# No More Bumping Around

## NEW WORK ON COLLISION AVOIDANCE AND NAVIGATION RESULTS IN MORE REALISTIC MULTI-AGENT SIMULATIONS FOR GAMES

BY GARRET ROMAINE

THE VISUAL BEAUTY OF TODAY'S GAMES IS UNDENIABLE, AND POWERFUL CPUs AND GPUs ARE A BIG PART OF THAT SUCCESS. Deep inside the game engine, however, there are new and emerging technologies that can enhance a title's immersive appeal in ways not thought of before. For example, many widely used game engines are better at harnessing the laws of physics to make characters interact in more life-like ways, thanks to an energetic team of university researchers, with some help from Intel.

Most action-oriented games involve the physics of detecting collisions between objects, whether in shooting targets, careening through streets, or just using a paddle in *Pong** to bat a ball around. When clearing a level in a shooting game, such as those built using current game engines, the targets must be realistic, unless the goal is to mow down mindless zombies, such as in the venerable *Resident Evil** franchise. Most games need more than potential targets plodding forward mechanically, jostling each other as they come into range. Those non-player characters (NPCs) might be fellow soldiers and must move smoothly to enhance the game's appeal.

Many long-time gamers can remember certain titles where characters would run around in clipped circles, bounce into unseen walls, or otherwise do dumb, unbelievable things. Those are unacceptable to today's players, and thanks to researchers at the University of North Carolina (UNC) at Chapel Hill, plus some top Intel application engineers, those days are over. The UNC team has developed a set of libraries for C, C++, and C# with tools that are updated and optimized for multi-core and many-core processors, bringing more realism to the gaming experience.

Intel has a long history of funding graphics research at major institutions and pushing the state of the art. Intel's visual computing product development and enabling teams have supported research in visual computing at UNC for more than a decade. This support includes fellowships and stipends for graduate students at UNC working on their PhDs; hosting professors and summer interns from UNC at Intel sites in Santa Clara, California, and Hillsboro, Oregon; equipment support; and employing many MS and PhD graduates.

*The above image demonstrates the application of UNC's crowd simulation library in an urban scene. This scene has more than a hundred agents and many high agents are in close proximity to each library. RVO2 Library can reliably perform collision avoidance and local navigation for each agent in a few microseconds, and thereby enable real-time crowd simulation.*

A recent technical paper, "Reciprocal Collision Avoidance and Navigation for Video Games," co-authored by four members of the Department of Computer Science at UNC and two Intel engineers, describes how games can take advantage of multi-core architectures to incorporate large numbers of virtual agents in game levels with increasing visual fidelity. The rest of this article reviews the work in that paper.

## Make Way for My Agent

The ability to anticipate and measure collisions between two or more objects is crucial to computer animation, modeling, and virtual environments. Game developers and researchers have been working on designing efficient techniques for generating realistic avoidance behaviors between objects and characters, and their environments. Without this work, the entire playing experience suffers.

Traditional methods for generating the behavior or trajectory of characters—or agents—often rely on force-based models, which generate the trajectory of each agent based on a particle system in which each particle applies a force on nearby particles. The laws of physics define the way the particles move, while developers create the rules for specific behaviors such as separation and alignment.

Velocity-based collision avoidance, a concept originally developed in robotics and recently extended to multi-agent and crowd simulation, computes the new velocity for each agent in a distributed manner at each time-step of the simulation. The original concept of velocity obstacle was designed for a single robot navigating among moving obstacles. Its application to multi-agent and crowd simulation can result in not-so-smooth motion for agents in close proximity, and it can cause congestion in resulting crowd simulation.

