

## Multiresolution Triangular B-Spline Surfaces

A. Dreger, M. H. Gross, J. Schlegel

Department of Computer Science

Swiss Federal Institute of Technology (ETH), Zurich, 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 decomposition and reconstruction schemes operate in principle on a tensor product grid in 3D, the sparsity of the arrangement enables us to design efficient linear time algorithms. The resulting basis functions are used to approximate triangular surfaces and provide 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 literature. One of the pioneers (Forsy et al. [13]) 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 [21], designed sets of semi-orthogonal wavelets to span the associated difference spaces and used them in the context of curve and surface design (Finkelstein [12]), or to implement surface compression (Stadt [25]) and triangulation [15]. Due to the limitations of tensor product constructions, various researchers proposed multiresolution surface representations over triangular domains. Here Lounsbery [18] or Eck [11], for instance, proposed linear bi-orthogonal wavelet bases to efficiently describe triangular meshes. Others, such as Schroeder [22] or Nielson [20], 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 [24]). 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 [10], Loop [17], or Doo [9], turned out to be a promising alternative. Using them (Schroeder [22]) in combination with generalized subsampling operators (Taubin [28]), 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 the barycentric domain (Seidel [23]), we recalled some fundamental work on box splines from Boehm [2] and de Boor [3], [4]. Here, the basic idea was to construct a B-spline basis function over a triangular domain by projecting a 3D tensor product spline onto the barycentric plane. The line integration inherent to the projection operator raises the degree of the resulting triangular B-spline. 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  $C^2$  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. Here we accept an over-representation in the pyramid, that is, we do not sample critically. 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 [14]) 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 [16] 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 coordinates parametrizing a 2-dimensional triangular domain. We assume, in the discrete setting, the functional values to be given at each triangle vertex. In order to define a multiresolution analysis (MRA), we recall the notion of barycentric coordinates, which span the diagonal plane in a 3D Euclidean space  $(x, y, z)$ , and follow the relationship:

$$u + v + w = Const \quad (1)$$

Many years ago, de Boor ([3], [4]) discovered that triangular B-splines - so-called box splines - can be constructed from 3D tensor product relatives by projection into the barycentric plane  $E_b$ . 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 upper interpretation allows one to map sets of functional values defined over regular triangulations onto the nodes of an equally-spaced 3D tensor product grid.

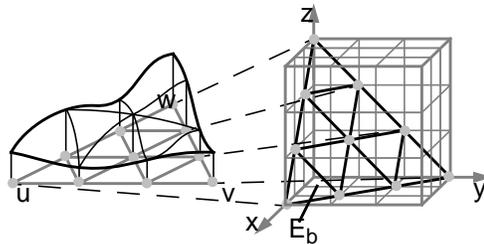
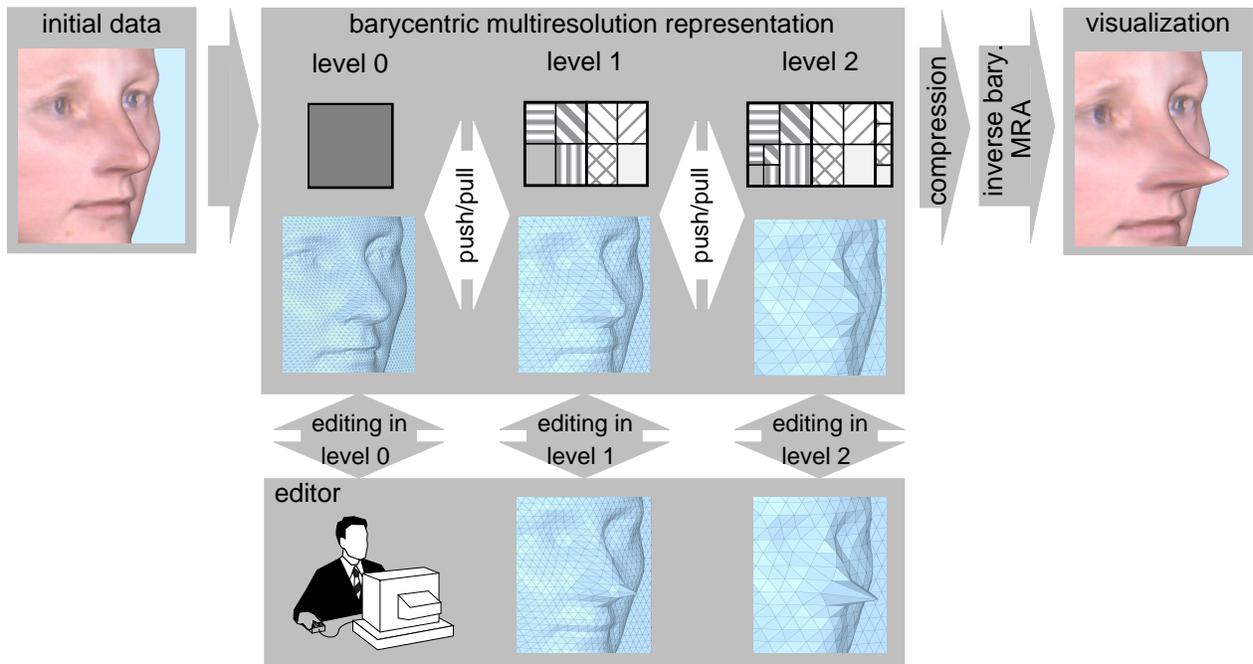


Figure 1: Barycentric plane spanning a 2D subspace in 3D

In principle, this 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.

Generally, the complexity of discrete convolution algorithms scales 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 in  $O(n)$ .

Using these relationships, we can implement a barycentric MRA implicitly in terms of a 3D tensor product MRA. Because tensor product wavelet constructions compute 7 independent wavelets, the barycentric projection



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

generates 7 different triangular wavelet types, such as the ones of Fig. 9. A 4:1 subsampling scheme, along with the projected wavelets, provides an overrepresentation, 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 product 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  $\phi_{m,i,j,k}$  see [19]), denoted by :

$$c_{m,i,j,k} = \int_V f(x, y, z) \cdot \varphi_{m,i,j,k}(x, y, z) dV \quad (2)$$

$m$ : iteration depth.

$i, j, k$ : indices.

In order to compute the projection of the tensor product basis onto the barycentric plane we have to integrate along a line  $t$  which is perpendicular to the barycentric plane. From here, the respective barycentric basis  $\tilde{\varphi}_{m,i,j,k}(x, y, z)$  yields as follows:

$$\tilde{\varphi}_{m,i,j,k}(x, y, z) = \int_t \varphi_{m,i,j,k}(x+t, y+t, z+t) dt \quad (3)$$

It's inner product with any  $L_2$  function defined over the triangular coordinates  $(u, v, w)$  is computed conformingly by

$$c_{m,i,j,k} = \int_{E_b} f(u, v, w) \cdot \tilde{\varphi}_{m,i,j,k}(u, v, w) dE_b \quad (4)$$

This approach can easily be applied to arbitrary wavelets. In section 2.5 we will use B-spline wavelets of order 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.

Furthermore, due to our approach of using a specialized mapping of the coefficients (see Section 3.1) from the barycentric plane to the 3D-space and vice versa, we deduce, that the barycentric basis functions form a basis in the barycentric plane.

As explained in the upcoming section the discrete implementation does not require an explicit mathematical representation of the basis, which can, however, be given in the case of B-splines.

### 2.3 Two-Scale-Relationship

The two-scale-relationship [19] 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:

$$\begin{aligned} \varphi_{m, x_0}(x) \cdot \varphi_{m, y_0}(y) \cdot \varphi_{m, z_0}(z) = & \left( \sum_i a(i - 2x_0) \varphi_{m-1, i}(x) \right) (5) \\ & \cdot \left( \sum_i a(i - 2y_0) \varphi_{m-1, i}(y) \right) \\ & \cdot \left( \sum_i a(i - 2z_0) \varphi_{m-1, i}(z) \right) \end{aligned}$$

$x_0, y_0, z_0$ : offset.

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

Exploiting the line integration property of (3) yields

$$\begin{aligned} \tilde{\varphi}_{m, x_0, y_0, z_0}(x, y, z) = & (6) \\ \int_t \varphi_{m, x_0, y_0}(x+t) \cdot \varphi_{m, y_0}(y+t) \cdot \varphi_{m, z_0}(z+t) dt & \\ = \sum_{i, j, k} a(i - 2x_0) \cdot a(j - 2y_0) \cdot a(k - 2z_0) & \\ \cdot \int_t \varphi_{m-1, i, j, k}(x+t, y+t, z+t) dt & \end{aligned}$$

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

$$\begin{aligned} \tilde{\varphi}_{m, x_0, y_0, z_0}(x, y, z) = & (7) \\ \sum_{i, j, k} \sum_l a(i+l-2x_0) \cdot a(j+l-2y_0) \cdot a(k+l-2z_0) & \\ \cdot \int_t \varphi_{m-1, i+l+x_0, j+l+y_0, k+l+z_0}(x+t, y+t, z+t) dt & \end{aligned}$$

$l$ : 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 upper integral by a barycentric basis in  $(i, j, k)$  thereby establishing the desired relation between two adjacent levels:

$$\tilde{\varphi}_m(x, y, z) = \quad (8)$$

$$\sum_{i, j, k} \left( \sum_l a(i+l-2x_0) \cdot a(j+l-2y_0) \cdot a(k+l-2z_0) \right) \cdot \tilde{\varphi}_{m-1, i, j, k}(x, y, z)$$

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

### 2.4 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  $\langle \cdot, \cdot \rangle$  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. As an example, we compare the basis functions from Fig. 9, which are computed from 3D Haar wavelets. In this case, for instance, the inner product of  $\varphi$  with wavelet  $\psi^1$  cancels to zero,

$$\langle \varphi, \tilde{\psi}^1 \rangle = 0 \Rightarrow \tilde{\varphi} \perp \tilde{\psi}^1 \quad (9)$$

while other combinations do not fulfill the orthogonal property, such as

$$\langle \tilde{\varphi}, \tilde{\psi}^3 \rangle \neq 0 \Rightarrow \tilde{\varphi} \not\perp \tilde{\psi}^3 \quad (10)$$

The examination of linear bases yields similar results, particularly since these functions are semi-orthogonal in 1D.

