks presented in §3.2 result in simple algorithms given by templates.

3.1 6-fold symmetry of dyadic refinement filter banks and wavelets

Let

$$\begin{aligned} &L_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; L_1 = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}, \; L_2 = \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}, \\ &L_3 = -L_0, \; L_4 = -L_1, \; L_5 = -L_2, \end{aligned} \tag{3.1}$$

and denote

$$R_1 = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad R_j = (R_1)^j, 0 \le j \le 5.$$

Then for a j, $0 \le j \le 5$, $\{p_{\mathbf{k}}\}$ is symmetric around the symmetry axis S_j in Fig. 1 if and only if $p_{L_j \mathbf{k}} = p_{\mathbf{k}}$; and $\{p_{R_j \mathbf{k}}\}$ is the $\frac{j\pi}{3}$ (anticlockwise) rotation of $\{p_{\mathbf{k}}\}$. Thus, with

$$h^{(1)}(\boldsymbol{\omega}) = e^{-i(\omega_1+\omega_2)} q^{(1)}(\boldsymbol{\omega}), \tag{3.2}$$

$\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if

$$\begin{aligned} &p(L_j^{-T}\boldsymbol{\omega}) = p(\boldsymbol{\omega}), \; 0 \le j \le 5, \; h^{(1)}(L_0\boldsymbol{\omega}) = h^{(1)}(L_3^{-T}\boldsymbol{\omega}) = h^{(1)}(\boldsymbol{\omega}), \\ &q^{(2)}(\boldsymbol{\omega}) = q^{(1)}(R_2^{-T}\boldsymbol{\omega}), \; q^{(3)}(\boldsymbol{\omega}) = q^{(1)}(R_4^{-T}\boldsymbol{\omega}). \end{aligned} \tag{3.3}$$

Observe that

$$L_j = R_j L_0, \; 0 \le j \le 5.$$

Thus, instead of considering all $L_j, 0 \le j \le 5$, we need only consider L_0, R_1 when we discuss the 6-fold axial symmetry of a filter bank. In particular, one has that $p(L_j^{-T}\boldsymbol{\omega}) = p(\boldsymbol{\omega}), 0 \le j \le 5$, is equivalent to $p(R_1^{-T}\boldsymbol{\omega}) = p(L_0\boldsymbol{\omega}) = p(\boldsymbol{\omega})$.

Proposition 1. *A filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if it satisfies*

$$[p, q^{(1)}, q^{(2)}, q^{(3)}]^T(R_1^{-T}\boldsymbol{\omega}) = S_1(2\boldsymbol{\omega})[p, q^{(1)}, q^{(2)}, q^{(3)}]^T(\boldsymbol{\omega}), \tag{3.4}$$

$$[p, q^{(1)}, q^{(2)}, q^{(3)}]^T(L_0\boldsymbol{\omega}) = S_0[p, q^{(1)}, q^{(2)}, q^{(3)}]^T(\boldsymbol{\omega}), \tag{3.5}$$

where

$$S_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\omega_2} \\ 0 & e^{-i(\omega_1+\omega_2)} & 0 & 0 \\ 0 & 0 & e^{i\omega_1} & 0 \end{bmatrix}, \ S_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.6}$$

Proof. For a filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$, let $h^{(1)}(\boldsymbol{\omega})$ be the filter given in (3.2), and define

$$h^{(2)}(\boldsymbol{\omega}) = e^{i\omega_1} q^{(2)}(\boldsymbol{\omega}), \ h^{(3)}(\boldsymbol{\omega}) = e^{i\omega_2} q^{(3)}(\boldsymbol{\omega}).$$

From $R_2^{-T}\boldsymbol{\omega} = [\omega_2, -\omega_1 - \omega_2]^T, \ R_4^{-T}\boldsymbol{\omega} = [-\omega_1 - \omega_2, \omega_1]^T$, we have

$$h^{(1)}(R_2^{-T}\boldsymbol{\omega}) = e^{i\omega_1} q^{(1)}(R_2^{-T}\boldsymbol{\omega}), \ h^{(1)}(R_4^{-T}\boldsymbol{\omega}) = e^{i\omega_2} q^{(1)}(R_4^{-T}\boldsymbol{\omega}),$$

which imply that $q^{(1)}(R_2^{-T}\boldsymbol{\omega}) = q^{(2)}(\boldsymbol{\omega})$ and $q^{(1)}(R_4^{-T}\boldsymbol{\omega}) = q^{(3)}(\boldsymbol{\omega})$ are equivalent to $h^{(1)}(R_2^{-T}\boldsymbol{\omega}) = h^{(2)}(\boldsymbol{\omega})$ and $h^{(1)}(R_4^{-T}\boldsymbol{\omega}) = h^{(3)}(\boldsymbol{\omega})$ respectively. Therefore, $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if

$$p(R_1^{-T}\boldsymbol{\omega}) = p(L_0\boldsymbol{\omega}) = p(\boldsymbol{\omega}), \tag{3.7}$$
$$h^{(1)}(L_0\boldsymbol{\omega}) = h^{(1)}(L_3^{-T}\boldsymbol{\omega}) = h^{(1)}(\boldsymbol{\omega}), \tag{3.8}$$
$$h^{(2)}(\boldsymbol{\omega}) = h^{(1)}(R_2^{-T}\boldsymbol{\omega}), \ h^{(3)}(\boldsymbol{\omega}) = h^{(1)}(R_4^{-T}\boldsymbol{\omega}). \tag{3.9}$$

From $L_3 = -L_0$ and $L_0^{-1} = L_0$, one has that (3.8) implies $h^{(1)}(-\boldsymbol{\omega}) = h^{(1)}(\boldsymbol{\omega})$. Thus, with the fact $R_1^3 = -I_2$, we have that (3.8) and (3.9) imply the following:

$$\begin{aligned}
&h^{(1)}(R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(-R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(R_4^{-T}\boldsymbol{\omega}) = h^{(3)}(\boldsymbol{\omega}), \\
&h^{(2)}(R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(R_2^{-T}R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(R_3^{-T}\boldsymbol{\omega}) = h^{(1)}(-\boldsymbol{\omega}) = h^{(1)}(\boldsymbol{\omega}), \\
&h^{(3)}(R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(R_4^{-T}R_1^{-T}\boldsymbol{\omega}) = h^{(1)}(R_5^{-T}\boldsymbol{\omega}) = h^{(1)}(-R_2^{-T}\boldsymbol{\omega}) = h^{(1)}(R_2^{-T}\boldsymbol{\omega}) = h^{(2)}(\boldsymbol{\omega}).
\end{aligned}$$

Furthermore, with the fact $R_2^{-T}L_0 = L_0R_4^{-T}$ and $L_0R_2^{-T} = R_4^{-T}L_0$, (3.8) and (3.9) also imply that

$$\begin{aligned}
h^{(2)}(L_0\boldsymbol{\omega}) = h^{(1)}(R_2^{-T}L_0\boldsymbol{\omega}) = h^{(1)}(L_0R_4^{-T}\boldsymbol{\omega}) = h^{(1)}(R_4^{-T}\boldsymbol{\omega}) = h^{(3)}(\boldsymbol{\omega}), \\
h^{(3)}(L_0\boldsymbol{\omega}) = h^{(1)}(R_4^{-T}L_0\boldsymbol{\omega}) = h^{(1)}(L_0R_2^{-T}\boldsymbol{\omega}) = h^{(1)}(R_2^{-T}\boldsymbol{\omega}) = h^{(2)}(\boldsymbol{\omega}).
\end{aligned}$$

Therefore, (3.7)-(3.9) imply

$$[p, h^{(1)}, h^{(2)}, h^{(3)}]^T(R_1^{-T}\boldsymbol{\omega}) = [p, h^{(3)}, h^{(1)}, h^{(2)}]^T(\boldsymbol{\omega}), \tag{3.10}$$
$$[p, h^{(1)}, h^{(2)}, h^{(3)}]^T(L_0\boldsymbol{\omega}) = [p, h^{(1)}, h^{(3)}, h^{(2)}]^T(\boldsymbol{\omega}). \tag{3.11}$$

On the other hand, it easy to show that (3.10) and (3.11) imply (3.7)-(3.9). Therefore, $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if (3.10) and (3.11) hold.

With $h^{(1)}(\boldsymbol{\omega}) = e^{-i(\omega_1+\omega_2)}q^{(1)}(\boldsymbol{\omega})$, $h^{(2)}(\boldsymbol{\omega}) = e^{i\omega_1}q^{(2)}(\boldsymbol{\omega})$, $h^{(3)}(\boldsymbol{\omega}) = e^{i\omega_2}q^{(3)}(\boldsymbol{\omega})$, one can easily show that (3.10) and (3.11) are equivalent to (3.4) and (3.5). Hence, $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if (3.4) and (3.5) hold, as desired. $\diamondsuit$

For an FIR filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$, with $q^{(0)}(\boldsymbol{\omega}) = p(\boldsymbol{\omega})$, we write $q^{(\ell)}(\boldsymbol{\omega}), 0 \le \ell \le 3$ as

$$q^{(\ell)}(\boldsymbol{\omega}) = \frac{1}{2}(q_0^{(\ell)}(2\boldsymbol{\omega}) + q_1^{(\ell)}(2\boldsymbol{\omega})e^{i(\omega_1+\omega_2)} + q_2^{(\ell)}(2\boldsymbol{\omega})e^{-i\omega_1} + q_3^{(\ell)}(2\boldsymbol{\omega})e^{-i\omega_2}),$$

where $q_k^{(\ell)}(\boldsymbol{\omega})$ are trigonometric polynomials. Let $V(\boldsymbol{\omega})$ denote the polyphase matrix of $\{p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})\}$ defined by

$$V(\boldsymbol{\omega}) = \begin{bmatrix} p_0(\boldsymbol{\omega}) & p_1(\boldsymbol{\omega}) & p_2(\boldsymbol{\omega}) & p_3(\boldsymbol{\omega}) \\ q_0^{(1)}(\boldsymbol{\omega}) & q_1^{(1)}(\boldsymbol{\omega}) & q_2^{(1)}(\boldsymbol{\omega}) & q_3^{(1)}(\boldsymbol{\omega}) \\ q_0^{(2)}(\boldsymbol{\omega}) & q_1^{(2)}(\boldsymbol{\omega}) & q_2^{(2)}(\boldsymbol{\omega}) & q_3^{(2)}(\boldsymbol{\omega}) \\ q_0^{(3)}(\boldsymbol{\omega}) & q_1^{(3)}(\boldsymbol{\omega}) & q_2^{(3)}(\boldsymbol{\omega}) & q_3^{(3)}(\boldsymbol{\omega}) \end{bmatrix}. \tag{3.12}$$

Then

$$[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T = \frac{1}{2} V(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}),$$

where $I_{00}(\boldsymbol{\omega})$ is defined by

$$I_{00}(\boldsymbol{\omega}) = [1, e^{i(\omega_1+\omega_2)}, e^{-i\omega_1}, e^{-i\omega_2}]^T. \tag{3.13}$$

The next proposition presents a characterization of the 6-fold axial symmetry of a filter bank in terms of its polyphase matrix. First we observe that 1-tap filter bank $\{1, e^{i(\omega_1+\omega_2)}, e^{-i\omega_1}, e^{-i\omega_2}\}$ has 6-fold symmetry. Thus, $I_{00}(\boldsymbol{\omega})$ defined above satisfies (3.4) and (3.5).

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

$$V(R_1^{-T}\boldsymbol{\omega}) = S_1(\boldsymbol{\omega}) V(\boldsymbol{\omega}) S_2(\boldsymbol{\omega}), \tag{3.14}$$

$$V(L_0\boldsymbol{\omega}) = S_0 V(\boldsymbol{\omega}) S_0, \tag{3.15}$$

where S_1 and S_0 are given by (3.6) and $S_2(\boldsymbol{\omega}) = S_1(\boldsymbol{\omega})^{-1}$:

$$S_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & e^{i(\omega_1+\omega_2)} & 0 \\ 0 & 0 & 0 & e^{-i\omega_1} \\ 0 & e^{-i\omega_2} & 0 & 0 \end{bmatrix}. \tag{3.16}$$

Proof. From the definition of $V(\boldsymbol{\omega})$,

$$[p, q^{(1)}, q^{(2)}, q^{(3)}](R_1^{-T}\boldsymbol{\omega}) = \frac{1}{2} V(2R_1^{-T}\boldsymbol{\omega}) I_{00}(R_1^{-T}\boldsymbol{\omega}) = \frac{1}{2} V(2R_1^{-T}\boldsymbol{\omega}) S_1(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}).$$

Thus (3.4) is equivalent to

$$\frac{1}{2} V(2R_1^{-T}\boldsymbol{\omega}) S_1(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}) = S_1(2\boldsymbol{\omega}) \frac{1}{2} V(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}),$$

or,

$$V(2R_1^{-T}\boldsymbol{\omega}) S_1(2\boldsymbol{\omega}) = S_1(2\boldsymbol{\omega}) V(2\boldsymbol{\omega}),$$

that is

$$V(R_1^{-T}\boldsymbol{\omega}) = S_1(\boldsymbol{\omega}) V(\boldsymbol{\omega}) S_1(\boldsymbol{\omega})^{-1},$$

which is (3.14).

