Technical Report

UCAM-CL-TR-417
ISSN 1476-2986

Number 417

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Selective mesh refinement for interactive terrain rendering

Peter J.C. Brown

February 1997

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Selective Mesh Refinement for Interactive Terrain Rendering

Peter J. C. Brown
Computer Laboratory, University of Cambridge
Pembroke St, Cambridge, CB2 3QG, UK
Peter.Brown@cl.cam.ac.uk

February 1997

Abstract

Terrain surfaces are often approximated by geometric meshes to permit efficient rendering. This paper describes how the complexity of an approximating irregular mesh can be varied across its domain in order to minimise the number of displayed facets while ensuring that the rendered surface meets pre-determined resolution requirements. We first present a generalised scheme to represent a mesh over a continuous range of resolutions using the output from conventional single-resolution approximation methods. We then describe an algorithm which extracts a surface from this representation such that the resolution of the surface is enhanced only in specific areas of interest. We prove that the extracted surface is complete, minimal, satisfies the given resolution constraints and meets the Delaunay triangulation criterion if possible. In addition, we present a method of performing smooth visual transitions between selectively-refined meshes to permit efficient animation of a terrain scene.

1 Introduction

Interactive terrain rendering is a key operation in a number of applications such as flight simulation and geographical information systems (GIS). A terrain scene is notable in rendering terms for two reasons. Firstly, the underlying database is usually a large set of data points arranged in a regular grid. Secondly, the disparity between distances in object-space from the viewpoint to the closest and furthest points of the rendered surface is often large.

This first observation implies that if a terrain surface is to be rendered efficiently, it must first be approximated. Traditional approximation techniques can be divided into those which retain the regular grid nature of the original datapoints and those which produce a Triangulated Irregular Network (TIN). The efficiency with which a TIN can represent a surface for rendering purposes for rendering purposes was noted in [FL79], [FEKR90] and [DP95]. Simple subsampling can be used to approximate terrain in a grid-based fashion ([CMR90], [FZPM93]), but this may lead to significant features of the surface being discarded or modified [SP92].

The second characteristic of terrain scenes, the range of distances between the viewpoint and facets in the scene, indicates that the efficiency of the rendering process could be improved further by enhancing the level of detail of the rendered model only in specific areas of its domain. Such *selective refinement* could increase the resolution of the rendered surface close to the viewer, in areas around critical lines of the model (peaks, pits, passes, etc ([Dou86]) and in other areas of interest (AOIs) while reducing the resolution of the surface in less important regions. Performing selective refinement on a grid-based representation of terrain is relatively simple ([CMR90], [TB94], [LKR+96]) but this imposes significant restrictions on the approximation process. The method outlined below permits the production of a selectively-refined TIN and thus the benefits of using a triangle-based representation can be realised.

In addressing the task of selective mesh refinement, this paper makes the following contributions:

- A framework is presented within which most existing single-resolution terrain approximating techniques can perform selective mesh refinement with respect to object- and screen-space rendering requirements.
- We show how a global resolution criterion can be specified to ensure that the level of detail of specific surface features is enhanced.

- We demonstrate how to smoothly transition between different selective refinements of a mesh in order to maintain the temporal continuity of an animated sequence.

We obtain a selectively-refined mesh by combining seamlessly a set of terrain surface approximations which have been obtained from an unspecified simplification process without reference to the techniques peculiar to that process. In this way we decouple the mesh simplification and selective refinement processes and hence permit a particular approximation technique to be used when it is advantageous for a specific terrain rendering application. In particular, we can extract a selectively-refined mesh from a set of single-resolution Triangulated Irregular Networks (TINs) [FL79] and hence obtain the representation and rendering benefits of modelling a surface as a triangulation [FEKR90]. We describe data structures which represent a surface as a coarse approximation together with a set of *refinement operations* which incrementally refine portions of the surface. This is a generalisation of previous continuous-resolution surface representation techniques ([DP95], [CPS95], [Hop96]).

Our selective refinement algorithm extracts a surface from this representation such that the resolution of each facet in the extracted surface is guaranteed to meet given resolution criteria. These minimum resolution constraints are specified by a *Resolution Control Function* (RCF) defined in object-space. The resolution required in a rendered surface is often described solely by local screen-space considerations, such as the projected size of triangles ([Hop96], [LKR+96]). By using a global object-space resolution constraint, which can be supplemented by screen-space requirements, we can ensure that features of the terrain are displayed at directly-queryable object-space resolutions. This is essential for applications such as navigation and nap-of-the-earth flight simulation ([Sch83], [Kle90]).

We prove that a selectively-refined surface generated by this method has the following properties:

1. The extracted surface satisfies the RCF.
2. The extracted surface completely covers the domain of the original.
3. The extracted surface is the smallest set of facets which can form a complete surface which satisfies the RCF.
4. If the coarse approximation and refinement operations are Delaunay triangulations[1] then the extracted surface will also satisfy the Delaunay criterion.

Furthermore, we demonstrate how to smoothly transition between adjacent frames in a dynamic display of a selectively-refined mesh by modifying the geometry of the displayed surface. These *geomorphs* [Hop96] maintain a complete surface during the transition period and hence no blending is required to minimise temporal aliasing.

The next section contains a discussion of the background to our work in the field of multiresolution terrain modelling. The following sections discuss the terminology, data structures, algorithms and proofs connected with our selective refinement and geomorphing techniques. We conclude with results from an implementation of these techniques.

2 Multiresolution Modelling

A number of recent papers have tackled the problem of "multiresolution terrain modelling" and have demonstrated data structures and algorithms which permit the extraction of multiple single-resolution approximations as well as the extraction of surfaces at variable resolution. It is the latter problem which was labelled "selective refinement" by Hoppe [Hop96].

De Floriani and Puppo [DP95] generalise the previous work of De Floriani *et al*, Scarlatos [SP92], Ferguson [FEKR90], and others, in their formal description of a *Hierarchical Triangulation* from which a selectively-refined generalised triangulation can be obtained. This generalised triangulation satisfies the given precision requirement at each point in the domain but cannot guarantee continuity between triangles. A further triangulation step is required to produce a complete surface, but this may produce a mesh which violates the previously-satisfied precision requirement.

De Berg and Dobrindt [dD95] present a system of linking a hierarchical sequence of Delaunay triangulations such that a selectively-refined surface can be extracted which satisfies the Delaunay

[1] A Delaunay triangulation $\mathcal{T}$ is such that for each triangle t in $\mathcal{T}$, there is no vertex of $\mathcal{T}$ in the interior of t's circumcircle [PS85]

property. As [DP95] notes, though, the resulting surface is not guaranteed to meet the specified accuracy requirements.

The algorithm of Cignoni *et al* [CPS95] overcomes the above problems by extracting a surface from what the authors call a *HyperTriangulation (HT)*, in which the refinements required to modify each Level of Detail (LOD) produced by the approximating process are successively "pasted" onto the coarsest approximation. A variable resolution surface can be extracted from the HT such that it satisfies a simple resolution function, *viz* the approximation error in the extracted triangles is proportional to their distance from a specified viewpoint. As [DP95] notes, this may be suitable for applications such as flight simulators, but not if we wish to have a more complex resolution function. For example, we may wish to ensure that the resolution of the rendered surface is high around the viewer, in regions containing critical lines of the model (peaks, pits, passes, etc [Dou86]) and in other Areas of Interest (AOIs). A surface which satisfies a resolution function incorporating all of these constraints can be extracted by our selective refinement algorithm.

The *progressive mesh* (PM) approach of Hoppe [Hop96] permits a 3D surface to be represented in a lossless, continuous-resolution form upon which certain operations can be performed, including selective refinement and geomorphing. Hoppe's selective refinement method applied to terrain rendering has three disadvantages over the algorithm presented below. Firstly, the minimal data structure employed by Hoppe limits the selective refinement process to producing only an approximation of the surface required by the input resolution constraints (Hoppe states that one of the items of future work is a spatial data structure to permit efficient selective refinement). Secondly, the PM data structure also prevents geomorphing between two different selectively-refined meshes. Lastly, the PM approach requires the use of a particular simplification strategy and this produces approximations which use optimised vertex positions rather than the original data points. This may not be acceptable to particular rendering applications and our method provides a framework within which selective refinement can be performed on the output of most approximation techniques, including that used by progressive meshes.

