Variable Resolution Terrain Surfaces

E. Puppo
Istituto per la Matematica Applicata – Consiglio Nazionale delle Ricerche
Via de Marini, 6 (Torre di Francia) – 16149 Genova, ITALY
Email: puppo@ima.ge.cnr.it

Abstract

A model for the multiresolution decomposition of planar domains into triangles is introduced, which is more general than other multiresolution models proposed in the literature, and can be efficiently applied to the representation of a polyhedral terrain at variable resolution. The model is based on a collection of fragments of plane triangulations arranged into a partially ordered set. Different decompositions of a domain can be obtained by combining different fragments from the model. A data structure to encode the model is presented, and an efficient algorithm is proposed that can extract in linear time a polyhedral terrain representation, whose accuracy over the domain is variable according to a given threshold function. Furthermore, the size of the extracted representation is minimum among all possible polyhedral representations that can be built from the model, and that satisfy the threshold function. A major application of these results is in real time rendering of terrains in flight simulation.

1 Introduction

Multiresolution geometric models can be used in several application fields to support the representation and processing of geometric entities at different levels of resolution. The case of topographic surfaces is especially attractive for its impact on applications like geographic information systems, and virtual reality contexts. For instance, visualization in flight simulation can be made faster by rendering portions of terrain close to the observer at high resolution, while far portions are rendered at lower resolution.

In this paper we consider polyhedral terrains defined by a triangulation of a plane domain, where each vertex has an elevation value, and each triangle corresponds to a triangular patch approximating the elevation of terrain over the area of the triangle. In this context, a key concept is that the resolution of a model is somehow proportional to the refinement of the domain decomposition, hence to the number of its vertices. The power of a multiresolution model derives from its ability to adapt a representation to a required resolution, while minimizing its size.

Although many different multiresolution models have been proposed in the literature, the extraction of a representation at variable resolution has been investigated only recently. For this problem, a threshold function is defined over the terrain's domain, which specifies for each point a threshold for the error of the model in approximating elevation at that point. A terrain representation satisfying such a threshold must be extracted from the multiresolution model. It is also important to obtain a representation that is as small as possible.

The model presented here is based on a multiresolution decomposition of a planar domain, called a *multi-triangulation*, and it is thought as a generalization over a broad class of multiresolution models. The basic idea underlying the model is that a large number of different subdivisions of a planar domain can be obtained on the basis of a relatively small set of atomic components, called *fragments*, which can be combined in different ways to cover the domain. Fragments can be partially overlapping, and they are arranged into a partially ordered set, where the order relation is dependent on interferences of fragments on the plane, and on the possibility to combine them to obtain triangulations.

The model gives support to variable resolution, allowing an application to extract a representation of minimum size for an arbitrary threshold function in linear time. This is the first proposal addressing the minimality of the representation extracted in a strong sense: indeed, the algorithm proposed warrants that the representation extracted is the smallest possible that can be built from the triangles of the model.

The rest of the paper is organized as follows. In Section 2 related work is briefly reviewed. In Section 3 some terminology and notations are introduced, while the definition of multi-triangulations is given in Section 4. In Section 5, a data structure to encode multi-triangulations is described. In Section 6, an algorithm for extracting a triangulation at variable resolution from a multi-triangulation is described and analyzed. In Section 7, some examples and applications are discussed. Finally, in Section 8 some concluding remarks are given.

2 Related work

Several multiresolution terrain models have been proposed in the literature: see [6] for a recent survey. Here, we focus only on most recent models, and on issues relevant to the support of variable resolution.

For all models considered here, we assume the following: each triangle of a triangulation defining a polyhedral terrain is tagged with an *accuracy*, corresponding to the maximum error made in approximating the terrain over its area with its corresponding linear patch; given a *threshold* function defined over the terrain's domain, a representation is said to satisfy such a threshold if the accuracy of each triangle is smaller than the minimum of the function on the triangle itself.

Most multiresolution models support only constant thresholds. A few models supporting more general functions, called *variable resolution models*, were proposed recently by de Berg and Dobrindt [4], Cignoni et al. [3], and De Floriani and Puppo [5].

The *hierarchical representation* proposed in [4] is defined as a pyramid of triangulations, whose structure is essentially based on an earlier hierarchical triangulation scheme proposed by Kirkpatrick [9]. The pyramid is built bottom-up: each layer is obtained by removing a constant fraction of the vertices from the previous layer. A traversal algorithm extracts a representation at variable resolution based on an arbitrary threshold function, in time linear in its output size. The algorithm is based on a top-down traversal of the pyramid, and on a greedy construction of the result. Unfortunately, the greedy approach does not warrant that the desired accuracy is satisfied everywhere: indeed, because of the current configuration at an intermediate step, the algorithm can be obliged to accept into the solution triangles whose accuracy is worse than required.

