Simulation Levels of Detail for Real-time Animation [†]

Deborah A. Carlson and Jessica K. Hodgins

College of Computing and Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
[debbie|jkh]@cc.gatech.edu

Abstract

When sufficient computing power is available, dynamic simulation can be used as a source of motion for real-time, interactive virtual environments. In this paper, we explore techniques for reducing the computational cost of simulating groups of creatures by using less accurate simulations for individuals when they are less important to the viewer or to the action in the virtual world. The less accurate, or lower level of detail, simulations can be dynamic with fewer degrees of freedom, hybrid kinematic/dynamic, or purely kinematic. As a test of the effectiveness of this approach, we implemented an environment with dynamically simulated legged creatures. Because the creatures switch smoothly among different levels of detail for the underlying simulation, we can achieve real-time performance for a larger group of creatures than would be possible if each creature were dynamically simulated. To be useful in this test case, the method must meet two criteria: the outcome of the game must be essentially unchanged and the viewer's perception of the motion must be the same. While it is not possible to make definitive measurements of these criteria, we assess the performance gain from using different levels of detail and make a preliminary assessment of the effect that the decreased accuracy has on the outcome of the sample game.

Keywords: level of detail, dynamic simulation, real-time simulation.

1 Introduction

With the increased computing power of current workstations, dynamic simulation has become a viable option for generating motion in interactive virtual environments. Dynamic simulation offers several potential advantages over other sources of motion by providing realistic and natural-looking motion in a variety of scenarios. In contrast to libraries of pre-computed or recorded motion, a simulation will respond interactively to changes in the environment and to the actions of the user.

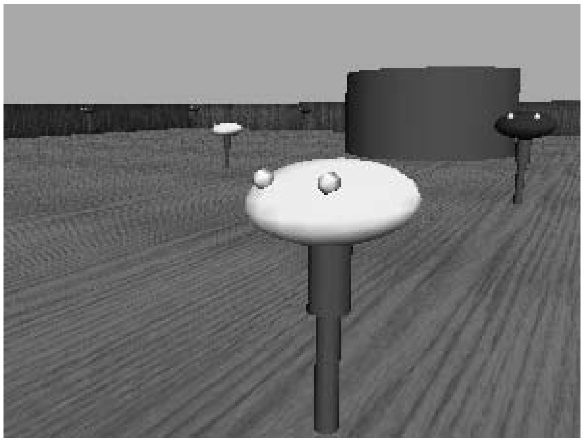

Figure 1: A frame from the environment used to demonstrate the computational savings achieved by using simulation levels of detail. The light grey creatures are simulated using rigid-body dynamics and the darker creatures are hybrid kinematic/dynamic.

Although computers are now powerful enough to compute the motion of such complex creatures as a rigid-body model of a human bicyclist in real time, dynamic simulation remains expensive. The computational cost is compounded when we want to use many interacting creatures to create interesting and compelling virtual environments. One approach to this problem is to distribute the creature simulations among many processors (see [15] for a description of one such system). A second potential solution is to select a level of detail or accuracy for each simulation that depends on such factors as the dynamic state of the simulation, its proximity to the important action in the scene, and its position in the user's field of view. In this paper, we explore the potential computational gains achieved by using multiple simulations at different levels of detail, a technique known as "multimodeling" in the simulation literature[6]. We borrow the term "level of detail" from the research on levels of detail in geometric models and rely on that work for inspiration in designing rules for switching between levels. We also explore appropriate criteria for success including the outcome of an example game and the viewer's perception of

---

[†] To appear in *Graphics Interface '97*

the motion in the environment. The testbed for our experiments is a virtual world in which a number of legged creatures attempt to escape a giant puck (figure 1).

The available levels of detail for a particular simulation can take such forms as pre-recorded motion sequences, procedurally generated motion, and dynamic simulations. In this paper, we consider simulations that are point masses, hybrid kinematic/dynamic, and full rigid-body dynamic simulations. Given that several levels of detail are possible in a simulation, we need rules for selecting a level of detail for each creature at each instant of time. The most appropriate rule may depend on the application domain. If the primary goal is visual realism, then the system should switch to simpler simulations when the creatures are out of view or too far away to be seen clearly. If the dynamic behavior of the creatures is important, then the system should select the most physically accurate simulation for creatures where dynamic events such as collisions are imminent.

