Variable Resolution 4–k Meshes: Concepts and Applications

Luiz Velho Jonas Gomes

IMPA – Instituto de Matemática Pura e Aplicada

Abstract

In this paper we introduce hierarchical 4–k meshes, a powerful structure for the representation of geometric objects at multiple levels of detail. It combines most properties of other related descriptions with several advantages, such as more flexibility and greater expressive power. The main unique feature of the 4–k mesh structure lies in its variable resolution capability, which is crucial for adaptive computation.

We also describe different methods for constructing the hierarchical 4–k mesh representation, as well as the basic algorithms necessary to incorporate it in modeling and graphics applications.


Keywords: multiresolution, four-directional grids, restricted quad-trees, multi-triangulations, adapted meshes.

1 Introduction

Hierarchical structures are the embodiment of fundamental abstraction mechanisms that allow us to deal with complexity. For this reason, such structures are an integral part of many tools in practically every area of human activity.

Hierarchies reflect dependency relations between entities at different levels. The specific nature of these relationships is determined by the application area, and by the problem to be solved.

In Geometric Modeling and Computer Graphics, hierarchical structures are often used to represent objects with multiple levels of detail. This type of hierarchy makes it possible to process the object at different resolutions. Thus, hierarchical structures are essential for most algorithms that require adapted computations. A typical example is the visualization of 3D polygonal surfaces, where the size of polygonal facets should be proportional to the projected area on the screen.

The importance of multiple levels of detail representations has motivated the development of various hierarchical structures which, in one way or another, support that capability.

In this paper, we present the hierarchical 4–k mesh structure. It combines most properties of other multiple level of detail representations and offers several advantages over them.

2 Basic Concepts

This section gives some definitions and basic notions that are used throughout the paper.

2.1 Meshes

A mesh is a cell complex, \( K = (V, E, F) \), where \( V, E \) and \( F \) are respectively sets of vertices \( v \in V \), edges \( (v_i, v_j) \in E \), and faces \( (v_i, v_j, \ldots, v_k) \in F \). The complex \( K \) provides a topological structure for the decomposition of two-dimensional domains.

The \( 1 \)-neighborhood \( N_1(v) \) of a vertex \( v \), consists of the set of vertices that share a face with \( v \). The valence (or degree) of a vertex, \( v \), is the number of edges incident in \( v \).

The size of a mesh, denoted by \( |K| \), is the number of faces in the set \( F \) of \( K \).

A geometric realization of the mesh \( K \) is created, by associating to each vertex \( v \) a coordinate value, \( p(v) \in \mathbb{R}^n \). When \( n = 2 \), \( K \) is a planar mesh and when \( n = 3 \), \( K \) is a surface in 3D.

A mesh is called conforming when faces that are spatially adjacent share exactly edges and vertices on common boundaries.

A mesh can be classified according to various criteria. Here we focus on the following three: cell type; mesh structure; and mesh geometry.

According to cell type, we usually work with homogeneous meshes, where 2D cells are \( n \)-sided faces, with \( n \) constant. The most common ones are: triangle meshes (\( n = 3 \)), and quadrilateral meshes (\( n = 4 \)). Note that it is always possible to triangulate an \( n \)-sided face. Therefore, it is sufficient to consider triangle meshes.

The mesh structure is related with the types of 1-neighborhoods in the mesh. In a regular mesh, the valence of all vertices is the same, while in an irregular mesh the valence may differ from vertex to vertex in an arbitrary way. A vertex with regular valence is called ordinary, otherwise it is called extraordinary.

The geometry of the mesh depends on its metric properties. An uniform mesh is a tessellation by regular \( n \)-gons, i.e. all edges have the same size. Meshes without this property are called non-uniform. Note that uniform meshes are also regular.

A related geometric property is the aspect ratio, which measures how close a face is from a regular \( n \)-gon. We remark that there are many ways to define this quantity.

The resolution of an uniform mesh is the number of vertices per unit length. The resolution of an irregular mesh can be determined locally from the length of its edges.

Two meshes \( K_m \) and \( K_n \) are compatible, if there is a subset of faces \( F_m \subset K_m \), that when it replaces a corresponding subset of faces in \( K_n \), the result is a conforming mesh. Correspondence in this case means spatial overlap.
2.2 Hierarchical Structures

A mesh hierarchy, $H$, is a sequence of meshes, $H = (K^j)_{j=1, \ldots, n}$, such that the size of the mesh $K^j$ increases monotonically with the index $j$. Furthermore, there is a dependency relation between faces at two subsequent levels $j$ and $j + 1$, whose support overlap.