The *hypertriangulation* model proposed in [3] is built from fragments by recording the history of the construction of a triangulation, e.g., through an on-line Delaunay triangulation algorithm [8, 12]. Fragments are arranged into a two-dimensional simplicial complex embedded in three-dimensional space, which is obtained by assigning a third coordinate to each vertex, whose value correspond to an iteration counter. Each time a vertex is inserted into a triangulation, the set of triangles updating the triangulation is stored as a new fragment that form a "dome" over the portion of the current triangulation that is updated. It is straightforward to obtain an analogous structure through a dynamic procedure that iteratively eliminates vertices from a given triangulation. An algorithm for extracting variable resolution representations from hypertriangulations is proposed, which is valid only for a special class of threshold functions, namely those monotonically increasing with distance from a given viewpoint, which are suitable to flight simulation. The algorithm is based on a breadth-first traversal of the domain starting at the viewpoint, and an incremental construction of the representation. The traversal technique ensures that the extracted model will satisfy the threshold function everywhere, but the computational complexity is suboptimal.

In [5] hierarchical triangulated models are discussed, whose general structure is a tree: each node is a triangulation with a triangular domain, refining the domain covered by a triangle in its parent node. Two algorithms for variable resolution surface extraction are proposed. The first algorithm is a simple top-down visit of the tree which accepts triangle as soon as its accuracy lies below the threshold. The resulting structure is a subivision called a generalized triangulation, in which some triangles are added new vertices along their edges. A triangulation of such generalized triangles is performed next to obtain a triangulated surface, and the whole algorithm is completed in time linear in its output size. However, the approximating function is changed by the triangulation of generalized triangles, hence the accuracy of the final structure might be worse than desired. The second algorithm is essentially an adaptation of that of Cignoni et al. to hierarchical triangulated models. The accuracy of the result is warranted, but the algorithm works only for the special class of threshold functions described above, and its computational complexity is suboptimal.

3 Preliminaries

Posets: Let $\mathcal{C}$ be a finite set. A *partial order* on $\mathcal{C}$ is a reflexive, antisymmetric and transitive relation $\leq$ on its elements. A pair $(\mathcal{C}, \leq)$ is called a *poset.* For every $c, c' \in \mathcal{C}$, the following notations are used: $c < c'$ means $c \leq c'$ and $c \neq c'$; $c \prec c'$ means $c < c'$ and $\nexists\, c''$ such that $c < c'' < c'$.

$c \in \mathcal{C}$ is a *minimal* element of $\mathcal{C}$ if $\nexists\, c' \in \mathcal{C}$ such that $c' < c$; if there exists a unique minimal element in $\mathcal{C}$, it is called the *least* of $\mathcal{C}$. A subset $\mathcal{C}' \subseteq \mathcal{C}$ is called a *lower set* if $\forall c' \in \mathcal{C}'$, $\forall c \leq c'$ then $c \in \mathcal{C}'$. For any $c \in \mathcal{C}$, the set $\mathcal{C}_c = \{c' \in \mathcal{C} \mid c' \leq c\}$ is the smallest lower set containing c, and it is called the *down-closure* of c. We also define the *sub-closure* of c as $\mathcal{C}_c^- = \{c' \in \mathcal{C} \mid c' < c\} = \mathcal{C}_c \setminus \{c\}$.

Given a lower set $\mathcal{C}' \subseteq \mathcal{C}$, a *compatible ordering* on $\mathcal{C}'$ is any total order $\leq_{\mathcal{C}'}$ on its elements such that $\forall c, c' \in \mathcal{C}', c \leq c' \Rightarrow c \leq_{\mathcal{C}'} c'$.

Triangulations: Given a generic set of triangles $T = \{t_1, \ldots, t_N\}$ in $\mathbb{R}^2$, called a *t-set*, we use the following notation: $|T| = N$ is the *size* of T; $\Delta(T) = \cup_{i=1}^{N} t_i$ is the *domain* of T; $V(T)$ is the set of vertices of the triangles of T; $E(T)$ is the set of edges of the triangles of T; $\forall t \in T$, $i(t)$ is the interior of t.

