MANAGING COMPLEXITY AND FEATURE PLACEMENT IN PLANETARY TERRAIN SYNTHESIS

Alan Murta, James Miller and Simon Embley
Advanced Interfaces Group
Department of Computer Science
University of Manchester
Manchester M13 9PL, UK.
amurta@cs.man.ac.uk

ABSTRACT

This paper presents a case study in the generation of a planetary terrain model. Several novel techniques were developed as part of the work. A new view-dependent surface tiling method for representing spherical bodies is first described. Secondly, a simple method which uses image-driven modulation of fractal functions to support the integration of synthetic surface detail with planned topographical features is presented. Finally a deterministic method for the incorporation of randomly placed discrete structures is outlined. A model of the lunar surface is presented as an illustration of these techniques.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.5 [**Computer Graphics**]: Object Modeling; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism.

Additional Key Words and Phrases: Fractals, natural phenomena, terrain models, sphere tiling.

1. INTRODUCTION

The representation of natural forms such as trees, clouds and landscapes is a challenging area of computer graphics modelling. Whereas man-made objects may be described using relatively simple Euclidean shapes, natural forms tend to exhibit considerable geometric complexity with intricate structure present at a wide range of scales. Rather than attempting to represent the entirety of such forms in rigorous geometric detail, current approaches attempt to model the underlying *character* of the objects under study. In particular, fractal methods have been employed widely in the characterisation of natural complexity [Mande82].

A variety of procedural techniques have been developed to reproduce the forms described by such fractal models. This is particularly so in the context of generating synthetic landscapes [Voss85]. Some approaches are based upon the recursive subdivision and controlled displacement of initially flat regions [Fourn82], while others create surface roughness via the summation of multi-dimensional noise functions [Musgr93]. Some can only be used to model finite areas [Lewis87], whereas others may allow the demand-driven synthesis of potentially infinite terrains [Saupe89]. Certain techniques are limited to the generation of landscapes with planar topologies, while others are more general and may be used to apply terrain-like features to the surfaces of spheres and other solid bodies [Perli94].

In almost all cases, the terrain modelling techniques described in the literature have been evaluated only in the context of producing completely synthetic scenes which possess a limited set of topographical features. Very little has been published on how procedural methods may be combined with other techniques to produce feature-rich landscapes on a large scale. This paper will consider the challenges of generating such models.

1.1. Objectives

The task of generating a detailed graphical representation of an entire planetary surface is to be addressed. The production of such a model poses several research problems. First, a suitable method of representing the surface geometry must be determined. This requires a mesh based approach if standard polygonal rendering methods are to be used. The method of mesh decomposition should be appropriate to the tiling of spherical bodies, and should support the representation of many levels of detail to allow surface viewing over a wide range of distances.

Producing a representation of a known planetary body, rather than a purely synthetic terrain with an arbitrary arrangement of features, requires a method of controlling the surface generation process. This paper will describe an approach to modulating the behaviour of a fractal generator in order to produce terrains with planned large-scale structure.

Real landscapes are likely to contain a number of distinct geological features which form part of the terrain itself (such as mountains, impact craters and volcanic domes) or items which lie on the surface (rocks, trees, etc.). A research objective is to consider effective methods for the incorporation of such discrete structures within large scale models.

2. SURFACE TILING

An established method of tiling a spherical body is to begin with an octahedron, icosahedron or dodecahedron and to apply a triangle based subdivision to each component face [ORour97]. However, such decompositions are ill-suited to the representation of spherical surfaces when viewed from extremely close up: extensive facet subdivisions would be required to adequately represent nearby surface detail, while naive attempts at mesh simplification by merging polygons in more distant surface regions may introduce unwanted distortions or tears.

This work introduces a new method of sphere tiling which may be employed over a wide range of viewing distances. Level of detail efficiencies are supported through the use of a *view-dependent* surface decomposition: nearby features are naturally represented by small tiles, while more distant surface parts are made up of larger primitives. Similar view-dependent decomposition techniques have recently become popular in the representation of digital terrain maps [Linds96].

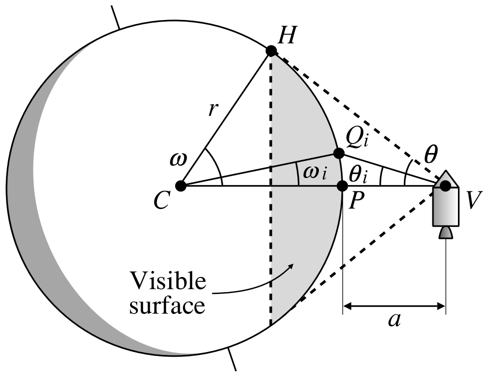

Figure 1: Viewing geometry

The tiling method is as follows: Assume the viewer is located at V, at an altitude a km above a point P on the surface of the sphere (figure 1). Only a limited portion of the sphere's surface is visible from any given viewpoint, corresponding to the shaded region in the figure. When the sphere is viewed from a very large distance this visible region covers almost an entire hemisphere; at low viewing altitudes it resembles a upturned circular bowl which stretches from horizon to horizon. Let θ represent the apparent angular elevation of the viewer's horizon H, measured with respect to the line VP. It may be calculated as

$$\theta = \arcsin\left(\frac{r}{(a+r)}\right) \quad (1)$$

where r is the sphere radius.

The visible surface is the only portion of the sphere requiring decomposition for a given view position. It is first divided into a number of concentric zones centered on P (figure 2). The zone boundary radii are chosen so that all zones project an equal *apparent* width as seen from the viewer's position. The *actual* width of each zone will increase towards the outer edge of the visible surface, due to the combined effects of perspective foreshortening and the sphere's surface curvature. The increasing zone size will result in the use of larger tiles near the horizon, thereby providing a significant reduction in the total polygon count.

The zone spacing is calculated by splitting the horizon angle θ into n equal parts. The view-

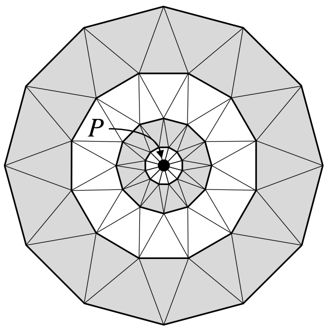

Figure 2: Tiling the visible surface

centred angle θ_i of the i^{th} subdivision at Q_i is simply

$$\theta_i = \frac{i\theta}{n} \tag{2}$$

This must be converted into a sphere-internal angle ω_i, so that surface tiling coordinates may be calculated:

$$\omega_i = \arcsin\left(\frac{(a+r)\sin\theta_i}{r}\right) - \theta_i \tag{3}$$

Each zone boundary is then divided into equal parts using a fixed number of azimuthal samples, as shown in figure 2. The application of a skew to samples on every alternate zone boundary prevents the occurrence of visually distracting radial edges in the tiling. In this example the number of tiles is set artificially low for illustrative purposes (four zones with twelve samples per zone). In a real planetary surface model around 600 zones each with 800 azimuthal samples would typically be required for a detailed terrain representation. The surface images presented later in this paper were generated using a tiling of this complexity.

3. INCORPORATING TOPOGRAPHICAL FEATURES

With a sufficiently fine polygonal decomposition and an interpolated shading model, the tiling method described will result in a smooth sphere representation. Some surface roughness must be added if a planetary terrain is to be modelled. The work presented here uses a spectral synthesis method of terrain modelling [Musgr93]. Each surface vertex is displaced in a radial direction, with the amplitude of individual displacements being controlled by a 3D noise function [Perli85]. Successive additions of noise to the surface (with noise wavelength and amplitudes being reduced on each iteration) will result in a surface character similar to that found in real terrains.

3.1. Terrain modulation

The above surface generation technique requires modification for it to support the production of terrain models which incorporate planned topographical structure. The moon, for example is not a simply a roughened sphere – it exhibits a differentiated surface of mountain ranges, volcanic flood plains and other geological features arranged in a known pattern.

