Uniform Powell–Sabin Spline Wavelets

Joris Windmolders
Evelyne Vanraes
Paul Dierckx
Adhemar Bultheel

Report TW 335, February 2002



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Uniform Powell–Sabin Spline Wavelets

Joris Windmolders
Evelyne Vanraes
Paul Dierckx
Adhemar Bultheel

Report TW 335, February 2002

Department of Computer Science, K.U.Leuven

Abstract

This paper discusses how the subdivision scheme for uniform Powell–Sabin spline surfaces makes it possible to place those surfaces in a multiresolution context. We first show that the basis functions are translates and dilates of one vector of scaling functions. This defines a sequence of nested spaces. We then use the subdivision scheme as the prediction step in the lifting scheme and add an update step to construct wavelets that describe a sequence of complement spaces. Finally, as an example application, we use the new wavelet transform to reduce noise on a uniform Powell–Sabin spline surface.

Keywords : wavelets, lifting, Powell–Sabin splines, B-splines, subdivision
AMS(MOS) Classification : 42C40, 65T60, 65D07, 65D17, 68U07

Uniform Powell–Sabin Spline Wavelets

Joris Windmolders, Evelyne Vanraes,
Paul Dierckx, Adhemar Bultheel

Department Computerwetenschappen, Kath. Universiteit Leuven.
Joris.Windmolders@cs.kuleuven.ac.be[1]

Abstract

This paper discusses how the subdivision scheme for uniform Powell–Sabin spline surfaces makes it possible to place those surfaces in a multiresolution context. We first show that the basis functions are translates and dilates of one vector of scaling functions. This defines a sequence of nested spaces. We then use the subdivision scheme as the prediction step in the lifting scheme and add an update step to construct wavelets that describe a sequence of complement spaces. Finally, as an example application, we use the new wavelet transform to reduce noise on a uniform Powell–Sabin spline surface.

1 Introduction

CAGD applications rely on powerful mathematical representations of curves and surfaces which, in general, share a number of interesting properties:

- affine invariance,
- control points can be associated with it, giving us insight in the shape,
- predictable and local changes can be made to shape the curve or surface.

Widespread nowadays in commercial CAGD packages is the B–spline representation for curves, and correspondingly the tensor product B–spline representation for surfaces. The latter is, however, essentially restricted to rectangular domains, and therefore is not well suited for designing surfaces with an arbitrary number of edges. Farins Bézier triangles [3] are a worthwile alternative to represent piecewise polynomials on polygonal domains, but imposing smoothness conditions between the patches requires a great number of nontrivial relations between the coefficients to be satisfied. Another approach is a B–spline representation for Powell–Sabin (PS)–splines by Shi *et al.* [12], but their construction method has some serious drawbacks from the numerical point of view.

Dierckx [2] presented a normalized B–spline basis for PS–splines, which guarantees global C^1 smoothness for any choice of the coefficients, and resolves the numerical problems mentioned before. The particular case of PS–splines on uniform triangulations, has a number of advantages ([11]), e.g., a subdivision scheme has been published for it.

We will show that this subdivision scheme enables us to put uniform Powell–Sabin (UPS)–splines in a multiresolution context. Indeed, subdivision amounts to describing the given surface with respect to translated and dilated basis functions at a finer resolution level.

Lounsbery *et al.* [6] proved that with every subdivision scheme, a sequence of nested linear spaces with corresponding basis functions, the scaling functions, can be associated. It is then possible to construct wavelets, that is, a set of functions that span the sequence of orthogonal complement spaces

[1] Corresponding author

by imposing a least–squares condition when restricting the approximation to the scaling functions. Locally supported wavelets are obtained by relaxing the condition that the wavelets should lie in the orthogonal complement spaces. This is related to the technique of the lifting scheme [9, 1] in which locally supported biorthogonal wavelets are build using consecutive lifting steps.

In this paper we use the lifting scheme to construct Powell–Sabin spline wavelets. We restrict, in contrast to Lounsbery *et al.*, the method to functions defined on $\mathbb{R}^2$. Compact surfaces of arbitrary topological type can be represented by parametric UPS–splines and wavelets, and by componentwise application of the technique presented here.

2 Notation and Background

In this Section we settle the notation and provide the necessary background on Powell–Sabin splines on uniform triangulations. The notation of triangular lattices introduced in Section 2.1 is based on [5]. Section 2.2 recalls the necessary fundamentals of Bézier triangles and Section 2.3 introduces the normalized B–spline basis for PS–splines.

2.1 Signals and lattices

Consider the triangular lattice $\mathcal{K} = \Gamma\mathbb{Z}^2$ where Γ is the invertible 2×2 matrix:

$$\Gamma = \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix}, \tag{2.1}$$

and the 2×2 matrix with integer coefficients

$$D = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \tag{2.2}$$

We can find a sublattice of $\mathcal{K}$ as $D\mathcal{K}$. There are $|D| - 1$ distinct cosets of this first sublattice $D\mathcal{K}$ each of the form $D\mathcal{K} + t_i$, with $t_i \in \mathbb{Z}^2$ and $1 \leq i \leq 3$. We know that

$$\mathcal{K} = \bigcup_{i=0}^{M-1} (D\mathcal{K} + t_i), \tag{2.3}$$

where $t_0 = 0$ and the cosets in the union are disjoint. Figure 1 shows the decomposition of $\mathcal{K}$ into its four disjoint sublattices according to (2.2).

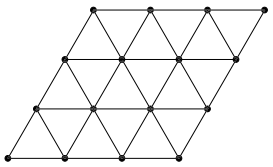
A signal S on $\mathcal{K}$ is, in general, a sequence of real-valued numbers:

$$S = \{S_k \in \mathbb{R} | k \in \mathcal{K}\}.$$

In this paper we focus on signals representing uniform Powell–Sabin (UPS)–splines, i.e., piecewise quadratic polynomials on a subset $\Delta \subset \mathcal{K}$ with polygonal boundary $\delta\Omega$. Δ is also called the triangulation of the *domain polygon* $\Omega \subset \mathbb{R}^2$, defined by $\delta\Omega$. It consists of a number of triangles $\rho_j, j = 1, \ldots, t$ having vertices $V_i, i = 1, \ldots, n$ with Cartesian coordinates (x_i, y_i). The triangles ρ_j are equilateral with edge length A, called the *base* of the triangulation.
Powell–Sabin splines have a normalized B–spline representation

$$S(x, y) = \sum_{i=1}^{n} \sum_{j=1}^{3} c_{i,j} B_i^j(x, y) \ , \quad (x, y) \in \Omega. \tag{2.4}$$

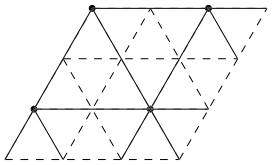
This representation will be discussed in detail in section 2.3. The signal can be thought of as a sequence of coefficients $c_{i,j}, i = 1, \ldots, n, \ j = 1, 2, 3$, that is, three coefficients per vertex V_i. Later we will show that each of these three coefficients can be associated with the corner of a triangle T_i. We will use the terms *signal* and *UPS–spline* interchangebly.



