Simplicial-based techniques for multi-resolution volume visualization: an overview

Rita Borgo[1], Paolo Cignoni[1], and Roberto Scopigno[1]

Istituto di Scienza e Tecnologia dell'Informazione (ISTI), Consiglio Nazionale
delle Ricerche, via G. Moruzzi 1, 5614 Pisa, Italy.
Email: borgo, cignoni@iei.pi.cnr.it, roberto.scopigno@cnuce.cnr.it

1 Introduction

A number of approaches exist to support multi-resolution management: naïve sub sampling, wavelet techniques, methods based on hierarchical space subdivisions (e.g. octrees), methods based on simplicial decomposition. The term multi-resolution is often used to indicate either discrete or continuous level of detail (LOD) representation. We will cover mostly the second aspect, and therefore we point our attention to those methods that allow to manage selective refinements (or the inverse operation, i.e. selective coarsening) in a dynamic manner, according to the interactive requests of the application. Simplicial meshes have been often used in the visualization of volume dataset. The simplicity of the basic cell allows to easily manage isosurface extraction (field is linearly interpolated, no ambiguity) and to implement in an efficient manner direct volume rendering (DVR) solutions. Moreover, tetrahedral-based DVR solution can now be implemented using off-the-shelf graphics hardware, gaining impressive speed-ups with respect to software solutions. Therefore, simplicial decompositions have been often considered in the design of multi-resolution methods, not only because easy to render but also because they easily adapt to different shapes or to the data field structure/topology. This paper presents an overview and a comparison of the different approaches based on multi-resolution simplicial decomposition, which have been proposed in the context of volume visualization. We subdivide the existing methods in two main classes, which depend on the refinement kernel used to manage the selective refinement/coarsening: regular or irregular. Regular techniques starts from a coarse irregular base domain and apply recursive regular refinement, resulting in large meshes organized as uniform grid patches. Irregular techniques are independent from the topology of the underlying mesh and as consequence during the refinement procedure new vertices can be inserted at more convenient locations instead of predefined positions; not being forced to follow a regular subdivision scheme irregular techniques result to be more flexible and suitable to resolve complex geometric features and geometry changes.

2 Using Irregular Refinement Techniques

As introduced in the previous section, multiresolution models for generic tetrahedral meshes has evolved to manage set of level of details in a more flexible, efficient and compact way. The main idea behind these methods is to exploit, in some way, the information that can be collected during the simplification of a volume dataset. The assumption is that we use an iterative simplification algorithm that progressively reduces/refines the dataset by means of small local operations. This sequence of small modifications is organized in a structure that encodes all the temporal dependencies among themselves and makes it possible to apply backward the small modifications in a different order. In this way is possible, for example, selectively refine, or simplify, only certain portions of the domain, according to the user needs.
These techniques are therefore strongly related with multiresolution and simplification techniques for generic three-dimensional surface meshes where those ideas were firstly developed (see, e.g., [10] for a survey of surface multiresolution). For this reason in the next section we shortly review the existing techniques for the simplification of a tetrahedral complex.

2.1 Simplification of a simplicial complex

One of the first approaches to the construction of a simplified representation of a tetrahedral dataset was proposed in [3]; it exploits a basic coarse-to-fine refinement strategy, an early technique developed in the two-dimensional case and widely used for approximating natural terrains [7, 11]. An on-line algorithm for Delaunay tetrahedralization is used together with a selection criterion to refine an existing Delaunay mesh by inserting one vertex at a time. The selection strategy at each iteration is aimed to refine the tetrahedron that causes the maximum warping/error in the current approximation: this is obtained by selecting the datum v_{max} corresponding to the maximum error as a new vertex. After a point is added to the dataset, the tetrahedron that contains it is split and a sequence of flipping actions (see Fig. 1) is performed until the resulting mesh satisfies the Delaunay criterion.
This refinement procedure always converges since the number of points in V is finite; total accuracy is warranted when all of them are inserted as vertices of Σ.
This approach is limited to datasets whose domain is convex, because the result of a Delaunay tetrahedralization is always convex. We have proposed the extension of this approach in [5] to deal also with non-convex curvilinear datasets; in this case the Delaunay tetrahedralization is computed in the computational domain (i.e. the underlying grid), while its image through lifting gives the corresponding mesh in the physical domain.

The vertex insertion method described above is difficult to adapt to the case of generic non-convex irregular datasets. Major difficulties arise in finding

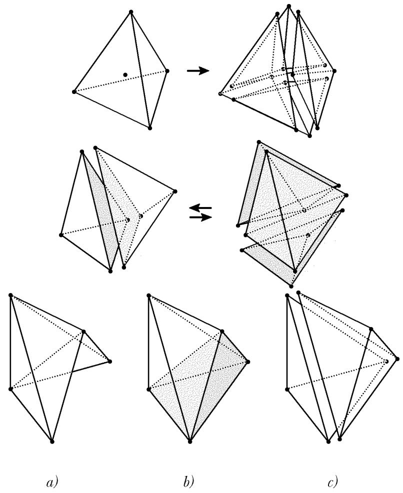


Fig. 1. A tetrahedra *split*, due to a new vertex selection and insertion in the mesh (top image); the two classes of tetrahedra *flip* actions: the 2_*to*_3 flip, which produces three cells out of two (center image); the 3_*to*_2 flip, needed when the two tetrahedra present a non convex union (a), and a third cell (b) has to be included in the flip action.

an initial coarse mesh to approximate the original domain Ω of the dataset and in the estimation of warping. Delaunay triangulation is not applicable to non-convex polyhedra; moreover even if we have an approximation of the boundary of the starting domain finding a tetrahedralization of this polyhedron, without adding new points, is an NP-complete problem [19].
Experience in the approximation of non-convex surfaces through 2D triangular meshes suggests that a decimation technique might be more appropriate to the case of non-convex irregular volume datasets (see, for example, [1,12,20]). Similarly for the 2D surface case in the decimation approach we start from the reference mesh Γ, and vertices are iteratively discarded as long as the error introduced by removing them does not exceed a given a accuracy threshold. Gross and Staadt [21] present a simplification technique based on collapsing an edge to an arbitrary interior point, and propose various cost functions to drive the collapsing process. Cignoni et al. [5] propose an algorithm based on collapsing an edge to one of its extreme vertices (called half-edge collapse see Fig. 2), in which the simplification process is driven by a combination of the geometric error introduced in simplifying the shape of the domain and of the error introduced in approximating the scalar field with fewer points. This approach has been extended in [2] by defining a framework for the unified management of the two errors (related to geometric domain and scalar field) and by proposing some techniques for an efficient evaluation and forecast of such errors.