Based on these dependency relations, it is possible to construct a hierarchical structure that defines the increasing sequence $H$. It is also possible to define the reverse of the hierarchy, which is the sequence in reverse order, where the mesh size is decreasing.

A mesh hierarchy is usually constructed by local modifications that either refine or simplify an initial mesh. Thus, one can start with a coarse mesh and subdivide it by applying a refinement operator; or, alternatively, one can start with a fine mesh and coarsen it by applying a simplification operator. Figure 1 shows a scheme of this process.

![Figure 1: Mesh hierarchy and construction mechanisms.](image)

Note that the modification operator provides the dependency relations necessary to build a hierarchical structure encoding the mesh hierarchy.

The nature of these operators and the method by which they are applied determines the properties of the hierarchy.

Here, we distinguish between hierarchical structures of two kinds: adaptive and non-adaptive.

A non-adaptive hierarchical structure defines only one mesh hierarchy. Examples of this kind of structure are multiresolution and progressive meshes.

In a multiresolution mesh, the modifications are applied in parallel to a set of independent regions that completely cover the mesh. Each step of this process changes the mesh resolution globally. The corresponding hierarchical data structure is a tree. A multiresolution structure is usually constructed using refinement [17].

In a progressive mesh, the modifications are applied sequentially to only one region of the mesh at a time. Each step of the process changes the mesh resolution locally. The corresponding data structure is a list. The progressive structure is usually constructed using simplification [12].

An adaptive hierarchical structure defines a family of mesh hierarchies. One example of this kind of structure is a variable resolution mesh.

In a variable resolution mesh, local modifications are applied to a set of independent regions, such that, the boundary of each region remains unchanged. Note that this set of regions may not cover the mesh completely.

Because of the boundary constraint, there is no interference between local modifications at each level, which can be applied independently of each other. The above property makes it possible to generate many sequences of meshes using permutations of independent local modifications.

The corresponding data structure is a directed acyclic graph (DAG), that encodes dependencies across levels.

A variable resolution mesh can be constructed either by refinement or simplification [16].

3 Four Directional Grids

This section gives some background on four directional tessellations of the plane, which are the basis of hierarchical 4-k meshes.

3.1 [4.8^2] Laves Tilings

Laves tilings are crystallographic groups that generalize regular planar tilings [10]. They are monohedral tilings with regular vertices.

In a monohedral tiling, all tiles are congruent to a single tile, called a prototile. Therefore, all tiles have the same shape and size.

A vertex, $v$, of a tiling is called regular if the angle between consecutive edges incident in $v$ is $2\pi/d$, where $d$ is the valence of $v$.

Laves tilings are classified by the valences of the vertices of their prototiles.

There are eleven types of Laves tilings. Here, we focus on the [4.8^2] Laves tiling, for which the prototile is an isosceles triangle with vertices of valence 4, 8, and 8. This tiling is shown in Figure 2.

![Figure 2: [4.8^2] Laves tiling](image)

This tiling has a rich fourfold set of symmetries. Also note that the [4.8^2] Laves tiling is a triangulated quadrangulation. Thus, it has the advantages of triangular and quadrilateral meshes.

3.2 Quincunx Lattices

The quincunx lattice [2] is the set of points $Q = \{ Mx; x \in \mathbb{Z} \times \mathbb{Z} \}$, where $M$ is the quincunx matrix.

$$ M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1} $$

In a [4.8^2] Laves tiling, we can divide the vertices into two classes: valence 4 vertices, $v \in V_4$, with $\deg(v) = 4$; and valence 8 vertices, $v \in V_8$, with $\deg(v) = 8$.

The vertices $V = V_4 \cup V_8$ of the [4.8^2] tiling belong to two interleaved quincunx lattices. That is, $v \in V_4 \Rightarrow p(v) \in Q_0 = \{ Mx; x \in \mathbb{Z}^2 \}$, and $v \in V_8 \Rightarrow p(v) \in Q_1 = \{ (1, 0) + Mx; x \in \mathbb{Z}^2 \}$. See Figure 3. Note that, the union of $Q_0$ and $Q_1$ is the integer lattice $\mathbb{Z} \times \mathbb{Z}$.

