

# Biorthogonal Wavelets for Subdivision Volumes

Martin Bertram<sup>1</sup>

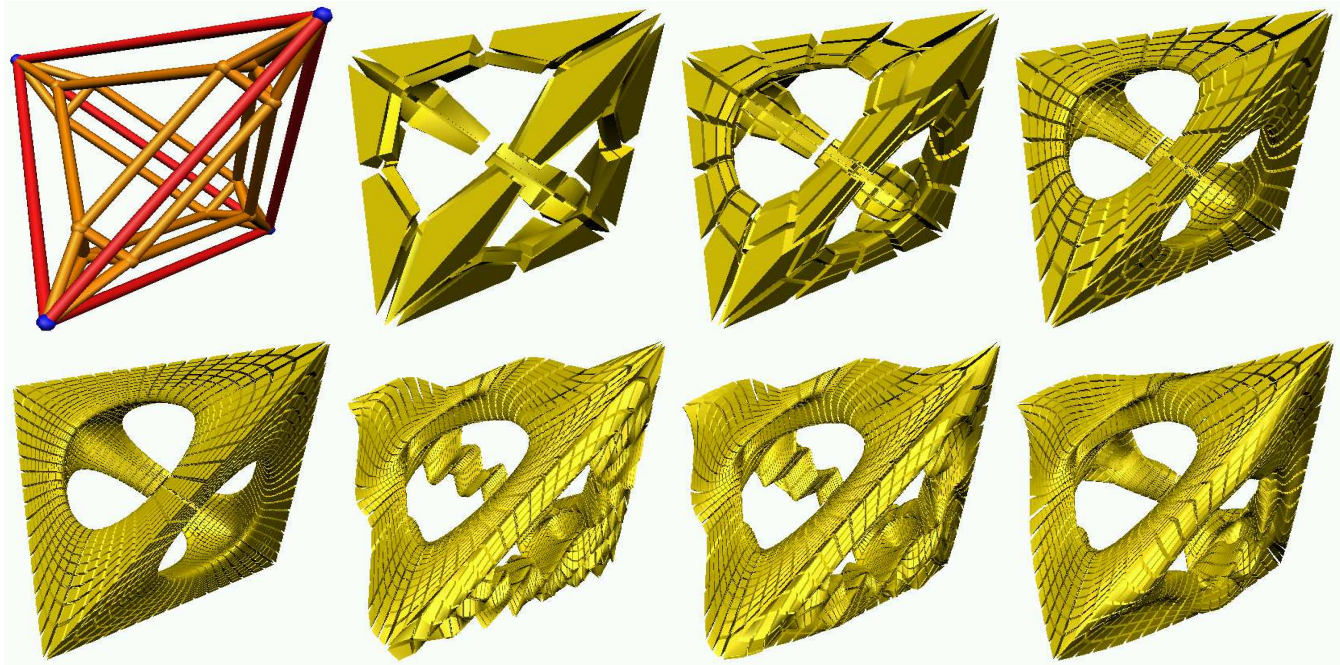


Figure 1: Volume subdivision, manipulation, and fitting. A lattice (top left) is recursively subdivided and re-shaped at the fourth subdivision level. This shape is low-pass filtered by removing fine-resolution wavelet coefficients (bottom right).

## Abstract

We present a biorthogonal wavelet construction based on Catmull-Clark-style subdivision volumes, like multi-linear cell averaging (MLCA). Our wavelet transform is the three-dimensional extension of a previously developed construction of subdivision-surface wavelets that was used for multiresolution modeling of large-scale isosurfaces. Subdivision surfaces provide a flexible modeling tool for geometries of arbitrary topology and for functions defined thereon. Wavelet representations add the ability to compactly represent large-scale geometries at multiple levels of detail. Our wavelet construction based on subdivision volumes extends these concepts to trivariate geometries, such as time-varying surfaces, free-form deformations, and solid models with non-uniform material properties. The domains of the represented trivariate functions are defined by lattices composed of arbitrary polyhedral cells. These are recursively subdivided based on stationary rules converging to piecewise smooth limit-shapes. Sharp features and boundaries, defined by specific polygons, edges, and vertices of a lattice are explicitly represented using modified subdivision rules. Our wavelet transform provides the ability to reverse the subdivision process after a lattice has been re-shaped at a very fine level of detail, for example using an automatic fitting method. During this coarsening process all geometric detail is compactly stored in form of wavelet coefficients from which it can be reconstructed without loss.

<sup>1</sup>University of Kaiserslautern, Department of Computer Science, P.O. Box 3049, D-67653 Kaiserslautern, Germany.

**CR Categories:** E.4 [Coding and Information Theory]: Data Compaction and Compression; F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems—Computation of Transforms; G.1.2 [Numerical Analysis]: Approximation—Approximation of Surfaces and Contours, Wavelets and Fractals

**General Terms:** Algorithms, Performance, Design, Theory.

**Keywords:** Arbitrary Topology, B-Spline Wavelets, Geometry Compression, Hierarchical B-Splines, Multiresolution Modeling, Subdivision Surfaces, Subdivision Volumes.

## 1 Introduction

Subdivision surfaces provide a flexible tool for solid modeling. In contrast to boundary representations (B-rep's) based on multiple trimmed non-uniform rational B-spline (NURBS) surfaces, the boundary of a solid can be represented as a single subdivision surface. This simple and homogeneous representation has many advantages: First, it eliminates the need for cumbersome consistency checks to avoid “dangling” edges and faces, as well as holes in a B-rep. Second, there are no interpolation constraints involved and no trimming is necessary to ensure that incident faces share a unique edge. These edges are often prone to numerical errors in a B-rep, since the boundary curves of incident trimmed NURBS patches do not exactly match in general. Problems of this kind are completely avoided when using subdivision. We note that trimming operations

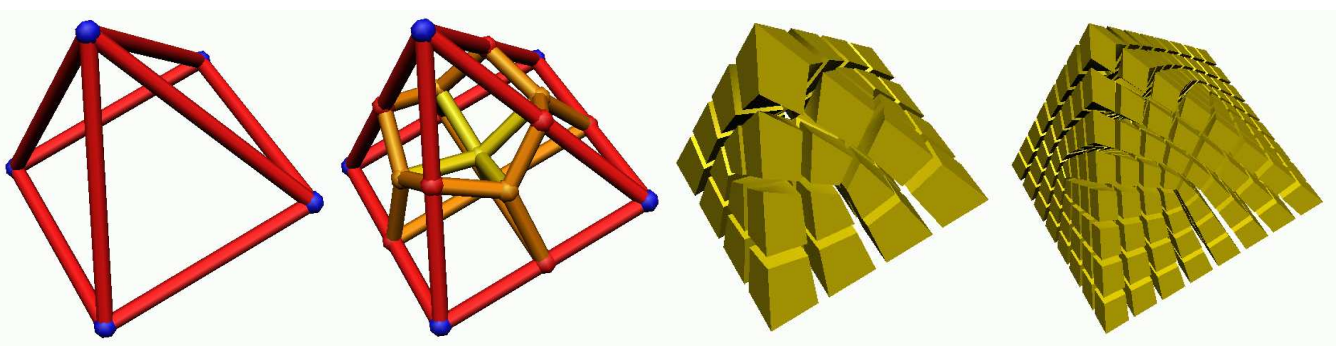


Figure 2: Subdividing a pyramid. Most cells generated by subdivision are hexahedra, except for one row of *type* – 4 cells connecting the top and the center of the pyramid.

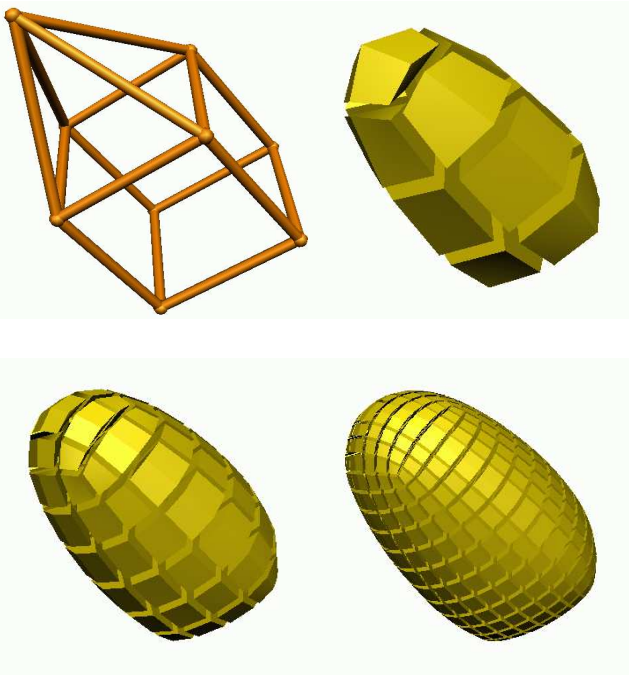


Figure 3: Subdivision volume defined by two cells. The boundary is a single face-feature without sharp edges or vertices.

exist as well for subdivision surfaces [25].

The domain for such a subdivision surface is given by a coarse polyhedral mesh defining a two-manifold topology. This coarse mesh, denoted as *base mesh*, is recursively subdivided by splitting edges and polygons and inserting new vertices. The locations of these vertices, as well as the new locations of the existing vertices, are locally computed by linear subdivision rules. This means that the coordinates of every vertex obtained from a subdivision step depend linearly on the coordinates of a local stencil of vertices prior to this subdivision step. A subdivision scheme is called *stationary*, if the weights of this linear combination are invariant under the level of subdivision and only depend on local mesh connectivity.

