An Experiment in Rule-Based Crowd Behavior for Intelligent Games

Conference Paper · December 2009

DOI: 10.1109/ICCIT.2009.239 · Source: IEEE Xplore

3 authors:

Seongdong Kim
Kaywon University of Art and Design

Christoph Hoffmann
Purdue University

Jae Moon Lee
Hansung University

PAPER

An Experiment in Rule-based Crowd Behavior for Intelligent Games

Seongdong Kim,† *member* and Christoph Hoffmann††, *non- member*

Summary
Today video games and animated movies and robotics involve individual and collective character behaviors. To this end, behavior algorithms are studied to identify underlying principles of how information flows between the characters and studies consider research in animal behavior. The prey-predator models could be based on simple attraction / repulsion. We will primarily be limiting ourselves to the behavior of prey-predator to finding mathematical models and simulating prey-predator interactions for application in computer games.
Key words:
Crowd behavior, rule-based behavior, collective behavior, interaction

1. Introduction

A crowd is a large group of people physically grouped (crowded) together. There are certain behaviors that individuals of a crowd assume because they are members of the crowd that differ from ordinary, individual behavior. The collective crowd behaviors are similar to those found in members of a flock of animals and include collision avoidance and maintaining crowd membership. For instance, a migrating group of geese will have a leader the group is following. A migrating herd of gazelles will proceed in a broad stream of individuals.

In a crowd of people an announcement may cause individuals to head in a particular direction or initiate specific maneuvers. Behavioral character is all about cognitive interaction with environment combined with the limitation imposed by simulation. Crowd behavior is not only needed to create an atmosphere but also should simulate intelligent actions of the group or individuals [1]. Modeling reflexive behavior, and intelligent behavior, is an open-ended task. Even simple flocking and pre-predator models can be difficult to control. Humans are even more complex and modeling their behavior is a task that Artificial Intelligence has been addressing for decades in many contexts [2]. In a [4], Couzin defined each individual animals follow three simple rules: repulsion, orientation and attraction zone [9].

In this paper we explore crowd behavior that is based on simple rules. Despite the simplicity of individual behavior rules, the internal dynamic of a crowd can result in complex collective behavior, and research in ethological biology seeks to explain observed collective behavior by postulating rules for individuals. We seek to invert this process, devising rules that lead to compelling crowd behavior for use in computer games, simulating prey-predator interaction to use the behavior algorithms and enhancing interaction of simulation by manipulating control parameters.

2. Steering and Flocking

Steering behaviors for autonomous characters draw on a long history of related research in other fields. Reynolds [1] used to refer to the improvisational and life-like actions of an autonomous character. He proposed a division of motion behavior for autonomous characters into a hierarchy of 3 layers: action selection, steering, and locomotion. While flocking behavior can be interesting, especially when interacting with obstacles in the environment, the objective is to produce a single uniform motion – the emergent behavior of the flock. It is one of the lowest forms of behavioral modeling. Members have only the most primitive intelligence that tells them how to be a member of a flock. From these local rules of the individual members, a global flocking behavior can emerge. While flocks typically consist of uniformly modeled members, prey-predator behavior may result from mixing two competing types of agents. To use the flock analogy but acknowledge that it refers to a more general concept, Reynolds uses the term boid to refer to a member of the generalized flock. To simulate the behavior of actual flocks, the animator can have the leader change periodically [2]. Actual flocks may change leaders because the wind resistance is strongest for the leader and rotating the job allows the birds to conserve energy as shown in Figure 1.

Manuscript received January xx, 20xx.
Manuscript revised March xx, 20xx.
† The author is with Kaywon School of Art & Design ,Korea, 437-712,
†† The author is with Purdue University, West Lafayette, 47907 USA.

(a)
(b)
(c)
(d)

Fig. 1 (a) Prey-predator scene in Galapagos (1st). (b) Korean crowd behavior at a beach of Haewoondae (2nd). (c) The winter behaviors of birds in Gachang, Korea (3rd). (d) Traffic flow in Seoul(4th) [8]

Assumptions

There are many factors that influence the behavior of the prey-predators and more research is needed to fully and completely model it. Prey and predators have a maximum age. In the case of prey it relates to the number of prey inversely, so that it can model the capacity of the environment to sustain a large herd. That would eliminate the need for the current maximum number of prey allowed. Likewise, fertility should be higher for a smaller number of prey and lower for a larger herd, for the same reasons. Predators should have offspring too, and should feed them to make them grow up. There would be a set amount of food needed over a certain time span to make the predator baby grow up and be viable. Like the prey, predators should die. In nature, herds can assume specific formations to defend themselves and their young. Likewise, many predators hunt in groups and execute a strategy for isolating individuals from a herd and hunting it down. Those behaviors are a particular challenge to model well and require geometric algorithms we are beginning to explore.

