An Application of Motion Design and Control for Physically-Based Animation

David Haumann Jakub Wejchert Kavi Arya Bob Bacon
Al Khorasani Alan Norton Paula Sweeney

Computer Sciences Department, IBM T. J. Watson Research Center

Abstract

We describe a variety of physically-based simulation techniques used to generate the motion of objects for computer animated scenes. Using classical mechanics, we model rigid, flexible and breakable objects and their interaction with wind fields. Complex motion can be controlled by specifying initial conditions, designing wind fields, and using an interactive previewing capability to view the results prior to final rendering. Examples are given of how simulation is used to create animated scenes that contain rigid and wind blown flexible objects.

Résumé

Nous présentons un ensemble de techniques de simulation de mouvement d'objets pour la génération de séquences d'animation. Les principes de mécanique classique sont utilisés pour modéliser des objets rigides, flexibles ou cassants et leurs interactions avec des champs de vecteurs qui représentent les effets du vent. Les mouvements peuvent être controlés par la spécification des conditions initiales et des caractéristiques du vent. Une pré-visualisation interactive permet l'analyse des séquences d'animation avant la génération finale des images. Des exemples de spécification de simulation de mouvements d'objets influencés par le vent sont présentés.

Keywords: animation, physically-based modeling, flexible objects, vector fields, simulation, dynamics.

CR Categories: I.3.7–Three dimensional graphics and realism (Animation); I.6.3–Simulation and modeling (Applications).

1 Introduction

1.1 Background

Physically-based computer animation uses physical principles to generate the motion of objects in a simulated world. This approach has advantages over traditional keyframed animation because it procedurally generates realistic motion and can reproduce the motion complexity of a large group of interacting objects. Current research employs physically-based techniques to broaden the range of natural phenomena that can be easily animated while exploring ways to provide the animator with explicit control over the motion produced.

The goal of a good physically-based animation model should be *generality* and *control.* A simulator should handle a wide range of objects and motions, and be able to simulate ordinary objects in a variety of typical everyday situations. The models should give a plausible visible presentation of objects responding to forces in their environment, and provide the user with adequate means of control. Although physics is the basis of such models, our concern with physical accuracy is limited to the point at which the desired visual and phenomenological effects have been achieved. In our approach, empirically defined physical laws may be approximated for motion control or exaggerated for visual effects. Thus, the objective of physically-based animation is to simulate a wide range of phenomena within one environment, and incorporate user controls over the generated motion.

A variety of methods have been used to simulate the motion of rigid bodies, flexible bodies and fluid-like behaviors. Particle systems have been used to simulate fluid-like activity of fire (Reeves 83), ocean foam (Fournier 86) and to model streams, fountains and blowing snow (Sims 90). Miller (Miller 89) uses a "globular" particle based method to simulate fluid flow, and Kass (Kass 90) solves shallow water equations to animate flowing water. The behavior of collections of animate objects such as birds (Reynolds 87, Amkraut 89a), individual motion of worms (Miller 88), and articulated figures (Girard 86, Hahn 88, Wilhelms 87, and Issacs, 87) have all drawn upon physical models to some extent. Hahn (Hahn 88), Barzel (Barzel 88) and Witkin (Witkin 88) have all been concerned with realistic simulation and control of rigid bodies. Platt (Platt 88) and Terzopoulos (Terzopoulos 87,88) have been concerned with modelling and controlling the motion of flexible objects. We have attempted to incorporate simulations of flexibility, fracture (Norton 90,91), and fluid effects into one general purpose animation environment, and provide the techniques necessary to control them.

1.2 Animation Pipeline

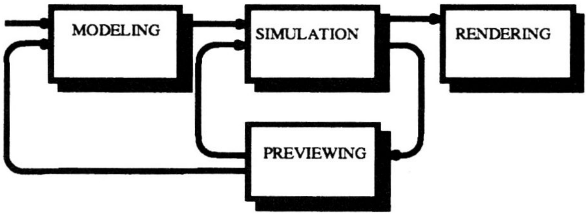


Figure 1: Animation Pipeline

Figure 1 shows the steps we take to generate an animation. The modeling phase assigns geometric and physical properties (such as mass, stiffness and damping) to objects to be animated. The models combined with initial and environmental conditions (such as position, velocity, and wind fields) are input to the simulation. The simulation generates the motion of the objects by integrating the physical equations over time. The previewer is used to obtain fast visual playback from the simulation so that the simulation or models may be modified. Once motion is acceptable, it is rendered and recorded.