The surface defined by a base mesh using certain subdivision rules is the limit surface obtained from the recursive subdivision process. In many cases, this limit surface is composed of piecewise polynomials and can be analytically computed as a function of some local parameters corresponding to a point on the polygonal domain [46, 51]. The property of polynomial precision can

be easily obtained when the subdivision rules are derived, for example, from the knot-insertion procedure for B-splines on regular rectilinear grids. The subdivision rules are then adapted to irregular mesh structures, such as *extraordinary* vertices, i.e. vertices that have a valence other than four in a quadrilateral mesh and a valence other than six in a triangle mesh. (The valence denotes the number of incident edges.) It is well known how to analyze the degree of continuity of subdivision surfaces at extraordinary points using the eigen structures of local subdivision matrices [39, 40, 37, 52, 53].

When modeling solids, it is necessary to consider “sharp” features in a surface, such as creases and cusps, where the degree of smoothness is reduced. These features are modeled using different subdivision rules for vertices located on certain edges of the mesh that define creases. For example, a sequence of consecutive edges in a base mesh can be subdivided such that it converges to a B-spline curve defining a feature within a surface [11, 5, 49]. The smooth patches of a subdivision surface interpolate their boundary curves due to the mesh connectivity without the need of specifying interpolation constraints.

Wavelets complete the concept of subdivision to a powerful multiresolution modeling paradigm that is designated to process large-scale geometries efficiently [28, 13, 4]. Considering the subdivision process only, the entire geometry of a surface would be defined by the (few) control points of its base mesh. The wavelet approach allows to perturb the mesh points after every subdivision step to approximate arbitrary shapes of high geometric detail. The *detail* added to a limit surface is expanded from a sparse set of wavelet coefficients. Due to the adaptive nature of the subdivision process, this geometric detail can be introduced gradually, providing multiple levels of resolution. The amount of detail can be locally adjusted, for example to satisfy a prescribed precision.

The inverse process of *subdividing* and *expanding* detail is implemented by a *fitting* and *compaction* procedure, which is called *wavelet transform*. In the wavelet literature, fitting / compaction is denoted as *decomposition* or *analysis*, whereas subdivision / expansion is known as *reconstruction* or *synthesis*. The wavelet transform starts with a subdivided mesh at fine resolution where the individual vertices correspond to points on a surface (these mesh points can be projected onto a given geometry by some algorithm or they can be manually edited). In reverse order of subdivision, the fitting process computes mesh points at each coarser level defining a sequence of surfaces. Due to the smaller number of control points, geometric detail is removed. This detail is compactly stored as wavelet coefficients at the vertices removed by fitting. For compression purposes, these wavelet coefficients can be quantized and encoded into a minimal bit sequence using, for example, an arithmetic coder [32].

Subdivision volumes, see figures 1–3, extend the subdivision approach to trivariate (three-manifold) domains defined by polyhedral lattices. A base lattice is composed of arbitrary polyhedral

cells sharing common faces (polygons), edges, and vertices. Each face, edge, or vertex can belong to a sharp feature (boundary) of a certain dimensionality. For example, a set of faces can define a smooth surface within the lattice or on its boundary. This surface is uniquely defined by the control points of its defining faces and the limit-shape of the subdivided lattice locally follows the shape of this surface. Edges can belong to a crease, i.e. a sharp curve, or to a surface feature as described above. Multiple surface features can share a sharp curve inside the lattice, thus defining a non-manifold feature. Vertices can be part of a surface feature, a crease, or a cusp (a fixed vertex). All other vertices that are not part of any feature are transformed by the volume subdivision rules.

Considering the range of applications for subdivision wavelets, the third dimension introduces a variety of feasibilities. First, we can represent the interior of a solid composed of different materials in addition to its boundary in a single homogeneous model. Second, we can define highly-detailed freeform deformations in order to automatically re-shape three-dimensional objects. The most interesting application, however, is a wavelet representation of time-varying surfaces. Therefore, the base mesh of a subdivision surface is swept (dragged along line segments), resulting in a lattice composed of prisms. Each layer of this lattice defines the surface at a certain time parameter corresponding to the sweeping distance. Local changes of surface topology can be modeled by splitting or collapsing primitives of the lattice on either side. For representing multiple transitions of surface topology this sweeping process can be repeated multiple times.

The contribution of the present work is a wavelet construction for subdivision volumes with individual synthesis and analysis procedures for features of different dimensionalities. We present a linear-time algorithm implementing our wavelet transform based on simple vertex manipulations. Our algorithm is very memory efficient, since it does not require adjacency information of cells. Instead, it uses a top-down connectivity with respect to the dimensionality of primitives, i.e. every cell is linked to its faces, every face is linked to its edges, and every edge is linked to its two defining vertices. This kind of data structure and our wavelet construction generalize nicely to higher dimensions.

In the next section we summarize related work on subdivision schemes and subdivision-based wavelet constructions. In Section 3, we review the construction of subdivision volumes with features and introduce a set of lifting operations that is used for wavelet construction in section 4. In addition to the technical details of our approach, we provide numerical examples and versatile applications for our method in Section 5.

## 2 Related Work

Subdivision surfaces based on uniform B-splines were initially described by Catmull / Clark [9] and Doo / Sabin [12]. Catmull-Clark subdivision generates bicubic B-splines on a regular grid and is used (with different weights like MLCA [1] in the extraordinary cases) for modeling surface features in our algorithm. Their algorithm inserts vertices inside each face and on each edge resulting in quadrilaterals that are regularly subdivided by the successive steps. Doo-Sabin subdivision generalizes biquadratic B-spline subdivision to arbitrary base meshes and uses the dual mesh structure of the meshes generated by Catmull-Clark subdivision. Both schemes provide  $C^1$ -continuous limit surfaces in the extraordinary case.

Loop [27] developed a subdivision scheme based on box splines for triangle meshes by inserting vertices only on the edges of a mesh, thus splitting each triangle into four. Loop's surfaces are  $C^1$ -continuous and approximate  $C^2$ -continuity at extraordinary points. This scheme can be improved satisfying  $G^2$ -continuity everywhere [38]. An interpolating scheme known as *butterfly subdivision*

[54, 14] uses the same mesh structure. A less "aggressive" scheme known as  $\sqrt{3}$ -subdivision [18] only triples the number of triangles at every subdivision step. An edge-splitting approach for polyhedral meshes is due to Peters / Reif [36]. Subdivision schemes generalizing uniform B-Splines of arbitrary order were recently constructed [45, 50].

Continuity degrees for different classes of subdivision surface schemes were analyzed by Prautzsch [39], Peters / Reif [37] Zorin [53], and by Levin [23]. The idea of using the eigen structure of local subdivision matrices for the limit analysis at extraordinary points goes back to Doo / Sabin [12]. This kind of limit analysis for volume subdivision has not been investigated, so far. The reason for this is a much broader range of irregular constellations in the volume case. Excluding the extraordinary points, the continuity degrees of schemes derived from B-spline subdivision are known, since the limit shape is locally a B-spline curve, surface, or volume. For our construction based on cubic B-spline subdivision,  $C^2$ -continuity is obtained almost everywhere, except for extraordinary points.

Subdivision rules for sharp features within piecewise smooth limit surfaces were introduced by Hoppe et al. [17]. Features can easily be incorporated into every subdivision scheme [11, 5, 6, 49]. Interpolating subdivision for quadrilateral meshes is due to Kobbelt [20]. Interpolation can be directly built into subdivision schemes or enforced by some kind of fitting method [16, 22, 33]. Algorithms for fitting subdivision surfaces to some prescribed geometry are well known [19, 3, 26] and can be used in combination with our technique.

A very useful approach for re-parametrization of triangle meshes is known as *multiresolution adaptive parametrization of surfaces* (MAPS) [21]. This algorithm converts an irregular triangle mesh into a pseudo-regular mesh with subdivision connectivity, thus providing the fine-resolution input for a wavelet transform. An interesting algorithm for re-shaping subdivision volumes based on physical principles (Lagrangian equation of motion) is presented by McDonnell / Qin [31]. A subdivided lattice, deformed by any of these methods, can be compressed and reconstructed using our wavelet transform.

Many subdivision algorithms use the Catmull-Clark mesh structure due to its regular refinement of quadrilateral patches. In addition, this structure nicely generalizes to volume subdivision, where for many types of base polyhedra only hexahedral cells (cubes) are issued by the first subdivision step. The first approach to volume subdivision by MacCracken / Joy [30] extends Catmull-Clark subdivision to three-dimensional lattices. Their subdivision approach is used to define free-form deformations in computer animation. It is observed that the number of vertices grows by a factor of eight for volume subdivision. A less aggressive scheme by Pascucci [35] splits every subdivision step into three smaller ones, reducing the growth to a factor of two. Wavelets for this style of subdivision were used for data compression based on regular lattices [24].

Another variant of volume subdivision applying trilinear subdivision followed by a second vertex-modification step (multi-linear cell averaging, MLCA) is due to Bajaj et al. [1]. Though, we developed the subdivision scheme for our wavelet construction independently of their work, both schemes are nearly identical. Hence, we provide a wavelet construction for MLCA subdivision.

