Real-Time Simulation of Autonomous Vehicles on Planet-Sized Continuous LOD Terrains

Selcuk SUMENGEN
Computer Graphics Laboratory
Sabanci University
selcuk@su.sabanciuniv.edu

Selim BALCISOY
Computer Graphics Laboratory
Sabanci University
balcisoy@sabanciuniv.edu

ABSTRACT
Real-Time visualization of interactive simulation environments using large datasets of height fields became feasible using current off the shelf graphics hardware. Our approach provides continuous level of detail rendering of high detailed, planet sized terrains using restricted quad-trees without re-sampling data points. The presented method preserves the original planet coordinate frame of the data gathered from the Mars Orbiter Laser Altimeter with 128 samples/degree resolution for vehicle simulation purposes. Furthermore the algorithm avoids discontinuities at the block boundaries occurring at latitudes.

Keywords
Terrain visualization, quad-tree, planet size simulation.

1. INTRODUCTION
Visualization of large scale planet-sized terrain datasets with interactive frame rate is a complex problem. The system should coupe with Gigatriangle sets. Moreover rendering of terrain is not sufficient in most of the application areas. The system should take care of other tasks such as simulation of the environment, vehicle navigation, collision detection and response. This paper presents a complete system which focuses on interactive terrain visualization and simulation of vehicles on it.

The presented algorithm preserves the co-ordinate system of the elevation data. Thus there is no necessity for further pre-processing. Moreover we preserve geometric continuity of the planet surface and therefore prevent cracks on the surface at every region of the planet.

2. TERRAIN VISUALIZATION
Terrain engine renders elevation data with several detail levels determined by surface slope and distance from the camera. The presented algorithm uses quadtree data structure to store and access data points as well as performing frustum culling and triangle tessellation.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

Restricted Quad-Tree (RQT) [Paj98] algorithm limits the level of detail difference between the neighbourhood nodes to one level. This approach reduces polygon count while preserving terrain detail without degrading the overall performance.

Previous approach on visualization of planet uses PBDAM [Cig03] (Planet-Sized Batched Dynamic Adaptive Meshes) structures to maintain geometric continuity, which presents mapping of a planet to a cube having six faces as a square patches. This approach requires pre-processing for re-sampling the elevation data which is an additional cost and a handicap for dynamic update of height fields.

Generating Restricted Quad-Tree
Arrangement of the elevation data is a rectangle form. However, planets have 180 latitudes and 360 longitudes after cylindrical projection, and therefore their elevation data sets can not be used with quad tree structures directly.

We position the rectangular sample frame into a square by overlapping centres. Restricted Quad Tree will be formed using square mapping. Some of the points in the square will remain unmapped after placing the rectangle frame into a square frame. Marked regions in the Figure 1 are formed by these unmapped points but they will also remain unreferenced after branching the tree because available coordinates for viewing the terrain is inside the latitude 0 – 360 and longitude 0 – 180 and these nodes of the tree will never collapse with a little modification on branching algorithm. This method will degrade the memory usage and prevent unnecessary calculations.

Restricted Quad Tree is always successful for preserving geometric continuity inside the square; however there might be discontinuities on the boundaries such as transition from longitude 0 directly to longitude 359 will cause cracks caused by level of detail changes. This is avoided with the limitation of the maximum viewing area to 180 longitudes.

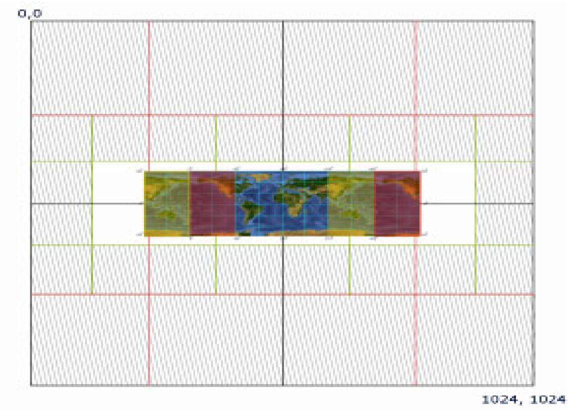

Figure 1: Square Region Mapping: Mapping of elevation data to a square after cloning rightmost and leftmost regions to preserve geometric continuity at boundaries.

Branching of Restricted Quad-Tree

Elevation data samples with resolution of N can be placed in a square sized (1024 x N + 1) x (1024 x N + 1), so every 1º x 1º region corresponds to nodes at level 10. The corresponding quad tree structure will be branched to level 10 and level 10 is preserved as base level. This produces enough number of triangles while viewing half of the planet as a whole. Expansion of nodes that are below level 10 should be determined after checking viewing volume, viewers distance from the planet, and slope of the corresponding planet surface.

While expanding the nodes to several levels (Figure 2 a-b-c), LOD difference between the neighbouring regions should be concerned to avoid discontinuities and cracks. If there is a LOD difference between the neighbours the difference must not be more than one (Figure 2 c-d). This constraint can be satisfied easily using any recursive branching algorithm. Another requirement is marking the triangle on the neighbouring side of lower level region for split operation from its bisector if there is a level of detail difference. Reason of this split operation can be seen from figure 2 where the regions with higher level of detail have extra vertices on shared sides with neighbours having lower level of detail. If these vertices on the same side are on different planes, there will be cracks on the surface. These cracks are prevented by splitting the corresponding triangle of lower level region from its bisector. After this operation smooth transitions of different level of details achieved (Figure 2f).

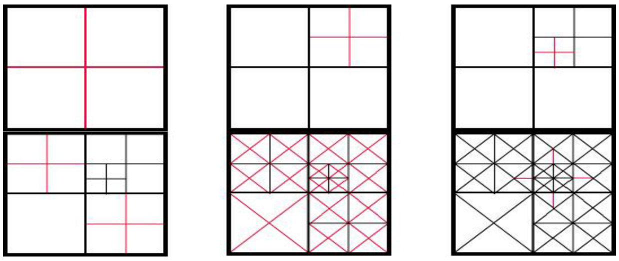

Figure 2: Branching Restricted Quad-Tree: from left to right: a-) 1. LOD b-) 2.LOD, c-) 3.LOD, d-) Level Refinement, e-) Triangulatization, f-) Connecting levels

3. Results

We used NASA's 128 sample/degree data set for Mars surface (1.93GB) and 4 sample/degree data set for Moon surface. The Figure 3 is a snapshot from real-time rendering of Mars surface with 400 ground and air vehicles with 60 frames/seconds.

Figure 3: Mars Simulation Environment

4. REFERENCES

[Cig03] Cignoni, P., F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio and R. Scopigno, "Planet--sized batched dynamic adaptive meshes (P-BDAM).", In Proceedings IEEE Visualization, Conference, Seattle USA, October 2003.

[Paj98] Pajarola, R., "Large scale terrain visualization using the restricted quadtree triangulation.", IEEE Visualization'98, North Carolina, 1998