Visualizing Parametric Surfaces at Variable Resolution

Leila De Floriani[1], Paola Magillo[1], Enrico Puppo[2]

[1]Computer and Information Science Department (DISI), University of Genova,
Via Dodecaneso, 35 – 16146 Genova, Italy,
email: {deflo,magillo}@disi.unige.it
[2]Institute for Applied Mathematics (IMA), National Research Council,
Via de Marini, 6 (Torre di Francia) – 16149 Genova, Italy,
email: puppo@ima.ge.cnr.it

Abstract. A multiresoltion model is presented capable of handling different piecewise-linear approximations of a boundary representation of a solid object where faces are described by parametric surfaces. The level of detail of an approximation may be variable over different portions of the boundary, and the continuity of the surface across different patches is guaranteed.

1 Introduction

The approximation of curved surfaces through piecewise-linear (PL) surfaces made of triangular patches is a common practice when modeling solid objects. Triangle meshes are suitable representations for visualization purposes, since graphics hardware works on triangles. The *Level of Detail* (*LOD*) of a PL approximation is related to the error made in approximating each portion of the original surface with a corresponding linear patch. LOD generically depends on the resolution of the corresponding mesh, i.e., on the size and number of its triangles. In the applications, it is important to handle approximations whose LOD is variable over the surface: for instance, in real time visualization of objects with many faces (terrains, ships, aircrafts, cars, etc.), LOD will be higher near the viewpoint, and progressively lower at increasing distance from it. In general, a threshold function specifies the accuracy required at each point in space.

Building a PL surface whose LOD satisfies a given threshold, while its size is maintained as low as possible, is not an easy problem. Many existing methods (e.g., [Coh96, Hop96, Kle95, Kle96b, Sch92]) are based on iterative techniques that either simplify an initial mesh at a high resolution, or refine an initial mesh at a coarse resolution, according to some error-driven criterion. Although such methods are usually able to generate satisfactory results, they are computationally intensive. Therefore, they cannot be applied to recompute a new representation each time a new LOD is required in real-time applications.

A technique for mesh simplification can be used as a basis for the construction of a multiresolution model. The main idea is to have data organized into a structure that is built off-line, and contains information necessary to extract representations at different LODs in real time. Any given LOD is obtained through

a simple traversal of the model, involving a minimal amount of numerical computations.

In a multiresolution model, it is important to achieve a good tradeoff between the amount of information stored, and the time spent in information retrieval. Proposed models vary from explicit collections of representations at different LODs [VRML96], to compact models that implicitly provide information to iteratively modify an initial coarse mesh in order to achieve the desired LOD [Hop96, Kle96b].

In this paper, we present a multiresolution model that extends our recent proposal for scalar fields [Pup96, DeF96] to the case of boundary representations with curved surfaces. Our model can support variable LOD with high efficiency in terms of both query time and quality of the result. It is also fairly compact, not requiring much more storage space than a simple representation at the maximum available LOD. The main idea is that a large number of different PL surfaces can be obtained from a relatively small set of atomic components, which can be combined in different ways. The model encodes basic components arranged into a partially ordered set, which gives sufficient information to find all possible PL surfaces that can be built from triangles of the model. A compact data structure to encode our model, and an algorithm that can extract a representation of minimal size for a given LOD, on-line in time linear in the output size, are described in [Pup96, DeF97]. In this paper, we show how such model can be constructed from parametric boundary representations of solid objects.

2 Preliminaries

A *solid object* $\mathcal{O}$ is a compact subset of the physical space $\mathbb{R}^3$, whose boundary $\delta(\mathcal{O})$ is a 2-manifold. A *boundary representation* (*BRep*) describes $\mathcal{O}$ through its boundary $\delta(\mathcal{O})$. In a BRep, the boundary surface of $\mathcal{O}$ is partitioned into a finite set of patches, called *faces*. Faces are defined in such a way that any two faces do not intersect, except at their common boundaries, and the intersection of (the boundaries) of two faces is a finite set of lines and isolated points. The union of all faces covers the whole object boundary. The isolated points and the endpoints of the lines where the boundaries of two faces meet are called *vertices*, and the portions of intersection lines lying between two vertices are called *edges*. Faces, edges and vertices define a two-dimensional cell complex on the boundary of $\mathcal{O}$; a BRep encodes such a cell complex.

Here, we consider BReps where each face is a parametric surface. A *parametric surface* is the image of a continuous function $\phi : \Omega \longrightarrow \mathbb{R}^3$, where $\Omega \subseteq \mathbb{R}^2$ is a compact domain. For simplicity, we also assume that ϕ is injective. The model can be easily extended to the case of ϕ injective in the interior of Ω, the inverse image of any point p contains only a finite number of points, and only a finite set of points have more than two points in their inverse image. This latter class of functions captures most parametric surfaces in practical applications. We denote a parametric surface by $\phi(\Omega)$, a compact notation that specifies both the function and its domain. The space $\mathbb{R}^3$, in which a parametric surface $\phi(\Omega)$ is embedded, is called the *physical space*, while the space $\mathbb{R}^2$ containing the domain

Ω of ϕ is called the *parameter space*. In a parametric BRep, topological relations between vertices, edges and faces are defined both in physical and in parameter space.

We consider a *triangle mesh* $\mathcal{T}$ as a collection of triangles in 3D space, whose union is a 2-manifold surface (with or without boundary). Given a generic BRep $\mathcal{BR}$, a *piecewise-linear approximation* of $\mathcal{BR}$ is a triangle mesh, where each triangle is labelled with the name of a face of $\mathcal{BR}$, such that:

- the union of all triangles labelled with the name of a face $\phi_i(\Omega_i)$ of $\mathcal{BR}$ is a mesh $\mathcal{T}_i$ homeomorphic to $\phi_i(\Omega_i)$;
- for any pair of faces $\phi_i(\Omega_i)$ and $\phi_j(\Omega_j)$ adjacent along an edge e in $\mathcal{BR}$, the corresponding meshes $\mathcal{T}_i$ and $\mathcal{T}_j$ intersect at a chain of segments homeomorphic to edge e;
- for any set of faces $\{\phi_i(\Omega_i)\}$ incident at a vertex v in $\mathcal{BR}$, the corresponding meshes $\{\mathcal{T}_i\}$ share a corresponding vertex.

The *accuracy* of $\mathcal{T}_i$ in approximating $\phi_i(\Omega_i)$ is referred to as the *distance* between the two surfaces. Several methods can be used to measure such distance. For instance, if the triangle mesh is parametrized in the same space as the original face, an L_∞ *norm* can be used [Kle95]; another possibility is using the *Hausdorff distance* between $\phi_i(\Omega_i)$ and $\mathcal{T}_i$ [Kle96a, Coh96]. Our multiresolution boundary representation is parametric on the kind of distance used.
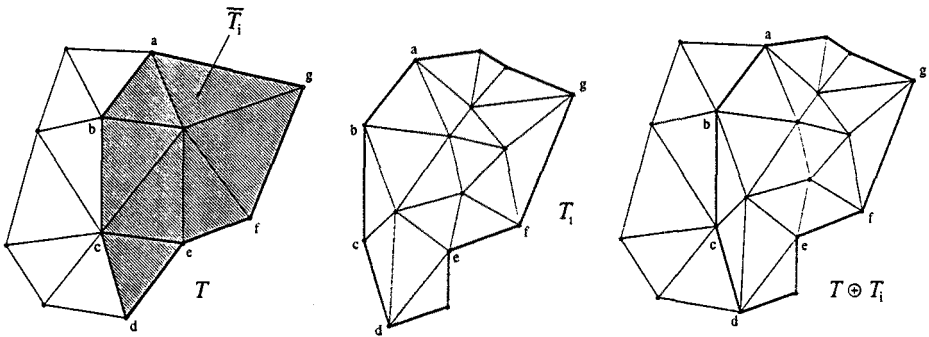
3   Multiresolution Boundary Representation

Fig. 1. Compatible meshes and their combination.