There exists no straight-forward volume construction generalizing triangle-based schemes, like Loop subdivision. The reason for this is that tetrahedra cannot be split into smaller primitives of congruent shape, like triangles, rectangles, and hexahedra. A tetrahedron can be split into four tetrahedra and an octahedron, which can be split into tetrahedra, again. This way, a pseudo-regular subdivision hierarchy with few different cell types is constructed. There still remains the problem of constructing a (polynomial) subdivision scheme for such a pseudo-regular mesh hierarchy.

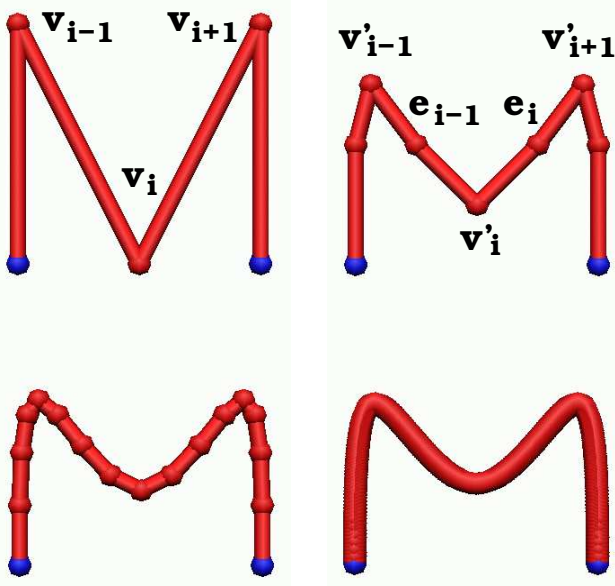


Figure 4: Recursive subdivision by knot-insertion for a cubic B-spline curve.

Biorthogonal wavelet constructions for many different subdivision surface schemes were introduced by Lounsbery et al. [28, 29]. Further subdivision-surface wavelet constructions for functions defined on triangulated spherical domains are described by Schröder / Sweldens [43], Nielson et al. [34], and Bonneau [8, 7]. Their approaches can be extended to more general domains, but they are mainly used for constructing functions on given geometries rather than representing these geometries.

Considering wavelet constructions for B-splines, there exist a variety of different constructions. Semi-orthogonal approaches [10, 15, 47] transform curves and surfaces into a hierarchy of least-squares approximations. Least-squares fitting requires the solution of a global system of equations in the case of non-orthogonal scaling functions (B-Splines are non-orthogonal for any order greater than one). In the case of surfaces defined by regular, rectilinear control meshes, the corresponding matrices are banded such that the transform has linear-time complexity. In the case of arbitrary meshes, these matrices are sparse, but not generally banded, implying a less efficient inversion.

Our approach is a biorthogonal wavelet construction where the transform and its inverse are computed by local filtering operations, providing linear time complexity and the advantage of local computation for out-of-core algorithms. The coarse approximations are consequently not least-squares approximations of the finest level. It is possible, however, to obtain a local least-squares fit when using biorthogonal constructions [29, 42]. The fitting is performed either with respect to the geometric limit shape [28, 29] or with respect to the control polygon minimizing a discrete norm [13, 42, 41].

In a previous paper we introduced a wavelet construction for Catmull-Clark-style subdivision for large-scale isosurface representation [3, 4]. This wavelet construction is extended to three-dimensional lattices in the present work. The idea of this construction is to compute the transform by a small number of simple vertex updates (*lifting* operations [48]) in order to obtain highest efficiency and locality for processing large meshes. The time- and memory requirements for our algorithm depend linearly on the number of transformed coefficients / control points.

### 3 Subdivision Volumes

The subdivision scheme used for our wavelet approach is identical to multi-linear cell averaging (MCLA) by Bajaj *et al.* [1], except for the treatment of type- $n$  cells. In this section, we describe the construction of subdivision curves, surfaces, and volumes including features [17, 11, 6, 49] and introduce the simple building blocks (lifting operations) of our wavelet transform. We start with our construction in one dimension and then extend this approach to regular meshes and lattices as a tensor product. The individual vertex updates implementing the subdivision process are then adapted to the irregular settings, e.g. to extraordinary vertices.

#### 3.1 Subdivision Curves

Considering a cubic B-spline curve defined by control points  $\mathbf{v}_i$  and a uniform knot sequence, a subdivision scheme can be derived by inserting a new knot in the center of every knot interval. Recursive subdivision will create a sequence of control polygons converging rapidly to the B-spline curve, see figure 4. Every subdivision step is computed by inserting a new vertex  $\mathbf{e}_i$  on every edge  $\mathbf{v}_i\mathbf{v}_{i+1}$  and by modifying the coordinates of the existing vertices. This subdivision scheme is computed as follows:

$$\begin{aligned} \mathbf{e}_i &= \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_{i+1}); \\ \mathbf{v}'_i &= \frac{1}{2}\mathbf{v}_i + \frac{1}{4}(\mathbf{e}_{i-1} + \mathbf{e}_i). \end{aligned} \quad (3.1)$$

The boundary vertices are not modified by this subdivision process. Vertex features (cusps) inside the curve can be modeled analogously. For the subsequent subdivision step, the vertices of the new control polygon are re-named to  $\mathbf{v}_i$ .

We now define the basic building blocks of our wavelet transform and re-write the subdivision process using the following three operations:

$$\mathbf{w}\text{-lift}(a): \quad \mathbf{e}_i \leftarrow \mathbf{e}_i + a(\mathbf{v}_i + \mathbf{v}_{i+1}), \quad (3.2)$$

$$\mathbf{s}\text{-lift}(a): \quad \mathbf{v}_i \leftarrow \mathbf{v}_i + a(\mathbf{e}_{i-1} + \mathbf{e}_i), \quad (3.3)$$

$$\mathbf{s}\text{-scale}(a): \quad \mathbf{v}_i \leftarrow a\mathbf{v}_i. \quad (3.4)$$

In the context of a wavelet transform, the  $\mathbf{e}$ -vertices already exist prior to a subdivision step and are associated with wavelet coefficients carrying geometric detail. Hence, we call the manipulation of  $\mathbf{e}$ -vertices *w-lift*, inspired by Sweldens' *lifting scheme* [48]. The  $\mathbf{v}$ -vertices representing a data set at a certain level of resolution are the coefficients of so-called *scaling functions* in the wavelet context. Hence, we call the  $\mathbf{v}$ -manipulating operations *s-lift* and *s-scale*.

For the subdivision process, our wavelet coefficients  $\mathbf{e}_i$  are zero prior to a subdivision step. The cubic subdivision procedure from equation(3.1) can now be written as a sequence of lifting operations:

$$\begin{aligned} &\mathbf{w} - \text{lift}\left(\frac{1}{2}\right); \\ &\mathbf{s} - \text{lift}\left(\frac{1}{2}\right); \\ &\mathbf{s} - \text{scale}\left(\frac{1}{2}\right). \end{aligned} \quad (3.5)$$

This procedure remains valid for subdividing meshes and lattices. For this purpose, however, we need to re-define the individual lifting operations. Analogously to the above construction, we can formulate subdivision rules for B-Splines of other polynomial degrees.

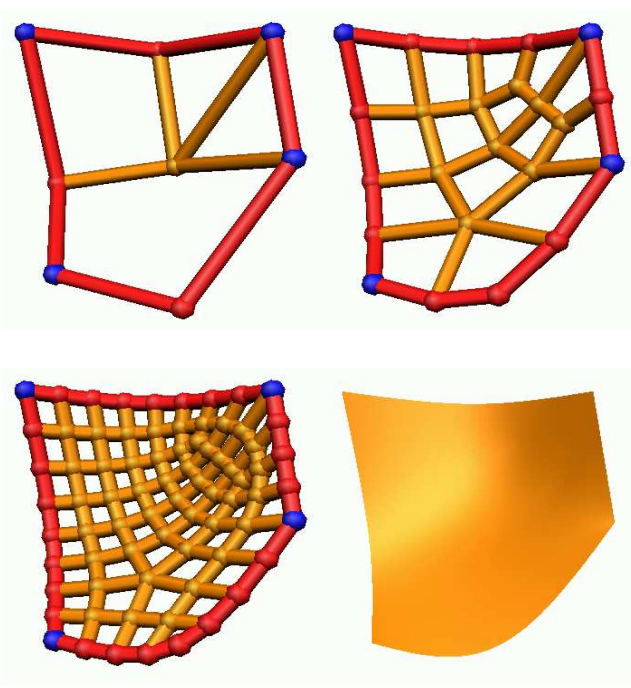


Figure 5: Subdividing a polyhedral mesh by inserting a new vertex for each face and each edge. Blue vertices denote vertex features. Red edges and vertices correspond to curve features and orange primitives denote face features.

The simpler linear subdivision, for example, is defined by

$$\mathbf{w} = \text{lift}\left(\frac{1}{2}\right); \quad (3.6)$$

Similar constructions exist for higher polynomial degrees (for the quintic case see [4]).