3 Preliminaries

3.1 Selective Refinement Overview

The two inputs to our selective refinement algorithm are:

(A) The output from a surface approximation method, in the form of a coarse single-resolution approximation to a terrain surface and a set of *refinement operations*, i.e. the incremental modifications which were made to the base mesh in order to approximate the original dataset over a continuous range of (increasing) resolutions. Such a refinement history, of which Hoppe's PM approach [Hop96] and Cignoni's HyperTriangulation [CPS95] are examples, is independent of the approximating process and can be generated by both refinement and simplification techniques [CPS95]. Indeed, all of the algorithms discussed in [HG95] can provide this approximation information apart from Rossignac & Borrel's grid-sampling method [RB93], Turk's point repulsion procedure [Tur92] and Lounsbery *et al*'s wavelets-based multiresolution analysis [LDW94] since these methods cannot provide a correspondence between triangles in different LODs.

Section 3.3 formalises this continuous-resolution surface representation.

(B) A Resolution Control Function (RCF) which governs the extraction process. The RCF is a single-valued bivariate function defined over the terrain's domain which specifies the minimum resolution required of each facet in the extracted surface. Section 3.4 expands on the concept of an RCF and presents an example of how an RCF may be constructed.

The output from our algorithm is a geometric mesh, representing the terrain data which was approximated by input (A), which completely covers that input's domain and is such that the degree of refinement of each facet in the mesh satisfies the RCF across the domain of that facet.

3.2 Triangulations

We assume in the remainder of this paper that our algorithm's input (A) is a set of Triangulated Irregular Networks (TINs) from which it must produce a selectively-refined triangulation, although

the algorithm can handle any polygonal approximating mesh. The efficiency with which a TIN can represent a surface for rendering purposes was noted in [FL79], [FEKR90] and [DP95]. In contrast, the cost and complexity of handling multiple levels of detail of TINs conventionally was discussed in [CMR90], [TB94], and [LKR+96]. The results we present later demonstrate that the efficiency of our selective refinement technique could permit interactive terrain rendering systems to take advantage of TINs.

To provide a framework for our manipulation of triangulations, we adapt the notation of [DP95]. Let $V = \{v_1, \ldots, v_n\}$ be a finite set of points in the Euclidean plane $\mathbb{E}^2$. A *triangulation* of set V is a maximal plane straight-line graph $G = (V, E)$, where E is a set of non-crossing line segments with endpoints in V [PS85]. A triangulation of V can be expressed as a tuple $\mathcal{T} = (V, E, T, A)$, where T is the set of triangular faces induced by G on the plane. The faces of T cover the interior of the convex hull of V; such a region is called the *domain* of triangulation $\mathcal{T}$, and denoted $D(\mathcal{T})$. The interior of the domain, i.e. excluding the convex hull of V, is denoted by $I(\mathcal{T})$. A is the attribute information of the triangulation; A will include the elevation values associated with each element of V and may also contain discrete and scalar attributes such as material identifiers, normals, etc [Hop96].

We can now use these terms to state the problem tackled by our algorithm. We produce a selective refinement of a high-resolution triangulation $\mathcal{T} = (V, E, T, A)$ of the original terrain data. The input to the algorithm consists of a base triangulation, $\mathcal{T}^0 = (V^0, E^0, T^0, A^0)$, which covers the terrain domain Ω, together with a sequence of *refinement operations*, $\{R^1, \ldots, R^m\}$. The output is a triangulation $\hat{\mathcal{T}} = (\hat{V}, \hat{E}, \hat{T}, \hat{A})$, approximating $\mathcal{T}$, which completely covers Ω and is such that the degree of refinement of each triangle $t_i \in \hat{\mathcal{T}}$ satisfies the RCF across the domain of that triangle, $D(t_i)$. Note that we use $I(t_i)$ to indicate the interior of triangle t_i.

3.3 Mesh Representation

We represent a mesh in a continuous-resolution form by storing it as a base triangulation together with a history of the refinements which are required to improve the base mesh with respect to some resolution metric (Section 3.3.1). The selective refinement algorithm needs additional information about the spatial relationships between the refinement operations and this is represented in a Directed Acyclic Graph (Section 3.3.2).

3.3.1 Refinement Operations

A refinement operation corresponds to the insertion of one or more points into a previous approximation. Hence a refinement operation can be completely specified by the set of triangles which its point insertions affect, together with the retriangulation of the domain of those triangles. Formally, we define a refinement operation $R^i = (T^i_{pre}, \mathcal{T}^i_{post})$ such that, for $1 \leq i \leq m$:

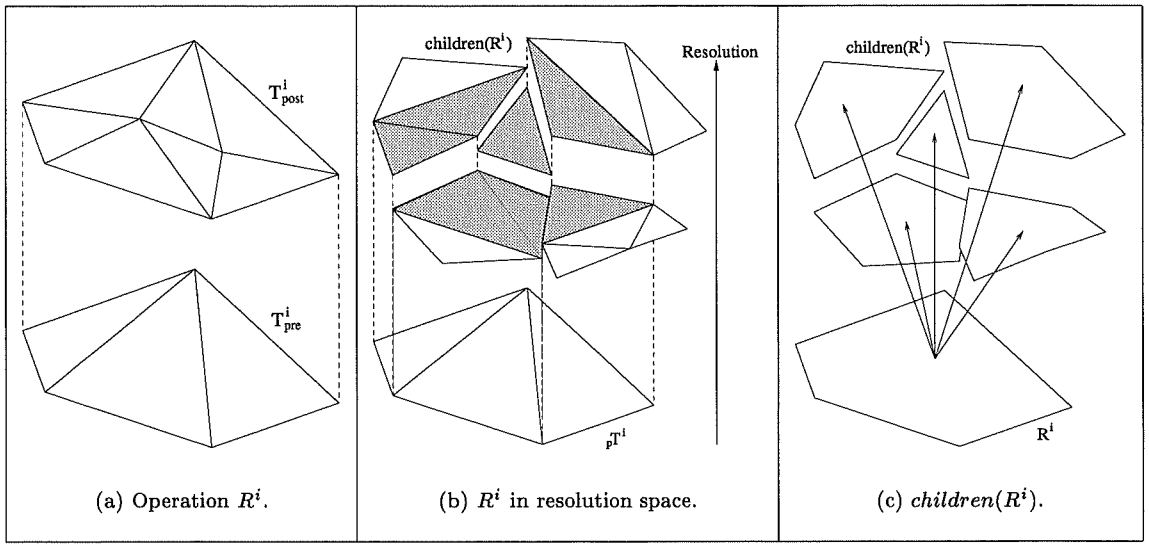
- $\mathcal{T}^i_{post} = \{V^i_{post}, E^i_{post}, T^i_{post}, A^i_{post}\}$ is the retriangulation, of a portion of Ω, introduced by the refinement operation, and;
- T^i_{pre} is the set $\{t_j : t_j \in T^k_{post}, 0 \leq k < i, I(t_j) \cap D(T^i_{post}) \neq \emptyset$ and $I(t_j) \cap D(T^l_{post}) = \emptyset, k < l < i\}$ of triangles which existed immediately previous to the invocation of R^i and which were affected by that operation. Note that we let R^0 be a pseudo refinement operation representing $\mathcal{T}^0$, where $T^0_{pre} = \emptyset$ and $\mathcal{T}^0_{post} = \mathcal{T}^0$.
- $|T^i_{pre}| < |T^i_{post}|$ since we are considering refinement operations created by the addition of points to a coarse triangulation. Refinement operations obtained from a simplifying approximation process can be "reversed" to this notation by simply swapping the sets T^i_{pre} and T^i_{post} in each region R^i and modifying V^i_{post}, E^i_{post} and A^i_{post} appropriately.

To complete our specification, we also require that input refinement operations must be both *complete* and *minimal*, where we define:

- R^i is *complete* $\iff$ $D(T^i_{pre}) = D(\mathcal{T}^i_{post})$, i.e. the domain of the set of triangles which the operation affects coincides with the domain of its specified retriangulation.

- R^i is *minimal* $\iff \nexists S \subset T^i_{pre}$ s.t. $D(S) \cap D(T^i_{post}) = D(S \cap T^i_{post})$, i.e. there is no subset of triangles in T^i_{pre} which can be completely replaced by a subset of T^i_{post} and hence the refinement operation is specified as locally as possible.

A refinement operation R^i which performs the transformation of a set of triangles T^i_{pre} into a new set of triangles T^i_{post} is visualised in Figure 1(a).



(a) Operation R^i. (b) R^i in resolution space. (c) *children*(R^i).

Figure 1: Visualisations of a refinement operation.

The refinement operations can be viewed as a set of transformations which produce a sequence of single-resolution triangulations from the base triangulation $\mathcal{T}^0$, i.e.

$$\mathcal{T}^0 \xrightarrow{R^1} \mathcal{T}^1 \xrightarrow{R^2} \ldots \xrightarrow{R^m} \mathcal{T}^m$$

We assume that this ordering of refinement operations corresponds to an increasing minimum resolution among the triangulations $\mathcal{T}^0, \ldots, \mathcal{T}^m$, i.e. $\forall t_j \in T^i, resolution(t_j) > \varepsilon^i$, for some $\varepsilon^i, 0 \leq \varepsilon^i < 1$, and $\varepsilon^i < \varepsilon^{i+1}$ for $0 \leq i < m$, where $resolution(t)$ is the resolution metric used by the surface approximation process over triangle t. We assume that the range of $resolution(t)$ is $[0, 1]$, where a value of 1 indicates that triangle t is in the highest-resolution approximation, $\mathcal{T}^m$.