A *plane triangulation* is a regular simplicial complex of order two embedded in $\mathbb{R}^2$: a triangulation is characterized by a t-set T such that for each pair of triangles $t_i, t_j \in T$, with $t_i \neq t_j$, then $t_i \cap t_j$ is either empty, or an edge or a vertex of both t_i and t_j. In the following we will use a triangulation and its t-set interchangeably.

A triangulation T whose domain is a (polygonal) region Ω is also called a *covering* of Ω. Given a generic t-set T' having Ω as domain, any covering of Ω formed of triangles of T' is called a triangulation *generated by* T'.

4 Multi-triangulations

All results in this section are stated without any proof. Complete proofs are given in [10].

Given two triangulations T_i and T_j, we define the *interference* $\otimes$, the *subtraction* $\ominus$, and the *pasting* $\oplus$, respectively, as follows:

$$T_i \otimes T_j = \{t \in T_i \mid \exists t' \in T_j,\ i(t) \cap t' \neq \emptyset\} \quad (1)$$
$$T_i \ominus T_j = T_i \setminus (T_i \otimes T_j) \quad (2)$$
$$T_i \oplus T_j = (T_i \ominus T_j) \cup T_j. \quad (3)$$

If $T_i \otimes T_j \neq \emptyset$ then we say T_i and T_j are *interfering*, otherwise we say they are *independent.* If $T_0 \oplus T_1$ is also a triangulation, and $\Delta(T_0 \oplus T_1) = \Delta(T_0) \cup \Delta(T_1)$, then T_1 is said *compatible over* T_0, and $T_0 \oplus T_1$ is said a *modification* of T_0. If T_1 is compatible over T_0, and no subset $T_1' \subset T_1$ is compatible over T_0, then T_1 is said *minimally compatible* over T_0.

Given a sequence of triangulations $T_0, \ldots, T_k$ we define its *upward pasting* as the successive pasting of its elements $\oplus_{i=0}^{k} T_i = T_0 \oplus T_1 \oplus \ldots \oplus T_k$. We say that $T_0, \ldots, T_k$ is an *upward compatible sequence* if $\forall j = 1, \ldots, k$, T_j is compatible over $\oplus_{i=0}^{j-1} T_i$.

Definition 4.1 *Let Ω be a polygonal domain in $\mathbb{R}^2$. A* multi-triangulation *(MT) on Ω is a poset $(\mathcal{T}, \leq)$ where $\mathcal{T} = \{T_0, \ldots, T_h\}$ is a set of triangulations, and $\leq$ is a partial order on $\mathcal{T}$ satisfying the following conditions:*

1. *$\forall i = 0, \ldots, h$, $\Delta(T_i) \subseteq \Omega$;*
2. *$\forall i, j = 0, \ldots, h$, $i \neq j$,*
   (a) *$T_i \prec T_j \Rightarrow T_i \otimes T_j \neq \emptyset$;*
   (b) *$T_i \otimes T_j \neq \emptyset \Rightarrow T_i$ is in relation with T_j (i.e., either $T_i < T_j$ or $T_j < T_i$).*
3. *$\forall$ lower set $\mathcal{T}' \subseteq \mathcal{T}$, if $[T_0' <_{\mathcal{T}'} \ldots <_{\mathcal{T}'} T_k']$ is a compatible ordering of the elements of $\mathcal{T}'$, then $T_0', \ldots, T_k'$ is an upward compatible sequence, and $\oplus_{i=0}^{k} T_i'$ is a covering of Ω.*

The elements of $\mathcal{T}$ are called fragments. *The t-set $T_{\mathcal{T}} = \cup_{i=0}^{h} T_i$, i.e., the set of all triangles of the multi-triangulation, is called the* associated t-set *of $\mathcal{T}$.*

In the following, a MT will be denoted simply by its set of fragments $\mathcal{T}$, while the ordering $\leq$ will be omitted, whenever no ambiguity arises.

Lemma 4.2 *A multi-triangulation $\mathcal{T}$ has always a least element T_i such that $\Delta(T_i) = \Omega$.*

Proof: omitted.

Lemma 4.3 *The upward pasting of a lower set $\mathcal{T}' \subseteq \mathcal{T}$ is indipendent of the specific compatible ordering.*

Proof: omitted.

Without loss of generality, in the following we will assume that T_0 is the least element. Being independent of the ordering, the upward pasting obtained from any compatible ordering of a lower set $\mathcal{T}'$ will be simply denoted $\oplus\mathcal{T}'$. The upward pasting $\oplus\mathcal{T}$ of the whole set $\mathcal{T}$ will be called the *top* of $\mathcal{T}$.