As an example of the animation pipeline in practice, consider a scene that consists of a collection of leaves blowing in the wind. A first step is to design the leaf geometry and assign it physical properties (such as weight and stiffness). Then one has to create a set of wind fields, and assign initial positions and velocities to the leaves. A test simulation can be run and the results previewed. Changes to the simulation input (such as the wind field velocity, position, or the leaf properties) can then be made. This cycle is repeated until a satisfactory motion results. Only then is the polygonal description rendered and recorded.

2 Simulation

2.1 Rigid, Flexible and Brittle objects

In our simulations, rigid, flexible and brittle objects and their interactions with wind fields have been modeled using Newtonian mechanics. Our objects are modeled as collections of mass particles in networks arranged into geometrical shapes (such as teapots or cylinders). Rigid object motion is modeled as in (Hahn 88); the net force and net torque on the object are computed by summing the contributions from all mass particles in the object. Flexible materials are modeled using a 3D mesh of interconnected masses and springs. The masses accelerate according to the sum of the forces applied to them and the time evolution of the system is carried out by integrating Newton's second law $\mathbf{F} = m\mathbf{a}$ as follows. At each time step Δt, and for each mass point m_i, the total force $\mathbf{F}_i$ acting on the point is computed. This determines the acceleration $\mathbf{a}_i$ of that mass point and using a first order difference approximation (Euler's Method) the extrapolated velocity $\mathbf{v}_i$ and position $\mathbf{r}_i$ at time $t+\Delta t$ is given as follows:

$$\mathbf{a}_i = \mathbf{F}_i / m_i. \tag{1}$$

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \mathbf{a}_i \Delta t, \tag{2}$$

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t) \Delta t. \tag{3}$$

Once all the positions have been updated the same cycle is repeated for succeeding time steps until the simulation is completed.

Forces on objects can be classified as either internal or external. External forces acting on objects are environmental forces of gravity, friction, or wind. The internal forces in a flexible object are modeled by the stretching of the springs. To simulate the breaking or tearing of objects, a threshold is associated with every spring. A spring breaks if it is stretched beyond a threshold, and on a macroscopic scale the breaking of many bonds causes a fracture or tear.

2.2 Wind Fields

Using wind velocity vector fields, we can simulate the motion of objects in wind. A wind field is a function that maps a position in space to the velocity of the wind at that point. This velocity can be used to determine the wind induced forces on an object at that position.

The use of fields is not new to computer animation; they have been used to model collision avoidance and to control motion. Terzopoulos suggested using potential energy fields to prevent interpenetration of flexible models (Terzopoulos 87). Haumann used a time varying spatially uniform field to model a flag blowing in the wind (Haumann 1988). In the film Eurhythmy (Amkraut 89b), force fields were used to direct the motion of a flock of birds and to simulate collision avoidance behavior between individual birds. Pintado (1989) used nonphysical fields to interactively control the motion of objects. Karl Sims used velocity and acceleration operators to control the motion of particles (Sims 90). Like the particle approach used by Sims we use velocity fields to model objects carried along by the surrounding fluid. However, because our approach computes the forces on *surfaces* in an orientation dependent manner, the geometry and current orientation of the object will affect its subsequent motion.

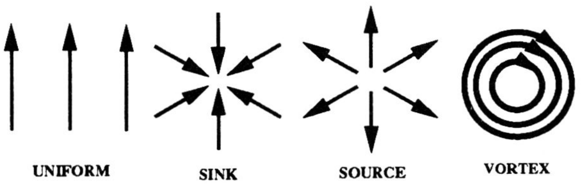
A physically correct wind velocity field must satisfy the Navier-Stokes equations. These equations relate the time evolution of the velocity field to the pressure, density, and viscosity of the fluid. Given the complexity of solving these non-linear equations we make several simplifying assumptions. First, we assume that the wind fields are not affected by the objects that are placed in them. Thus, our simulations will not exhibit wind "shadowing" effects where an upwind object shields objects downwind. Secondly, we restrict the fluid flow to be non-turbulent. Following along the lines suggested by Feynman (1965) we assume that: a) the fluid is inviscid (viscosity is zero), b) incompressible (the density is constant), and c) the flow is irrotational (infinitesimal fluid volume elements do not rotate, although the fluid as a whole may circulate). Imposing these conditions reduces the problem to finding a solution to Laplace's equation which is stated as follows:

$$\nabla^2 \phi = 0 \tag{4}$$

If ϕ is a scalar field that satisfies Laplaces' equation, then the gradient of ϕ is a vector field ($\mathbf{v}$)

$$\mathbf{v} = \nabla \phi \tag{5}$$

which is a solution of the Navier-Stokes equations (with the above conditions imposed). Because the Laplacian is a *linear* second order differential equation, the linear combination of any two analytic solutions is also a solution. Thus, we can define a set of wind velocity field primitives which are simple analytic solutions of our "restricted" Navier-Stokes equations, and then superimpose them to obtain more complicated velocity fields. We use vortices, sinks, sources, and uniform flows as our primitive wind fields (figure 2). Sources are points from which fluid moves out in all directions. Sinks are points toward which fluid moves uniformly from all directions and then disappears. A vortex is a field in which the fluid moves in concentric circles about the center. By linear addition of these