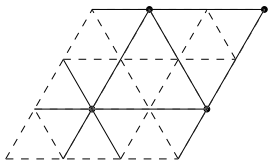
$$\Gamma = \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix} \qquad D = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$M = |D| = 4$$

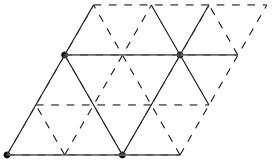
$$t_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



$$t_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



$$t_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



$$t_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

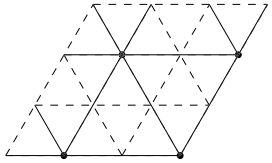


FIG. 1. Triangular two-dimensional lattice $\mathcal{K} = \Gamma\mathbb{Z}^2$ and sublattices.

2.2 Bézier triangles

Using standard multi–index notation, let $\lambda = (\lambda_1, \lambda_2, \lambda_3)$, $|\lambda| = \lambda_1 + \lambda_2 + \lambda_3 = d$, $\lambda_i \in \{0, 1, \ldots, d\}$. Consider a nondegenerate triangle $\mathcal{T}(T_1, T_2, T_3)$ in the plane. Any point P in that plane can be expressed in terms of barycentric coordinates $\tau = (\tau_1, \tau_2, \tau_3)$ with respect to $\mathcal{T}$: $P = \sum_{i=1}^{3} \tau_i T_i$ where $\sum_{i=1}^{3} \tau_i = 1$.

A Bézier polynomial of degree d [3] over that triangle is defined by

$$b_{\mathcal{T}}^d(P) = b_{\mathcal{T}}^d(\tau) = \sum_{|\lambda|=d} b_\lambda B_\lambda^d(\tau), \tag{2.5}$$

in which b_λ are called Bézier ordinates, and

$$B_\lambda^d(\tau) = \frac{d!}{\lambda_1!\lambda_2!\lambda_3!} \tau_1^{\lambda_1} \tau_2^{\lambda_2} \tau_3^{\lambda_3} \tag{2.6}$$

are the Bernstein polynomials on the triangle.

The piecewise linear interpolant to the points $(\frac{\lambda}{d}, b_\lambda)$ is the Bézier net or control net, and is displayed schematically in Figure 2 for the case $d = 2$. The points $\frac{\lambda}{d}$, marked on the figure, will be called Bézier domain ordinates. This representation has the convex hull property, pseudo–local control, is affine invariant, and is invariant under affine parameter transformations. The latter two properties indicate that, first, the shape of the surface does not change if the control net is translated, and second,

$$b_{\mathcal{T}}^d(P) = b_{M\mathcal{T}+N}^d(MP + N), \tag{2.7}$$

where $MX + N$ denotes an affine transformation of X, thus, e.g., scaling the domain triangle with respect to a point results in a dilated surface.

Representing complex shapes requires to use patch complexes with a great number of Bézier triangles. Keeping up continuity conditions between neighbouring patches results in nontrivial rela-

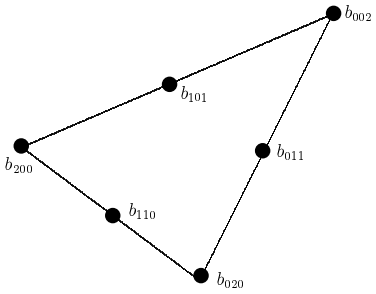


FIG. 2. Positions of Bézier ordinates for $d = 2$.

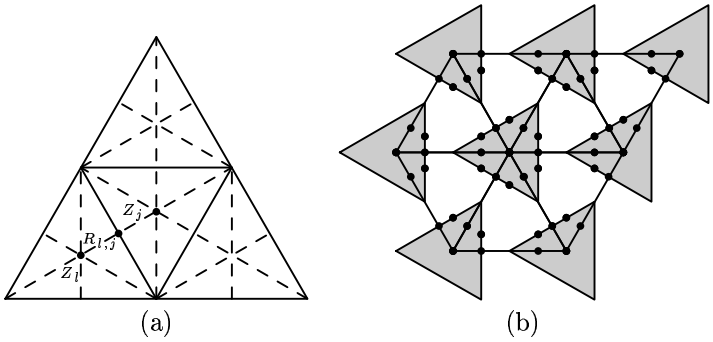


FIG. 3. (a) The dashed lines form a Uniform PS-refinement of the triangulation in solid lines. (b) PS–points and a fixed form for the PS–triangles containing the PS–points.

tions between their Bézier ordinates. We will overcome this problem by using split triangles in the next section.

2.3 A normalized B–spline basis for UPS–splines

This section recalls the definition of PS–splines and their normalized B–spline representation. We will restrict ourselves to the case of uniform triangulations Δ. A general description of PS–splines can be found in [2].

Let Δ^* be a Powell–Sabin refinement ([8]) of Δ, which divides each macro triangle ρ_j into six smaller triangles with a common vertex Z_j. In the case of a uniform triangulation, a PS–refinement can easily be found by drawing, for each triangle ρ_j, the bisector of its vertices (see figure 3(a)).

Now we consider the space of piecewise quadratic polynomials with C^1–continuity on Δ^*, denoted by $S_2^1(\Delta^*)$. Each of the $6t$ triangles resulting from the PS–refinement becomes the domain triangle of a quadratic Bézier polynomial. Powell and Sabin [8] proved that the dimension of the space $S_2^1(\Delta^*)$ equals $3n$, so there exists a unique solution $S(x, y) \in S_2^1(\Delta^*)$ for the interpolation problem

$$S(V_k) = f_k, \quad \frac{\partial S}{\partial x}(V_k) = f_{x,k}, \quad \frac{\partial S}{\partial y}(V_k) = f_{y,k}, \quad k = 1, \ldots, n. \tag{2.8}$$

Thus given the function values and derivative values at each point V_i, the Bézier ordinates on the domain subtriangles are uniquely defined.