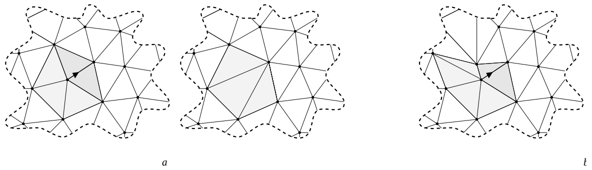


Fig. 2. Half-Edge collapse in 2D: (a) a valid collapse; (b) an inconsistent collapse.

In fact, selecting a vertex to be removed involves an estimation of the amount of field error and geometric domain error due to the removal: the criterion usually adopted is that the vertex causing the smallest increase in error/warping should be selected at each iteration. An exact estimation of the change in error and warping can be obtained by simulating deletion of all vertices in the current mesh. This would be computationally expensive, since each vertex has 24 incident tetrahedra on average. This may involve relocating many points lying inside such tetrahedra. We prefer to use heuristics to estimate apriori how a vertex removal affects error and warping. Such an estimation is computed for all vertices before decimation starts, and it is updated for a vertex each time one or more of its incident tetrahedra change.

Trotts et al. [22] perform half-edge collapse as well. They control the quality of the simplified mesh by estimating the deviation of the simplified scalar field from the original one, and by predicting increase in deviation caused by a collapse. They also provide a mechanism to bound the deformation of the domain boundary.

Another approach for the simplification of generic simplicial complexes, called Progressive Simplicial Complex (PSC), has been proposed by Popovic and Hoppe [16], as an extension of the Progressive Meshes (PM) model [12]. The PM models are based on the simplification of a mesh with a sequence of edge-collapse transformations, this sequence of operations is encoded in the PM structure. Given a PM, a mesh can be reconstructed by applying, in the right order, a series of vertex-split transformations (the reverse of edge-collapse). The PSC codifies in a similar manner the sequence of more general edge-collapse transformations. It should be noted that while the PSC are quite general, they have been conceived for the management of possibly degenerated 2D surfaces rather than simplicial complexes representing a volume dataset, so the conditions of legality of a sequence of vertex-split operation of a generic complex are not specified, and the problem of evaluating the approximation error introduced in the volume field representation is not considered.

2.2 From simplification to Multiresolution Models

In this section we introduce the framework of multiresolution simplicial models (MSM) introduced by De Floriani, Puppo and Magillo [8] as a multidimensional extension of the two dimensional structure described by Puppo in [17].

Definitions Let S be a finite set. A partial order on S is an antisymmetric and transitive relation $<$ on its elements. A pair $(S,<)$ is called a *partially ordered set (poset)*. For every $s, s' \in S$ with $s <' s$ we mean that $s < s'$ and $\nexists s''$ such that $s < s'' < s'$. A subset $S' \subseteq S$ is called a *lower set* if $\forall s' \in S', s < s' \Rightarrow s \in S'$. Intuitively S' is a lower set if it contains all the elements that precede each of its elements. The algebraic structure of a poset $(S,<)$ can be described by a direct acyclic graph (DAG), where nodes represent elements of S and arcs encode the $<'$ relation. For any $s \in S$ the set $S_s = \{s' \in S | s' \leq s\}$ is the smallest lower set containing s and it is called the *down-closure* of s. The *sub-closure* of s is defined as $S_s^- = S_s \backslash \{s\}$.
We call any finite set of d-simplexes in $\mathbb{E}^n$ a *d-set*; a regular d-simplicial complex Σ is completely characterized by the collection of its d-simplexes i.e. by the d-set associated with Σ.
When managing a collection of representation of the same complex at different resolutions, as done in the previous section, we need a measure of the error we commit. With $\mu(\sigma)$ we denote a function $\mu : \Sigma \rightarrow [0,1]$ measuring this error, $\mu(0)$ means exact representation. With $\mu(\Sigma)$ we denote the maximum error among all the tetrahedra of Σ: $\max_{\sigma \in \Sigma} \mu(\sigma)$. **Operators.** We

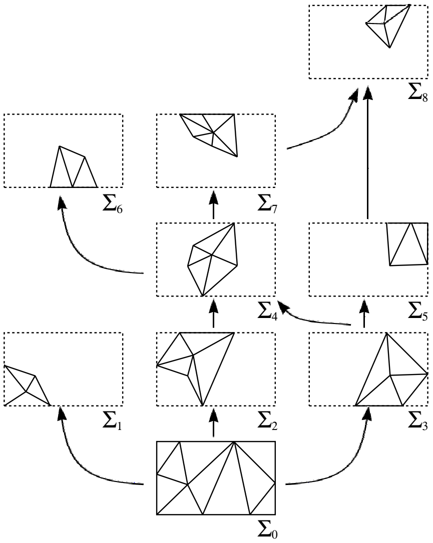


Fig. 3. The DAG describing a simple two-dimensional MSM.

define two operators on d-sets: the interference operator $\otimes$ and the combination operator $\oplus$. Both operators take two d-sets as arguments and produce a

d-set. The *interference* operator of two d-sets is defined as:

$$\Sigma_i \otimes \Sigma_j = \{\sigma \in \Sigma_i | \exists \sigma' \in \Sigma_j, i(\sigma) \cap \sigma' \neq \emptyset\}$$

In other words, the interference of two d-sets Σ_i, Σ_j is the set of the simplexes of Σ_i that have a proper intersection with some simplexes of Σ_j.
The *combination* operator of two d-set is defined as:

$$\Sigma_i \oplus \Sigma_j = \Sigma_i \backslash (\Sigma_i \otimes \Sigma_j) \cup \Sigma_j$$

