published in:
Computer Animation and Simulation '96(Proc. Eurographics Workshop), R. Boulic and G. Hégron eds., Springer Wien, New York, 1996, pp.183–197.

Plausible Motion Simulation for Computer Graphics Animation

Ronen Barzel
University of Washington

John F. Hughes
Brown University

Daniel N. Wood
University of Washington

Abstract

Accuracy is the ubiquitous goal of dynamic simulation, in order to yield the "correct" motion. But for creating animation, what is really of interest is "plausible" motion, which is somewhat different. We discuss what we mean by *plausible* simulation, how it differs from "accurate" simulation, and why we think it's a worthwhile area to study. The discussion touches on questions of *physically plausible* vs. *visually plausible* motion, plausible simulation in a noisy or textured environment, and probability measures for motion, as well as issues for forward and inverse problems.

1 Introduction

Simulation is generally used in the context of a predictive model of behavior: given a precise description of a real-world situation, try to determine computationally what would really happen. When designing airplane parts, for example, accuracy of the model and of the simulation are critically important.

However computer graphics animation has different needs, requiring a slightly different outlook. In particular, we claim:

- For computer graphics animation we don't need a predictive model of what will happen. The creator of an animation decides a priori what is to happen: the purpose of simulation is to provide tools to make it happen in a way that seems real.
- Simulated motion often looks "sterile," because it lacks the variation caused by small details that are left out of the models. These are generally omitted because their inclusion would render the simulation computationally intractible, or because simulation methods for handling such detail are not known.

We thus introduce the idea of *plausible* motion: motion that could happen, given what is (un)known about the system. Many motions may be plausible for given conditions; this can give us latitude in creating or choosing a particular motion that is desired.

We have found that recasting the animation simulation problem domain to be one of plausible rather than accurate motion opens up a variety of interesting and promising directions for investigation. We will discuss some of these directions, and give some preliminary results. We hope to stimulate interest in an area that we have only just begun to explore.

2 Thought Experiments

We discuss some simple thought experiments, to motivate the claims of the last section and to examine the notion of plausibility. The experiments lead to various conjectures about implementing and taking advantage of plausibility.

2.1 The Importance of Detail

Thought experiment: A Superball

Consider a typical CG simulation of a sphere released from rest above a ground plane: it bounces up and down above a single point on the plane.

Now consider a real-world superball that is held and released above a floor: it bounces and skitters every which way. Repeat the experiment; the ball will travel on a very different path, but the overall "character" of the skittering will be roughly the same.

There are a variety of factors that probably contribute to the skittering of a real-world superball: slight initial spin imparted when the ball is released, eccentricities in the shape of the ball, inhomogeneities in its mass distribution, non-horizontal floor plane, non-planar floor, and dirt and other particles on the floor.

Thinking about the bouncing ball, we observe:

- Simulations of CG models tend to look mechanical and sterile, no matter how accurate the computation. This is because the model is itself "sterile": a perfect flat plane and a perfect sphere.
- It would be hard to take into account all factors and imperfections to accurately model and simulate a real superball on a real floor. Furthermore, no matter how carefully we measure the superball and floor in question, there will be some uncertainty in the initial conditions (unless we build a precise mechanical release mechanism rather than simply dropping the ball by hand), which will lead to uncertainty in the resulting motion path.
- It doesn't matter exactly which motion path the ball takes if one is only concerned with the *appearance* of realism. Each real-world trial is different anyway. But it is essential that the path displays the same type of skittering that is caused by the imperfections in the real world.

It may be too hard to accurately *simulate* all the "details" of the real world. But we get a big win if we can *mimic* the details: we would make plausible, non-sterile motion.

Conjecture: We can mimic real-world imperfections by starting with the usual flat, smooth CG model, and introducing an appropriate amount of variability into the simulation process. This approach is well-known in the world of rendering: One provides *texture* to make models appear more realistic. Texture adds detail most effectively when the "character" of the detail is important, but the actual data in the detail is not (e.g., it matters that the wall is marble, but not where the veins in the marble lie).

2.2 The Futility of Accuracy

Thought experiment: Rolling Dice (a)

Consider rolling a die to see who plays first in a game of Monopoly. This extremly simple physical system—roughly, a cube and flat plane—is commonly considered to be a good source of random numbers.

Thinking about simulating a die roll leads us to two observations:

- Since behavior of this system is random,[1] there's no point in trying to simulate it "accurately."
- A typical simulation of a rolling die would produce the same result each time it is run (for a given set of initial conditions). In some sense, to really be "accurate," a simulation should produce a different result each time it is run.