2 Background

Several researchers have used geometric levels of detail to increase the maximum complexity of models in virtual environments while maintaining an acceptable frame rate. Heckbert and Garland[10] present a useful survey of existing techniques for geometric levels of detail. Although the techniques that researchers have used to develop lower resolution models are specific to systems for geometric levels of detail, the techniques they developed for selecting the appropriate model to render are relevant to systems that vary the level of detail of dynamic simulation.

Funkhouser and Séquin[7] use a cost/benefit heuristic to determine the appropriate geometric level of detail for each object in order to achieve a fast, constant frame rate in building walkthroughs. Their heuristic primarily relies on estimating the screen size of an object to determine the level of detail for the underlying model. In addition, the virtual environment developer is allowed to specify the inherent importance of an object. For example, walls are likely to be more important than pencils in the building walkthrough application. The viewer's focus of attention is taken into account by considering objects in the center of the view to be more significant than objects on the periphery. Quickly moving objects are regarded as less important because the viewer will perceive them as blurred regardless of the level of detail. The final factor they used in selecting a level of detail is hysteresis because it is essential to avoid the popping caused by rendering an object with a different level of detail on several successive frames. We can use similar criteria for deciding which simulation level to choose when visual accuracy and frame rates are important.

Researchers at the University of Pennsylvania[9] and at the Naval Postgraduate School[5] used motion levels of detail for real-time rendering of dismounted infantry within the Distributed Interactive Simulation protocol. The University of Pennsylvania system used scripted motion for a detailed human model with 134 degrees of freedom and generated motion scripts for models with 50 and 21 degrees of freedom. The system used an iterative, off-line process to map the posture of the most detailed model to the simpler models. Beginning with an initial configuration in which the joints of the detailed model were linearly combined, the system iteratively solved constraints to align landmark sites such as hand positions. To reduce the required storage, the system stored the sampled motion for lower level of detail models at a lower frequency. While we share with these researchers the goal of generating the motion of virtual characters in real time, we concentrate on levels of detail for simulations rather than for procedurally generated motion.

Earlier work in computer animation explored various simulation models and procedural approaches in an effort to reduce the cost of computing the motion of a single creature without adversely affecting the quality of the motion[3, 11, 12]. For example, Bruderlin and Calvert[3] relied on a simplified dynamic model and control algorithms to generate the motions of a walking human. The leg model included a telescoping leg with two degrees of freedom for the stance phase and a compound pendulum model for the swing phase. A foot, upper body, and arms were added to the model kinematically and made to move in an oscillatory pattern similar to that observed in humans. We use a similar approach for our hybrid kinematic/dynamic models.

Girard and Maciejewski[8] developed animation techniques for running and dancing legged creatures that used a point-mass dynamic model to control the position of the body. The legs of the creatures were animated using inverse kinematics and constraints to position the feet on the ground. Bruderlin and Calvert[2] took this approach a step further when they implemented a kinematic runner whose body followed a parabolic trajectory in the air. Incorporating data from the biomechanical literature, their system encapsulated information about how people run and gave the animator control over stylistic parameters of the running motion.

More recently, Perlin[12] used stochastic noise functions to create subtle human movements like shifting of weight and fidgeting while standing. The fundamental motion of each joint is sinusoidal with additive noise used to prevent the motion from becoming repetitive. The procedurally generated motion of the human figures can be computed in real time for use in virtual environments.