In other words, the combination of two d sets Σ_i, Σ_j is the set of the simplexes that can be obtained by adding to Σ_j the simplexes of Σ_i not interfering with Σ_j.
If $\Sigma_i \oplus \Sigma_j$ is a d-simplicial complex and $\Delta(\Sigma_i \oplus \Sigma_j) = \Delta(\Sigma_i) \cup \Delta(\Sigma_j)$, then the complex Σ_j it is said to be *compatible* over Σ_i.
Let $[\Sigma_0, \ldots, \Sigma_k]$ be a sequence of d-complexes. the *combination* $\oplus_{i=0}^{k} \Sigma_k$ is defined as:

- if $k = 0$ then $\oplus_{i=0}^{0} \Sigma_k = \Sigma_0$
- if $k > 1$ then $\oplus_{i=0}^{k} \Sigma_k = (\oplus_{i=0}^{k-1} \Sigma_k) \oplus \Sigma_k$

Multiresolution Simplicial Model A Multiresolution Simplicial Model (MSM) on Ω is a poset $(\mathcal{S}, <)$, where $S = \{\Sigma_0, \ldots, \Sigma_h\}$ is a set of d-complexes and $<$ is a partial order on $\mathcal{S}$ satisfying the following conditions:

1. $\Delta(\Sigma_0) = \Omega$,
2. $\forall i, j = 0..h, i \neq j,$,
   a) $\Sigma_i <' \Sigma_j \Rightarrow \Sigma_i \otimes \Sigma_j \neq \emptyset$
   b) $\Sigma_i \otimes \Sigma_j \neq \emptyset \Rightarrow \Sigma_i$ is in relation with Σ_j
3. the sequence $[\Sigma_0, \ldots, \Sigma_h]$ of all complexes of $\mathcal{S}$ defines a consistent order w.r.t. relation $<$ and $[\Sigma_0, \ldots, \Sigma_h]$ is a compatible sequence.

The meaning of the second condition becomes clearer if we consider the DAG encoding relation $<'$ for the set $\mathcal{S}$:

- if two complexes Σ_i and Σ_j are connected by an arc, they are interfering;
- if two complexes Σ_i and Σ_j interfere then they are connected through a path.

The elements of $\mathcal{S}$ are called components or fragments; intuitively the fragments describe a portion of the domain Ω at a certain resolution. For example, if we think to an iterative refinement procedure on a simplicial mesh, the set of simplexes, derived from the substitution of a complex with a more refined one, can be considered as a fragment that is combined over the existing complex. Combining a lower set of $\mathcal{S}$ give us a complete description of Ω at various resolutions.
The following properties holds for MSM's:

Lemma 1. Σ_0 *is unique minimum element of* $(\mathcal{S}, <)$.

In other words Σ_0 is the starting coarsest complex. Given any subset $\mathcal{S}' \subseteq \mathcal{S}$ the total order of its elements consistent with the sequence $[\Sigma_0, \ldots, \Sigma_h]$ is called the *default order* of $\mathcal{S}'$.

Lemma 2. *The default order of any lower set $\mathcal{S}' \subseteq \mathcal{S}$ is a consistent order.*

Lemma 3. *In a MSM $(\mathcal{S}, <)$, the combination of a lower set $\mathcal{S}' \subseteq \mathcal{S}$ is independent of the specific consistent order.*

Since the combination obtained from any consistent order is unique, it will simply called the combination of $\mathcal{S}'$ and denoted with $\oplus\mathcal{S}$. Let Σ_i be a component of $(\mathcal{S}, <)$; the combination of the subclosure $\mathcal{S}^-_\Sigma$ is called the *support* of Σ_i; the set of the simplices of the support that are interfering with Σ_i, $(\oplus\mathcal{S}^-_\Sigma) \otimes \Sigma_i$is called the *floor* of Σ_i.

The following definitions permit us to consider a particular class of MSM's where the order relations provide control over the size in terms of number of simplexes. A MSM $(\mathcal{S}, <)$ is *increasing* if and only if for every pair of lower sets $\mathcal{S}', \mathcal{S}''$ holds: $(\mathcal{S}' \subset \mathcal{S}'' \Rightarrow |\oplus\mathcal{S}'| < |\oplus\mathcal{S}''|$. Similarly is defined the concept of *decreasing* MSM; an increasing or decreasing MSM is said *monotone*. In other words a MSM is increasing (decreasing) if and only if the size of each fragment is larger (smaller) than the size of its floor.

In Figure 3 is shown a simple multiresolution simplicial model for the two dimensional case; the arrows in the figure correspond to the relation $<'$. The fragments of a MSM can be used to build different triangulations of the domain Ω through the paste operator. The intuitive meaning of the $<$ relation is of dependence between the pasting of the fragments: if we use a fragment Σ_i in a triangulation then all the fragments $\Sigma_j < \Sigma_i$ must also be used.

In Figure 4 is shown the triangulation resulting from the pasting/combination of a subset $\mathcal{S}'$ of fragments in a consistent order $\Sigma_0, \Sigma_2, \Sigma_3, \Sigma_4, \Sigma_7$; note that any other consistent order of pasting of $\mathcal{S}'$ (like $\Sigma_0, \Sigma_3, \Sigma_2, \Sigma_4, \Sigma_7$) builds the same triangulations.

2.3 MSM for Volume Visualization

Each simplification algorithm described in Section 2.1 can be used to build a MSM. Both a decimation or a refinement algorithm for simplifying a tetrahedral complex can be regarded as producing an "historical" sequence of tetrahedra, namely all tetrahedra that appear in the current mesh Σ during its construction. An historical sequence can be also viewed as the sequence of all subdivisions of the whole domain that are obtained through changes, or as an initial subdivision plus a sequence of fragments reflecting the local changes iteratively done to the mesh, i.e. subdivisions of portions of the domain, which can be partially overlapping and are pasted one above the other to update the existing structure.