We adapt the *birth error* and *death error* terms of Cignoni *et al* [CPS95] to conform to our resolution-based representation. We say that a refinement operation R^i has a birth resolution, $birthres^i$, which is the maximum resolution in the overall triangulation just before R^i was performed (we assume $birthres^0 = 0$); and each triangle t in T^i_{post} has a death resolution, $deathres(t)$, which is the maximum resolution in the overall triangulation just before t was retriangulated by another refinement operation (or 1 if t is part of $\mathcal{T}^m$).

A region R^i is visualised in resolution space (i.e the vertical axis represents increasing resolution) in Figure 1(b), where the triangles of T^i_{pre} are shown at their birth resolution, $birthres^i$. The triangles in T^i_{post} are drawn shaded at their death resolutions together with the T^j_{pre} for each of the R^j in which these triangles lie. Note that this implies that the T^j_{pre} are shown at their corresponding $birthres^j$.

3.3.2 The Refinement Region DAG

The birth and death resolution attributes are used to order the *refinement regions* (a term used to describe the triangles produced by a refinement operation, i.e. T^i_{post}, or interchangeably with "refinement operation" where no confusion will arise) in a directed acyclic graph (DAG) which captures the overlapping nature of these regions.

We define the sets of *children* and *parents* of a refinement region R^i as

$$children(R^i) = \{R^k : T^i_{post} \cap T^k_{pre} \neq \emptyset \text{ and } \exists t \in T^i_{post} \text{ s.t. } deathres(t) = birthres^k\}$$

$$parents(R^i) = \{R^k : T^i_{pre} \cap T^k_{post} \neq \emptyset \text{ and } \exists t \in T^k_{post} \text{ s.t. } deathres(t) = birthres^i\}$$

The sets of *ancestors* and *descendants* of a particular region can be defined as an extension of these sets in the usual way. The arcs in Figure 1(c) indicate the relationships which are represented by the set $children(R^i)$.

The refinement region DAG can then be defined as the set of nodes $\{R^0, \ldots, R^m\}$ together with the set of arcs $\{(R^i, R^j) : R^j \in children(R^i)\}$. The root of the DAG is R^0.

3.4 The Resolution Control Function

We specify the minimum resolution required of each of the triangles in the extracted surface as a global function, $RCF : D(\Omega) \rightarrow [0, 1]$. A triangle t in the extracted surface is said to *satisfy* the RCF if $\forall p \in D(t), deathres(t) \geq RCF(p)$.

We define two terms which relate refinement regions to the RCF:

- R^i *is finer than* the RCF if $\forall t \in T^i_{post}, p \in D(t), deathres(t) \geq RCF(p)$, i.e. if the post-refinement set of triangles in operation R^i satisfy the RCF;
- R^i *is completely finer than* the RCF if $\forall p \in D(R^i), birthres^i \geq RCF(p)$, i.e. if the pre-refinement set of triangles in operation R^i satisfy the RCF.

Example components from which a practical RCF could be constructed are visualised in resolution space in Figure 5. This RCF is used in the Results section to extract a mesh which is selectively-refined for a particular view of the Mt St Helens dataset, where the viewpoint is close to the edge of the dataset and the viewer is looking horizontally towards the centre of the domain. The first three RCF components shown in Figure 5 are view-dependent: the minimum resolution of the triangles in the view frustum is specified to be 0.5 by component (a); a "Head Up Display" in the centre of the screen is simulated by requiring the triangles which lie in the corresponding object-space cone (b) to be at a higher resolution; and the viewpoint is specified as an AOI by a Gaussian function centred on that point in (c). Figure 5(d) shows the result of using a critical line detection technique [Dou86] to obtain a function which requires the resolution of the ridge around the crater in the dataset to be increased. Components (b), (c) and (d) were clipped by the view frustum and then added to (a) to produce the RCF of Figure 5(e).

Hoppe [Hop96] presents local screen-space heuristics which determine the degree of refinement of a selectively-refined progressive mesh, without regard for object-space resolution accuracy. Such heuristics could be used to supplement the above object-space definition of an RCF.

4 Selective Refinement Algorithm

4.1 Algorithm

The selective refinement algorithm proceeds in two stages:

1. **Region Expansion**
   We construct a list, X, of refinement regions ordered by increasing birth resolution by performing a partial traversal of the DAG. X must contain sufficient regions to permit a complete surface to be extracted from these regions such that the surface satisfies the RCF. The traversal proceeds by placing R^0 on X and continues until:

   $$R^i \in X \iff (\exists R^j \in X : R^i \in children(R^j), R^j \text{ is not finer than the RCF and } R^i \text{ is not completely finer than the RCF}) \text{ or } \exists R^k \in X : R^i \in ancestors(R^k)$$

   Informally, this means that a) the children of every region in X which does not satisfy the RCF are also in X if they themselves are not completely finer than the RCF and b) the ancestor regions of every region in X are also in X.

   This can be implemented by placing R^0 on X and then performing *RegionExpand* on each element in X, in coarsest to finest order, where the *RegionExpand* routine is defined below.

```
Function: RegionExpand
  In:      R^i: region which may be expanded
  In/Out:X: expanded regions list
begin
  if R^i is not finer than the RCF then
    for each child, R^j, of R^i do
      if R^j is not completely finer than the RCF then
        X := X + R^j + ancestors(R^j)\X
      endif
    endfor
  endif
end
```

2. **Surface Extraction**
   A surface is extracted from X by traversing its members in order of decreasing resolution and applying an operation *Infill*(R^i) to each region R^i, where *Infill*(R^i) is defined as:

$$\hat{\mathcal{T}} := \hat{\mathcal{T}} + \{t \in T^i_{post} : I(t) \cap I(\hat{\mathcal{T}}) = \emptyset\}$$

   We will show in Section 4.3 that the region traversal order guarantees that *Infill*(R^i) is never called when triangles in T^i_{post} partially overlap $\hat{\mathcal{T}}$ and hence that a complete surface is extracted.

   Our implementation of *Infill* operates as follows. We associate an *edgelist*, L^i, with each refinement region R^i. During the extraction process, each L^i contains a set of directed edges which is a subset of E^i_{post}. L^i and the boundary of R^i form a complete chain of directed edges which bound the area within T^i_{post} which remains to be extracted. The rules applied by *Infill* are:

   (a) If L^i is empty and either $children(R^i) = \emptyset$ or no member of $children(R^i)$ has previously been encountered then R^i is a "starting point" around which the surface will be extracted and T^i_{post} is added to the current extracted surface, $\hat{\mathcal{T}}$.

   (b) When a boundary edge, e, of a refinement region is extracted then e, directed appropriately, is added to L^j where j is such that R^j is the parent of R^i whose E^j_{post} contains e.

   (c) If L^i is not empty then the triangles in T^i_{post} bounded by L^i and the boundary of R^i are added to $\hat{\mathcal{T}}$.

4.2 Algorithmic Complexity

The complexity of the above algorithm is $O(m+|\hat{\mathcal{T}}|)$ where m is the number of refinement operations contained in the input to the algorithm and $|\hat{\mathcal{T}}|$ is the number of triangles in the extracted surface. This arises from the observations that the region expansion step can expand at most m regions and that the surface extraction process extracts exactly $|\hat{\mathcal{T}}|$ triangles. Thus the order of the algorithm depends on m, which is a factor introduced by the approximation process which generates the input triangulations. The approximation method which is used by our implementation (described in Section 6) generates refinement operations such that $m \approx \frac{1}{2}|V^m|$, where $|V^m|$ is the number of vertices in the highest resolution triangulation in the input.

4.3 Proofs of Extracted Surface Properties

Proofs of the following theorems are presented:

- The extracted surface is complete.
- The extracted surface satisfies the RCF throughout the domain of the terrain.
- The extracted surface is the smallest set of triangles which can form a complete surface which satisfies the RCF.

- If the given triangulations $\mathcal{T}^0, \ldots, \mathcal{T}^m$ satisfy the Delaunay criterion then the extracted surface will also satisfy this criterion.

Lemma 4.1 $\forall p \in \Omega, \exists t \in T^i_{post}$ s.t. $R^i \in X$ and $deathres(t) \geq RCF(p)$, i.e. the specification of X ensures that for every point in the domain there is a triangle which contains that point, is finer than the RCF and is in the regions on X.