Enter the innovative reciprocal velocity obstacle (RVO) concept designed by the researchers at UNC. The concept of an RVO involves anticipating other agents' reactions and is performed in a distributed manner. In effect, each agent in an impending collision takes some of the responsibility for avoiding the collision. However, problems can arise if, for example, two agents walking toward each other try to pass on the same side. It is the same problem as when two pedestrians get into a "dance" as they try to keep moving, but each one keeps stepping into the other's way. To keep agents moving freely, researchers have further refined mathematical calculations and developed the hybrid reciprocal velocity obstacle (HRVO) method and the optimal reciprocal collision avoidance (ORCA) method based on optimization techniques. Both these approaches can generate smooth trajectories for each agent, as well as many emergent behaviors observed in real-world crowds. The ORCA algorithm and its implementation, RVO2 Library, are used in the game *Warhammer 40,000: Space Marine**.

The UNC team knew that the existing approaches to collision detection and avoidance for multiple objects could be relatively slow. The team also realized that the utilization of multi-core and many-core architectures in PCs and consoles would allow large numbers of agents to be simulated at higher-quality levels if they could develop enhanced versions of the RVO-based algorithms that could use the multiple cores and data parallelism capabilities of current Intel® processors. So the team set out to determine how to create libraries for different languages so that game developers could automatically generate realistic avoidance behaviors for groups of agents.

## Part of the Engine Now

Publishing a paper and gaining recognition at conferences is important for academics because it helps measure their impact in their research field. Another recognition point is when their work actually shows up in a public product. RVO2 Library, a collection of resources including pre-written code, subroutines, and classes, has already been downloaded by more than 3,000 users and integrated into several games or game engines to either perform local collision avoidance and navigation or improve upon the default implementations in the game engines.

The UNC team has released the following related libraries:

- HRVO Library for C++
- RVO2 Library for C++
- RVO2 Library for C#
- RVO2-3D Library for C++

The acclaim has started to roll in, starting with THQ's inclusion of RVO2 Library for C++ in the game *Warhammer 40,000: Space Marine** from developer Relic Entertainment. On their web site, at http://gamma.cs.unc.edu/RVO2/, the UNC team quotes from a review of the game by the (London) *Daily Telegraph* in September 2011:

*"It's worth noting, however, that no matter how many orks are barreling your way (and often there will be a lot) the frame-rate never takes a hit."*

## The Gaming Triangle

In the 1950s, the area around Raleigh-Durham, North Carolina, became known as the "Research Triangle" because of the many prestigious universities located within a roughly three-sided geography. Thanks to the number of technically skilled graduates emerging from schools such as the University of North Carolina at Chapel Hill, North Carolina State University, and Duke University, the region has attracted numerous large employers, developing a reputation for high-paying, interesting jobs.

Lately, the sheer number of gaming companies springing up or moving to the area has resulted in the area acquiring a new name: the Gaming Triangle. There are over 30 gaming companies, including Epic Games, Electronic Arts, Red Storm Entertainment, and Insomniac Games. In addition, the triangle is home to Joystick Labs, an accelerator that launches startup video-game studios through a mix of early-stage seed funding, mentorship, education, and networking.

At least 1,000 employees work in the area, which boasts the developers of two of the top five game engines. There is the annual East Coast Game Conference in Raleigh for developers and game industry professionals, plus the popular Carolina Games Summit. The State of North Carolina now offers tax incentives for game companies and other businesses developing interactive and digital media, and the Triangle Game Initiative was formed as a trade association for the Raleigh-Durham, North Carolina, interactive entertainment industry.

Deepak Vembar, application engineer for Intel's Visual Computing Group, said working with the UNC team, both with the optimization and in co-authoring the paper, was a privilege. "Intel and UNC have worked together to fund a lot of graphics and simulation research," he said. "They came up with a more elegant solution for simulating crowds or multiple agents, and that's an interesting topic in itself. There's a lot of research going into it."

According to Vembar, it was exciting to work on a project that affected the entire gaming ecosystem. "Their code isn't just pure research. It's part of many games. We're trying to help


## Intel® VTune™ Amplifer XE to the Rescue

The University of North Carolina at Chapel Hill team received plenty of help from Intel's visual computing team to tune and optimize their RVO2 Library. They used Intel® VTune™ Amplifier XE to analyze existing bottlenecks and take advantage of multi-core processors. The result is that RVO2 Library neatly splits the algorithm steps for each virtual agent using either OpenMP* or Intel® Threading Building Blocks.