3. The Model

Video games, autonomous machines, animations in movies and robotics are subject areas interested in studies that explore individual and group behaviors. We propose as a modified mathematical model of Lotka-Volterra equation for the behavior of prey-predator interaction [7]. We are trying to understand the natural interactions between individuals. If we let N***(t)*** and ***M(t)*** represent the number of prey and predators, respectively, that are alive at time ***t***, then the prey-predator model with linear per capita growth rate is

$$\frac{dN}{dt} = (k - gM)N \quad \text{and} \quad \frac{dM}{dt} = (wN - q)M \qquad (1)$$

The parameter ***k*** is the growth rate of the species ***n*** of prey, in the absence of interactions with the species ***M*** of predators. Prey numbers are diminished by these interactions. The per capita growth rate decreases here linearly with increasing ***M***, possibly becoming negative. The parameter ***g*** measures the impact of predation and ***q*** is the death rate of species ***M*** in the absence of interaction with species N. The term ***wN*** denotes the net rate of growth of predator population in response to the size of the prey population.

4. Description of the Experiment

4.1 Prey-predator behavior description

At a minimum, the reasoning component for prey-predator models could be based on simple attraction/repulsion [**Error! Reference source not found.**]. One character class is attracted to the other trying to eat it while the other class is repulsed by the first and seeks to avoid being eaten. While a simple force field model can produce interesting motion, incorporating some predictive reasoning ability in one or both character classes can make the situational behavior more realistic. The rule used in the experiment is to populate the field with ***N*** preys as follows (see Fig.2). The spawn point is a random location within the test field for mature prey. Prey may spawn babies, and in this case the baby prey is spawned in the vicinity of the parent. The starting and maximum speeds are initialized to be random values with a global default maximum for each of them. As prey ages, they acquire greater speeds, so simulating the vulnerability of new offspring. Next, ***M*** predators are spawned, also in a random location in the test field, and again the current and maximum speed is initialized as in the case of prey. Here the maximum speed is constant and does not increase with time.

4.2 Experimental implementation

We give a few notes explaining the prey-predator algorithm:

- The default behavior of prey is to group together and exhibit flocking but highest precedence is given to obeying boundary constraints of the field. So, even when engaged by a predator, a prey first ensures that it does not violate the test field boundary limits.
- The first update step involves refreshing the current max speed because this grows constantly for a prey. So when it is born the current max speed is relatively low, but as it ages, the speed is increased to reflect growth

*Eqn:curMaxSpeed = (min(curMaxSpeed + 0.15f * elapsedTime, maxSpeed));*

- The next step is to obey boundary constraints. If the prey is currently outside the field limits it steers to seek the center of the field thus automatically causing to turn back into the field.

Code: if(fabs(position().x) > FIELD_SIZE || fabs(position().z) > FIELD_SIZE)

*totalSteer += 10.0f * xxxsteerForSeek(Vec3::zero).perpendicularComponent (forward());*

- Next, a check is performed to find out if the prey is currently under pursuit by a predator. Right now this is pretty straightforward: get the list of neighbors from the spatial database and check if any of them is a predator. If so, find an exact opposite direction vector and steer in that direction. There are also some random calls to a 'wander' steering behavior for better realism.

*Code:Vec3 flee = ((*i)*

->predictFuturePosition(elapsedTime)) - position();

*flee *= -1.0f;*

*flee = flee.normalize() * 13.0f;*

- If not being pursued, then the next step would be to get into a flock with the neighboring prey. This is done in 3 steps:

  a. Separation: Try to maintain a minimum amount of distance from the neighbors so no collision takes place.

     *totalSteer += 8.0f * steerForSeparation(4.0f, -0.707f, nb);*

  b. Cohesion: Try to move towards the center of gravity of the neighboring prey so as to form a closed group.