Similarly, we have that (3.5) is equivalent to

$$V(2L_0\boldsymbol{\omega}) = S_0 V(2\boldsymbol{\omega}) S_0^{-1},$$

which is (3.15). Therefore, (3.4) and (3.5) are equivalent to (3.14) and (3.15). Hence, from Proposition 1, we know $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry if and only if $V(\boldsymbol{\omega})$ satisfies (3.14) and (3.15). $\diamondsuit$

Proposition 3. *Suppose an FIR filter bank $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ has 6-fold axial symmetry. Let ϕ be the associated scaling function with dilation matrix $M = 2I_2$ and $\psi^{(\ell)}, \ell = 1, 2, 3$ be the functions define by (2.11) with $q^{(\ell)}$. Then*

$$\phi(L_j\mathbf{x}) = \phi(\mathbf{x}),\ 0 \le j \le 5, \tag{3.17}$$

$$\psi^{(2)}(\mathbf{x}) = \psi^{(1)}(R_2\mathbf{x}),\ \psi^{(3)}(\mathbf{x}) = \psi^{(1)}(R_4\mathbf{x}), \tag{3.18}$$

and

$$\psi^{(1)}(L_0\mathbf{x}) = \psi^{(1)}(\mathbf{x}),\ \psi^{(1)}(L_3\mathbf{x}) = \psi^{(1)}(\mathbf{x} - (1,1)), \tag{3.19}$$
$$\psi^{(1)}(L_2\mathbf{x}) = \psi^{(1)}(R_4\mathbf{x}),\ \psi^{(1)}(L_4\mathbf{x}) = \psi^{(1)}(R_2\mathbf{x}). \tag{3.20}$$

Proof. From (2.10) (with $M = 2I_2$), we have $\widehat{\phi}(\boldsymbol{\omega}) = p(\frac{\boldsymbol{\omega}}{2})\widehat{\phi}(\frac{\boldsymbol{\omega}}{2})$. Thus $\widehat{\phi}(\boldsymbol{\omega}) = \Pi_{k=1}^{\infty} p(2^{-k}\boldsymbol{\omega})\widehat{\phi}(0)$. Therefore, $p(L_j^{-T}\boldsymbol{\omega}) = p(\boldsymbol{\omega})$, $0 \le j \le 5$, imply

$$\widehat{\phi}(L_j^{-T}\boldsymbol{\omega}) = \Pi_{k=1}^{\infty} p(2^{-k}L_j^{-T}\boldsymbol{\omega})\widehat{\phi}(0) = \Pi_{k=1}^{\infty} p(2^{-k}\boldsymbol{\omega})\widehat{\phi}(0) = \widehat{\phi}(\boldsymbol{\omega}),$$

which is (3.17).

From (2.11) (with $M = 2I_2$), we have $\widehat{\psi}^{(\ell)}(\boldsymbol{\omega}) = q^{(\ell)}(\frac{\boldsymbol{\omega}}{2})\widehat{\phi}(\frac{\boldsymbol{\omega}}{2}),\ \ell = 1, 2, 3$. Thus for $j = 1, 2$,

$$\widehat{\psi}^{(1)}(R_{2j}^{-T}\boldsymbol{\omega}) = q^{(1)}(R_{2j}^{-T}\boldsymbol{\omega}/2)\widehat{\phi}(R_{2j}^{-T}\boldsymbol{\omega}/2) = q^{(j+1)}(\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2) = \widehat{\psi}^{(j+1)}(\boldsymbol{\omega}).$$

Therefore, (3.18) holds.

Finally, let us prove (3.19) and (3.20). From $q^{(1)}(L_0\boldsymbol{\omega}) = q^{(1)}(\boldsymbol{\omega})$ and $\phi(L_0\mathbf{x}) = \phi(\mathbf{x})$, we have

$$\widehat{\psi}^{(1)}(L_0\boldsymbol{\omega}) = q^{(1)}(L_0\boldsymbol{\omega}/2)\widehat{\phi}(L_0\boldsymbol{\omega}/2) = q^{(1)}(\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2) = \widehat{\psi}^{(1)}(\boldsymbol{\omega}),$$

which is the first equation in (3.19).

For the proof of the second equation in (3.19), from $h^{(1)}(-\boldsymbol{\omega}) = h^{(1)}(\boldsymbol{\omega})$ as shown in the proof of Proposition 1, where $h^{(1)}(\boldsymbol{\omega}) = e^{-i(\omega_1+\omega_2)}q^{(1)}(\boldsymbol{\omega})$, we have

$$q^{(1)}(-\boldsymbol{\omega}) = e^{-i2(\omega_1+\omega_2)}q^{(1)}(\boldsymbol{\omega}).$$

This, together with $L_3 = -L_0, \psi^{(1)}(L_0\mathbf{x}) = \psi^{(1)}(\mathbf{x})$ and $\widehat{\phi}(-\boldsymbol{\omega}) = \widehat{\phi}(\boldsymbol{\omega})$ (followed from (3.17)), leads to that

$$\begin{aligned}&\widehat{\psi}^{(1)}(L_3^{-T}\boldsymbol{\omega}) = \widehat{\psi}^{(1)}(-L_0\boldsymbol{\omega}) = \widehat{\psi}^{(1)}(-\boldsymbol{\omega}) = q^{(1)}(-\boldsymbol{\omega}/2)\widehat{\phi}(-\boldsymbol{\omega}/2)\\ &= e^{-i(\omega_1+\omega_2)}q^{(1)}(\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2) = e^{-i(\omega_1+\omega_2)}\widehat{\psi}^{(1)}(\boldsymbol{\omega}),\end{aligned}$$

which is the second equation in (3.19).

The proof for two equations in (3.20) is similar. Here we provide the proof for the first one. From $L_2 = R_2L_0$ and $q^{(2)}(L_0\boldsymbol{\omega}) = q^{(3)}(\boldsymbol{\omega})$ (see (3.5)), we have

$$\begin{aligned}&\widehat{\psi}^{(1)}(L_2^{-T}\boldsymbol{\omega}) = \widehat{q}^{(1)}(L_2^{-T}\boldsymbol{\omega}/2)\widehat{\phi}(L_2^{-T}\boldsymbol{\omega}/2) = q^{(1)}(R_2^{-T}L_0\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2)\\ &= q^{(2)}(L_0\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2) = q^{(3)}(\boldsymbol{\omega}/2)\widehat{\phi}(\boldsymbol{\omega}/2) = \widehat{\psi}^{(3)}(\boldsymbol{\omega}).\end{aligned}$$

Thus $\psi^{(1)}(L_2\mathbf{x}) = \psi^{(3)}(\mathbf{x})$. This and (3.18) lead to $\psi^{(1)}(L_2\mathbf{x}) = \psi^{(1)}(R_4\mathbf{x})$, as desired. $\diamondsuit$

3.2 Biorthogonal FIR filter banks and wavelets with 6-fold axial symmetry

In this subsection we present biorthogonal FIR filter banks with 6-fold symmetry. These filter banks have block structures, namely, they are given by simple blocks and a simple initial filter bank. As mentioned above, 1-tap filter bank $\{1, e^{i(\omega_1+\omega_2)}, e^{-i\omega_1}, e^{-i\omega_2}\}$ has 6-fold symmetry and hence, it could be used as the initial filter bank. So the key to obtain the block structures is to find suitable blocks which satisfy (3.14) and (3.15). In the following we present several types of such blocks. First observe that two filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are biorthogonal if and only if

$$V(\boldsymbol{\omega})\widetilde{V}(\boldsymbol{\omega})^* = I_4, \quad \boldsymbol{\omega} \in \mathbb{R}^2, \tag{3.21}$$

where $V(\boldsymbol{\omega})$ and $\widetilde{V}(\boldsymbol{\omega})$ are their polyphase matrices defined by (3.12).

In the following we use the notations:

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

Thus an FIR filter $p(\boldsymbol{\omega})$ can be written as a polynomial of x, y. Denote

$$B(\boldsymbol{\omega}) = \begin{bmatrix} \gamma + \rho(x + xy + y + \frac{1}{x} + \frac{1}{xy} + \frac{1}{y}) & \alpha(1+xy) + \beta(x+y) & \alpha(1+\frac{1}{x}) + \beta(y + \frac{1}{xy}) & \alpha(1+\frac{1}{y}) + \beta(x + \frac{1}{xy}) \\ \tau(1+\frac{1}{xy}) & 1 & 0 & 0 \\ \tau(1+x) & 0 & 1 & 0 \\ \tau(1+y) & 0 & 0 & 1 \end{bmatrix}, \tag{3.22}$$

where $\rho = \tau(\alpha + 2\beta)$, $\alpha, \beta, \gamma, \tau$ are constants with $\gamma \neq 6\alpha\tau$. One can verify that $B(\boldsymbol{\omega})$ satisfies (3.14) and (3.15). Thus filter banks built by $B(\boldsymbol{\omega})$ have 6-fold axial symmetry. For example, the filter bank given by $\frac{1}{4}B(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega})$ has 6-fold axial symmetry. Furthermore, the determinant of $B(\boldsymbol{\omega})$ is $\gamma - 6\alpha\tau$, a nonzero constant. Thus, the inverse of $B(\boldsymbol{\omega})$ is a matrix whose entries are also polynomials of x, y. More precisely, $\widetilde{B}(\boldsymbol{\omega}) = (B(\boldsymbol{\omega})^{-1})^*$ is given by

$$\widetilde{B}(\boldsymbol{\omega}) = \frac{1}{\gamma - 6\alpha\tau} \times \begin{bmatrix} 1 & -\tau(1+xy) & -\tau(1+\frac{1}{x}) & -\tau(1+\frac{1}{y}) \\ -(\alpha + \frac{\alpha}{xy} + \frac{\beta}{x} + \frac{\beta}{y}) & \xi(\frac{1}{xy}, y) & \tau(1+\frac{1}{x})(\alpha + \frac{\alpha}{xy} + \frac{\beta}{x} + \frac{\beta}{y}) & \tau(1+\frac{1}{y})(\alpha + \frac{\alpha}{xy} + \frac{\beta}{x} + \frac{\beta}{y}) \\ -(\alpha + \alpha x + \beta xy + \frac{\beta}{y}) & \tau(1+xy)(\alpha + \alpha x + \beta xy + \frac{\beta}{y}) & \xi(x, y) & \tau(1+\frac{1}{y})(\alpha + \alpha x + \beta xy + \frac{\beta}{y}) \\ -(\alpha + \alpha y + \beta xy + \frac{\beta}{x}) & \tau(1+xy)(\alpha + \alpha y + \beta xy + \frac{\beta}{x}) & \tau(1+\frac{1}{x})(\alpha + \alpha y + \beta xy + \frac{\beta}{x}) & \xi(y, x) \end{bmatrix}, \tag{3.23}$$

where $\xi(x, y) = \gamma - 4\alpha\tau + \beta\tau(y + \frac{1}{y} + xy + \frac{1}{xy}) + \alpha\tau(x + \frac{1}{x})$. Therefore, the filter bank given by $\widetilde{B}(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega})$ is biorthogonal to that given by $\frac{1}{4}B(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega})$. Furthermore, it also has 6-fold axial symmetry since its polyphase matrix $2\widetilde{B}(\boldsymbol{\omega})$ satisfies (3.14) and (3.15).

We may use other blocks. For example, we may use

$$C(\boldsymbol{\omega}) = \begin{bmatrix} \gamma + \rho(x + xy + y + \frac{1}{x} + \frac{1}{xy} + \frac{1}{y}) & \alpha(1+xy) & \alpha(1+\frac{1}{x}) & \alpha(1+\frac{1}{y}) \\ \tau(1+\frac{1}{xy}) + \sigma(\frac{1}{x} + \frac{1}{y}) & 1 & 0 & 0 \\ \tau(1+x) + \sigma(xy + \frac{1}{y}) & 0 & 1 & 0 \\ \tau(1+y) + \sigma(xy + \frac{1}{x}) & 0 & 0 & 1 \end{bmatrix}, \tag{3.24}$$

where $\rho = \alpha(\tau + 2\sigma)$, $\alpha, \beta, \gamma, \sigma$ are constants with $\gamma \neq 6\alpha\tau$, or we may use

$$D(\boldsymbol{\omega}) = \begin{bmatrix} 1 & \eta(\frac{1}{xy}, y) & \eta(x, y) & \eta(y, x) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.25}$$

where

$$\eta(x,y) = -w(1+\frac{1}{x}) - u(y+\frac{1}{xy}), \tag{3.26}$$

for some constants u, w.

The determinants of $C(\boldsymbol{\omega})$ and $D(\boldsymbol{\omega})$ are $\gamma - 6\alpha\tau$ and 1 respectively. Thus, the inverse of $C(\boldsymbol{\omega})$ and $D(\boldsymbol{\omega})$ are matrices whose entries are also polynomials of x, y with $\widetilde{C}(\boldsymbol{\omega}) = (C(\boldsymbol{\omega})^{-1})^*$ given by

$$\widetilde{C}(\boldsymbol{\omega}) = \tfrac{1}{\gamma-6\alpha\tau}\times \\ \begin{bmatrix} 1 & -\tau(1+xy)-\sigma(x+y) & -\tau(1+\frac{1}{x})-\sigma(y+\frac{1}{xy}) & -\tau(1+\frac{1}{y})-\sigma(x+\frac{1}{xy}) \\ -\alpha-\frac{\alpha}{xy} & \zeta(\frac{1}{xy},y) & \alpha(1+\frac{1}{xy})(\tau+\frac{\tau}{x}+\frac{\sigma}{y}+\frac{\sigma}{xy}) & \alpha(1+\frac{1}{xy})(\tau+\frac{\tau}{y}+\frac{\sigma}{x}+\frac{\sigma}{xy}) \\ -\alpha-\alpha x & \alpha(1+x)(\tau+\tau xy+\sigma x+\sigma y) & \zeta(x,y) & \alpha(1+x)(\tau+\frac{\tau}{y}+\sigma x+\frac{\sigma}{xy}) \\ -\alpha-\alpha y & \alpha(1+y)(\tau+\tau xy+\sigma x+\sigma y) & \alpha(1+y)(\tau+\frac{\tau}{x}+\frac{\sigma}{y}+\frac{\sigma}{xy}) & \zeta(y,x) \end{bmatrix}, \tag{3.27}$$

where $\zeta(x,y) = \gamma - 4\alpha\tau + \alpha\sigma(y+\frac{1}{y}+xy+\frac{1}{xy}) + \alpha\tau(x+\frac{1}{x})$, and $\widetilde{D}(\boldsymbol{\omega}) = (D(\boldsymbol{\omega})^{-1})^*$ given by

$$\widetilde{D}(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\eta(xy,\frac{1}{y}) & 1 & 0 & 0 \\ -\eta(\frac{1}{x},\frac{1}{y}) & 0 & 1 & 0 \\ -\eta(\frac{1}{y},\frac{1}{x}) & 0 & 0 & 1 \end{bmatrix}. \tag{3.28}$$

One can show that $C(\boldsymbol{\omega}), D(\boldsymbol{\omega})$, $\widetilde{C}(\boldsymbol{\omega})$, and $\widetilde{D}(\boldsymbol{\omega})$ satisfy (3.14) and (3.15).

