Construction of Multiresolution Triangular B-Spline Surfaces using Hexagonal Filters

A. Dreger, M. H. Gross, J. Schlegel
Department of Computer Science
Swiss Federal Institute of Technology (ETH), Zürich, Switzerland
eMail: dreger@inf.ethz.ch, grossm@inf.ethz.ch

Abstract

We present multiresolution B-spline surfaces of arbitrary order defined over triangular domains. Unlike existing methods, the basic idea of our approach is to construct the triangular basis functions from their tensor product relatives in the spirit of box splines by projecting them onto the barycentric plane. The scheme works for splines of any order where the fundamental building blocks of the surface are hierarchies of triangular B-spline scaling functions and wavelets spanning the complement spaces between levels of different resolution. Although our bases functions have been deduced from the corresponding 3D-bases, our decomposition and reconstruction scheme operates directly on the triangular mesh using hexagonal filters. The resulting basis functions are used to approximate triangular surfaces and possess many useful properties, such as multiresolution editing, local level of detail, continuity control, surface compression and much more. The performance of our approach is illustrated by various examples including parametric and nonparametric surface editing and compression.

Keywords: *Triangular B-spline wavelets, box splines, multiresolution editing, hierarchical surface representation, surface compression, decomposition, reconstruction.*

1 Introduction

Multiresolution surface representations have become a fundamental paradigm in geometric modeling and computer graphics, primarily because they enable one to design and edit a surface at different levels of detail. In addition, the multiresolution approach comes along with many other useful properties such as local and global level of detail, efficient surface compression, progressively refinable reconstruction, error bounds, and mostly fast and handy computational algorithms.

Many different approaches to multiresolution and hierarchical surface representations can be found in the literature. One of the pioneers (Forsey et al. [14]) constructed hierarchies of B-spline bases, whose tensor product extensions can be used for surface approximations at multiple levels of detail. Others, such as Chui [7] or Quak [22], designed sets of semi-orthogonal wavelets to span the associated difference spaces and used them in the context of curve and surface design (Finkelstein [13]), or to implement surface compression (Staadt [26]) and triangulation [16]. Due to the limitations of tensor product constructions, various researchers proposed multiresolution surface representations for arbitrary control nets. Here Lounsbery [19] or Eck [12], for instance, proposed linear bi-orthogonal wavelet bases to efficiently describe triangular meshes. Others, such as Schroeder [23] or Nielson [21], introduced different forms of bi-orthogonal or orthogonal Haar bases on triangular spherical domains and employed them for various tasks in modeling, rendering and visualization. Nonseparable sampling schemes can also be found in the signal processing literature (Simoncelli [25]). Unfortunately, most of the existing explicit construction schemes for triangular wavelets only work out for low degrees and must find a balance between different fundamental mathematical properties, such as vanishing moments or continuity. Thus, for efficient surface modeling, higher order smooth triangular B-spline wavelets are highly desirable.

Whereas the compact wavelet representation imposes tight constraints on the design of the associated decomposition and reconstruction operators, more general subdivision schemes, such as Dyn [11], Loop [18], or Doo [9], turned out to be a promising alternative. Using them, Schroeder [23], in combination with generalized subsampling operators (Taubin [29]), allows one to build sophisticated multiresolution mesh editors. In these cases, however, the freedom taken for the operator design goes at the cost of over-representations, and the basic building blocks are no longer splines.

Since B-splines have been a fundamental concept in surface modeling and CAGD, the motivation for our research was to point out a simple and efficient alternative for the construction of multiresolution triangular B-spline surfaces of arbitrary polynomial order. Unlike contemporary approaches operating directly on triangles (Seidel [24]), we recalled some fundamental work on box splines from Boehm [2] and de Boor [3], [4]. Here, the basic idea is to construct a bivariate B-spline basis function consisting of triangular polynomial segments from trivariate tensor product splines by a line integration. This integration leads to splines whose polynomial degree is higher than the degree of the trivariate B-Splines. This powerful concept can be extended to hierarchies of B-spline functions and wavelets. We will demonstrate that a multiresolution hierarchy of triangular scaling functions and wavelets can be constructed effectively by projection of their 3D tensor product relatives. The raise of degree and continuity of the bases allows us to generate linear triangular spline wavelets using 3D Haar wavelets; consequently, we obtain C^2 quartic spline surfaces over triangular domains by using linear building blocks in 3D. However, a fundamental difference to immediate construction is that we create 7 types of triangular wavelets (instead of 3). A two-scale relationship in the barycentric plane determines the corresponding decomposition and reconstruction operators. Interestingly, the sparsity of the data allows one to perform the respective push and pull operation in linear time and generates a hierarchy of control points for surface editing at different levels of detail. In addition, oracles (Gross [15]) can help to identify and to reject unimportant basis functions, thus governing the compression rate. It should be noted that a similar type of projection was used by Lippert [17] to compute hierarchical splats for volume rendering. In this case, however, the projection was computed via Fourier transforms of the underlying splines and the resulting splat functions were not considered the bases of a barycentric wavelet transform.

The remainder of this paper is organized as follows: Followed by a brief overview, section 2 addresses the mathematical details underlying our approach including projection, two-scale relationship and the resulting barycentric bases. Section 3 introduces efficient decomposition and reconstruction algorithms. Section 4 discusses some issues concerning implementation. Finally, section 5 illustrates the performance of our scheme for multiresolution representation and editing of various types of triangular surfaces.

2 Construction of Triangular B-Spline Bases

2.1 Overview

The basic concept of the construction scheme is depicted in Fig. 1. Let *(u,v,w)* be the barycentric coordinates parametrizing a 2-dimensional triangular domain. We assume the functional values to be given at each triangle vertex. In order to define a multiresolution analysis (MRA), we recall that the barycentric coordinates u, v, w viewed as affine space coordinates represent points on the plane $u + v + w = 1$. However, instead of using these barycentric coordinates, we make use of the homogenous coordinates, which span the diagonal plane in a 3D Euclidean space *(x,y,z)*, and follow the relationship:

$$u + v + w = Const \tag{1}$$

Many years ago, de Boor ([3], [4]) discovered that triangular B-splines - special box splines - can be constructed from 3D tensor product relatives by projection into the barycentric plane E_b (where $u + v + w = Const$). The same idea can be carried over to tensor product B-spline wavelets (Chui [6] or [7]), which are by definition linear combinations of tensor product B-spline bases. The fundamental problem is to define an MRA for the discrete setting.

Conversely, we observe that the above interpretation allows one to map functional values defined over regular triangulations onto the nodes of an equally-spaced 3D tensor product grid.

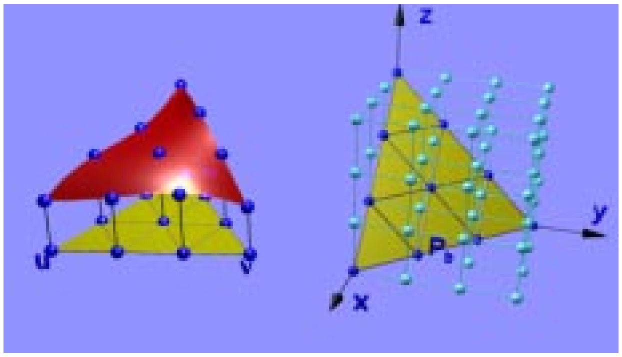


Figure 1: Barycentric plane spanning a 2D subspace in 3D

In principle, such a correspondence enables one to run discrete tensor product algorithms in 3D thereby raising the dimension of the problem by one. A fundamental prerequisite, however, is an appropriate scheme to project the data values onto the nodal positions in 3D and vice versa.

This scheme of mapping functional values over a regular triangular grid to 3D and back is shown is in Fig. 3: Starting with initial values on a triangular mesh $\tilde{c}_0(u, v, w)$

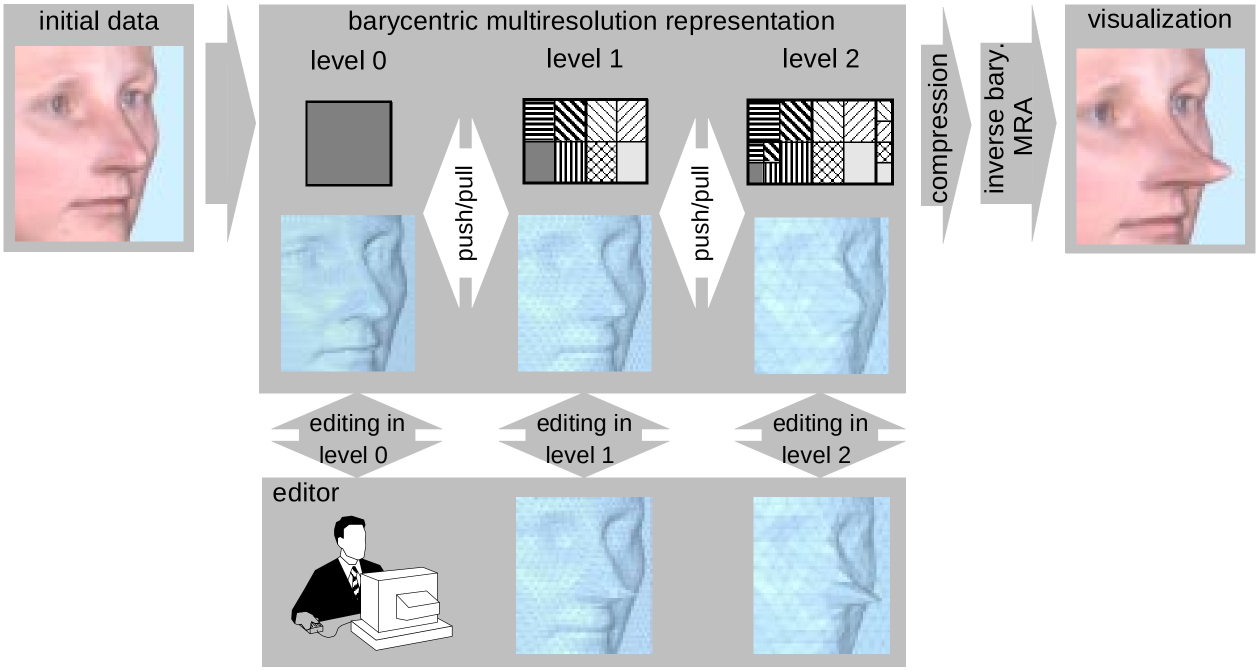


Figure 2: Conceptual components of a multiresolution editor for triangular B-splines.