     *totalSteer += 6.0f * steerForCohesion(9.0f, -0.15f, nb);*

  c. Alignment: Try to orient self in the average forward direction of the neighbors. This ensures the future movement of the collective herd is synchronized.

*totalSteer += 6.0f * steerForAlignment(7.45f, 0.7f, nb);*

- Finally, if the prey is a newborn of an existing live prey (its mother), it tries to stick with it. This is first taken care of by spawning the baby in the vicinity of the mother. The flocking behavior ensures that the baby tries to stay close to its mother. But this could be disrupted by a predator chase or collision avoidance measures amongst the flock. In such a case when a baby's distance from its mother exceeds a particular threshold, it performs a 'steer to seek' action until it is within an acceptable distance from its mother.

- The first step of a predator's update loop is the same as the prey's obeying boundary limits. This is exactly as shown above.

    a. The next step is to find the easiest prey to chase. This involves taking into account 2 things, the distance to the prey and the age of the prey (the younger it is, the slower it is).

5. Results

We have simulated a prey-predator behavior management within a specific area with scripted and ruled-based behaviors.

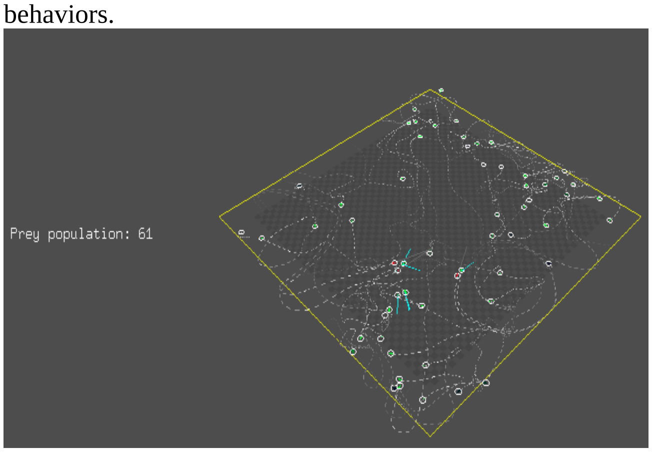


Fig.2 Prey-predator intelligent behavior

To find the optimal prey, a list of neighboring prey is first retrieved. Then for each prey, a heuristic is used to decide its optimal candidate value. The distance to the prey is divided by the Min (0.1f, age), where the age ranges from [0, 1.0]. So far this has turned out to be a pretty good heuristic. We are currently extending the simulation in several ways. Predators are given the ability to communicate with each other up to a certain distance. They are then negotiating which prey to pursue and may collaborate. To collaborate, they first execute a stalking strategy, staying just beyond the distance below which prey notices their presence. Concurrent with stalking, predators need a positioning strategy that surrounds the group of prey and seeks to isolate some of its individuals. If unsuccessful, predators will die from starvation after some time. For enhancing interaction with the simulation, we let the user manipulate the controlling parameters through the control interface shown in Figure 3. The number of prey-predators, of max prey-predators and prey-predator life span can be modified with the control interface as shown in Figure 3. These parameters and modalities are controlled by a user interface, allowing us to explore the effect of various settings on the overall emergent properties of the system.