Definition 4.4 *A multi-triangulation $\mathcal{T}$ is in* canonical form *if every fragment T_i of $\mathcal{T}$ is minimally compatible over the upward pasting of its sub-closure $\oplus\mathcal{T}_{T_i}^-$.*

It is easy to show that any MT can be transformed into a MT in canonical form having the same associated t-set: this is done by breaking each fragment into pieces, each of which is minimally compatible over its sub-closure. Henceforth, we will always assume that a MT is in canonical form.

Definition 4.5 *A multi-triangulation $\mathcal{T}$ is* non-redundant *if*

1. *there are no duplicate triangles, i.e., each triangle of the associated t-set belongs to exactly one fragment;*
2. *$\forall i, j = 0, \ldots, h$, if e is an edge common to T_i and T_j, and $T_j < T_i$, then e is an edge of $\oplus\mathcal{T}_{T_i}^-$.*

The meaning of non-redundancy is the following. Triangles, which are the atoms of the structure, are not replicated in different fragments to preserve a sort of minimality. Edges can be replicated, since they provide an interface for pasting triangulations. However, if different triangulations share a common edge, they must form a sequence in the poset. This latter condition is fundamental to guarantee that the MT supports all possible triangulations generated by its associated t-set. Such a property is stated by the following theorem.

Theorem 4.6 *Let $\mathcal{T}$ be a non-redundant multi-triangulation. Then for any triangulation T generated by $T_{\mathcal{T}}$ there exists a lower set $\mathcal{T}' \subseteq \mathcal{T}$ such that $T = \oplus\mathcal{T}'$.*

Proof: omitted.

The above theorem states the power of a MT of expressing different coverings of the same domain on the basis of a relatively small set of atomic entities (i.e., its triangles). In practice, we will be also interested in controlling the size of each covering versus its resolution. Therefore, in the following sections we will focus on a special class of MTs, for which the order relation provides control over the size.

Definition 4.7 *A multi-triangulation $\mathcal{T}$ is* increasing *if and only if*

1. *$\forall\ \mathcal{T}', \mathcal{T}''$ lower sets,*
$$(\mathcal{T}' \subset \mathcal{T}'') \Rightarrow |\oplus\mathcal{T}'| < |\oplus\mathcal{T}''|;$$
2. *$\forall\ T_i$ fragment, $\forall e \in E(T_i) \setminus E(\oplus\mathcal{T}_{T_i}^-)$, at least one endpoint of e belongs to $V(T_i) \setminus V(\oplus\mathcal{T}_{T_i}^-)$.*

A decreasing *multi-triangulation is defined similarly. A multi-triangulation which is either increasing or decreasing is said* monotone.

In the above definition, the first requirement warrants that each modification will increase the size of the triangulation, while the second requirement warrants that modifications are caused only by insertion of vertices, i.e., no edges flip occur on pairs of existing triangles.

A further interesting property of MTs is that also the collection of interferences of each fragment over its sub-closure is a MT, called the *reverse*. Such a structure can be encoded together with the primary one, and, having the same associated t-set, it generates the same set of triangulations. In case of monotone MTs, the reverse has a monotonicity opposed to the primary structure (see [10] for details).

Definition 4.8 *An increasing multi-triangulation $\mathcal{T}$ has* linear growth *if and only if for each lower set $\mathcal{T}' \subseteq \mathcal{T}$ the size of $\mathcal{T}'$ is linear in the size of its upward pasting. A decreasing multi-triangulation has linear growth if and only if its reverse has linear growth.*

In Section 6, we will see that linear growth is a desirable property since it permits to achieve optimal output sensitive time complexity in visiting the structure. A sufficient condition to achieve linear growth for increasing [decreasing] MTs is that vertices are never removed [added] by modifications, and the size of each fragment is linear in the number of its internal [removed] vertices.

5 A data structure for MT

For the purpose of the algorithm presented in the next section, a multi-triangulation $\mathcal{T}$ can be encoded by a data structure based on interferences, which maintains directly relations between triangles and fragments, and indirectly relations between fragments. Three sets are maintained: the set of all vertices $V(T_{\mathcal{T}})$; the set of all triangles $T_{\mathcal{T}}$; the set of all fragments $\mathcal{T}$.

Each fragment T_i in the fragment set, contains the following information: a list of (pointers to) triangles composing T_i, called the *ceiling*; a list of (pointers to) triangles composing $\oplus\mathcal{T}^-_{T_i} \otimes T_i$, called the *floor*. The least element T_0 has an empty floor. A dummy fragment with an empty ceiling and a floor containing all triangles of the top of $\mathcal{T}$ is also added to the structure. In fact, the collection of all floors corresponds to the set of fragments of the reverse of $\mathcal{T}$.