![Figure 3: Interleaved quincunx lattices and [4.8^2] tiling](image)
3.3 The Four Directional Grid and Box Splines

4–8 tessellations are closely related with the four directional grid that is generated by the set of vectors \((e_1, e_2, e_1 + e_2, e_1 - e_2)\), where \(e_1 = (1, 0)\) and \(e_2 = (0, 1)\). See Figure 4.

The four directional grid is well known in the theory of Box splines [4]. Box splines are piecewise polynomial functions created by convolution along a prescribed set of directions. The simplest smooth box spline over a four directional grid is the Zwart-Powell basis [25]. This function is piecewise quadratic, with \(C^1\) continuity across grid lines. They are defined by the set of directions \(\mathbf{D}\), shown in matrix form below

\[
D = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix}
\]  

(2)

Note that this is the same set of vectors associated with the four directional grid.

Figure 5 shows the support of the Zwart-Powell function on the underlying grid.

4 Multiresolution 4–8 Structures

This section discusses how to define a representation for multiresolution meshes based on 4–8 tessellations.

4.1 Construction

4–8 tessellations are refnible tilings. This property means that it is possible to subdivide a coarse 4–8 tiling and obtain a finer self-similar tiling. Therefore, we can construct a multiresolution 4–8 tessellation using refinement.

There are two alternative construction methods: quaternary subdivision and interleaved binary subdivision.

The quaternary subdivision refinement procedure of a 4-8 mesh \(K = (V, E, F)\), is as follows:

1. Split all edges \(e \in E\) at their midpoints \(m\):
2. Subdivide all faces \(f \in F\) into four new faces, by linking the degree 4 vertex, \(v \in V_4\), to the midpoint \(m\) of the opposite edge, and also linking \(m\) to the midpoints of the two other edges.

These data structures are called hierarchy of right triangles [7, 14, 6]. Note that, in the regular case, the structure does not need to be explicitly represented.

Alternatively, we can group adjacent triangles into quadrilaterals and represent the multiresolution 4–8 mesh as a triangulated quadtree [17, 11].
Note that, because of the special structure of the 4–8 tessellation, each node of the quadtree consists of four triangles created by subdivision of one diagonal of the triangulated quadrilateral. In fact, we have two interleaved quadtrees, one rotated by 45 degrees in relation to the other. See Figure 9.

5 4–8 Meshes

This section generalizes planar 4–8 tessellations into a family of mesh structures with similar properties. These meshes are all constructed by some form of 4–8 refinement.

5.1 Regular 4–8 Meshes

A regular 4–8 mesh is a homogeneous simplicial complex that is homeomorphic to a $[4,8]$ tiling. All vertices in a regular 4–8 mesh have valence 4 or 8, and are called ordinary vertices. Moreover, the 1-neighborhood of every vertex of valence 4 has only neighbors of valence 8, and the 1-neighborhood of every vertex of valence 8 consists of a ring of vertices with alternating valences 4 and 8.

A finer regular 4–8 mesh can be obtained from a coarse regular 4–8 mesh by refinement. (See Section 4.) Figure 10 shows a regular 4–8 mesh.

5.2 Semi-Regular 4–8 Meshes

A semi-regular 4–8 mesh is a tessellation which contains isolated extraordinary vertices with valence different than 4 or 8.

This mesh structure is created from a coarse irregular mesh by applying a semi-regular 4–8 refinement method that introduces only ordinary vertices [24]. In that way, as the mesh is refined, extraordinary vertices from the initial mesh are surrounded by ordinary vertices with regular valence 4 or 8.

Figure 11 shows a semi-regular 4–8 mesh.

5.3 Quasi-Regular 4–8 Meshes

A quasi-regular 4–8 mesh is a tessellation in which most vertices have regular valence 4 or 8, but irregular vertices are not guaranteed to be isolated. Therefore, this mesh does not possess the 1-neighborhood structure of a regular 4–8 mesh.

This type of mesh is created by processes that almost always introduce vertices with regular valence [23].

6 Variable Resolution Triangulations

This section defines more precisely some basic notions concerning adaptive hierarchical structures.