Let $\mathcal{T}$ and $\mathcal{T}_i$ be two triangle meshes, where $\mathcal{T}_i$ is a 2-manifold with boundary, and $\mathcal{T}$ may be either with or without boundary. $\mathcal{T}_i$ is said to be *compatible* over $\mathcal{T}$ if and only if:

1. the boundary of $\mathcal{T}_i$ can be decomposed into chains of edges such that for each chain $\mathcal{C}$:
   - either $\mathcal{C}$ coincides with a chain $\bar{\mathcal{C}}$ of edges in $\mathcal{T}$, or
   - $\mathcal{T}$ contains a chain $\bar{\mathcal{C}}$ of *boundary edges* having the same endpoint vertices as $\mathcal{C}$;

2. the collection $\bar{\mathcal{T}}_i$ of triangles of $\mathcal{T}$ enclosed by chains $\bar{\mathcal{C}}$ is a triangle mesh homeomorphic to $\mathcal{T}_i$ (see Fig. 1).

Provided that $\mathcal{T}_i$ is compatible over $\mathcal{T}$, the *combination* of $\mathcal{T}_i$ over $\mathcal{T}$ is a new triangle mesh, denoted by $\mathcal{T} \oplus \mathcal{T}_i$, and defined as $(\mathcal{T} \setminus \bar{\mathcal{T}}_i) \cup \mathcal{T}_i$ (see Fig. 1).
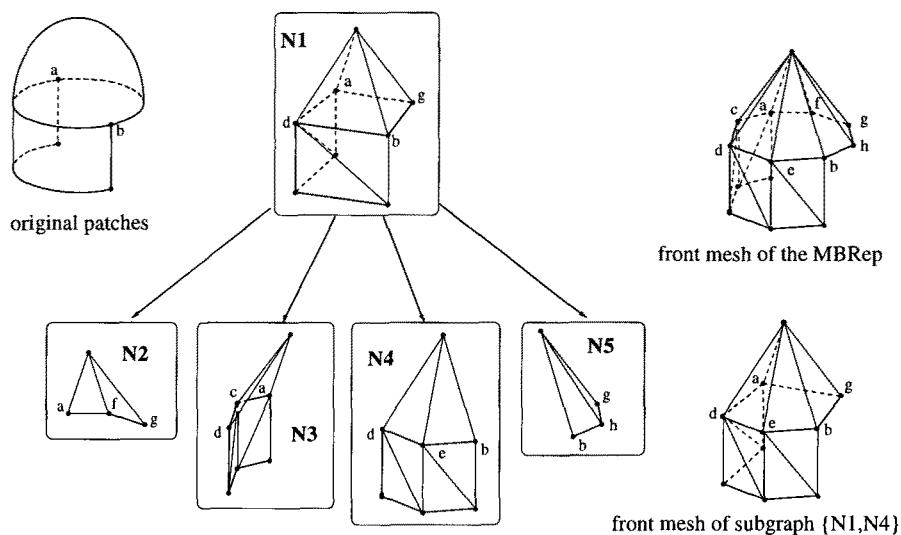

Fig. 2. An MBRep for two adjacent parametric surfaces.

A *multiresolution BRep (MBRep)* for a given generic BRep $\mathcal{BR}$ consists of a collection of triangle meshes, each characterized by a certain approximation error. Such meshes are organized into a partial order based on their compatibility, and can be combined in several ways to obtain approximations of the original BRep $\mathcal{BR}$ at different LODs. The partial order is represented as a rooted DAG $\mathcal{MB} = (\mathcal{N}, \mathcal{A})$, where each node in $\mathcal{N}$ is a triangle mesh. The class of DAGs which are MBReps is defined inductively as follows:

1. The DAG containing only one node (the root node $\mathcal{T}_0$) is an MBRep, and the associated *front mesh* is $\mathcal{T}_0$ itself.
2. Let $\mathcal{MB} = (\mathcal{N}, \mathcal{A})$ be an MBRep, and $\mathcal{T}$ be its associated front mesh. Let $\mathcal{T}_i$ be a mesh compatible over $\mathcal{T}$. Then, the DAG $\mathcal{MB}' = (\mathcal{N} \cup \{\mathcal{T}_i\}, \mathcal{A} \cup \mathcal{A}')$ is an MBRep, where $\mathcal{A}'$ contains an arc $(\mathcal{T}_j, \mathcal{T}_i)$ for each $\mathcal{T}_j \in \mathcal{N}$ such that $\mathcal{T}_j \cap \bar{\mathcal{T}}_i$ contains at least one triangle; the *front mesh* associated with $\mathcal{MB}'$ is $\mathcal{T} \oplus \mathcal{T}_i$.