### 3.2 Subdivision Surfaces

A polyhedral mesh is defined by a set of vertices, a set of edges, and a set of faces. Every edge has two incident vertices and every face corresponds to a cycle of edges. Catmull-Clark-style subdivision inserts a new vertex for every face and for every edge and connects these to define smaller quadrilateral faces, see figure 5. We denote the sets of vertices corresponding to faces, edges, and (old) vertices by  $F$ ,  $E$ , and  $V$ , respectively. For every  $\mathbf{f} \in F$ ,  $\mathbf{e} \in E$ , and  $\mathbf{v} \in V$ , we use the following neighborhood definitions:

- $NF(\mathbf{v})$ : set of  $\mathbf{f}$ -vertices corresponding to incident faces of  $\mathbf{v}$ .
- $NE(\mathbf{v})$ : set of  $\mathbf{e}$ -vertices corresponding to incident edges of  $\mathbf{v}$ .
- $NF(\mathbf{e})$ : set of  $\mathbf{f}$ -vertices corresponding to faces sharing the edge associated with  $\mathbf{e}$ .
- $NV(\mathbf{e})$ : set of two incident  $\mathbf{v}$ -vertices of the edge.
- $NE(\mathbf{f})$ : set of  $\mathbf{e}$ -vertices corresponding to edges that define the face associated with  $\mathbf{f}$ .
- $NV(\mathbf{f})$ : set of  $\mathbf{v}$ -vertices defining the face.

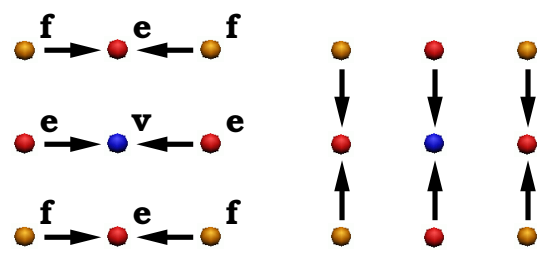


Figure 6: Tensor product computation of an s-lift operation.

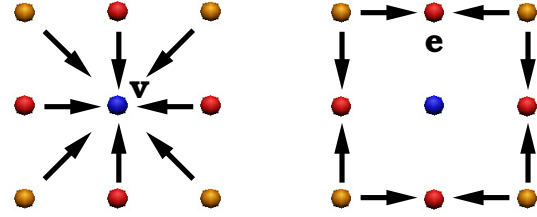


Figure 7: Different order of computation for s-lift operation.

Considering a regular, rectilinear mesh, a subdivision surface is defined by the tensor products of the one-dimensional operations. In this case, the one-dimensional lifting steps are applied to the rows and columns (or vice versa), as illustrated in figure 6. In the case of an s-lift, we can modify the order of computation such that we first update all  $\mathbf{v}$ -vertices and then all  $\mathbf{e}$ -vertices, according to figure 7. To obtain the same computation result, we must consider that the contribution of a  $\mathbf{f}$ -vertex to a neighboring  $\mathbf{v}$ -vertex traverses exactly one  $\mathbf{e}$ -vertex and is multiplied twice with the lifting parameter  $a$ . Thus, we re-define the s-lift operation for regular meshes:

regular 2d **s-lift**( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow \mathbf{v} + a^2 \sum_{\mathbf{f} \in NF(\mathbf{v})} \mathbf{f} + a \sum_{\mathbf{e} \in NE(\mathbf{v})} \mathbf{e}; \\ \mathbf{e} &\leftarrow \mathbf{e} + a \sum_{\mathbf{f} \in NF(\mathbf{e})} \mathbf{f}. \end{aligned} \quad (3.7)$$

In the case of an extraordinary  $\mathbf{v}$ -vertex, we want to normalize the contribution of neighboring  $\mathbf{e}$ - and  $\mathbf{f}$ -vertices by the valence. Therefore, we define the centroids

$$\begin{aligned} \bar{\mathbf{e}}_{\mathbf{v}} &:= \frac{1}{|NE(\mathbf{v})|} \sum_{\mathbf{e} \in NE(\mathbf{v})} \mathbf{e}; \\ \bar{\mathbf{f}}_{\mathbf{v}} &:= \frac{1}{|NF(\mathbf{v})|} \sum_{\mathbf{f} \in NF(\mathbf{v})} \mathbf{f}; \\ \bar{\mathbf{f}}_{\mathbf{e}} &:= \frac{1}{|NF(\mathbf{e})|} \sum_{\mathbf{f} \in NF(\mathbf{e})} \mathbf{f}. \end{aligned} \quad (3.8)$$

Using these centroids instead of summations, equation (3.7) can be written as

2d **s-lift**( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow \mathbf{v} + 4a^2 \bar{\mathbf{f}}_{\mathbf{v}} + 4a \bar{\mathbf{e}}_{\mathbf{v}}; \\ \mathbf{e} &\leftarrow \mathbf{e} + 2a \bar{\mathbf{f}}_{\mathbf{e}}. \end{aligned} \quad (3.9)$$

Analogously to the s-lift, we define generalized two-dimensional w-lift and s-scale operations. For a w-lift the dual mesh is considered, i.e., the  $\mathbf{v}$ - and  $\mathbf{f}$ -vertices simply change their roles. The

centroids  $\bar{\mathbf{e}}_f$ ,  $\bar{\mathbf{v}}_f$ , and  $\bar{\mathbf{v}}_e$  are defined analogously to equation (3.8). For an s-scale operation, the e-vertices are multiplied once with the parameter  $a$ , and the v-vertices are multiplied twice:

2d w-lift( $a$ ):

$$\begin{aligned} \mathbf{f} &\leftarrow \mathbf{f} + 4a^2\bar{\mathbf{v}}_f + 4a\bar{\mathbf{e}}_f; \\ \mathbf{e} &\leftarrow \mathbf{e} + 2a\bar{\mathbf{v}}_e. \end{aligned} \quad (3.10)$$

2d s-scale( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow a^2\mathbf{v}; \\ \mathbf{e} &\leftarrow a\mathbf{e}. \end{aligned} \quad (3.11)$$

### 3.3 Modeling Features

Feature lines within a subdivision surface are obtained by using the one-dimensional lifting operations for vertices corresponding to certain edges in a base mesh. To represent features of different kind, we define a *feature-dimensionality* for every base-mesh primitive. During the subdivision process, the newly generated faces, edges, and vertices inherit the feature-dimensionality from the corresponding primitives of the prior subdivision step. A vertex, for example, can be associated with these feature-dimensionalitys:

$$\dim(\mathbf{v}) = \begin{cases} 0, & \text{if } \mathbf{v} \text{ represents a vertex - feature;} \\ 1, & \text{if } \mathbf{v} \text{ is part of an edge - feature;} \\ 2, & \text{if } \mathbf{v} \text{ is part of a face - feature;} \\ 3, & \text{otherwise.} \end{cases} \quad (3.12)$$

The dimensionality of every edge (and correspondingly of an e-vertex) is at least one and the dimensionality of every face is at least two. We assume that for every face, the feature-dimensionalitys of its defining edges and vertices is less or equal the face's dimensionality. Analogously, the feature-dimensionality of a vertex is typically less or equal than the dimensionality of its incident edges.

It can be observed that in the case of a w-lift the manipulated vertices have greater dimensionalitys than the manipulating ones. Thus, the w-lift operation has no impact on the shape of features. For s-lift and s-scale operations, however, the vertex manipulation is directed from higher dimensionalitys to lower ones. In the case of an edge-feature, we apply the one-dimensional lifting operations to the vertices defining the feature and reject any contribution from higher-dimensional primitives. In the case of a vertex-feature, all subdivision rules are ignored.

The resulting subdivision rules are defined as follows:

2d w-lift( $a$ ):

$$\begin{aligned} \mathbf{f} &\leftarrow \mathbf{f} + 4a^2\bar{\mathbf{v}}_f + 4a\bar{\mathbf{e}}_f; \\ \mathbf{e} &\leftarrow \mathbf{e} + 2a\bar{\mathbf{v}}_e; \end{aligned} \quad (3.13)$$

2d s-lift( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow \mathbf{v} + \begin{cases} 4a^2\bar{\mathbf{f}}_v + 4a\bar{\mathbf{e}}_v & \text{if } \dim(\mathbf{v}) = 2; \\ 2a\bar{\mathbf{e}}_v^{(1)} & \text{if } \dim(\mathbf{v}) = 1; \\ 0 & \text{otherwise;} \end{cases} \\ \mathbf{e} &\leftarrow \mathbf{e} + \begin{cases} 2a\bar{\mathbf{f}}_e & \text{if } \dim(\mathbf{e}) = 2; \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (3.14)$$

2d s-scale( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow a^{\dim(\mathbf{v})}\mathbf{v}; \\ \mathbf{e} &\leftarrow a\mathbf{e} \quad \text{if } \dim(\mathbf{e}) = 2. \end{aligned} \quad (3.15)$$

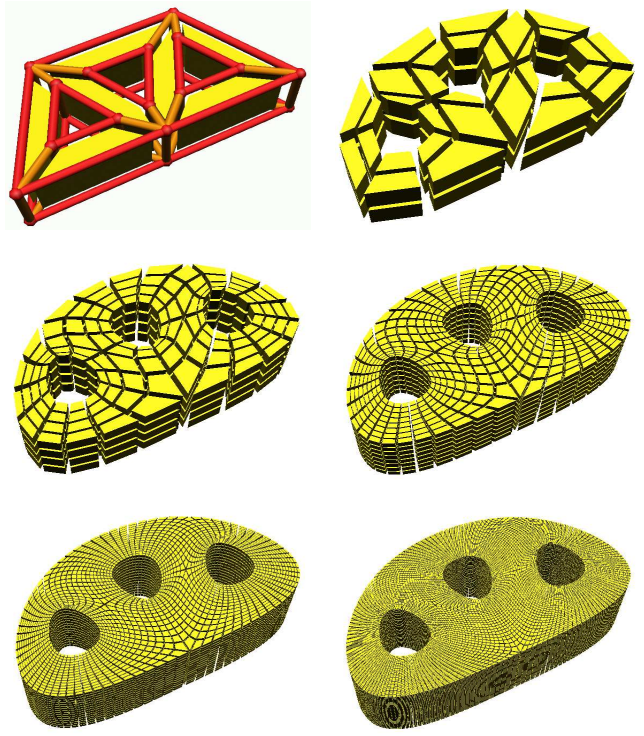


Figure 8: A subdivision volume defined by sweeping a polyhedral mesh (duplicating the mesh and connecting corresponding vertices).