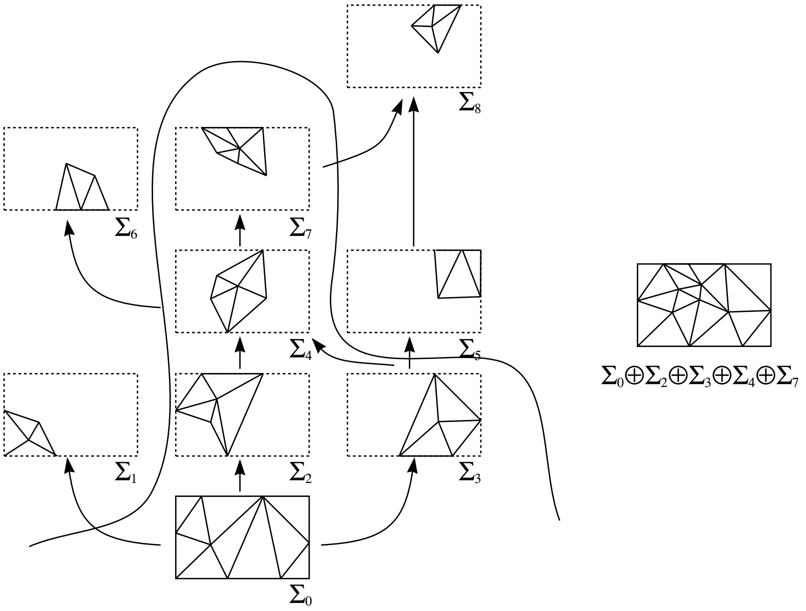


Fig. 4. A subset $\mathcal{S}' \subset \mathcal{S}$ combined in a consistent order builds a triangulation

For example, if we follow the *refinement* heuristics, the minimum of the poset is the starting coarse triangulation; when we insert a new point v_i in the complex the new tetrahedra that are built form a new fragment Σ_i; the floor of this fragment is constituted by the tetrahedra that were destroyed by the insertion of v_i.

Following the MSM all these fragments, represented by a tetrahedral complex covering a small part of the whole domain Ω, are arranged in a poset where the order relation between fragments is dependent on their interferences in 3D space. The minimum fragment Σ_0, the coarsest representation of our mesh, has an empty floor. Similarly all the triangles on the top of $\mathcal{S}$, representing the dataset at its full resolution, have no upper fragments.

A simple data structure to encode a generic MSM was presented in [6]; in [4] a much more compact data structure has been proposed for a three-dimensional tetrahedral MSM built based on a sequence of general edge collapses; this structure, customized to the needs of volume visualization, requires three times less storage space with respect to a simple indexed data structure encoding the original mesh at full resolution, and 5.5 times less space than a data structure for the original mesh encoding both connectivity and adjacency information (as required, e.g., by direct volume rendering algorithms based on cell projection).

2.4 Extracting a variable resolution model

The algorithm for the extraction of a variable resolution model here presented are a straightforward extension to tetrahedral complexes of the one presented in [17]. We suppose that our MSM is monotone, to extract a variable resolution model we need a boolean acceptance function $c(\sigma)$ in order to decide, for a given tetrahedron, if its accuracy is sufficient or if we need to further refine that part of the domain.

The algorithm for the extraction of a variable resolution model tries to incrementally build the desired solution by adding new fragments to the current solution. The algorithm is based on a breadth-first traversal of the DAG representing the MSM. The traversal starts from the coarsest fragment Σ_0, root of the DAG, and fragments above the current solution are progressively traversed and marked. The current solution is maintained as a list of tetrahedra Σ_{Out}. For each fragment Σ that we encounter in the traversal of the tree, the following two loops are executed:

- we search for fragments before Σ, still not visited and, if found, they are added to the traversal queue Q. All the fragments before Σ can be found by checking, for each tetrahedron $\sigma \in \Sigma$, if the corresponding lower fragment Lower(σ), has been marked.
- for each tetrahedron $\sigma \in \Sigma$, if it satisfies the acceptance function $c(\sigma)$ then σ is added to the current solution, else we add the upper fragment of σ to the traversal queue Q and mark it to be removed from the solution.

The correctness of this algorithm has been proved in [6].

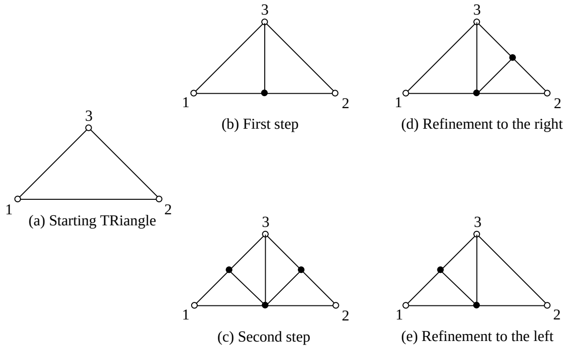
3 Using Regular Refinement Techniques

Within regular subdivision the world is divided if not into equal sized voxel at least into regular shaped entity following a recurrent pattern. Simple mathematical rules withhold the basis of a subdivision scheme that is applied in a recursive manner.The adoption of a predefined rule to perform the subdivision introduces some constrains on the topology of the local region to be refined/simplified, and in effect this type of approach adapts just to regular dataset. Initially, this approach has been proposed for terrain data represented by gridded [9], and then been extended to represent voxels dataset with a simplicial decomposition constructed via a regular subdivision rule [13, 14, 23]. Essentially the refinement heuristic consists of a uniform recursive subdivision of the volume data, driven by a set of simple mathematical rules that guarantee a progressive update of the accuracy of the intermediate representation in an error-controlled manner. The subdivision process in general starts by subdividing the bounding box of the volume (i.e. a single hexahedral cell) in simplicial cells. The subdivision proceeds recursively and can be described either as a vertex-adding process or, analogously, as a cell subdivision process. Each step picks up a vertex from the original volume and divides the cell containing it (or a group of adjacent cells) in two (or more) simplicial cells. Because of the regular and hence predictable parametric structure of the regular refinement, these techniques always generate meshes with bijective mapping between the coarsest levels and the finest levels. Therefore, going from fine/coarse to coarse/fine levels correspond in executing a regular fusion/subdivision of simplices allowing for smooth and continuous changes between different levels of details. Recursive subdivision schemes automatically achieve hierarchical multi-resolution representations, hence, regular refinement techniques guaranteeing an almost everywhere regular structure allows for efficient tree based data structures organization with higher performances on modern processors. The beauty of regular refinement techniques basically lies in their elegant mathematical formulation and in the simplicity of the rules for generating different representations. In the next paragraph we introduce in detail some of the main contribution in the field of regular refinement techniques.