Uniform (Cartesian coords): $\phi = ax + by + cz$;
$\mathbf{v}_x = a$; $\mathbf{v}_y = b$; $\mathbf{v}_z = c$

Source (Spherical coords): $\phi = a\ln(r)/2\pi$;
$\mathbf{v}_r = a/2\pi r$; $\mathbf{v}_\psi = 0$; $\mathbf{v}_\theta = 0$

Sink (Spherical coords): $\phi = -a\ln(r)/2\pi$;
$\mathbf{v}_r = -a/2\pi r$; $\mathbf{v}_\psi = 0$; $\mathbf{v}_\theta = 0$

Vortex (Cylindrical coords): $\phi = a\theta/2\pi$;
$\mathbf{v}_r = 0$; $\mathbf{v}_\theta = a/2\pi r$; $\mathbf{v}_z = 0$

Figure 2: Field primitives with their potential and velocity vector component equations (where a, b, c are constants).

basic types, complicated flows can be easily constructed and used to affect the motion of objects in an animated scene. For example, it is possible to construct fields of limited spatial extent. If we place a large number of sinks on plane A and on a parallel plane B (all of equal strength), then to a first approximation we have a field that is zero between A and B and non-zero everywhere else. Such "bounded" fields were useful for confining object motion to a limited volume.

2.3 Field-Object Interaction

Consider a particle interacting with a velocity field representing the surrounding fluid. The magnitude of the force on the particle is related to the difference between the particle velocity and the field velocity. Given a velocity field $\mathbf{G}(x, y, z)$, the relative velocity of the mass particle with respect to the field velocity is:

$$\mathbf{v}_i^r = \mathbf{G}_i - \mathbf{v}_i \qquad (6)$$

where $\mathbf{G}_i$ refers to the field at the x, y, z position of the particle, and $\mathbf{v}_i$ is the particle velocity. The force of the wind acting on that particle is then:

$$\mathbf{F}_i = \alpha \mathbf{v}_i^r, \qquad (7)$$

where α is a chosen constant relating force to relative velocity.

Representing an object as a single mass particle will not result in any rotational effects due to the wind. To achieve this effect, a surface model is used. Wind forces acting on a surface depend on the surface area and the orientation of a triangular surface with respect to the relative velocity (see figure 3). Given a mass particle whose position defines one corner of a triangular surface of area A, we resolve the relative velocity of the particle into the normal and tangential components with respect to the surface. Thus:

$$\mathbf{v}_i^r = \mathbf{v}_i^n + \mathbf{v}_i^t \qquad (8)$$

where $\mathbf{v}_i^n$ is the normal component and $\mathbf{v}_i^t$ is the tangential component. We use these to compute normal and tangential forces as follows:

$$\mathbf{F}_i^n = \alpha^n A \mathbf{v}_i^n \qquad (9)$$

$$\mathbf{F}_i^t = \alpha^t A \mathbf{v}_i^t \qquad (10)$$

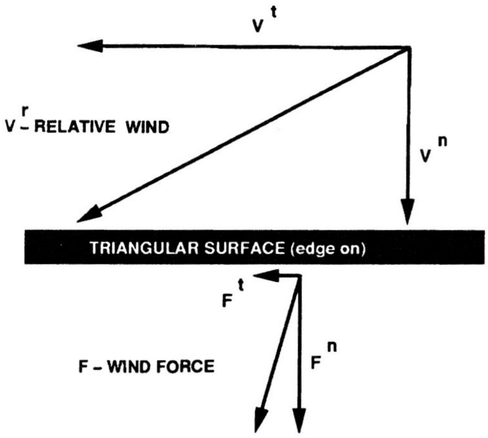


Figure 3: The relative wind is resolved into components normal and tangential to the surface. Assigning different weights (α's) gives rise to orientation dependent wind forces.