The idea of a variable resolution triangulation was introduced independently by Floriani et al. [9] and De Berg et al. [3]. Subsequently, Puppo developed an extensive theoretical framework for general variable resolution structures [15], which he called Multi-Triangulations.

6.1 Definitions

As mentioned in Section 2, hierarchical mesh structures are based on local modifications. In the variable resolution setting, it is necessary to employ a restricted class of local modifications: the ones that are minimally compatible

A minimally compatible local modification, $W(K_s)$, to a sub-mesh $K_s \subseteq K$ of a mesh $K = (V, E, F)$, is a substitution of $K_s$ by $W(K_s)$ in $K$, such that:

1. The boundary edges of $K_s$ are not altered;
2. The interior edges of $K_s$ are totally replaced by new edges.

The sub-meshes $K_s$ and $W(K_s)$ are, respectively, the domain and range of the modification operator $W$.

Compatibility is enforced by condition (1). Since the boundary $\partial K_s$ does not change, the new sub-mesh $W(K_s)$ is compatible with $K_s$ and the modification operator produces a conforming mesh.

Minimality is addressed by condition (2). Since the interior of $K_s$ changes completely, there is minimal redundancy between the sub-meshes $K_s$ and $W(K_s)$.

The modification operator $W$ is called increasing if $|W(K_s)| > |K_s|$. This means that $W$ is a refinement operator. Similarly, $W$ is called decreasing if $|W(K_s)| < |K_s|$. In this case, it is a simplification operator.

Figure 9: Interleaved quadtrees.

Figure 10: Regular 4–8 mesh.

Figure 11: Semi-Regular 4–8 mesh.
A compatible sequence of meshes, \((K^0, K^1, \ldots, K^n)\), is generated by the application of a sequence of modifications \((W_1, W_2, \ldots, W_{n-1})\), starting with an initial mesh \(K^0\). This produces the sequence of meshes \((K^0, W_1(K_1), \ldots, W_{n-1}(K_{n-1}))\), where

\[ K^j = W_{j-1}(W_{j-2}(\cdots W_1(K_1))) \]  

for \(j > 0\).

Note that, given an intermediate mesh, \(K^m\), and two independent modifications \(W_j\) and \(W_i\) that are compatible with \(K^m\), we can apply either one of them to \(K^m\), in order to produce a new mesh \(K^{m+1} = W_j(K_i)\) or \(K^{m+1} = W_i(K_j)\), with \(K_j, K_i \subset K^m\).

The purpose of a variable resolution structure is to encode all possible mesh hierarchies that can be generated from a sequence of modifications \((W^i)_{i=1,\ldots,n-1}\). In order to achieve this goal, we need to distinguish between dependent and independent modifications.

A variable resolution mesh, \(V = (K^0, W; \leq)\) is defined by an initial mesh \(K^0\), a set of minimally compatible local modifications \(W = \{W_1, W_2, \ldots, W_{n-1}\}\), and a partial order relation \(\leq\) on \(W\), that satisfies the following conditions:

1. **Dependency:** \(W_i \prec W_j\), if and only if there is a face \(f \in F_i\) in the domain \(K_i\) of \(W_i\) that belongs to the range \(W_j(K_j)\) of \(W_j\). In other words, precedence is determined by compatibility of dependent modifications.

2. **Non-redundancy:** \(f \in F_i\) of \(W_i(K_i)\) implies that \(f \notin F_j\) of \(W_j(K_j)\) for all \(j \neq i\). In other words, there are no duplicate faces.

### 6.2 Representation

The partial order relations can be described by a directed acyclic graph (DAG), where the nodes are associated with modifications \(W_i\), and there is an arc from \(W_i\) to \(W_j\) whenever \(W_j\) is the successor of \(W_i\) according to the partial order relation \(\leq\).

We construct a lattice representation of a variable resolution mesh by adding a source and a drain to the DAG.

In this representation, each face, \(f\), is referenced by exactly two nodes. It appears in the range and in the domain of a modification. The node having \(f\) in its domain is called successor of \(f\), and the node having \(f\) in its range is called predecessor of \(f\).

The source node is associated with a constructor of the initial mesh \(K^0\), and the drain node is associated with the application of all modifications \(W_i, i = 1, \ldots, n-1\), to \(K^0\), that produces the final mesh \(K^n\). Appropriate arcs are added to and from these two special nodes.