For the block $D(\boldsymbol{\omega})$, one may use other $\eta(x,y)$ such that $D(\boldsymbol{\omega})$ satisfies (3.14) and (3.15). For example, we may use

$$\eta(x,y) = -w(1+\frac{1}{x}) - u(y+\frac{1}{xy}) - w_1(xy+\frac{y}{x}+\frac{1}{y}+\frac{1}{x^2y}) - u_1(x+\frac{1}{x^2}). \tag{3.29}$$

Theorem 1. *Suppose FIR filter banks $\{p, q^{(1)}, q^{(2)}, \widetilde{q}^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are given by*

$$\begin{aligned} [p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T &= V_n(2\boldsymbol{\omega})V_{n-1}(2\boldsymbol{\omega})\cdots V_0(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega}), \\ [\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega})]^T &= \frac{1}{4}\widetilde{V}_n(2\boldsymbol{\omega})\widetilde{V}_{n-1}(2\boldsymbol{\omega})\cdots \widetilde{V}_0(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega}) \end{aligned} \tag{3.30}$$

for some $n \in \mathbf{Z}_+$, where $I_{00}(\boldsymbol{\omega})$ is defined by (3.13), each $V_k(\boldsymbol{\omega})$ is a $B(\boldsymbol{\omega})$ in (3.22) or a $\widetilde{B}(\boldsymbol{\omega})$ in (3.23) for some parameters $\alpha_k, \beta_k, \gamma_k, \tau_k$, or a $C(\boldsymbol{\omega})$ in (3.24) or a $\widetilde{C}(\boldsymbol{\omega})$ in (3.27) for some parameters $\alpha_k, \beta_k, \gamma_k, \sigma_k$, or a $D(\boldsymbol{\omega})$ in (3.25) or a $\widetilde{D}(\boldsymbol{\omega})$ in (3.28) for some parameters w_k, u_k, and $\widetilde{V}_k(\boldsymbol{\omega}) = (V_k(\boldsymbol{\omega})^{-1})^*$ is the corresponding $\widetilde{B}(\boldsymbol{\omega})$ in (3.23) ($B(\boldsymbol{\omega})$ in (3.22), $\widetilde{C}(\boldsymbol{\omega})$ in (3.27), $C(\boldsymbol{\omega})$ in (3.24), $\widetilde{D}(\boldsymbol{\omega})$ in (3.28), or $D(\boldsymbol{\omega})$ in (3.25) accordingly), then $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ are biorthogonal FIR filter banks with 6-fold axial symmetry.

In the following we illustrate in examples that the above filter banks lead to biorthogonal wavelets. When we construct biorthogonal wavelets, we will construct the synthesis scaling function $\widetilde{\phi}$ to have a higher smoothness order. Smoothness of $\widetilde{\phi}$ is in general more important than that for ϕ. For example, when the filter banks are applied to subdivision surface multiresolution processing, certain smoothness of $\widetilde{\phi}$ is required to assure the reconstructed surfaces to have some smoothness. Here we consider the Sobolev smoothness of ϕ and $\widetilde{\phi}$. We say a function f on $\mathbb{R}^2$ to be in the Sobolev space W^s for some $s > 0$ if its Fourier transform $\hat{f}$ satisfies $\int_{\mathbb{R}^2}(1+|\boldsymbol{\omega}|^2)^s|\hat{f}(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} < \infty$. The Sobolev smoothness of ϕ can be given by the eigenvalues of the transition operator matrix T_p associated with the corresponding lowpass filter p, see [18, 17].

Example 1. *Let* $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ *and* $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ *be the biorthogonal filter banks given by (3.30) for* $n = 1$ *with*

$$[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T = B_1(2\boldsymbol{\omega})\widetilde{B}_0(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega}), \tag{3.31}$$
$$[\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega})]^T = \frac{1}{4}\widetilde{B}_1(2\boldsymbol{\omega})B_0(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega}),$$

where $\widetilde{B}_0(\boldsymbol{\omega}), \widetilde{B}_1(\boldsymbol{\omega})$*, and* $B_0(\boldsymbol{\omega}), B_1(\boldsymbol{\omega})$ *are given by (3.23) and (3.22) for some parameters* $\gamma_0, \alpha_0, \beta_0, \tau_0$ *and* $\gamma_1, \alpha_1, \beta_1, \tau_1$ *respectively. By solving the equations of sum rule order 1 for* $p(\boldsymbol{\omega}), \widetilde{p}(\boldsymbol{\omega})$*, we have*

$$\tau = 4(\alpha_1 + \beta_1 - \frac{1}{8}),\ \gamma_1 = 2(\alpha + \beta - \tau_1 + 3\tau_1\alpha_1),\ \gamma = 6\tau(2\tau_1 - 2\beta - \alpha) + 2(\alpha + \beta - \tau_1).$$

Because of the symmetry of $p(\boldsymbol{\omega}), \widetilde{p}(\boldsymbol{\omega})$*, the conditions in (2.13) for* $p(\boldsymbol{\omega}), \widetilde{p}(\boldsymbol{\omega})$ *with* $(\alpha_1, \alpha_2) = (1, 0), (0, 1)$ *are automatically satisfied. Thus the resulting* $p(\boldsymbol{\omega})$ *and* $\widetilde{p}(\boldsymbol{\omega})$ *actually have sum rule order 2. If we choose*

$$\alpha = -\frac{1}{8},\ \alpha_1 = \frac{127}{512},\ \beta = \frac{1}{64},\ \beta_1 = -\frac{7}{64},\ \tau_1 = -\frac{61}{256},$$

then the resulting ϕ *is in* $W^{0.0241}$*, and* $\widetilde{\phi}$ *in* $W^{1.8515}$*. One may choose other values such that the resulting* $\widetilde{\phi}$ *is smoother. But* $\widetilde{\phi}$ *cannot gain a big increment of smoothness order as long as its dual* ϕ *is in* $L^2(\mathbb{R}^2)$.

Example 2. *Let* $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ *and* $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ *be the biorthogonal filter banks given by (3.30) for* $n = 1$ *with*

$$[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T = D(2\boldsymbol{\omega})\widetilde{B}(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \tag{3.32}$$
$$[\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega})]^T = \frac{1}{4}\widetilde{D}(2\boldsymbol{\omega})B(2\boldsymbol{\omega})I_0(\boldsymbol{\omega}),$$

where $B(\boldsymbol{\omega})$*,* $\widetilde{B}(\boldsymbol{\omega}), D(\boldsymbol{\omega})$*, and* $\widetilde{D}(\boldsymbol{\omega})$ *are given by (3.22), (3.23), (3.25), and (3.28) for some parameters* $\gamma, \alpha, \beta, \tau$ *and* w, u *respectively. If*

$$\tau = -\frac{1}{2} - 4w - 4u,\ \alpha = \frac{1}{2} - \beta,\ \gamma = 1 - 6\tau\beta - 3\tau, \tag{3.33}$$

then both $p(\boldsymbol{\omega})$ *and* $\widetilde{p}(\boldsymbol{\omega})$ *have sum rule order 2. If in addition,*

$$\beta = \frac{1}{8},\ u = -w - \frac{3}{20},$$

then $\widetilde{p}(\boldsymbol{\omega})$ *has sum rule order 4. In this case,*

$$\gamma = \frac{5}{8},\ \alpha = \frac{3}{8},\ \beta = \frac{1}{8},\ \tau = \frac{1}{10},\ \rho = \frac{1}{16},$$

and $\widetilde{p}(\boldsymbol{\omega}) = \frac{1}{64}e^{2i(\omega_1+\omega_2)}(1 + e^{-i\omega_1})^2(1 + e^{-i\omega_2})^2(1 + e^{-i(\omega_1+\omega_2)})^2$*, the mask for Loop's scheme [25]. We find that for this* $\widetilde{p}(\boldsymbol{\omega})$*, it is impossible to choose the left parameter* w *such that the analysis scaling function* ϕ *is in* $L^2(\mathbb{R}^2)$*. We also used a different* $D(\boldsymbol{\omega})$ *with* $\eta(x, y)$ *given in (3.29) with more parameters* w, u, w_1, u_1*. However, even in this case, we still cannot find a* ϕ *in* $L^2(\mathbb{R}^2)$ *provided that* $\widetilde{p}(\boldsymbol{\omega})$ *is the mask of Loop's scheme. Thus we should choose other parameters for* $\widetilde{p}(\boldsymbol{\omega})$ *(with sum rule order 2). If we choose*

$$\beta = \frac{17}{256},\ w = -\frac{1}{4},\ u = \frac{57}{512}, \tag{3.34}$$

then the resulting ϕ *is in* $W^{0.0029}$*, and* $\widetilde{\phi}$ *in* $W^{1.8672}$*. We numerically verify that the resulting* $\widetilde{\phi}$ *is* C^1*. One can choose other parameters such that* $\widetilde{\phi} \in C^1$ *and* $\phi \in L^2(\mathbb{R}^2)$. $\diamondsuit$

Example 3. *Let $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ be the biorthogonal filter banks given by*

$$
\begin{aligned}
[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega}), q^{(3)}(\boldsymbol{\omega})]^T &= \widetilde{D}_1(2\boldsymbol{\omega}) D(2\boldsymbol{\omega}) \widetilde{B}(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}), \\
[\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega}), \widetilde{q}^{(3)}(\boldsymbol{\omega})]^T &= \frac{1}{4} D_1(2\boldsymbol{\omega}) \widetilde{D}(2\boldsymbol{\omega}) B(2\boldsymbol{\omega}) I_{00}(\boldsymbol{\omega}),
\end{aligned}
\tag{3.35}
$$

where $B(\boldsymbol{\omega})$, $\widetilde{B}(\boldsymbol{\omega}), D(\boldsymbol{\omega})$, and $\widetilde{D}(\boldsymbol{\omega})$ are given by (3.22), (3.23), (3.25), and (3.28) for some parameters $\gamma, \alpha, \beta, \tau$ and w, u respectively, and $\widetilde{D}_1(\boldsymbol{\omega})$, and $D_1(\boldsymbol{\omega})$ are given by (3.28), and (3.25) with parameters w_1, u_1.

When

$$
\tau = -\tfrac{1}{2} + \tfrac{1-2\alpha-2\beta+2u_1+2w_1}{6(\alpha+\beta)(w_1+u_1)}, \gamma = 6\tau\alpha + \tfrac{2(\alpha+\beta)(1-6\tau)}{1+2u_1+2w_1}, u = -w - \tfrac{1-2\alpha-2\beta+2u_1+2w_1}{24(\alpha+\beta)(u_1+w_1)}, \tag{3.36}
$$

both $\widetilde{p}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ have sum rule order 2. Furthermore, if

$$
\begin{aligned}
&u_1 = (8w_1\beta + 3\beta - \alpha)/(8\alpha), \\
&w = \tfrac{16(16w_1-3)\alpha^3 + \{11+8(12\beta+80w_1\beta-31w_1-32w_1^2)\}\alpha^2 - (8w_1+3)(80w_1\beta+45\beta-16w_1-48\beta^2-2)\alpha + 3\beta(8w_1+3)^2(1-2\beta)}{12(8w_1(\alpha+\beta)+3\beta-\alpha)^2(2\alpha+2\beta-1)},
\end{aligned}
\tag{3.37}
$$

then $\widetilde{p}(\boldsymbol{\omega})$ has sum rule order 4. If

$$
\alpha = 0.4616,\ \beta = 0.0217,\ w_1 = -0.0266, \tag{3.38}
$$

then the resulting $\phi \in W^{0.0014}$ and $\widetilde{\phi} \in W^{2.9743}$. We numerically verify that this $\widetilde{\phi}$ is in C^2. We can choose other α, β, w_1 such that $\phi \in L^2(\mathbb{R}^2)$ and $\widetilde{\phi} \in C^2$.

3.3 Dyadic multiresolution algorithms for triangle surface processing

In this subsection, we first show how to represent decomposition and reconstruction algorithms as templates for surface multiresolution processing. Then we show that some filter banks presented in §3.2 result in simple algorithms given by templates. Here we consider the regular triangle surface, which can be represented (locally) as a regular mesh with nodes in $\mathcal{G}$ in the 2D plane as shown on the left of Fig. 6.

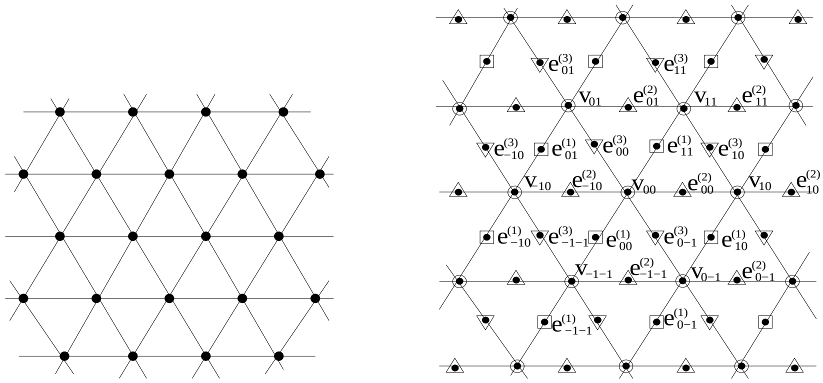

Figure 6: *Left: Regular triangle mesh; Right: Initial data separated into 4 groups: $\{v_{\mathbf{k}}\}$, $\{e^{(1)}_{\mathbf{k}}\}$, $\{e^{(2)}_{\mathbf{k}}\}$, $\{e^{(3)}_{\mathbf{k}}\}$*

To obtain templates corresponding to given decomposition reconstruction algorithms (with both lowpass and highpass filters), we first separate the *type E* nodes of $\mathcal{G}$ into three groups. Recall that the nodes $\mathbf{g} = k_1\mathbf{v}_1 + k_2\mathbf{v}_2$ of the unit regular lattice $\mathcal{G}$ defined by (2.1) are labelled as $\mathbf{k} = (k_1, k_2) \in \mathbf{Z}^2$. Thus $2\mathbf{Z}^2 = \{(2k_1, 2k_2), (k_1, k_2) \in \mathbf{Z}^2\}$ is the set of the labels for the *type V* nodes, the nodes in $\mathcal{G}_4$, and $\mathbf{Z}^2\backslash(2\mathbf{Z}^2)$ is that for the *type E* nodes, the nodes in $\mathcal{G}\backslash\mathcal{G}_4$. We separate *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 of Fig. 6, where squares, $\triangle$ and ∇ denote these three groups of *type E* nodes (the big circles denote *type V* nodes).