3.1 Refinement of Tetrahedral Grids

Regular subdivision scheme allows for the organization of the space of the data in a hierarchical manner. As the subdivision proceeds a more detailed version of the surface is produced causing each level of the hierarchy to hold implicitly a multi-resolution organization of the dataset itself. A hierarchical organization of the dataset allows as primary consequence for a selective traversal of the representation and indeed fast construction of adaptive levels of detail. For what concern volume visualization not many attempt has been tried to accomplish good balance between efficient subdivision techniques and easiness of implementation. Techniques for tetrahedra bisection mostly recall similar patterns of subdivision, for this reason we have individuated three principal contribution in the area of regular volume refinement techniques based on simplicial complexes.
As first work we introduce the one from Plaza and Carey [15] in which refinement and conformity are protracted together.

Definitions *Skeleton.* Let Ω be a bounded set in $R^n$with non empty interior and τ an n-simplicial mesh of Ω. The set $skt(\tau) = \{\partial(t_i) : t_i \in \tau\}$ will be called *skeleton* or (n-1)-skeleton of τ with ∂ being the boundary in R^n. The skeleton of a tetrahedralization in 3D dimensions is comprised of the faces of the tetrahedra, in 2D is the set of edges of the triangles and in 1D it is the set of the points which define the partition into segments.
Edge bisection. Let $S =< X_0, X_1, \ldots, X_n >$ be an n-simplex in R^n, with edge $< X_k, X_{k'} >$ having midpoint $A = (X_k + X_{k'})/2$. Then two new simplices $S_1 =< X_0, \ldots, X_{k-1}, A, X_{k+1}, \ldots, X_{k'}, \ldots, X_n >$ and $S_2 =<$



(a) Starting TRiangle

(b) First step

(d) Refinement to the right

(c) Second step

(e) Refinement to the left

Fig. 5. 2D version of the 4T Rivara's algorithm

$X_0, \ldots, X_k, \ldots, X_{k'-1}, A, X_{k'+1}, \ldots, X_n >$ may be formed such that the interiors are disjoint and $S = S_1 \cup S_2$. This defines a subdivision of S by edge bisection.

Refinement of Tetrahedral Grids: Plaza and Carey Technique Plaza and Carey's starting point is a modified version of the 4T algorithm from M.C. Rivara [18]. In 2D such techniques can be seen as triangle refinement based on a longest edge bisection criteria and works as follows.
Considering a generic triangle t, the 4T algorithm divides t in 4 new triangles as in Fig. 5. The subdivision consists of two steps showed respectively in Fig. 5b-c. In the first step a selection of the longest edge of the triangle t is performed and the triangle is bisected in two halves by the line joining the bisection point and the "opposite vertex". The second step proceeds with the subdivision of the triangles produced in step 1 following the same bisection-criteria as for the original triangle.

The 4T algorithm is used by Plaza as refining technique of what is called *"skeleton"* of the mesh to be refined. A point worth noting is that to guarantee the conformity of the mesh Plaza induces the subdivision of all simplices that contain the bisected edge.
The 3D version of the 4T algorithm applies to tetrahedral meshes in the same way the 4T algorithm applies to triangular meshes.
Let $T^n = \{\tau_1 < \tau_2 < \ldots < \tau_n\}$ be a sequence of nested three dimensional grids with τ_1 be the initial mesh and τ_n the finest mesh in the sequence, τ_{n+1}, with $\tau_n < \tau_{n+1}$, is generated applying a 2-dimensional refinement algorithm to the skeleton of τ_n. The algorithm works as follows:

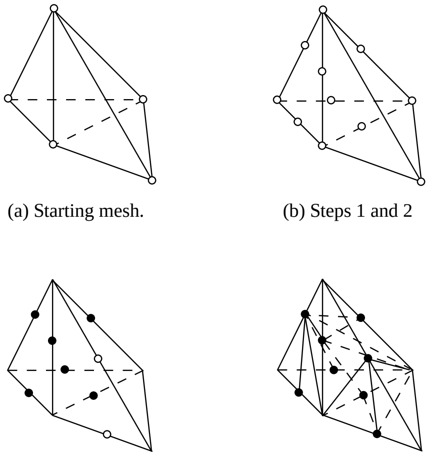
- Step 1: all the $t \in \tau_n$ that need to be refined are selected;

- Step 2: a node is added at the midpoint of each edge of each selected t;

- Step 3: the conformity of each $t \in \tau_n$ is checked. If a t is non-conforming a node is added at the midpoint of the longest-edge of each non-conforming face of t and also to the midpoint of the longest edge of t;

- Step 4: each $t \in skt\tau_n$ is subdivided by the 4T algorithm of Rivara;

- Step 5: each $t \in \tau_n$ is properly subdivided,


(a) Starting mesh.

(b) Steps 1 and 2

Fig. 6. Application of the 3D algorithm

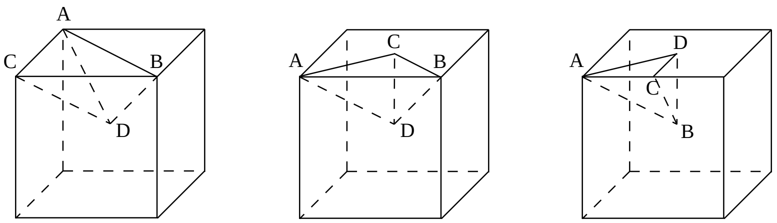
the result produced can be seen in 6e.

In Plaza's algorithm the most difficult case is represented by the presence of regular faces or elements, in this case further heuristics need to be formalized: the first edge in which has been introduced a node for division is chosen as the longest edge of the face/element. This kind of choice avoids breaks in the refinement area and limits the growth of the subdivision because of required conformity. The five steps of the algorithm are showed in Fig. 6.

As second contribution we decided to introduce a work from Zhou et al. [23]. Quite different from Plaza's algorithm, Zhou work still adopts a longest edge bisection criteria approach to perform the subdivision of the volume.