The term  $\bar{\mathbf{e}}_v^{(1)}$  denotes the centroid taken from only those neighboring e-vertices that have a feature-dimensionality of one. Inserting these definitions of the individual lifting operations in equation (3.5) provides the generalized bicubic subdivision rules including features for arbitrary base meshes. These meshes are not limited to manifold topologies. For example, more than two faces can share an edge-feature and more than two edges can share an incident vertex-feature.

### 3.4 Subdivision Volumes

In the trivariate case we consider base-lattices composed of polyhedral cells. The subdivision process inserts a new c-vertex for every cell, an f-vertex for every face, and an e-vertex for every edge, see figure 2. The first subdivision step issues only quadrilateral faces and mostly hexahedral cells. In some cases, however, so-called *type-n* cells [30] composed of  $2n$  faces and  $2n+2$  vertices ( $n > 3$ ) are generated.

A hexahedron is a type-3 cell. Type- $n$  cells with  $n > 3$  are created when the valence of a v-vertex is  $n > 3$  considering only edges belonging to a certain cell. An example for a type-4 cell is the topmost cell in a subdivided pyramid, see Figure 2. When subdividing a type- $n$  cell, two type- $n$  cells and  $2n$  hexahedra are generated. In the limit, an entire curve is defined by a sequence of type- $n$  cells, see Figure 9. Type- $n$  cells with  $n > 3$  can be completely avoided by using only certain cell types in a base mesh. All v-vertices of these cells must have valence three, which is the case for hexahedra, prisms, and tetrahedra.

For generalizing our lifting operations to volume subdivision, we first consider the regular, hexahedral case. Here, the individual lifting steps are tensor-products of one-dimensional operations applied to the three canonical grid directions. In the case of an s-lift, for ex-

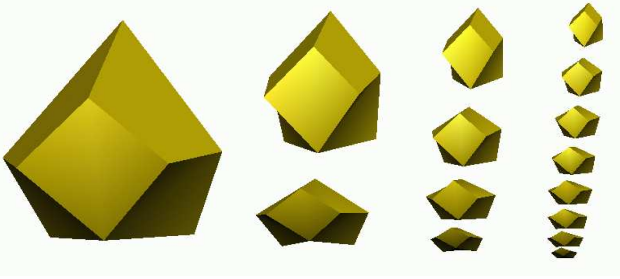


Figure 9: Recursive subdivision of a type-4 cell. A type- $n$  cell contains two vertices of valence  $n$ . Hence, subdivision produces two smaller type- $n$  cells (and  $2n$  hexahedra). In the limit, the row of type- $n$  cells converges to a curve inside the volume.

ample, a  $\mathbf{v}$ -vertex has 8 cell-neighbors, 12 face-neighbors, and 6 edge-neighbors. Every cell-neighbor contributes with a weight of  $a^3$  (traversing an  $\mathbf{f}$ -vertex and an  $\mathbf{e}$ -vertex), every face neighbor contributes with weight  $a^2$  (traversing an  $\mathbf{e}$ -vertex), and every  $\mathbf{e}$ -vertex contributes directly to  $\mathbf{v}$  with weight  $a$ . In addition to  $\mathbf{v}$ -vertices, the  $\mathbf{e}$ - and  $\mathbf{f}$ -vertices are also modified by an s-lift operation. Similar considerations as for the bivariate case provide the generalized trivariate subdivision rules.

Considering features of different dimensionality, the lifting operations for volume subdivision are defined as

3d **w-lift**( $a$ ):

$$\begin{aligned} \mathbf{c} &\leftarrow \mathbf{c} + 8a^3\bar{\mathbf{v}}_{\mathbf{c}} + 12a^2\bar{\mathbf{e}}_{\mathbf{c}} + 6a\bar{\mathbf{f}}_{\mathbf{c}} \\ \mathbf{f} &\leftarrow \mathbf{f} + 4a^2\bar{\mathbf{v}}_{\mathbf{f}} + 4a\bar{\mathbf{e}}_{\mathbf{f}}; \\ \mathbf{e} &\leftarrow \mathbf{e} + 2a\bar{\mathbf{v}}_{\mathbf{e}}; \end{aligned} \quad (3.16)$$

3d **s-lift**( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow \mathbf{v} + \begin{cases} 8a^3\bar{\mathbf{c}}_{\mathbf{v}} + 12a^2\bar{\mathbf{f}}_{\mathbf{v}} + 6a\bar{\mathbf{e}}_{\mathbf{v}} & \text{if } \dim(\mathbf{v}) = 3; \\ 4a^2\bar{\mathbf{f}}_{\mathbf{v}}^{(2)} + 4a\bar{\mathbf{e}}_{\mathbf{v}}^{(2)} & \text{if } \dim(\mathbf{v}) = 2; \\ 2a\bar{\mathbf{e}}_{\mathbf{v}}^{(1)} & \text{if } \dim(\mathbf{v}) = 1; \\ 0 & \text{otherwise;} \end{cases} \\ \mathbf{e} &\leftarrow \mathbf{e} + \begin{cases} 4a^2\bar{\mathbf{c}}_{\mathbf{e}} + 4a\bar{\mathbf{f}}_{\mathbf{e}} & \text{if } \dim(\mathbf{e}) = 3; \\ 2a\bar{\mathbf{f}}_{\mathbf{e}}^{(2)} & \text{if } \dim(\mathbf{e}) = 2; \\ 0 & \text{otherwise.} \end{cases} \\ \mathbf{f} &\leftarrow \mathbf{f} + \begin{cases} 2a\bar{\mathbf{c}}_{\mathbf{f}} & \text{if } \dim(\mathbf{f}) = 3; \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (3.17)$$

3d **s-scale**( $a$ ):

$$\begin{aligned} \mathbf{v} &\leftarrow a^{\dim(\mathbf{v})}\mathbf{v}; \\ \mathbf{e} &\leftarrow a^{\dim(\mathbf{e})-1}\mathbf{e} \quad \text{if } \dim(\mathbf{e}) \geq 2; \\ \mathbf{f} &\leftarrow a\mathbf{f} \quad \text{if } \dim(\mathbf{f}) = 3. \end{aligned} \quad (3.18)$$

The centroids  $\bar{\mathbf{f}}_{\mathbf{v}}^{(2)}$ ,  $\bar{\mathbf{e}}_{\mathbf{v}}^{(2)}$ , and  $\bar{\mathbf{f}}_{\mathbf{e}}^{(2)}$  are computed from vertices of feature-dimensionality two, only. The neighborhood definitions are extended to include  $\mathbf{c}$ -vertices. The set of subdivision rules is now completely defined. Examples for this subdivision process are illustrated in figures 1 and 8.

### 3.5 Continuity at Extraordinary Points

Due to the wide range of special irregular cases (even when considering only type-3 cells) in volume-subdivision structures, it is hard

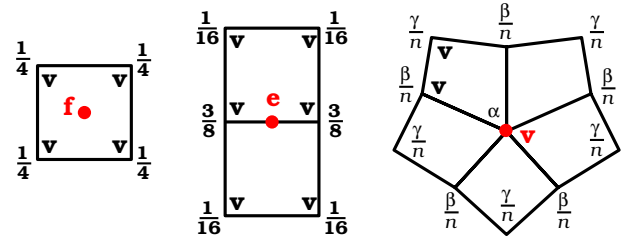


Figure 10: Subdivision masks for surfaces generalizing bicubic B-splines. Different choices for  $\alpha$ ,  $\beta$ , and  $\gamma$  are feasible.

to predict the degree of continuity of the limit shape at extraordinary points. In all ordinary points, our subdivision scheme produces  $C^2$ -continuous shapes (except across features), since it is derived from cubic B-splines.

Considering surface features, our subdivision is  $C^1$  at extraordinary points which can be verified by analyzing the eigenstructure of a local subdivision matrix [37]. Our bicubic scheme uses three lifting operations computing the control points  $\mathbf{f}$ ,  $\mathbf{e}$ , and  $\mathbf{v}'$  from coarse-level points  $\mathbf{v}$ ,

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{e} \\ \mathbf{v}' \end{pmatrix} = \mathbf{S}_3 \mathbf{S}_2 \mathbf{S}_1 \mathbf{v}, \quad (3.19)$$

where the matrices  $\mathbf{S}_i$  correspond to the individual lifting steps in equation 3.5. Combining these operations into one matrix  $\mathbf{S} = \mathbf{S}_3 \mathbf{S}_2 \mathbf{S}_1$  provides the subdivision masks for  $\mathbf{f}$ ,  $\mathbf{e}$ , and  $\mathbf{v}'$ -vertices, as shown in Figure 10. For extraordinary vertices, there exist multiple valid choices for the weights  $\alpha + \beta + \gamma = 1$ .

