Hexagonal Geometry Clipmaps for Spherical Terrain Rendering

Shiben Bhattacharjee* CVIT, IIIT Hyderabad

P.J. Narayanan† CVIT, IIIT Hyderabad

Abstract

Terrains can be rendered efficiently with rectangular 2D grid of heights. Terrain on spheres, on the other hand, can be rendered using Hierarchical Triangular Mesh (HTM) but the representation does not fit directly with 2D grid of heights. We present a unified representation of HTM and clipmapping using Hexagonal Geometry Clipmaps. This provides one to one correspondence of vertices and heights, low and constant memory usage, less storage space requirements, no pole singularity, fast vertex look-ups, and large range of view distances. Hexagonal clipmaps fit equilateral triangles of HTM and achieve uniform triangle count on the screen. We keep the clipmaps on the GPU for fast access from shaders and use vertex buffer objects for fast triangle rendering.

Keywords: Spherical Terrain Rendering, Hierarchical Triangular Mesh, Hexagonal Geometry Clipmaps

1 Introduction

Terrains are of great interest in flight simulators, geographic information systems, games, etc. Earlier terrains have been rendered with triangulated irregular network but recently have been given away to regular grid representation especially on fast graphics hardware. Losasso and Hoppe [2004] introduced a multiresolution, fixed memory size scheme for efficient representation and rendering of large terrains, called the Geometry Clipmaps. A 2D grid seems to correspond directly with a latitude and longitude based representation of spherical terrains, but this does not distribute the samples regularly at all latitudes and has singularity at the poles. Hierarchical Triangular Mesh (HTM) [2005] can subdivide a sphere into similar spherical triangles providing near uniform sampling of the whole surface of the sphere. A combination of clipmaps and HTM can provide a uniform, singularity free representation and fast rendering. We introduce Hexagonal Geometry Clipmaps which pack equilateral triangles perfectly while retaining clipmap's low and constant memory usage and large view ranges. The HTM underneath ensures uniform samples over the surface of the sphere. This reduces the storage space requirements at the high latitudes compared to rectangular grids (Fig 1(a)).

2 Design and Implementation

An octahedron contains eight equilateral triangles. These *base triangles* are subdivided recursively [2005] until the detail of sphere required is reached (Fig 1(b)). Any two adjacent base triangles collectively contain a 30° sheared 2D grid of samples. Terrain data contained in these base triangles can be kept as a usual heightmap. We create pairs of adjacent base triangles picking one from northern hemisphere and another from southern hemisphere. Four pairs form four 2D images and can represent the whole sphere. To match the terrain data available, we transform the data so that the equator aligns with the diagonals of the four images and the poles as the two complementary corners (Fig 1(c)). We handle each of these four terrains separately. For these 2D terrains, we form square geometry clipmaps centered around the camera position with high resolution closer to center (viewer) and low resolutions in outer rings. These square clipmaps are, however, rhombi in the world coordinates. We discard the two sharp corners of each rhombus to form a hexagonal geometric mesh (Fig 1(d)) thus forming hexagonal geometry

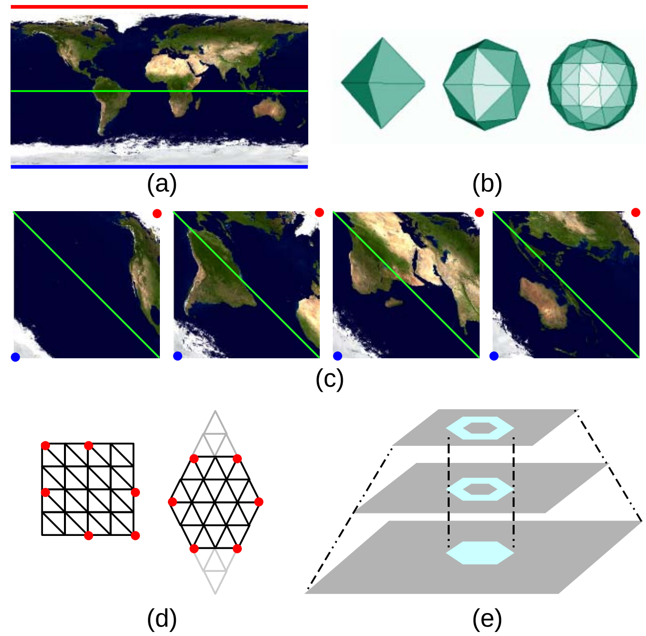

Figure 1: Representation scheme for Hexagonal Clipmaps

clipmaps (Fig 1(e)). (Note that we utilize $9/16$ of the data available in the low resolution clipmap textures). The clipmaps are updated every frame torroidally [2004] according to viewpoint shifts. A hexagonal clipmap can be broken into six trapezia which help in view frustum culling. Generic triangle lists of the shape of trapezium are used for the tessellation of all the trapezia in the view. We use spherical interpolation to calculate the vertices which perform better than polar coordinate calculations involving sine and cosine operations. Hexagonal clipmaps give optimal rendering throughput and steady rendering same as square clipmaps do. Distances from the edges of the hexagonal clipmaps to the viewpoint are less varying than in case of square clipmaps. This provides better uniformity in triangle count on the screen at any camera yaw angle. Square clipmaps (255×255) for flat terrains (Puget Sound dataset) achieve 630 average fps on a GeForce 8800GT GPU. We created a sphere with 4 Puget Sounds (one for each pair of base triangles, flip-flopped for continuity) and achieved 520 average fps with clipmap size of 257×257 on the same GPU.

3 Conclusion

We presented a Spherical Terrain Rendering algorithm which provides uniform sampling of points over the surface and fast rendering with low memory usage. Applications like Google Earth/Virtual Earth, space simulators, 3D social networks (e.g. Second Life) or spacecraft involving games can show seemless journey from ground to space and back using this method.

References

Losasso, F., and Hoppe, H. 2004. Geometry clipmaps: terrain rendering using nested regular grids. In *ACM Transactions on Graphics, Vol 23, Issue 3*.

Szalay, A., Gray, J., Fekete, G., Kunszt, P., Kukol, P., and Thakar, A. 2005. Indexing the sphere with the hierarchical triangular mesh. In *MSR-TR-2005-123*.

*e-mail:shiben@research.iiit.ac.in
†e-mail:pjn@iiit.ac.in