A cut of a DAG consists of a set of nodes disconnecting it. A front in a lattice is a cut which contains exactly one arc for each path from the source to the drain.

Figure 12 shows an illustration of the lattice representation of a variable resolution mesh.

### 7 Hierarchical 4–k Meshes

This section describes a hierarchical structure to encode the family of multiresolution 4–8 meshes discussed in Section 5. This structure also allows to generate a larger class of mesh hierarchies, which we call variable resolution 4–k meshes.

A variable resolution 4–k mesh is a special case of the variable resolution triangulation, defined in Section 6. Because of its particular nature, it has unique desirable properties not available in general hierarchical structures.

#### 7.1 Variable Resolution 4–k Structure

The variable resolution 4–k mesh is built from a restricted set of local modifications defined on a cluster of two triangular faces. These two modifications are:

1. **Internal edge split:** the edge shared by two adjacent faces is subdivided, and the two faces are replaced by four faces. See Figure 13.

2. **Internal edge swap:** The edge shared by two adjacent faces is replaced by another edge linking the opposite vertices in each face. See Figure 14.
Note that these modifications make sense only if the two-face cluster is convex. Also note that modification (i) is exactly the binary subdivision step of 4–8 refinement applied to adjacent faces. The inverse of (i) is an edge collapse. It can be shown that these operations are sufficient to make any topology preserving transformation to a mesh [13].

Another important observation is that both (i) and (ii) are edge-based modifications. We exploit this fact to design data structures for representing variable resolution meshes.

The description combines edge and face elements. Modifications of type (i) are associated with an edge that splits or collapses. The edge points out to the two faces sharing it. Additionally, a face points out to its parent and two children.

This representation is illustrated in Figure 15.

The specification of these data structures in pseudo-C is given below. A face is represented by the structure:

```c
Face {
    Hedge* edge[3];
    Face* parent, children[2];
}
```

where we adopt the convention that the split edge of a face is `edge[0]` (i.e. `split_edge(f) := f.edge[0]`).

A edge is represented using an augmented half-edge data structure:

```c
Hedge {
    Vertex* point;
    Hedge* mate;
    Face* fbase;
}
```

These two data structures provide a compact way to encode the variable resolution 4–8 mesh, as well as its inverse. They also make possible the efficient implementation of all relevant operations. For example, the domain of a refinement \( W(e) \) is

```c
Set domain_w(Hedge e) {
    return make_set(e.fbase, e.mate.fbase);
}
```

The range of a refinement \( W(e) \) is

```c
Set range_w(Hedge e) {
    return make_set(e.fbase.children[0], e.fbase.children[1], e.mate.fbase.children[0], e.mate.fbase.children[1]);
}
```

The successor refinement of a face \( f \) is

```c
Hedge* successor_f(Face f) {
    return split_edge(f);
}
```

The predecessor refinement of a face \( f \) is

```c
Hedge* predecessor_f(Face f) {
    return split_edge(f.parent);
}
```

The representation of type (ii) modification, corresponding to an edge swap, it uses the same data structures. The implementation is very similar. We take advantage of the fact the each face has only one child in the context of this operation. Thus, we set `f.children[1]=NULL`.

7.2 Properties

The effectiveness of a variable resolution structure can be analyzed according to the following criteria [15]:

- **Expressive Power**: the number of different meshes that can be built from a variable resolution structure. It is equal to the number of distinct fronts in the lattice representation. This property is relevant to the adaptivity of the mesh;

- **Depth**: the number of levels of the longest path from source to drain in the lattice representation. This property is relevant to structure traversal operations, such as point location.

- **Growth Rate**: the ratio between the size of the longest sequence of modifications and the size of its cumulative application to a mesh. When this rate is bounded by a constant the growth is linear. This property is relevant to the performance of selective refinement operations.

According to the above criteria, the variable resolution 4–8 mesh structure has all the desirable properties, as will be shown below. A node of the DAG in the lattice representation of a 4–8 mesh structure has some special characteristics because of the nature of the 4–8 refinement operator \( W \). The number of faces in the range \( Wi(K) \) of \( W \) is always 4, and the number of faces in the domain of \( K \) of \( Wi \) is always 2. As a consequence, a node \( Wi \) has exactly two incoming arcs (the two nodes that generate the faces in the domain of \( Wi \)), and four outgoing arcs (the four nodes that reference one of the faces in the range of \( Wi \)). This is illustrated in Figure 16.