Proof: The definition of X on page 6 states that for every triangle t in a region R^i on X such that t is not finer than the RCF, we can find another region with a higher birth resolution whose domain intersects t, i.e.

$$\forall t \in T^k_{post} \text{ where } R^k \in X \text{ and } \exists p \in D(t) \text{ s.t. } deathres(t) < RCF(p),$$
$$\exists R^i \in X \text{ s.t. } k < i \text{ and } birthres^i = deathres(t) \text{ and } D(R^i) \cap D(t) \neq \emptyset$$

Hence by induction, $\forall p \in D(\mathcal{T}^0), \exists t \in T^i_{post}$ s.t. $R^i \in X$ and $deathres(t) \geq RCF(p)$.

Lemma 4.2 On entry to *Infill*(R^i), no triangles of T^i_{post} partially cover those in $\hat{\mathcal{T}}$, i.e. $I(T^i_{post}) \cap I(\hat{\mathcal{T}}) = I(T^i_{post} \cap \hat{\mathcal{T}})$

Proof: This is trivially true before the first call to *Infill* since $\hat{\mathcal{T}} = \emptyset$ at this point.

Assume true on entry to *Infill*(R^j) for some j, $0 < j \leq m$.

We must now prove that, on entry to the next call to *Infill*, which will be made for the region R^r where $R^r \in X$ and $\nexists l : r < l < j$ s.t. $R^l \in X$, it is true that $I(T^r_{post}) \cap I(\hat{\mathcal{T}}) = I(T^r_{post} \cap \hat{\mathcal{T}})$.

- If $I(T^r_{post}) \cap I(\hat{\mathcal{T}}) = \emptyset$, then this is indeed true.
- If $I(T^r_{post}) \cap I(\hat{\mathcal{T}}) \neq \emptyset$, then $\forall t \in T^r_{post} : I(t) \cap I(\hat{\mathcal{T}}) \neq \emptyset$, $\exists R^k \in X$ s.t. $R^k \in descendants(R^r)$ and $I(t) \cap I(R^k) \neq \emptyset$

$$\begin{aligned}
\text{Hence } I(T^r_{post}) \cap I(\hat{\mathcal{T}}) &= I(T^r_{post}) \cap I(\cup\{R^s \in X : R^s \in descendants(R^r)\}) \\
&= I(T^r_{post}) \cap I(\cup\{R^s \in X : R^s \in children(R^r)\}) \\
&= I(T^r_{post}) \cap I(\cup\{T^s_{pre} : R^s \in X \text{ and } R^s \in children(R^r)\}) \\
&= I(\{t : t \in (T^r_{post} \cap T^s_{pre}), \forall R^s \in X \text{ s.t. } R^s \in children(R^r)\}) \\
&= I(T^r_{post} \cap \hat{\mathcal{T}})
\end{aligned}$$

Corollary 4.1 The order of applying *Infill* to the regions in X, i.e. finest to coarsest, ensures that on entry to *Infill*(R^0), $\hat{\mathcal{T}}$ is a complete triangulation.

Corollary 4.2 $t \in T^i_{post}$ is extracted by $Infill(R^i) \iff I(t) \cap I(\hat{\mathcal{T}}) = \emptyset$ i.e. a triangle is extracted iff no other triangles which overlap its domain have already been extracted.

Theorem 4.1 The surface extraction process outlined above extracts a complete triangulation, $\hat{\mathcal{T}}$, of the domain, Ω.

Proof: The definition of X asserts that X always contains R^0. By Corollary 4.1, on entry to *Infill*(R^0), $I(R^0) \cap I(\hat{\mathcal{T}}) = I(R^0 \cap \hat{\mathcal{T}})$ and $\hat{\mathcal{T}}$ is a complete triangulation. It follows from the definition of *Infill* that *Infill*(R^0) will complete the triangulation over Ω.

Theorem 4.2 The extracted surface satisfies the RCF throughout the domain, i.e.

$$\forall p \in D(t), \exists t \in \widehat{T} \text{ s.t. } deathres(t) \geq RCF(p) \tag{1}$$

Proof: We again use induction to prove this theorem, examining the first call to *Infill* and then a later call to the same function, and hence show that statement (1) is a loop invariant of the iterations on *Infill* in the surface extraction process.

On the first call to *Infill*(R^i), the finest region in X is extracted completely. The definition of X on page 6 proves that (1) is true.

Assume that (1) is true at some stage during the extraction process - after performing *Infill*(R^j), say, for $0 < j \leq m$.

Now we must prove that statement (1) holds after the next call to *Infill*, which will be made for the region R^r where $R^r \in X$ and $\nexists l : r < l < j$ s.t. $R^l \in X$

If we assume that *Infill*(R^r) extracts a triangle t from T^r_{post} which invalidates (1), i.e. $\exists p \in D(t)$ s.t. $deathres(t) < RCF(p)$, then by Lemma 4.1, for such $p \in D(t)$,

$$\exists \bar{t} \in T^s_{post} : R^s \in X, r < s < m \text{ s.t. } p \in D(\bar{t}) \text{ and } \forall q \in D(\bar{t}), deathres(\bar{t}) \geq RCF(q)$$

and hence, by Corollary 4.2, triangle $\bar{t}$ or a refinement of $\bar{t}$ would have already been extracted and this contradicts our assumption that t is extracted by *Infill*(R^r).

Theorem 4.3 $\hat{\mathcal{T}}$ is the smallest triangle set which can be extracted from $R^0, \ldots, R^m$ such that $\hat{\mathcal{T}}$ is a complete triangulation over Ω and every triangle in $\hat{\mathcal{T}}$ is finer than the RCF.

Proof: We will show by induction that a loop invariant of the surface extraction process is that after *Infill*(R^i), $\hat{\mathcal{T}}$ is the smallest triangle set which could be extracted from R^0, ..., R^m such that $\hat{\mathcal{T}}$ satisfies the RCF and completely covers $\bigcup_{j=0}^{j=i} D(R^j)$.

After the first call to *Infill*(R^i), where R^i is the finest region in X, the definition of X implies that R^i must be finer than the RCF but not completely finer. Thus there must be at least one triangle in T^i_{pre} which is not completely finer than the RCF and hence at least one triangle, t, in T^i_{post} which is the only one in $T^0_{post}, \ldots, T^i_{post}$ which satisfies the RCF over $D(t)$. The minimal property of refinement operations (page 4) implies that the smallest set of triangles which can form a triangulation (i.e. are compatible) with t over $\bigcup_{j=0}^{j=i} D(R^j)$ is T^i_{post} and hence the loop invariant holds.

Assume the loop invariant is true after *Infill*(R^j) for some j, $0 < j \leq m$.

Now we must prove that the loop invariant holds after the next call to *Infill*, which will be made for the region R^r where $R^r \in X$ and $\nexists l : r < l < j$ s.t. $R^l \in X$

- If $I(R^r) \cap I(\hat{\mathcal{T}}) = \emptyset$ then this can be treated in the same way as the initial R^i.
- If $I(R^r) \subset I(\hat{\mathcal{T}})$ then no elements of T^i_{post} will be added to $\hat{\mathcal{T}}$ by Corollary 4.2 and therefore the loop invariant remains true.
- If $I(R^r) \cap I(\hat{\mathcal{T}}) \neq \emptyset$ and $I(R^r) \not\subset I(\hat{\mathcal{T}})$ then the definition of *Infill* states that we will extract those triangles t in T^i_{post} for which $I(t) \cap I(\hat{\mathcal{T}}) = \emptyset$. We have already shown that such t will satisfy the RCF (Theorem 4.2) and will be compatible with $\hat{\mathcal{T}}$ (Corollary 4.2). Assume we can replace the triangles in $D(T^r_{post}) \cap D(\hat{\mathcal{T}})$ by triangles from some regions $R^{k_1}, \ldots, R^{k_u}$, $0 \leq k_i < r$, $i = 1, \ldots, u$. This, though, would contradict the minimal property of refinement operations and hence the loop invariant holds.

Theorem 4.4 If the input base triangulation, $\mathcal{T}^0$, and sequence of refinement operations, $\{R^1, \ldots, R^m\}$ satisfies the Delaunay criterion [PS85], then the output triangulation, $\hat{\mathcal{T}}$, also satisfies this criterion.

Proof: In the similar way to the Delaunay-satisfiability proof in [dD95], we note that the vertices in each refinement region $R^i, 1 \leq i \leq m$, cannot have an influence, in terms of the Delaunay criterion, outside that region. Hence when we add triangles to the current triangulation, $\hat{\mathcal{T}}$, these triangles will not affect the Delaunay nature of $\hat{\mathcal{T}}$.