3 Simulation levels of detail

To demonstrate the effectiveness of levels of detail for simulated creatures, we built a virtual world in which a number of legged creatures try to escape a giant puck (figure 1). The puck slides passively around a rectangular court bouncing off the walls. The creatures attempt to avoid the puck, the walls, and other creatures. If a creature collides with an obstacle or another creature while trying to

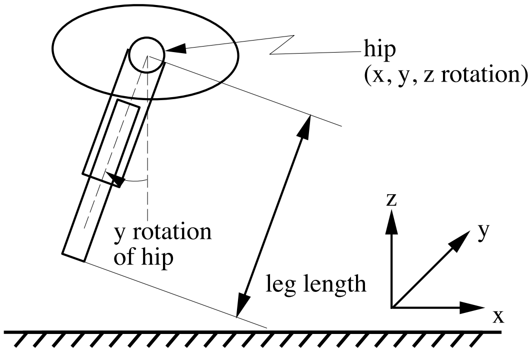

Figure 2: The reference angles for the degrees of freedom of the one-legged creature. The controlled degrees of freedom are the three degrees of freedom at the hip and the length of the leg.

escape, the collision is simulated dynamically and the creature may recover its balance or it may fall down. If it falls down, it is removed from the court. To keep the simulation load constant, creatures that are removed are restarted in another part of the court. This simple virtual world provides an environment in which to explore and evaluate levels of detail for the simulations and the rules for switching among them.

To reduce the computational cost of a dynamic simulation, we reduce the complexity of the dynamic representation of the creature. We incorporate three levels of detail: rigid-body dynamics, a hybrid kinematic/dynamic representation, and a point-mass simulation. For the one-legged creature, the most complete and computationally expensive level of detail is a rigid-body dynamic simulation with three bodies and four controlled degrees of freedom. The middle level of detail uses a point-mass model to determine the location of the body on the plane and a kinematic model to determine the joint angles for the legs given the current running speed. The lowest level of detail uses a point-mass model for the body and holds the leg at a constant angle. A similar hierarchy of levels of detail was also used for a bipedal creature with telescoping legs.

3.1 Dynamic model

We formulated the equations of motion for the legged creatures using a commercially available package[14]. The model is shown in figure 2. The locomotion algorithm controls flight duration, body attitude, and forward and sideways velocity. Flight duration is controlled by extending the telescoping leg during stance to make up for losses in the system. Body attitude (pitch, roll, and yaw) is controlled by exerting a torque between the body and the leg during stance. The forward and sideways velocities are controlled by the position of the foot with respect to the hip at touchdown. For a constant velocity, the foot is positioned in the center of the distance that the body is expected to travel while the foot is on the ground. To increase the speed, the foot is positioned closer to the hip. To decrease the speed, the foot is positioned farther from the hip. Raibert[13] gives the details of the locomotion control algorithm.

3.2 Hybrid kinematic/dynamic model

The second model relies on a simple dynamic model for the motion of the body and computes the motion of the leg kinematically. The position of the body on the playing field is computed using a point-mass model with an acceleration limit that closely matches that of the dynamic simulation. The body is oriented to face in the direction of travel. The height, roll, and pitch of the body and the length and hip angles for the leg are computed using a table lookup and linear interpolation. To create the table of body orientations and joint angles, we recorded motion from the dynamic simulation with forward velocities of 0, 1, 2, and 3 m/s. The joint angles were recorded at a rate of 30 samples/second for the period of time from one touchdown to the next. To calculate the kinematic parameters for other speeds and frame rates, the system linearly interpolates between the stored values. This simple interpolation works well because the running frequency is nearly independent of forward velocity.

For steady-state hopping, it is very difficult to visually distinguish between the motion generated by the full rigid body simulation and this hybrid simulation. When the creature is accelerating or turning, however, this simpler simulation violates physical laws during the flight phase by arbitrarily changing the forward, sideways, and angular velocities of the center of mass. Therefore, in some situations, the motion of the hybrid simulation appears much less realistic than the dynamic simulation. Figure 3 compares the velocity of the dynamically simulated creature with the velocity of the hybrid kinematic/dynamic creature in response to step changes in desired velocity. While the actual velocities are similar, the hybrid simulation responds more quickly to changes in velocity because it is able to change direction instantaneously while the dynamic simulation can only change direction when the foot is on the ground. Even small differences in velocity can result in substantial changes in final position after 8 seconds of simulation. Figure 4 shows the path on the ground plane of both simulations for this experiment.