Dierckx [2] showed that each piecewise polynomial $S(x, y) \in S_2^1(\Delta^*)$ has a unique normalized B–spline representation (2.4) where the basis functions form a convex partition of unity and have

local support: $B_i^j(x,y)$ is nonzero only on the so–called *molecule* M_i of V_i, being the set of triangles ρ_l that have V_i as a vertex. The number of triangles in M_i is called the *molecule number* m_i.

Furthermore, one can prove that the control triangles, defined as $T_i(c_{i,1}, c_{i,2}, c_{i,3})$, $i = 1, \ldots, n$, are tangent to the surface at $S(V_i)$. We thus have a representation that shares the properties mentioned in the introduction of tensor–product splines, but that is not restricted to rectangular domains. The advantage over Bézier triangles is that we have global C^1–continuity regardless of the choice of $c_{i,j}$. Dierckx also provides an algorithm to calculate the Bernstein-Bézier representation of the surface in an efficient and numerically stable way.

3 Uniform Powell–Sabin splines: a multiresolution

In Section 3.1, we have a closer look at the relation between the geometry of Δ^* and the basis functions. In Section 3.2 we show that the B–spline basis functions for UPS–splines are translates and dilates of three so–called scaling functions and in Section 3.3 we recall an algorithm for subdividing uniform PS–splines.

3.1 A closer look at the basis functions

We are interested in how the geometry of the PS–refinement determines the shape of the B–spline basis functions precisely. We will therefore briefly recall the algorithm for constructing a normalized B–spline basis from [2]:

(1) For each vertex $V_i \in \Delta$, we start by identifying its PS–points. To do so, consider all edges in the PS–refinement Δ^* that have V_i as a vertex. The Bézier domain ordinate on each such edge which is closest to V_i, is a PS–point. V_i itself is a PS–point as well. For example, Figure 3(b) shows the PS–points for each vertex.

(2) For each vertex V_i, find a triangle $t_i(Q_{i,1}, Q_{i,2}, Q_{i,3})$ containing all the PS–points of V_i. Denote its vertices $Q_{i,j}(X_{i,j}, Y_{i,j})$. The triangles $t_i, i = 1, \ldots, n$ are called PS–triangles. In [11] we have shown that for uniform PS–splines a fixed form for the PS–triangles can be used, instead of determining the PS–triangle locally for each vertex. Figure 3(b) shows the PS–triangles containing the PS–points at each vertex. It turns out that this fixed form is a equilateral triangle itself, having V_i as barycenter.

(3) Given the PS–triangle of a vertex V_i, three linearly independent triplets of real numbers can be found as follows:

(a) $\alpha_i = (\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3})$ are the barycentric coordinates of V_i with respect to t_i,

(b) $\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3}) = \left(\frac{Y_{i,2}-Y_{i,3}}{e}, \frac{Y_{i,3}-Y_{i,1}}{e}, \frac{Y_{i,1}-Y_{i,2}}{e}\right)$,

(c) $\gamma_i = (\gamma_{i,1}, \gamma_{i,2}, \gamma_{i,3}) = \left(\frac{X_{i,3}-X_{i,2}}{e}, \frac{X_{i,1}-X_{i,3}}{e}, \frac{X_{i,2}-X_{i,1}}{e}\right)$,

where

$$e = \begin{vmatrix} X_{i,1} & Y_{i,1} & 1 \\ X_{i,2} & Y_{i,2} & 1 \\ X_{i,3} & Y_{i,3} & 1 \end{vmatrix}, \tag{3.1}$$

and $|\alpha_i| = 1$, and $|\beta_i| = |\gamma_i| = 0$. Since the form of the PS–triangles is fixed, these vectors are independent of i:

$$\alpha_i := \alpha = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \quad \beta_i := \beta = \left(\frac{-4}{3A}, \frac{2}{3A}, \frac{2}{3A}\right), \quad \gamma_i := \gamma = \left(0, \frac{-2\sqrt{3}}{3A}, \frac{2\sqrt{3}}{3A}\right).$$

(4) The basisfunction $B_i^j(x,y)$ is the unique solution of the interpolation problem (2.8) with all $(f_k, f_{x,k}, f_{y,k}) = (0,0,0)$ except for $(f_i, f_{x,i}, f_{y,i}) = (\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j})$.

The Bézier ordinates of this solution can be found explicitely, and by a result of [2] they depend solely on:

- the barycentric coordinates of the PS–points with respect to the corresponding PS–triangle,
- the barycentric coordinates of the PS–refinement points Z_j and $R_{j,l}$ with respect to the corresponding triangle ρ_j.

Therefore, the Bézier ordinates of the uniform basis functions B_i^j do not depend on the index i, nor on the base A of the triangulation.

3.2 A multiresolution

From the previous, it follows that for a given uniform triangulation Δ, the basis functions are translates of three functions B_i^0:

$$\mathbf{B}^0(u) = \begin{bmatrix} B_1^0(u) \\ B_2^0(u) \\ B_3^0(u) \end{bmatrix}, \quad u \in \mathbb{R}^2. \tag{3.2}$$

Consider the refinement $D^{-1}\Delta$ of Δ, which can be obtained by midedge subdivision, and is a subset of $D^{-1}\mathcal{K}$. On Figure 4, Δ is displayed on the left and $D^{-1}\Delta$ on the right, together with their respective PS–triangles. $D^{-1}\Delta$ is also a uniform triangulation; its base is $A/2$. The Bézier ordinates of the basis $\mathbf{B}^1(u)$ on $D^{-1}\Delta$ are exactly the same as those on Δ. The support of a basis function on $D^{-1}\Delta$ and the support of a basis function on Δ are both equilateral hexagons, with edge length $A/2$, resp. A, so they are equal up to an affine transform. Thus by (2.7), the basis functions on $D^{-1}\Delta$ are translates and dilates of the functions $\mathbf{B}^0(u)$. In general, the basis functions on all standard refinements $D^{-j}\Delta$ of Δ can be written as

$$\mathbf{B}_k^j(u) = \mathbf{B}^0(D^j u + k),\ k \in \mathcal{K}. \tag{3.3}$$

So we can decompose $L_2(\mathbb{R}^2)$ into a sequence of nested subspaces $\mathcal{V}_j = S_2^1(D^{-j}\Delta^*) \subset L_2(\mathbb{R}^2), j \in \mathbb{Z}$, such that

$$\mathcal{V}_j \subset \mathcal{V}_{j+1},\ j \in \mathbb{Z}, \tag{3.4}$$

the closure of their union is $L_2(\mathbb{R}^2)$

$$\overline{\lim_{j\to\infty} \mathcal{V}_j} = \overline{\bigcup_{j\in\mathbb{Z}} \mathcal{V}_j} = L_2(\mathbb{R}^2) \tag{3.5}$$

and their intersection contains only the zero function

$$\lim_{j\to-\infty} \mathcal{V}_j = \bigcap_{j\in\mathbb{Z}} \mathcal{V}_j = \{0\}. \tag{3.6}$$