5 Geomorphing

When the above technique is used to render a selectively-refined surface in an animation, the RCF may change between frames. If, for example, the RCF contains view-dependent components, such as those described in Section 3.4, then movement of the viewpoint will induce a change in the RCF. This may result in different selectively-refined surfaces being extracted for successive frames and if these were displayed without modification, the temporal discontinuities would be distracting to the viewer. We therefore require a method of geomorphing between two such surfaces. (A *geomorph* is specified by Hoppe [Hop96] as a "smooth visual transition" between two geometric meshes.)

The geometric correspondence between surfaces extracted according to two different RCFs is not immediately obvious. For example, Figure 3 demonstrates the triangles in a simple selectively-refined surface which are affected by the movement of an AOI. The AOI peak in the RCF is moved towards the viewer between Figures 3(a) and (d) (in which the extracted surface triangles are coloured according to the refinement region of which they are part), with the result that the low-resolution triangles in the foreground cannot satisfy the new RCF and hence some of these must be replaced by higher resolution triangles. Figures 3(c) and (f) highlight the triangles which must be geomorphed, and the retriangulation into which they must be morphed, due to the modification of the RCF. By considering the differences between the set of expanded regions from which the pre- and post-morph surfaces can be extracted, we can identify the geometric correspondence between such surfaces.

We assume that we have two functions, *MorphCoarsen*(R^i) and *MorphRefine*(R^i), which perform a smooth geometric transition between the post-refinement triangles of operation R^i and the pre-refinement triangles of R^i, and the reverse, respectively. A method of implementing such functions was described by Cohen-Or and Levanoni [COL96]. Discrete and scalar attributes associated with the triangles could be interpolated in a similar fashion to that described by Hoppe [Hop96] for the Progressive Mesh approximation format.

The functions *MorphCoarsen()* and *MorphRefine()* can only be invoked if all of the triangles which are the starting points for their transitions exist in the current surface. This governs the overall geomorphing strategy.

We first require a minor addition to the above surface extraction algorithm: if we extract a refinement region completely, as we do in step (2a) on page 7, then the region is regarded as a "starting point" for extraction and it is added to a list, C, of regions which could be coarsened in the event of a suitable geomorph. C is initialised to an empty list at the start of the surface-extracting process.

5.1 Algorithm

The geomorphing algorithm proceeds in two steps (note that lists X and C must have been previously created by the selective refinement algorithm):

1. **Coarsening**
   We repeatedly identify regions in X which have been completely extracted in the current surface and hence are candidates for coarsening.

   On each iteration, we traverse list C in order of decreasing resolution until we find a region $R^i \in C$ which is completely finer than the RCF. We then render the surface by performing the surface extraction part of the selective refinement algorithm (as above) and call *MorphCoarsen*(R^i) when R^i is encountered during the extraction process. Once the visual morph of this region has been completed, we can delete R^i from X so that X now reflects the set of regions from which the displayed surface originates. Note that the action of extracting a surface will have modified the list C in preparation for the next iteration of this coarsening step.

   When no regions in C are completely finer than the RCF, X contains the set of refinement regions which can be traversed by the surface extraction algorithm in order to produce a surface which satisfies the minimum value of the previous and current RCFs at every point in the domain.

2. **Refining**
   We traverse the list of expanded regions, X, from its coarsest element to its finest in order to further expand some of these regions and hence ensure that X contains the set of regions from which a surface can be extracted which will satisfy the new RCF.

If a region $R^i \in X$ is not finer than the RCF then we perform a set of refining morphs which are compatible with the current surface and which result in all of the members of $children(R^i)$ which are not completely finer than the RCF being members of X. For such a region R^i, we create a set containing the elements of $children(R^i)$ which are not completely finer than the RCF as well as the ancestors of these regions which are not already in X. This set of regions is then morphed into the current surface by performing *MorphRefine* on the elements of the set in order of increasing resolution.

The simplicity of the refining step is demonstrated by the following pseudo-code.

```
Function: GeomorphRefining
  In/Out:X: expanded regions list
begin
  for each R^i ∈ X in coarsest to finest order do
    Y := ∅
    for each R^j ∈ children(R^i) do
      if R^j is not completely finer than the RCF and R^j ∉ X then
        Y := Y + R^j + ancestors(R^j)\X
      endif
    endfor
    while Y ≠ ∅ do
      remove R^k from Y where R^k is the element of Y with lowest birth resolution
      X := X + R^k
      extract surface, calling MorphRefine(R^k) when R^k encountered
    endwhile
  endfor
end
```

5.2 Proofs of Geomorphed Surface Properties

We will show that the set of regions in X after a geomorph are those which would have been placed on X if the region expansion step of the selective refinement algorithm had been applied to the same RCF. It follows that a geomorphed surface satisfies the properties identified in Section 4.3.

We assume that the region expansion step of the selective refinement algorithm has been performed for an RCF, RCF_{old}, and that the geomorphing step transforms the set of regions on X into one from which a surface can be extracted which will satisfy RCF_{new}.

Lemma 5.1 The regions in X after the coarsening step are those which would have been placed on X if the region expansion step of the selective refinement algorithm had been performed for a "common denominator" RCF, where this is defined $\forall p \in \Omega$ as:

$$RCF_{cd}(p) = \min(RCF_{old}(p), RCF_{new}(p))$$

Proof: We are trying to verify that after the coarsening step the regions in X satisfy:

$$R^i \in X \iff (\exists R^j \in X : R^i \in children(R^j), R^j \text{ is not finer than } RCF_{cd} \text{ and } R^i \text{ is not completely finer than } RCF_{cd}) \text{ or } \exists R^k \in X : R^i \in ancestors(R^k) \quad (2)$$

We will first prove the necessary case of statement (2), i.e.

$$(\exists R^j \in X : R^i \in children(R^j), R^j \text{ is not finer than } RCF_{cd} \text{ and } R^i \text{ is not completely finer than } RCF_{cd}) \text{ or } \exists R^k \in X : R^i \in ancestors(R^k) \Rightarrow R^i \in X$$

The coarsening step removes regions from X only if they have been placed on list C and are completely finer than RCF_{new}. Hence no region which is not completely finer than RCF_{cd} will be removed from X and thus

$$\exists R^j \in X : R^i \in children(R^j), R^j \text{ is not finer than } RCF_{cd} \text{ and } R^i \text{ is not completely finer than } RCF_{cd} \Rightarrow R^i \in X$$

will remain true.

Also, the definition of the surface extraction "starting point" regions which are placed on C implies that $children(R^i) \cap X \neq \emptyset \Rightarrow R^i \notin C$ and therefore the coarsening step will never remove a region which is an ancestor of another on X. Hence the statement $\exists R^k \in X : R^i \in ancestors(R^k) \Rightarrow R^i \in X$ will remain true.

The sufficient case can be proved by contradiction. Assume that after the coarsening step a region R^i exists in X which is completely finer than RCF_{cd} and $\nexists R^k \in X : R^i \in ancestors(R^k)$. Then R^i will have been placed on list C and, since R^i must be completely finer than RCF_{new}, it will have been removed from X by the coarsening step. This contradicts our assumption.

Theorem 5.1 After the refining step, the set of regions in X are those which would have been placed on X if the region expansion step of the selective refinement algorithm had been applied to RCF_{new}, i.e.

$$R^i \in X \iff (\exists R^j \in X : R^i \in children(R^j), R^j \text{ is not finer than } RCF_{new} \text{ and } R^i \text{ is not completely finer than } RCF_{new}) \text{ or } \exists R^k \in X : R^i \in ancestors(R^k) \quad (3)$$

Proof: The necessary case of statement (3) holds trivially by the definition of the *GeomorphRefining* routine.

Lemma 5.1 demonstrated that statement (2) holds after the coarsening step and hence the sufficient case of statement (3) is also true at that point. The definition of *GeomorphRefining* implies that the only regions which are added to X (if they are not already present) are the children of any region in X which is not finer than the RCF, if they themselves are not completely finer than the RCF, together with their ancestors. Hence the sufficient case will remain true.

6 Results

To prepare a continuous-resolution mesh which could be input to our implementation of the selective refinement algorithm, we used a modification of the *Delaunay selector* approximation method presented in [DP95]. Our modification is that point insertion in a refinement region continues until the global approximation error has been reduced to a value lower than the error before the region was created. Figures 4(a),(b) and 6(a),(b) show the highest resolution triangulations in the resulting approximations of Digital Elevation Model [EC84] datasets of Mt St Helens (WA.) and Emory Peak (TX.).