Therefore, each triangle t is referenced by two fragments: the fragment T_i containing t (in its ceiling), called the *lower* fragment; the fragment containing t in its floor, called the *upper* fragment. Each triangle in the triangle set contains pointers to its upper and lower fragments, as well as pointers to its three vertices. Each vertex is simply characterized by its two cartesian coordinates.

Note that relation $\prec$ between fragments is induced by the links between fragments and triangles: given a fragment T_i, fragments preceding it are the lower fragments of triangles of the floor of T_i, while fragments following it are the upper fragments of triangles of the ceiling of T_i. The same data structure encodes also the reverse MT.

The following operations can be implemented in the above data structure with linear complexity in their output size: CEILING(T_i), and FLOOR(T_i) return the sets of triangles composing the ceiling, and the floor of a fragment T_i, respectively; LOWER(t), and UPPER(t) return the lower, and upper fragments of t, respectively; LEAST($\mathcal{T}$) returns the least fragment T_0, while TOP($\mathcal{T}$) returns the top fragment.

6 Extracting a triangulation at variable resolution

In this section we consider only monotone MTs, and we focus on the extraction of variable resolution representations based on an arbitrary threshold function.

In order to extract variable resolution coverings, we need a test to either accept or discard (the resolution of) a given triangle. In order to be generic about the application, we assume that a Boolean condition $c()$ is defined on the triangles of $\mathcal{T}$, such that for a given triangle t, $c(t)$ is true if and only if the resolution of t is acceptable. Similarly, the notation $c(T)$ means that all triangles of a triangulation T satisfy $c()$. We consider the following problem:

> *Given a non-redundant and monotonically increasing multi-triangulation $\mathcal{T}$, and a Boolean condition $c()$, find the smallest triangulation T generated by $T_{\mathcal{T}}$ such that $c(T)$ is true.*

The algorithm we propose to resolve such problem works by traversing $\mathcal{T}$ starting at its least fragment, visiting the elements of the MT in breadth-first order, and marking all triangles that cannot be part of the solution. A queue of fragments that must be visited is maintained, which is initialized with the least fragment T_0, while triangles selected for a potential solution are added to a list. After traversal, such a list will contain all triangles of the solution, plus some extra (marked) triangles that are purged through a single scan.

Traversal is performed through a loop controlled by the content of the queue. At each iteration, the current fragment is extracted from the queue, and triangles composing its floor and its ceiling are visited. Each non-marked triangle of the floor is marked, and if its corresponding lower fragment has not been visited yet, then such a fragment is added to the queue. For each non-marked triangle of the ceiling of the fragment, if its upper fragment has been visited already, then the triangle is marked, otherwise it is tested against condition $c()$. If a triangle t passes the test, it is added to the potential solution, otherwise it is marked, and its upper fragment is added to the queue. Note that a triangle can possibly be marked after its insertion in the potential solution, when visiting the floor of its upper fragment. Traversal stops when either the queue becomes empty, or a triangle in the top level fails the test: in the latter case the algorithm returns an empty solution. After traversal, the list of potential solution is scanned, and only triangles not marked are given in output.

In Figure 1 we give a detailed pseudo-code of the algorithm, which is based on the data structure described in the previous section. Besides the primitives on MTs outlined above, we make use of some standard procedures acting on generic lists. Let Q be a generic list, and let e be a generic element, the following primitives are used: MAKE_EMPTY(Q), IS_EMPTY(Q), FIRST(Q), ADD(Q, e), REMOVE(Q, e). Notice that REMOVE(Q, e) removes the current element e of list Q during list scan, hence it can implemented with constant time complexity. Finally, we use primitives to mark and test generic elements: MARK(e), MARKED(e), NULL(e). Note that both triangles and fragments can be marked: marking a triangle means that it cannot be part of the solution; marking a fragment means that it has been visited. All the primitives above can be implemented with constant time complexity.

The time complexity analysis of the algorithm is straightforward and omitted here. If a solution exists, the algorithm runs in time linear in the size of the lower set generating the solution, otherwise it gives a negative answer in time at most linear in the size of the MT. Therefore, for MTs having linear growth an existing solution can be found in optimal time, i.e., linear in the output size.

The correctness of the algorithm is proven here only for the case in which a solution exists. The negative case is straightforward, hence omitted. First, we show that the output t-set T is indeed a covering satisfying $c()$. Next, we show that any other covering satisfying $c()$ is necessarily larger than T.