In the following analysis we will consider the case of a regular 4–8 variable resolution mesh, which features optimal properties among the family of 4–8 meshes.
In a regular 4–8 mesh, the initial mesh $K^0$ has arbitrary size, $|K^0| = n$. For a hierarchical structure with $m$ levels, at each refinement step, $j = 1, m$, binary subdivision is applied to an independent set of two-face clusters that completely cover the mesh. Moreover, all clusters at subsequent levels $j$ and $j + 1$ are interleaved. As a consequence, there are $2^j$ nodes in the DAG at each even level $j$. The size of the refined mesh produced by applying all modifications up to level $j$ is $2^j n$.

The variable resolution structure of a regular 4–8 mesh has the following properties:

- **High expression power**: It can be shown that the number, $p$, of distinct meshes produced by the 4–8 structure with $m$ levels is equal to

$$p = \sum_{j=1}^{m} \sum_{k=0}^{j} \left(\frac{2^j}{k}\right)$$

(4)

As an example, for $m = 6$, the expression power is $p = 1844674407800484724$.

- **Logarithmic depth**: the number of levels of a 4–8 structure with $q = 2^m$ nodes is approximately $l = \log_2 q$.

- **Linear Growth**: the growth rate is bounded by the ratio between the sizes of the range and domain of the modifications, which in the case of internal edge split is 2. It can be shown that the growth rate $g$ of a 4–8 structure is bounded by

$$g = \frac{n + 2}{n + 1}$$

(5)

### 8 Construction Methods

This section gives an overview of the methods used to generate a variable resolution 4–k mesh.

We remark that it is important to have a variety of construction methods, so that they can be applied in distinct situations, such as free form modeling, surface approximation, and conversion of representations, to name a few.

The main categories of methods are the ones based on refinement and simplification.

#### 8.1 Refinement-Based Methods

We subdivide the refinement-based methods into three types: semi-regular, quasi-regular, and irregular.

The **semi-regular** refinement method employs topology based subdivision. It generalizes the regular 4–8 mesh refinement and uses interleaved edge splits. A complete description of the algorithm can be found in [24].

The method produces semi-regular meshes suitable for implementing stationary subdivision schemes. Figure 18 shows various subdivision surfaces generated with such schemes. The shape in this example is the “Stanford Bunny”. The control polyhedron, shown in Figure 18(a), is a coarse mesh obtained from the original data through simplification [21].

The most natural scheme to implement using 4–8 semi-regular meshes is a generalization of subdivision for Box splines defined on four directional grids [24]. Figure 18(e) shows a $C^0$ subdivision surface based on the Zwart-Powell basis. Figure 18(f) shows a $C^4$ subdivision surface based on the degree 6 Box spline.

Because of the quadrangulated structure of semi-regular 4–8 meshes, it is also suitable for the implementation of subdivision schemes originally designed for quadrilateral meshes [5, 1]. This is achieved through a decomposition of primal and dual quadrilateral refinement into interleaved binary subdivision steps [20]. Figure 18(c) shows a biquadratic B-spline surface based on the Doo-Sabin scheme. Figure 18(d) shows a bicubic B-spline surface based on the Catmull-Clark scheme.

The **quasi-regular** refinement method employs geometry sensitive subdivision. At each level, it covers the mesh with two-face clusters selected using an edge length criteria. This method produces a mesh that combines quasi-regular 4–8 topology with almost uniform geometric features. A complete description of the algorithm can be found in [23].

The quasi-regular mesh structure allows the implementation of quasi-stationary subdivision schemes. Figure 18(d) shows an example of a quasi 4–8 subdivision surface.

The **irregular** refinement method employs adaptive subdivision. It is based on multiresolution edge sampling. This method produces hierarchical meshes that conform to the shape of existing objects. A complete description of the algorithm can be found in [19].

The irregular 4–8 mesh structure is suitable for adaptive surface tessellation. Because the subdivision algorithm is very general, it can work with both parametric or implicit surface descriptions.

Figure 19 gives some examples of surfaces approximated by adapted irregular 4–8 meshes.

Figures 19(a) and (b) show a torus, defined implicitly by

$$f(x, y, z) = (x^2 + y^2 + z^2 - r^2 - 1)^2 - 4r^2(1 - z^2),$$

$x, y \in [-3, 3], z \in [-1, 1], r = 1.6$.