Figure 4 presents the inputs and outputs of an application of the selective refinement algorithm to the Emory Peak approximation. The complete set of input refinement operations is visualised embedded in resolution space in Figure 4(c). The input RCF which specified an increased resolution in the foreground and background of a particular view is shown in Figure 4(d). The region expansion and surface extraction processes used this RCF to produce the selectively-refined surface which is viewed from its intended viewpoint in Figure 4(e). Its plan view (Figure 4(f)) demonstrates where the output triangle density was increased due to the influence of the RCF. Figure 4(g), which shows the extracted surface's triangles at their individual death resolutions, demonstrates that the surface satisfies the RCF since all of the triangles lie above the RCF in resolution space.

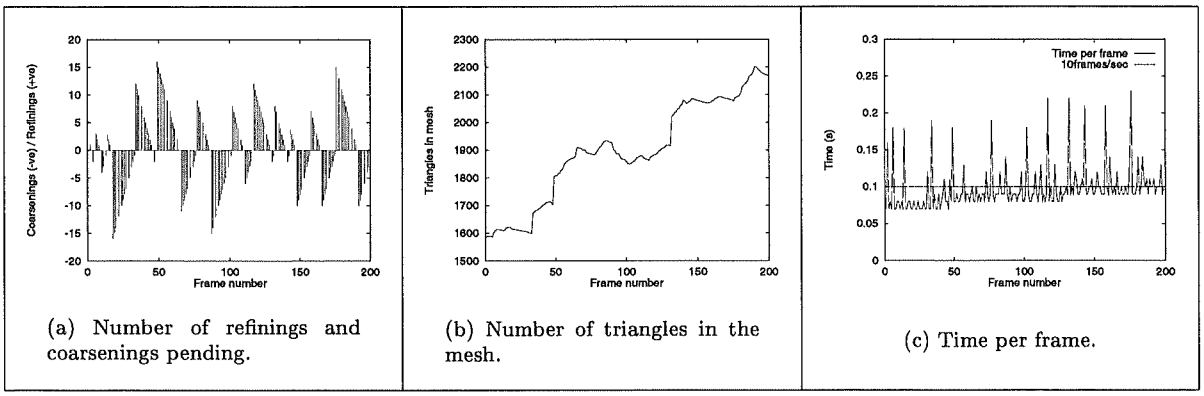
The example RCF discussed in Section 3.4 (Figure 5(e)) was applied to the Mt St Helens approximation to produce a selectively-refined scene (Figure 6(f)) which can be compared with the same view of the highest resolution triangulation (Figure 6(d)). Figures 6(f),(g) and (h) demonstrate that the RCF has ensured that the resolutions of the crater's silhouette, the region in the "Head Up Display" and the foreground of the scene have been enhanced with respect to the surrounding area.

The same RCF was modified to match the view frustum of Figure 6(e), i.e. the view frustum, Head Up Display and viewpoint AOI components of the RCF were moved forward with respect to the original view direction. The output surface from this RCF is viewed as intended in Figure 6(j). Figures 6(k),(l) and (m) demonstrate the differences between this extracted surface and the previous one.

Timings for these applications of the selective refinement algorithm, together with the use of another RCF on the Mt St Helens data, were obtained using OpenGL on an SGI Indigo^2 175Mhz R10000 Solid Impact (Table 1). This demonstrates that the rendering time ("Extraction Time") for each selectively-refined mesh is significantly less than for the same view of the corresponding highest resolution triangulation. Even when the total processing time for each selectively-refined mesh ("Expansion time" + "Extraction Time") is considered, a static scene which meets the user's resolution requirements can be rendered up to 57% faster than a view of the highest resolution triangulation.

Input $\mathcal{T}^m$	Emory Peak dataset (approximated by 6000pts and 11875 triangles)	Mt St Helens dataset (approximated by 7200pts and 14247 triangles)		
View of $\mathcal{T}^m$	Fig 4(a)	Fig 6(d)	Fig 6(e)	Similar to Fig 6(d)
Time to render $\mathcal{T}^m$	0.43s	0.57s	0.56s	0.59s
RCF	Fig 4(d)	Fig 5(e)	Fig 5(e) with reduced view frustum	Viewpoint and crater AOIs
Expansion time	0.27s	0.32s	0.20s	0.20s
$\lvert X \rvert$	702	437	414	236
Extraction time	0.11s	0.07s	0.06s	0.04s
$\lvert \widehat{T} \rvert$	2511	1587	1510	1014

Table 1: Selective mesh refinement applied to various datasets, views and RCFs.



(a) Number of refinings and coarsenings pending.

(b) Number of triangles in the mesh.

(c) Time per frame.

Figure 2: Geomorphing results during a traversal over the Mt St Helens mesh.

The numerical results of geomorphing during a traversal over the Mt St Helens dataset are given in Figure 2. The set of refinement regions which produced the static selectively-refined mesh shown in Figure 6(f) was pre-calculated (in 0.32s). The viewpoint was then stepped forward and the RCF was modified accordingly (as described above). The end-point of the traversal was Figure 6(j).

The numbers of refinement region refinings and coarsenings which were visible and pending during each frame are graphed in Figure 2(a), where the number of refinings pending are on the positive axis and the number of coarsenings on the negative. The "saw-tooth" nature of this graph reflects the sequence of operations described in Section 5, namely that, repeatedly, a sequence of coarsenings was applied, the next set of necessary refinings was determined and then these refinings were applied. The potential for parallelizing the geomorphs has not yet been realised in our implementation and hence the number of morphs pending decreases uniformly. It can be seen that only a small proportion of refinement regions (at most 16, out of over 400) were marked as pending throughout the transition.

The number of triangles in the mesh (Figure 2(b)) was correspondingly increased and decreased by these refinings and coarsenings. This graph has a generally increasing nature because only the first 200 frames of the animation are depicted and the coarsenings marked in the blue region of

Figure 6(l) had yet to take place. The three abrupt increases in the number of triangles in the mesh are due to a number of refinings having been applied simultaneously because they lay outside the current view.

The total time per frame is given in Figure 2(c), together with a line indicating a rendering time which would permit 10 frames/sec. Our rendering time is generally below this line; the peaks above this line occur at the points where the next set of refinings had to be determined. The rendering time of each frame is still significantly less than the average time to render a scene of the animation from the highest resolution triangulation (0.57s).

7 Related Work

Independently, and in parallel with this work, Puppo [Pup96] has developed a theoretical method for meshing a set of single-resolution TINs into a static variable resolution representation of a surface. His algorithm differs from our selective refinement process in that it combines the region expansion and surface extraction steps. This prevents any further operations on the set of expanded regions, such as the above geomorphing technique. Puppo does not present an implementation of his theory.

Cohen-Or and Levanoni [COL96] also describe an approach to selective mesh refinement which is based on merging triangulated subsets of the domain from different levels of detail, and they tackle the temporal continuity problem. They combine De Berg and Dobrindt's work [dD95] with a tree structure to represent the overlaps between refinement regions, rather than the DAG which we have described. Thus the method is limited to the use of terrain approximations which have been produced by decimating Delaunay triangulations. Cohen-Or restricts the depth of the tree structure to only three or four levels which means that a) only a limited number of resolution levels can be accurately represented by an extracted surface; and b) refinement regions containing large numbers of triangles will have to be considered during a geomorphing step.

A limited form of RCF to control the level of detail in an approximation was presented by Gross *et al* [GGS95]. This paper applies a Gaussian filter to the coefficients of a wavelet transform of terrain data in order to enhance the resolution of an AOI. Again, this method is restricted to a specific method of approximating terrain meshes, namely repeated point removal from a quad-mesh.

8 Conclusions and Future Work

A framework has been presented within which an approximation to a terrain can be selectively refined according to a global object-space Resolution Control Function and this selective refinement can then be modified smoothly to adapt to dynamic rendering requirements. The foundation for this technique is a generalised continuous-resolution representation for surfaces which permits our selective refinement method to be applied to most existing single-resolution terrain approximating schemes.

We have presented results which demonstrate the efficiency of our scheme. Our current implementation was not primarily designed for speed, yet it can render a flight over a geomorphing selectively-refined surface at almost interactive frame rates. There are opportunities for performance enhancements in a number of areas, such as constraining geomorphing considerations to the subgraph of the refinement regions' DAG which relates to the current scene; and parallelisation of independent geomorphs.

Other areas of future work include extending our geomorphing technique of post-processing a previously expanded set of refinement regions to other simulation tasks such as database transmission and dynamic terrain effects. Also, Hoppe's progressive mesh approach [Hop96] can be utilised by our selective refinement framework for terrain rendering, but if a PM representation of a 3D model is to be selectively refined in this way then an adaptation of the RCF concept is required.

References

[CMR90] Michael A. Cosman, Allen E. Mathisen, and John A. Robinson. A new visual system to support advanced requirements. In *IMAGE V Conference*, pages 370–380. Image Society Inc, June 1990.