Since repeatability is a desirable property of computer programs, having a simulation truly produce random results is probably not ideal. More useful—though probably hard to compute—would be a simulation that reports (in some form) the space of possible results, allowing the user to choose a specific element as desired; the program could also choose one arbitrarily[2] or randomly.

Thought experiment: Rolling Dice (b)

Consider the following scenario: a pair of dice are on a table, showing seven; something knocks them to the floor, they end up showing "snake eyes." This is a plausible occurrence, but one that is hard to arrange or predict.

Suppose we want to create an animation of the above scenario. To produce a believable motion for the dice, simulation is a natural choice. But thinking about how to meet the final-state condition, we note:

- Since behavior of this system is random, we can't expect an accurate predictive model. And we can't expect to easily adjust the initial conditions to guarantee the desired final results.
- To create an animation, we don't need a predictive model, since we know what we want to have happen. But we do need a way to create a plausible motion.

One might imagine trying to create this animation by simulating it repeatedly with various parameter values—since the odds of "snake eyes" are 1:36, we'd expect that within, say, 50 tries we'd get the desired result.

But what if the director specified not only the initial and final scores of the dice, but also their positions? We now have a highly constrained (or multi-point boundary value) problem, for a highly unstable system.

Conjecture: The easiest way to create an animation of scenarios such as the above is to exploit the variability and instability within the system: At each moment when there is variability, can adjust the outgoing state within the bounds that are derived from the plausible set of incoming states and the instability in the system.

[1] More precisely: given the limits of precision in the initial conditions and the computation, and given the instability in the system, the results of this (or comparable situations) are provably indeterminate [HS92].

[2] Thus returning us to the usual behavior of simulation.

2.3 The Flexibility of Believability

Thought experiment: Baseball Hit

Consider a viewer in the bleachers watching a baseball[3] pitch, swing, and hit. Can the viewer guess where and how far the ball will travel—straight to the left fielder? over his head? inside the foul line? out of the park? Can the viewer measure how high the ball flies, or whether it travels in a parabolic arc?

To create an animation of a baseball, we can probably choose from a wide range of trajectories—even ones that are physically infeasible—without compromising the believability (for example, have the ball "hang" in the air for a moment longer than it ought to, to increase the dramatic tension).

Again, making a few observations:

- People are not always terribly accurate predictors of motion. (How many novice outfielders have run forward to catch a fly ball, only to have it pass over their heads?)
- In some cases, people are willing to accept surprising behavior by relying on unseen forces: in particular, the effects of wind on the ball are significant on balls hit a long way, and miscalculations and fielding errors can often be blamed on the wind.
- Believability depends on viewing angle. From the bleachers above home plate, the initial direction (towards left or right field?) is probably not as free to be adjusted as is the initial elevation (grounder, line-drive, or fly ball?), while from behind first base the opposite would be true.

Of course, there are limits to how much variation can be introduced before breaking the viewer's belief that the simulation is "real." For example, if the batter bunts, holding the bat still, it would be implausible for the ball to gain enough energy that it could fly to the outfield.

Conjecture: For computer graphics animation, we can in many cases introduce ad-hoc variation of motion without comprimising believability.

3 Characterizing Plausibility

In the previous section we considered some intuitive notions about plausibility. We now attempt to characterize plausibility somewhat more precisely.

3.1 Sources of Variability

There are several factors that contribute to variability in a model and simulation:

- Numerical error in computation. The result of a solver is one element of the set of solutions accurate to within given tolerances—but there may be others.
- Approximations in the abstraction. (Are the bodies really rigid? How appropriate is the Poisson model of collision [KR66] or the Coulomb model of friction [Rei71]?)
- Inaccuracy in the data. (How precisely do we know the mass? How accurate is the initial velocity?)
- Missing details in the model. (Is the ball really round? Is the floor really flat?)

[3] Any Europeans in the audience, consider instead kicking a football or soccer ball.

These factors mostly cause continuous variability in the solution. But collisions introduce an entirely different class of variability: Each collision can non-linearly magnify all the other variabilities, producing macroscopic, discontinuous effects due to small changes in parameters.[4] Thus we have an additional source of variablitiy:

- Instability in the system.

Conjecture: Since collisions can magnify invisible variations, in many cases it may be sufficient, for generating plausible motion, to neglect explicit consideration of the "primary" variations, and introduce ad-hoc variation on the results of each collision instead.

Slightly abusing the English language, we refer collectively to the various sources of variability in a system as the *variabilities* of the system.[5]

3.2 Physical vs. Visual Plausibility

We consider a motion path to be *physically plausible* if it lies within the range of motions allowed by known error bounds on the above variabilities.