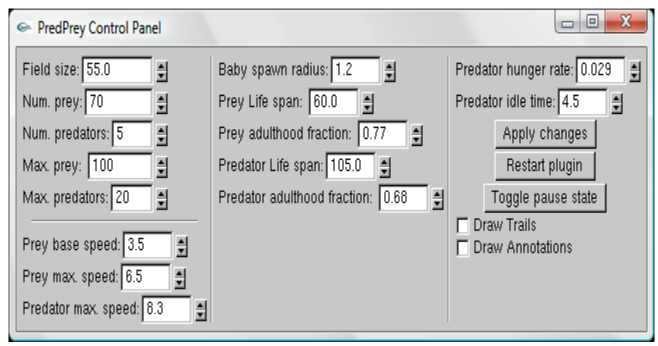
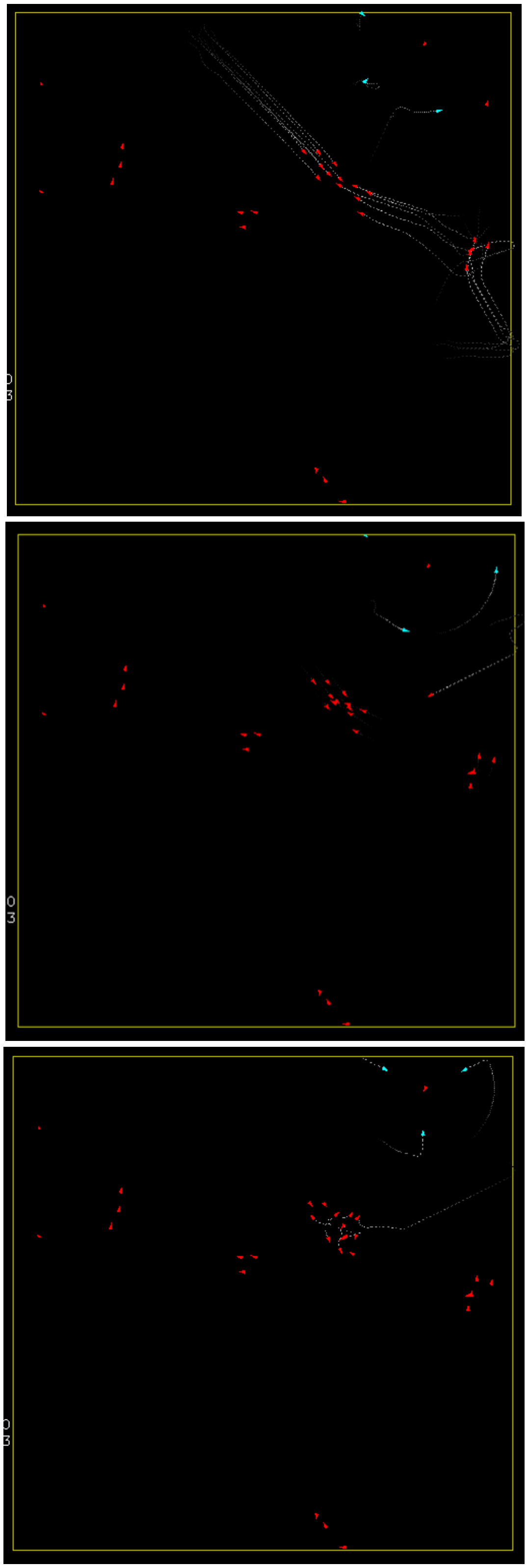


Fig.3 Prey-predator behavior parameters

6. Pack Hunting and other Extensions

The experiment reported uses only very simple behavioral strategies. More behavior traits are desirable and can be implemented. In particular, the Open Steer Library of Reynolds [2] offers tools for overlaying over the basic behavior more complex pattern. One such pattern would be to let predators hunt in packs and try to single out individuals from the crowd, separate them spatially, and then attack. In the current version of the implementation we have begun to implement such strategies. In particular, predators are seeking to communicate with their peers and surround a targeted individual:



(a)
(b)
(c)

Fig.4 Predators (cyan) surrounding 3 prey (red).
(a): two predators decide to join with a third to hunt.
(b): waiting while the predators surround the prey.
(c): closing in on the prey from three sides.

In this implementation, the predators start out as individuals, but a pack-seeking behavior activates other predators are nearby and there is a chance to hunt prey together. Such a behavior has to be weighed against other behavior. For instance, when the pack forms, they do so for the purpose of hunting a common target. There may be an opportunity target that a predator would forgo if the pack formation has too high a priority.

Figure 4 illustrates the pack-hunting process. In the top panel, (a), two predators encounter a third who is getting ready to attack the prey at the top right. They decide to join. Since predator vision extends farther than prey vision, the prey is still unaware of this development. Elsewhere, two flocks of prey decide to join.

Next, in the middle panel, (b), the two predators who have joined surround the prey just outside the circle of awareness, getting into position. The two other herds of prey have now combined into a larger herd.

Finally, in the bottom panel, (c), the predators close in on the prey which unsuccessfully tries to escape. This pack-hunting behavior is in addition to the behavior characterized before. Its activation has to be carefully considered, so that realistic overall behavior results. In particular, the decision to hunt separately or in a pack has to be considered. In our implementation, predators hunt individually unless there are other predators nearby, in which case they join forces and run together. In nature, one can observe more complex patterns in which packs of hunting predators split, separately pursuing different prey, and abandoning the chase of the other prey when a kill has been made.