we define a forward mapping function F

$$\begin{tikzcd}
\text{Barycentric MRA} & & \text{Optimized 3D-MRA} \\
\tilde{c}_0(u, v, w) \arrow[rr, shift left, "F"] \arrow[d, "\text{baryc. MRA}"] & & c_0(x, y, z) \arrow[ll, shift left, "F^{-1}"] \arrow[d, "\text{3D MRA}"] \\
\begin{array}{c}\tilde{c}_m(u, v, w) \\ \tilde{d}^l_m(u, v, w)\end{array} \arrow[rr, shift left, "F"] \arrow[d, "\text{inverse baryc. MRA}"] & & \begin{array}{c}c_m(x, y, z) \\ d^l_m(x, y, z)\end{array} \arrow[ll, shift left, "F^{-1}"] \arrow[d, "\text{inverse 3D-MRA}"] \\
\tilde{c}_0(u, v, w) \arrow[rr, shift left, "F"] & & c_0(x, y, z) \arrow[ll, shift left, "F^{-1}"]
\end{tikzcd}$$

Figure 3: Computational scheme for the barycentric MRA

$$F\colon\ c_m(x, y, z) \rightarrow \tilde{c}_m\left(x - \frac{x+y+z}{3}, y - \frac{x+y+z}{3}, z - \frac{x+y+z}{3}\right) \quad (2)$$

which computes 3D coefficients c_m and perform a 3D-MRA algorithm.

Conformingly, the initial underlying barycentric function $\tilde{f}(u, v, w)$ is mapped into the 3D space by

$$F\colon\ f(x, y, z) = \tilde{f}\left(x - \frac{x+y+z}{3}, y - \frac{x+y+z}{3}, z - \frac{x+y+z}{3}\right) \quad (3)$$

Thus, we obtain coefficients with similar values along the perpendiculars of the barycentric plane. Analyzing the inverse function F^{-1}, it is obvious, that, on the condition that these values are equal along the perpendiculars (with arbitrary i):

$$F^{-1}\colon\ \tilde{c}_m(u, v, w) \rightarrow c_m(u+i, v+i, w+i) \quad (4)$$

It is easy to prove, that one step of the MRA algorithm preserves this condition!

Generally, the complexity of discrete convolution algorithms increases with the dimension of the data. However, as we will demonstrate, the decomposition and reconstruction algorithms can be implemented immediately on the barycentric plane. Thus, it is possible to keep the computational and storage costs to within $O(n)$.

Using these correspondences along with the two-scale-relationship, we can implement a barycentric MRA operating directly on the triangular mesh instead of realizing the 3D-algorithm. Since tensor product wavelet constructions compute 7 independent wavelets, the barycentric projection generates 7 different triangular wavelet types, such as the ones of Fig. 6. A 4:1 subsampling scheme provides an over-representation, which, however, does not affect the performance of the approach, since most wavelet coefficients will vanish.

We implemented this scheme for a multiresolution editor for triangular surfaces, whose conceptual components are depicted in Fig. 2. The input data, initially defined over a uniform triangular grid is decomposed using the barycentric MRA. More precisely, individual push and pull operators enable one to move up and down the hierarchy and generate a hierarchical set of scaling function control points for editing the surface at different resolutions.

The approximation error and the compression gain are governed by some global oracles which reject unimportant coefficients from the hierarchy. The remainder of the paper elaborates on the mathematical and implementation details of the method.

2.2 Constructing Bases by Projection

As explained above, we deduce the barycentric multiresolution analysis from the 3D-MRA. To obtain a barycentric representation, we first have to recall some details of the 3D-wavelet transform (WT). Here the coefficients $c_{m,i,j,k}$ of the transform are computed by inner products < , > of a function *f* with the set of 3D tensor product bases $\varphi_{m,i,j,k}$ (for definition of tensor product wavelets $\psi_{m,i,j,k}$ and scaling functions $\varphi_{m,i,j,k}$ see [20]), denoted by:

$$c_{m,i,j,k} = \int_V f(x,y,z)\cdot\varphi_{m,i,j,k}(x,y,z)dV \tag{5}$$

m: decomposition level.

i, j, k: indices.

The 3D function defined in equation (3) can be reparametrized using the barycentric coordinates (*u*,*v*,*w*) and a parameter *t* running perpendicular to it. Furthermore, it follows that *f* is constant along the lines perpendicular to the barycentric plane, or

$$f(u,v,w) = f(u+t, v+t, w+t)\text{, for all } t \tag{6}$$

Substituting the integration variables

$$dV = dxdydz = dE_b dt \tag{7}$$

in equation (5) we obtain the following formulation for the inner products

$$c_{m,i,j,k} = \int_{E_b}\int_t f(u+t, v+t, w+t)\cdot\varphi_{m,i,j,k}(u+t, v+t, w+t)dE_b dt \tag{8}$$

The special structure of *f* allows us to separate the above integration as follows:

$$c_{m,i,j,k} = \int_{E_b} f(u,v,w)\int_t \varphi_{m,i,j,k}(u+t, v+t, w+t)dE_b dt \tag{9}$$

From here, the respective barycentric basis $\tilde{\varphi}_{m,i,j,k}(x,y,z)$ yields as follows:

$$\tilde{\varphi}_{m,i,j,k}(x,y,z) = \int_t \varphi_{m,i,j,k}(x+t, y+t, z+t)dt \tag{10}$$

Its inner product with any L_2 function defined over the triangular coordinates (*u*,*v*,*w*) is computed corrrespondingly by

$$c_{m,i,j,k} = \int_{E_b} f(u,v,w)\cdot\tilde{\varphi}_{m,i,j,k}(u,v,w)dE_b \tag{11}$$

The same algebraic transforms can be applied in reverse order using the inverse mapping function F^{-1} to get back from 3D into the barycentric world.

This approach can easily be applied to arbitrary wavelets. In Section 2.6 we will use B-spline wavelets of orders 1 and 2.

Note that the integration along the direction of projection raises the degree of the triangular B-spline bases. As a consequence, we obtain C^0 continuous surfaces for 1D-Haar bases and C^2 continuous surfaces in case of 1D linear B-splines.

Note furthermore, that the above equations define a stable framework for barycentric decomposition and reconstruction using the 3D setting and provides that the underlying 3D function *f* exhibits the special structure explained above. In this case $\tilde{\varphi}_{m,i,j,k}$ provides a basis for the corresponding barycentric function $\tilde{f}$.

2.3 Two-Scale-Relationship

The two-scale-relationship [20] links basis functions of different resolution and is fundamental to the design of decomposition and reconstruction algorithms. In order to derive a two-scale-relationship for barycentric scaling functions we start from their 3D relatives given by:

$$\begin{aligned}\varphi_{m,n}(x)\cdot\varphi_{m,o}(y)\cdot\varphi_{m,p}(z) = {} & \left(\sum_i a(i-2n)\varphi_{m-1,i}(x)\right)\\ & \cdot\left(\sum_j a(j-2o)\varphi_{m-1,j}(y)\right)\\ & \cdot\left(\sum_k a(k-p)\varphi_{m-1,k}(z)\right)\end{aligned} \tag{12}$$

n, o, p: offset.

$\{a_i\}$: discrete filter sequence.

Exploiting the line integration property of (10) yields

$$\begin{aligned}\tilde{\varphi}_{m,n,o,p}(x,y,z) = {} & \\ & \int_t \varphi_{m,n}(x+t)\cdot\varphi_{m,o}(y+t)\cdot\varphi_{m,p}(z+t)dt\\ & = \sum_{i,j,k} a(i-2n)\cdot a(j-2o)\cdot a(k-2p)\\ & \cdot\int_t \varphi_{m-1,i,j,k}(x+t, y+t, z+t)dt\end{aligned} \tag{13}$$

with

$$\varphi_{m-1,i,j,k}(x,y,z) = \varphi_{m-1,i}(x)\cdot\varphi_{m-1,j}(y)\cdot\varphi_{m-1,k}(z)$$

To simplify notation a further index transform gives

$$\tilde{\varphi}_{m,n,o,p}(x,y,z) = \tag{14}$$

$$\sum_{\substack{i,j,k \\ i+j+k=0}} \sum_l a(i+l-2n)\cdot a(j+l-2o)\cdot a(k+l-2p)$$
$$\cdot \int_t \varphi_{m-1,\,i+l+n,\,j+l+o,\,k+l+p}(x+t,y+t,z+t)dt$$

l: index.