[COL96] Daniel Cohen-Or and Yishay Levanoni. Temporal continuity of levels of detail in Delaunay triangulated terrain. In *7th IEEE Visualization Conference*. IEEE, October 1996.

[CPS95] P. Cignoni, E. Puppo, and R. Scopigno. Representation and visualization of terrain surfaces at variable resolution. In R. Scateni, editor, *Proceedings International Symposium on Scientific Visulization*, September 1995.

[dD95] Mark de Berg and Katrin T. G. Dobrindt. On levels of detail in terrains. In *11th ACM Symposium on Computational Geometry*, pages 26–27, June 1995. Also published in longer version as *Technical Report* UU-CS-1995-12, Utrecht University, Dept. of Computer Science, April 1995.

[Dou86] David H. Douglas. Experiments to locate ridges and channels to create a new type of digital elevation model. *Cartographica*, 23(4):29–61, 1986.

[DP95] Leila De Floriani and Enrico Puppo. Hierarchical triangulation for multiresolution surface description. *ACM Transactions on Graphics*, 14(4):363–411, 1995.

[EC84] Atef A. Elassal and Vincent M. Caruso. USGS digital cartographic data standards - Digital Elevation Models. Technical Report Geological Survey Circular 895-B, US Geological Survey, 1984.

[FEKR90] R. L. Ferguson, R. Economy, W. A. Kelly, and P. P. Ramos. Continuous terrain level of detail for visual simulation. In *IMAGE V Conference*, pages 144–151. Image Society Inc, June 1990.

[FL79] Robert J. Fowler and James J. Little. Automatic extraction of irregular network digital terrain models. In *SIGGRAPH 79*, Computer Graphics Proceedings, Annual Conference Series, pages 199–207. ACM SIGGRAPH, 1979.

[FZPM93] John S. Falby, Michael J. Zyda, David R. Pratt, and Randy L. Mackey. NPSNET: Hierarchical data structures for real-time three-dimensional visual simulation. *Computers and Graphics*, 17(1):65–69, 1993.

[GGS95] M. H. Gross, R. Gatti, and O. Staadt. Fast multiresolution surface meshing. In *6th IEEE Visualization Conference*. IEEE, October 1995.

[HG95] Paul S. Heckbert and Michael Garland. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, Carnegie Mellon University, August 1995.

[Hop96] Hugues Hoppe. Progressive meshes. In *SIGGRAPH 96*, Computer Graphics, Annual Conference Series, pages 99–108. ACM SIGGRAPH, August 1996.

[Kle90] James A. Kleiss. Terrain visual cue analysis for simulating low-level flight: A multidimensional scaling approach. In *IMAGE V Conference*, pages 60–67. Image Society Inc, June 1990.

[LDW94] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. Technical Report TR 93-10-05b, Dept. of Computer Science and Engineering, University of Washington, January 1994.

[LKR+96] Peter Lindstrom, David Koller, William Ribarsky, Larry F. Hodges, Nick Faust, and Gregory Turner. Real-time, continuous level of detail rendering of height fields. In *SIGGRAPH 96*, Computer Graphics, Annual Conference Series, pages 109–118. ACM SIGGRAPH, August 1996.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction.* Springer Verlag, 1985.

[Pup96] Enrico Puppo. Variable resolution terrain surfaces. In *8th Canadian Conference on Computational Geometry*, August 1996.

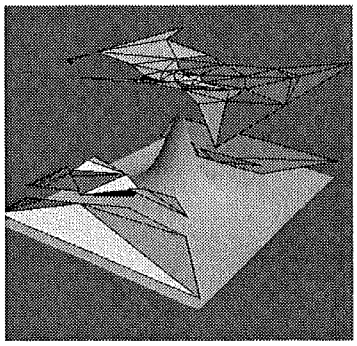
[RB93] Jarek Rossignac and Paul Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics: Methods and Applications*, pages 455–465. Springer-Verlag, June 1993. Also available as IBM Research Report RC 17697 (# 77951), Yorktown Heights, NY 10598, February 1992.

[Sch83] Bruce J. Schachter. *Computer Image Generation.* John Wiley and Sons, 1983.

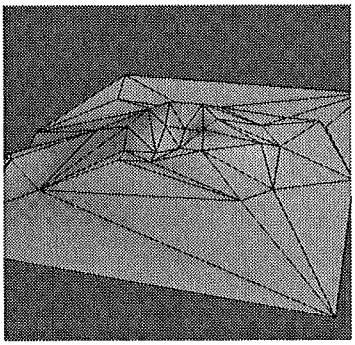
[SP92] Lori Scarlatos and Theo Pavlidis. Hierarchical triangulaton using cartographic coherence. *CVGIP: Graphical Models and Image Processing*, 54(2):147–161, March 1992.

[TB94] David C. Taylor and William A. Barrett. An algorithm for continuous resolution polygonalizations of a discrete surface. In Wayne A. Davis and Barry Joe, editors, *Graphics Interface 94*, pages 33–42. Canadian Human-Computer Communications Society, May 1994.

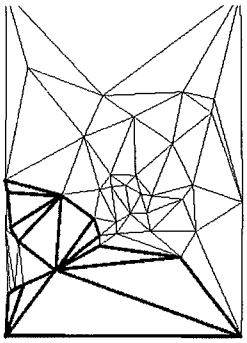
[Tur92] Greg Turk. Re-tiling polygonal surfaces. In *SIGGRAPH 92*, Computer Graphics Proceedings, Annual Conference Series, pages 55–64. ACM SIGGRAPH, July 1992.



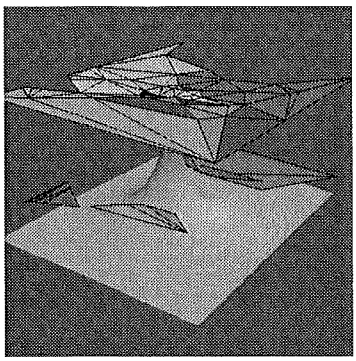
(a) Pre-morph RCF together with the extracted triangles at their death resolutions.



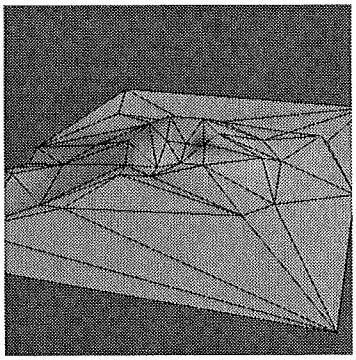
(b) Pre-morph perspective view of the extracted surface.



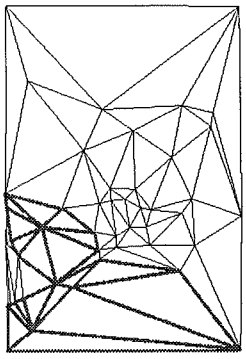
(c) Pre-morph aerial view; the triangles which will be refined during the geomorph are highlighted.



(d) Post-morph RCF together with the new set of extracted triangles at their death resolutions.

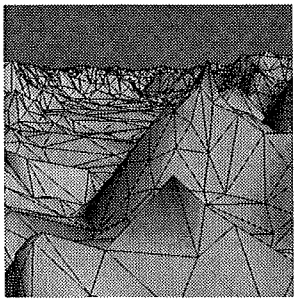


(e) Post-morph perspective view of the new extracted surface.

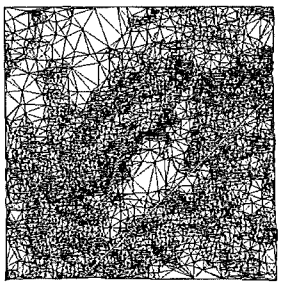


(f) Post-morph aerial view; the triangles which were refined during the geomorph are highlighted.

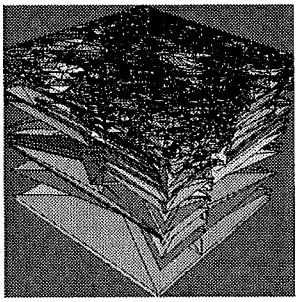
Figure 3: Simple demonstration of the operation of a geomorph on a low-resolution representation of the Mt St Helens dataset. The geomorph was caused by the movement of an AOI in the RCF.



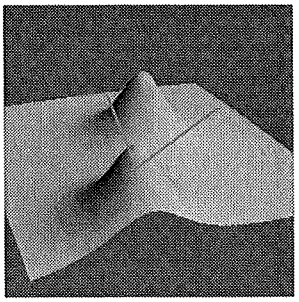
(a) Perspective view of highest resolution triangulation in 6000pt approximation (11875 triangles).



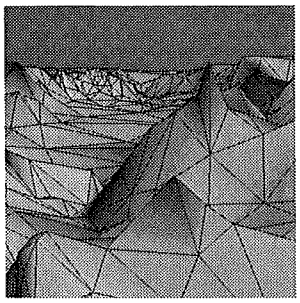
(b) Highest resolution aerial view indicating the extent of the view frustum in scenes (a) and (e).