In Figure 19(a), we have an orthogonal projection of the base mesh together with the 3D grid; and Figure 19(b), the polygonal approximation, which contains 1324 triangles. The base mesh was constructed using a Coxeter-Freudenthal decomposition on a $4 \times 4 \times 2$ grid.

Figures 19(c) and (d), show the same torus, defined parametrically by

$$x = \cos u(r + \cos v), y = \sin u(r + \cos v), z = \sin v,$$

$u, v \in [0, 2\pi], r = 1.6$.

In Figure 19(c) we have the adapted decomposition of the parameter domain; and in Figure 19(d), the polygonal approximation, which contains 516 triangles. The base mesh was simply the subdivision of the rectangle $[0, 2\pi] \times [0, 2\pi]$ along its diagonal into two triangles. The algorithm has structured the parameter domain into a 4–8 hierarchy with three layers.

Note that the algorithm produces consistent results using either the parametric or implicit description of a surface.

Figures 19(e) and (f), show a digitized bust of Spock. The Cyberware data had 87040 points, structured into a regular cylindrical grid. In Figure 19(f), we have an adaptive mesh which approximates the surface within a prescribed tolerance, and in Figure 19(e), we have the corresponding domain decomposition. The facial details are clearly visible, because the regions of high curvature are sampled more densely than the rest of the surface.

#### 8.2 Simplification-Based Methods

Simplification-based methods construct the reverse of an increasing variable resolution 4–k mesh. They start with a fine mesh and coarsify it using the inverse of an edge split operation – an edge collapse. Therefore, they produce a decreasing hierarchical structure. For several reasons, it is advisable to establish the convention that the canonical lattice representation is an increasing structure, in which the source is a coarse mesh and the drain is a fine mesh. In this context, a simplification method builds the variable resolution representation “bottom-up”.
In order to perform the simplification of a mesh with 4–8 connectivity, it is sufficient to apply the internal edge collapse operator that transforms a cluster of four faces into a cluster of two faces. Moreover, the simplification procedure has to ensure that clusters at subsequent levels are interleaved. This type of method is only practical for regular 4–8 meshes.

In the case of arbitrary meshes, it is necessary to use also the edge swap operator. The reason is that, since an irregular mesh does not have 4–8 connectivity, it may not be possible to cover the mesh with clusters of four faces sharing a degree 4 vertex \( v \in V_i \). The edge swap operator is used to modify the mesh at each level, producing the required set of four-face clusters that cover most of the mesh [21].

Figure 20 shows an example of 4–k simplification. It is a cow model distributed with SGI’s powerflip demo. The initial mesh, shown in Figure 20(a) contains 5800 triangles. The sequence of simplified meshes at levels 3 to 7, is shown in Figures 20(b) through (f). They contain respectively, 1200, 700, 400, 300, and 200 faces.

9 Level of Detail Operations

This section considers the application of variable resolution 4-k meshes for managing level of detail of large geometric models. It defines the relevant operations and gives some examples.

9.1 Variable Resolution Queries

A level of detail operation consists in extracting a mesh \( \mathcal{K} \) from a variable resolution structure \( \mathcal{V} = (\mathcal{K}^0, \mathcal{W}, \leq) \). As we have seen in section 6, this mesh \( \mathcal{K} \subset \mathcal{V} \), corresponds to a front in the lattice representation of \( \mathcal{V} \), i.e., a set of arcs containing exactly one arc for each path from the source to the drain.

The collection of all nodes which can be reached from the source without traversing the arcs of the front, correspond to modifications to the mesh that are consistent with the partial order \( \leq \) and produce the extracted mesh \( \mathcal{K} \). See Figure 17.

![Figure 17: A front in the lattice representation.](image)

We can abstract this level of detail operation as a geometric query, \( \mathcal{Q} \), to the variable resolution structure \( \mathcal{V} \).

This general query operation can be specified by the following parameters [8].

- An adaptation function: \( \tau : K^0 \rightarrow \{0, 1\} \), that computes some measure over \( \mathcal{V} \) to determine if a face \( f \) produced by a modification should be accepted or not.

- A focus set: \( S \subseteq \mathbb{R}^3 \), that defines a region of interest where \( \tau(f) \) is evaluated.

