Scalable Motion Simulation
65

Stephen Chenney
Okan Arikan
D. A. Forsyth
University of California, Berkeley
schenney@cs.berkeley.edu

Simulation is an important method for animating virtual environments, but its computational cost limits the size and complexity of highly dynamic worlds. We are developing new algorithms for simulation culling that are applicable to objects that move over large distances and exhibit group behaviors, such as the approach described here for cars in a city. Through culling, we can scale simulation to very large environments while paying little for motion out of view.

In previous work,[1] we describe simulation culling for independent objects with small spatial ranges, which precludes cars in a city. For simulation level-of-detail, other authors[2,3,4] have used cheap proxy simulations that still work for every object on every frame and compute full group behaviors on the approximated motion. For dependent, wide-ranging objects, we improve upon previous results in two ways:

- Only objects in view are fully simulated on every frame, while objects out of view are updated at a lower frequency with much lower cost.
- Probabilistic models are used to capture the overall effect of group behaviors on objects out of view without explicitly tracking interactions.

We experient with automatically generated cities populated by cars that roam the streets making random turns at each intersection. The cars obey the rules of stop-sign intersections and avoid collisions by adjusting their speed. On each frame, the potentially visible roads are made known to the car simulation, which then maintains a list of potentially visible cars. Every other car makes a guarantee, marking roads that it may be on up to some future expiration time. The guarantees are computed by doing breadth-first search on the road network to locate roads the car may be able to reach in the future. Bounds on maximum velocities allow us to say how long it will take to reach any given road, and hence compute expiration times.

On each frame we perform the following operations:

- Make new guarantees for all cars with expired guarantees.
- Determine the location of each car whose guarantee includes potentially visible roads, and, if visible, add the car to the potentially visible set or make a new guarantee.
- Simulate the potentially visible cars as normal.
- Remove any cars that are no longer visible from the potentially visible set and make guarantees for them.

To position a car when its guarantee overlaps the view, we associate a probability distribution with the travel time for each roa, and do random walk with sampled travel times. The distributions are based on statistics gathered from an off-line simulation of all the cars, so they directly encode the cumulative effects of group behaviors. Also associated with each road are various approximation functions that allow us to set the state of the car after it has been out of view.

Without culling, we can simulate at most a few hundred cars on a PC. With culling, we can scale up to thousands of cars. The performance of a culling algorithm is best defined by its efficiency, which is the ratio of work done for systems in view to total work done. We see efficiencies ranging from 86% to 96% in experiments performed with a constant-sized city and a variable number of cars. In experiments using larger cities with up to 2,400 cars, efficiencies fall to 35% because we track all the objects that may ever be seen in the world. In future work, we expect to address this by tracking only those objects the viewer has seen recently and adding and deleting other objects as needed.

References

1. Chenney, S. Ichnowski, J. & Forsyth, D. (1999). Dynamics modeling and culling. IEEE Computer Graphics and Applications, 19 (2), 79-87.
2. Grzeszczuk, R., Terzopoulosm, D., & Hinton, G. (1998). NeuroAnimator: Fast neural network emulation and control of physics-based models. Computer Graphics, Proc. of SIGGRAPH 98, 9-20.
3. Carlson, D.A. & Hodgins, J.K. (1997). Simulation levels of detail for real-time animation. Graphics Interface '97, 1-8.
4. Yu, Q. & Terzopoulos, D. (1999). Synthetic motion capture: Implementing an interactive virtual marine environment. The Visual Computer, 377-394.

Caption:

Screen snapshots from a city simulation. The top image shows all of the city, populated by 100 cars with culling turned off. Red roads are potentially visible, and the shaded cone indicates the viewer's location. The middle image is a close up while culling is in progress. Only cars on or near the visible roads are being simulated. The green shades indicate how many non-visible cars may be on the other roads. The bottom image is the walkthrough view corresponding to the middle map. Note the car stopped at the intersection waiting for another to pass.