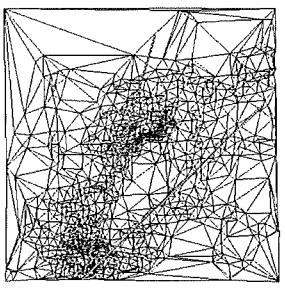
(c) Post-refinement triangles of all refinement operations in 6000pt approximation, in resolution space.



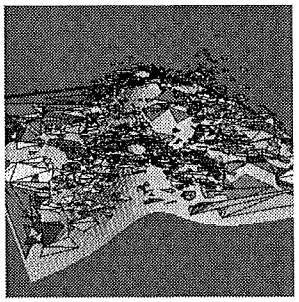
(d) RCF used to generate the selectively-refined surface in (e); the extent of the view frustum is indicated.



(e) Selectively-refined version of scene in (a).

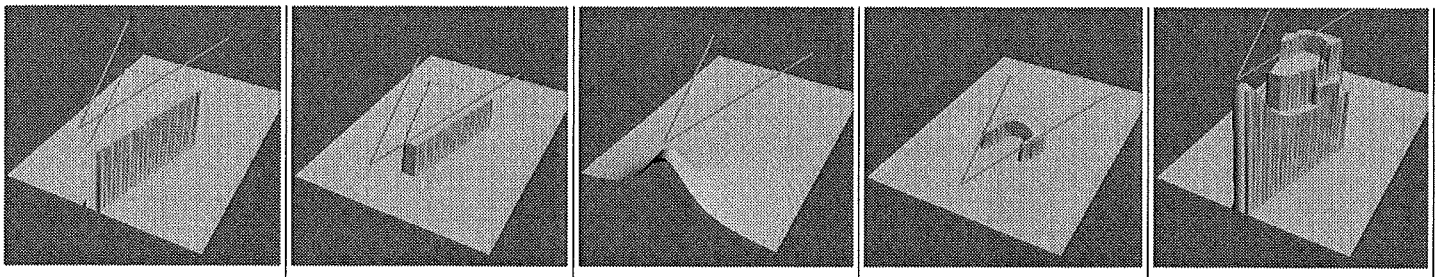


(f) Aerial view of the surface extracted to produce (e).



(g) RCF, together with the extracted triangles from (e) at their death resolutions, in resolution space.

Figure 4: Emory Peak selective refinement example inputs - (a) to (d) - and outputs - (e),(f),(g).



(a) View frustum component.

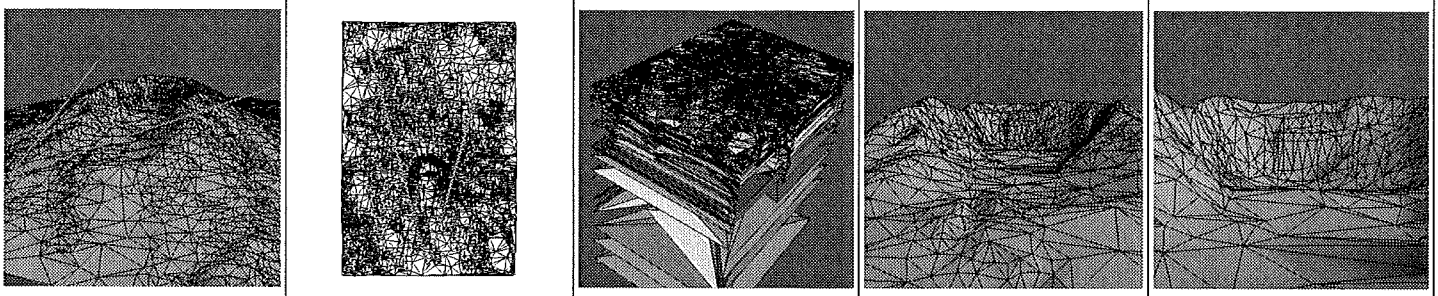
(b) 'Head Up Display' component.

(c) Area Of Interest centred on the viewpoint.

(d) Critical lines' enhancement component.

(e) RCF obtained by combining these components.

Figure 5: Components of the RCF used to produce the selectively-refined view in Figure 6(f). The orange lines denote the extent of the view frustum in the xy-plane.



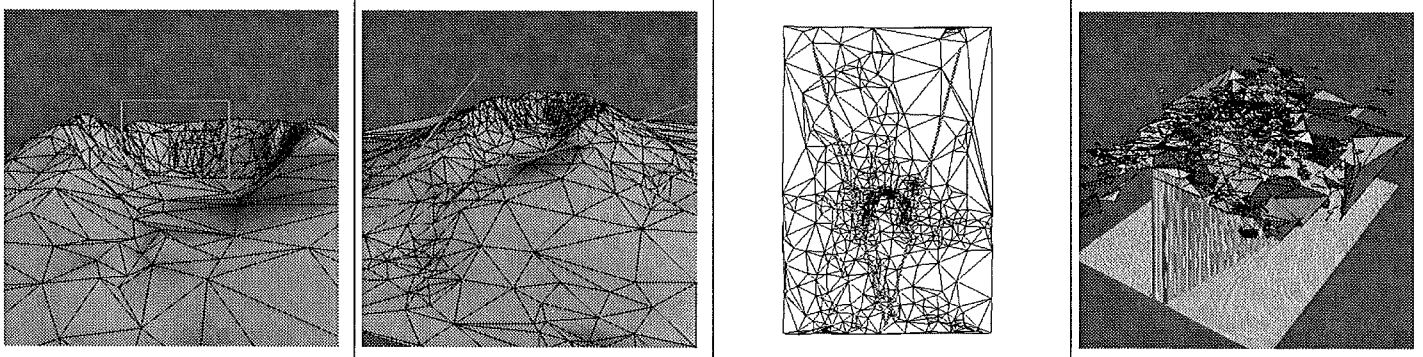
(a) Highest resolution triangulation in 7200pt approximation; the view frustum for (d) and (f) is illustrated.

(b) Highest resolution aerial view indicating the extent of the view frustum in scenes (d) and (f).

(c) Post-refinement triangles of all refinement operations in 7200pt approximation, in resolution space.

(d) A view of the crater from the highest resolution triangulation in (a).

(e) Similar view of (a) with the viewpoint moved forward.

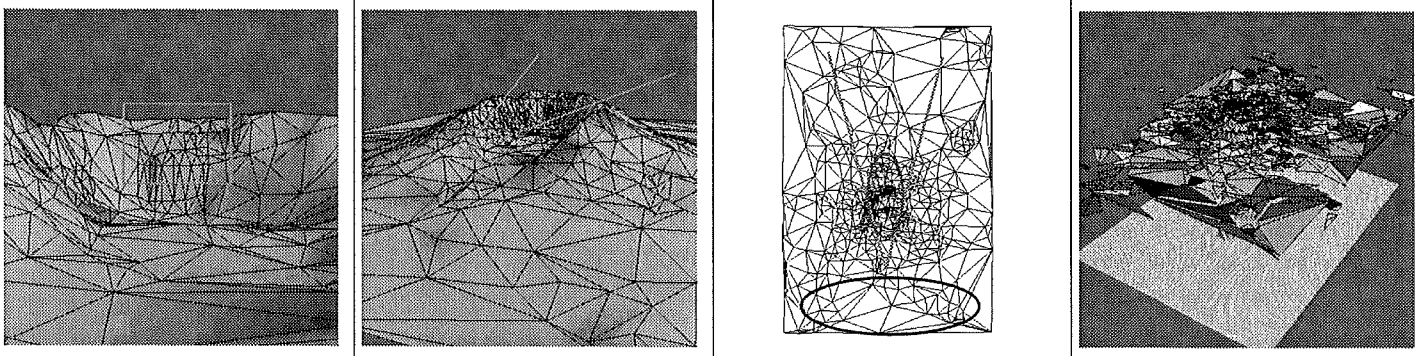


(f) Selectively-refined version of scene in (d); the simulated Head Up Display screen region is indicated.

(g) Perspective view of the surface extracted to produce (f).

(h) Aerial view of (g).

(i) RCF used to produce (f), together with the triangles from (f) at their death resolutions, in resolution space.



(j) Selectively-refined version of scene in (e); the simulated Head Up Display screen region is indicated.

(k) Perspective view of the surface extracted to produce (j).

(l) Aerial view of (k) for comparison with (h); note coarsening in the blue region and refining in the red region.

(m) RCF used to produce (j), together with the triangles from (j) at their death resolutions, in resolution space.

Figure 6: Views of a single-resolution approximation of Mt St Helens ((a)-(e)) and two sets of selectively-refined outputs.