Refinement of Tetrahedral Grids: Zhou, Chen and Kaufman Technique Starting point of Zhou's algorithm is the bounding box of the volume seen as a cube with faces parallel to a coordinate plane. The bounding box is subdivided into tetrahedra. The center of the box is added to the volume and connected to all the 8 vertices forming 6 pyramids. Each pyramid can be easily divided into two tetrahedra by connecting the diagonal of the base face generating 12 tetrahedra. After this initial step the algorithm proceed as a regular scheme for tetrahedra subdivision. Each produced tetrahedra is recursively subdivided adding a midpoint to its longest edge and connecting the new vertex with the opposite one lying on the "base" face. The tetrahedra generated by the subdivision can be grouped in three main classes:

- Class 1 cell: there is only one face parallel to a coordinate plane and there exists only one edge l of the face not parallel to any coordinate axis;
- Class 2 cell: there is only one face parallel to a coordinate plane and there exists only one edge l of the face parallel to a coordinate axis;
- Class 3 cell: there are only two faces parallel to coordinate planes (the edge that does not belong to any of these two faces is denoted by l).



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 7. Types of tetrahedra generated by the subdivision.

Figure 7 depicts the three classes just described. Edge AB of tetrahedron $ABCD$ correspond to edge l. In each different case the midpoint of corresponding edge l is selected as the dividing point. In Fig. 8 is shown the entire subdivision process. After a few steps in which the cube subdivision is performed the cycle ends with 12 tetrahedra all belonging to Class 1 (Fig. 8c), that subdivided during step 3 produce tetrahedra belonging to

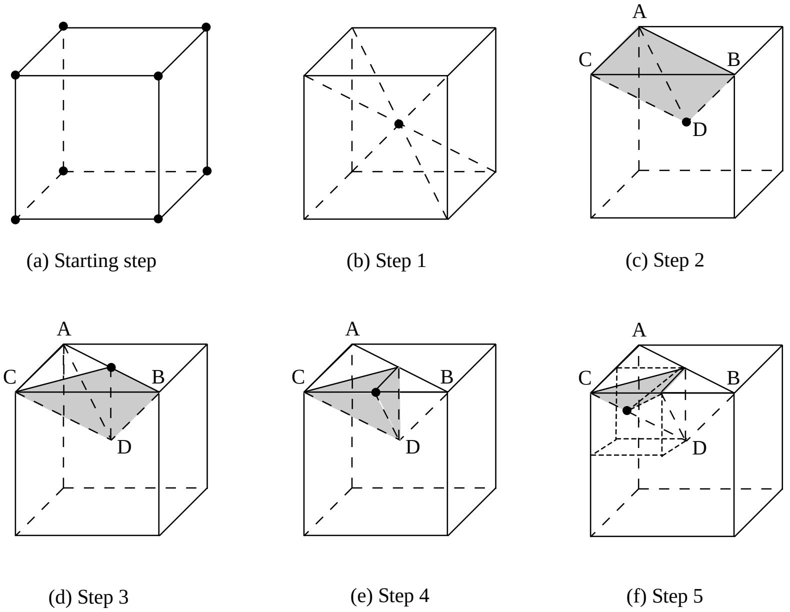
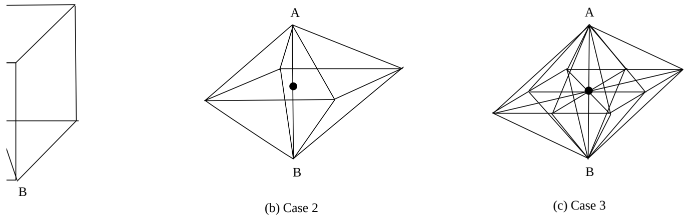


Fig. 8. Cube subdivision by Zhou et al.

Class 2 (Fig. 8d) that subdivided again during step 4 of the subdivision process produce tetrahedra belonging to Class 3 (Fig. 8e). After the subdivision of Class 3 tetrahedra the configuration recursively returns to Step 2. It is worth noting that each type of tetrahedra belongs to a different step of the subdivision process and that the overall process has a cyclic behavior.

Third and last contribution is the one from Pascucci and Borgo that introduces a subdivision schema recalling the one from Zhou et al. but that differs deeply in the subdivision structure.

Refinement of Tetrahedral Grids: Pascucci Technique As for Zhou et al. the starting point of Pascucci [13] techniques is the volume bounding box, a cubic cell, divided in function of its center. The center of the box is added to the volume and connected to all the 8 vertices forming 6 pyramids. Pascucci do not divides the cube into tetrahedra instead he considers the new entity formed by the six pyramids, generated by the subdivision, with the pyramids generated by adjacent subdivided cubes (Fig. 10b). This new entity corresponds to an hexahedral shaped cell characterized by a center that corresponds to the center of one of the faces of the cube. The center of the hexahedra is added to the volume and connected to all the 6 vertices dividing the cell into six tetrahedra. This six tetrahedra correspond each to the eighth part of different octahedral shaped cells whose centers correspond