We consider a motion path to be *visually plausible* if it looks convincing. This is of course a weak definition, since it depends on a variety of perception and cognition factors, or more broadly, on who's doing the looking and in what context. Still, we can attempt to make some general characterizations of what makes motion visually plausible or implausible:

- In most cases, visual plausibility seems to require instant-to-instant "correctness." In our context, we can consider motion to be visually plausible if it lies within the range of motion allowed by arbitrarily chosen bounds on the above variabilities, so long as the end result looks convincing. Visual plausibility can allow temporal variation of parameters that would otherwise be considered to be constant (e.g., an object could possibly change mass slightly over time) .

- Visual plausibility may depend on the state of system and viewing parameters: If a ball approaches a surface perpendicularly, it should leave mostly perpendicularly. But if it arrives at, say, 47 degrees it can probably leave anywhere from 40 to 55 degrees without the viewer noticing—more if the ball is traveling directly away from the camera or if the camera is aimed along the wall thus confusing the viewer's ability to judge angles.

- Reliance on invisible forces can help provide visual plausibility, if the viewer expects that such forces might exist (e.g., wind effecting a baseball).

- Reliance on invisible forces can reduce visual plausibility, if the viewer doesn't expect such forces to exist.

- Visual plausibility depends on how well the viewer can see. Objects that are moving very fast, or are are very far away, or are obscured, or are poorly lit, provide extra opportunities for variability.[6]

In general, visual plausibility is a looser requirement than physical plausibility. But visual plausibility may occasionally be a stronger requirement: physical systems are sometimes non-intuitive, or behave unexpectedly due to invisible or otherwise unknown internal state. There may be some cases in which a visually

[4] Indeed, Huberman and Struss [HS92] describe a simple billiard-table-with-obstacle system whose behaviour is chaotic in large areas.

[5] The more obvious terms such as *variables* or *parameters* would suggest a model of the system that is more formal or complete than we wish to employ.

[6] Thus to some extent, we can consider visual plausibility to be the same as physical plausibility: motion allowed by known tolerances—but based on the viewer's perception of the system, rather than the model-builder's knowledge.

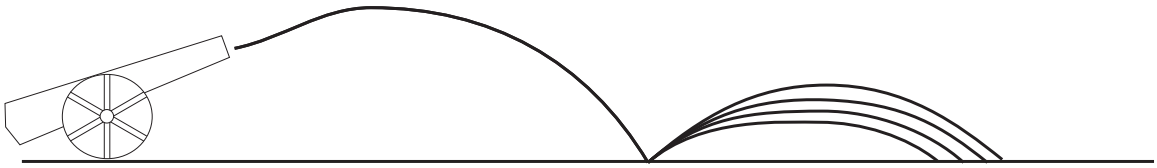

Figure 1: The set of plausible paths for a cannonball that bounces once. Here, the uncertainty in the initial conditions is very small, so the first arcs are nearly identical. Following the bounce, the subsequent arcs are many and varied.

plausible simulation has to "break" physics—to do something not allowed by physical law—to avoid surprising the viewers.

What about the limits of visual plausibility? It would be interesting to learn how such things as viewing angle, speed, and impact angle affect viewers' impressions of plausibility, perhaps by conducting user studies. In general, however, it is likely to require the storytelling talent of human animators to know just how much implausibility or coincidence one can "get away with" in a given context.

Finally, note that we're not considering so-called "cartoon physics"—the Coyote walking off a cliff for several paces before falling, or Ricochet Rabbit's bullets chasing a villain around corners—to be within the realm of visual plausibility. The point of such gags is exactly their implausibility.

3.3 Motion Path Cones

The behavior of a system can of course be described as a path through state space or phase space [FW80]. We consider the plausible motion of a body to be a bundle of paths, or generalized cone, through state or phase space, as illustrated in Figure 1. The bounds of the cone are determined by the ranges of the variabilities in the model.

More precisely, a motion path cone starts at the region of phase space that describes the initial conditions (a region rather than a point, since we include error bounds or tolerances). We sweep out a volume by extruding the region over time, evolving it in keeping with physical law.[7]

Cones do not necessarily continue indefinitely: If friction slows down the object, it may come to rest, as in the case of the cannonball of Figure 1.

Whenever the path cone of a body intersects some other body so that they can collide, there is a branch point in the plausible motion. Thus there is a tree of cones, with "shadows" of obstacles in the parent cone.