### 2.5 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  $\varphi$  and wavelets  $\psi$  as shown in Fig. 3, results in piecewise linear  $C^0$ -continuous functions. Fig. 9 depicts the bases functions using an intensity plot. We observe eight different types of bases, four of which are displayed in Fig. 10.

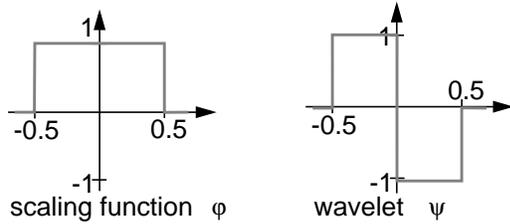


Figure 3: 1D Haar scaling functions and wavelets.

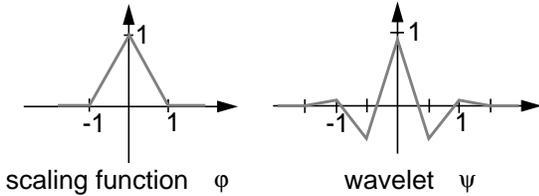


Figure 4: 1D linear B-spline bases according to [7].

### Quartic barycentric B-spline bases

One of the many advantageous properties of B-splines is the automatic continuity control. Therefore, the use of linear tensor product B-spline bases implies  $C^2$  continuity and an increase of the support. Fig. 4 illustrates the 1D versions.

The corresponding barycentric children are presented in Fig. 11. As a fundamental observation, we distinguish 4 different types of basis functions, as given in Fig. 12.

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

## 3 Analysis / Synthesis Algorithms

### 3.1 Barycentric convolution

As mentioned above, the analysis and synthesis algorithms for the barycentric approach are constructed by using the filter sequences  $\{a_i\}$  and  $\{b_i\}$  of the 1D-MRA.

In principle, it is possible to run the respective tensor product algorithm in 3D. Here we recall that the 3D MRA is computed by sequential convolution and subsampling along the 3 principal axes  $x, y, z$ .

Our implementation represents an optimization of the original 3D-approach. In order to illustrate the fundamental differences, let us compare a 2D tensor product MRA to a 1D-barycentric multiresolution analysis which can be thought of as operating along the barycentric line. Fig. 5 illustrates the setting.

The coefficients on the lines orthogonal to the barycentric line are set to equal values (indicated by the same

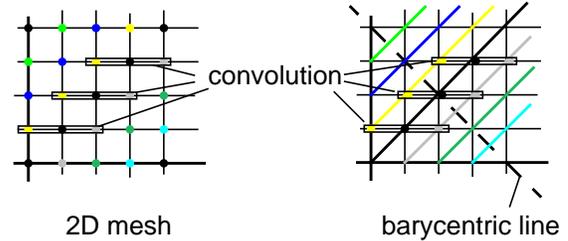


Figure 5: Illustration of the sampling scheme in 2D.

color in Fig. 5).

A thorough investigation of the convolution process of the 2D-MRA reveals that the convolution (i.e. in  $x$ -direction at every point of a unicolored line) operates on the same coefficients and, consequently, computes the same results. That is, the computation along one line must be performed only once.

Another fundamental aspect is the direction of convolution. Again, assuming equal coefficients along the orthogonal lines, a convolution in  $x$ -direction evokes the same result as a direct convolution on the barycentric line, as indicated in Fig. 6. In addition, we observe from Fig. 6 that the barycentric line obviously needs to be resampled at every intersection point of our orthogonal lines with the barycentric line. This conforms to a retriangulation in 3D. Note that the vertices of the triangle mesh do not necessarily lie entirely in the barycentric plane (for example refer to the samples on the yellow line in Fig. 6).

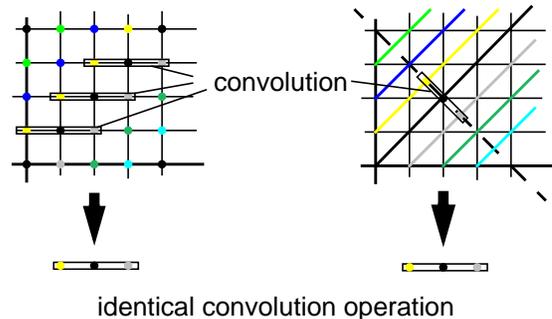


Figure 6: Convolution on the barycentric plane.

The consequences for the overall computational performance of our 3D algorithms can be summarized as follows:

- *Sparsity*: The 3D convolution required to implement the decomposition and reconstruction is sparse; therefore, 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, it is sufficient to design appropriate 2D data structures.

