INTERACTIVE REMESHING FOR NAVIGATION OVER LANDSCAPE MODELS

M. V. CIFUENTES †, M. J. VENERE‡ and A. CLAUSSE‡

† *CICPBA - Universidad Nacional del Centro, 7000 Tandil, Argentina*
‡ *CNEA – Universidad Nacional del Centro, 7000 Tandil, Argentina*
{cifuente, venerem, clausse}@exa.unicen.edu.ar

***Abstract*— The performance of an algorithmic procedure to visualize digital landscapes based in the dynamic control of the size of the model is presented. Approximate representations are produced taking into account criteria relative to the observer position and the local curvature of the terrain. The algorithm leads to substantial reductions of the size of the final meshes, lowering the rendering costs down to 5%. An application of the algorithm is tested on a model of the Colorado Canyon is shown.**

***Keywords*— surface simplification, landscape visualization, computational geometry, polygonal mesh.**

I. INTRODUCTION

Virtual reality applications based on outdoor scenarios demand realistic and efficient topographic representations. The current trend is to abandon models based on special effects such as projected images or synthetic representations, resorting instead to real digital elevation models (DEMs). Basically a DEM is a grid where each cell is associated with a pair of geo-referenced coordinates (e.g. geodesic or Gauss Kruger) and a corresponding terrain height, leading to a digital representation of a fraction of the earth surface within a given resolution. Current DEMs use tens and even hundreds of millions of cells. Such a huge volume of data is necessary in many applications, but it is hard to visualize interactively, particularly if interactive navigation is involved (Duchaineau *et al.*, 1997; Gobbetti *et al.*, 2006 and Livny *et al.*, 2007). The mesh data often exceed the available memory, and this problem is more dramatic if texture maps are involved (Döllner *et al.*, 2000).

In the last decades several efficient algorithms and techniques were developed to digitally represent terrains preserving the quality of the visualizations while keeping fast access to the data. One of the most interesting strategies is mesh simplification by recursive decomposition. In particular, the adaptive simplification algorithms start from the simplest version of the original model and subsequently add greater detail enhancing the quality of the primal model (Balmelli *et al.*, 1998; Hoppe, 1998; Pajarola, 1998; Röttger *et al.*, 1998; Lindstrom and Pascucci, 2002; Pajarola *et al.*, 2002 and Pajarola and Gobbetti, 2007). DeHaemer and Zyda (1991) have successfully applied this technique in DEMs with an algorithm that preserves the topology.

In this article, a variant of the adaptive simplification methods is presented and applied to DEMs visualization. A hierarchy quadtree with restrictions and templates associated to the end nodes is used to generate the final surface ensuring conformal triangulation and avoiding major changes in the size of the mesh polygons. The basic guiding criterion to the coarsening process is the local curvature of the represented surface (Cifuentes *et al.*, 2004). Subsequent enhancement is achieved by introducing a refinement according to the position of the observer (Gerstner, 1999). The result is a dynamic mesh that is extracted from the quadtree hierarchy which mutates following the movement of the observer. The size of the problem can be reduced one order of magnitude; and even more if some quality loss is accepted.

II. COARSENING GUIDED BY CURVATURE

Let us start with a DEM supported by a regular mesh of squared pixels. The height of the terrain, h_{ij}, is provided for every $\Delta\times\Delta$ cell located in each vertex (i,j). One reasonable criterion to simplify this model is to decrease the number of pixels in regions where the curvature is small (Miao *et al.*, 2009). The underlying principle is that in order to obtain similar visualization qualities smoother regions can be represented by fewer cells than rougher regions.

To construct a simple local curvature indicator, we propose to use the Frobenius norm of the curvature tensor in each vertex, given by:

$$\kappa_{ij}^2 = \left(Gx_{i+1,j} - Gx_{i-1,j}\right)^2 + \left(Gy_{i+1,j} - Gy_{i-1,j}\right)^2 + \left(Gx_{i,j+1} - Gx_{i,j-1}\right)^2 + \left(Gy_{i,j+1} - Gy_{i,j-1}\right)^2, \quad (1)$$

where Gx_{ij} and Gy_{ij} are the gradient components in each direction and can be efficiently approximated by

$$Gx_{ij} = \frac{1}{2\Delta}\left(h_{i+1,j} - h_{i-1,j}\right), \quad (2)$$

$$Gy_{ij} = \frac{1}{2\Delta}\left(h_{i,j+1} - h_{i,j-1}\right), \quad (3)$$

Accordingly, the cumulative curvature of a given set S of neighbour vertices is defined as

$$K(S) = \sum_{(i,j)\in S} \kappa_{ij}, \quad (4)$$

Then, the procedure to simplify the mesh is as follows: