



Fast Line-of-Sight Computations in Complex Environments using GPUs

Department of Computer Science

University of North Carolina at Chapel Hill

November 2005

The Challenge

We present a method to accelerate line-of-sight computation for computer generated forces (CGF) with a high number of entities. Our algorithm uses graphics processing units (GPUs) to accelerate the line of sight (LOS) queries. We use the GPU to limit the number of queries for which intersection tests must be performed. We present a hybrid algorithm that uses the GPU in a preprocess to determine regions between which there is no LOS. Our approach is well suited to LOS computations in complex urban environments.

Our previous GPU/CPU hybrid LOS system used the GPU to remove queries with definite line of sight. A conservative rasterization of the terrain and line segments representing LOS queries determined which queries passed above the terrain. Such queries had definite LOS and do not need further computation. This system reduced the workload of CPU which resolved the LOS of the non-culled queries using ray-casting. Our new region-based method layers on top of this system to further reduce both GPU and CPU workload by culling away queries between regions where there is no possible LOS.

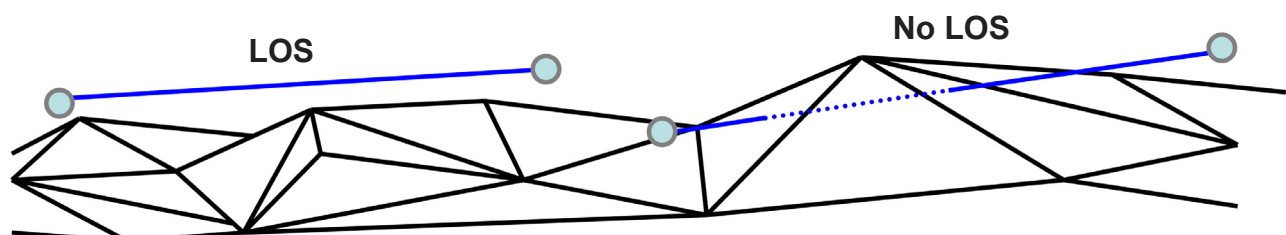
Military simulations can contain thousands of entities that must be checked for line of sight. If each unit must be checked against all other units this can amount to millions of LOS queries that must be performed. We use a new region-based method to reduce the number of LOS queries to be resolved.

Highlights

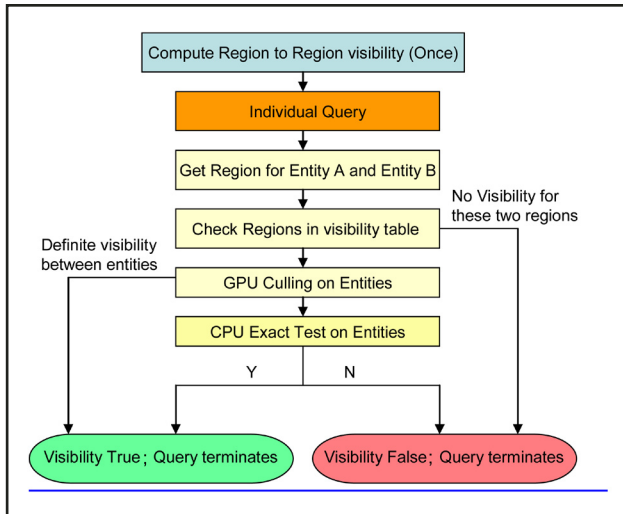
- Line of sight (LOS) is a visibility query between two entities with respect to a terrain and other entities
- Our previous system was able to conservatively cull queries with definite LOS
- We use a new region-based preprocess to create a visibility table that specifies pairs of cells completely occluded from each other
- In urban environments, this approach is found to further reduce millions of LOS queries and accelerate the LOS computation by an additional factor of 4 to 10, for a large-scale military simulation containing thousands of entities
- Demonstrated 100-200x speed up of LOS calls in OneSAF on a single CPU/GPU machine
- Demonstrated 10x overall speedup improvement in OneSAF system performance
- GPU LOS Computations in Complex Environments transitioning into Block D Build 24 of OneSAF

The Approach

Our LOS solution subdivides the environment into cells and determines pairs of cells that are fully blocked from each other. For entities in such pairs of cells there is definitely no LOS. During the simulation runtime there is no need to perform any further CPU or GPU computations for these queries. We store this cell-to-cell visibility information in a visibility table and each LOS query is first tested against this table. In highly occluded urban environments many queries can be culled away using this technique.



A line-of-sight (LOS) query is a point-to-point visibility query with respect to a terrain or an urban environment. Given a large environment with N moving entities, we used GPU-based acceleration techniques to overcome the $O(N^2)$ bottleneck.



Our algorithm culls queries twice. First a visibility table is used to remove queries with out LOS. Next the GPU culling removes queries with LOS. Finally the CPU based ray casting technique tests the un-culled queries.

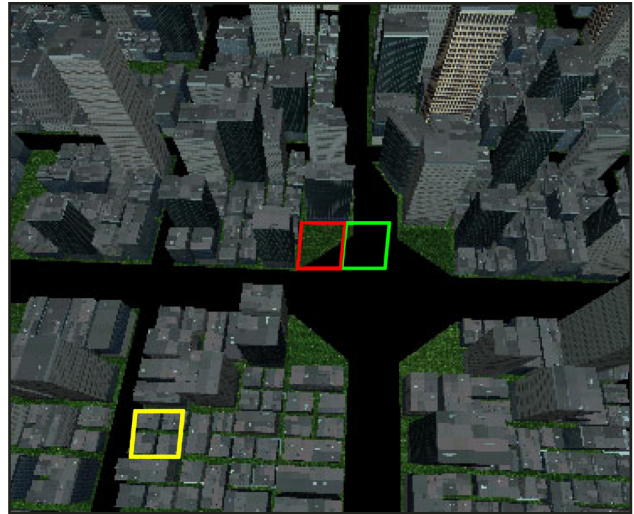
A preprocess performs the spatial subdivision of the environment and the cell-to-cell visibility computations. Determining the visibility between two cells exactly is computationally intensive. Instead, our algorithm uses a conservative approximation that detects most cell pairs for which all LOS queries are blocked. Queries not resolved by the visibility table are tested using our previous GPU/CPU technique [Salomon et al. 2004] to ensure that the correct LOS result is always returned.

The GPU is used to compute the visibility between cells. We process each cell individually and compute the conservative set of cells blocked from the current cell. Each pixel in the frame buffer can represent a subset of rays emanating from the cell being tested. We rasterize each environment polygon into the buffer, so that it covers only those pixels representing rays that intersect it. The framebuffer stores blockage for every ray emanating from the current cell. By rasterizing the other cells into this same framebuffer we can determine which cells are fully blocked by obstacles in the environment.

Rather than rasterizing every polygon, we traverse a spatial hierarchy containing the environment polygons. We traverse this subdivision in a hierarchical front to back order from the current cell. By detecting that nodes of a hierarchy are occluded from the current cell we can avoid testing any of the contained polygons.

Results

We have tested our region-to-region LOS acceleration algorithm on a highly occluded urban environment with several thousands of moving entities. In this environment, we are able to cull approximately 80% of the LOS queries using the visibility table. The average query time was 0.6 microseconds using the region-based culling vs. 3 microseconds without.



If a unit is in the red cell, and another is in the green cell, then the LOS Query will still need to be performed. From the red cell to the yellow cell the query does not need to be performed because a table lookup shows that there is no possible LOS from the red cell to the yellow cell.

Project Leaders

Dinesh Manocha, professor, UNC

Ming Lin, professor, UNC

Government Project Leader

Maria Bauer, RDECOM

Project Members

Sean Hanlon, graduate research assistant

Brian Salomon, graduate research assistant

Dave Tuft, research staff

Other Investigators

Angel Rodriguez, RDECOM

Jason Munro, Eric Root, Marlo Verdesca, SAIC

Research Sponsors

Defense Advanced Research Projects Agency
Battle Command Simulation and Experimentation
Office

RDECOM

U.S. Army Research Office

Selected Publications

Govindaraju, N., M. Lin, and D. Manocha. "Fast and Reliable Collision Culling using Graphics Processors," *Proc. of ACM VRST*, 2004.

B. Salomon, N. Govindaraju, A. Sud, R. Gayle, M. Lin, D. Manocha, B. Butler, M. Bauer, A. Rodriguez, L. Eifert, A. Rubel and M. Macedonia, "Accelerating Line of Sight Computation using Graphics Processing Units", *Proc. of Army Science Conference*, 2004.