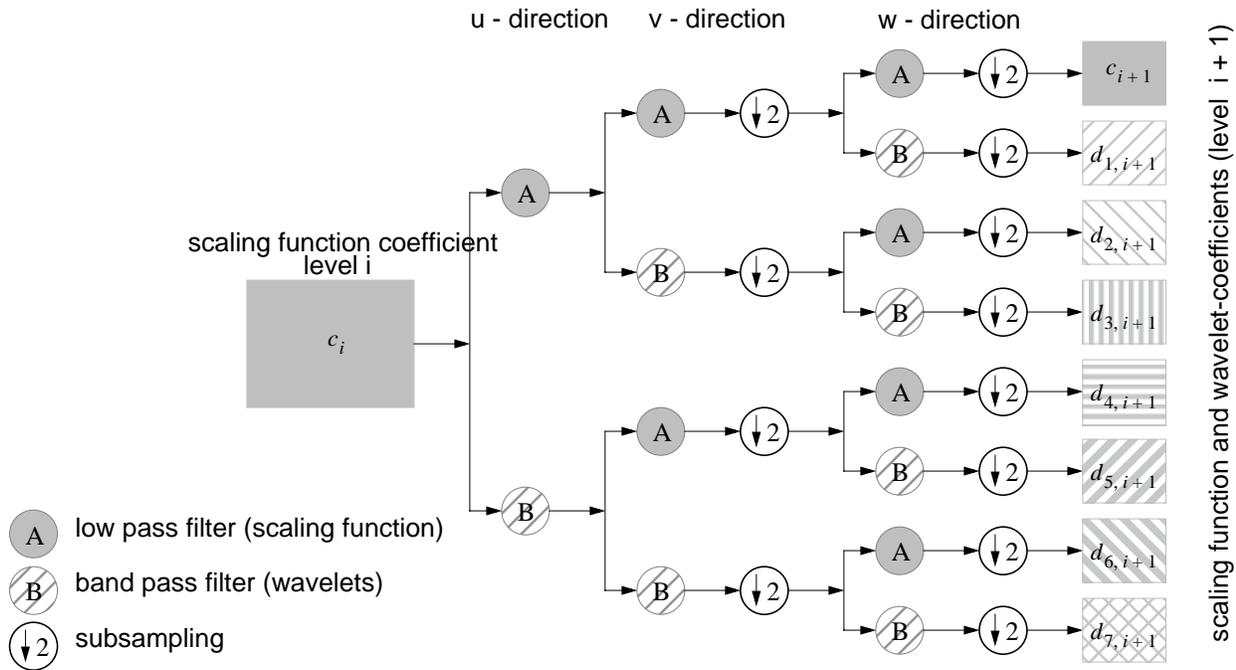


Figure 7: Analysis pipeline of the barycentric MRA.

In this case, a convolution along one of the main axes is expressed by a convolution along one direction of the triangular mesh. Thus, convolutions in  $x$ -,  $y$ - and  $z$ -direction are equivalent to corresponding operations in  $u$ -,  $v$ - and  $w$ -directions on the barycentric plane.

Unlike convolution, the subsampling has to be modified for direct 2D implementations. In this case, the last subsampling step is redundant on the triangular mesh, whereas it is needed on the 3D tensor product grid. To cope with this problem, we restrict subsampling to the last two convolution directions.

### 3.2 Decomposition

The resulting decomposition and analysis scheme for the direct barycentric MRA is shown in Fig. 7. The omission of the subsampling in  $u$ -direction leads to a non-critically sampled pyramid whose estimation is shown in Fig. 8.

The corresponding decomposition algorithm can be outlined in pseudo-code as follows:

```

Initial data:
 $c_0[u,v]$ : 1- or 3-dimensional array of function values
 $a[t]$ ,  $b[t]$ : scaling and wavelet filter kernels for the decomposition

push()
{
  for  $i \leftarrow 0$  to  $N-1$  // MRA-level  $i$ 
    // convolution in  $u$ -direction using filter  $a$  and  $b$ 
    // without subsampling
    for  $u \leftarrow u_{min}$  to  $u_{max}$ 

```

```

      for  $v \leftarrow v_{min}$  to  $v_{max}$ 
        temp1[u,v] = Convolution( $a[t]$ ,  $c_i[u+t,v]$ ,  $t$ )
        temp2[u,v] = Convolution( $b[t]$ ,  $c_i[u+t,v]$ ,  $t$ )
      // convolution in  $v$ -direction using filter  $a$  and  $b$ 
      // including subsampling
      for  $u \leftarrow u_{min}$  to  $u_{max}$  step 2
        for  $v \leftarrow v_{min}$  to  $v_{max}$  step 2
          temp3[u/2,v/2] = Convolution( $a[t]$ , temp1[u,v+t],  $t$ )
          temp4[u/2,v/2] = Convolution( $b[t]$ , temp1[u,v+t],  $t$ )
          temp5[u/2,v/2] = Convolution( $a[t]$ , temp2[u,v+t],  $t$ )
          temp6[u/2,v/2] = Convolution( $b[t]$ , temp2[u,v+t],  $t$ )
        // convolution in  $w$ -direction using filter  $a$  and  $b$ 
        // including subsampling
        for  $u \leftarrow u_{min}$  to  $u_{max}$  step 2
          for  $v \leftarrow v_{min}$  to  $v_{max}$  step 2
             $c_{i+1}[u/2,v/2]$  = Convolution( $a[t]$ ,temp3[u+t,v+t],t)
             $d_{1,i+1}[u/2,v/2]$  = Convolution( $b[t]$ ,temp3[u+t,v+t],t)
             $d_{2,i+1}[u/2,v/2]$  = Convolution( $a[t]$ ,temp4[u+t,v+t],t)
             $d_{3,i+1}[u/2,v/2]$  = Convolution( $b[t]$ ,temp4[u+t,v+t],t)
             $d_{4,i+1}[u/2,v/2]$  = Convolution( $a[t]$ ,temp5[u+t,v+t],t)
             $d_{5,i+1}[u/2,v/2]$  = Convolution( $b[t]$ ,temp5[u+t,v+t],t)
             $d_{6,i+1}[u/2,v/2]$  = Convolution( $a[t]$ ,temp6[u+t,v+t],t)
             $d_{7,i+1}[u/2,v/2]$  = Convolution( $b[t]$ ,temp6[u+t,v+t],t)
          }
    }
}