To test their library, the team relied on help from Intel to create a sample scenario consisting of 1,000 virtual agents, arranged on the perimeter of a circle of known radius with the goal to navigate cleanly to the other side. The benchmark scenario was run on an Intel® Core™ i7 processor Extreme Edition 980X with six cores each with hyper-threading technology (six cores/12 threads). Analysis of concurrency and hotspots by Intel VTune Amplifier XE showed that the application is well threaded with the major significant hotspots in the code having optimal CPU utilization time. The team determined that all 12 threads in the CPU were fully subscribed, with very few inactive periods.

push the boundaries of parallel programming using APIs like Intel® Threading Building Blocks for tasking and to parallelize the workloads. We optimized the code to use Intel Threading Building Blocks so it scales from four cores to six cores and beyond, and it's optimized for scaling across cores. And it's built to take advantage of distributing the load across the CPUs in the system."

The paper covers collision avoidance and agent-velocity computation in detail, with comparison tables and screenshots from the Intel® tools used in the optimization phase. The net result was the optimization of both HRVO Library and RVO2 Library. As the paper states:

*"Both HRVO Library and RVO2 Library choose a new velocity by computing the velocity that is closest to the preferred velocity and is collision-free. If the goal position of the virtual agent is visible, then the preferred velocity is in the direction of the goal. If the goal position is not visible, the preferred velocity should be directed to the nearest node on the waypoint graph to the goal (or to some point on the nearest edge on a navigation mesh path) or roadmap that leads to the goal."*

## Intel on Campus

Intel works with top graphics researchers to help move the state of the art forward, according to Mitchell J. Lum, an Intel product marketing engineer. "Choosing which projects to fund isn't easy, but the team at UNC is very well-published and recognized," he said. "That made it easier to select them. Some research will pay off in the near term, and some will take years to mature."

Just as Intel researchers work on problems that are several years out, faculty members decide where they want to be on the research spectrum. Some work, by its nature, is highly speculative and may never result in marketable products. According to Lum, the UNC team, led by Ming C. Lin and Dinesh Manocha, ". . . has chosen to conduct research that could create breakthroughs relatively soon, in a one-to-three year time frame, as opposed to something so revolutionary that it won't show up in a game for five years."

### HERE IS THE ALGORITHM USED IN RVO2 LIBRARY:

```
input [A] : list of agents, t : time step
loop
     for all A in [A] do
            Get current position and velocity for A
            Compute n nearest neighbors [A'] in [A]
            for A
            for all A' in [A'] do
                  Get current position and velocity
                  for A'
                  Construct velocity obstacle VO(A,A')
                  Construct ORCA(A,A')
            end for
            Construct ORCA(A) from intersection of all
            ORCA(A,A')
            Compute preferred velocity pv for A
            Compute new velocity for A in ORCA(A)
            closest to pv using 2D linear programming
            if 2D linear program is infeasible then
                  Compute new velocity for A closest
                  to pv using 3D linear programming
            end if
     end for
     for all A in [A] do
            Set velocity to new velocity for A
            Increment position for A by t * new
            velocity
     end for
end loop
```

As Lum points out, the UNC team has worked on crowd and multi-agent simulation for quite some time; Dr. Lin's 1993 PhD thesis at the University of California, Berkeley, was entitled "Efficient Collision Detection for Animation and Robotics." UNC researchers understand the challenges involved in designing such algorithms and systems. The UNC team has released a number of libraries for collision detection over the last 15 years. These libraries have been downloaded by more than 100,000 users and licensed by 50 commercial vendors. In the same manner, "Crowd and multi-agent simulation is picking up traction now," Lum said. "This is something that game companies are working on. It's nice to see the practical application of UNC's research." ■