A similar issue of activating different elements of behavior arises when, in the pursuit of prey 1, a predator passes by a prey 2 that is apparently closer. Here, the time discretization can lead to unexpected results. Suppose we want to implement that a predator switches from prey 1 to prey 2 if, in the pursuit of prey 1, prey 2 is nearer at some point in time. The natural implementation will evaluate the nearby prey 2 by determining the expected distances at time $t+\Delta t$. If the chase is high-speed, the point at which prey 2 is nearer may be in-between time t and time $t+\Delta t$. This makes the predator ignore prey 2 and continue to chase prey 1, a situation that is visually unconvincing.

Finally, if the predator-prey simulation is to reflect animal communities, we have to include spawning offspring that, on the predator side, has to be fed and trained, and on part of the prey has to stay close to a parent. This means in particular that the urge to stay with the crowd is stronger in youth and could diminish later-on, since the relatively slower and vulnerable offspring of prey is an easier target. That is especially the case when grown-up prey are difficult or impossible to attack successfully [11].

7. Temporary Separation

Obstacles present a special problem for autonomous behavior since it is difficult to endow autonomous characters with a true sense of space. For example, using only Reynold's rules, a flock of boids that encounters a set of pillars in its path may have to split into subgroups until the pillars have been passed. A true flock would at that point re-unite, but explicit bounds on the group attraction can cause difficulties. The problem has been investigated in [10] assuming a variety of individuals and accounting for their physical characteristics. The types of individuals considered include bicyclists; one-legged, hopping robots; and abstract point-mass individuals. Since the work accounts for the physical characteristics, it is not surprising that the separation distance for bicyclists has to be made greater than for the other types of individuals, in order to avoid collisions.

Since obstacles must be avoided, as must be other members in the crows, the separation rule has to be adjusted near obstacles. This can be done by defining an influence region around the obstacle of appropriate size, and assigning weights to the urgency with which to avoid the obstacle that grows with diminishing distance from the obstacle. The main difficulty is to ensure a form of progress beyond the obstacle. For example, in order to avoid bouncing aimlessly along an obstructing wall, Reynolds aims for the obstacle silhouette, creating a goal for the motion of the crowd. Once a silhouette location is achieved, the individual must switch to a more distant goal so as to continue along the path and not stop next to the obstacle. This approach of setting goals can be adapted to our predator/prey simulation as follows.

1. Predators seek out prey wherever it may be found. No specific global goal needs to be added.
2. When nearing an obstacle, a subgoal is formulated by which to steer towards the silhouette of the obstacle. When this subgoal has been achieved, we can reinstate the search for the global goal.
3. Prey seeks out pasture for grazing. Again, this is a global goal.
4. Obstacles are treated in the same way as prey do, and when the subgoal has been achieved, the global goal is again reinstated.

We assume here that the formulation of the global goal is derived from the geometry of the obstacle and the location of the global goal, as indicated in Figure 5:

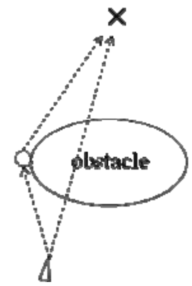


Fig.5: Subgoal formulation by global goal:
The nearer silhouette point is chosen

When the path towards the goal intersects an obstacle, the silhouette points seen from the current position are computed. The nearest silhouette point is then chosen as the subgoal to pursue. When the subgoal has been reached, navigation switches back to the global goal. By choosing the nearest silhouette, complex paths can result as shown in Figure 6.

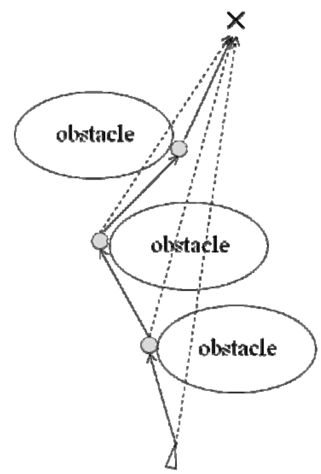


Fig.6: Navigation past several obstacles by subgoal choosing the nearest silhouette point. Solid arrows show the final path.

We have not investigated the competence of this heuristic experimentally. In particular, the case of a moving global goal may require adjustments.

It is important to realize that this heuristic cannot cope with all situations. It uses a simple obstacle avoidance strategy that is very efficient, but it is well known that path planning has high complexity, thus this simple heuristic must fail in certain situations; e.g., [12]. Furthermore, it does not account for the geometry of the moving individual which may not fit through tight passages.