If a triangle is added to the list T during the algorithm, it necessarily satisfies $c()$. Therefore, it is sufficient to show that T is a covering. Let us consider the set $\mathcal{T}'$ of fragments visited by the algorithm: $\mathcal{T}'$ is a lower set of $\mathcal{T}$. Indeed, when visiting the floor of a fragment, we make sure that all fragments preceding it are also visited. Let us now consider the set of triangles of $T_{\mathcal{T}'}$: each such triangle t belongs to T if and only if there does not exist a fragment of $\mathcal{T}'$ having t in its floor. Indeed, if there exists one such fragment T_i, then t would be marked when visiting the floor of T_i. Conversely, if no such fragment exists, then the upper fragment of t is never visited. This means that t cannot be marked, since marking t would either cause, or be caused by visiting its upper fragment. Hence, t must be a triangle of T. In conclusion, the triangles of T are all and only those that have no upper fragment in $\mathcal{T}'$. It follows that T is indeed the upward pasting of $\mathcal{T}'$.

From Theorem 4.6, we know that any covering generated by $\mathcal{T}$ can be obtained by an upward past-

```
Algorithm EXTRACT(𝒯,c(),out T)
begin
   local var Q : queue; F, F1 : fragment; t : triangle;
   MAKE_EMPTY(Q);
   MAKE_EMPTY(T);
   F = BOTTOM(𝒯);
   MARK(F);
   ADD(T, F);
   while not IS_EMPTY(Q) do
      F = FIRST(Q);
      for every t ∈ FLOOR(F) do
         if not MARKED(t) then
            MARK(t);
            F1 = LOWER(t);
            if not (NULL(F1) or MARKED(F1))
            then
               MARK(F1);
               ADD(Q, F1);
            end if
         end if
      end for;
      for every t ∈ CEILING(F) do
         if not MARKED(t) then
            F1 = UPPER(t);
            if MARKED(F1) then
               MARK(t)
            else
               if c(t) then
                  ADD(T, t)
               else
                  if (F1 == TOP(𝒯)) then
                     MAKE_EMPTY(T);
                     exit()
                  end if;
                  MARK(t);
                  MARK(F1);
                  ADD(Q, F1)
               end if
            end if
         end if
      end for
   end while;
   for every t ∈ T do
      if MARKED(t) then REMOVE(T, t) end if
   end for
end.
```

Figure 1: The algorithm for extracting a triangulation at variable resolution.

ing on a lower set $\mathcal{T}'' \subseteq \mathcal{T}$. Therefore, in order to prove the minimality of T, we show that all fragments visited by the algorithm will necessarily belong to the lower set $\mathcal{T}''$ generating the solution. We prove this fact inductively. Since $\mathcal{T}''$ is a lower set, the least element T_0 of $\mathcal{T}$ is certainly a fragment of $\mathcal{T}''$. Now, let us assume that the first k fragments visited by the algorithm belong to $\mathcal{T}''$. We show that the $(k+1)$-th fragment T_j visited must also belong to $\mathcal{T}''$. There are two possible causes for visiting T_j. Either (first **for** loop) T_j is preceding a fragment already visited, hence T_j must be part of $\mathcal{T}''$, which is a lower set; or (second **for** , innermost **if**) some triangle t belonging to the ceiling of a fragment of $\mathcal{T}''$, and to the floor of T_j, fails the test. In this case, it means that t cannot be part of the solution, hence there must exist a fragment in $\mathcal{T}''$ that is pasted over t. The only such possible fragment is indeed T_j.

Now, let us suppose that $\mathcal{T}'$ is a proper subset of $\mathcal{T}''$. Since $\mathcal{T}$ is increasing, this means that the size of $\oplus\mathcal{T}''$ would be larger than the size of $\oplus\mathcal{T}'$, which contradicts the fact that $\mathcal{T}''$ is a solution. Hence, we must have $\mathcal{T}' \equiv \mathcal{T}''$.

All the above proves the following theorem.

Theorem 6.1 *Given an increasing non-redundant multi-triangulation $\mathcal{T}$, and a Boolean condition* c() *on $\mathcal{T}$, it is possible to decide whether there exists a covering generated by the associated t-set $T_{\mathcal{T}}$, and satisfying* c(), *and to find the smallest such covering, in time linear in the size of the lower set generating it (i.e., at most linear in the size of $\mathcal{T}$).*

If $\mathcal{T}$ has linear growth, an existing solution can be found in time linear in its output size.