A signal $S(u),\ u \in \mathbb{R}^2$ that belongs to one of these subspaces $\mathcal{V}_j$ satisfies the dilation and translation property:

$$S(u) \in \mathcal{V}_j \Leftrightarrow S(Du) \in \mathcal{V}_{j+1},\ j \in \mathbb{Z},\ u \in \mathbb{R}^2, \tag{3.7}$$

$$S(u) \in \mathcal{V}_0 \Leftrightarrow S(u+k) \in \mathcal{V}_0,\ k \in \mathcal{K},\ u \in \mathbb{R}^2. \tag{3.8}$$

Furthermore, the basis functions $\mathbf{B}^0(u)$ form a stable basis for the space $\mathcal{V}_0$. This fact, together with (3.4)–(3.8) form the definition of a multiresolution analysis.

3.3 Subdivision of uniform PS–splines

The goal of subdivision is essentially to calculate the B–spline representation (2.4) of a PS–spline surface on a refinement of the given Δ. The new basis functions have smaller support and thus give the designer more local control when manipulating surfaces. Furthermore, since the linear interpolant of the barycenters of the control triangles converge to the surface itself, it is a common technique for displaying surfaces graphically.

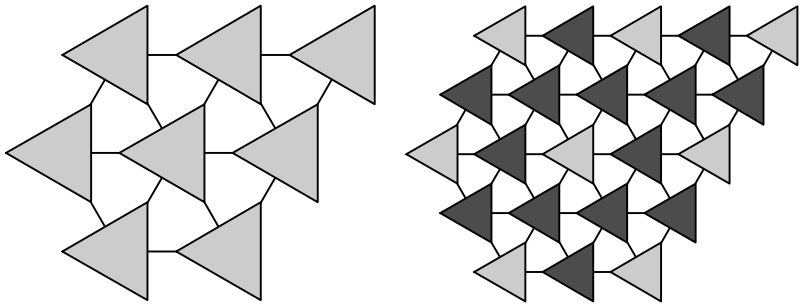


FIG. 4. Subdivision displayed schematically. (a) PS–triangles on the coarser grid. (b) Grid obtained by midedge subdivision, together with its PS–triangles.

Windmolders and Dierckx [11] presented a subdivision scheme for PS–splines on uniform triangulations. Consider again a triangulation Δ and its refinement $D^{-1}\Delta$, as illustrated in Figure 4. During subdivision, a new control triangle has to be calculated for each edge midpoint of Δ, becoming a vertex of $D^{-1}\Delta$. These are the darker triangles in Figure 4. Next, the original control triangles have to be rescaled. The algorithm uses convex barycentric combinations of the coefficients only. The new control triangles along an edge $V_iV_j \in \Delta$ are calculated as illustrated in Figure 5, for the bottom edge of a triangle $\rho(V_i, V_j, V_k) \in \Delta$:

$$\begin{cases} c_1 &= \frac{1}{2}(c_{i,2} + c_{i,3}) \\ c_2 &= \frac{1}{2}c_{j,1} + \frac{1}{4}(c_{i,2} + c_{j,2}) \\ c_3 &= \frac{1}{2}c_{j,1} + \frac{1}{4}(c_{i,3} + c_{j,3}). \end{cases} \tag{3.9}$$

Rescaling the original control triangles is accomplished by

$$\begin{cases} c'_{i,1} &= \frac{2}{3}c_{i,1} + \frac{1}{6}(c_{i,2} + c_{i,3}) \\ c'_{i,2} &= \frac{2}{3}c_{i,2} + \frac{1}{6}(c_{i,3} + c_{i,1}) \\ c'_{i,3} &= \frac{2}{3}c_{i,3} + \frac{1}{6}(c_{i,1} + c_{i,2}) \end{cases} \tag{3.10}$$

for the control triangle at V_i.

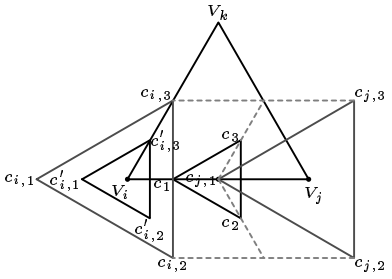


FIG. 5. Subdivision for the bottom edge of a triangle $\rho(V_i, V_j, V_k) \in \Delta$.

4 Uniform Powell–Sabin Spline Wavelets

In this Section we design a lifting scheme based on UPS–spline subdivision, yielding a wavelet transform for data on uniform triangulations. Section 4.1 introduces the lifting scheme, that splits a signal into its high and low–frequency components, step by step. Section 4.2 shows how the basis

functions of these components can be found. Section 4.3 focusses on the multi–wavelet aspect in this setting, and finally Section 4.5 generalizes our findings to parametric splines .

4.1 Lifting

The goal is to transform a given signal S into a number of components, capturing either the high–frequency, or the low–frequency characteristics of S. It must be possible to recover the original signal from the different components (this is called *perfect reconstruction*). The lifting scheme (Figure 6) is a method for constructing such a transform, yielding *biorthogonal wavelets* as the basis functions of the high–frequency components [1, 9]. It starts from a trivial wavelet transform and enhances its properties using a number of lifting steps. In our construction we use two lifting steps: a *prediction* step and an *update* step. Lifting has a number of advantages over the classical method of constructing wavelets, including the fact that the inverse transform can easily be found by running the scheme backward, from right to left, changing each + into a − and vice versa, and inverting each scaling operation (see also [5]).