(b) Case 2

(c) Case 3

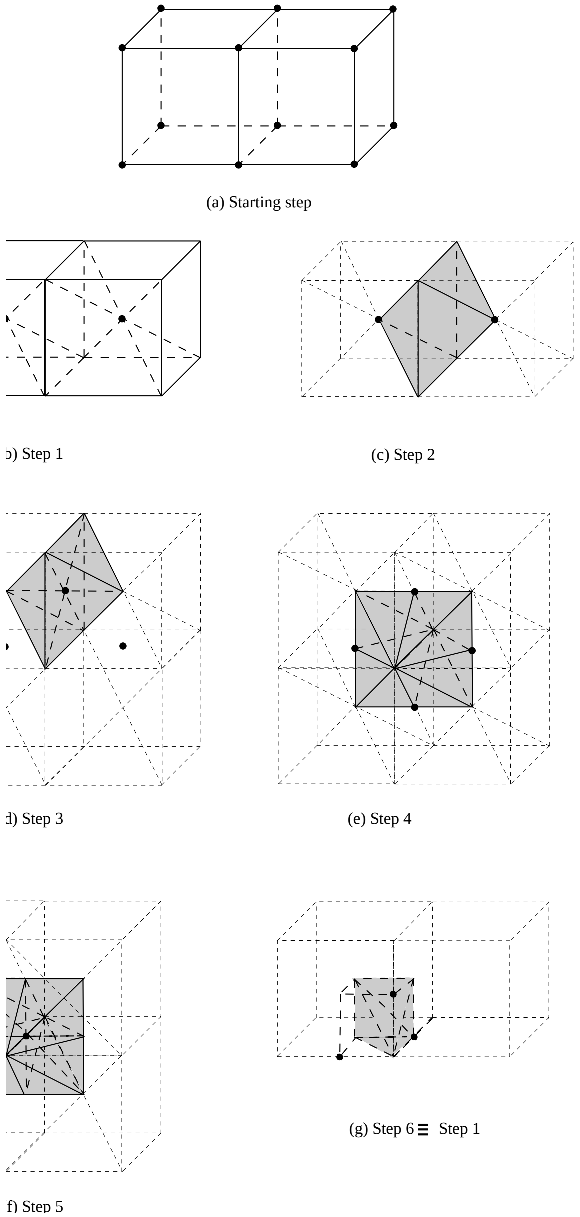
Fig. 9. Cell types.

to the midpoints of the cube edges (Fig. 10c). The cell generated by the subdivision can be grouped in three main classes:

- Class 1 cell: the cell is cube shaped;
- Class 2 cell: the cell is hexahedral shaped and can be oriented along x, y or z axis;
- Class 3 cell: the cell is octahedral shaped and can be oriented along x, y or z axis;.

Figure 9 depicts the three classes just described. Each cell is characterized by a center that correspond to the midpoint of edge AB, longest edge of the cell. The midpoint of AB is always selected as the dividing point and correspond respectively to the center of the cube (Step 1 Fig. 9b), to the center of the cube faces (Step 2 Fig. 9d) and to the midpoint of the cube edges (Step 3 Fig. 9f). In Fig. 10 is shown the entire subdivision process summarized in three steps. The subdivision of Class 1 cells (Fig. 10a) produces cells belonging to Class 2 (Fig. 10b) that subdivided again produce tetrahedra belonging to Class 3 (Fig. 10c). After the subdivision of Class 3 cells the configuration recursively returns to Step 1.
Like Zhou subdivision each type of cell belongs to a different step of the subdivision process and the overall process has a cyclic behavior. The difference between the two approaches relies mainly in the cell shape and in the way a cell can be identified. Point worth noting of Pascucci approach is in the fact that to characterize a cell (i.e. type, orientation and refinement level it belongs to) he just need to know its center.

All the approaches presented work under a politics of per-vertex adding process. The subdivision process is always recursive and ends when no more refinement is needed or if all the vertexes belonging to the mesh has been added. As mentioned before the regularity of the processes allows for an organization of the results in hierarchical data structure suitable of efficient LOD's extraction and adaptive traversal.



(a) Starting step

b) Step 1

(c) Step 2

d) Step 3

(e) Step 4

(g) Step 6 ≡ Step 1

f) Step 5

Fig. 10. Cube subdivision by Pascucci et al.

4 Comparative Evaluation

A comparative evaluation of the two techniques is not straightforward. Both approaches present strength and weakness on different sides. To perform an effective evaluation we have then identified six main point of interest that visualization techniques are required to satisfy in the best possible way.

- *adaptivity*: with adaptivity we indicate how well each approach is able to produce different resolution representation of the initial model, which either can be: (a) give a good approximation of the dataset shape or (b) give a good representation of the field distribution encoded in the initial pointset. Irregular techniques allow for better adaptivity due mainly to the freedom in the choice of the best suitable pattern for refinement purpose. In passing form high resolution to low resolution, and vice-versa, the area on which irregular techniques needs to operate the update can be restricted to a quite small number of tetrahedra. Regular techniques instead need to propagate the change to a wider area because in most cases those type of techniques allows for at most one level of difference in the refinement of adjacent cells.
- *flexibility*: with flexibility we indicate to what extent each method allows to change the parameters used to define the simplification criterion. This is a critical feature for what concern the visualization of datasets with multiple field values like, for example dataset coming from fluidodynamic simulations, in which more than parameters (pressure, temperature) influences the error-based refinement. Regular techniques suit better the possibility of changing refinement constraints at dynamic level. The regular pattern on which relies the overall refinement hierarchy allows if needed for fast update of the hierarchy itself. In irregular techniques instead the construction of the refinement "patterns" is error-driven so a changing in the refinement parameters, at dynamic time, requires to start the all refinement process from scratch.
- *data access*: with data access we indicate how well data can be organized in memory to allow for efficient data access. Irregular techniques require random access to the data on disk, on the contrary regular techniques can be easily organized in memory in such a way that guarantees locality in memory access especially during refinement actions known only at runtime.
- *space complexity*: with space complexity we indicate how well data can be organized in memory to allow for efficient storing. The adoption of a regular schema obviously allows for an efficient representation of the schema itself. With regular techniques it is possible to avoid an explicit representation of the topology of the mesh or piece of mesh under refinement and often also of the . Irregular techniques requires instead an explicit representation also of the interdependency relationships that exists between the elements that make up the multiresolution structure.

	Adaptivity	**Flexibility**	**Data Access**	**Space Complexity**
Regular	bad	bad	good	good
Irregular	good	good	bad	bad

Two more point of analysis worth to be mentioned would be the implementation easiness and adaptability to different type of datasets. Referring to easiness of implementation it would seem that implementation and debugging of techniques based on regular approaches should be preferable nevertheless reality shows to be different. As the depth of the refinement augment, even on dataset Mega sized, the structural complexity of the refinement structure grows exponentially both for the regular and irregular case making quite hard any debugging attempt in both of the approaches. For what concern adaptability irregular techniques show to be more flexible. Regular techniques in fact performs well only on regular datasets and result to be difficult to generalize to irregular ones. Re-sampling an irregular dataset on a regular grid is not convenient for several reason. It is common that the ratio between the sizes of the smallest and the largest cell in such a dataset is about $1 : 10,000$, and large cell may lay in any position across the domain. This would require an adaptive re-sampling strategy that often makes it hard using standard techniques peculiar to the regular case. Second most irregular datasets have non-convex domains and the regular grid should properly contain the original data domain. In this case the field has good chances to have a sharp discontinuity cross the boundary of such domain since it is unknown outside the boundary.