The weights for our approach are the same as for MLCA subdivision [1]:

$$\alpha = \frac{9}{16}, \quad \beta = \frac{3}{8}, \quad \text{and} \quad \gamma = \frac{1}{16}. \quad (3.20)$$

Catmull / Clark suggested different weights, depending on the valence  $n$  of a  $\mathbf{v}$ -vertex [9]:

$$\alpha = 1 - \frac{7}{4n}, \quad \beta = \frac{3}{2n}, \quad \text{and} \quad \gamma = \frac{1}{4n}. \quad (3.21)$$

Both choices provide  $C^1$ -continuous limit surfaces at extraordinary points, see [37]. In the three-dimensional case, the subdivision used in our construction is still identical to MLCA, except for cases where type- $n$  cells ( $n > 3$ ) are involved. We do not use Catmull-Clark subdivision, since our lifting-based construction would not have an inverse when vertices of valence three occur.

When using wavelets, surfaces and volumetric shapes can be approximated arbitrarily closely by adding geometric detail expanded from wavelet coefficients. This implies that discontinuities of high-order derivatives across a represented shape can be reduced to a minimum at fine levels of resolution. Hence, the degree of smoothness at extraordinary vertices becomes a less important issue. Smooth representations are often preferred, however, since they reduce the number of wavelet coefficients required to approximate certain shapes.

## 4 Wavelet Construction

In the preceding section we defined stationary subdivision rules for arbitrary base lattices. Recursively applying these subdivision rules, denoted here by the operator  $\mathbf{S}$ , results in a (piecewise) smooth limit shape defining continuous basis functions associated with the control points of a lattice. In addition to subdivision, a wavelet transform provides the following three operators:

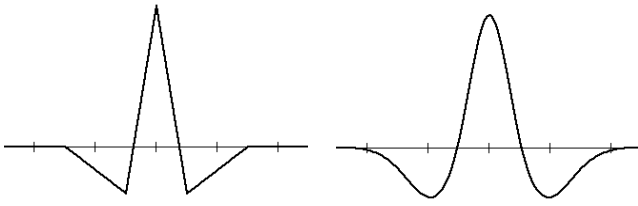


Figure 11: Linear and cubic B-spline wavelets.

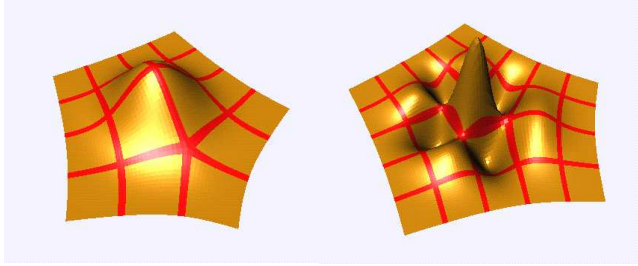


Figure 12: Two-dimensional scaling function and (face-)wavelet located near an extraordinary point.

- E** Expanding detail. At every level of refinement, geometric detail can be added to a subdivision volume. This detail is compactly stored as wavelet coefficients and can be expanded from these during subdivision.
- F** Fitting. This operation reverses a subdivision step. Based on all vertices on a fine level, the vertices on the next coarser level are predicted such that they provide a good approximation to the fine level when applying subdivision, again.
- C** Compacting detail. The difference between two levels of resolution, i.e., the displacement of mesh vertices when applying **F** followed by **S**, is compactly stored in the wavelet coefficients replacing the removed vertices.

The operation **E** is computed together with **S** by an extension of the subdivision rules. Subdivision and expansion of detail represent the reconstruction procedure of a wavelet transform. Recursively applied reconstruction steps correspond to the inverse wavelet transform.

The operators **F** and **C** provide the inverse of **E** and **S** and define a decomposition step. Repeated decomposition starting with a fine-resolution lattice and ending at the base lattice corresponds to the wavelet transform. For our wavelet construction, the reconstruction process is computed by local vertex manipulations whose inverse can be computed locally, too. Hence, the decomposition process is obtained by inverting every single vertex manipulation in the reverse order of reconstruction.

In a previous work, we have constructed wavelets for bilinear and bicubic subdivision surfaces that we used for multiresolution modeling of large-scale isosurfaces [3]. The computation of this wavelet transform is based on the lifting operations described in the preceding section. These are the reconstruction rules for our wavelet constructions:

**Linear B-spline wavelet reconstruction:**

$$\begin{aligned} s &- \text{lift}\left(-\frac{1}{4}\right); \\ w &- \text{lift}\left(\frac{1}{2}\right); \end{aligned} \tag{4.1}$$

**Cubic B-spline wavelet reconstruction:**

$$\begin{aligned} s &- \text{lift}\left(-\frac{3}{8}\right); \\ w &- \text{lift}\left(\frac{1}{2}\right); \\ s &- \text{lift}\left(\frac{1}{2}\right); \\ s &- \text{scale}\left(\frac{1}{2}\right). \end{aligned} \tag{4.2}$$

One- and two-dimensional basis functions of the wavelet transform are depicted in figures 11 and 12. The shape of recursively generated basis functions can be visualized by modifying a single coefficient corresponding to a vertex in the lattice and then applying the inverse wavelet transform.

We observe that the additional s-lift operations in equations (4.1) and (4.2) do not modify the subdivision process. When the wavelet coefficients, i.e. the **e**-, **f**-, and **c**-vertices, are zero prior to subdivision, then the first lifting operation does not have an impact. The first lifting operation determines the shape of the wavelets that are composed of scaling functions (corresponding to **v**-vertices) of the next finer level.

It is possible to obtain different wavelet constructions based on the same subdivision scheme by modifying this first lifting step. Our biorthogonal wavelet construction is driven by the need for efficiency and simplicity considering large meshes. Its fitting operation could be further improved, however, by a more expensive first lifting operation taking into account a broader stencil of manipulating vertices. For example, it is possible to approximate orthogonality between wavelets and scaling functions on the same level, resulting in a least-squares fitting operator **F**, following Lounsbery’s approach [28, 29]. Since the computation of inner products of basis functions required for orthogonalization is computationally expensive, Duchaineau [13] and Samavati / Bartels [42] suggest to use a discrete norm minimizing the coefficients rather than the geometric distance. Both approaches would work with our subdivision scheme, as well.

The time complexity of our wavelet transform is linear in the number of transformed coefficients / control points. At every level transition, each coefficient is modified by a fixed (small) number of lifting steps, such that decomposition and reconstruction for one level transition are linear-time operations. Considering a fine-to-coarse transition, the number of coefficients transformed again at the coarser levels is at most one half of all coefficients (for one dimension and less for higher dimensions). Hence, the computation time of the entire transform is linear, since  $O(n) + O(\frac{n}{2}) + O(\frac{n}{4}) + O(\frac{n}{8}) + \dots = O(n)$ .

In the remainder of this section we provide a few implementation details for our wavelet transform. The use of our specific lifting operations makes the required data structures fairly simple. In particular, we need to store only *downwards connectivity* with respect to the dimensionality of primitives. This means that cells are only linked to their faces, faces are only linked to edges, and edges to vertices. There is no need for *upwards connectivity*. The only upwards information needed for vertices, edges and faces are the numbers of incident primitives that have the same feature-dimensionality. No explicit links to these primitives are required.

When computing an s-lift operation, we traverse the cells, faces and edges of a mesh and transport their contributions downwards to the primitives of lower dimensionality. To add these contributions with the correct weights, the low-dimensional primitives like vertices and edges need to “know”, how many primitives of the same type they are connected to. When computing a w-lift, we traverse again the cells, faces, and edges and collect the contributions of the lower-dimensional manipulating primitives.

When implementing the decomposition rules, every individual vertex manipulation must be inverted by subtracting the terms that were added during the reconstruction process. In the case of an s-scale operation, the coefficients would be divided by the scaling pa-

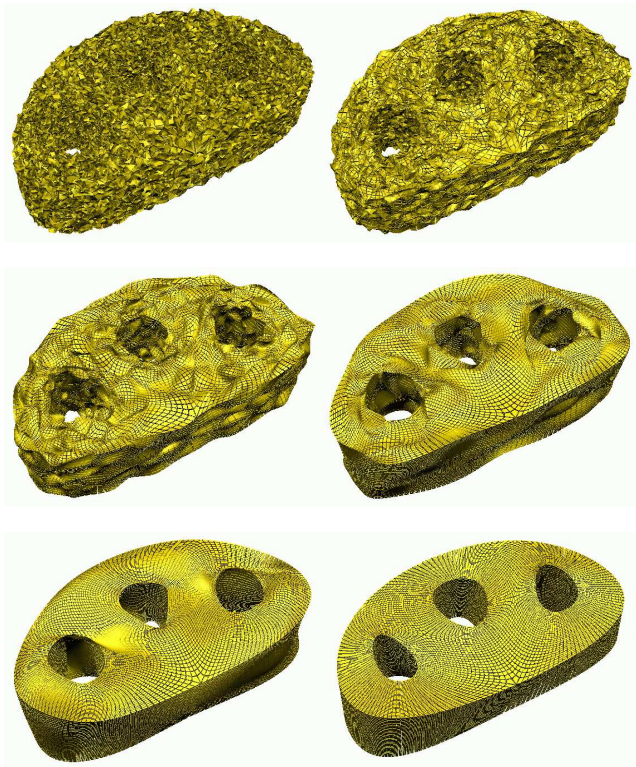


Figure 13: The noise-filtering experiment examining the stability of our wavelet construction.