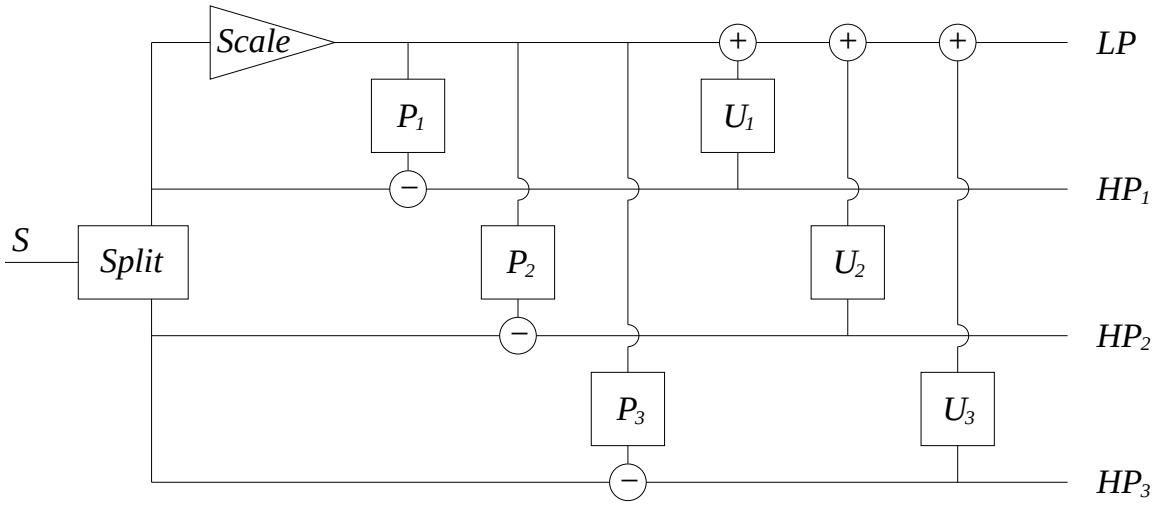


FIG. 6. The lifting scheme starts with a trivial wavelet i.e. just splitting the signal in samples on the four distinct sublattices. It then gradually builds a new wavelet with improved properties using lifting steps.

Split The trivial wavelet transform splits the given signal into four parts, according to the four disjunct sublattices $D\Delta + t_j$ of the original triangulation Δ.

Prediction Consider the subsignal on the zeroth sublattice $D\Delta$ on the upper branch in Figure 6. We try to predict the signals on the other sublattices from this subsignal: first rescale the control triangles on $D\Delta$ by the inverse transform of (3.10). This brings them to a lower resolution level. From this level, we use the subdivision rule (3.9) to estimate the values on the other sublattices. We call this the *prediction operator* P_i for the ith sublattice. Each prediction $P_i(S)$ is subtracted from the real values on $D\Delta + t_i$. The difference tells how much the prediction fails to be correct. We call these differences the *detail coefficients* or *wavelet coefficients*. For quadratic polynomials, the prediction will always be exact; the wavelet coefficients are equal to zero. Referring to the wavelet terminology, we say that the UPS–spline wavelet transform has three dual vanishing moments. We can say that the upper branch of the scheme contains the low–frequency component of the signal S, we call these the *scaling coefficients*. Thinking of representation (2.4), these are just B–spline coefficients of a uniform PS–spline on a coarser grid than the one we started from. It is still possible to recover

the original signal: add the predictions $P_i(S)$ to the wavelet coefficients on $D\Delta + t_i, i = 1, 2, 3$ and rescale the control triangles on $D\Delta$ using (3.10).

Update In general, the signal on $D\Delta$, which results from the split and prediction steps, does not have the same average as the original signal. In this final lifting step, we make sure that the DC–component of the signal is maintained in the low–frequency component. Again, referring to the wavelet terminology we say that the wavelet transform has one primal vanishing moment. This goal can be achieved by adding update operators U_i, which update the signal on $D\Delta$ from the values on the other sublattices. We propose that a detail coefficient corresponding to a control triangle T_l, on some branch $D\Delta + t_i$, updates its two neighbouring control triangles T_m and T_n on the zeroth sublattice. These are the control triangles from which T_l was predicted in the previous step. More exactly, a coefficient $c_{l,j}$ will increase the values of $c_{m,j}$ and $c_{n,j}$ with an amount $k * c_{l,j}$. The value of k is determined such that the average of the scaling coefficients is maintained, or equivalently, such that the integral of the wavelet functions is equal to zero [10]. As pointed out in the next subsection a wavelet function can be found by doing an inverse wavelet transform on a sequence of zeros with one particular wavelet coefficient set to one. The resulting scaling coefficients are the PS–representation of the wavelet function. We require now that the sum of the barycenters of the control triangles, which lie on the wavelet function, equals zero. This yields the value of k as illustrated in Figure 7. The control triangles whose values are changed during a certain step, have been shaded.

Figure 7(a) : one wavelet coefficient, in this case $c_{l,1}$, is set to one.

Figure 7(b) : inverse update. The neighbouring control triangles along this branch are updated, i.e., $c_{m,1} \leftarrow c_{m,1} - k * c_{l,1}$ and $c_{n,1} \leftarrow c_{n,1} - k * c_{l,1}$.

Figure 7(c) : inverse prediction. The figure shows the situation before the scaling step. Since the latter does not affect the sum of the coefficients, it will be ignored here.

Demanding that the sum of the coefficients equals zero finally yields

$$k = \frac{1}{8}. \tag{4.1}$$

This value is also valid for the other predicion steps (corresponding to $c_{l,2}, c_{l,3}$ and the other branches). However, note that until now our lifting scheme has not taken the edges of Δ into account. For prediction, this is not a problem since it only involves two neighbouring control triangles. For update on the other hand, it might well be the case that not all the control triangles in Figure 7(c) belong to the domain under consideration. Nevertheless, the same reasoning as before can be applied for any possible boundary case which is not covered here. It appears that only six different situations occur, i.e., six situations where another value of k is used during update. Finally we note that it can be proven that, if the average of the scaling coefficients is maintained, the integral of the PS–spline surface is invariant over the low–frequency component.

4.2 The wavelets

The wavelet transform decomposes a given signal into a low–frequency component and three high frequency components . The same scheme can be applied repeately to the low frequency component. In general, we have a n–step wavelet transform.

Running the inverse wavelet transform starting from a sequence of zeroes and one scaling coefficient equal to one at a particular position, yields, in the limit, a control net converging to a scaling function, i.e., a B–spline basis function corresponding to that position. In fact, this is nothing else than repeated subdivision.