3.3 Point-mass model

The lowest level of detail incorporates the same point-mass simulation for the motion of the body on the playing field but eliminates the table lookup and interpolation for the leg angle and pitch and roll of the body. Because this level of detail is so visually different from the other two, it is only appropriate for situations in which the creature is not visible but where the motion of the creature continues to influence the motion of visible creatures.

3.4 Switching among simulation levels

We explored two criteria for determining the most appropriate level of detail for each creature as

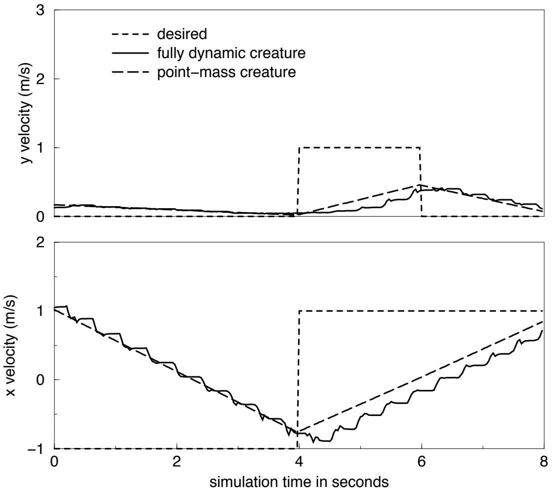

Figure 3: The velocities of the dynamic simulation and the point-mass simulation in response to a step change in desired velocity. The point-mass simulation responds to the step change with less delay. The velocity of the dynamic simulation remains nearly constant during the flight phase and changes only during stance. The velocity of the point-mass simulation increases linearly.

Simulation LOD	CPU Time (ms)
Rigid-body dynamics	2.4
Point-mass/kinematic joints	0.024
Point-mass/no kinematics	0.0025

Table 1: The computation time required for 0.033 second of motion for each of the three simulation levels of detail on a Silicon Graphics R10000 Maximum Impact workstation. The measured computation time includes the simulation and low-level control algorithms but not the computation time for graphics or for such higher-level behaviors as obstacle avoidance.

the motion occurs in the virtual world. The first criterion evaluates the importance of the dynamic behavior of the creature and chooses the simulation level of detail accordingly. Each creature or object has an area of influence representing the space where dynamic interactions are likely. For a creature or the puck, the area of influence is represented by a circle distorted in the direction of travel (figure 5). For the wall, the area of influence includes any point within a fixed distance of the wall. If a creature intersects with the area of influence of another creature, the puck or the wall, then the creature should be dynamically simulated. The areas of influence represent an approximate measure of the probability of a collision occuring within a short period of time.

The second criterion evaluates the importance of a creature to the viewer by determining whether it is in the field of view and if so, how close it is to the viewer. When a creature is not visible to the viewer or is in the field of view but not close to the viewer, a lower level of detail for the simulation may not affect the viewer's perception of the game. For creatures that are out of view, the point-mass simulation is used. For creatures that are in view but far from the viewer, the hybrid kinematic/dynamic simulation is used so that the distinctive motion of the leg is retained. Figure 6 shows an overhead view and the user's view of the playing field when this criterion is used.

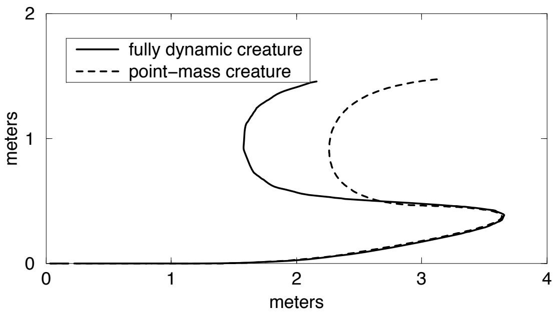

Figure 4: The trajectories of the two creatures on the ground plane during the 8-second simulation. Although the velocities of the two simulations are similar, the final locations of the creatures are substantially different. When the location of a particular creature is crucial to the outcome of a game, a different level of detail may change the outcome.