The more detail or precision we have in the model, the narrower the cones. However, even with very narrow initial cones, the instability caused by collisions can quickly widen the cones farther down the tree. If the object is confined to some bounded region of space, in many cases the cone may widen to cover all points of the space [KMS85].[8]

How does all this relate to simulation? A typical initial-value forward simulation chooses one path from within the tree of path cones. In thinking about simulations, it may often be worth remembering that any given solution has neighboring paths within the cones, that describe other plausible solutions. Although it is not in general feasible to compute all complete path cone trees—they can clearly grow too large and complex to represent, compactly or at all—in some cases computing a subtree might well be feasible and useful, as will be discussed below. If to find the "right" animation one is going to have to simulate repeatedly, then it

[7] Note that the boundaries of the cone aren't necessarily determined by the conditions at the start; there may be continuously variable parameters such as atmospheric disturbance or gravity that can affect the boundaries.

[8] Although not necessarily all points of the phase space. For example, a billiard ball, bouncing about on a frictionless pool table with bumpers that absorb 10% of the energy on each collision will (with high probability) eventually pass very near to every point of the table, although only a small initial segment of its trajectory will be covered at a high speed.

might reduce the overall computation to compute the cone tree initially, and then simply pick one path within it that meets one's goals.

3.4 Probability and Plausibility

While a wide range of motion paths may be plausible, are they all really likely? We consider probabilities in conjunction with motion paths.

We start by introducing probability distributions as a model of underlying variability. For example, in a collision with a rough surface, the statistical variation in a normal vector might have a Gaussian distribution.

These underlying distributions induce probability distributions over each path cone. At each branching point in the tree of cones, we propagate probability by integrating the parent probability over the cross-section of the target area, to get the probability p of taking that branch. The probabilites in the child branch are scaled by p; the probabilities for paths not following that branch are scaled by $1 - p$.

Of course, in a continuous space of motion paths, the probability of any given trajectory is negligible (even if it's a pretty reasonable choice). This can be addressed by defining the probability *density* over the cone, and only assigning probability to *sets* of motion paths, by integrating the density over the range of paths in a set.

In discussing what makes a path acceptably plausible, it can be useful to take a Bayesian approach [HU93]. That is, if a viewer saw a given path, would the viewer's belief about the simulated world change? If, for example, one sees a ball dropped on a gravel driveway, and it took an odd bounce to the North, one might think nothing of it ("there was probably a stone that caught it just the wrong way"). But if it took seven odd bounces in a row, all to the North, one would begin to think that something was "rigged." From a Bayesian viewpoint, the implicit hypothesis that the gravel's normal vectors are uniformly distributed becomes less likely as one observes the string of unusual bounces.

Having a probability metric for path cones has the potential to be useful for minimizing computation. For example, in many cases, rather than compute an entire tree of path cones, it would be appropriate to compute only those that represent reasonably likely occurrences.

4 Applications

We discuss here a few ways in which the ideas of plausible motion can be applied.

4.1 Simulating with Texture

Perhaps the most basic idea discussed thus far is that for simulations to look realistic, they must add "texture" to the otherwise too-simple models, where we use the term "texture" in analogy with the texture-mapping commonly used in rendering: apparent detail added as an after-effect. Texture in image generation is produced by texture mapping of scanned images, by procedural textures, and by randomly-generated textures (among other methods), and we can do likewise. As per the conjecture of Section 3.1, we concentrate the texture in collision parameters:

- We can plausiblly forward-simulate bouncing bodies by simply adding a certain amount of ad-hoc random variation to the initial conditions and to the normal vector associated with each bounce.
- We can perform experiments, e.g., measure real superball bounce directions, and use a statistical tabulation of the results as a source for distribution of variability, similarly to rendering's bidirectional reflection distribution functions or microfacet slope distributions [FvDFH90].

- We can borrow directly from rendering, and use texture maps and bump maps—in many cases, the very maps that are used by the renderer. For example, we can vary the coefficient of restitution based on surface texture, and use the bumped normal for the collision.

We suspect that the best results will arise from combining randomness with stored or computed textures. For example, wherever there is a bump, widen the range of normals. Thus for a tile floor, with raised tiles bedded in grout, we might assign somewhat variable normals for the grout region, and substantially variable normals in the areas where the bump-map tells us that the normal is changing rapidly (i.e., on the curved edges of each tile).

In adding texture to simulations, we need to be careful not to add (too much) energy to the system, as this would likely destroy (visual) plausibility. This depends on whether there are active elements in the system, such as pinball machine bumpers.

4.2 Animation Control

By exploiting variability we have the potential to give animators direct-manipulation style control over animation, while automatically maintaining physical plausibility. We describe a few approaches:

- *Motion construction.* A program shows the user the next motion path cone for an object, given its current state and the various variabilities; the user chooses from within the cone. Thus the user builds the animation step-by-step.

- *Motion adjustment.* Given the motion path cones describing a particular path (which may have been computed by forward simulation or built step-by-step), the user can interactively manipulate the final position, and the program automatically propagates the necessary variation back up along the cone tree. The user might, in the course of doing this, come up against the cone boundaries. This is analogous to inverse kinematics (IK) limb manipulation with joint limits.