Let $\mathcal{C}=\{c_{\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ be the data sampled on $\mathcal{G}$. Thus $\{c_{2\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ is the set of data associated with *type V* nodes of $\mathcal{G}_4$, $\{c_{2\mathbf{k}-(1,1)}\}_{\mathbf{k}\in\mathbf{Z}^2}$, $\{c_{2\mathbf{k}+(1,0)}\}_{\mathbf{k}\in\mathbf{Z}^2}$ and $\{c_{2\mathbf{k}+(0,1)}\}_{\mathbf{k}\in\mathbf{Z}^2}$ are three sets of data associated with the above three groups of *type E* nodes. Denote

$$v_{\mathbf{k}}=c_{2\mathbf{k}},\ e^{(1)}_{\mathbf{k}}=c_{2\mathbf{k}-(1,1)},\ e^{(2)}_{\mathbf{k}}=c_{2\mathbf{k}+(1,0)},\ e^{(3)}_{\mathbf{k}}=c_{2\mathbf{k}+(0,1)},\ \mathbf{k}\in\mathbf{Z}^2. \tag{3.39}$$

Refer to the right of Fig. 6 for these four groups of data.

The multiresolution decomposition algorithm is to decompose the original data $\mathcal{C}=\{c_{\mathbf{k}}\}_{\mathbf{k}}$ with the analysis filter bank into the "approximation" $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$ and the "details" $\{d^{(1,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ $\{d^{(2,1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{d^{(3,1)}_{\mathbf{k}}\}_{\mathbf{k}}$, while the (prefect) multiresolution reconstruction algorithm is to recover exactly $\mathcal{C}$ from $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$, $\{d^{(1,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ $\{d^{(2,1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{d^{(3,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ with the synthesis filter bank. Denote

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

Then, the decomposition algorithm can be written as

$$\begin{aligned}\widetilde{v}_{\mathbf{k}}&=\tfrac14\textstyle\sum_{\mathbf{k}'\in\mathbf{Z}^2}p_{\mathbf{k}'-2\mathbf{k}}c_{\mathbf{k}'},\ \widetilde{e}^{(1)}_{\mathbf{k}}=\tfrac14\sum_{\mathbf{k}'\in\mathbf{Z}^2}q^{(1)}_{\mathbf{k}'-2\mathbf{k}}c_{\mathbf{k}'},\\ \widetilde{e}^{(2)}_{\mathbf{k}}&=\tfrac14\textstyle\sum_{\mathbf{k}'\in\mathbf{Z}^2}q^{(2)}_{\mathbf{k}'-2\mathbf{k}}c_{\mathbf{k}'},\ \widetilde{e}^{(3)}_{\mathbf{k}}=\tfrac14\sum_{\mathbf{k}'\in\mathbf{Z}^2}q^{(3)}_{\mathbf{k}'-2\mathbf{k}}c_{\mathbf{k}'}\end{aligned} \tag{3.40}$$

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

$$c_{\mathbf{k}}=\sum_{\mathbf{k}'\in\mathbf{Z}^2}\{\widetilde{p}_{\mathbf{k}-2\mathbf{k}'}\widetilde{v}_{\mathbf{k}'}+\widetilde{q}^{(1)}_{\mathbf{k}-2\mathbf{k}'}\widetilde{e}^{(1)}_{\mathbf{k}'}+\widetilde{q}^{(2)}_{\mathbf{k}-2\mathbf{k}'}\widetilde{e}^{(2)}_{\mathbf{k}'}+\widetilde{q}^{(3)}_{\mathbf{k}-2\mathbf{k}'}\widetilde{e}^{(3)}_{\mathbf{k}'}\},\ \mathbf{k}\in\mathbf{Z}^2. \tag{3.41}$$

Considering $c_{\mathbf{k}}$ in (3.41) with $\mathbf{k}$ in four different cases: $(2j_1,2j_2)$, $(2j_1-1,2j_2-1)$, $(2j_1+1,2j_2)$, $(2j_1,2j_2+1)$, we write the reconstruction algorithm (3.41) as

$$\begin{aligned}v_{\mathbf{k}}&=\textstyle\sum_{\mathbf{n}\in\mathbf{Z}^2}\{\widetilde{p}_{2\mathbf{n}}\widetilde{v}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(1)}_{2\mathbf{n}}\widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(2)}_{2\mathbf{n}}\widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(3)}_{2\mathbf{n}}\widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\},\\ e^{(1)}_{\mathbf{k}}&=\textstyle\sum_{\mathbf{n}\in\mathbf{Z}^2}\{\widetilde{p}_{2\mathbf{n}-(1,1)}\widetilde{v}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(1)}_{2\mathbf{n}-(1,1)}\widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(2)}_{2\mathbf{n}-(1,1)}\widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(3)}_{2\mathbf{n}-(1,1)}\widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\},\\ e^{(2)}_{\mathbf{k}}&=\textstyle\sum_{\mathbf{n}\in\mathbf{Z}^2}\{\widetilde{p}_{2\mathbf{n}+(1,0)}\widetilde{v}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(1)}_{2\mathbf{n}+(1,0)}\widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(2)}_{2\mathbf{n}+(1,0)}\widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(3)}_{2\mathbf{n}+(1,0)}\widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\},\\ e^{(3)}_{\mathbf{k}}&=\textstyle\sum_{\mathbf{n}\in\mathbf{Z}^2}\{\widetilde{p}_{2\mathbf{n}+(0,1)}\widetilde{v}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(1)}_{2\mathbf{n}+(0,1)}\widetilde{e}^{(1)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(2)}_{2\mathbf{n}+(0,1)}\widetilde{e}^{(2)}_{\mathbf{k}-\mathbf{n}}+\widetilde{q}^{(3)}_{2\mathbf{n}+(0,1)}\widetilde{e}^{(3)}_{\mathbf{k}-\mathbf{n}}\}.\end{aligned}$$

Thus the decomposition algorithm is to decompose the original data $\mathcal{C}=\{v_{\mathbf{k}},e^{(1)}_{\mathbf{k}},e^{(2)}_{\mathbf{k}},e^{(3)}_{\mathbf{k}}\}_{\mathbf{k}}$ into $\{\widetilde{v}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(2)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(3)}_{\mathbf{k}}\}_{\mathbf{k}}$, and the reconstruction algorithm to recover $\mathcal{C}$ from $\{\widetilde{v}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(2)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(3)}_{\mathbf{k}}\}_{\mathbf{k}}$.

Next we represent multiresolution analysis and synthesis algorithms as templates. The key to do this is to associate the outputs $\{\widetilde{v}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(1)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(2)}_{\mathbf{k}}\}_{\mathbf{k}}$, $\{\widetilde{e}^{(3)}_{\mathbf{k}}\}_{\mathbf{k}}$ appropriately with the nodes of $\mathcal{G}$. Clearly, we should associate the "approximation" $\{\widetilde{v}_{\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ with $\mathcal{G}_4$ consisting of the *type V* nodes of $\mathcal{G}$ with labels $(2k_1,2k_2)$. Then we associate the "details" $\widetilde{e}^{(1)}_{\mathbf{k}},\widetilde{e}^{(2)}_{\mathbf{k}},\widetilde{e}^{(3)}_{\mathbf{k}}$ with the *type E* nodes with labels $(k_1-1,k_2-1),(k_1+1,k_2)$ and (k_1,k_2+1) resp. With such association, we can obtain the analysis and synthesis templates corresponding to analysis and synthesis algorithms, and vice versa.

When the decomposition and reconstruction algorithms (templates) are used to process surface with extraordinary vertices, these templates are required to have high symmetry. Roughly speaking,

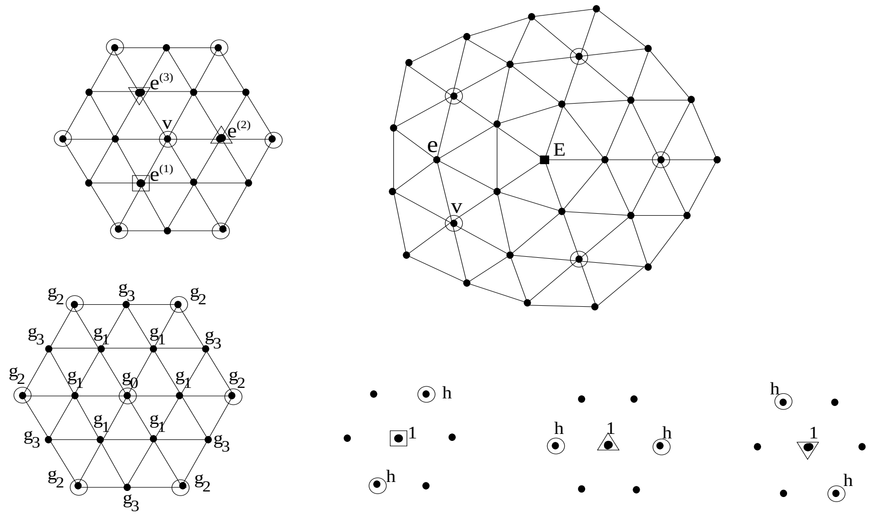

Figure 7: *Top-left: 3 types of type E nodes around a type V node; Top-right: Extraordinary vertex E; Bottom 1st picture (from left): Coefficients of lowpass filter; Bottom 2nd to 4th pictures: Coefficients of highpass filters*

these templates are independent of the orientations of the nodes. For example, the analysis templates for *type E* nodes $e^{(1)}, e^{(2)}, e^{(3)}$ around a *type V* node v in the top-left picture of Fig. 7 with analysis highpass filters $q^{(1)}, q^{(2)}, q^{(3)}$ must be identical. The reason is that to design multiresolution algorithms for extraordinary vertices, these 3 types of *type E* nodes must be treated uniformly. For example, in the top-right of Fig. 7, where E is an extraordinary node with valence 5 and e is a *type E* node around a regular *type V* node v, the algorithms for all *type E* nodes must be the same, whether they are of type 1, 2 or 3. Otherwise, we cannot design a consistent algorithm for E. Clearly, from this picture, we also see the template for the regular *type V* node v must be also orientation invariant. The biorthogonal filter banks provided in §3.2 do result in both analysis and synthesis templates with the required symmetry. Next, as an example, let us look at a very simple case.

Let $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ be the filter bank defined by $\frac{1}{4}B(2\boldsymbol{\omega})I_{00}(\boldsymbol{\omega})$, where $B(\boldsymbol{\omega})$ is defined by (3.22). Then the nonzero coefficients of $\widetilde{p}$ and these of $\widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}$ are shown in the bottom of Fig. 7 with $g_0 = \gamma, g_1 = \alpha, g_2 = \rho (= \tau(\alpha + 2\beta)), g_3 = \beta$ and $h = \tau$. When this filter bank is used as the analysis filter bank, then the pictures for coefficients of filters are the templates of the decomposition algorithm (up to a constant $\frac{1}{4}$). Clearly, the decomposition algorithm (equivalently the analysis template) for the *type V* node is orientate invariant, and all algorithms for *type E* nodes around v are the same and have certain symmetry. If the above $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ is used as the synthesis filter bank, then the reconstruction algorithm is

$$
\begin{aligned}
v_{\mathbf{k}} &= g_0 \widetilde{v}_{\mathbf{k}} + h(\widetilde{e}^{(1)}_{\mathbf{k}} + \widetilde{e}^{(1)}_{\mathbf{k}+(1,1)} + \widetilde{e}^{(2)}_{\mathbf{k}} + \widetilde{e}^{(2)}_{\mathbf{k}-(1,0)} + \widetilde{e}^{(3)}_{\mathbf{k}} + \widetilde{e}^{(3)}_{\mathbf{k}-(0,1)}) \\
&\quad + g_2(\widetilde{v}_{\mathbf{k}-(1,1)} + \widetilde{v}_{\mathbf{k}-(0,1)} + \widetilde{v}_{\mathbf{k}+(1,0)} + \widetilde{v}_{\mathbf{k}+(1,1)} + \widetilde{v}_{\mathbf{k}+(0,1)} + \widetilde{v}_{\mathbf{k}-(1,0)}), \\
e^{(1)}_{\mathbf{k}} &= \widetilde{e}^{(1)}_{\mathbf{k}} + g_1(\widetilde{v}_{\mathbf{k}} + \widetilde{v}_{\mathbf{k}-(1,1)}) + g_3(\widetilde{v}_{\mathbf{k}-(0,1)} + \widetilde{v}_{\mathbf{k}-(1,0)}), \\
e^{(2)}_{\mathbf{k}} &= \widetilde{e}^{(2)}_{\mathbf{k}} + g_1(\widetilde{v}_{\mathbf{k}} + \widetilde{v}_{\mathbf{k}+(1,0)}) + g_3(\widetilde{v}_{\mathbf{k}-(0,1)} + \widetilde{v}_{\mathbf{k}+(1,1)}), \\
e^{(3)}_{\mathbf{k}} &= \widetilde{e}^{(3)}_{\mathbf{k}} + g_1(\widetilde{v}_{\mathbf{k}} + \widetilde{v}_{\mathbf{k}+(0,1)}) + g_3(\widetilde{v}_{\mathbf{k}+(1,1)} + \widetilde{v}_{\mathbf{k}-(1,0)}).
\end{aligned}
$$

Thus the reconstruction algorithms can be expressed by two templates to update v and to calculate e as shown on the left and right of Fig. 8 resp. (The algorithms to calculate $e^{(1)}_{\mathbf{k}}, e^{(2)}_{\mathbf{k}}, e^{(3)}_{\mathbf{k}}$ are identical, and we need only one template to represent these algorithms.) Clearly the templates are also highly symmetric.