```

### 3.3 Reconstruction

Analogous to the decomposition, we derive the reconstruction algorithm in pseudo-code:

```

Initial data:
Scaling coefficients (level  $N$ )
 $c_N[u,v]$ 
Wavelet coefficients (level  $1, \dots, N$ )
 $d_{j,i}[u,v]$ ,  $i=1, \dots, N$ ;  $j=1, \dots, 7$ 
 $p[t]$ ,  $q[t]$ : scaling and wavelet filter kernels for the reconstruction
(semi-orthogonal)

```

```

pull()
{
for i ← N-1 to 0 // MRA-level i
// convolution in w-direction using filter p and q
// including upsampling
for u ← umin to umax
for v ← vmin to vmax
temp1[u,v] = Convolution(p[t], ci+1[2u+t,2v+t],t)
+ Convolution(q[t], d1,i+1[2u+t,2vt],t)
temp2[u,v] = Convolution(p[t], d2,i+1[2u+t,2v+t],t)
+ Convolution(q[t], d3,i+1[2u+t,2vt],t)
temp3[u,v] = Convolution(p[t], d4,i+1[2u+t,2v+t],t)
+ Convolution(q[t], d5,i+1[2u+t,2vt],t)
temp4[u,v] = Convolution(p[t], d6,i+1[2u+t,2v+t],t)
+ Convolution(q[t], d7,i+1[2u+t,2vt],t)
// convolution in v-direction using filter p and q
// including upsampling
for u ← umin to umax
for v ← vmin to vmax
temp5[u,v] = Convolution(p[t], temp1[2u,2v+t],t)
+ Convolution(q[t], temp2[2u,2v+t],t)
temp6[u,v] = Convolution(p[t], temp3[2u,2v+t],t)
+ Convolution(q[t], temp4[2u,2v+t],t)
// convolution in u-direction using filter p and q
// without upsampling
for u ← umin to umax
for v ← vmin to vmax
ci[u,v] = Convolution(p[t], temp5[u+t,v],t)
+ Convolution(q[t], temp6[u+t,v],t)
}

```

Note, however, that editing or rejection of individual coefficients in the hierarchy requires some additional post-processing, because modifying a coefficient leads to inconsistencies along the lines orthogonal to the barycentric plane.

In our implementation, we modified the synthesis algorithm so that the last convolution step is performed on an extended mesh. This mesh consists of several parallel planes depending on the support of the bases, and the reconstruction algorithm operates on a sparse 3D slice around the barycentric plane.

### 3.4 Compression and Complexity

One major advantage of the wavelets is their compression performance resulting from the vanishing moments. The rejection of unimportant coefficients is a non-trivial, discrete, global optimization problem in semi-orthogonal settings [14], 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 [22] which employs 4 basis functions, the 3D 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. 8.

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

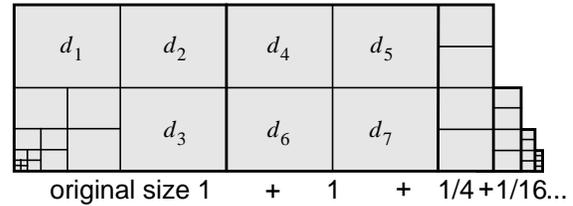


Figure 8: Pyramidal data structure used for the barycentric MRA.

This over-representation requires to investigate possible linear dependencies between individual types of basis functions. An analysis yields a linear dependency for the basis functions for Haar-based constructions.

## 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 heightfields 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. 13, where we lifted one scaling function coefficient for linear and quartic representations.

The surface region affected by an individual scaling function conforms to its spatial support and scales according to (8) 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.

## 5 Results

This section illustrates the usefulness and performance of our 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. 14, we reconstructed a digital terrain model of Matterhorn, Switzerland, 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. 14 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 percentage of error energy. We contrasted linear and quartic bases in the left and right columns, respectively. The difference in smoothness between the  $C^0$  and  $C^2$  continuous surfaces strikes in particular for high compression rates, such as in Fig. 14 d or h.