Completely analogous algorithm and proof are valid for a monotonically decreasing MT. In this case, the reverse MT is visited. Since the data structure encodes also the reverse MT, it is not necessary to recompute it explicitly. In this case, running the algorithm for the reverse triangulation is equivalent to visit $\mathcal{T}$ starting at the floor of its top, and swapping floor and ceiling, as well as upper and lower, in the code.

7 Examples and applications

Multi-triangulations can be used provided that one can build them. In this section, we review some MTs based on Delaunay triangulations, which are easy to build, and can be useful for variable resolution terrain modeling.

Definition 7.1 *A* Delaunay multi-triangulation *is a multi-triangulation $\mathcal{T} = \{T_0, \ldots, T_h\}$ such that for each $i = 0, \ldots, h$ the upward pasting $\oplus_{j=0}^{i} T_j$ is a Delaunay triangulation.*

Among such MTs, we are interested in those that can be built by dynamic algorithms, and that are monotone. Consider for instance the on-line construction of a Delaunay triangulation of n sites [12, 8]. Let us assume that an initial triangulation T_0 built over $n-h$ such sites, and covering the convex hull of all sites is given. Let us consider the sequence of operations that build the triangulation of all n sites starting at T_0. At each step i, for $i = 1, \ldots, h$, a new site is inserted and the current triangulation is updated: let us define T_i as formed by the set of new triangles that update the triangulation. It is easy to see that the resulting set of fragments $\mathcal{T} = \{T_0, \ldots, T_h\}$ is a MT, whose structure is analogous to that of the hypertriangulation proposed by Cignoni et al. [3]. It is also easy to show that such a structure, called a *historical Delaunay multi-triangulation* is in canonical form, non-redundant, and increasing. Therefore, such a MT is suitable for the application of the algorithm described in the previous section. If, moreover, a historical Delaunay MT is built through a randomized algorithm, such as those proposed by Guibas et al. [8], or by Boissonnat and Teillaud [1], it follows from the randomized analysis of such algorithms that it will be built in optimal time, and it will have linear growth, with high probability. Therefore, such a randomized structure will also give an expected optimal output sensitive time complexity for the variable resolution extraction.

A further interesting property, which is a consequence of the previous ones, is that any lower set of a historical Delaunay nulti-triangulation $\mathcal{T}$ is itself a historical Delaunay MT, i.e., that all triangulations generated by $T_{\mathcal{T}}$ are Delaunay triangulations.

Similar results are obtained by considering an equivalent structure which encodes a historical sequence of operations which simplify a triangulation of all the n sites by eliminating one vertex at a time. In this case, the resulting MT would be decreasing.

As a second example, let us consider the variable resolution model of de Berg and Dobrindt: this is in fact a reverse historical Delaunay MT. Let us define

T_0 as the triangulation of all n sites, and let us consider a sequence of vertex deletions which is compatible with the ordering of levels, and which is arbitrary for the vertices of a given level. This makes sense since the vertices eliminated at each level form an independent set. Therefore, the result of each such elimination is always a Delaunay triangulation, and the influence polygon of each vertex eliminated forms a fragment. It is easily seen that the resulting structure is a MT with all properties defined above, including linear growth. Hence, all results stated previously apply also to such a structure.

Other models proposed in the literature can be interpreted as special cases of MTs. This subject is discussed in more detail in [7].

Applications of MTs to varialbe resolution terrain modeling is straightforward. The model can be built with any of the strategies described above. For each vertex in the model, its elevation z is encoded as an additional information. For each triangle t, its accuracy ε_t is also encoded, which corresponds to the maximum error in approximating the elevation of data points contained in t. Given a threshold function $\tau : \Omega \rightarrow \mathbb{R}$, the condition for algorithm EXTRACT is defined as $c(t) = (\varepsilon_t \leq \min_{p \in t} \tau(p))$. For applications such as flight simulators, function τ is usually decreasing with distance from the viewpoint. A possible function used in [3] is simply $\tau(p) = K|p - v|$, where v is the viewpoint, $| \cdot |$ is the distance in $\mathbb{R}^2$, and K is a suitable constant.

Another possible application of multi-triangulations is in domain decompositions for finite element methods. In this case, a possibility is that a domain must be decomposed into triangles whose size satisfies a user-defined density function [2]. In this case, algorithm EXTRACT could be used with a condition $c(t) = r(t) \leq \delta(o_t)$, where $\delta : \Omega \leftarrow \mathbb{R}$ is the density function, $r(t)$ is the circumradius of t, and o_t is its circumcenter.