As shown by the above simple example, the biorthogonal filter banks provided in §3.2 result in

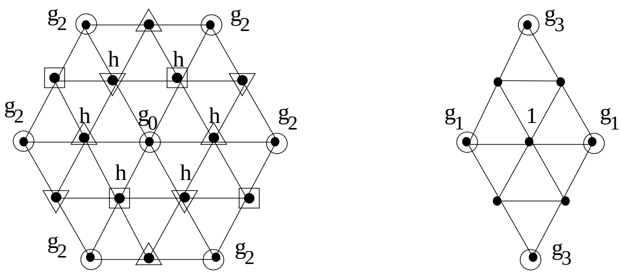

Figure 8: *Left: Template to reconstruct type V node v; Right: Template to reconstruct type E node e*

both analysis and synthesis templates with the required symmetry. Thus, based on these templates one can design the algorithms for extraordinary vertices. In the rest of this subsection, we show that multiresolution algorithms resulted from the biorthogonal filter banks built with blocks $D(\boldsymbol{\omega})$ can be described in a simpler way, as that in [3].

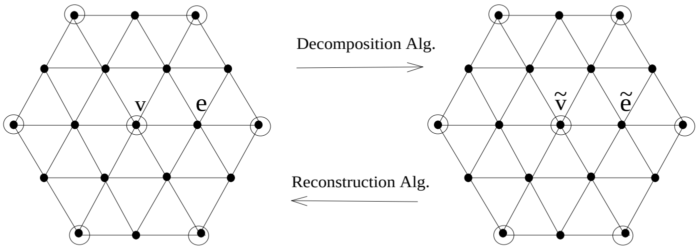

Figure 9: *Decomposition and reconstruction algorithms*

Since 6-fold symmetric filter banks result in 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 $e^{(1)}_{\mathbf{k}}, e^{(2)}_{\mathbf{k}}, e^{(3)}_{\mathbf{k}}$ are also identical, we may simply let e denote the original data associated with the *type* E nodes, and use $\widetilde{e}$ to denote the "details" after the decomposition algorithm. Thus, the decomposition algorithm is to decompose the original data $\{v\} \cup \{e\}$ into $\{\widetilde{v}\}$ and $\{\widetilde{e}\}$, and the reconstruction algorithm to recover $\{v\} \cup \{e\}$ from $\{\widetilde{v}\}$ and $\{\widetilde{e}\}$, see Fig. 9. In the following we simply use v, e and $\widetilde{v}, \widetilde{e}$ to describe the algorithms.

For given $\mathcal{C}$ sampling on $\mathcal{G}$ (or equivalently, for given $\{v\}$ and $\{e\}$), the multiresolution decomposition algorithm is given by (3.42)-(3.45) and shown in Fig. 10, where a, b, d, w, u, s and r are constants to be determined. Namely, first we replace all v associated with *type* V nodes of $\mathcal{G}_4$ by v'' given by formula (3.42). Then, based on v'' obtained, we replace all e associated with *type* E nodes in $\mathcal{G}\backslash\mathcal{G}_4$ by e'' given in formula (3.43). After that, based on e'' obtained in Step 2, all v'' in Step 1 are updated by $\widetilde{v}$ given in formula (3.44). Finally, based on $\widetilde{v}$ obtained in Step 3, all e'' in Step 2 are updated by $\widetilde{e}$ given in formula (3.45). The algorithm is simple and efficient.

Decomposition Algorithm:

Step 1. $v'' = \frac{1}{b}(v - d\sum_{k=0}^{5} e_k)$ (3.42)

Step 2. $e'' = e - a(v''_0 + v''_1) - c(v''_2 + v''_3)$ (3.43)

Step 3. $\widetilde{v} = v'' - w\sum_{k=0}^{5} e''_k - u\sum_{k=6}^{11} e''_k$ (3.44)

Step 4. $\widetilde{e} = e'' - s(\widetilde{v}_0 + \widetilde{v}_1) - r(\widetilde{v}_2 + \widetilde{v}_3)$. (3.45)

The multiresolution reconstruction algorithm is given by (3.46)-(3.49) and shown in Fig. 11, where a, b, d, w, u, s and r are the same constants in the multiresolution decomposition algorithm.

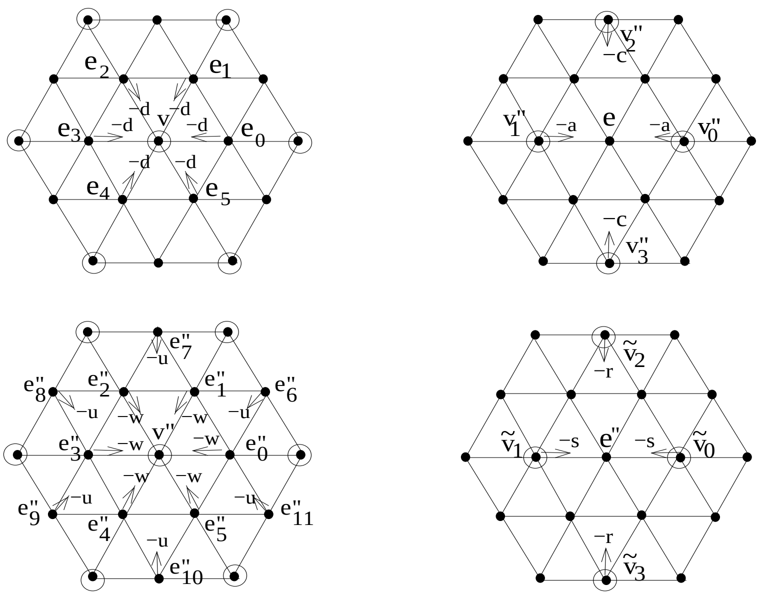

Figure 10: *Top-left: Decomposition Alg. Step 1; Top-right: Decomposition Alg. Step 2; Bottom-left: Decomposition Alg. Step 3; Bottom-right: Decomposition Alg. Step 4*

The reconstruction algorithm is the reverse algorithm of the decomposition algorithm. More precisely, first we update all $\widetilde{e}$ associated with *type E* nodes of $\mathcal{G}\backslash\mathcal{G}_4$ with the resulting e'' given by in (3.46). Then, based on e'' obtained, we replace all $\widetilde{v}$ associated with *type V* nodes of $\mathcal{G}_4$ by v'' given by formula (3.47). After that, based on v'' obtained, we update all e'' obtained in Step 1 by e given in (3.48). Finally, based on e obtained in Step 3, all v'' in Step 2 are updated with the resulting v given by formula (3.49). Again, the reconstruction algorithm from $\widetilde{v}, \widetilde{e}$ to v, e is simple and efficient.

Reconstruction Algorithm:

Step 1. $e'' = \widetilde{e} + s(\widetilde{v}_0 + \widetilde{v}_1) + r(\widetilde{v}_2 + \widetilde{v}_3)$ (3.46)

Step 2. $v'' = \widetilde{v} + w\sum_{k=0}^{5} e''_k + u\sum_{k=6}^{11} e''_k$ (3.47)

Step 3. $e = e'' + a(v''_0 + v''_1) + c(v''_2 + v''_3)$ (3.48)

Step 4. $v = bv'' + d\sum_{k=0}^{5} e_k.$ (3.49)

When the constants a, b, c, d, u, w, s, r are appropriately chosen, the decomposed $\{\widetilde{v}\}$ is the "approximation" of the initial data $\mathcal{C}$, and $\{\widetilde{e}\}$ is the "detail" of $\mathcal{C}$. The decomposition algorithm can be applied repeatedly to the "approximation" to get further "approximation" and "details" of the data, and reconstruction algorithm assures that the original data can be recovered exactly from the "approximation" and "details".

When $\widetilde{e} = 0$, the reconstruction algorithm is the subdivision algorithm to produce finer and finer meshes from the initial mesh with vertices $\widetilde{v}$. The subdivision schemes obtained by such an algorithm as that in (3.46)-(3.49) are studied in [30] and [29], where such schemes are called the composite subdivision schemes.

When $s = r = 0$, the above decomposition algorithm consists of three steps: Step 1 to Step 3 of (3.42)-(3.44) (with $\widetilde{e} = e''$), and the corresponding reconstruction algorithm also consists of three steps: Step 2 to Step 4 of (3.47)-(3.49) (with $e'' = \widetilde{e}$). These algorithms are proposed in [3], where d, b, a, c are chosen to be

$$d = \frac{1}{10},\ b = \frac{2}{5},\ a = \frac{3}{8},\ c = \frac{1}{8}, \tag{3.50}$$

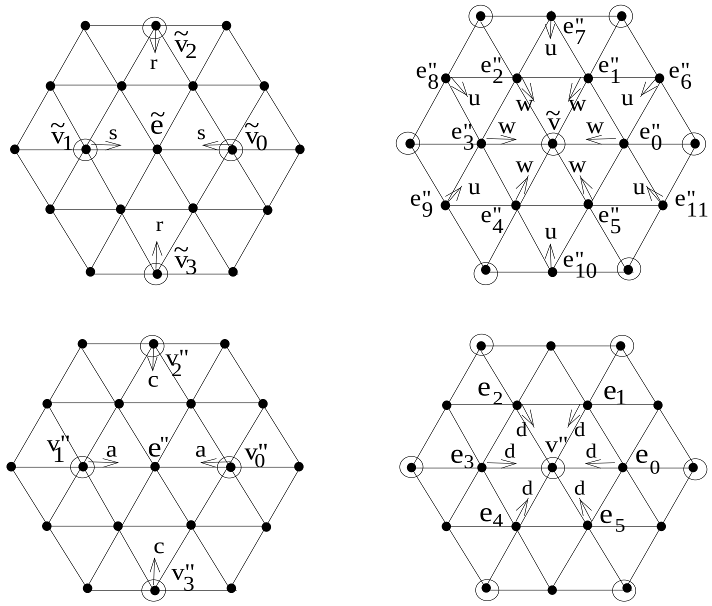

Figure 11: *Top-left: Reconstruction Alg. Step 1; Top-right: Reconstruction Alg. Step 2; Bottom-left: Reconstruction Alg. Step 3; Bottom-right: Reconstruction Alg. Step 4*

and $w = -0.284905, s = 0.071591$. For such choices of d, b, a, c, the corresponding subdivision scheme is the Loop's scheme. Thus, the wavelets in [3] associated with these algorithms are called the Loop-scheme based wavelets.

With the formulas in (3.40) and (3.41), and by careful calculations, one can obtain the filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ corresponding to the algorithms (3.42)-(3.49) with $s = r = 0$ are exactly these given by (3.32) in Example 2 with

$$a = \alpha,\ c = \beta,\ d = \tau,\ b = \gamma - 6\alpha\tau. \tag{3.51}$$

When d, b, a, c are given by (3.50), $\alpha = \frac{3}{8}, \beta = \frac{1}{8}, \tau = \frac{1}{10}$. From Example 2, we know the corresponding analysis scaling function ϕ is not in $L^2(\mathbb{R}^2)$. Therefore, these filter banks cannot generate biorthogonal bases for $L^2(\mathbb{R}^2)$. Hence, we should use other parameters. For example, if the parameters given by (3.34) in Example 2 are used, then d, b, a, c, w, u are

$$d = \frac{7}{128},\ b = \frac{43}{64},\ a = \frac{111}{256},\ c = \frac{17}{256},\ w = -\frac{1}{4},\ u = \frac{57}{512}.$$

In this case, the algorithms generate the biorthogonal bases for $L^2(\mathbb{R}^2)$ and the corresponding $\widetilde{\phi}, \widetilde{\psi}^{(j)}, j = 1, 2, 3$ are in C^1.

With the formulas in (3.40) and (3.41) again, we also find the filter banks $\{p, q^{(1)}, q^{(2)}, q^{(3)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}, \widetilde{q}^{(3)}\}$ associated with the algorithms (3.42)-(3.49) with nonzero s, r are these given by (3.35) in Example 3 with d, b, a, c defined by (3.51) and $w_1 = -s, u_1 = -r$. From Example 3, we may use the filter banks with parameters given by (3.38). In this case, the corresponding d, b, a, c, w, u, s, r are given by

$$\begin{aligned} &d = 0.104524,\ b = 0.494004,\ a = 0.461600,\ c = 0.021700, \\ &w = -0.267159,\ u = 0.116028,\ s = 0.026600,\ r = 0.108622. \end{aligned}$$

To obtain scaling functions and wavelets with a high approximation order or smooth order, we may use more steps as the above algorithms (3.42)-(3.49) with more parameters. The corresponding filter banks are given as those in (3.35) of Example 3 but with more blocks $D(2\boldsymbol{\omega})$ and/or $\widetilde{D}(2\boldsymbol{\omega})$.

With the filter banks available, then one may use the method discussed in Examples 1-3 to choose suitable parameters.

4 $\sqrt{3}$-refinement wavelets with 6-fold axial symmetry

In this section we consider the $\sqrt{3}$-refinement biorthogonal wavelets. In §4.1, we recall a family of 6-fold symmetric biorthogonal $\sqrt{3}$-refinement FIR filter banks in [20] and present a result on the symmetry of the associated scaling functions and wavelets. In §4.2 we show that the multiresolution algorithms from 6-fold symmetric $\sqrt{3}$-refinement FIR filter banks can be described by templates.