- *Constrained Motion adjustment.* The program finds the plausible solutions of a constraint problem (e.g., a two-point boundary-value problem such as "the ball bounces from here to there"), showing the user the set of path cones that all meet the constraints. The user can choose among them, again like manipulating an IK jointed limb that is fixed at both ends.

In the above cases, lacking an automatic metric for visual plausibility, we can simply provide the user with a knob or knobs to widen (or narrow) the ranges of the variabilities.

All the above could also compute and display the likelihood of each chosen solution, if we have probability distributions for each variable[9] (or, perhaps more informative, show the likelihood that, given such a motion-path, the user would believe the hypothesis that the chosen values for the variabilities are actually uncorrelated.)

The computation of motion path cones is reminiscent of cone traced rendering [Ama84], and indeed, many of the ideas there have analogs in motion path cones.

4.3 Motion Synthesis

The motion synthesis problem, creating animation to meet desired goals, is often cast as an *inverse* simulation problem: given the desired end result, what initial conditions and forces should be applied?

[9] One might imagine using this this type of system for non-animation applications; e.g., car crash reconstruction: what is the likelihood that a given scenario could happen? But this is a case where getting the correct answer can really matter, and as such our loose discussion of plausibility is not appropriate. (Also note that cars contain drivers, and are thus active volitional objects that can vary their behavior due to internal state, greatly complicating the problem.)

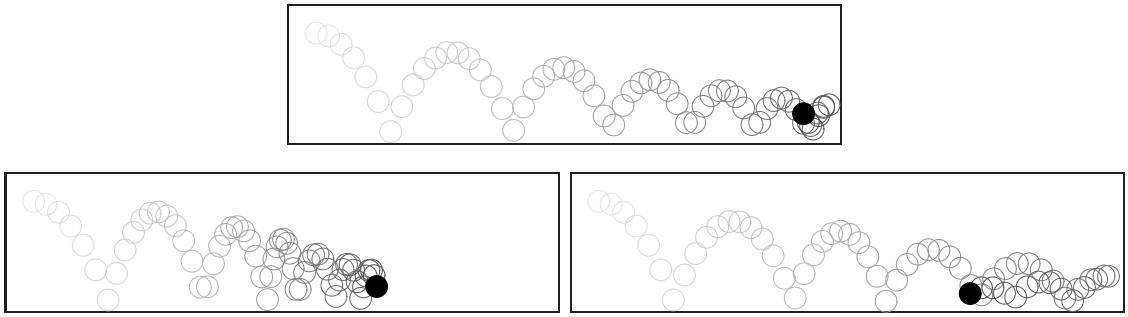

Figure 2: Simulation of a bouncing ball in a box, side view. *Top:* No texture added—the bounce is regular. *Bottom:* Collision normals are perturbed randomly up to 8 degrees.

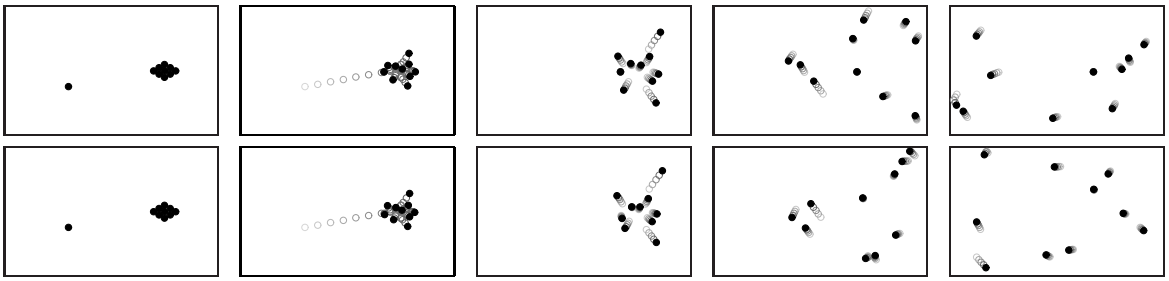

Figure 3: Simulation of a 9-ball pool break. *Left:* "Accurate" solution. *Right:* Collision normals are perturbed randomly by up to 8 degrees.

Inverse problems can be very complex. For example, Tang et al. [TNM95] describe a simple system, in which one is to have the balls on a billiard table end up in a particular configuration, and the problem is to determine appropriate initial velocities. This is a daunting task,[10] made particularly difficult by the search for accurate solutions, which are highly unstable and sensitive to variation of the initial velocities.

From our point of view, slight errors in the initial velocity can be masked by using a system that allows for variabilities. At first glance, adding such variabilities is both good and bad: it increases the size of the solution space, but also increases the number of degrees of freedom to be searched.