Multiresolution surface editing is depicted in Fig. 15 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. 15 a, c represent the mesh at level  $m=1$ . Pushing further down the hierarchies as in Fig. 15 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. 15 e and g. 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 out on 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 on some of the fundamental mathematical properties of the basis functions, partly omitted in this paper. In addition, focus will be

given to a reduction of the over-representation and to boundary problems by designing new generations of decomposition and reconstruction operators. 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 Computer Graphics Center, Darmstadt, Germany, 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] 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.
- [11] 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.
- [12] A. Finkelstein and D. H. Salesin. "Multiresolution Curves." *Proceedings of SIGGRAPH '94*, 261-268, July 1994.
- [13] D. Forsey and R. Bartels. "Hierarchical B-spline refinement." *Computer Graphics*, 22(4):205-212, 1988.

- [14] 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>.
- [15] 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.
- [16] L. Lippert, M. Gross, and C. Kurmann. “Compression domain volume rendering for distributed environments.” In *Proceedings of Eurographics '97*, 95-107, 1997.
- [17] C. Loop. “Smooth Spline Surfaces over Irregular Meshes.” *Computer Graphics Proceedings*, 303-310, 1994.
- [18] J. M. Lounsbery. “Multiresolution Analysis for Surfaces of Arbitrary Topological Type.” PhD thesis, University of Washington, Seattle, 1994.
- [19] S. Mallat. “A theory for multiresolution signal decomposition: The wavelet representation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 674-693, 1989.
- [20] 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.
- [21] 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.
- [22] P. Schroeder and W. Sweldens. “Spherical Wavelets: Efficiently Representing Functions on the Sphere.” *ACM SIGGRAPH '95 Proceedings*, 161-172, 1995.
- [23] 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.
- [24] E. Simoncelli and E. Adelson. “Non-Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions.” *Proceedings of the IEEE*, 78(4):652-664, 1990.
- [25] O. G. Staadt, M. H. Gross and R. Weber. “Multiresolution Compression and Reconstruction.” *IEEE Visualization Proceedings '97*, 337-346, 1997.
- [26] E. J. Stollnitz, T. D. DeRose, and D. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, Inc., 1996.
- [27] 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).
- [28] 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 dependency of the wavelets

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

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

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

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

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

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

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

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

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

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

$$\left( \sum_l g(i+l) \cdot g(j+l) \cdot g(k+l) \right)_{\substack{i,j,k \\ i+j+k=0}} \quad (11)$$

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:

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

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

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

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

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

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

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

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

From here we analyze a possible linear dependency by computing the nullspace of the basis Matrix **B** with:

$$\mathbf{B} = \begin{bmatrix} \tilde{\Phi} & \tilde{\Psi}_1 & \tilde{\Psi}_2 & \tilde{\Psi}_3 & \tilde{\Psi}_4 & \tilde{\Psi}_5 & \tilde{\Psi}_6 & \tilde{\Psi}_7 \end{bmatrix} \quad (12)$$

For Haar bases **B** is given by:

$$\mathbf{B} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 2 & 0 & 2 & 2 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \quad (13)$$

Its nullspace can be computed straightforwardly to

$$\text{Nullspace}(\mathbf{B}) = \{[0, 1, 1, 0, 1, 0, 0, 1]\} \quad (14)$$

Obviously, the linear barycentric wavelets constructed from the Haar basis are linear dependent. Note, that we only determined the linear dependency of basis functions in one point of the barycentric plan. Thus, in order to estimate the dimension of linear dependency, all translations of the wavelets and the scaling functions need to be considered. For purpose of brevity, we omitted a detailed elaboration.

## B EXAMPLES

### Haar basis (1D)

The operators for Haar wavelets are trivial and given by

$$a = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} \quad (15)$$

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

$$a = [a_{10}, a_9, a_8, a_7, a_6, \dots, a_1, a_0, a_1, \dots, a_{10}] \quad (16)$$