4.1 Biorthogonal $\sqrt{3}$-refinement FIR filter banks and wavelets with 6-fold axial symmetry

For a ($\sqrt{3}$-refinement) FIR filter bank $\{p, q^{(1)}, q^{(2)}\}$ with dilation matrix M, with $q^{(0)}(\boldsymbol{\omega}) = p(\boldsymbol{\omega})$, we write $q^{(\ell)}(\boldsymbol{\omega})$ as

$$q^{(\ell)}(\boldsymbol{\omega}) = \frac{1}{\sqrt{3}}(q_0^{(\ell)}(M^T\boldsymbol{\omega}) + q_1^{(\ell)}(M^T\boldsymbol{\omega})e^{-i\omega_1} + q_2^{(\ell)}(M^T\boldsymbol{\omega})e^{i\omega_1}),$$

where $q_k^{(\ell)}(\boldsymbol{\omega}), 0 \le k \le 2$ are trigonometric polynomials. Let $V(\boldsymbol{\omega})$ denote the polyphase matrix (with dilation matrix M) of $\{p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega})\}$:

$$V(\boldsymbol{\omega}) = \begin{bmatrix} p_0(\boldsymbol{\omega}) & p_1(\boldsymbol{\omega}) & p_2(\boldsymbol{\omega}) \\ q_0^{(1)}(\boldsymbol{\omega}) & q_1^{(1)}(\boldsymbol{\omega}) & q_2^{(1)}(\boldsymbol{\omega}) \\ q_0^{(2)}(\boldsymbol{\omega}) & q_1^{(2)}(\boldsymbol{\omega}) & q_2^{(2)}(\boldsymbol{\omega}) \end{bmatrix}. \tag{4.1}$$

Then

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

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

$$I_0(\boldsymbol{\omega}) = [1, e^{-i\omega_1}, e^{i\omega_1}]^T. \tag{4.2}$$

First we recall the following two propositions on the characterizations of 6-fold symmetric filter banks in [20].

Proposition 4.[20] *A filter bank $\{p, q^{(1)}, q^{(2)}\}$ has 6-fold axial symmetry if and only if it satisfies*

$$[p, q^{(1)}, q^{(2)}]^T(R_1^{-T}\boldsymbol{\omega}) = N_1(\boldsymbol{\omega})[p, q^{(1)}, q^{(2)}]^T(\boldsymbol{\omega}), \tag{4.3}$$
$$[p, q^{(1)}, q^{(2)}]^T(L_0\boldsymbol{\omega}) = N_2(\boldsymbol{\omega})[p, q^{(1)}, q^{(2)}]^T(\boldsymbol{\omega}), \tag{4.4}$$

where

$$N_1(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & e^{-i(2\omega_1+\omega_2)} \\ 0 & e^{i(2\omega_1+\omega_2)} & 0 \end{bmatrix}, \ N_2(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{i(\omega_1-\omega_2)} & \\ 0 & 0 & e^{-i(\omega_1-\omega_2)} \end{bmatrix}. \tag{4.5}$$

Proposition 5.[20] *An FIR filter bank $\{p, q^{(1)}, q^{(2)}\}$ has 6-fold axial symmetry if and only if its polyphase matrix $V(\boldsymbol{\omega})$ (with dilation matrix $M = M_1$) satisfies*

$$V(R_1^{-T}\boldsymbol{\omega}) = N_0(\boldsymbol{\omega})V(\boldsymbol{\omega})N_0(\boldsymbol{\omega}), \tag{4.6}$$
$$V(L_0\boldsymbol{\omega}) = J_0V(\boldsymbol{\omega})J_0, \tag{4.7}$$

where

$$N_0(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & e^{-i\omega_1} \\ 0 & e^{i\omega_1} & 0 \end{bmatrix}, \ J_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4.8}$$

Based on Proposition 4, we have the following result on the symmetry of the scaling function and wavelets associated with a symmetric filter bank.

Proposition 6. *Suppose an FIR filter bank $\{p, q^{(1)}, q^{(2)}\}$ has 6-fold axial symmetry. Let ϕ be the associated scaling function with dilation matrix $M = M_1$ and $\psi^{(\ell)}, \ell = 1, 2$ be the functions define by (2.11) with $q^{(\ell)}$. Then*

$$\phi(L_j\mathbf{x}) = \phi(\mathbf{x}), \ 0 \le j \le 5, \tag{4.9}$$

$$\psi^{(2)}(\mathbf{x}) = \psi^{(1)}(-\mathbf{x}), \tag{4.10}$$

and

$$\psi^{(1)}(L_0\mathbf{x}) = \psi^{(1)}(-\mathbf{x}), \psi^{(1)}(L_2\mathbf{x}) = \psi^{(1)}((1,0) - \mathbf{x}), \psi^{(1)}(L_4\mathbf{x}) = \psi^{(1)}((0,1) - \mathbf{x}). \tag{4.11}$$

Proof. For (4.9), we need only to prove $\phi(R_1\mathbf{x}) = \phi(\mathbf{x})$ and $\phi(L_0\mathbf{x}) = \phi(\mathbf{x})$. From (2.10), we have $\widehat{\phi}(\boldsymbol{\omega}) = p(M^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}\boldsymbol{\omega})$. Thus $\widehat{\phi}(\boldsymbol{\omega}) = \Pi_{k=1}^{\infty} p((M^{-T})^k\boldsymbol{\omega})\widehat{\phi}(0)$. When $M = M_1$ given in (2.2), we have

$$MR_1 = R_1M, \ ML_0 = L_0R_1M.$$

Thus $M^kR_1 = R_1M^k, M^kL_0 = L_0R_1^kM^k$, which implies $(M^{-T})^kR_1^{-T} = R_1^{-T}(M^{-T})^k$, $(M^{-T})^kL_0 = L_0(R_1^{-T})^k(M^{-T})^k$. Therefore,

$$\begin{aligned}
&\widehat{\phi}(R_1^{-T}\boldsymbol{\omega}) = \Pi_{k=1}^{\infty} p((M^{-T})^kR_1^{-T}\boldsymbol{\omega})\widehat{\phi}(0) \\
&= \Pi_{k=1}^{\infty} p(R_1^{-T}(M^{-T})^k\boldsymbol{\omega})\widehat{\phi}(0) \\
&= \Pi_{k=1}^{\infty} p((M^{-T})^k\boldsymbol{\omega})\widehat{\phi}(0) = \widehat{\phi}(\boldsymbol{\omega}),
\end{aligned}$$

and

$$\begin{aligned}
&\widehat{\phi}(L_0\boldsymbol{\omega}) = \Pi_{k=1}^{\infty} p((M^{-T})^kL_0\boldsymbol{\omega})\widehat{\phi}(0) \\
&= \Pi_{k=1}^{\infty} p(L_0(R_1^{-T})^k(M^{-T})^k\boldsymbol{\omega})\widehat{\phi}(0) \\
&= \Pi_{k=1}^{\infty} p((M^{-T})^k\boldsymbol{\omega})\widehat{\phi}(0) = \widehat{\phi}(\boldsymbol{\omega}).
\end{aligned}$$

Hence, $\phi(R_1\mathbf{x}) = \phi(\mathbf{x})$ and $\phi(L_0\mathbf{x}) = \phi(\mathbf{x})$, and (4.9) holds.

Next, let us consider (4.10). From (2.11), we have

$$\widehat{\psi}^{(\ell)}(\boldsymbol{\omega}) = q^{(\ell)}(M^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}\boldsymbol{\omega}), \ \ell = 1, 2. \tag{4.12}$$

Thus, $q^{(2)}(-\boldsymbol{\omega}) = q^{(1)}(\boldsymbol{\omega})$ and $\phi(-\mathbf{x}) = \phi(\mathbf{x})$ lead to that

$$\widehat{\psi}^{(2)}(-\boldsymbol{\omega}) = q^{(2)}(-M^{-T}\boldsymbol{\omega})\widehat{\phi}(-M^{-T}\boldsymbol{\omega}) = q^{(1)}(M^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}\boldsymbol{\omega}) = \widehat{\psi}^{(1)}(\boldsymbol{\omega}),$$

which is $\psi^{(2)}(-\mathbf{x}) = \psi^{(1)}(\mathbf{x})$, as desired.

For (4.11), here we give the proof for the second equation. From $ML_2 = R_1^2L_0R_1M$ and (4.12) with $\ell = 1$, we have

$$\begin{aligned}
&\widehat{\psi}^{(1)}(L_2^{-T}\boldsymbol{\omega}) = q^{(1)}(M^{-T}L_2^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}L_2^{-T}\boldsymbol{\omega}) \\
&= q^{(1)}((R_1^{-T})^2L_0R_1^{-T}M^{-T}\boldsymbol{\omega})\widehat{\phi}((R_1^{-T})^2L_0R_1^{-T}M^{-T}\boldsymbol{\omega}) \\
&= q^{(1)}((R_1^{-T})^2L_0R_1^{-T}M^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}\boldsymbol{\omega}).
\end{aligned}$$

From (4.3) and (4.4), one can get

$$q^{(1)}((R_1^{-T})^2 L_0 R_1^{-T}\boldsymbol{\omega}) = e^{-i(2\omega_1+\omega_2)} q^{(2)}(\boldsymbol{\omega}) = e^{-i(2\omega_1+\omega_2)} q^{(1)}(-\boldsymbol{\omega}),$$

which implies that $q^{(1)}((R_1^{-T})^2 L_0 R_1^{-T} M^{-T}\boldsymbol{\omega}) = e^{-i\omega_1} q^{(1)}(-M^{-T}\boldsymbol{\omega})$. Thus

$$\begin{aligned}&\psi^{(1)}(L_2^{-T}\boldsymbol{\omega}) = e^{-i\omega_1} q^{(1)}(-M^{-T}\boldsymbol{\omega})\widehat{\phi}(M^{-T}\boldsymbol{\omega})\\ &= e^{-i\omega_1} q^{(1)}(-M^{-T}\boldsymbol{\omega})\widehat{\phi}(-M^{-T}\boldsymbol{\omega}) = e^{-i\omega_1}\widehat{\psi}^{(1)}(-\boldsymbol{\omega}).\end{aligned}$$

Therefore, we have the second equation in (4.11).

The proof for the first and third equations in (4.11) is similar. For the proof of these two equations, instead of using $ML_2 = R_1^2 L_0 R_1 M$ for the proof of the second equation, we should use $ML_0 = L_0 R_1 M$ and $ML_4 = -R_1 L_0 R_1 M$. The details of the proof are omitted here. $\diamondsuit$

Based on Proposition 5, one can easily construct blocks to build symmetric filter banks. For example, [20] uses

$$W(\boldsymbol{\omega}) = \begin{bmatrix} \gamma + \rho(x + xy + y + \frac{1}{x} + \frac{1}{xy} + \frac{1}{y}) & \alpha(1 + \frac{1}{x} + y) & \alpha(1 + x + \frac{1}{y}) \\ \frac{\rho}{2\alpha}(1 + x + \frac{1}{y}) & 1 & 0 \\ \frac{\rho}{2\alpha}(1 + \frac{1}{x} + y) & 0 & 1 \end{bmatrix}, \tag{4.13}$$

where α, ρ, γ are constants with $\alpha \neq 0, \gamma \neq 3\rho$, and

$$Z(\boldsymbol{\omega}) = \begin{bmatrix} 1 & -w(1 + \frac{1}{x} + y) - u(xy + \frac{1}{xy} + \frac{y}{x}) & -w(1 + x + \frac{1}{y}) - u(xy + \frac{1}{xy} + \frac{x}{y}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.14}$$

where w, u are constants. Both $W(\boldsymbol{\omega})$ and $Z(\boldsymbol{\omega})$ satisfy (4.6) and (4.7). Furthermore, the inverses of $W(\boldsymbol{\omega})$ and $Z(\boldsymbol{\omega})$ are matrices whose entries are also polynomials of x, y. More precisely, $\widetilde{W}(\boldsymbol{\omega}) = (W(\boldsymbol{\omega})^{-1})^*$ is given by

$$\widetilde{W}(\boldsymbol{\omega}) = \frac{1}{\gamma - 3\rho} \times \begin{bmatrix} 1 & -\frac{\rho}{2\alpha}(1 + \frac{1}{x} + y) & -\frac{\rho}{2\alpha}(1 + x + \frac{1}{y}) \\ -\alpha(1 + x + \frac{1}{y}) & \gamma - \frac{3}{2}\rho + \frac{\rho}{2}(x + xy + y + \frac{1}{x} + \frac{1}{xy} + \frac{1}{y}) & \frac{\rho}{2}(1 + x + \frac{1}{y})^2 \\ -\alpha(1 + \frac{1}{x} + y) & \frac{\rho}{2}(1 + \frac{1}{x} + y)^2 & \gamma - \frac{3}{2}\rho + \frac{\rho}{2}(x + xy + y + \frac{1}{x} + \frac{1}{xy} + \frac{1}{y}) \end{bmatrix}, \tag{4.15}$$

and $\widetilde{Z}(\boldsymbol{\omega}) = (Z(\boldsymbol{\omega})^{-1})^*$ is given by

$$\widetilde{Z}(\boldsymbol{\omega}) = \begin{bmatrix} 1 & 0 & 0 \\ w(1 + x + \frac{1}{y}) + u(xy + \frac{1}{xy} + \frac{x}{y}) & 1 & 0 \\ w(1 + \frac{1}{x} + y) + u(1 + \frac{1}{xy} + \frac{y}{x}) & 0 & 1 \end{bmatrix}. \tag{4.16}$$

Clearly both $\widetilde{W}(\boldsymbol{\omega})$ and $\widetilde{Z}(\boldsymbol{\omega})$ also satisfy (4.6) and (4.7). Therefore, if FIR filter banks $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ are given by

$$\begin{aligned}[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega})]^T &= U_n(M^T\boldsymbol{\omega})U_{n-1}(M^T\boldsymbol{\omega})\cdots U_0(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}), \qquad (4.17)\\ [\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega})]^T &= \frac{1}{3}\widetilde{U}_n(M^T\boldsymbol{\omega})\widetilde{U}_{n-1}(M^T\boldsymbol{\omega})\cdots \widetilde{U}_0(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega})\end{aligned}$$

for some $n \in \mathbf{Z}_+$, where $I_0(\boldsymbol{\omega})$ is defined by (4.2), each $U_k(\boldsymbol{\omega})$ is a $W(\boldsymbol{\omega})$ in (4.13) or a $\widetilde{W}(\boldsymbol{\omega})$ in (4.15) for some parameters a_k, b_k, d_k, and $\widetilde{U}_k(\boldsymbol{\omega}) = (U_k(\boldsymbol{\omega})^{-1})^*$ is the corresponding $\widetilde{W}(\boldsymbol{\omega})$ in (4.15) or $W(\boldsymbol{\omega})$ in (4.13), then their polyphase matrices are respectively $\sqrt{3}U_n(\boldsymbol{\omega})U_{n-1}(\boldsymbol{\omega})\cdots U_0(\boldsymbol{\omega})$ and

$\frac{1}{\sqrt{3}}\widetilde{U}_n(\boldsymbol{\omega})\widetilde{U}_{n-1}(\boldsymbol{\omega})\cdots\widetilde{U}_0(\boldsymbol{\omega})$, both satisfying (4.6) and (4.7). Hence, $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ are biorthogonal FIR filter banks and both have 6-fold axial symmetry.