We note that the degrees of freedom added by variabilities are not only in initial conditions, but are distributed throughout the simulation: we can "tune" each bounce. Thus to some extent, we can convert the difficulties of solving a two-point boundary value problem into a simpler one of leaping from stepping-stone to stepping-stone.

There is one further point to be made: it's tempting to say "Of course adding variability makes the problem easier—it lets us cheat to get the result we want!" But if the variabilities we include are physically plausible, then we are not cheating at all—the solution we get is one which, to the limits of our understnading of the system is one of the many plausible solutions that *could* arise. And if we're in the domain of visual plausibility, then it's OK to cheat!

[10] Tang et al. describe an algorithm to reduce their roughly 10^{13} candidate solutions to only 10^6 trials that must be searched.

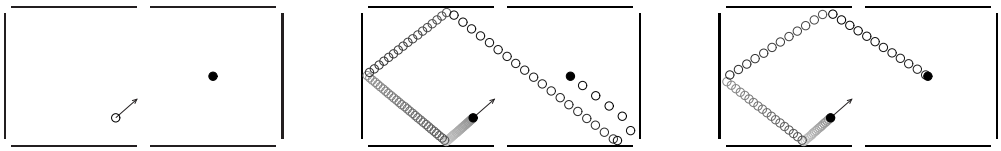

Figure 4: *Left:* The problem: give the ball (black circle) an initial velocity so that ends up in the desired position with the desired velocity; there is no "accurate" solution on a table with friction. *Middle:* Introducing a variability of 2 degrees to collision normals allows two "five-rail" shots. *Right:* Increasing the variability to 5 degrees finds a simpler "three-rail" solution.

5 Preliminary results

We have just begun to explore the ideas of plausible motion simulation. Our first steps have been to attempt to validate some of the conjectures for the simplest of test cases: perfect spheres on flat surfaces. We report here the results of four preliminary experiments and what we have learned from them.[11]

The first two experiments are simple checks of the conjecture that adding random variability to the simulation can create plausible motion "texture," and moreover that we can do so by ad-hoc variation of collision parameters. The second two experiments test the conjecture that inverse problems are easier if we allow variability.

5.1 Bouncing ball

First, we simulated bouncing balls, as described in Section 2.1. The surface normal at each collision could be perturbed randomly in a solid angle. The size of the angle was adjustable. Figure 2 shows a side view of one ball bouncing, with and without random texture.

The results of this test are straightforward:

- As expected, we found that adding a small amount of variation looked plausible and not as mechanical as the unperturbed control.
- The variation could be adjusted up to roughly 8 degrees without breakdown of plausibility.

The motion of this experiment, although interesting, instantly suggests the richness that could be added via texture mapping, so that we can have areas of gravel with higher variability, and so forth, as per Section 4.1. We have not yet implemented this, but expect it to work well.

5.2 Pool Break

Next we simulated billiard balls, in particular a "nine-ball break"—nine balls are arranged in a diamond shape, and a tenth ball is hit into them. As before, the collision normals could be subject to random variation (in the plane). The initial break is asymmetric. Figure 3 shows the results.

- In this case, the complexity of the underlying non-random system coupled with asymmetry in the initial conditions gives a plausible, non-mechanical feel, but it does so only once. When the simulation is repeated the results repeat exactly, quickly creating a mechanical feel.
- Adding roughly 10 degrees of random variation was completely unnoticeable in terms of the visual plausibility of the motion. But now the simulation is different each time, succesfully keeping a "natural" feel.

[11] Animations of the results are available online from: `http://www.cs.washington.edu/homes/ronen`

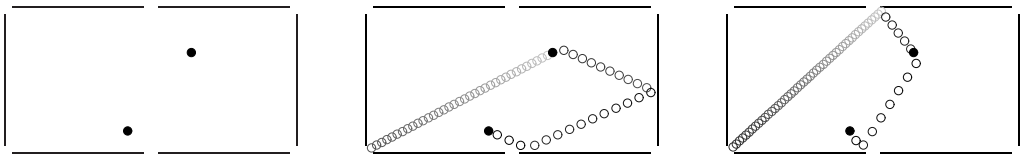

Figure 5: *Left:* A cue ball, and a target ball to sink in the corner pocket. *Middle:* Solution without perturbing normals. *Right* Another solution introduced with 8 degree perturbation of collision normals.

From this we learn that in an already complex system, introducing extra variability "texture" does not harm the visual plausibility. This frees us to introduce variability into motion synthesis problems, as discussed in Section 4.3 and tried in the next two experiments.