A simple method of local control in the fractal generation process is described here, which permits the controlled synthesis of a variety of surface types within a single terrain model. It requires the use of an external 'map' (2D image) which contains a pictorial encoding of the various land types and their arrangement. A Mercator, or similar mapping projection relates mesh vertex locations to points in the image. The map code corresponding to each mesh vertex is consulted during the surface displacement process, and its value may be used to control the behaviour of the fractal generator. For example, surface regions which associate with 'mountain' encodings in the map may be encouraged to exhibit additional roughness and given a predisposition to rise above average terrain levels.

The approach is clearly related to the techniques of image-driven texture, bump or displacement mapping which are used widely in the graphics community to modulate surface characteristics [Heckb86]. It also shares similarities with Musgrave's 'multifractal' terrain generator [Musgr94]. This is a fractal synthesis technique which applies automatic regional adjustments to modelling parameters, resulting in artificial landscapes which may feature both rugged mountains and smooth valley floors.

4. STOCHASTIC FEATURE PLACEMENT

The model described so far consists of a continuous fractal generator with regional control over topographical properties. Realistic environments are likely to feature a number of essentially discrete structures which occur at 'random' locations. Craters provide an example of a discrete structure in a lunar surface model. Other environments may have equivalent features, such as trees, cities and rocks.

A method for the stochastic generation and placement of discrete structures is presented here. Any feature generator of this kind should promote *statistical integrity* – the creation of structure populations with properties which are consistent with those found in nature, *population management* – suppressing the representation of unnecessary features, and *repeatability* which ensures deterministic behaviour.

4.1. Statistical integrity

Populations of discrete structures are generally subject to a number of empirical properties which govern the characteristics of individual members. Lunar craters, for example, all have diameters which fall within a finite range of permitted values. Crater populations also exhibit a geometric relationship between profusion and scale. It has been observed that within a crater population, the number of craters with an area μ greater than some chosen value A generally conforms to the rule

$$Num(\mu > A) \propto \frac{1}{A} \tag{4}$$

i.e. there are many more small craters than large ones [Voss88]. Many other entities (vegetation, fauna, stars etc.) exhibit similar profiles relating population to component scale.

Crater populations, like other phenomena, additionally feature a correlation between shape and overall size – large craters tend to have shallower profiles compared to smaller ones [Rükl90]. The visual properties of entities within a simulated population must be statistically consistent with reality if a model is to be convincing.

4.2. Population management

The tiling method of section 2 provides a surface decomposition with a complexity appropriate to the amount of detail which is discernible from a given viewpoint. Economical management of the discrete structure generation process is another valuable way of limiting modelling complexity. This will typically involve measures to support the introduction and removal of small scale features in accordance with changes in viewing position.

In a lunar model, for example, it would be unreasonable to attempt to represent every single crater, boulder and pebble on the moon's visible surface when observing from an altitude of several thousand kilometres. It is desirable to include features of this kind within low altitude views, however.

4.3. Repeatability

The form and placement of stochastically generated features within an environment should not be completely haphazard. For example, structures which appear within a distant terrain view should also be present when a close-up view of the scene is computed. To achieve a distribution which appears indiscriminate, yet supports the perpetuation of repeatable feature arrangements under arbitrary model generation scales, requires careful control over the random generation functions used.

4.4. Discrete structures model

A deterministic method for the efficient, statistically controlled generation and placement of discrete structures is presented here. The specific process of applying random crater populations to a planetary surface is described. However, the technique may be adapted to support the generation and positioning of any feature population within an arbitrary n-dimensional space.

The planetary surface generation method is as follows: A bounding box for the visible surface region is first calculated in the latitude, longitude coordinate system, as depicted in figure 3(a). Within this bounding box a grid hierarchy is then constructed. Grid line positions are chosen so that each lies on a rational fraction of the latitude, longitude coordinate system units. The coarse top-level grid – shown using bold lines in figure 3(b) – is used for the placement of large-scale features. In the lunar model, craters with radii in the range r_{max} (the largest possible crater size) to $r_{max}/2$ may potentially be generated at these grid points. In the figure these are portrayed as large light-grey circles. The next level in the hierarchy is a 50% smaller grid (thin lines) on which craters with radii in the range $r_{max}/2$ to $r_{max}/4$ may be placed (medium-grey circles). A further grid generation is depicted using dotted lines and dark-grey circles. The hierarchy may be continued to successively finer levels to allow the plotting of ever-smaller features.