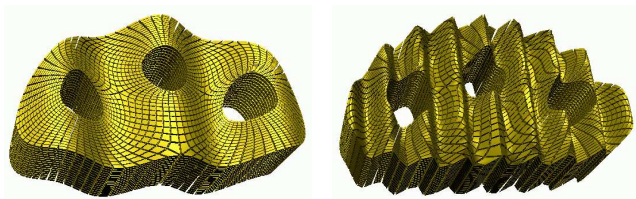


Figure 14: Geometric detail of a subdivided shape, obtained, for example by low- and high-frequency deformations is compactly represented by wavelet coefficients.

parameter, rather than multiplied. All lifting operations used in equation 4.2 have an inverse of the same type that is easily constructed. When inverting the individual lifting steps of the reconstruction, the order of operations needs to be reversed. The resulting inverse of equation 4.2 defines our wavelet decomposition.

We note that our transform can be computed in integer arithmetic for lossless compression purposes. Here, the terms added/subtracted would be rounded to closest integers in the same way for decomposition and reconstruction. The  $s$ -scale operation is ignored when using integer arithmetic, reducing the precision of low-level coefficients.

## 5 Results and Applications

We provide numerical examples for the stability of our wavelet construction and summarize the most relevant applications of our technique.

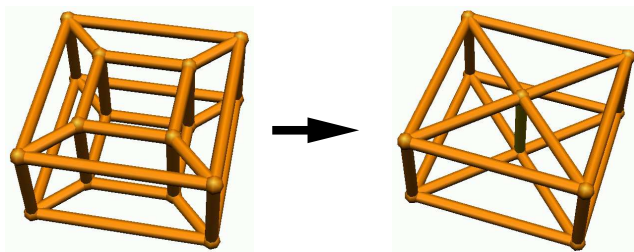


Figure 15: Transition of polyhedral meshes representing a time-varying surface. Four inner faces of the first mesh are collapsed to an edge in the second mesh where the genus of the surface is decremented. To obtain a manifold, the inner edge must be removed.

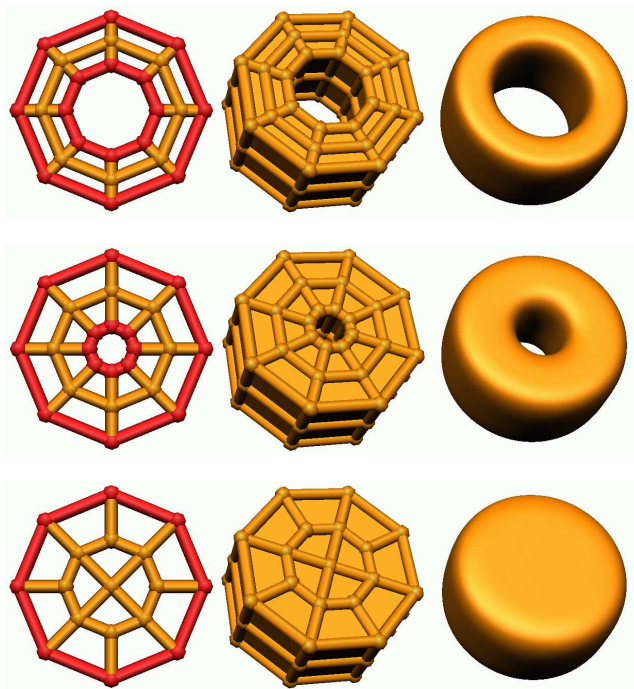


Figure 16: Subdivision volume representing a time-varying surface with genus reduction. This figure shows the footprints and the meshes for the first subdivision (with geometric detail added), as well as the limit surfaces.

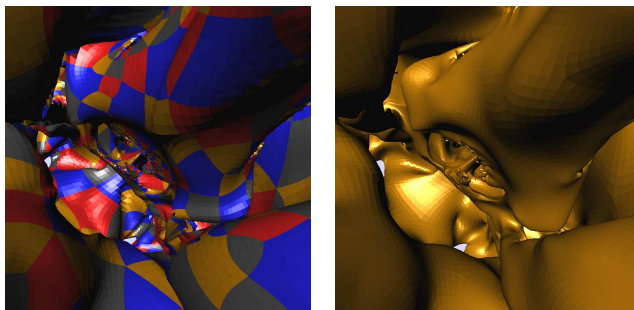


Figure 17: Subdivided base mesh shrink-wrapped to an isosurface (left) and a low-resolution representation using 1.6 percent of wavelet coefficients (right).

## 5.1 Stability

Considering the subdivision  $\mathbf{S}$ , we know that our scheme converges locally to polynomial segments. Due to the convex-hull property of B-splines that holds for our approach as well, there are no stability objections involved with the subdivision scheme.

Our fitting operator  $\mathbf{F}$  does not provide a least squares fit, since it is computed locally using a very small stencil of coefficients as input. A mathematical definition of stability, requiring that wavelet coefficients representing any bounded function are bounded (assuming an infinite number of levels) may not be satisfied by our transform. In practical applications, however, we have found that our transform does not exhibit any stability problems. Additionally, in the case of subdivision volumes the number of level transitions is typically low.

level index	number of vertices	max. displacement
5	277662	1.00
4	36686	1.23
3	5094	1.44
2	770	2.15
1	132	3.46
0	28	1.95

Table 1: Displacement of the lattice at different fitting levels due to noise added to representation at the fifth subdivision level.

A relevant measure for the stability of a fitting scheme is its ability to deal with noisy data. This is necessary to avoid the accumulation of numerical errors that may already be present in a data set or could be introduced by the transform. For a noise-removal experiment, we have randomly perturbed the control points of a subdivided lattice (see figure 8) at the fifth subdivision level. Then, for each level of fitting we recorded the maximal displacement of a vertex (at the fifth subdivision level). The results are shown in table 1 and in figure 13.

These results show that the noise amplitude grows moderately and is reduced again at the base level. In the case of a highly unstable transform, the noise amplitude would rapidly grow such that the shape of the represented object could not be recognized anymore. Our linear wavelet construction provides a greater level of stability, but it leads to much larger wavelet coefficients when representing smooth shapes [4].

## 5.2 Applications

Our wavelet transform provides a sparse representation for subdivided lattices that are re-shaped by some process, see figure 14. The shape modification can be performed on a very fine subdivision level, or it can be done gradually during the subdivision process. Due to the fitting/compaction procedure, detail of high frequency can be preserved while the coarse shape is edited at a low resolution. Some of the most interesting applications for our technique are described in the following

- Freeform deformations. Algorithms exploiting volume subdivision of lattices as a freeform-deformation tool have already been devised by MacCracken/Joy [30]. With the additional operations provided by a wavelet transform, we can easily apply highly-detailed deformations of three-dimensional objects based on low and high frequencies at the same time.
- Multiresolution editing. The Shape of a solid object can be created at multiple levels of resolution. When the coarse shape is modified, the high-resolution detail is still present in the geometric model. Using volumes rather than surfaces will extend these techniques to the representation of inner structures of solids, like material compositions.

- Visualization of large-scale geometries. Subdivision surfaces can be fitted to implicitly defined geometries, such as isosurfaces, see figure 17. Multiresolution representations of this kind are used for visualization of large structures on low-end graphics hardware and for view-dependent rendering.
- Time-varying surfaces can be represented as a sequence of meshes embedded in a single lattice. Every layer of the lattice produced by subdivision can be considered as a surface at a certain time. It is feasible to model local topology changes by collapsing or subdividing polygons of surface meshes corresponding to consecutive time steps. An example for a genus transition is shown in figures 15 and 16. This technique has been successfully applied to represent one-dimensional contour sets of scalar fields [2].
- Lossless compression. Vertices represented at finite precision (or snapped to a certain regular grid) can be represented by integer coordinates. The lifting scheme is then computed using integer arithmetic such that no quantization of wavelet coefficients is necessary for compression purposes.

## 6 Conclusions and Future Work

We have introduced a promising technique for multiresolution modeling of arbitrary volumes combining smooth subdivision surfaces with biorthogonal wavelets. Our representation is  $C^2$ -continuous almost everywhere and is capable of representing volumes with features of different dimensionalities at multiple levels of resolution.

One of the most interesting applications of our technique is the adaptive representation of time-varying surfaces. In the past, a variety of fitting approaches have been developed using subdivision surfaces and wavelets for the representation of highly-detailed geometries. New algorithms need to be devised for the fitting of lattices to certain geometric shapes such that our new technique can develop its full potential.

## References

- [1] C. Bajaj, S. Schaefer, J. Warren, and G. Xu, *A subdivision scheme for hexahedral meshes*, The Visual Computer, special issue on subdivision, vol. 18, 2002, pp. 343–356.
- [2] M. Bertram, D.E. Laney, M.A. Duchaineau, C.D. Hansen, B. Hamann, and K.I. Joy, *Wavelet representation of contour sets*, Proceedings of IEEE Visualization, Oct. 2001, pp. 303–310 & 566.
- [3] M. Bertram, M.A. Duchaineau, B. Hamann, and K.I. Joy, *Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization*, Proceedings of IEEE Visualization, Oct. 2000, pp. 389–396 & 579.
- [4] M. Bertram, *Multiresolution Modeling for Scientific Visualization*, Ph.D. Thesis, University of California at Davis, July 2000. <http://daddi.informatik.uni-kl.de/~bertram/>
- [5] H. Biermann, I.M. Martin, D. Zorin, and F. Bernardini, *Sharp features on multiresolution subdivision surfaces*, Proceedings of Pacific Graphics 2001, IEEE, pp. 140–149.
- [6] H. Biermann, A. Levin, and D. Zorin, *Piecewise smooth subdivision surfaces with normal control*, Computer Graphics, Proceedings of Siggraph 2000, ACM, pp. 113–120.