5.3 One Pool Ball

We attempt a simple motion synthesis problem: The cue ball starts at rest; we strike it with the pool cue, and we want it to reach a given location traveling in a given direction (without "scratching" by entering a pocket).

Our simulation includes frictional losses as the ball rolls across the table, inelasticity in the collisions with the walls, and a maximum initial speed. These together effectively prune our search and make it far simpler than the problems considered by Tang et al. [TNM95], whose balls moved on a frictionless surface with no energy losses, and thus could continue bouncing forever.[12]

Our algorithm is a backwards search: we start at the desired end position and velocity of the cue-ball, and draw a (very narrow) cone of trajectories along which it must travel to arrive at that position and velocity. When this cone reaches a wall, the tree branches as the cone reflects. (Note that the cone reflects separately from each half of the long walls, since the ball can bounce from the walls but not from the side pocket.) The reflected cones include the variability of reflection, and therefore spread faster than the parent cone.

We search recursively, computing the required speed in each cone given the frictional table and collisions, and prune the search when the required speed exceeds a predefined maximum. All successful results (those where the cone encloses the ball, and the ball speed is below the maximum) are reported. For each match, we trace forward from the branch to the root of this cone tree, piecing together a sensible path. We start by connecting the cones roughly apex-to-apex, then perform a relaxation process to minimize the overall eccentricity of the collisions.

As illustrated in Figure 4, it is easy to encounter circumstances in which the "accurate" approach has no solution at all. But adding as little as 2 degrees of variation at collisions makes it easily soluble.

For this test, the variability approach works well: allowing a small amount of variation per collision quickly builds up enough controllability, due to magnification at subsequent collisions, that the cones eventually encompass the solution. As we mentioned before, this isn't solving the problem by "cheating," it's finding a solution that is plausibile within the stated limits of knowledge of details in the model.

5.4 Two Pool Balls

Finally, we consider a more complex problem: given a cue ball and a target ball, we must "sink" the target ball into a chosen pocket, by giving the cue ball an initial velocity.

We use the same algorithm as used in Section 5.3, but here we start with a cone from the pocket, and have branch points additionally when a cone encounters the target ball. Some results are shown in Figure 5.

Notice that in this case, the instability of the ball-to-ball collision allows many solutions, even without perturbing normals: (ignoring rotation) the cue ball can send the target ball towards, say, the East, by hitting

[12] Interestingly, having a more complex model simplifies the computation!

Figure 6: Two pool shots. Both are physically plausible. The one on the right is a much harder shot.

the target on its west side; the cue ball can be headed in any direction from due East (direct hit) to almost vertical (glancing hit).

When we introduce variability into the problem, we do find many new solutions. However, these solutions tend to be on the fringes of acceptability—situations such as the lower right of Figure 5, in which the "accurate" solution would scratch, while the variability allows one to *just barely* make the shot by using the end of the rail. This result is perhaps somewhat disappointing, but not surprising in retrospect: the wide spread of incident cue angle at the collision overshadows the benefits of the small additional spread variation we introduce.

In using our solver, we see that many solutions (in both the "pure" case and the one with variability) are, although physically plausible, certainly in the realm of the remarkable. Consider, for example, the two shots shown in Figure 6. The first, in which the cue ball hits the target ball and sends it directly into the pocket, is the sort of shot that anyone can make. The second, in which the cue-ball hits four rails and finally grazes the target ball, causing it to roll slowly into the corner pocket, might have been executed by a well-tuned machine (or by Minnesota Fats), but the viewer might still describe the shot as "remarkable." We conjecture that two factors contribute to the second shot's being surprising:

- Can a pool table really support that kind of precision? Along the lines of the discussion of Section 3.4, if the table was marble we might believe that all the angles will work out exactly as needed; but for a felt-covered table, we'd expect the random effects to make it impossible for anyone to deliberately make the shot.

- Can a pool player really hit the ball with that kind of precision? Given the instability of the system, among all possible initial conditions, the measure of those that lead to sinking the target ball at the end is almost vanishingly small.

If we want to eliminate the "Minnesota Fats solutions" from our solver, a general approach would be to prune based on probability, as discussed in Section 3.4: Before descending each branch of the path cone tree, integrate the probability over that cone, and make sure it remains above threshold; and for the candidate solutions, compute the Bayesian probability to make sure that the variabilities along the way are behaving in a manner consistent with our expectations of their probability distributions.

A more specific approach to eliminating surprising solutions for this problem is to simply place an arbitrary limit on the spread of incident angles for the ball collision, to rule out grazing shots. This is a situtation where visual plausbility is stricter than physical plausibility, as discussed in Section 3.2.