Due to the infinite range of the integration the integral of the basis function according to point (i,j,k) equals the integral of the basis function through $(i+l,j+l,k+l)$.

This allows one to replace the above integral by a barycentric basis in (i,j,k) thereby establishing the desired relation between two adjacent levels:

$$\tilde{\varphi}_m(x,y,z) = \tag{15}$$

$$\sum_{\substack{i,j,k \\ i+j+k=0}} \left(\sum_l a(i+l-2n)\cdot a(j+l-2o)\cdot a(k+l-2p)\right)$$
$$\cdot\, \tilde{\varphi}_{m-1,i,j,k}(x,y,z)$$

Wavelets can be constructed similarly from their 3D counterparts, where 7 different prototypes are obtained for each level.

2.4 Barycentric Filter Kernels

In order to design an optimized barycentric multiresolution algorithm we have to construct a direct WT over the triangular mesh.

This implies the need for barycentric filters according to the barycentric scaling functions and different wavelets introduced above. We will show that, using the two-scale-relationship, there is an easy and elegant way to compute the corresponding filter coefficients.

We define the filter kernels $\tilde{a}, \tilde{b}^1, \ldots, \tilde{b}^7$ for the barycentric scaling function and the 7 wavelets as follows:

$$\begin{aligned}
\tilde{a}(i,j,k) &= \sum_l a(i+l)\cdot a(j+l)\cdot a(k+l)\\
\tilde{b}^1(i,j,k) &= \sum_l a(i+l)\cdot a(j+l)\cdot b(k+l)\\
\tilde{b}^2(i,j,k) &= \sum_l a(i+l)\cdot b(j+l)\cdot a(k+l)\\
\tilde{b}^3(i,j,k) &= \sum_l b(i+l)\cdot a(j+l)\cdot a(k+l)\\
\tilde{b}^4(i,j,k) &= \sum_l a(i+l)\cdot b(j+l)\cdot b(k+l)\\
\tilde{b}^5(i,j,k) &= \sum_l b(i+l)\cdot a(j+l)\cdot b(k+l)\\
\tilde{b}^6(i,j,k) &= \sum_l b(i+l)\cdot b(j+l)\cdot a(k+l)\\
\tilde{b}^7(i,j,k) &= \sum_l b(i+l)\cdot b(j+l)\cdot b(k+l)
\end{aligned} \tag{16}$$

i, j, k: indices.

Utilizing these definitions, the two-scale-relationship for the scaling function can be expressed by the following equation:

$$\tilde{\varphi}_{m,n,o,p}(u,v,w) = \tag{17}$$

$$\sum_{\substack{i,j,k \\ i+j+k=1}} \tilde{a}(i-2n,j-2o,k-2p)\cdot \tilde{\varphi}_{m-1,i,j,k}(u,v,w)$$

The two-scale-relationship for the 7 different wavelets ($g = 1, \ldots, 7$) yields conformingly.

$$\tilde{\psi}^g_{m,n,o,p}(u,v,w) = \tag{18}$$

$$\sum_{\substack{i,j,k \\ i+j+k=0}} \tilde{b}^g(i-2n,j-2o,k-2p)\cdot \tilde{\psi}^g_{m,n,o,p}(u,v,w)$$

$g = 1, \ldots, 7$.

The resulting barycentric filter kernels are depicted for linear barycentric wavelets in Section 2.6 (Fig. 8).

These filters enable us to construct a barycentric wavelet transformation which operates directly on the triangular mesh. The analysis and synthesis algorithms are discussed in detail in Section 3.

2.5 Orthogonality

Although a rigorous mathematical analysis of the orthogonality properties of the bases is omitted here for brevity, it is necessary to briefly discuss this issue.

Using the inner product operator < , > it is easy to prove that even in the case of 3D tensor product Haar wavelets, orthogonality of both wavelets and scaling functions gets lost. However, since we obtained the barycentric basis from the 3D-MRA, it is necessary to analyze the basis functions in 3D. We will show later, that although we lost orthogonality in the barycentric setting, the *same* operators can be used for decomposition and reconstruction.

2.6 Examples

In our implementation we use cardinal B-spline wavelets [7], since they form a canonic extension of B-splines which are fundamental in geometric modeling. It has to be noted, however, that the construction scheme from above is not restricted to a particular type of wavelet.

Especially Haar and linear spline bases have enormous practical importance, because they enable one to represent piecewise linear and quartic surfaces in the barycentric plane.

Linear Barycentric B-Spline Bases

Applying our scheme to Haar bases, with scaling functions φ and wavelets ψ as shown in Fig. 4, results in piecewise linear C^0-continuous functions. Fig. 6 depicts the bases functions using an intensity plot. We observe eight different types of bases, four of which are displayed in Fig. 7. The filters for the linear barycentric B-Spline bases are presented in Fig. 8, the corresponding 1D filter coefficients are outlined in Appendix B.

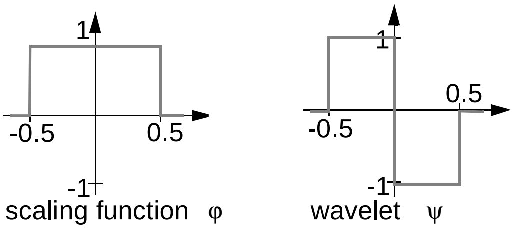


Figure 4: 1D Haar scaling functions and wavelets.

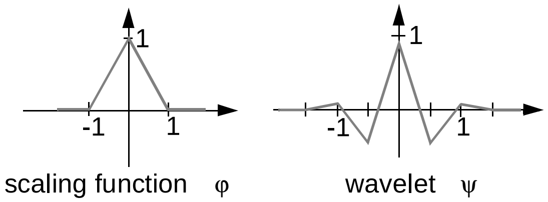


Figure 5: 1D linear B-spline bases according to [7].

Note, that following equation (16) the filters for decomposition and reconstruction are identical, since the 1D filters are equal for analysis and synthesis. The non-orthogonality of the barycentric filters in Fig. 8 and the lack of (different) dual barycentric filters is not a contradiction, because they are only used to simplify the 3D-MRA.

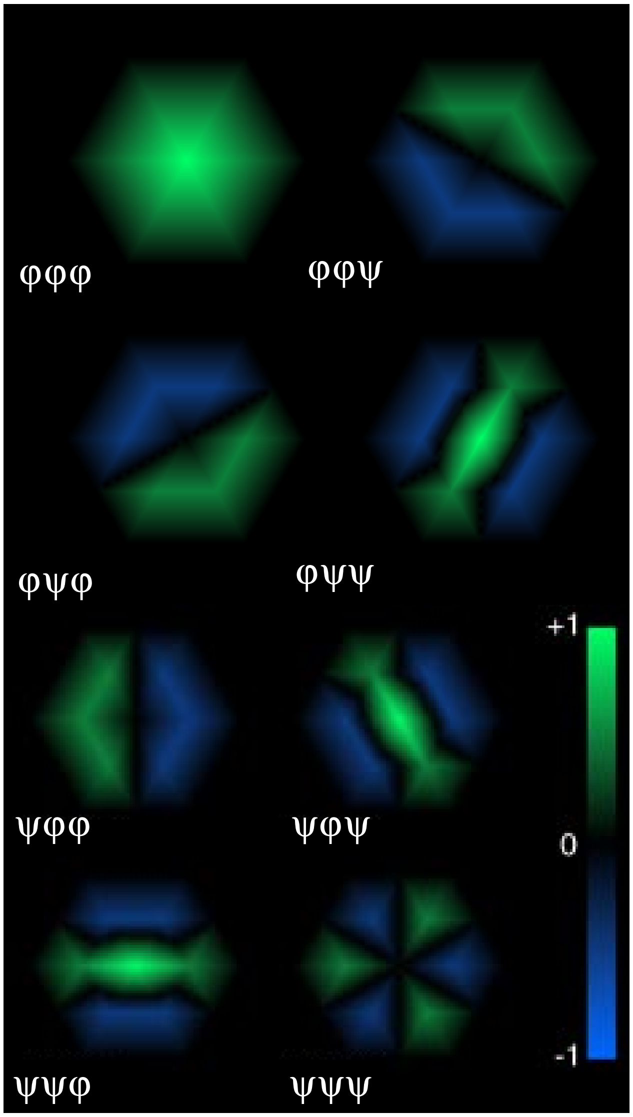


Figure 6: Linear barycentric B-spline bases (intensity plot) (C^0).

The linear dependence of the basis functions is analyzed in Appendix A.