$$b = [b_9, b_8, b_7, \dots, b_1, b_0, b_1, \dots, b_9] \quad (17)$$

and

$$p = [p_2, p_1, p_2] ; \quad q = [q_3, q_2, q_1, q_2, q_3] \quad (18)$$

with given constants

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

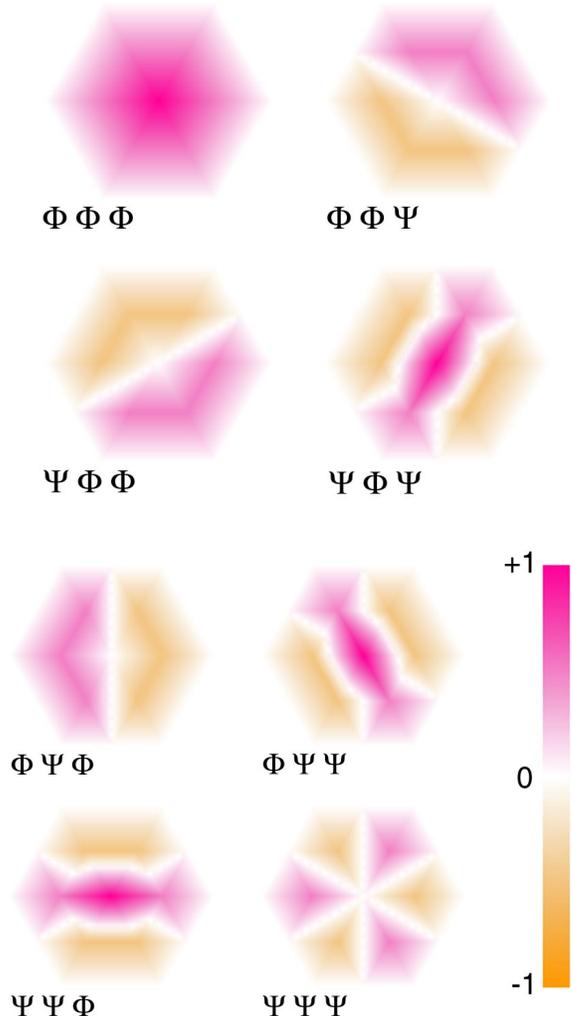


Figure 9: Linear barycentric B-spline bases (intensity plot) ( $C^0$ ).

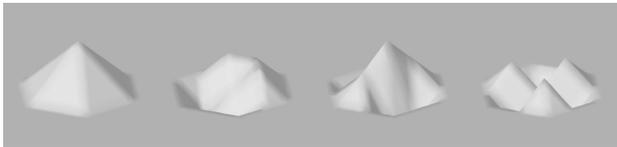


Figure 10: 4 types of linear barycentric B-spline bases.

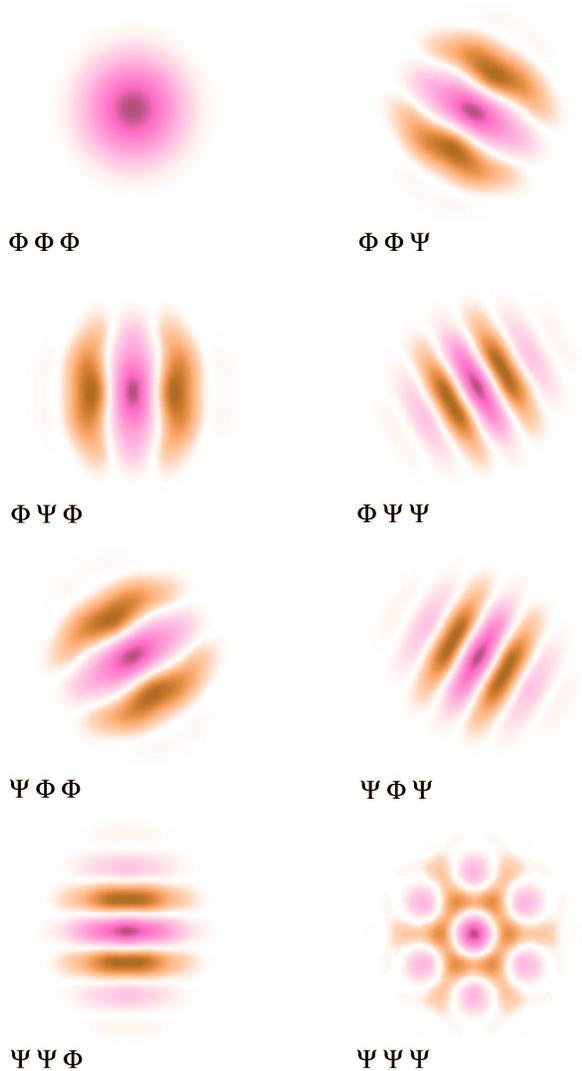


Figure 11: Quartic barycentric B-spline bases (intensity plot) ( $C^2$ ).

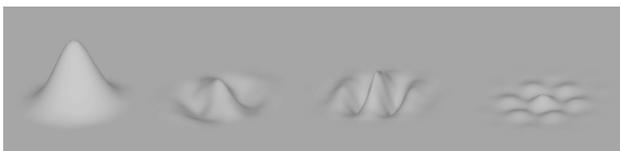


Figure 12: 4 different types of quartic barycentric B-spline bases.

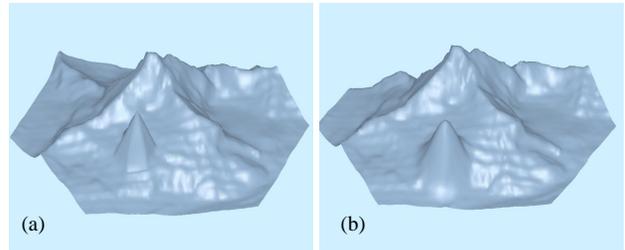


Figure 13: Editing of an individual coefficient in decomposition level  $m=3$  using linear bases (a) and quartic bases (b).