The switch between levels of detail in the simulation must be smooth to avoid distracting or annoying the viewer. For geometric levels of detail, smooth shifts in color and geometric blending between objects are used to make a smooth visual transition. With simulation levels of detail, the motion must be smoothly blended during a transition. We achieve a smooth transition by allowing switching only during a particular part of the flight phase of the running cycle. This strategy works well for this particular simulation and should work for any running or walking system, but it may not generalize easily to a more complicated, non-cyclic simulation. Interpolating joint angles between the two levels of detail for a few motion cycles may provide a better solution.

4 Results and Discussion

Table 1 shows the computation time required for each of the three levels of detail for the simulation on a Silicon Graphics R10000 Maximum Impact workstation. The computation times in this table include the simulation and low-level control algorithms but do not include rendering or such higher-level behaviors as obstacle avoidance. These additional computations will affect the reduction in computation time for a given level of detail.

Figure 7 shows the number of creatures that can

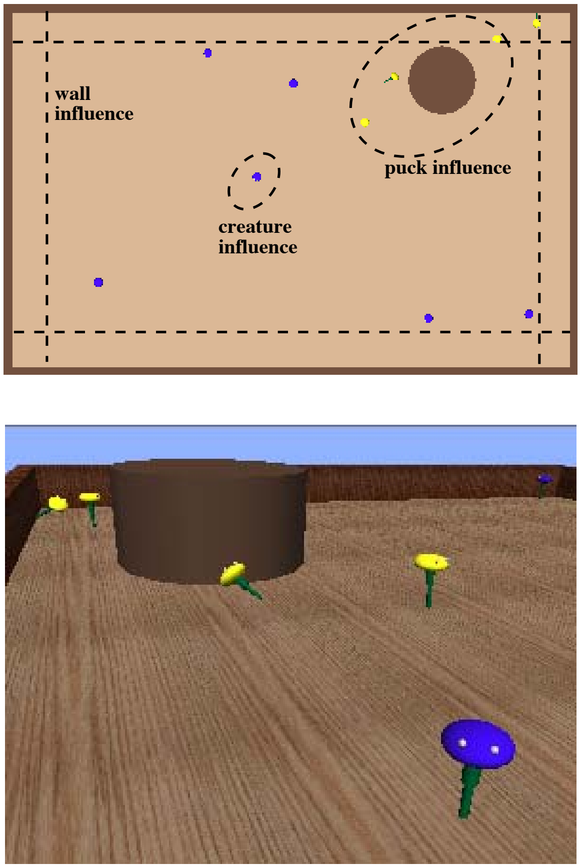

Figure 5: The top image is an overhead view of the court during a game. The bottom image shows the user's view of the playing field. Creatures within a specified range of the puck, walls, or other creatures use rigid-body dynamic simulation and are colored yellow. The rest of the creatures use hybrid kinematic/dynamic simulation and are colored blue.

be simulated in real time at 30 frames per second as a function of the percentage of creatures that use full dynamics. In this test, all of the creatures are visible and the lowest level of detail (point-mass simulation with no kinematic motion of the legs) is not used. At one extreme, using only dynamic simulations, the system can simulate 8 creatures at 30 fps. The system can animate 24 creatures when all of the simulations are hybrid. To show how the speedup is affected by the computation time for the Open Inventor graphics, the bottom graph in figure 7 illustrates the same test without graphics. Without the graphics, the system can simulate 12 dynamic creatures or 112 hybrid creatures in real time–more than a ninefold speedup. Simulation levels of detail, therefore, will provide a greater speedup in systems that use culling or have faster graphics pipelines.

Using the puck virtual world as a testbed, we measured how the frame rate was affected by the rules for selecting the level of detail for each creature.