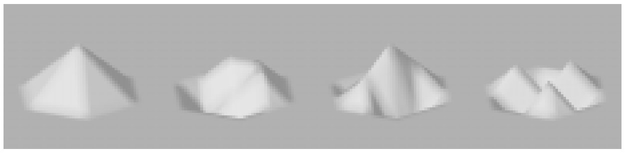


Figure 7: 4 types of linear barycentric B-spline bases.

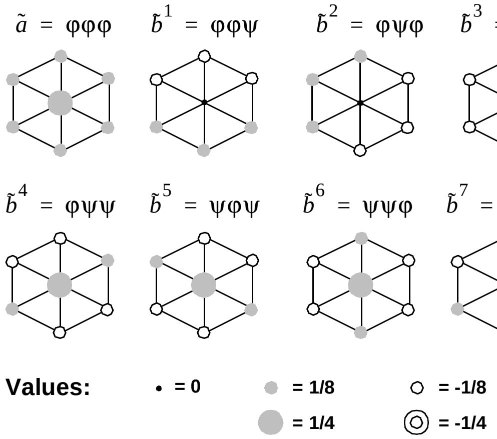


Figure 8: Linear barycentric filter kernels.

Quartic barycentric B-spline bases

One of the many advantageous properties of B-splines is the automatic continuity control. Therefore, the use of trilinear tensor product B-spline bases implies C^2 continuity for the bivariate functions optained by the integration defined above and it furthermore implies an increase of the support. Fig. 5 illustrates the 1D versions.

The corresponding barycentric children (the 8 different barycentric B-Spline bases) are presented in Fig. 9. As a fundamental observation, we distinguish again four different types of basis functions, as given in Fig. 10. We omit the description of the barycentric quartic filter kernels, however, one can calculate those kernels straightforwardly using equation (16). In this case, dual kernels must be computed as well.

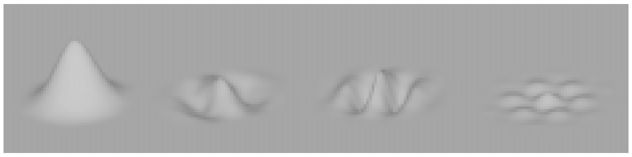


Figure 10: 4 different types of quartic barycentric B-spline bases.

3 Analysis / Synthesis Algorithms

In an earlier version of this algorithm [10] the construction of the barycentric MRA was realized by a 3D multiresolution analysis. The following algorithms represent a substantial improvement of the initial schemes, since they exploit the barycentric definitions from above and operate directly on the triangular mesh. In Appendix A we briefly describe the definition of the barycentric convolution using the 3D convolution.

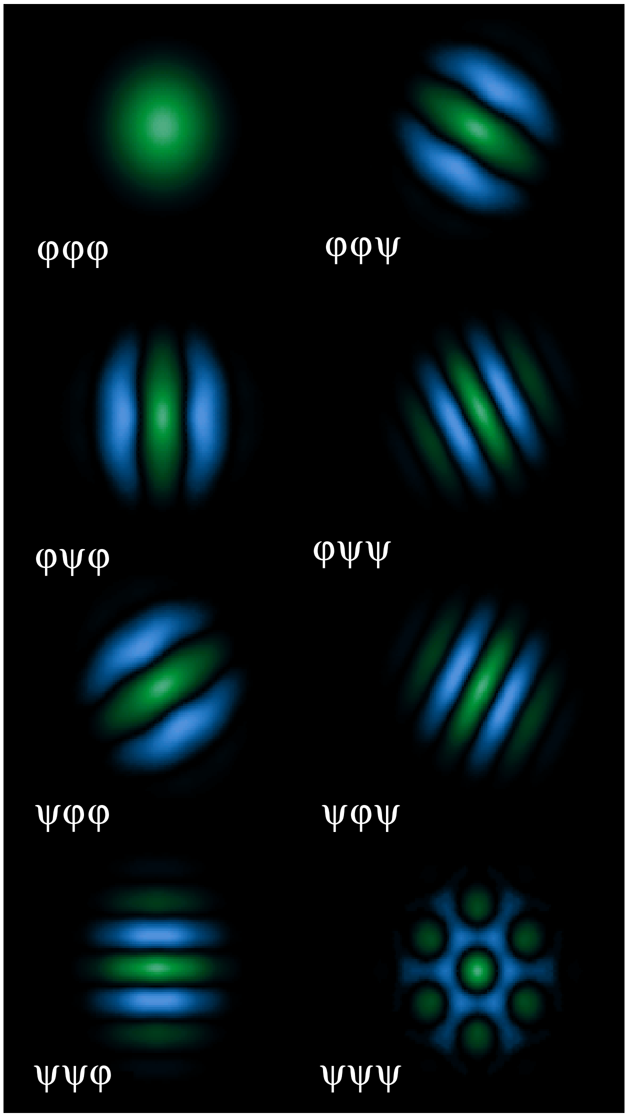


Figure 9: Quartic barycentric B-spline bases (intensity plot) (C^2).

3.1 Barycentric Convolution and Sampling

As already explained earlier, the analysis and synthesis algorithms for the barycentric approach are constructed using the 1D filter sequences $\{a_i\}$ and $\{b_i\}$.

Fig. 11 illustrates that the barycentric MRA, like the conventional MRA, consists of two operations, namely *convolutio*n (with possibly different analysis and synthesis filters) and *sampling* (sub-/upsampling).

In order to calculate the coefficients of a particular level of detail we apply barycentric convolution using sums over barycentric coordinates (Appendix A).

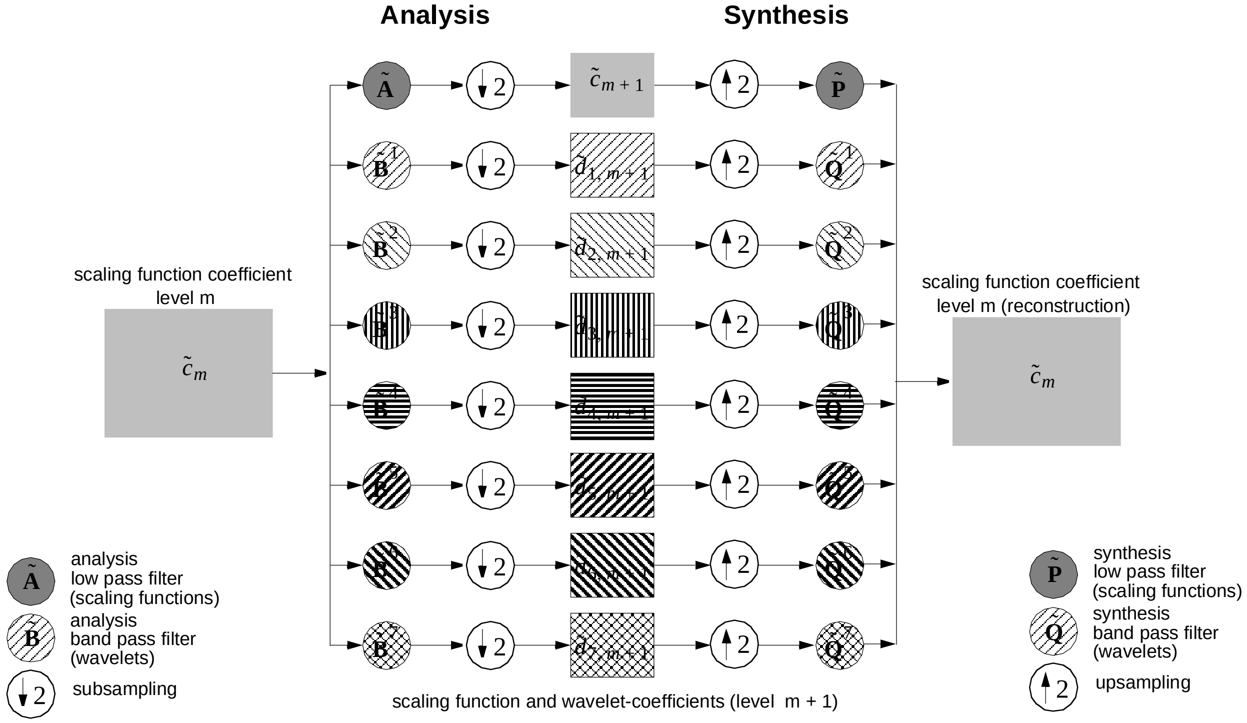


Figure 11: Analysis and synthesis pipeline of the barycentric MRA.

The sampling scheme is illustrated in Fig. 12. We reduce the resolution of the triangular mesh by a factor of 2 in all directions, where, in fact, the 3rd direction is redundant. Thus we obtain a subsampling rate of 1:4, which implies, that - due to the number of basis functions - the data is over-represented. The data organization of the non-critically sampled pyramid is shown in Fig. 15.