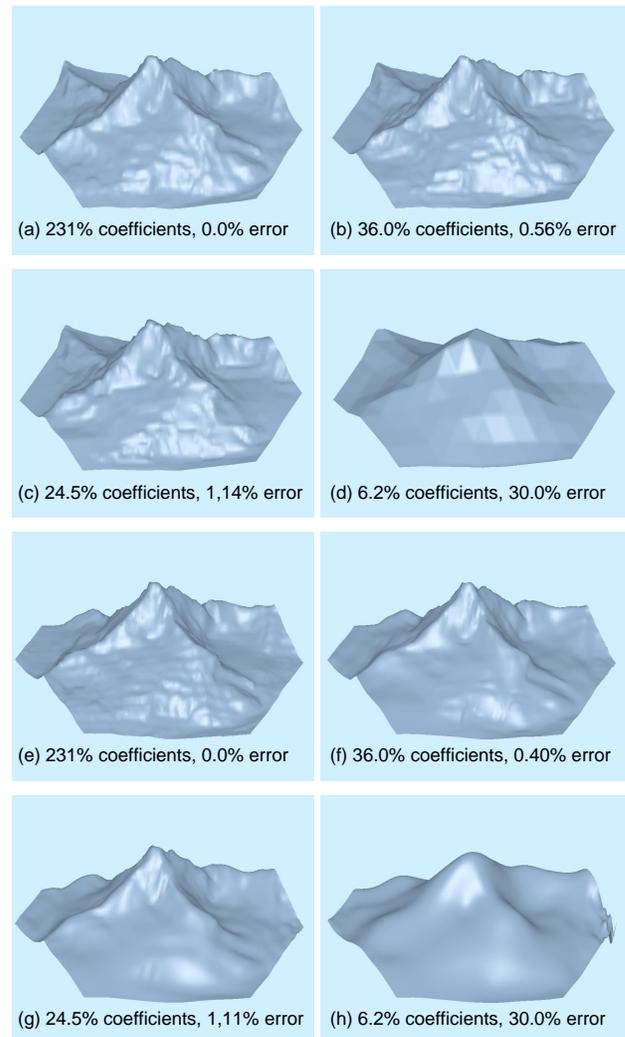
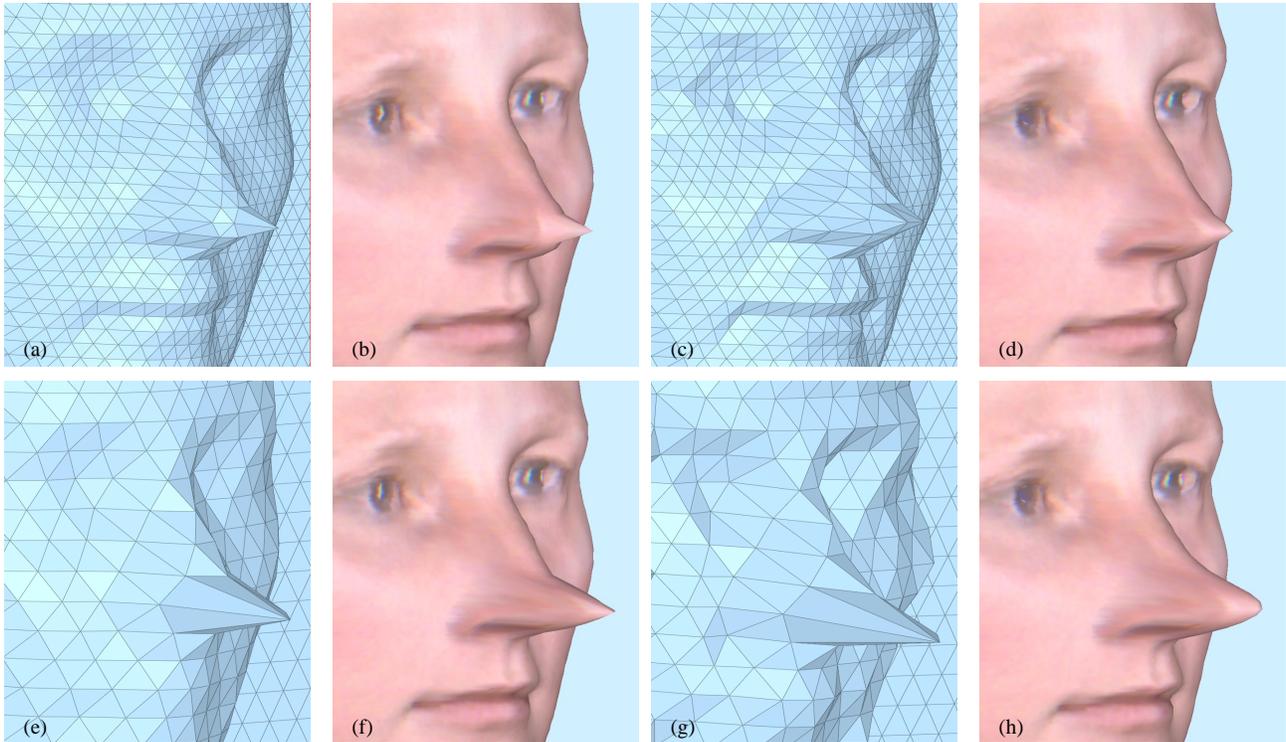


Figure 14: Compression performance of the method on a digital terrain data set: MRA with maximum decomposition level 3, using linear bases ((a)-(d)) and quartic bases ((e)-(h)).



**Figure 15: Editing Silvia's face in different resolutions: level 1 ((a)-(d)) and level 2 ((e)-(h)) using linear bases (a),(b),(e),(f) and quartic bases (c),(d),(g),(h).**