The use of a regular grid to control feature placement appears to be at variance with the 'random distribution' philosophy. However, it is known that regular structure is difficult to perceive when grid-points are sampled aperiodically. This phenomenon is exemplified within error-diffusion based methods for digital halftoning, in which dot placement is grid based, yet an overall

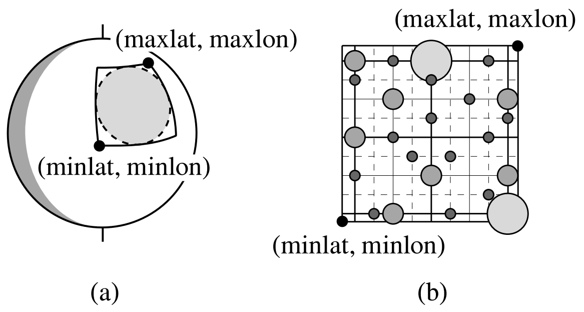

Figure 3: Grid based plotting of random features

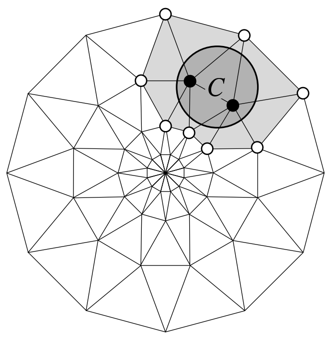

Figure 4: Adding a crater

random character is exhibited [Ulich88].

Each intersection point (*lat*, *lon*) in the grid represents a potential crater plotting location. The decision whether or not to plot at a given point is made by first hashing the grid coordinate onto a pseudo-random number generator seed value. If `random` is a random number function which returns values in the range [0, 1], and `seed` sets the generator state given an argument in the same range, the following routine may be used as the hashing function:

```
void 2D_seed(double lat, double lon)
{
  double r1, r2;
  seed(K1 * lat); r1= random();
  seed(K2 * lon); random(); r2= random();
  seed(K3 * (r1 + r2));
}
```

The constants `K1`, `K2` and `K3` are used to scale the seed arguments to the required range. Note that *two* `random` calls are made following the longitude seeding to preclude reflectional symmetry about the `lat` = `lon` axis. Alternative methods of random seed hashing can be found in [Peach94]. Once the generator has been seeded a random number is produced, and its value is compared against a probability threshold to steer the crater plotting decision:

```
2D_seed(lat, lon);
if (random() > THRESHOLD * cos(lat))
  plot_random_crater(lat, lon);
```

The cosine factor is included to compensate for the bunching of longitude lines in the polar regions, thereby maintaining a constant crater probability density over the entire globe.

In the lunar model, the decision to plot a crater in a given location results in the following behaviour. First, the surface vertices which lie within the crater disc C are determined (figure 4). The radial distance between each included vertex and the crater centre is then calculated, and used to index a table based crater profile function. This returns a vertical displacement value which is applied to the vertex. The net result is to modulate the terrain over the shaded region. The coarse tiling illustrated will result in an acute under-sampling of the crater profile; this is less of an issue when finer mesh decompositions are used.

5. EVALUATION

The techniques described have been used in the construction of a surface model of the earth's moon. A coarse hand-drawn map of the major lunar features was used to modulate the generation of surface topography, as outlined in section 3.1. The map contained four basic region types representing: *Mare* – relatively flat lowland regions; *Rilles* – valley-like formations; *Terra* – areas of moderate roughness, and *Montes* – mountainous regions exhibiting large-scale irregularity. The map was additionally used to control surface colour properties; Mare regions being darker than other terrain types.

A random crater population was applied to the surface using the stochastic placement methods described in section 4. The lunar feature map was also employed within this process to introduce regional variations in plotting probability thresholds. This promoted sparse crater densities in Mare regions (which are 'young' in geological terms) compared with the older and more heavily cratered Terra regions.