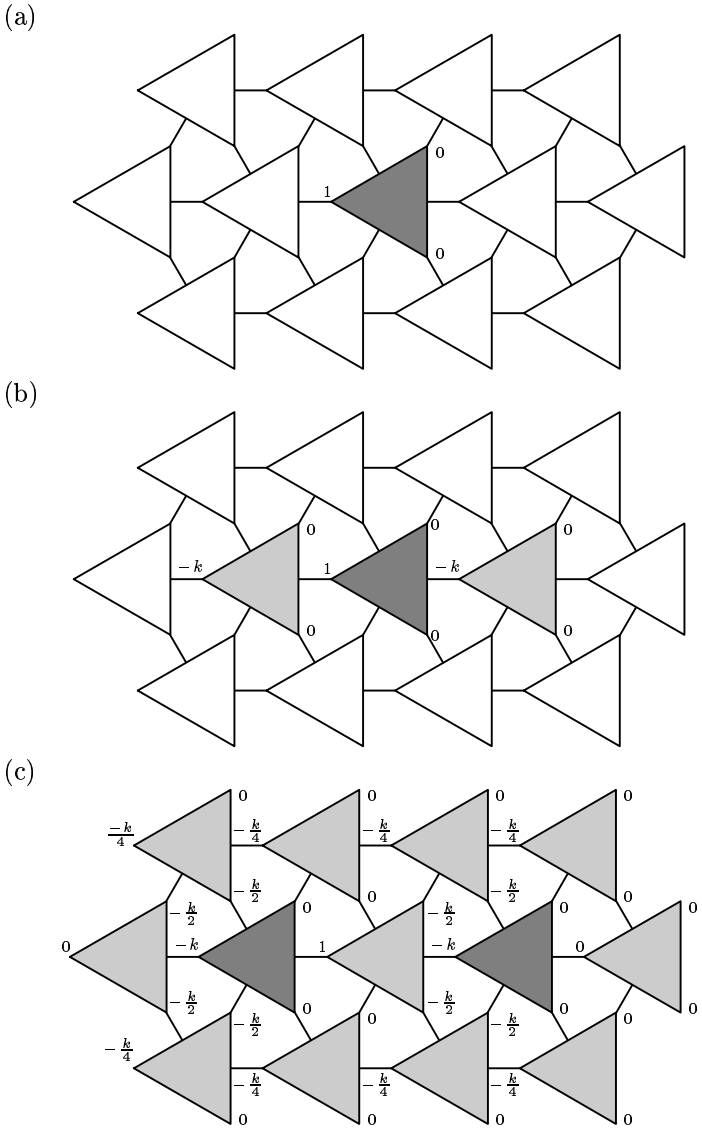
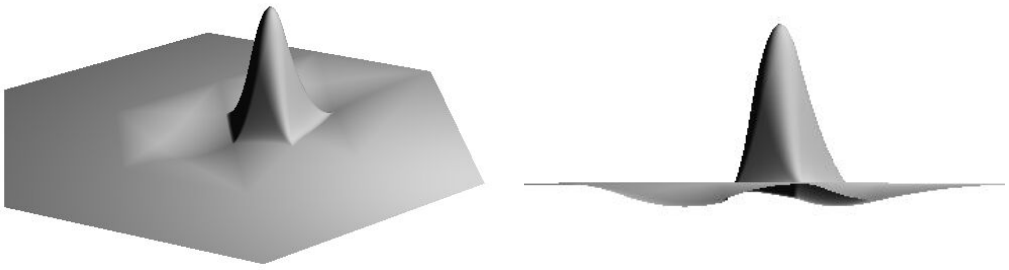


FIG. 7. Calculation of the update weights. (a) One wavelet coefficient is set to one. (b) Situation after inverse update. (c) Situation after inverse prediction and before inverse scaling.

If analogously one wavelet coefficient is set to one, the inverse wavelet transform yields the B–spline coefficients of the corresponding wavelet, or, in the limit, a control net converging to this wavelet. Figure 8 shows a wavelet, (a) seen from the top, (b) seen from aside. The latter clearly shows the parts above and beneath the plane $z = 0$, which have the same integral in absolute value.

4.3 Multiwavelets

In the standard wavelet theory, the scaling functions are translates and dilates of one function. For uniform PS–splines, we have, by (3.2)–(3.3), translates and dilates of three scaling functions, one corresponding to each corner of the control triangles. This is called a *multi–scaling function.*



(a) (b)

FIG. 8. A UPS–spline wavelet. (a) View from the top. (b) Sideview, clearly showing the parts above and beneath the plane $z = 0$.

Correspondingly, the wavelets resulting from the algorithm in section 4.2, corresponding to one of the three HP–branches, are translates and dilates of three wavelets, the *multi–wavelet*. Without going into details, multi–wavelets (see [4, 7], and references therein) allow more complex designs of the predict– and update–operators than the one considered in this paper (see appendix A for a brief comment on this). The multi–scaling function can be written in terms of translates and dilates of itself. This *refinement equation* for UPS–splines is given in appendix B.

4.4 The wavelet decomposition

Referring to the standard notation in the wavelet literature, we will denote the multi–scaling function in this section as $\mathbf{\Phi}(u) := \mathbf{B}^0(u)$, and the three multi–wavelets corresponding to each branch as $\mathbf{\Psi}_i(u) := [\psi_{i,1}(u), \psi_{i,2}(u), \psi_{i,3}(u)]^T$, $i = 1, 2, 3$. Section 4.1 then suggests an algorithm for decomposing a given signal $S \in V_{j+1}$ into a low–resolution signal $LP(S) \in V_j$ and three high resolution components. We already noticed that the space V_j is spanned by the translates and dilates of $\mathbf{\Phi}(u)$, i.e.

$$\mathbf{\Phi}_{j,k}(u) = \mathbf{\Phi}(D^j u - k) \quad \text{with} \quad k \in \mathcal{K}. \tag{4.2}$$

The three high frequency components are each captured in subspaces $W_{i,j}$ spanned by the translates and dilates of $\mathbf{\Psi}_i(u)$, i.e.,

$$\mathbf{\Psi}_{i;j,k}(u) = \mathbf{\Psi}_i(D^j u - k) \quad \text{with} \quad k \in \mathcal{K}. \tag{4.3}$$

These spaces form the complement of V_j in V_{j+1}

$$V_j \oplus \left(\bigoplus_{i=1}^{3} W_{i;j} \right) = V_{j+1}. \tag{4.4}$$

Suppose the signal S belongs to V_J, i.e., has resolution level J. It can be decomposed by the repeated wavelet transform such that the eventual lowest resolution level is L. The expansion of S then reads

$$S = \sum_{j=L}^{J} \sum_{k \in D^j \mathcal{K}} \sum_{i=0}^{M-1} W_{i;j,k} \mathbf{\Psi}_{i;j,k} + \sum_{k \in D^J \mathcal{K}} C_{L,k} \mathbf{\Phi}_{L,k}, \quad W_{i;j,k}, C_{L,k} \in \mathbb{R}^{3\times 1}. \tag{4.5}$$

4.5 Parametric UPS–splines

The lifting scheme presented in this paper can be generalized in a straightforward way to parametric uniform PS–splines

$$\mathbf{S}(u,v) = \begin{bmatrix} S_x(u,v) \\ S_y(u,v) \\ S_z(u,v) \end{bmatrix} = \sum_{i,j} \mathbf{c}_{i,j} B_i^j(u,v), \quad (u,v) \in \Omega, \quad \mathbf{c}_{i,j} \in \mathbb{R}^3, \tag{4.6}$$

by applying it componentwise.

5 Application: noise removal with a thresholding algorithm

We now use the newly developed UPS–spline wavelet transform in an algorithm for smoothing a noisy surface. Suppose we are given a parametric UPS–spline $\mathbf{S}(u,v)$ on Δ, such that the number of triangles ρ_j along each edge of Ω is, up to a constant, the same power of two, say 2^f. Such a surface can be constructed, for example, by applying f subdivision steps to an initial surface, and editing the control points afterwards to modify the surface locally. We call this surface $\mathbf{S}^f(u,v)$.

The thresholding algorithm begins with a forward wavelet transform with exactly f steps. This means that first $\mathbf{S}^f$ is decomposed into $\mathbf{S}^{f-1}$, a picture of the given surface on a lower resolution level, and $\mathbf{W}^1 = (W_x^1, W_y^1, W_z^1)$, a series of wavelet coefficients for each component in the parametric representation. $\mathbf{S}^{f-1}$ is the low–frequency part and also a PS–spline surface. $\mathbf{W}^1$ captures the high frequency parts. Next the same decomposition is applied to the low frequency part: $\mathbf{S}^{f-1} \leftarrow (\mathbf{S}^{f-2}, \mathbf{W}^2)$, etc. The result is a surface $\mathbf{S}^0(u,v)$, together with f series of wavelet coefficients $\mathbf{W}^k, k = 1, \ldots, f$. Thereafter follows f thresholding steps . During each such step $k = 1 \ldots, f$, the user can provide a triple of real positive thresholds $(\epsilon_x^k, \epsilon_y^k, \epsilon_z^k)$. All the wavelet coefficients belonging to $W_l^{f-(k-1)}$ which have an absolute value smaller then $\epsilon_l^k, l = x, y, z$, are replaced by zeros. For the remaining wavelet coefficients of $W_l^{f-(k-1)}$ we consider two possibilities:

Hard thresholding leaves the wavelet coefficients with an absolute value higher than the threshold unchanged,

Soft thresholding decreases the absolute value of the remaining wavelet coefficients from $W_l^{f-(k-1)}$ with ϵ_l^k, leaving their sign unchanged.

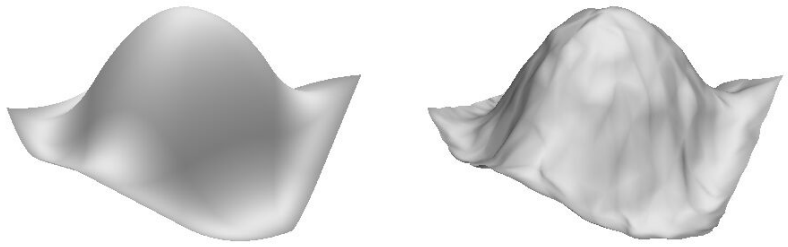
Applying the inverse wavelet transform on the thresholded coefficients results in the denoised surface.

We have tested this algorithm on an initially smooth UPS–spline surface, to which noise has been added. The given smooth surface has been subdivided f times, after which the B–spline coefficients $c_{i,j}$ were replaced by $c_{i,j} + \delta c_{i,j}$ with $\delta c_{i,j}$ uniformly distributed on a user specified interval $[-c, c]$. Figure 9 shows the original surface (left) and the surface after adding noise to the coefficients (right). Figure 10 shows the results for $f = 3$, with ϵ_l^k equal to 10% of the maximal wavelet coefficient of $W_l^{f-(k-1)}$ in absolute value, for $l = x, y, z$. The result for hard thresholding is displayed left and the result for soft thresholding right. On first view, one can see that the latter yields the best results. The following table shows the percentage of wavelet coefficients that have been eliminated during each soft thresholding step.

k	S_x	S_y	S_z
1	27.78%	22.23%	13.89%
2	48.41%	50.00%	38.10%
3	73.50%	75.43%	74.79%

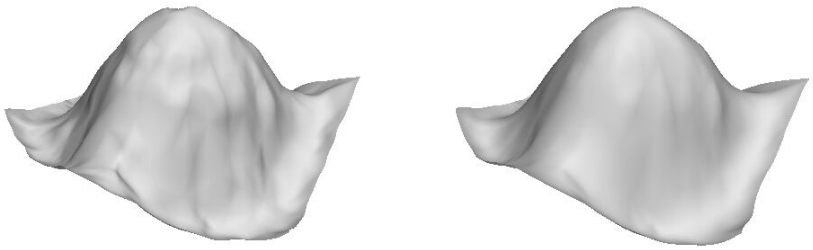
6 Conclusion

In this paper we presented a new wavelet transform for signals on regular triangulations, which can be modelled by normalized uniform Powell–Sabin B–splines. We used the existing subdivision algorithm as a prediction step in a standard lifting scheme, such that quadratic polynomial data are exactly reproduced, and designed an update step such that the DC–component of the data is retained. This enabled us to use wavelet techniques on PS–surfaces, which was not possible before. As an example, we developed a thresholding algorithm for smoothing UPS–spline surfaces. However, some questions remain unanswered. We therefore point out some possible directions for further



(a) (b)

FIG. 9. (a) The original surface. (b) Noise added to the smooth surface.



(a) (b)

FIG. 10. (a) Hard thresholding. (b) Soft thresholding, both after three steps.

research. The multi–wavelet aspect can be further elaborated in a more sophisticated update step. Next, we wonder whether existing techniques such as noise removal and compression are directly applicable to UPS–spline wavelets. Referring to the thresholding algorithm, it would be interesting to have a measure of the smoothness of PS–spline surfaces, and to investigate the choice of the threshold ϵ_l^k. Finally, would a subdivision scheme for non–uniform Powell–Sabin splines exist, and does it give raise to 2D wavelets on arbitrary triangulations?

Acknowledgement

This work is partially supported by the Belgian Program on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, and by the Flemish Fund for Scientific Research (FWO VLaanderen) project MISS (G.0211.02). The scientific responsibility rests with the authors.