8 Concluding remarks

The optimal time behavior of the algorithm for variable resolution surface extraction, the simplicity of its implementation, as well as the minimality of the representation it extracts, make MTs a valid support to enhance the quality of surfaces that can be rendered in real time. In this perspective, it is also easy to modify algorithm EXTRACT to restrict the search to a given region of the domain, such as a window corresponding to the field of view, in order to achieve a further speedup.

The data structure for encoding MTs, as well as algorithms for building one such structure, and the extraction algorithm presented here are under implementation. We plan to build next a small prototype for the real-time rendering of topographic surfaces based on MTs.

The reconstruction of the whole topological structure of representation extracted have not been considered, since for rendering applications the list of triangles is sufficient. However, it is easy to address such a subject by completing the data structure encoding the MT with adjacences between triangles, such as those used in [3].

The fact that a MT encodes all possible triangulations that can be built from its triangles is a result that goes beyond the scope of this paper. However, this has been proved only for the special class of non-redundant MTs. On a more general perspective, it would be interesting to study the relationships between a generic set of triangles T and all possible triangulations that can be built from it. The t-set associated to a MT has indeed a quite special structure, while addressing the problem for a generic t-set seems much harder.

Although MTs in their current form have been defined on planar domains, hence supporting only the representation of functional surfaces, they can be extended to variable resolution modeling of general surfaces embedded in three-dimensional space. The availability of simplification algorithms for triangulated surfaces, such as the one proposed by Schroeder et al. [11], offers also immediate means to build such structures easily. However, the generalization to non-planar domains involves some theoretical difficulties, since some properties that are immediate in the planar case are no longer valid for non-planar domains. In particular, it is not easy to provide control over the construction and traversal of an MT to warrant an extracted surface free of self-intersections: such undesirable sitations can appear in some cases because of warping caused by the approximation of a general surface through a piecewise-linear triangulated surface. In practice, self-intersections will not be frequent, therefore MTs can be extended immediately for practical purposes. However, a rigorous study would be required to make such an extension also theoretically sound.

Finally, MTs can be extended to an arbitrary dimension to obtain variable resolution decompositions of multidimensional domains, for applications in scientific visualization and finite element analysis. This subject is the topic of a companion paper [7], in which we also show how a number of multiresolution models proposed in the literature can be interpreted as special cases of our model.

Acknowledgments

Leila De Floriani and Paola Marzano are gratefully acknowledged for many helpful discussions and suggestions on the matter of this paper.

References

[1] J.D. Boissonnat, M. Teillaud, 1993, On the randomized construction of the Delaunay tree, *Theoretical Computer Science*, 112, pp.339-354.

[2] L.P. Chew, 1993, Guaranteed-quality mesh generation for curved surfaces, *Proceedings 9th ACM Symposium on Computational Geometry*, pp.274-280.

[3] P. Cignoni, E. Puppo, R. Scopigno, 1995, Representation and visualization of terrain surfaces at variable resolution, *Proceedings International Symposium on Scientific Visualization*, R. Scateni, Ed., World Scientific, Cagliari (Italy), September 1995.

[4] M. de Berg, M., K.T.G. Dobrindt, 1995, On the levels of detail in terrains, *11th ACM Symposium on Computational Geometry*, Vancouver, BC (Canada), June 5-7, 1995, pp.c26-c27. Also published in longer version as *Techical Report* UU-CS-1995-12, Utrecht University, Dept. of Computer Science, April 1995.

[5] L. De Floriani, E. Puppo, 1995, Hierarchical triangulation for multiresolution surface description, *ACM Transactions on Graphics*, 14, 4, pp.363-411.

[6] L. De Floriani, P. Marzano, E. Puppo, Multiresolution models for topographic surface description, *The Visual Computer*, (to appear).

[7] L. De Floriani, E. Puppo, A formal approach to multiresolution modeling, (in preparation).

[8] L.J. Guibas, D.E. Knuth, M. Sharir, 1992, Randomized incremental construction of the Delaunay and Voronoi diagrams, *Algorithmica*, 7, pp.381-413.

[9] D.G. Kirkpatrick, 1983, Optimal search in planar subdivisions, *SIAM Journal of Computing*, 12, pp.28-35.

[10] E. Puppo, 1996, Variable resolution terrain surfaces, *Techincal Report* N.6/96, Istituto per la Matematica Applicata, C.N.R., Genova, Italy, April 1996.

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

[12] D.F. Watson, 1981, Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes, *The Computer Journal*, 24, pp. 167-171.