The answer to the query \( \mathcal{K}_i \) is the smallest conforming mesh such that \( \tau(f) = 1 \) and \( f \cap F \neq \emptyset \), for all \( f \in K^0 \).

We remark that \( \mathcal{K}_i \) could be either a mesh representing the whole surface, or a sub-mesh containing just the elements inside the region of interest. In the first case, the query is called globally defined and in the second case, locally defined [8].

9.2 Adapted Mesh Extraction

Examples of variable resolution query operations are: point location; region intersection; neighbor search and adapted mesh extraction. The last one is particularly important, because it appears in many graphics applications, such as, progressive rendering, real-time visualization and interactive modeling.

Adaptive mesh extraction is implemented through a selective mesh refinement procedure using the variable resolution structure [8, 22].

This procedure can use a non-incremental or an incremental algorithm. The non-incremental algorithm starts with an initial front that contains all the arcs leaving the source node, and gradually advances the front, in a top-down fashion, based on the evaluation of the adaptation function and intersection with the focus set.

The incremental algorithm uses an existing front, and updates it, moving the front up or down if necessary, according to the adaptation function.

We remark that the variable resolution structure guarantees that an extracted mesh is consistent by construction.

Another nice feature of this framework is that, the mesh extraction procedure is independent of the query specification. As a consequence, it is straightforward to incorporate it in completely different application domains. This gives a lot of flexibility from the systems design point of view.

In that context, what distinguishes two adapted mesh extraction operations is the nature of the adaptation function. Some common types of applications are related to: shape approximation; view dependent geometry, etc.

The practical performance of level of detail operations is highly influenced by the properties of the underlying structure.

Next, we demonstrate the capabilities of the 4–k mesh structure in the context variable resolution queries.

Figure 21 exhibits few examples of adapted mesh extraction, using a variety of adaptation functions, as well as, variable resolution meshes constructed using different methods.

Figures 21(a) and (b) show two meshes representing a “saddle" surface that was defined parametrically by

\[
\begin{align*}
x &= u, & y &= v, & z &= (uv)^3 \\
u &\in [0, 1], & v &\in [0, 1].
\end{align*}
\]

The variable resolution structure was constructed using adaptive refinement. The adaptation criteria used in Figure 21(a) was triangle size. In Figure 21(b) the criteria was intersection with a rectangular region in the parametric domain.

Figures 21(c) and (d) show two versions of the “Stanford Bunny". The one in Figure 21(c) was constructed using the \( C^0 \) Box-spline subdivision scheme; and the one in Figure 21(d) was constructed using simplification. The adaptation criteria is the same.
for both models: it is a linear ramp in the horizontal direction determining triangle size.

Figures 21(e) and (f) show an example of point location using the cow model of Figure 20. In Figure 21(e) we have the complete mesh, in which the smallest triangle was picked by pointing at the screen. A detail of the area surrounding this point is shown in Figure 21(d).

We close this section with some remarks about a useful capability of the 4–8 mesh structure that allows the construction of a triangle strip representation of the extracted mesh [18]. Similarly to selective refinement, this algorithm starts with a path, defined on the coarsest mesh, and the path is refined while traversing the variable resolution structure. In particular, if the model has semi-regular 4–8 connectivity, it is possible to maintain a Hamiltonian path for all extracted meshes. See Figure 22 for examples.

10 Conclusions

This section concludes the paper with a review of the results and a discussion of future work.

A framework for variable resolution description of surfaces was presented. It is based on the hierarchical 4–8 mesh structure. This representation has several desirable properties for multiresolution applications.

We described various methods for constructing the 4–8 representation that contemplate most modeling situations.

We also demonstrated the practical use of the 4–8 representation, for the implementation of level of detail operations.

Future work in this area includes: hierarchical parametrizations; multiresolution decomposition; mesh compression; and the development of an integrated applications framework.

References


Figure 18: Surfaces generated by different subdivision schemes based on quasi 4–8 refinement (b), and semi-regular 4–8 refinement (c-f)
Figure 19: Surface Approximations using adaptive 4–8 refinement of implicit (a-b), parametric (c-d), and sampled (e-f) models.
Figure 20: 4-k Simplification of a cow model.
Figure 21: Adapted variable resolution 4-k mesh extraction.
Figure 22: Hamiltonian paths on semi-regular 4–8 meshes.