In order to improve the visual authenticity of the lunar model, a database containing the locations, sizes and types of approximately one thousand of the moon's more prominent craters was used to supplement the stochastic crater population.

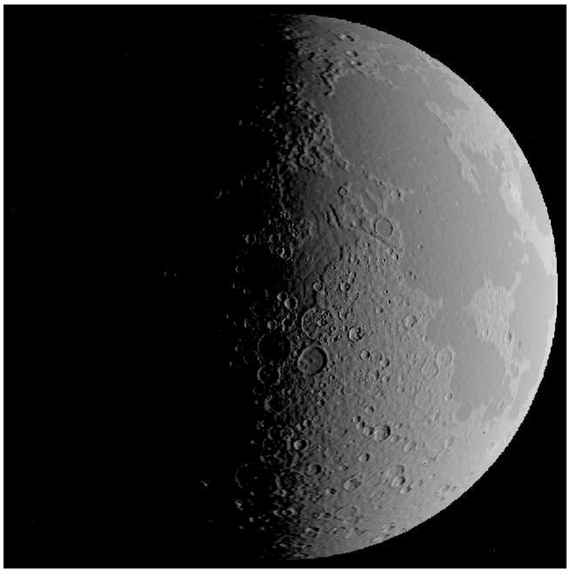

Figure 5: The lunar model at first quarter

Figure 5 shows a view of the lunar model from an altitude 3,500km above the surface. Many of the large craters visible from this distance are sourced from the crater database, with smaller craters being generated using the stochastic placement algorithm. Darker, smoother and lightly cratered Mare regions are visible in the upper-right of the image.

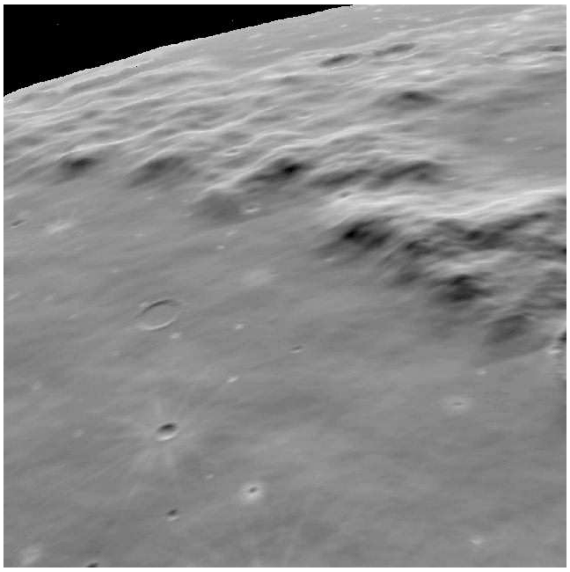

Figure 6: Low altitude view

A close-up of the synthetic lunar surface is shown in figure 6. It presents an eastward view of the Apenninus mountain range from an altitude 100km above the Mare Imbrium. Most of the craters appearing in this image are generated stochastically. Some craters have associated ray systems which are rendered using scaled texture maps. In this oblique terrain view much of the generated visible surface disc lies outside the view frustum.

5.1. Performance

This approach to terrain modelling is considered viable for the off-line generation of animation frames. Model generation times scale almost linearly with the number of vertices (V) used in the surface decomposition. If t_n is the per vertex cost of computing the surface noise displacement, F is the number of random features which appear at a given level of detail, and t_f is the per vertex cost of feature addition

$$Generation\ time \propto V(t_n + F.t_f) \tag{5}$$

A tiling of 600 zones each split into 800 samples (480,000 vertices) takes around one minute to compute on a 200MHz R4400 system. Rendering the surface mesh may be accomplished in around one second on a high-end graphics workstation. Since a mesh is produced specifically for a given view *position*, changes in view *direction* or lighting arrangements do not require the surface model to be regenerated. This can lead to significant compute time savings within animation sequences. Minor changes in viewer altitude, latitude and longitude without recomputation of the surface mesh may also be tolerated before economies in the modelling method become noticeable (discerning the rim of the finite modelled surface, or perceiving level of detail deficiencies, for example).