5.5 Additional Results

To quantify the effects of introducing variability into the inverse problems of Sections 5.3 and 5.4, we ran them on a series of random trials, and accumulated statistics. The table lists the average number of solutions of a given length, for the "accurate" solution, and for 5 and 10 degrees of normal perturbation:

Figure 7: A collision with random normal perturbation. *Left:* incident at 5.1 degrees, plausibly reflected at 6.84 degrees. *Right:* incident at 5.1 degrees, implausibly reflected at -2.3 degrees.

# bounces:	1	2	3	4	5	6	7
"accurate":	0.04	0.09	0.07	0.05	0.01	0.00	0.00
5 degrees:	0.21	0.52	0.88	1.18	0.43	0.00	0.00
10 degrees:	0.32	1.00	2.14	1.92	0.44	0.00	0.00

ONE BALL

"accurate":	2.16	3.66	2.34	0.70	0.18	0.04	0.00
5 degrees:	2.23	4.68	3.64	2.33	0.63	0.00	0.00
10 degrees:	2.42	5.89	6.21	3.51	0.72	0.00	0.00

TWO BALLS

As expected, for the one ball case, there were rarely any "accurate" solutions, and adding variability increased the number of solutions found. For the two ball case, increasing the variability mostly helps in finding longer solutions. In both cases, the maximum speed and friction of the table ruled out most any solutions with six or more collisions.

Finally, we came across occasional situations in which our small normal perturbation yields visually implausible results, illustrated in Figure 7: Although it was generally fine for collision normals to be perturbed, for a nearly-perpendicular collision, occasionally the ball would fail to cross the centerline of the collision; lacking any spin, this is implausible. This points out the need for care in choosing limits on plausibility. We suspect that some of the ideas of qualitative physics [Bob85] will be of use for this.

6 Conclusion

Computer graphics animation systems do not need to be predictive modelers of the physical world. Instead, they have the quite distinct task of being *plausible mimickers* of the physical world, while at the same time providing flexibility and controllability for animators. We believe that

- *detail* in motion, just like the detail in geometry and in light-reflection models, is an essential chacteristic of effective mimicry,
- that randomness can often provide effective motion detail, and
- that the ability to select from a distribution can help provide effective controllability.

The addition of variable detail to generate plausible motion has complimentary effects: on the one hand, it eliminates the need to accurately model and simulate the detail; on the other hand, it enlarges the space of possible motion paths that one may consider, and, at least for the approaches discussed in this paper, requires that we consider cones of motion paths rather than single paths, thus enlarging the work done in simulation. In trade for this added work, there is a new controllability in the generated paths, making the solving of inverse problems more tractable.

Limitations, Future Work

All of the ideas presented in this paper are preliminary, and the examples are all deliberately simple. For more complex systems, the notions we discussed may need to be drastically reorganized, for example with some sort of "factorization" of complex systems into simpler parts, each of which may be amenable to plausible simulation.

We have also only discussed rigid bodies and passive motion. Will active or volitional motion complicate or break the ideas we've described? Will deformable or fluid body interactions add new kinds of variability that are more difficult to describe or control? We find such questions intriguing, and intend to do further research on the topic.

Acknowledgements

The authors are grateful to Leslie Kaelbling for helping us make sense of our intuitive notions of Bayesian probability. We would like to dedicate this paper to the late Minnesota Fats (Rudolf Wanderone Jr., d. Jan 18, 1996).

References

[Ama84] J. Amanatides. Ray tracing with cones. *Computer Graphics*, 18(3), 1984.

[Bob85] Daniel Bobrow. *Qualitative Reasoning About Physical Systems*. MIT Press, 1985.

[FvDFH90] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 2nd edition, 1990.

[FW80] Alexander L. Fetter and John Dirk Walecka. *Theoretical Mechanics of Particles and Continua*. McGraw-Hill, Inc., 1980.

[HS92] Bernardo A. Huberman and Peter Struss. *Chaos, Qualitative Reasoning, and the Predictability Problem*. MIT Press, 1992.

[HU93] Colin Howson and Peter Urbach. *Scientific Reasoning: The Bayesian Approach*. Open Court Publishing Company, 1993.

[KMS85] S. Kerckhoff, H. Masur, and J. Smillie. A rational billiard flow is uniquely ergodic in almost every direction. *Bulletin of the Americal Mathematical Society*, 13(2):141–142, Oct 1985.

[KR66] C.W. Kilmister and J.E. Reeve. *Rational Mechanics*. American Elsevier Publishing Company, 1966.

[Rei71] Arnold L. Reimann. *Physics: Mechanics and Heat*. Barnes and Noble, 1971.

[TNM95] Diane Tang, J. Thomas Ngo, and Joe Marks. N-body spacetime constraints. *Journal of Visualization and Computer Animation*, 6:143–154, 1995.