5 Conclusions

In this work we have addressed the problem of efficiently managing large volume datasets using multiresolution structure based on regular and irregular techniques. We have presented the main contribution currently present in literature and analyzed strength and weakness of both the approaches. We have seen how irregular techniques, at expense of a more complex structure, suit better approximation and refinement of any kind of dataset while regular techniques mainly allows for efficient refinement of regular datasets. On their side regular techniques deals better with all the issues arising from the necessity of working in out-of-core. The adoption of regular subdivision pattern allows for efficient data organization to improve memory occupancy and efficient access policy to secondary memory properties of prime importance in the visualization of very large datasets. Visualization results from both techniques are considerable even if irregular techniques perform better on adaptive refinement while regular techniques deals better with change of critical refinement constraint. Both approaches have been resulted valid for specific kind of problems it is still not possible to formulate a unique judgement of quality that could prefer one approach against the other neither this has ever been primary focus of this work.

References

1. A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, *Multiresolution decimation based on global error*, The Visual Computer **13** (1997), no. 5, 228–246.
2. P. Cignoni, D. Costanza, C. Montani, C. Rocchini, and R. Scopigno, *Simplification of tetrahedral volume with accurate error evaluation*, Proceedings IEEE Visualization'00, IEEE Press, 2000, pp. 85–92.
3. P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno, *Multiresolution modeling and rendering of volume data based on simplicial complexes*, Proceedings of 1994 Symposium on Volume Visualization, ACM Press, October 17-18 1994, pp. 19–26.
4. P. Cignoni, Paola Magillo, Leila De Floriani, Enrico Puppo, and R.Scopigno, *Memory-efficient selective refinement on unstructured tetrahedral meshes for volume visualization*, IEEE Transactions on Visualization and Computer Graphics **8** (2002), no. 3, To appear.
5. P. Cignoni, C. Montani, E. Puppo, and R. Scopigno, *Multiresolution modeling and visualization of volume data*, IEEE Transactions on Visualization and Computer Graphics **3** (1997), no. 4, 352–369.
6. L. De Floriani, P. Magillo, and E. Puppo, *Building and traversing a surface at variable resolution*, Proceedings IEEE Visualization 97 (Phoenix, AZ (USA)), October 1997, pp. 103–110.
7. L. De Floriani and E. Puppo, *Hierarchical triangulation for multiresolution surface description*, ACM Transactions on Graphics **14** (1995), no. 4, 363–411.
8. L. De Floriani, E. Puppo, and P. Magillo, *A formal approach to multiresolution modeling*, Theory and Practice of Geometric Modeling (R. Klein, W. Straßer, and R. Rau, eds.), Springer-Velrag, 1997, (to appear).
9. M. Duchaineau, Murray Wolinshy, David E. Sigeti, Mark C. Miller, Charles Aldrich, and Mark B. Mineev-Weinstein, *ROAMing terrain: Real-time optimally adapting meshes*, Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97) (Los Alamitos) (Roni Yagel and Hans Hagen, eds.), IEEE Computer Society Press, October 19–24 1997, pp. 81–88.
10. L. De Floriani, E. Puppo, and R. Scopigno, *Level-of-detail in surface and volume modeling*, IEEE Visualization '98, 1998, Tutorial Notes #6.
11. R.J. Fowler and J.J. Little, *Automatic extraction of irregular network digital terrain models*, ACM Computer Graphics (Siggraph '79 Proc.) **13** (1979), no. 3, 199–207.
12. H. Hoppe, *Progressive meshes*, Proceedings of SIGGRAPH '96 (1996), 99–108.
13. V. Pascucci, *Slow growing subdivision (sgs in any dimension: towards removing the curse of dimensionality*, (2002).
14. V. Pascucci and C. L. Bajaj, *Time critical isosurface refinement and smoothing*, Proceedings of the 2000 IEEE Symposium on Volume visualization 2000 (T. Ertl, B. Hamann, and A. Varshney, eds.), IEEE Computer Society Technical Committe on Computer Graphics, 2000.
15. A. Plaza and G.F. Carey, *About local refinement of tetrahedral grids based on local bisection*, 5th International Meshing Roundtable (1996), 123–136.
16. J. Popovic and H. Hoppe, *Progressive simplicial complexes*, ACM Computer Graphics Proc., Annual Conference Series, (SIGGRAPH 97), 1997, pp. 217–224.

17. E. Puppo, *Variable resolution terrain surfaces*, Proceedings Eight Canadian Conference on Computational Geometry, Ottawa, Canada, August 12-15 1996, pp. 202–210.
18. María-Cecilia Rivara, *Mesh refinement processes based on the generalized bisection of simplices*, SIAM Journal on Numerical Analysis **21** (1984), no. 3, 604–613. MR 85i:65159
19. J. Ruppert and R. Seidel, *On the difficulty of triangulating three-dimensional non-convex polyhedra*, Discrete Comput. Geom. **7** (1992), 227–253.
20. W.J. Schroeder, J.A. Zarge, and W.E. Lorensen, *Decimation of triangle meshes*, ACM Computer Graphics (SIGGRAPH 92 Proceedings) (Edwin E. Catmull, ed.), vol. 26, July 1992, pp. 65–70.
21. O.G. Staadt and M.H. Gross, *Progressive tetrahedralizations*, Proceedings IEEE Visualization '98, IEEE, 1998, pp. 397–402 (en).
22. I.J. Trotts, B. Hamann, and K.I. Joy, *Simplification of tetrahedral meshes with error bounds*, IEEE Transactions on Visualization and Computer Graphics **5** (1999), no. 3, 224–237.
23. Yong Zhou, Baoquan Chen, and A. Kaufman, *Multiresolution tetrahedral framework for visualizing regular volume data*, IEEE Visualization '97 (VIS '97) (Washington - Brussels - Tokyo), IEEE, October 1997, pp. 135–142.