9. Summary

We have investigated experimentally the behavior of individuals in a crowd and the emergence of crowd behavior. Using predator/prey simulations as vehicle, we have considered behavior that adds to Reynolds three basic rules (avoid collision, seek out your kind, and go along with your neighbors) higher-level behavioral traits in a hierarchical manner, as advocated before by Reynolds [2]. Our experience with developing this higher level behavior on the whole is good: New traits can be fitted well into the hierarchy of impulses. However, some aspects of the implementation, such as goal switching and subgoal formulation require more experimentation to accomplish simulations that reproduce, in fair measure, animal predator/prey behavior.

Acknowledgments

This work has been partially supported by NSF grant CCF-0722210, by the Northwest Indiana Computational Grid and by the Art and Engineering Research institute of K-SAD. We thank Varun Ramachandran, a graduate student of Computer Science at Purdue, for helping with the simulation code during the design and implementation stages of this study. Seongdong Kim thanks Prof. Hoffmann for supporting a temporary visit of Purdue University and to the NAVER website to support special images.

References

[1] C. Reynolds, “Flocks, Herds, and Schools: A Distributed Behavioral Model,” Computer Graphics (Proc. of SIGGRAPH 87), July 1987

[2] C. W. Reynolds, “ Steering Behaviors for Autonomous Characters”, GDC 1999, San Jose Ca. 1999

[3] Seongah Chin, Chung-yeon Lee, Seongdong Kim, "Leadership and Self-Propelled Behavior Based Autonomous Virtual Fish Motion," pp.336-339, 2008 3rd ICCIT, IEEE CS December 2008

[4] I.D.Couzin, J.Krause, R.James, G.D.Ruxton, N.R.Franks, “Collective Memory and Spatial Sorting in Animal Groups,” J. Theor. Biol. 2002 (218)

[5] C. Hoffmann, V. Popescu, S. Kilic and M. Sozen, “Modeling, Simulation and Visualization: The Pentagon on September 11th,” *IEEE Comp. in Sci. and Engr.*, Jan/Feb 2004, 52-60

[6] C. Hoffmann and B. Davenport-Sypher, “Game Development in a Larger Context“, Microsoft *3rd Academic Gaming Days for CS Education*, Feb 28, 2008

[7] http://www.stolaf.edu/people/mckelvey/envision.dir/lotka-volt.html

[8] http://photo.naver.com/view/2009010221584797829

[9] D.J.T Sumpter, “The principal of collective animal behavior”, Phil. Rans. R. Soc.B doi:10.1098/rstb. 2005

[10] Brogan, D.C., Hodgins, J. K., 1997. “Group Behaviors for Systems with Significant Dynamics”. *Autonomous Robots* 4(1), pp137-153.

[11] D. Attenborough, 2002. “The Life of Birds”. *BBC Video*.

[12] J. Canny, 1988. The complexity of motion planning. MIT Press.

Seongdong Kim received the B.E. and Ph.D from Kwangwoon University in Computer Engineering, Seoul, in 1981 and 2001, respectively and M.S. degree in Computer Science from City University of New York in 1991. From 2002 to 2003, he was a research scientist of Computer Vision Research

Laboratory at University of Notre Dame. Currently, he is a professor of Department of Gameware at Kaywon School of Art and Design and serves as a board member of Korea Game Society (KGC) and Korea Computer Game Society (KCGS). He has published more than 40 technical articles. His interests are group behavior problems, distributed behavioral model, interaction with groups, 3D simulation game, group decision making in games.

Christoph Hoffmann received the B.S. in mathematics from University Hamburg, Germany, in 1968 and M.S. degree in mathematics from Indiana University in 1970 and Ph.D degree in Computer Science from University of Wisconsin in 1974. During 1984-1986, he stayed a visiting professor of Department of Computer Science at Cornell University. He joined Department of Computer Science, Purdue University where he is now professor. He is well known for his seminal contributions to geometric modeling and constraint solving. Recently, he has led a massive simulation effort that analyzed the 9/11 attacks on the World Trade Center and the Pentagon. A YouTube version of the WTC North Tower impact has been viewed more than 5 million times. He is a regular at the *Academic Days on Game Development in CS Education* conference, co-sponsored by Microsoft and Siggraph, and presented an overview paper on games at the 2008 conference.