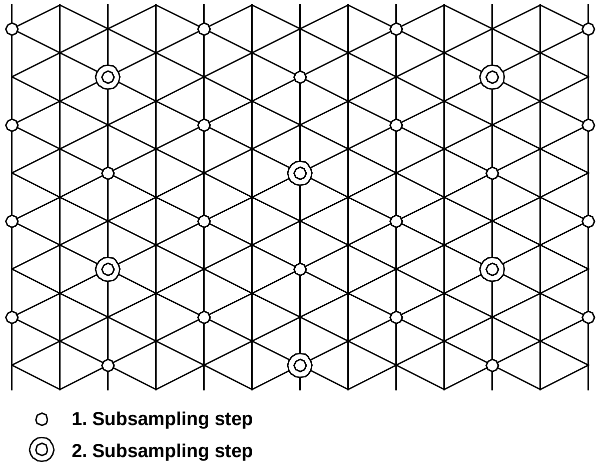


Figure 12: Subsampling of the triangular mesh.

The consequences for the overall computational performance of the barycentric algorithms can be summarized as follows:

- *Sparsity:* The barycentric convolution required to implement the decomposition and reconstruction is sparse; therefore, it requires only linear time O(n) with respect to the data.
- *Directness:* The computational scheme operates immediately on the barycentric plane; thus, rather than storing and maintaining a 3D array (if we perform a 3D MRA), it is sufficient to implement appropriate triangular 2D data structures.

3.2 Decomposition

The corresponding decomposition algorithm can be

$$\begin{tikzcd}
\tilde{c}_0 \arrow[r, "\tilde{\mathbf{A}}"] \arrow[dr, "\tilde{\mathbf{B}}^1"] \arrow[dddr, "\tilde{\mathbf{B}}^7"'] & \tilde{c}_1 \arrow[r, "\tilde{\mathbf{A}}"] \arrow[dr, "\tilde{\mathbf{B}}^1"] \arrow[dddr, "\tilde{\mathbf{B}}^7"'] & \tilde{c}_2 & \cdots \arrow[r, "\tilde{\mathbf{A}}"] \arrow[dr, "\tilde{\mathbf{B}}^1"] \arrow[dddr, "\tilde{\mathbf{B}}^7"'] & \tilde{c}_N \\
 & \tilde{d}_1^1 & \tilde{d}_2^1 & & \tilde{d}_N^1 \\
 & \vdots & \vdots & & \vdots \\
 & \tilde{d}_1^7 & \tilde{d}_2^7 & & \tilde{d}_N^7
\end{tikzcd}$$

Figure 13: Analysis pipeline

outlined in pseudo-code as follows:

```
Initial data:
1- or 3-dimensional array of function values:
$c_0$[u, v, w], (u+v+w=0)
scaling and wavelet filter kernels for the decomposition:
a[i, j, k], $b^l$[i, j, k],  (i+j+k=0, l=1,...,7)

push()
{
for m ← 0 to N-1   // MRA-level m
{
    // convolution using filters a and b
    // with subsampling
    for (u,v,w) ← ($u_{min}$, $v_{min}$, $w_{min}$) to ($u_{max}$, $v_{max}$, $w_{max}$),
            u+v+w=0
    {
        $c_{m+1}$[u/2,v/2,w/2]  = Convolution(a[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^1_{m+1}$[u/2,v/2, w/2] = Convolution($b^1$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^2_{m+1}$[u/2,v/2, w/2] = Convolution($b^2$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^3_{m+1}$[u/2,v/2, w/2] = Convolution($b^3$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^4_{m+1}$[u/2,v/2, w/2] = Convolution($b^4$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^5_{m+1}$[u/2,v/2, w/2] = Convolution($b^5$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^6_{m+1}$[u/2,v/2, w/2] = Convolution($b^6$[i,j,k], $c_m$[u+i,v+j,w+k])
        $d^7_{m+1}$[u/2,v/2, w/2] = Convolution($b^7$[i,j,k], $c_m$[u+i,v+j,w+k])
    }
}
```

This algorithm reflects an analysis pipeline with a maximum iteration depth of *N*-1 (Fig. 13).

3.3 Reconstruction

In analogy to the decomposition, the subsequent pseudo-code gives the reconstruction algorithm:

```
Initial data:
Scaling coefficients (level N)
$c_N$[u,v,w], (u+v+w=0)
Wavelet coefficients (level 1,...,N)
$d^l_m$[u,v,w], (u+v+w=0, m=1,...,N; l=1,..7)
Scaling and wavelet filter kernels for the reconstruction (semi-
orthogonal):
p[i,j,k], $q^l$[i,j,k], (i+j+k=0, l=1,...,7)

pull()
{
for m ← N-1 to 0  Step -1 // MRA-level m
{
    // convolution using filters p and $q^l$
    // including upsampling
    for (u,v,w) ← ($u_{min}$, $v_{min}$, $w_{min}$) to ($u_{max}$, $v_{max}$, $w_{max}$),
    {
            u+v+w=0
        $c_m$[u,v,w] = Convolution(p[i,j,k],  $c_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^1$[i,j,k], $d^1_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^2$[i,j,k], $d^2_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^3$[i,j,k], $d^3_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^4$[i,j,k], $d^4_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^5$[i,j,k], $d^5_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^6$[i,j,k], $d^6_{m+1}$[2u+i,2v+j,2w+k])
                + Convolution($q^7$[i,j,k], $d^7_{m+1}$[2u+i,2v+j,2w+k])
    }
}
```

The corresponding synthesis pipeline is depicted in Fig. 14.

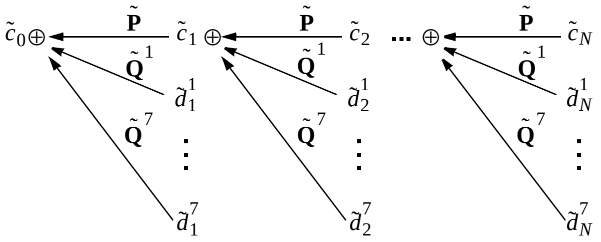


Figure 14: Synthesis pipeline

Note, that the MRA scheme allows one to edit or reject individual coefficients in the hierarchy without any restrictions.

3.4 Compression and Complexity

One major advantage of the wavelets is their compression performance resulting from the vanishing moments. However, the rejection of unimportant coefficients is a non-trivial, discrete, global optimization problem in semi-orthogonal settings [15], although experiments have shown that magnitude based rejection often performs well.

Storage expense and computational complexity are mainly influenced by the number of basis functions. Unlike the direct 2D-MRA [23] which employs 4 basis functions, the barycentric non-critically sampled setting used in our approach produces 7 coefficients at each level of resolution. This implies an over-representation of the data which can be organized in a modified pyramidal setup, such as the one presented in Fig. 15.

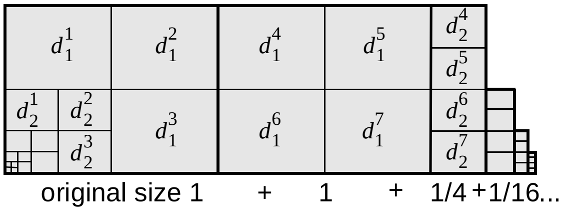


Figure 15: Pyramidal data structure used for the barycentric MRA.

As the maximum decomposition level increases, the storage expense converges to 2 1/3 of the initial data.

This over-representation requires to investigate possible linear dependencies between individual types of basis functions. An analysis yields a linear dependence for the basis functions for Haar-based constructions. In addition, it is possible to represent all 7 wavelet types using an infinite sequence of 3 derived wavelets and the scaling function (see Appendix A).

4 Implementation

The presented multiresolution analysis approach for triangular data was used to implement a multiresolution surface editor. Our prototype system comprises the following functionalities:

- Modification of triangular surfaces at several levels of detail (multiresolution editing).
- Piecewise linear and piecewise quartic basis functions.
- Pushing and pulling along the hierarchies.
- Editing of hightfields and parametric surfaces.

The scaling function coefficients generated at different levels in the hierarchy form the control points and enable editing of the shape at different resolutions. The smoothness of the surface and of the editing operation depends on the polynomial degree of the bases. This is illustrated in Fig. 16, where we lifted one scaling function coefficient for linear and quartic representations.

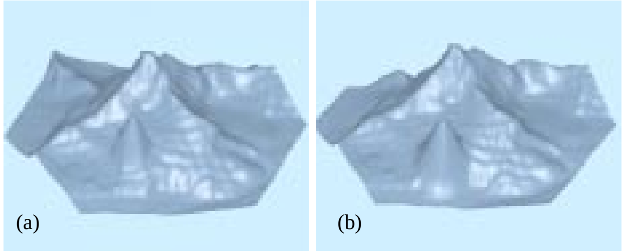


Figure 16: Editing of an individual coefficient at decomposition level *m*=3 using baryc. linear bases (a) and quartic bases (b).

The surface region affected by an individual scaling function conforms to its spatial support and scales according to (15) exponentially with the decomposition level. Here the power of the multiresolution editing concept allows the user to push/pull along the hierarchies in order to realize modifications from a rough, global scale to fine grain details.

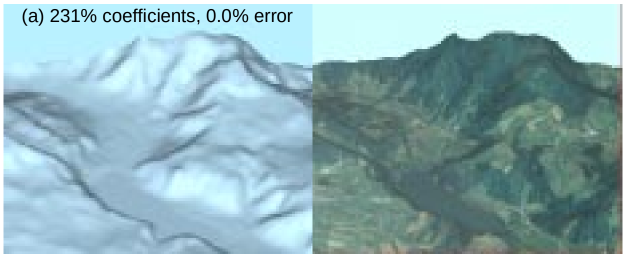


Figure 17: Compression performance of the method on a digital terrain data set: MRA with maximum decomposition level 3, using linear bases.

