CityScape, a Walkthrough System for Large Environments Using Stratified Rendering

Rebecca Xiong*
becca@graphics.lcs.mit.edu
MIT Laboratory for Computer Science

Real-time visual simulation of large environments such as cities can be useful for city planning, embedding of commercial databases, military training, and entertainment. Polygon-based representations of these environments allow for collision detection, re-illumination, and object manipulation. However, current hardware cannot render such large polygon models using brute force at interactive frame rates.

Visibility-based algorithms can reduce the number of rendered polygons by exploiting occlusions such as walls of a room [4]. For large, irregular outdoor environments where most objects are visible, a different approach is needed. Approximation algorithms render simplified level-of-details for the individual objects or clusters of objects [3, 2]. However, they require cost-benefit heuristics to select the best representation, and make no guarantee of the visual quality [1, 2].

Stratified Rendering

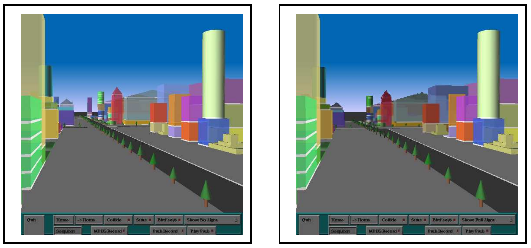
Improved hardware support for texture mapping has enabled a new approach – Stratified Rendering – applied in the CityScape Walkthrough System. The novel aspects of this approach are 1) it maintains, in physical memory, a constant amount of state for scenes of bounded density; 2) it dramatically reduces the number of polygons to be rendered; 3) it requires no cost-benefit calculation; 4) it exploits object spatial and temporal coherence; and 5) it makes an absolute guarantee of image quality (for a fixed display resolution).

Instead of using approximations for objects, clusters of objects, or cells of objects, approximations for large portions of the model are used. Stratified Rendering derives from a simple observation: as the viewpoint changes, *remote objects appear to change relative position more slowly than nearby objects.* In other words, for viewpoints that are close to each other, the rendered appearance of remote objects changes slowly. This temporal coherence can be exploited by pre-computing this appearance, and re-using it for many frames during subsequent walkthroughs.

The system consists of two stages. During pre-processing, the space of possible viewpoints are divided into **viewcells**. Then for each viewcell, using motion parallax, polygons in the scene are partitioned into **near** and **far** sets. The far polygons are projected onto the sides of a bounding box and stored as texture maps. During interactive simulation, as the synthetic viewpoint moves, its corresponding viewcell can be determined. Based on the viewcell, **near polygons** are retrieved, culled, and rendered as before. Instead of the original far polygons, only a constant number of

*Rebecca Xiong is advised by Prof. Seth Teller and is supported by an NSF Fellowship. Address: Room 226, 545 Technology Square, Cambridge MA 02139. Web: http://graphics.lcs.mit.edu/~becca

far texture maps are retrieved and rendered. Thus, the rendering load in each frame is significantly reduced. The new approach also allows the user to set an absolute bound on the angular error between the projection of an actual object and the projection of texture maps containing that object. Thus, the rendering accuracy during walkthroughs can be guaranteed.



(a) Brute force (b) Stratified

Figure 2: Image quality.

Results

Several tests were performed on a Silicon Graphics Onyx with four processors and one RealityEngine 2 Graphics Pipeline. The input consisted of a randomly generated cluster of city blocks with 83,000 polygons. Figure 1 shows that brute force rendering gives an average frame reate of 0.05 seconds/frame, and ideal Stratified Rendering 0.016 seconds/frame. This corresponded to frame rates of 20Hz and 60Hz, respectively. Stratified Rendering achieved an improvement factor of three in this case.

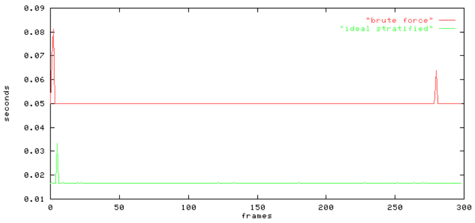


Figure 1: Performance data.

Figure 2 compares an image from brute force rendering with an image from Stratified Rendering with an angular error bound of 7.5°.

The difference between Figure 2(a) and 2(b) is strictly within the angular error bound specified. The near objects that the users tend to focus on are rendered the same, while the far objects rendered as far texture maps in the background of Figure 2(b) are slightly different from those in Figure 2(a).

In summary, by using the Stratified Rendering Algorithm, CityScape achieves both speed and realism for virtual walkthroughs of large, irregular environments.

References

[1] Thomas A. Funkhouser and Carlo H. Sequin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environment. In *Computer Graphics Proceedings*, volume 26. SIGGRAPH, August 1993.

[2] Paulo W.C. Maciel and Peter Shirley. Visual navigation of large environments using textured clusters. In *Computer Graphics Proceedings*. SIGGRAPH, August 1995.

[3] Jarek R. Rossignac and Paul Borrel. Multiresolution 3d approximations for rendering complex scenes. Technical report, IBM T.J. Watson Research Center, 1992.

[4] Seth J. Teller. Visibility preprocessing for interactive walkthroughs. In *Computer Graphics Proceedings*, volume 25. SIGGRAPH, 1991.