Bibliography

1. I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
2. P. Dierckx. On calculating normalized Powell–Sabin B–splines. *CAGD*, 15(3):61–78, 1997.
3. G. Farin. Triangular bernstein–bézier patches. *CAGD*, 3(2):83–128, 1986.
4. V. Strela G. M. Davis and R. Turcajová. Multiwavelets construction via the lifting scheme. In *Wavelet Analysis and Multiresolution Methods.* Marcel Dekker, 1999.
5. J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Image Proc.*, 9(3):480–496, March 2000.

6. Tony D. DeRose Michael Lounsbery and Joe Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, 1997.
7. G. Plonka and V. Strela. From wavelets to multiwavelets. In *Mathematical Methods for Curves and Surfaces II.* Vanderbilt University Press, Nashville, TN, 1998.
8. M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Transactions on Mathematical Software*, 3:316–325, 1977.
9. W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.
10. W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.
11. J. Windmolders and P. Dierckx. Subdivision of Uniform Powell–Sabin splines. *CAGD*, 16:301–315, 1999.
12. W. Wang X. Shi, S. Wang and R. H. Wang. The C^1–quadratic spline space on triangulations. *Report 86004, Department of Mathematics, Jilin University, Changchun*, 1986.

Appendix A: multiwavelet aspects

Apart from the fact that multi–wavelets allow for more flexibility by using translates and dilates of more than one scaling function and wavelet function, one can design more advanced update and prediction operators than in the single wavelet approach. For instance, recall that our update operator has the simple form $c_{m,1} \leftarrow c_{m,1} + k * c_{l,1}$. A more complex design such as $c_{m,j} \leftarrow c_{m,j} + k_{j,1} * c_{l,1} + k_{j,2} * c_{l,2} + k_{j,3} * c_{l,3}$ is however possible. In this section we show that our scale step actually takes advantage of this multi–wavelet aspect: the subdivision rule (3.10) can be rewritten as

$$\begin{cases} c'_{i,1} &= \frac{2}{3}c_{i,1} + \frac{1}{6}c_{i,2} + \frac{1}{6}c_{i,3} \\ c'_{i,2} &= \frac{1}{4}c'_{i,1} + \frac{5}{8}c_{i,2} + \frac{1}{8}c_{i,3} \\ c'_{i,3} &= \frac{1}{5}c_{i,1} + \frac{1}{5}c'_{i,2} + \frac{1}{40}c_{i,3}. \end{cases} \tag{6.1}$$

Using this formulation, the scale step in Figure 6 can be unraveled in a wiring diagram (Figure 11), from which it is clear that each coefficient of the multi–scaling function contributes to the new value of each other coefficient.

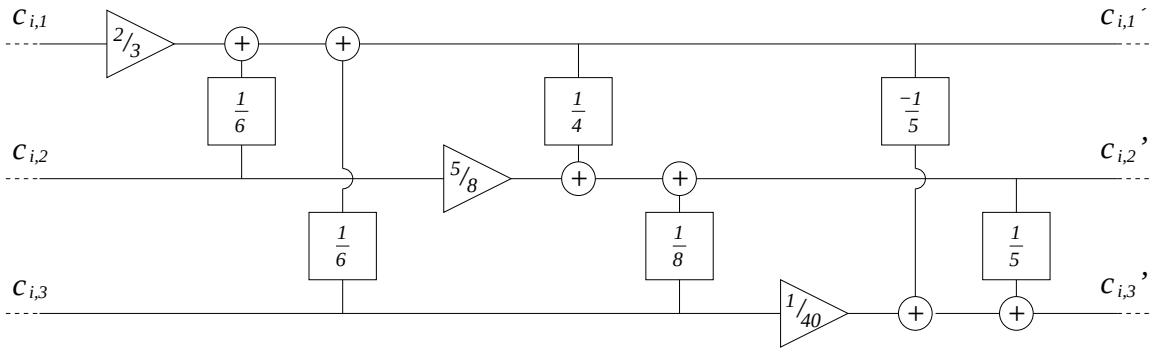


FIG. 11. Detail of the scale step factored in lifting steps.

Appendix B: the dilation equation

The dilation equation, which expresses the multi–scaling function in terms of translates and dilates of itself, can be found by applying the subdivision scheme directly to the UPS–spline basis functions.

$$
\begin{aligned}
\begin{bmatrix} B_1(u) \\ B_2(u) \\ B_3(u) \end{bmatrix} &= \begin{bmatrix} 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 2/3 \\ 1/6 & 1/6 & 2/3 \end{bmatrix} \begin{bmatrix} B_1(Du-k_0) \\ B_2(Du-k_0) \\ B_3(Du-k_0) \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} B_1(Du-k_1) \\ B_2(Du-k_1) \\ B_3(Du-k_1) \end{bmatrix} \\
&+ \begin{bmatrix} 1/4 & 0 & 1/2 \\ 0 & 1/4 & 1/2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_1(Du-k_2) \\ B_2(Du-k_2) \\ B_3(Du-k_2) \end{bmatrix} \\
&+ \begin{bmatrix} 1/4 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} B_1(Du-k_3) \\ B_2(Du-k_3) \\ B_3(Du-k_3) \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 1/4 & 0 \\ 1/2 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} B_1(Du-k_4) \\ B_2(Du-k_4) \\ B_3(Du-k_4) \end{bmatrix} \\
&+ \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 1/2 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} B_1(Du-k_5) \\ B_2(Du-k_5) \\ B_3(Du-k_5) \end{bmatrix} \\
&+ \begin{bmatrix} 1/4 & 1/2 & 0 \\ 0 & 0 & 0 \\ 0 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} B_1(Du-k_6) \\ B_2(Du-k_6) \\ B_3(Du-k_6) \end{bmatrix}, \qquad (6.2)
\end{aligned}
$$

where $k_0 = 0$, and

$$
\begin{aligned}
k_1 &= (-1/2, 0)^T, \quad k_2 = \left(-1/4, -\sqrt{3}/4\right)^T, \quad k_3 = \left(1/4, -\sqrt{3}/4\right)^T, \\
k_4 &= (1/2, 0)^T, \quad k_5 = \left(1/4, \sqrt{3}/4\right)^T, \quad k_6 = \left(-1/4, \sqrt{3}/4\right)^T.
\end{aligned}
$$

For the multi–wavelet functions $\mathbf{\Psi}_i(u)$ one can find a similar refinement relation with 3×3 matrices as filter coefficients.