Based on this family of biorthogonal filter banks, one can construct biorthogonal $\sqrt{3}$-refinement wavelets. For example, let $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ be the biorthogonal filter banks given by

$$\begin{aligned}
[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega})]^T &= Z(M^T\boldsymbol{\omega})\widetilde{W}(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}),\\
[\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega})]^T &= \frac{1}{3}\widetilde{Z}(M^T\boldsymbol{\omega})W(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}),
\end{aligned} \tag{4.18}$$

where $W(\boldsymbol{\omega})$, $\widetilde{W}(\boldsymbol{\omega}), Z(\boldsymbol{\omega})$, and $\widetilde{Z}(\boldsymbol{\omega})$ are given by (4.13), (4.15), (4.14), and (4.16) for some parameters α, ρ, γ and w, u respectively. Then, as discussed in [20], when

$$\alpha = \frac{1}{3},\ \gamma = 1 - 6\rho,\ w = -\frac{1}{9} - \frac{1}{2}\rho - u, \tag{4.19}$$

both $\widetilde{p}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ have sum rule order 2. Furthermore, if $\rho = \frac{1}{18}$, then $\widetilde{p}(\boldsymbol{\omega})$ has sum rule order 3 and $\widetilde{p}(\boldsymbol{\omega})$ is the subdivision mask in [23]. However, for this $\widetilde{p}(\boldsymbol{\omega})$, for any choice of u, the analysis scaling function ϕ is not in $L^2(\mathbb{R}^2)$. When

$$\rho = \frac{1}{27},\ u = \frac{1}{10}, \tag{4.20}$$

the resulting $\phi \in W^{0.0104}$ and $\widetilde{\phi} \in W^{1.9801}$. We check numerically that the resulting scaling functions $\widetilde{\phi}$ are in C^1. A few more examples are considered in [20]. Next, we consider another example.

Example 4. *Let $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ be the biorthogonal filter banks given by*

$$\begin{aligned}
[p(\boldsymbol{\omega}), q^{(1)}(\boldsymbol{\omega}), q^{(2)}(\boldsymbol{\omega})]^T &= \widetilde{Z}_1(M^T\boldsymbol{\omega})Z(M^T\boldsymbol{\omega})\widetilde{W}(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}),\\
[\widetilde{p}(\boldsymbol{\omega}), \widetilde{q}^{(1)}(\boldsymbol{\omega}), \widetilde{q}^{(2)}(\boldsymbol{\omega})]^T &= \frac{1}{3}Z_1(M^T\boldsymbol{\omega})\widetilde{Z}(M^T\boldsymbol{\omega})W(M^T\boldsymbol{\omega})I_0(\boldsymbol{\omega}),
\end{aligned} \tag{4.21}$$

where $W(\boldsymbol{\omega})$, $\widetilde{W}(\boldsymbol{\omega}), Z(\boldsymbol{\omega})$, and $\widetilde{Z}(\boldsymbol{\omega})$ are given by (4.13), (4.15), (4.14), and (4.16) for some parameters α, ρ, γ and w, u respectively, and $\widetilde{Z}_1(\boldsymbol{\omega})$, and $Z_1(\boldsymbol{\omega})$ are given by (4.16), and (4.14) with parameters w_1, u_1.

When

$$\begin{aligned}
\gamma &= \tfrac{1}{(w_1+u_1)}(u_1 - 2u_1\alpha - \tfrac{2}{3} + 2\alpha - 2w_1\alpha + w_1),\\
\rho &= \tfrac{1}{3(w_1+u_1)}(\tfrac{1}{3} - \alpha + u_1 + w_1 - 2w_1\alpha - 2u_1\alpha),\\
u &= -\tfrac{1}{18\alpha(w_1+u_1)}(\tfrac{1}{3} - \alpha + u_1 + w_1 + 18u_1w\alpha + 18w_1w\alpha),
\end{aligned} \tag{4.22}$$

both $\widetilde{p}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ have sum rule order 2. Furthermore, if

$$\begin{aligned}
u_1 &= \tfrac{1}{\alpha(1-6\alpha)}(\tfrac{4}{3} - 6\alpha + 3w_1 + 6w_1\alpha^2 - 10w_1\alpha + 6\alpha^2),\\
w &= -\tfrac{1}{486\alpha(w_1+u_1)^2}\{2 - 12\alpha + (21 - 69\alpha)(u_1 + w_1) + 18(w_1 + u_1 + 1)\alpha^2 + 45(u_1 + w_1)^2\},
\end{aligned} \tag{4.23}$$

then $\widetilde{p}(\boldsymbol{\omega})$ has sum rule order 3. If $\alpha = \frac{17}{64}, w_1 = -\frac{13}{64}$, then the resulting $\phi \in W^{0.0478}$ and $\widetilde{\phi} \in W^{2.8331}$; and if we choose $\alpha = \frac{25}{81}, w_1 = -\frac{11}{81}$, then the corresponding $\phi \in W^{0.1896}$ and $\widetilde{\phi} \in W^{2.4556}$. We also can choose α, w_1 such that $\widetilde{\phi}$ and ϕ have similar smoothness orders. For example, if we choose $\alpha = \frac{3}{10}, w_1 = -\frac{1}{5}$, then the resulting $\phi \in W^{1.4213}$ and $\widetilde{\phi} \in W^{1.4914}$.

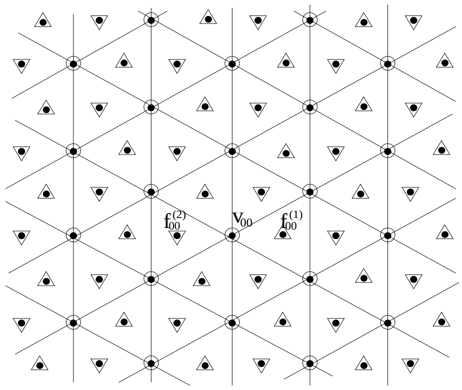

Figure 12: *Type V nodes, two types of type F nodes with* $\triangle$ *and* ∇

4.2 $\sqrt{3}$ multiresolution algorithms for triangle surface processing

In this subsection, we show how $\sqrt{3}$-refinement decomposition and reconstruction algorithms can be represented as templates by associating the outputs $c^1_{\mathbf{k}}, d^{(1,1)}_{\mathbf{k}}, d^{(2,1)}_{\mathbf{k}}$ after 1-level $\sqrt{3}$ decomposition appropriately with the nodes of $\mathcal{G}$. Then we show that some filter banks presented in §4.1 result in simple algorithms given by templates.

Let $M = M_1$ be the matrix defined by (2.2). For $\mathbf{k} = (k_1, k_2) \in \mathbf{Z}^2$, $M\mathbf{k} = (2k_1 - k_2, k_1 + k_2)$ are the labels for $(2k_1 - k_2)\mathbf{v}_1 + (k_1 + k_2)\mathbf{v}_2$, the *type V* nodes, the nodes of $\mathcal{G}_3$. We separate *type F* nodes, the nodes in $\mathcal{G}\backslash\mathcal{G}_3$, into two groups with labels in:

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

See Fig. 12, where $\triangle$ and ∇ denote these two groups of *type F* nodes (the big circles denote *type V* nodes).

Let $\mathcal{C} = \{c_{\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ be the data sampled on $\mathcal{G}$. Thus $\{c_{M\mathbf{k}}\}_{\mathbf{k}\in\mathbf{Z}^2}$ is the set of data associated with *type V* nodes of $\mathcal{G}_3$, $\{c_{M\mathbf{k}-(1,0)}\}_{\mathbf{k}\in\mathbf{Z}^2}$, $\{c_{M\mathbf{k}+(1,0)}\}_{\mathbf{k}\in\mathbf{Z}^2}$ are two sets of data associated with two groups of *type F* nodes. Denote

$$v_{\mathbf{k}} = c_{M\mathbf{k}},\ f^{(1)}_{\mathbf{k}} = c_{M\mathbf{k}-(1,0)},\ f^{(2)}_{\mathbf{k}} = c_{M\mathbf{k}+(1,0)},\ \mathbf{k} \in \mathbf{Z}^2. \tag{4.24}$$

The $\sqrt{3}$-refinement multiresolution decomposition algorithm is to decompose the original data $\mathcal{C} = \{c_{\mathbf{k}}\}_{\mathbf{k}}$ with the analysis filter bank into the "approximation" $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$ and the "details" $\{d^{(1,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ $\{d^{(2,1)}_{\mathbf{k}}\}_{\mathbf{k}}$, and the (prefect) multiresolution reconstruction algorithm is to recover $\mathcal{C}$ from $\{c^1_{\mathbf{k}}\}_{\mathbf{k}}$, $\{d^{(1,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ $\{d^{(2,1)}_{\mathbf{k}}\}_{\mathbf{k}}$ with the synthesis filter bank. Denote

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

Then, the decomposition algorithm can be written as

$$\widetilde{v}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbf{Z}^2} p_{\mathbf{k}'-M\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{f}^{(1)}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(1)}_{\mathbf{k}'-M\mathbf{k}} c_{\mathbf{k}'},\ \widetilde{f}^{(2)}_{\mathbf{k}} = \frac{1}{3}\sum_{\mathbf{k}'\in\mathbf{Z}^2} q^{(2)}_{\mathbf{k}'-M\mathbf{k}} c_{\mathbf{k}'} \tag{4.25}$$

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

$$c_{\mathbf{k}} = \sum_{\mathbf{k}'\in\mathbf{Z}^2} \{\widetilde{p}_{\mathbf{k}-M\mathbf{k}'} \widetilde{v}_{\mathbf{k}'} + \widetilde{q}^{(1)}_{\mathbf{k}-2\mathbf{k}'} \widetilde{f}^{(1)}_{\mathbf{k}'} + \widetilde{q}^{(2)}_{\mathbf{k}-M\mathbf{k}'} \widetilde{f}^{(2)}_{\mathbf{k}'}\}, \tag{4.26}$$

where $p_{\mathbf{k}}, q^{(1)}_{\mathbf{k}}, q^{(2)}_{\mathbf{k}}, \mathbf{k} \in \mathbf{Z}^2$ and $\widetilde{p}_{\mathbf{k}}, \widetilde{q}^{(1)}_{\mathbf{k}}, \widetilde{q}^{(2)}_{\mathbf{k}}, \mathbf{k} \in \mathbf{Z}^2$ are the coefficients of the analysis filter bank and the synthesis filter banks respectively.

We associate the "details" $\widetilde{f}^{(1)}_{\mathbf{k}}$ and $\widetilde{f}^{(2)}_{\mathbf{k}}$ with the *type F* nodes with labels $M\mathbf{k} + (1,0)$ and $M\mathbf{k} - (1,0)$ respectively. After separating $\mathcal{C} = \{c_{\mathbf{k}}\}$ into three groups $\{\mathbf{v}_{\mathbf{k}}\}, \{f^{(1)}_{\mathbf{k}}\}, \{f^{(2)}_{\mathbf{k}}\}$ and associating $\widetilde{\mathbf{v}}_{\mathbf{k}}, \widetilde{f}^{(1)}_{\mathbf{k}}, \widetilde{f}^{(2)}_{\mathbf{k}}$ with the suitable nodes as above, then we can represent $\sqrt{3}$ decomposition and reconstruction algorithms as templates. Furthermore, 6-fold symmetric biorthogonal $\sqrt{3}$-refinement filter banks result in templates independent of the orientation of nodes. Thus, based on these templates, one can design the multiresolution algorithms for extraordinary vertices. In the following, we show that multiresolution algorithms with the biorthogonal filter banks built with blocks $Z(\boldsymbol{\omega})$ can be described in a simple way with templates. Because of the symmetry of the filter banks, we may simply use v, f and $\widetilde{v}, \widetilde{f}$ to describe the multiresolution algorithms.