6. SUMMARY

This paper has described a variety of techniques which support the generation of complex terrain models. Two major themes have pervaded throughout the work: managing model complexity, and increasing the richness of synthesized scenes. Complexity management issues have arisen in the development of efficient mesh decomposition methods for spherical surfaces, and in the progressive refinement approach to discrete feature placement. Scenic richness has been promoted through the modulation of fractal generator functions to produce planned topographical heterogeneity, together with a controlled method for the inclusion of feature populations within the model.

A model of the lunar surface was created to evaluate the techniques described. Viewing distances may be continuously varied through a wide range of altitudes without level of detail transitions in the generation process being observed. The images which are produced exhibit an overall visual appearance which compare well with actual lunar photographs. The techniques described may be applied to the synthesis of other solid planetary bodies, including the earth, given appropriate extensions to model local geological forms.

ACKNOWLEDGEMENTS

We would like to thank the University of Manchester Advanced Interfaces Group members for their support, the anonymous reviewers for their constructive comments, and Ken Musgrave for his encouragement and inspiration.

REFERENCES

[Fourn82] Fournier, A., Fussel, D., Carpenter, L., Computer Rendering of Stochastic Models, *Communications of the ACM*, Vol.25, No.6, pp.371–384, 1982.

[Heckb86] Heckbert, P.S., Survey of Texture Mapping, *Computer Graphics and Applications*, Vol.6, No.11, pp.56–67, 1986.

[Lewis87] Lewis, J.P., Generalized Stochastic Subdivision, *ACM Transactions on Graphics*, Vol.6, No.3, pp.167–190, 1987.

[Linds96] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Faust, N., Real-Time, Continuous Level of Detail Rendering of Height Fields, *ACM Computer Graphics (SIGGRAPH '96 Proceedings)*, pp.109–118, 1996.

[Mande82] Mandelbrot, B.B., *The Fractal Geometry of Nature*, W.H. Freeman and Co., New York, 1982.

[Musgr93] Musgrave, F.K., Methods for Realistic Landscape Imaging, Yale University Department of Computer Science, TR-1038, May 1993.

[Musgr94] Musgrave, F.K., Procedural Fractal Terrains, in *Texturing and Modeling: A Procedural Approach*, Ebert, D.S. (ed.), Academic Press, pp.295–310, 1994.

[ORour97] O'Rourke, J., comp.graphics. algorithms Frequently Asked Questions, available via the Internet at http://www.exaflop.org/docs/cgafaq

[Peach94] Peachey, D., Building Procedural Textures, in *Texturing and Modeling: A Procedural Approach*, Ebert, D.S. (ed.), Academic Press, pp.5–96, 1994.

[Perli85] Perlin, K., An Image Synthesizer, *ACM Computer Graphics (SIGGRAPH '85 Proceedings)*, Vol.19, No.3, pp.287–296, 1985.

[Perli94] Perlin, K., Noise, Hypertexture, Antialiasing and Gesture, in *Texturing and Modeling: A Procedural Approach*, Ebert, D.S. (ed.), Academic Press, pp.193–248, 1994.

[Rükl90] Rükl, A., *Atlas of the Moon*, Kalmbach Publishing Co., 1990.

[Saupe89] Saupe, D., Point Evaluation of Multi-Variable Random Fractals, in *Visualisierung in Mathematik und Naturwissenschaften*, Jurgens, H. and Saupe, D. (eds.), Springer-Verlag, pp.114–126, 1989.

[Ulich88] Ulichney, R., *Digital Halftoning*, MIT Press, 1988.

[Voss85] Voss, R.F., Random Fractal Forgeries, in *Fundamental Algorithms for Computer Graphics*, Earnshaw, R.A. (ed.), Springer-Verlag, pp.805–835, 1985.

[Voss88] Voss, R.F., Fractals in Nature: From Characterization to Simulation, in *The Science of Fractal Images*, Peitgen, H.O. and Saupe, D. (eds.), Springer-Verlag, p.32, 1988.