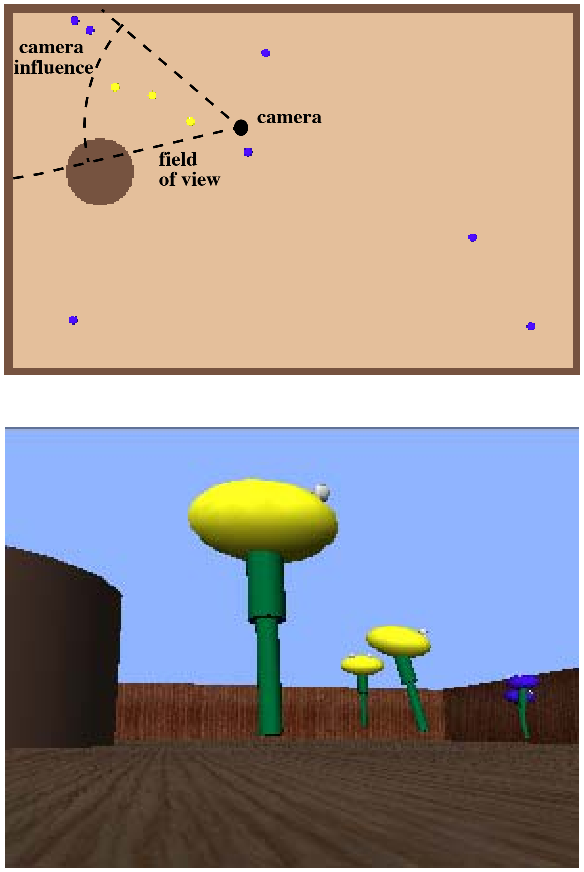

Figure 6: The top image is an overhead view of the court during a game. The bottom image shows the user's view of the virtual world. Creatures within the camera influence and inside the view cone are simulated with rigid-body dynamics and are colored yellow. Those outside the field of view use point-mass simulation and are colored blue. The remainder are hybrid kinematic/dynamic simulations and are also colored blue.

The first test involved varying a radius of influence surrounding the puck and therefore the number of creatures that were dynamic. Figure 8 shows the average and lowest frame rates for a 20-second experiment with the virtual world when the radius of influence of the puck is varied. In an actual application, the appropriate radius of influence could be conservatively determined by the average speeds of the creature and the puck.

For the second measurement, we used a moving camera attached to one of the creatures and always kept the puck in view. The radius of influence of the camera was varied to determine the effect of this switching rule on the performance of the system. Only the creatures within the camera's field of view and within a certain distance from the camera were dynamically simulated. For this experiment, the system used the lowest level of

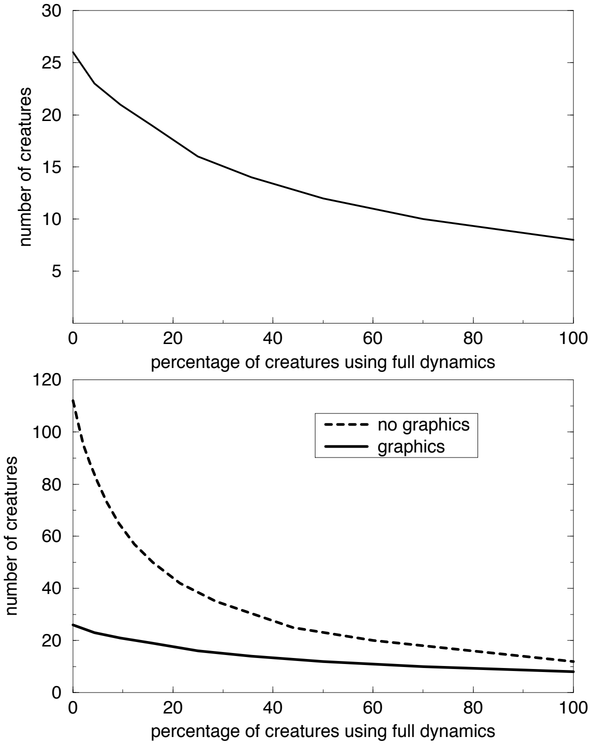

Figure 7: The top graph shows the number of one-legged creatures that can be simulated in real time at 30 fps. The number drops as the percentage of dynamically simulated creatures increases and the number of hybrid creatures decreases. The bottom graph shows the same experiment with and without graphics to demonstrate the effect of the graphics on simulation speed.

detail for the creatures that are out of view and we varied the distance from the camera at which the simulation switches between hybrid and dynamic. Figure 9 shows the average and lowest frame rates for a 20-second run of the environment.