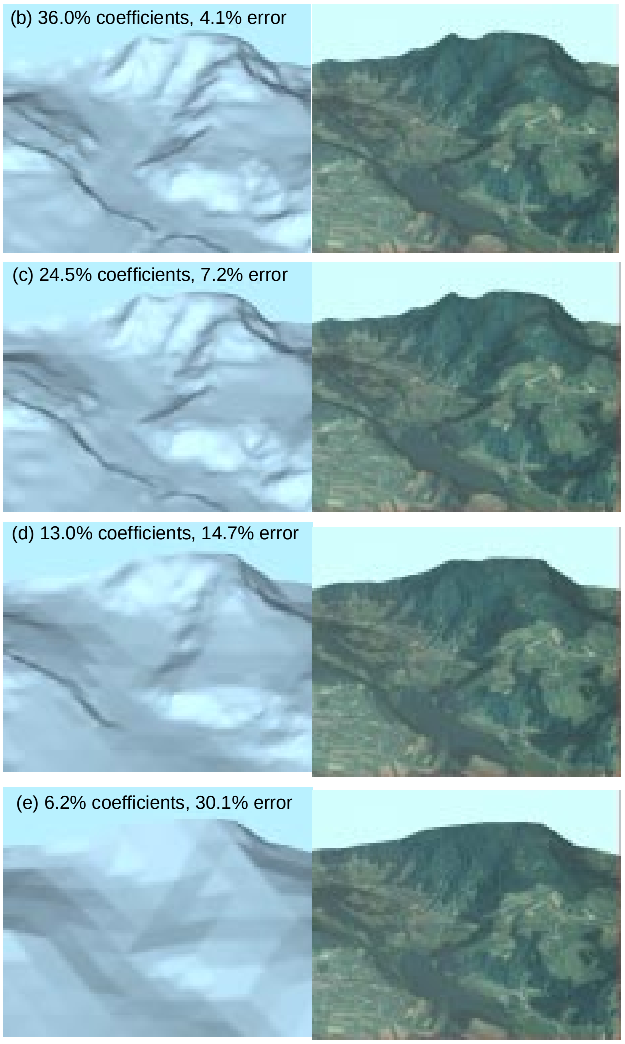
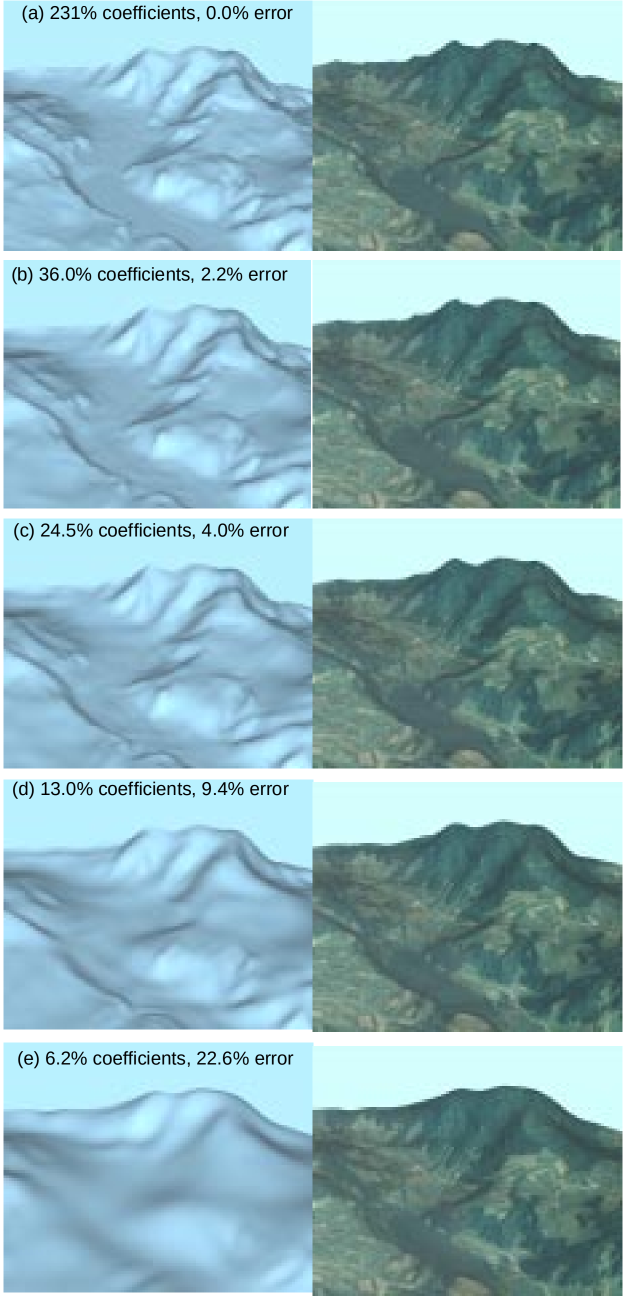

(b) 36.0% coefficients, 4.1% error
(c) 24.5% coefficients, 7.2% error
(d) 13.0% coefficients, 14.7% error
(e) 6.2% coefficients, 30.1% error

Figure 17: Compression performance of the method on a digital terrain data set: MRA with maximum decomposition level 3, using linear bases.

5 Results

This section illustrates the usefulness and performance of the approach in the context of surface design and editing. Therefore, we implemented both linear and quartic barycentric basis functions. In the first series of images, displayed in Fig. 17 and Fig. 18, we reconstructed a digital terrain model using different ratios of basis functions. Although a full reconstruction requires almost 231% of the original data, the oracle easily computes a 1:3 compression without notable degradation of surface quality, as presented in Fig. 17 b and Fig. 18 b. As expected, most of the computed coefficients do not contribute much to the overall surface quality and, hence, can be zeroed out. The corresponding L_2 errors are presented in terms of percent-



(a) 231% coefficients, 0.0% error

(b) 36.0% coefficients, 2.2% error

(c) 24.5% coefficients, 4.0% error

(d) 13.0% coefficients, 9.4% error

(e) 6.2% coefficients, 22.6% error

Figure 18: Compression performance of the method on a digital terrain data set: MRA with maximum decomposition level 3, using quartic bases.

age of error energy. We contrasted linear and quartic, respectively. The difference in smoothness between the C^0 and C^2 continuous surfaces is particularly striking for high compression rates, such as in Fig. 17 e and Fig. 18 e. For the right-hand side sequence of images, we added satellite images to enhance visual quality.

Multiresolution surface editing is depicted in Fig. 19 where a parametric range data set of a human face is edited at different levels of detail. The left-hand images show the corresponding control meshes of the surface, where Fig. 19 a, c represent the mesh at level m=1. Pushing further down the hierarchies as in Fig. 19 e, g enables one to edit the mesh at m=2. Again, the linear C^0 surfaces are contrasted against quartic C^2 representations. The difference in the smoothness of the surface is particularly striking when comparing Fig. 19 f and h. We observe that an editing operation at level m=1 affects the surface only locally in a small neighborhood of the control vertex. Since the local support of the barycentric spline bases increases dyadically at each level, the same operation performed at level m=2 affects a larger part of the surface. The push and pull operations explained in the previous sections allow one to switch between individual levels and to edit and design the surface at different scales. Higher levels allow interactions which have a global impact on the surface, whereas, lower levels are localized and allow to shape small surface details.

6 Conclusion and Future Work

We presented a novel approach for the construction of multiresolution B-spline surfaces over triangular domains. Rather than trying to compute the spline bases immediately we generated them through projection of 3D tensor product basis functions onto the barycentric plane. This powerful concept, which had already been used years ago to design box splines, provides an elegant way to generate triangular B-spline scaling functions and wavelets of arbitrary polynomial order. We have implemented this approach for linear and quartic bases in a multiresolution mesh editor and illustrated its usefulness and performance by various examples. It is our belief that multiresolution editing is an extremely powerful notion which may be used in future generations of CAD and modeling systems.

Future work must include proofs of some of the fundamental mathematical properties of the bases, such as stability and frame bounds. In addition, focus will be given to a reduction of the over-representation and to boundary problems. Furthermore, some research will be conducted to construct globally C^2 continuous representations of surfaces of arbitrary topological type.