It can be shown that any subgraph $\mathcal{MB}' = (\mathcal{N}', \mathcal{A}')$ of $\mathcal{MB}$ induced by a cut of the DAG (i.e., such that $\mathcal{T}_0 \in \mathcal{N}'$, and for every mesh $\mathcal{T}_i \in \mathcal{N}'$, $\mathcal{N}'$ contains the sources of all arcs ending in $\mathcal{T}_i$), is also an MBRep. The front meshes associated with the cuts of an MBRep provide representations of the original BRep at different resolutions (see Fig. 2).

Within a generic application, a LOD function $\tau : \mathbb{R}^3 \to \mathbb{R}$ is given, which defines the maximum error tolerance for the surface representation pointwise. An MBRep $\mathcal{MB}$ supports a simple and efficient algorithm which, given an arbitrary LOD function τ, generates the simplest PL approximation, made of triangles of $\mathcal{MB}$ which satisfy the given error requirements. Such approximation is the front mesh of the minimal subgraph $\mathcal{MB}'$ of $\mathcal{MB}$ induced by a cut where all the triangles have an acceptable resolution. The algorithm described in [DeF97] performs a breadth-first traversal of the DAG, starting at the root mesh, to determine the subgraph $\mathcal{MB}'$ giving the solution. Its running time is linear in the size of the visited subgraph.

4 Building a Multiresolution BRep

The construction of an MBRep relies on *incremental* methods for building piece-wise-linear approximations of a BRep. Such algorithms are based on the update of a current linear approximation: they either start from a coarse approximation and progressively refine it, or start with a fine approximation and progressively simplify it. Refinement and simplification are usually performed by inserting and discarding vertices into/from the model, respectively. An MBRep is built by creating an new node of the MBrep at each step of the incremental construction and by connecting it to the DAG constructed so far. Here, we follow an approach based on refinement. An equivalent approach based on simplification can be also adopted with small modifications.

PL approximations of a parametric surface $\phi(\Omega)$ are commonly obtained by lifting a triangulation of a polygonal approximation of the domain Ω into physical space: lifting is performed by mapping each vertex (u, v) to point $\phi(u, v)$. When several parametric surfaces compose a Brep, a main issue is ensuring matching between the linear approximations at the common edge of any pair of adjacent faces. In an MBRep, this property must be guaranteed at *any* LOD.

Our approach to the construction of an MBRep operates in two successive steps:

1. first, a multiresolution linear representation for the edges of the BRep is defined (see Fig. 3); this step involves all faces of the BRep simultaneously, thus ensuring boundary matching;
2. then, a multiresolution representation is built separately for the interior of each face, which conforms to the multiresolution representation of its edges (see Fig. 4).

Finally, the local MBReps of adjacent faces are glued together at boundary edges, thus obtaining an MBRep for the whole object.

Steps 1 and 2 are performed by iteratively insering points lying on an edge or in a face, respectively; in both cases, points to be inserted are selected according to a heuristic technique aimed at progressively reducing the approximation error: the algorithm iteratively selects the point causing the greatest approximation error. For step 1, we observe that there is no correlation between the warping - measured in parameter space - between a boundary and its polygonal

approximation, and the error - measured in physical space - between the para-
metric surface and the triangle mesh approximating it: a small warping (possibly,
zero) in paramater space may correspond to a large error in physical space, or
viceversa. We always measure the error in physical space: the error of the cur-
rent boundary approximation is evaluated based on the distance between the
piecewise image of the approximation and the image of the original boundary in
physical space.