Although these experiments provide a measure of the improvement in performance achieved through using simulation levels of detail, they do not assess how the quality of the viewer's experience is affected. One measure of success for a system such as this is whether the required computation can be reduced without affecting the visual quality of the motion[4]. Another measure is whether the fundamental behavior of the virtual world is changed. Brogan and Hodgins[1] explore the differences in the behavior of dynamically simulated one-legged creatures with the behavior of point masses in the context of multi-agent flocking behaviors. In our application, the outcome of the game might be an appropriate measure of the behavior of the virtual world. Figure 10 shows the change in the number of creatures that collide with the puck or the wall in a 20-second simulation

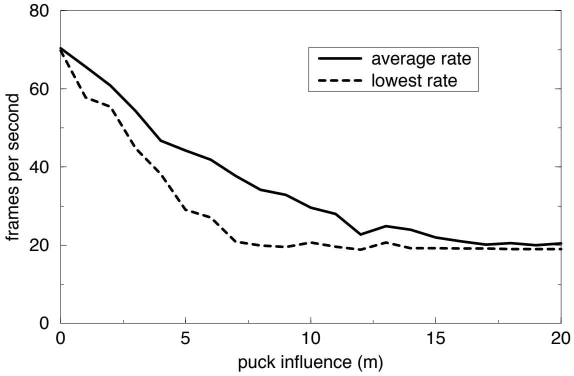

Figure 8: The average and lowest frame rate as a radius of influence for the puck is increased. When a creature is within the radius of influence of the puck, it is dynamically simulated; otherwise the hybrid simulation is used.

as a function of the radius of influence surrounding the puck. The dynamic creatures are less agile than the hybrid simulations and are therefore more likely to be forced into the wall or run over by the puck. Over time, this difference in performance substantially changes the state of the virtual world. Criteria for switching between levels of detail should take these factors into account in order to keep the outcome of the game as close as possible to the outcome when the most detailed simulations are used.

We plan to implement more complex switching rules to enable the system to make explicit tradeoffs between guaranteed frame rates and physical accuracy. For example, cost/benefit heuristics might be designed to assign a switching priority for each creature. Thus a consistent frame rate could be maintained by using lower levels of detail for the lowest priority creatures.

The simplicity of the one-legged creature simulation allowed us to simulate 12 creatures in real time on a single processor; however, this simplicity also constrained the number and variety of the levels of detail that could be implemented for this system. More humanlike simulations with 22-32 degrees of freedom run between real time and four times slower than real time on a MIPS R10000 processor. When we have computers that can run these more complex simulations in real time, we will be able to explore more levels of detail in the simulations. For example, a humanlike model might have several levels of dynamic simulation with varying numbers of degrees of freedom as well as several hybrid kinematic/dynamic simulations.

Several researchers have explored techniques for automatically generating geometric levels of detail. We believe that this will also be an important problem for simulation levels of detail. The simulation levels of detail described in this paper were designed by hand; however, we believe that by analyzing the

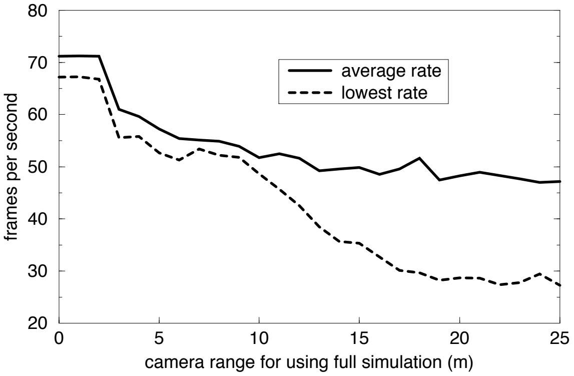

Figure 9: The average and lowest frame rate as the radius of influence of the camera is varied. When a creature is within the radius of influence of the camera, it is dynamically simulated; otherwise a hybrid simulation is used. When a creature is not in the field of view, the lowest level of detail is used.