7 Acknowledgment

The authors thank the Bundesamt für Landestopographie, Bern, Switzerland for providing the digital terrain model of Matterhorn. Further appreciation is given to the Swissphoto Vermessung AG for providing the digital terrain model of Albis and to the Computer Graphics Center,

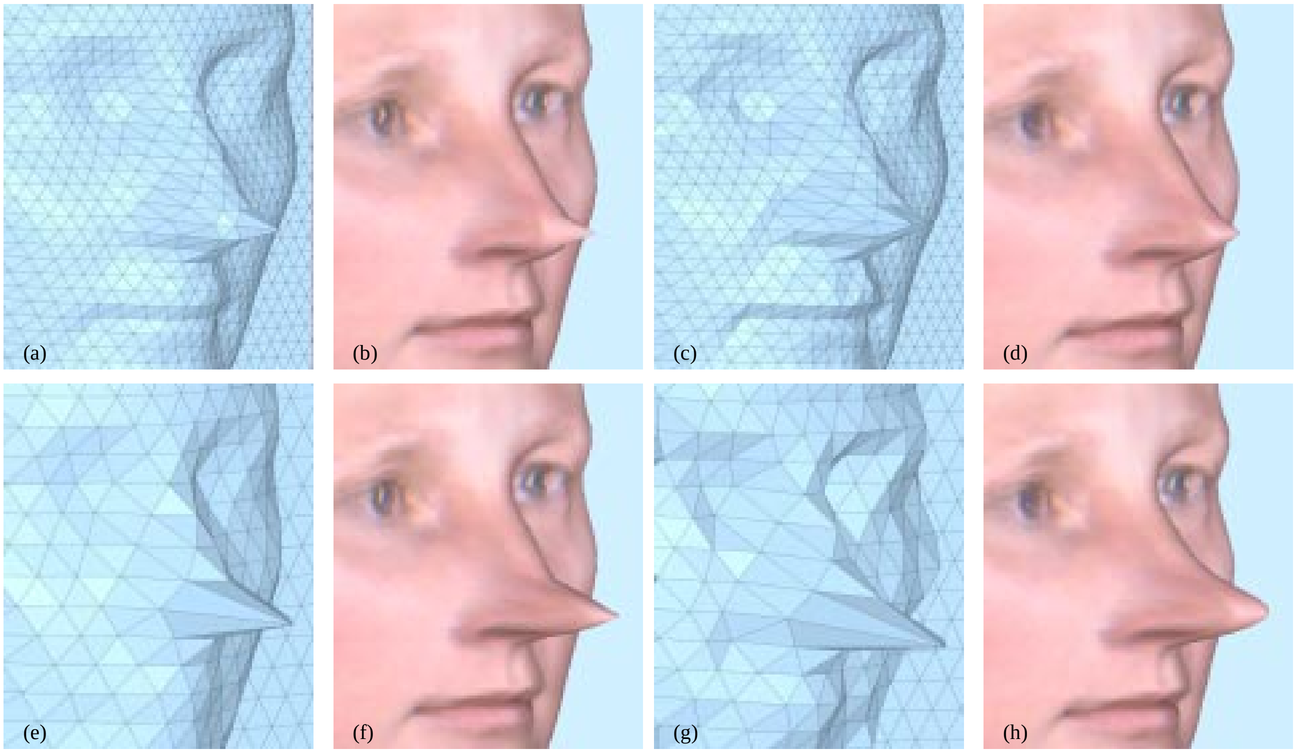


Figure 19: Editing Silvia's face at different resolutions: level 1 ((a)-(d)) and level 2 ((e)-(h)) using Haar bases (a),(b),(e),(f) and linear bases (c),(d),(g),(h).

Darmstadt, Germany, from where Sylvia's range image was obtained.

8 References

[1] W. Boehm, H. Prautzsch and P. Arner. "On Triangular Splines." *Constr. Approx*, 3:157-167, 1987.

[2] W. Boehm: "Calculating with box splines." *Computer Aided Geometric Design*, 1:149-162, 1984.

[3] C. de Boor. "On the evaluation of box splines." *Numerical Algorithms*, 5:5-23, 1993.

[4] C. de Boor, K. Hoellig and S. Riemenschneider. "Box Splines", *Applied Mathematical Sciences 98*, Springer-Verlag, 1993.

[5] M. D. Buhmann and C. A. Micchelli. "Spline prewavelets for non-uniform knots." *Numerische Mathematik*, 61(4):455–474, May 1992.

[6] C. K. Chui and E. Quak. "Wavelets on a bounded Interval." In D. Braess and L. L. Schumaker, editors, *Numerical Methods of Approximation Theory*, pages 1-24. Birkhaeuser Verlag, Basel, 1992.

[7] C. K. Chui and J. Z. Wang. "A cardinal spline approach to wavelets." *Proc. Amer. Math. Soc.*, 113:785–793, 1991.

[8] A. Cohen, I. Daubechies, and J. Feauveau. "Bi-orthogonal bases of compactly supported wavelets." *Comm. Pure and Applied Mathematics*, 45:485-560, 1992.

[9] D. Doo, and M. Sabin. "Analysis of the Behavior of Recursive Division Surfaces near Extraordinary Points." *Computer Aided Design*, 10(6):356-360. 1978.

[10] A. Dreger, M. H. Gross and J. Schlegel. "Multiresolution Triangular B-Spline-Surfaces." *Computer Graphics International 1998 Proceedings*, 166-177, June 1998.

[11] N. Dyn, D. Levin, and J. Gregory. "A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control." *Transactions on Graphics*, 9(2):160-169, April 1990.

[12] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. "Multiresolution analysis of arbitrary meshes." In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 173–182. ACM SIGGRAPH, Addison Wesley, Aug. 1995.

[13] A. Finkelstein and D. H. Salesin. "Multiresolution Curves." *Proceedings of SIGGRAPH '94*, 261-268, July 1994.

[14] D. Forsey and R. Bartels. "Hierarchical B-spline refinement." *Computer Graphics*, 22(4):205-212, 1988.

[15] M. H. Gross. "L^2 optimal oracles and compression strategies for semiorthogonal wavelets." Technical Report 254, Computer Science Department, ETH Zürich, 1996. http://www.inf.ethz.ch/publications/tr200.html.

[16] M. H. Gross, O. G. Staadt, and R. Gatti. "Efficient Triangular Surface Approximations using Wavelets and Quadtree Data Structures." *IEEE Transactions on Visualization and Computer Graphics*, 2(2): 130-143, 1996.

[17] L. Lippert, M. Gross, and C. Kurmann. "Compression domain volume rendering for distributed environments." In *Proceedings of Eurographics '97*, 95-107, 1997.

[18] C. Loop. “Smooth Spline Surfaces over Irregular Meshes.” *Computer Graphics Proceedings*, 303-310, 1994

[19] J. M. Lounsbery. “Multiresolution Analysis for Surfaces of Arbitrary Topological Type.” PhD thesis, University of Washington, Seattle, 1994.

[20] S. Mallat. “A theory for multiresolution signal decomposition: The wavelet respresentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 11(7), 674-693, 1989.

[21] G. M. Nielson. I. Jung, and J. Sung. “Haar Wavelets over Triangular Domains with Application to Multiresolution Models for Flow over a Sphere.” *IEEE Visualization Proceedings ‘97*, 143-149, 1997.

[22] E. Quak and N. Weyrich. “Decomposition and reconstruction algorithms for spline wavelets on a bounded interval.” *Applied and Computational Harmonic Analysis*, 1(3):217–231, June 1994.

[23] P. Schroeder and W. Sweldens. “Spherical Wavelets: Efficiently Representing Functions on the Sphere.” *ACM SIGGRAPH ‘95 Proceedings*, 161--172, 1995.

[24] H.-P. Seidel. “Polar Forms and Triangular B-Spline Surfaces.” Blossoming: The New Polar-Form Approach to Spline Curves and Surfaces, *SIGGRAPH ’91 Course Notes*, 26, 8.1-8.52, 1991.

[25] E. Simoncelli and E. Adelson. “Non-Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions.” *Proceedings of the IEEE,* 78(4):652-664, 1990.

[26] O. G. Staadt, M. H. Gross and R. Weber. “Multiresolution Compression and Reconstruction.” *IEEE Visualization Proceedings ‘97*, 337-346, 1997.

[27] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, Inc., 1996.

[28] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. “Wavelets for computer graphics: A primer.” *IEEE Computer Graphics and Applications*, 15(3):76–84, May 1995 (part 1) and 15(4):75–85, July 1995 (part 2).

[29] Taubin, G. “A Signal Processing Approach to Fair Surface Design.” *SIGGRAPH 95 Conference Proceedings*, R. Cook, Ed. Annual Conference Series, 351-358, August 1995.

A PROOFS

Linear dependence and over-representation

As proved by many signal processing applications, such as image or video encoding, over-representation is not necessarily a disadvantage. In order to further investigate the inherent over-representation of our scheme, we have to examine possible linear dependencies of individual basis functions. In this context, we followed two different strategies:

- The straightforward linear dependence of the 8 basis functions in one point of the grid at a particular level of detail, which might lead to a more compact representation of coefficients by direct linear combinations of others.
- A more general representation of individual types of basis functions (e.g one type of wavelets) using all other basis functions in the barycentric plane at the same level of detail.

Direct linear dependencelinear dependence of the 8 basis functions

To investigate the linear dependence of the 8 different kinds of barycentric basis functions we start from the two-scale relationship:

$$\tilde{\varphi}_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} a(i+l)\cdot a(j+l)\cdot a(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^1_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} a(i+l)\cdot a(j+l)\cdot b(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^2_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} a(i+l)\cdot b(j+l)\cdot a(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^3_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} b(i+l)\cdot a(j+l)\cdot a(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^4_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} a(i+l)\cdot b(j+l)\cdot b(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^5_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} b(i+l)\cdot a(j+l)\cdot b(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^6_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} b(i+l)\cdot b(j+l)\cdot a(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

$$\tilde{\psi}^7_m(X) = \sum_{\substack{i,j,k,l \\ i+j+k=0}} b(i+l)\cdot b(j+l)\cdot b(k+l)\cdot \tilde{\varphi}_{m-1,i,j,k}(X)$$

with

$$X = (x, y, z) \in R^3$$

The barycentric wavelets are linear combinations of shifted versions of one prototype function weighted with coefficients of type

$$\left(\sum_l g(i+l)\cdot g(j+l)\cdot g(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}} \tag{19}$$

where g stands for the filter kernels $\{a\}$ and $\{b\}$.