In step 1, vertices are iteratively inserted on each edge e of the BRep to
build polygonal chains approximating e at increasing LODs. All intermediate
straight-line segments created for an edge e are stored into a multiresolution
representation organized as a tree. Let $p_1, \ldots, p_k$ be the current polygonal ap-
proximation of an edge e; the insertion of a point q on e lying between p_i and p_{i+1}
causes a node representing segment $p_i p_{i+1}$ to become the parent of two nodes
corresponding to segment $p_i q$ and $q p_{i+1}$, respectively. Each node (straight-line
segment) of the tree stores its approximation error with respect to the portion
of e having the same endpoints. For a non-leaf node, such an error occurres ex-
actly at the point whose insertion created its two children. Note that, through
back-projection in paramater space, edge linearization defines an approximation
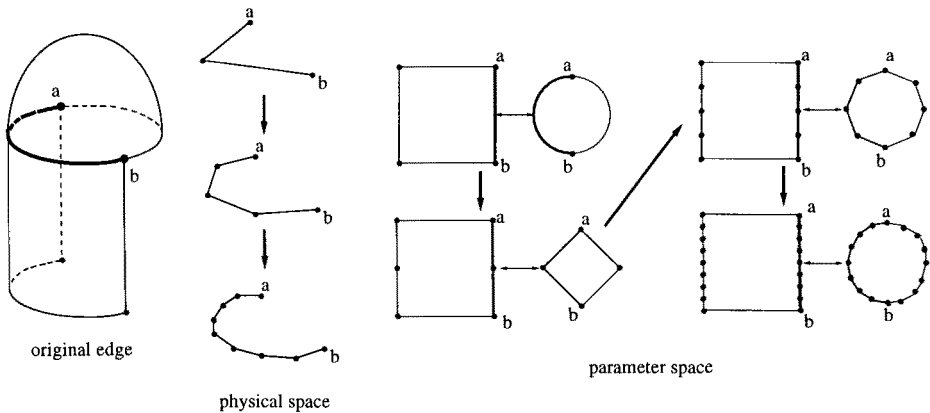of the domain of each face with a polygonal region.

Fig. 3. Iterative boundary refinement.

Step 2 consists of iteratively refining a domain triangulation for each face
separately, while using the multiresolution edge representations, built at step 1,
as constraints to ensure matching with adjacent faces. The insertion of points
lying in the interior of the current boundary discretization is interleaved with the
refinement of the domain boundary itself, driven by the tree-like representation
of each edge. At each step, we have a current polygonal approximation of the
domain, which corresponds to a cut on each tree representing an edge of the
domain. If the maximum error measured in the interior of the current polygonal

region is larger than the pre-computed error recorded in the nodes of the current cut, we insert the internal point with maximum error. Otherwise, we refine the boundary representation by replacing the tree node (segment) of maximum error with its two children: this corresponds to the insertion of a point lying on the boundary of the patch. Since the points to be inserted on edges are pre-computed, the method guarantees that both adjacent faces insert the same points, and in the same order, on their common edge.
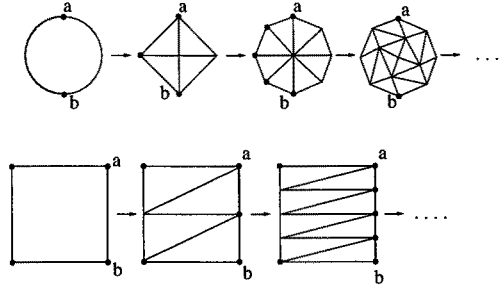
Fig. 4. Iterative face refinement.

At the insertion of a new point p, a new node, denoted by $\mathcal{T}_p$, is inserted in the MBRep of the current face: $\mathcal{T}_p$ is formed by the new triangles created in the update of the current triangulation. New DAG arcs, having $\mathcal{T}_p$ as their destination node, are also defined: the sources of such arcs are the nodes containing the triangles that have been removed from the current triangulation after the insertion of p. When the new point p lies on the boundary (i.e., p is one of the points stored in the tree representation of some edge e), then a reference to the newly created patch is recorded in the tree, to be used at step 3.