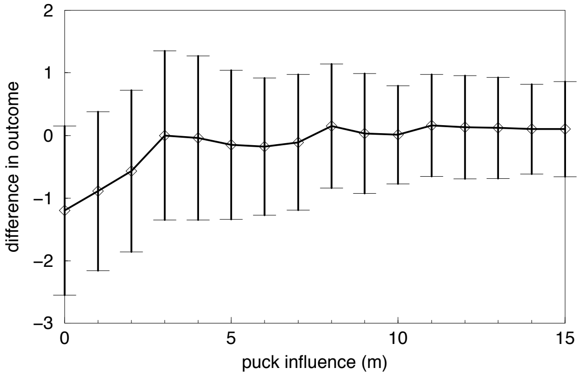

Figure 10: To maintain the realism of the environment, we need to determine if the simulation levels of detail affect the outcome. To compute the data points in this graph, we ran 100 trials for each radius of influence for the puck. Each of the 100 trials ran for 20 seconds starting from a unique initial condition. For each trial, we computed the number of creatures that collided with the puck or the wall and compared that number to the number of collisions for the identical trial when all creatures were full rigid body simulations. Each data point represents the mean and the variance for the difference in the collision count. The collision count remains approximately constant as the radius of puck influence is decreased from 15 to 8 meters. When the radius is less than 8 meters, the number of collisions decreases, reflecting the greater agility of the hybrid model.

structure of the creature, levels of detail could be constructed automatically for a broad class of systems.

Acknowledgments

This project was supported in part by NSF NYI Grant No. IRI-9457621, Mitsubishi Electric Research Laboratory, and a Packard Fellowship. Thanks to Bob Brown and Alias|Wavefront for providing support and facilities.

References

[1] D. C. Brogan and J. K. Hodgins. Group behaviors for systems with significant dynamics. *To appear in The Journal of Autonomous Robotics*, 1997.

[2] A. Bruderlin and T. Calvert. Knowledge-driven, interactive animation of human running. In *Graphics Interface '96, Proceedings, Toronto*, pages 213–221, May 1996.

[3] A. Bruderlin and T. W. Calvert. Goal-directed, dynamic animation of human walking. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 233–242, July 1989.

[4] S. Chenney and D. Forsyth. View dependent culling of dynamic systems in virtual environments. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, 1997.

[5] C. A. Chrislip and J. F. Ehlert, Jr. Level of detail models for dismounted infantry in NPSNET-IV.8.1. Master's thesis, Naval Postgraduate School, 1995.

[6] P. A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds.* Prentice Hall, Englewood Cliffs, NJ, 1995. ISBN 0-13-098609-7.

[7] T. A. Funkhouser and C. H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 247–254, August 1993.

[8] M. Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 263–270, July 1985.

[9] J. P. Granieri, J. Crabtree, and N. I. Badler. Production and playback of human figure motion for 3d virtual environments. In *VRAIS '95 Conference Proceedings*, pages 127–135, 1995.

[10] P. Heckbert and M. Garland. Multiresolution modeling for fast rendering. In *Proceedings of Graphics Interface '94*, pages 43–50, Banff, Alberta, Canada, May 1994. Canadian Information Processing Society.

[11] H. Ko and N. I. Badler. Straight-line walking animation based on kinematic generalization

that preserves the original characteristics. In *Proceedings of Graphics Interface '93*, 1993.

[12] K. Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, 1995.

[13] M. H. Raibert. *Legged Robots That Balance.* MIT Press, Cambridge, 1986.

[14] D. E. Rosenthal and M. A. Sherman. High performance multibody simulations via symbolic equation manipulation and kane's method. *Journal of Astronautical Sciences*, 34(3):223–239, 1986.

[15] A. Singla, U. Ramachandran, and J. K. Hodgins. Temporal notions of synchronization and consistency in beehive. In *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures*, 1997.