We rewrite the relations in a vector form, where the individual coefficients form the entries of the corresponding vector:

$$\tilde{\Phi} = \left(\sum_l a(i+l) \cdot a(j+l) \cdot a(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_1 = \left(\sum_l a(i+l) \cdot a(j+l) \cdot b(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_2 = \left(\sum_l a(i+l) \cdot b(j+l) \cdot a(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_3 = \left(\sum_l b(i+l) \cdot a(j+l) \cdot a(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_4 = \left(\sum_l a(i+l) \cdot b(j+l) \cdot b(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_5 = \left(\sum_l b(i+l) \cdot a(j+l) \cdot b(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_6 = \left(\sum_l b(i+l) \cdot b(j+l) \cdot a(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

$$\tilde{\Psi}_7 = \left(\sum_l b(i+l) \cdot b(j+l) \cdot b(k+l)\right)_{\substack{i,j,k \\ i+j+k=0}}$$

From here we analyze a possible linear dependence by computing the nullspace of the basis Matrix $\mathbf{B}_L$ with:

$$\mathbf{B}_L = \left[\tilde{\Phi}\ \tilde{\Psi}_1\ \tilde{\Psi}_2\ \tilde{\Psi}_3\ \tilde{\Psi}_4\ \tilde{\Psi}_5\ \tilde{\Psi}_6\ \tilde{\Psi}_7\right] \qquad (20)$$

For Haar bases $\mathbf{B}_L$ is given by:

$$\mathbf{B}_L = \frac{1}{2}\begin{bmatrix} 2 & 0 & 0 & 0 & 2 & 2 & 2 & 0 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \qquad (21)$$

Its nullspace can be computed straightforwardly to

$$Nullspace(\mathbf{B}_L) = \{[0, 1, 1, 1, 0, 0, 0, 1]\} \qquad (22)$$

Obviously, the barycentric wavelets constructed from the Haar basis are linear dependent.

A similar analysis for the quartic wavelets yields

$$Nullspace(\mathbf{B}_Q) = \{\ \} \qquad (23)$$

that is no direct linear dependence.

General linear dependence of the basis functions

The computations from above are not fully sufficient to determine all possible linear dependencies of basis functions. In order to estimate the dimension of linear dependence, all translations of the wavelets and the scaling functions of the same decomposition level must be considered.

It is possible to prove, that there is no finite representation of the wavelets, except for linear barycentric wavelet, presented above. For the sake of brevity we omit the full proof. As an example, we take the linear barycentric basis functions to construct an infinite basis of 4 functions, that represent the whole space. This case illustrates the fundamental ideas of the full proof.

Recalling the barycentric filter kernels, as described in equation (16), we deduce a set of new filters as shown in Fig. 20.

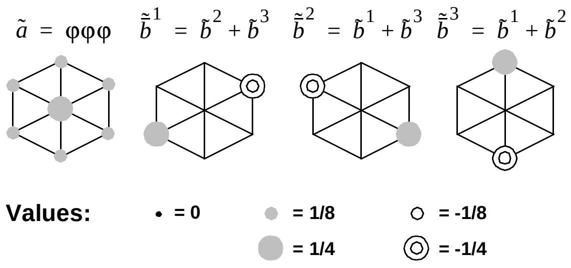

$\tilde{a} = \varphi\varphi\varphi$
$\tilde{\tilde{b}}^1 = \tilde{b}^2 + \tilde{b}^3$
$\tilde{\tilde{b}}^2 = \tilde{b}^1 + \tilde{b}^3$
$\tilde{\tilde{b}}^3 = \tilde{b}^1 + \tilde{b}^2$
Values:
• = 0
● = 1/8
○ = -1/8
● = 1/4
◎ = -1/4

Figure 20: Barycentric linear filter basis.

Note that these kernels are orthogonal.

Using the definitions

$$m = (i, j, k) \qquad (24)$$
$$u = (1, 0, 0);\ v = (0, 1, 0);\ w = (0, 0, 1)$$

the remaining filter coefficients can be determined as follows:

$$\begin{aligned} \tilde{b}^4 &= \tilde{a} + \sum_{l=1}^{\infty} [-\tilde{\tilde{b}}^1_{m+lu} + \tilde{\tilde{b}}^1_{m-lu} - \tilde{\tilde{b}}^2_{m+lu} + \tilde{\tilde{b}}^2_{m-lu}] \\ \tilde{b}^5 &= \tilde{a} + \sum_{l=1}^{\infty} [-\tilde{\tilde{b}}^1_{m+lu} + \tilde{\tilde{b}}^1_{m-lu} - \tilde{\tilde{b}}^3_{m+lu} + \tilde{\tilde{b}}^3_{m-lu}] \\ \tilde{b}^6 &= \tilde{a} + \sum_{l=1}^{\infty} [-\tilde{\tilde{b}}^2_{m+lu} + \tilde{\tilde{b}}^2_{m-lu} - \tilde{\tilde{b}}^3_{m+lu} + \tilde{\tilde{b}}^3_{m-lu}] \\ \tilde{b}^7 &= \tilde{b}^1 + \tilde{b}^2 + \tilde{b}^3 \end{aligned} \qquad (25)$$

The above equations express the linear dependencies of the filter kernels of the linear barycentric basis functions. From here, the linear dependencies of the associated basis functions follows immediately. We expect this observation to be a promising way to handle (remove) the over-representation and we will focus on this topic in our future work.

Barycentric convolution operator

In order to use the barycentric filters and the barycentric MRA, we have to establish the correspondences between the 3D- and barycentric convolution operator:

$$
\begin{aligned}
\tilde{c}_{m+1}(u, v, w) &= c_{m+1}(u, v, w)\\
&= \sum_{n, o, p} a(n)\cdot a(o)\cdot a(p)\cdot c_m(u+n, v+o, w+p)\\
&= \sum_{\substack{n, o, p\\ n+o+p = 0}} \sum_{l} a(n+l)\cdot a(o+l)\cdot a(p+l)\cdot c_m(u+n+l, v+o+l, w+p+l)\\
&= \sum_{\substack{n, o, p\\ n+o+p = 0}} \tilde{a}(n, o, v)\cdot c_m(u+n, v+o, w+p)\\
&= \sum_{\substack{n, o, p\\ n+o+p = 0}} \sum_{l} \tilde{a}(n, o, v)\cdot \tilde{c}_m(u+n, v+o, w+p)
\end{aligned}
\tag{26}
$$

The convolution operator in 3D is calculated by convolution in all 3 directions of the 3D grid. A simple transformation using the sum over the barycentric plane and building the convolution along the perpendiculars yields the convolution operator on triangular meshes. Here it is important to remember the equality of

$$c_m(u, v, w) = c_m(u+l, v+l, w+l) \tag{27}$$

which is obvious since we build the 3D coefficients using the mapping operator F as described in (2) and (4).

The convolution of the wavelet coefficients is calculated in the same way.

B EXAMPLES

Haar basis (1D)

The operators for Haar wavelets are trivial and given by

$$a = \left[\frac{1}{2}, \frac{1}{2}\right] \quad , \quad b = \left[\frac{1}{2}, -\frac{1}{2}\right] \tag{28}$$

where a denotes the low pass filter (scaling function) and b denotes the band pass filter (wavelet). Orthogonality forces the inverse operators to equal the transpose.

Linear basis (1D)

Linear B-splines, as proposed by [6] are semi-orthogonal. Hence, we have different filters for analysis and synthesis with the sequences

$$a = [a_{10}, a_9, a_8, a_7, a_6, \ldots, a_1, a_0, a_1, \ldots, a_{10}] \tag{29}$$

$$b = [b_9, b_8, b_7, \ldots, b_1, b_0, b_1, \ldots, b_9] \tag{30}$$

and

$$p = [p_2, p_1, p_2] \;\; ; \;\; q = [q_3, q_2, q_1, q_2, q_3] \tag{31}$$

with given constants

a_1 = 0.6830127	b_1 = 0.8660254	$p_1 = 1$
a_2 = 0.3169873	b_2 = -0.3169873	
a_3 = -0.1160254	b_3 = -0.2320508	$p_2 = \frac{1}{2}$
a_4 = -0.0849365	b_4 = 0.0849365	
a_5 = 0.0310889	b_5 = 0.0621778	$q_1 = \frac{10}{12}$
a_6 = 0.0227587	b_6 = -0.0227587	
a_7 = -0.0083302	b_7 = -0.0166605	$q_2 = -\frac{6}{12}$
a_8 = -0.0060982	b_8 = 0.0044642	
a_9 = 0.0022321	b_9 = -0.0016340	$q_3 = \frac{1}{12}$
a_{10} = 0.0016340		