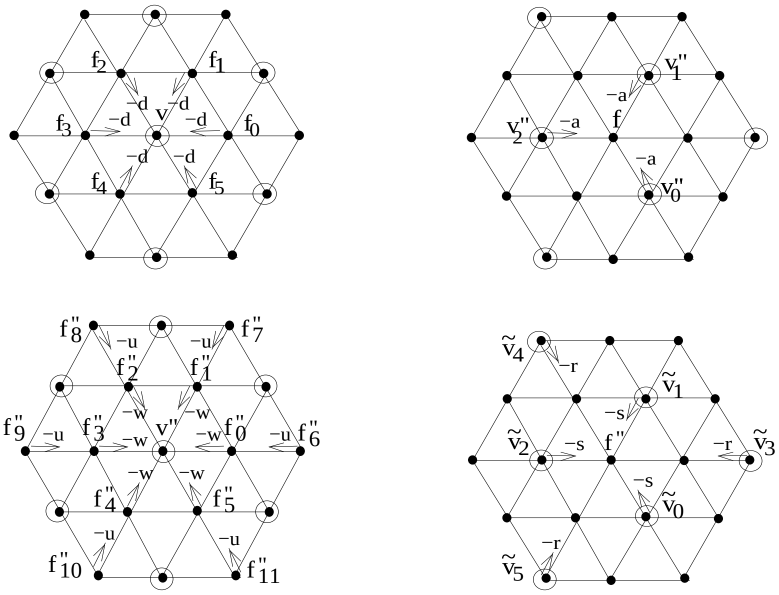

Figure 13: *Top-left: Decomposition Alg. Step 1; Top-right: Decomposition Alg. Step 2; Bottom-left: Decomposition Alg. Step 3; Bottom-right: Decomposition Alg. Step 4*

For given $\mathcal{C}$ sampled on $\mathcal{G}$ (or equivalently, for given $\{v\}$ and $\{f\}$), the multiresolution decomposition algorithm is given by (4.27)-(4.30) and shown in Fig. 13, where a, b, d, w, u, s and r are constants to be determined. More precisely, first we replace all v associated with *type V* nodes of $\mathcal{G}_3$ by v'' given by formula (4.27). Then, based on v'' obtained, we replace all f associated with *type F* nodes in $\mathcal{G}\backslash\mathcal{G}_3$ by f'' given in formula (4.28). After that, based on f'' obtained in Step 2, all v'' in Step 1 are updated by $\widetilde{v}$ given in formula (4.29). Finally, based on $\widetilde{v}$ obtained in Step 3, all f'' in Step 2 are updated by $\widetilde{f}$ given in formula (4.30).

Decomposition Algorithm:

Step 1. $v'' = \frac{1}{b}(v - d\sum_{k=0}^{5} f_k)$ (4.27)

Step 2. $f'' = f - a(v''_0 + v''_1 + v''_2)$ (4.28)

Step 3. $\widetilde{v} = v'' - w\sum_{k=0}^{5} f''_k - u\sum_{k=6}^{11} f''_k$ (4.29)

Step 4. $\widetilde{f} = f'' - s(\widetilde{v}_0 + \widetilde{v}_1 + \widetilde{v}_2) - r(\widetilde{v}_3 + \widetilde{v}_4 + \widetilde{v}_5)$. (4.30)

The multiresolution reconstruction algorithm to recover $\mathcal{C}$ associated with $\mathcal{G}$ (or equivalently, v and f associated with $\mathcal{G}_3$ and $\mathcal{G}\backslash\mathcal{G}_3$ respectively) from given $\widetilde{v}$ associated with $\mathcal{G}_3$ and given $\widetilde{f}$ asso-

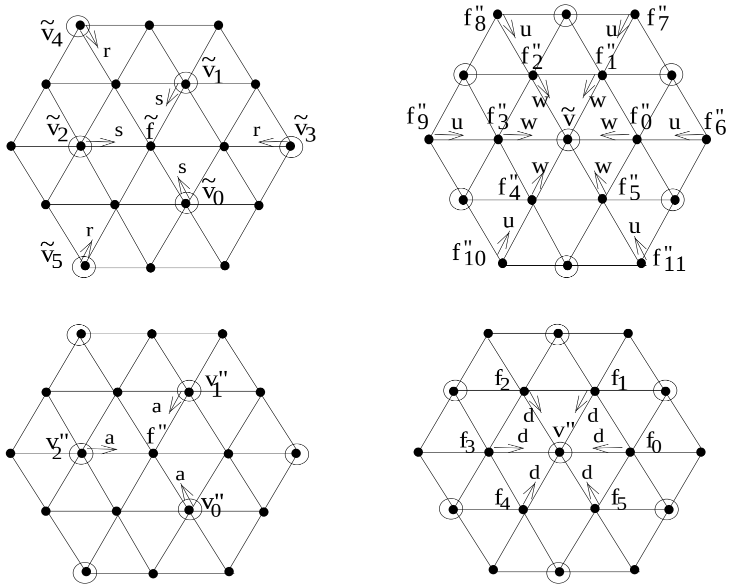

Figure 14: *Top-left: Reconstruction Alg. Step 1; Top-right: Reconstruction Alg. Step 2; Bottom-left: Reconstruction Alg. Step 3; Bottom-right: Reconstruction Alg. Step 4*

ciated with $\mathcal{G}\backslash\mathcal{G}_3$. The algorithm is given by (4.31)-(4.34) and shown in Fig. 14, where a, b, d, w, u, s and r are the same constants in the multiresolution decomposition algorithm. More precisely, first we update all $\widetilde{f}$ associated with *type F* nodes of $\mathcal{G}\backslash\mathcal{G}_3$ with the resulting f'' given by formula (4.31). Then, we update all $\widetilde{v}$ associated with *type V* nodes of $\mathcal{G}_3$ with the resulting v'' given by formula (4.32). After that, based on v'' obtained, we replace all f'' obtained in Step 1 by f with formula in (4.33). Finally, based on f obtained in Step 3, all v'' in Step 2 are updated with the resulting v given by formula (4.34).

Reconstruction Algorithm:

Step 1. $f'' = \widetilde{f} + s(\widetilde{v}_0 + \widetilde{v}_1 + \widetilde{v}_2) + r(\widetilde{v}_3 + \widetilde{v}_4 + \widetilde{v}_5)$ (4.31)

Step 2. $v'' = \widetilde{v} + w\sum_{k=0}^{5} f_k'' + u\sum_{k=6}^{11} f_k''$ (4.32)

Step 3. $f = f'' + a(v_0'' + v_1'' + v_2'')$ (4.33)

Step 4. $v = bv'' + d\sum_{k=0}^{5} f_k.$ (4.34)

When the constants a, b, d, u, w, s, r are appropriately chosen, the decomposed $\widetilde{v}$ is the "approximation" of the initial data $\mathcal{C}$, and $\widetilde{f}$ is the "details" of $\mathcal{C}$. The decomposition algorithm can be applied repeatedly to the "approximation" to get further "approximation" and "details" of the data, and reconstruction algorithm recovers the original data from the "approximation" and "details".

When $s = r = 0$, the above $\sqrt{3}$-refinement decomposition and reconstruction algorithms consist of three steps: Step 1 to Step 3 of (4.27)-(4.29) (with $\widetilde{f} = f''$), and Step 2 to Step 4 of (4.32)-(4.34) (with $f'' = \widetilde{f}$). These algorithms are studied in [40], where d, b, a are chosen to be

$$d = \frac{1}{12},\ b = \frac{1}{2},\ a = \frac{1}{3}. \tag{4.35}$$

With the formulas in (4.25) and (4.26), we find the filter banks $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ corresponding to the algorithms (4.27)-(4.34) with $s = r = 0$ are these given by (4.18) with

$$a = \alpha,\ d = \frac{\rho}{2\alpha},\ b = \gamma - 3\rho. \tag{4.36}$$

If d, b, a are given by (4.35), then, $\alpha = \frac{1}{3}, \rho = \frac{1}{18}, \gamma = \frac{2}{3}$, and hence, the resulting $\widetilde{p}(\boldsymbol{\omega})$ is that in [23]. As we mentioned above, the corresponding filter banks cannot generate ϕ in $L^2(\mathbb{R}^2)$, and therefore, they cannot generate biorthogonal bases for $L^2(\mathbb{R}^2)$. Thus, we should use other parameters. For example, when ρ, u are given by (4.20) (with α, γ, w given by (4.19)), d, b, a, w, u are

$$d = \frac{1}{18},\ b = \frac{2}{3},\ a = \frac{1}{3},\ w = -\frac{31}{135},\ u = \frac{1}{10}. \tag{4.37}$$

As discussed above, in this case, the corresponding filter banks generate biorthogonal bases for $L^2(\mathbb{R}^2)$ with the synthesis scaling function $\widetilde{\phi}$ and wavelets $\widetilde{\psi}^{(1)}, \widetilde{\psi}^{(2)}$ in $W^{1.9801}$.

With the formulas in (4.25) and (4.26) and careful calculations, one can obtain that the filter banks $\{p, q^{(1)}, q^{(2)}\}$ and $\{\widetilde{p}, \widetilde{q}^{(1)}, \widetilde{q}^{(2)}\}$ corresponding to the algorithms (4.27)-(4.34) with nonzero s, r are these given by (4.21) in Example 4 with d, b, a defined by (4.36) and $w_1 = -s, u_1 = -r$. With the values $\alpha = \frac{25}{81}, w_1 = -\frac{11}{81}$ in Example 4, the corresponding d, b, a, w, u, s, r are

$$d = \frac{4}{75},\ b = \frac{23}{27},\ a = \frac{25}{81},\ w = -\frac{3662}{18225},\ u = \frac{1313}{18225},\ s = \frac{11}{81},\ r = -\frac{91}{1863}. \tag{4.38}$$

In this case, as discussed in Example 4, the corresponding filter banks generate biorthogonal bases for $L^2(\mathbb{R}^2)$ with the synthesis scaling function $\widetilde{\phi}$ and wavelets $\widetilde{\psi}^{(1)}, \widetilde{\psi}^{(2)}$ in $W^{2.4556}$. One may also use other parameters as mentioned in Example 4 for the algorithms (4.27)-(4.34).

To obtain scaling functions and wavelets with a high approximation order or smooth order, we may use more steps as the above algorithms (4.27)-(4.34) with more parameters. The corresponding filter banks are given similarly to those in (4.21) with more blocks $Z(2\boldsymbol{\omega})$ or $\widetilde{Z}(2\boldsymbol{\omega})$. Then, we may use the above method to choose the parameters.

In this paper we consider the multiresolution algorithms for regular and triangle surface. The design of the corresponding multiresolution algorithms for extraordinary vertices, and the study of compactly supported wavelets for quadrilateral surface processing will be discussed elsewhere.

References

[1] J. D. Allen, Coding transforms for the hexagon grid, Ricoh Calif. Research Ctr., Technical Report CRC-TR-9851 (Aug. 1998).

[2] J. D. Allen, Perfect reconstruction filter banks for the hexagonal grid, in *Fifth International Conference on Information, Communications and Signal Processing 2005* (Dec. 2005), pp. 73–76.

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

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

[5] P. J. Burt, Tree and pyramid structures for coding hexagonally sampled binary images, *Computer Graphics and Image Proc.* **14** (1980) 271–80.

[6] C. Cabrelli, C. Heil, and U. Molter, Accuracy of lattice translates of several multidimensional refinable functions, *J. Approx. Theory* **95** (1998) 5–52.

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

[8] A. Cohen and I. Daubechies, A stability criterion for biorthogonal wavelet bases and their related subband coding scheme, *Duke Math. J.* **68** (1992) 313–335.

[9] A. Cohen and J. -M. Schlenker, Compactly supported bidimensional wavelets bases with hexagonal symmetry, *Constr. Approx.* **9** (1993) 209–236.

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

[11] S. D. Gertz, B. G. Bodmann, D. Vela, M. Papadakis, *et al*, Three-dimensional isotropic wavelets for post-acquisitional extraction of latent images of atherosclerotic plaque components from micro-computed tomography of human coronary arteries, *Academic Radiology* **17** (2007) 1509-1519.

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

[13] I. Guskov, Multivariate subdivision schemes and divided differences, Preprint, Princeton University (1998).

[14] X. J. He and W. J. Jia, Hexagonal structure for intelligent vision, in *Proceedings of the 2005 First International Conference on Information and Communication Technologies* (Aug. 2005), pp. 52–64.

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

[16] R. Q. Jia, Convergence of vector subdivision schemes and construction of biorthogonal multiple wavelets, in *Advances in Wavelets*, Ed. Ka-Sing Lau (Springer-Verlag, Singapore, 1999), pp. 199–227.

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

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

[19] Q. T. Jiang, FIR filter banks for hexagonal data processing, *IEEE Trans. Image Proc.* **17** (2008) 1512–1521.

[20] Q. T. Jiang, Orthogonal and biorthogonal $\sqrt{3}$-refinement wavelets for hexagonal data processing, *IEEE Trans. Signal Proc.* **57** (2009) 4304–4313.

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

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

[23] L. Kobbelt, $\sqrt{3}$-subdivision, in *SIGGRAPH Computer Graphics Proceedings* (2000), pp. 103–112.

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

[25] C. Loop, Smooth subdivision surfaces based on triangles, Master's Thesis, University of Utah, Department of Mathematics, Salt Lake City, 1987.

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

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

[28] L. Middleton and J. Sivaswarmy, *Hexagonal Image Processing: A Practical Approach* (Springer, 2005).

[29] P. Oswald, Designing composite triangular subdivision schemes, *Comput. Aided Geom. Design* **22** (2005) 659–679.

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

[31] M. Papadakis, B. G. Bodmann, S. K. Alexander, D. Vela, *et al*, Texture based tissue characterization for high-resolution CT-scans of coronary arteries, *Comm. in Numer. Methods in Engineering* **25** (2009) 597–613.

[32] D. P. Petersen and D. Middleton, Sampling and reconstruction of wave-number-limited functions in N-dimensional Euclidean spaces, *Information and Control* **5** (1962) 279–323.

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

[34] J. Romero, S. Alexander, S. Baid, S. Jain, and M. Papadakis, The geometry and the analytic properties of isotropic multiresolution analysis, *Advances in Computational Mathematics* **31** (2009) 282–328.

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

[36] 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.

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

[38] E. Simoncelli and E. Adelson, Non-separable extensions of quadrature mirror filters to multiple dimensions, *Proceedings of the IEEE* **78** (1990) 652–664.

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

[40] 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.

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

[42] D. White, A. J. Kimberling, and W. S. Overton, Cartographic and geometric components of a global sampling design for environmental monitoring, *Cartography and Geographic Information Systems* **19** (1992) 5–22.

[43] H. Xu, W. -S. Lu, and A. Antoniou, A new design of 2-D non-separable hexagonal quadrature-mirror-filter banks, in *Proc. CCECE*, (Vancouver, Sep. 1993), pp. 35–38.