- [7] G.-P. Bonneau, *Multiresolution analysis on irregular surface meshes*, IEEE Transactions on Visualization and Computer Graphics (TVCG), Vol. 4, No. 4, IEEE, Oct.-Dec. 1998, pp. 365–378.
- [8] G.-P. Bonneau, *Optimal triangular Haar bases for spherical data*, Proceedings of Visualization '99, IEEE, 1999, pp. 279–284 & 534.
- [9] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer-Aided Design, Vol. 10, No. 6, Nov. 1978, pp. 350–355.
- [10] W. Dahmen and C.A. Micchelli, *Banded matrices with banded inverses II: Locally finite decomposition of spline spaces*, Construction Approximation, vol. 9, 1993, pp. 263–281.
- [11] T. DeRose, M. Kass, and T. Truong, *Subdivision surfaces in character animation*, Computer Graphics, Proceedings of Siggraph '98, ACM, 1998, pp. 85–94.
- [12] D. Doo and M. Sabin, *Behaviour of recursive division surfaces near extraordinary points*, Computer-Aided Design, Vol. 10, No. 6, Nov. 1978, pp. 356–360.
- [13] M. Duchaineau, *Dyadic splines*, Ph.D. thesis, Department of Computer Science, University of California, Davis, 1996.
- [14] N. Dyn, D. Levin, and J.A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*, ACM Transactions on Graphics, Vol. 9, No. 2, April 1990, pp. 160–169.
- [15] A. Finkelstein and D.H. Salesin, *Multiresolution curves*, Computer Graphics, Proceedings of Siggraph '94, ACM, 1994, pp. 261–268.
- [16] M. Halstead, M. Kass, and T. DeRose, *Efficient, fair interpolation using Catmull-Clark surfaces*, Computer Graphics, Proceedings of Siggraph '93, ACM, 1993, pp. 35–44.
- [17] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, *Piecewise smooth surface reconstruction*, Computer Graphics, Proceedings of Siggraph 94, ACM, 1994, pp. 295–302.
- [18] L. Kobbelt, *Sqrt(3)-subdivision* Computer Graphics, Proceedings of Siggraph 2000, ACM, 2000, pp. 103–112.
- [19] L.P. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel, *A shrink wrapping approach to remeshing polygonal surfaces*, Proceedings of Eurographics '99, Computer Graphics Forum, Vol. 18, Blackwell Publishers, 1999, pp. 119–129.
- [20] L. Kobbelt, *Interpolatory subdivision on open quadrilateral nets with arbitrary topology*, Proceedings of Eurographics '96, Computer Graphics Forum Vol. 15, Blackwell Publishers, 1996, pp. 409–420.
- [21] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, *MAPS: multiresolution adaptive parameterization of surfaces*, Computer Graphics, Proceedings of Siggraph '98, ACM, 1998, pp. 95–104.
- [22] A. Levin, *Polynomial generation and quasi-interpolation in stationary non-uniform subdivision*, Technical report. <http://www.math.tau.ac.il/~levin/adi/>
- [23] Adi Levin, *Combined subdivision schemes*, Dissertation, School of Mathematical Sciences, Tel-Aviv University, 2000.
- [24] L. Linsen, V. Pascucci, M.A. Duchaineau, B. Hamann, and K.I. Joy, *Hierarchical representation of time-varying volume data with 4th-root-of-2 subdivision and quadrilinear B-spline wavelets*, submitted to: Pacific Graphics 2002, IEEE, 2002.
- [25] N. Litke, A. Levin, and P. Schroder, *Trimming for subdivision surfaces*, Computer-Aided Geometric Design, vol. 18, no. 5, Elsevier, 2001, pp. 463–481.
- [26] N. Litke, A. Levin, and P. Schröder, *Fitting Subdivision Surfaces*, Proceedings of IEEE Visualization 2001, pp. 319–324 & 568.
- [27] C.T. Loop, *Smooth subdivision surfaces based on triangles*, M.S. thesis, Department of Mathematics, University of Utah, 1987.
- [28] J.M. Lounsbery, *Multiresolution analysis for surfaces of arbitrary topological type*, Ph.D. thesis, Department of Mathematics, University of Washington, 1994.
- [29] M. Lounsbery, T. DeRose, and J. Warren, *Multiresolution analysis for surfaces of arbitrary topological type*, ACM Transactions on Graphics, Vol. 16, No. 1, ACM, Jan. 1997, pp. 34–73.
- [30] R. MacCracken and K.I. Joy, *Free-form deformations with lattices of arbitrary topology*, Computer Graphics, Proceedings of Siggraph '96, ACM, 1996, pp. 181–188.
- [31] K.T. McDonnell and H. Qin, *FEM-based subdivision solids for dynamic and haptic interaction*, Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, pp. 312–313, June 2001.
- [32] A. Moffat, R.M. Neal, and I.H. Witten, *Arithmetic coding revisited*, ACM Transactions on Information Systems, Vol. 16, No. 3, July 1998, pp. 256–294.
- [33] A.H. Nasri and M.A. Sabin, *Taxonomy of interpolation constraints on recursive subdivision surfaces*, Visual Computer, vol. 18, no. 4, Springer, 2002, pp. 382–403.
- [34] G.M. Nielson, I.-H. Jung, and J. Sung, *Haar wavelets over triangular domains with applications to multiresolution models for flow over a sphere*, Proceedings of Visualization '97, IEEE, 1997, pp. 143–150.
- [35] V. Pascucci, *Slow-growing subdivision (sgs) in any dimension: towards removing the curse of dimensionality*, Proceedings of Eurographics, 2002, pp. 451–460.
- [36] J. Peters and U. Reif, *The simplest subdivision scheme for smoothing polyhedra*, ACM Transactions on Graphics, Vol. 16, No. 4, 1997, pp. 420–431.
- [37] J. Peters and U. Reif, *Analysis of algorithms generalizing B-spline subdivision*, SIAM Journal on Numerical Analysis, Vol. 13, No. 2, April 1998, pp. 728–748.
- [38] H. Prautzsch and G. Umlauf, *Improved triangular subdivision schemes*, Proceedings of Computer Graphics International, Hannover, Germany, IEEE, 1998. pp. 626–632.
- [39] H. Prautzsch, *Analysis of  $C^k$ -subdivision surfaces at extraordinary points* Technical Report, University of Karlsruhe (Germany), 1995.
- [40] U.A. Reif, *A unified approach to subdivision algorithms near extraordinary vertices*, Computer-Aided Geometric Design, Vol. 12, No. 2, Elsevier, March 1995, pp. 153–174.

- [41] F.F. Samavati, N. Mahdavi-Amiri, and R.H. Bartels, *Multiresolution representation of surface with arbitrary topology by reversing Doo subdivision*, Computer Graphics Forum, vol. 21, no. 2, 2002, pp. 121-136.
- [42] F.F. Samavati and R.H. Bartels, *Multiresolution curve and surface editing: reversing subdivision rules by least-squares data fitting*, Computer Graphics Forum, vol. 18, no. 2, 1999, pp. 97-119.
- [43] P. Schröder and W. Sweldens, *Spherical wavelets: efficiently representing functions on the sphere*, Computer Graphics, Proceedings of Siggraph '95, ACM, 1995, pp. 161-172.
- [44] T.W. Sederberg, D. Sewell, and M. Sabin, *Non-uniform recursive subdivision surfaces*, Computer Graphics, Proceedings of Siggraph '98, ACM, 1998, pp. 287-394.
- [45] J. Stam, *On subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree*, Computer Aided Design (CAD), Special issue on subdivision surfaces, vol. 18, no. 5, pp. 383-396.
- [46] J. Stam, *Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values*, Computer Graphics, Proceedings of Siggraph '98, ACM, 1998, pp. 395-404.
- [47] E.J. Stollnitz, T.D. DeRose, D.H. Salesin, *Wavelets for Computer Graphics—Theory and Applications*, Morgan Kaufmann Publishers Inc., San Francisco, California, 1996.
- [48] W. Sweldens, *The lifting scheme: a custom-design construction of biorthogonal wavelets*, Applied and Computational Harmonic Analysis, Vol. 3, No. 2, pp. 186-200, 1996.
- [49] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*, Morgan Kaufmann, 2001.
- [50] D. Zorin and P. Schroder, *A unified framework for primal/dual quadrilateral subdivision schemes*, Computer-Aided Geometric Design, vol. 18, (no. 5), Elsevier, June 2001. pp. 429-454.
- [51] D. Zorin and D. Kristjansson, *Evaluation of piecewise smooth subdivision surfaces*, Visual Computer, vol. 18, no. 5-6, 2002, pp. 299-315
- [52] D. Zorin, *A method for analysis of  $C^1$ -continuity of subdivision surfaces*, SIAM Journal on Numerical Analysis, vol. 37, no. 5, 2000, pp. 1677-1708.
- [53] D. Zorin, *Smoothness of subdivision on irregular meshes*, Constructive Approximation, Vol. 16, No. 3, 2000, pp. 359-397.
- [54] D. Zorin, P. Schröder, and W. Sweldens, *Interpolating subdivision for meshes with arbitrary topology*, Computer Graphics, Proceedings of Siggraph '96, ACM, 1996, pp. 189-192.