At step 3, adjacent nodes of the local MBReps built at step 2 are glued together to obtain the final MBRep of the whole object. Meshes to be glued are detected by using the tree representations of edges, which, in step 2, have been enriched by references to nodes of the local MBReps. In any tree node we find references to two adjacent meshes $\mathcal{T}_1$ and $\mathcal{T}_2$, possibly belonging to local MBReps of different faces; we glue $\mathcal{T}_1$ and $\mathcal{T}_2$ into one node of the final MBRep, which inherits incoming and outgoing arcs from both $\mathcal{T}_1$ and $\mathcal{T}_2$.

5 Concluding Remarks

We have presented a multiresolution model that supports fast extraction of triangle meshes approximating BReps with parametric faces at arbitrary level of detail, possibly variable over the surface. Major applications of this model are in real-time visualization of objects with a large number of faces, and in the production of graded meshes for the finite elements. The main idea underlying our model is that many different representations can be built from a relatively small set of atomic entities (i.e., triangular patches), and that all possible combinations

of such elements to form approximate BReps follow from their clustering into fragments, and from dependencies among such fragments. Based on the results of [Pup96], we can assert that the algorithm for extracting a LOD returns the smallest representation satisfying the LOD among all possible representations that can be built from triangles of the model. Therefore, our model achieves a sort of optimality in its result. Our model is parametric on the kind of distance adopted to measure approximation accuracy, and on the kind of refinement strategy and triangulation algorithm adopted to build it. In fact, it can be built also through a simplification algorithm that progressively discards vertices from an initial PL surface at high resolution. Our previous experience with multiresolution models for scalar fields [DeF96] suggests that, in addition to the extraction of a LOD, other interesting geometric queries, such as point location and surface intersection, can be implemented efficiently within the same framework.

References

[Coh96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, W. Wright, 1996, Simplification envelopes, *ACM Computer Graphics (SIG-GRAPH 96)*, pp.119-128.

[DeF96] L. De Floriani, E. Puppo, P. Magillo, 1996, A formal approach to multiresolution modeling, in *Theory and Practice of Geometric Modeling*, W. Straßer, R. Klein, R. Rau (eds), Springer-Verlag.

[DeF97] L. De Floriani, P. Magillo, E. Puppo, Multiresolution Representation and Reconstruction of Triangulated Surfaces, *Proc. Int. Workshop on Visual Form*, Capri, Italy, May 1997. Also published in longer version as *Technical Report* PDISI-96-26, Computer and Information Science Dept. (DISI), Univ. of Genova, Italy, 1996.

[Kle95] R. Klein, W. Straßer, 1995, Large mesh generation from boundary models with parametric face representation, *Proc. 3rd Symp. on Solid Modeling and Applications*, pp. 431-440, ACM Press, C. Hoffmann and J. Rossignac (eds).

[Kle96a] R. Klein, G. Liebich, W. Straßer, 1996, Mesh reduction with error control, *Proc. ACM Visualizations '96*, R. Yagel (ed).

[Kle96b] R. Klein, W. Straßer, 1996, Generation of multiresolution models from CAD data for real time rendering, *Theory and Practice of Geometric Modeling*, W. Straßer, R. Klein, R. Rau (eds), Springer-Verlag.

[Hop96] H. Hoppe, 1996, Progressive meshes, *ACM Computer Graphics (SIGGRAPH 96)*, pp.99-108.

[Pup96] E. Puppo, 1996, Variable resolution terrain surfaces, *Proc. Canadian Conference on Computational Geometry*, Ottawa (Canada), pp.202-210. Also published in longer version as "Variable resolution triangulations", *Technical Report* N.12/96, Istituto per la Matematica Applicata, C.N.R., Genova, Italy, 1996.

[Sch92] W.J. Schroeder, J.A. Zarge, W. Lorensen, 1992, Decimation of triangle mesh, *ACM Computer Graphics (SIGGRAPH 92)*, 26, 2, pp. 65-70.

[VRML96] *The Virtual Reality Modeling Language Specification - Version 2.0*, August 1996. Available at: http://vag.vrml.org/.