$\mathbf{F}^n$ is the force experienced by a surface facing into the wind, while $\mathbf{F}^t$ is due to the viscous effects of fluid flowing across the surface; A is the area of the triangular element. Normally, α^n is chosen to be much larger than α^t because surfaces facing into the wind experience much larger forces than surfaces parallel to the wind. Note that a particle in a triangulated mesh usually forms a corner of several adjacent triangles, and hence will receive force contributions from each.

In summary, our simulation consists of masses, springs, and wind fields, all obeying classical mechanics. These basic building blocks can be assembled together to create a sophisticated animation environment.

3 Preview

Interactive previewing of simulation output is an indispensable capability in the animation pipeline. Short circuiting the rendering and recording steps saves time. As soon as the simulation data is available, decisions to change the simulation (or modeling) input can be made. This allows for a greater level of interactivity between the user and the simulation output. The previewing program allows the motion produced by the simulator to be animated at acceptable playback speeds by storing polygonal data in memory and utilizing workstation graphics hardware to render the frames. Thus, the viewpoint can be interactively changed as the animation is playing. A friendly interface performs all translations and rotations in user centered coordinates, allowing the user to easily pilot about and explore the entire space of simulation data.

4 Examples of Motion Design and Control

The wind field capability inspired us to create an animation of leaves blowing in the wind. Action to support our simple plot included playground swings and a trash bin that chases the leaves. Both the swings and the leaves were flexible objects controlled by wind fields, while the bin was a rigid body whose motion was partly keyframed and partly simulated.

4.1 Designing the Geometry

Our first task was to construct a physical model of a leaf. We did not set out to model a particular kind of leaf, rather, our goal was simply to create an adequate illusion of leaf motion. We chose a model that was as simple as possible (to limit simulation time) yet which retained enough geometrical complexity to produce interesting motion when affected by the wind fields. Figure 4 shows two prototypical leaf topologies consisting of six triangular surfaces with a mass at each vertex. With these prototype leaves as input, we used the simulator experimentally to refine them into leaves that exhibited realistic yet interesting and controllable motion. Each leaf prototype was duplicated on the order of a hundred times with small random variations in geometry, and physical properties such as mass distribution and stiffness. Variations in geometry were achieved by "folding" the (otherwise flat) leaf prototype along the shared edge between two triangles. Tests were performed on this varied collection by dropping it in still air (zero velocity uniform field). The previewer was used to examine the results and to identify those leaves exhibiting desirable motion characteristics. This process was repeated by applying the duplication/variation step to the newly selected leaves, but with a much smaller magnitude of variation. It was from this process of "manual selection" that the final collection of leaves "evolved".

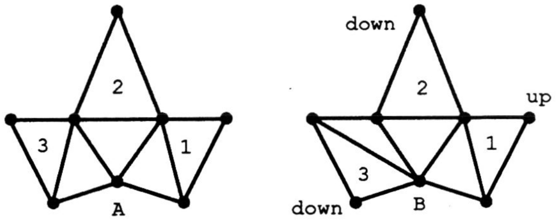


Figure 4: Two possible leaf topologies consisting six triangular surfaces with a mass at each vertex (springs not shown). Topology B was used in the final animation; the asymmetry probably contributing to its interesting lilting motion. Directions indicate how the outer triangles were folded (relative to the page).

It is interesting to note the variety of motion that we encountered using only the two prototypes shown. For example, leaf motion could be roughly classified into three types: leaves that did not rotate but seemed to float or glide gently down (we called them "floaters") ; leaves that rotated rapidly as they fell ("spinners"), and leaves that appeared to alternately rock, then rotate end over end ("lilters"). These properties were a result of how the outer triangles (labeled 1, 2, and 3 in figure 4) were folded and where the leaf center of gravity was placed. For example, leaves that had all three triangles (1, 2, and 3 on topology A) folded up (from the page) and a center of gravity near the leaf center were usually "floaters". "Spinners" tended to have their center of gravity off to one side and have triangles 1 and 3 (again, topology A) folded in opposite directions. The final animation used a descendent of topology B that had triangle 1 folded up while 2 and 3 were folded down. This gave rise to its interesting lilting motion clearly evident in the opening scene of our film "Leaf Magic" (Norton 90).

A set of playground swings and a trash bin were also modeled. Each flexible rope of the swing was modeled from a 3D mass-spring mesh formed in the shape of a long cylinder. The seat was a rectangular parallelepiped. The ropes were "glued" to the seat by attaching springs from seat masses to rope masses in the vicinity of their junction. The bin was modeled as a simple cylinder.

4.2 Controlling the Motion.

The animated story for this project was concerned with leaves being chased about a playground by a yard bin. In one scene it was required to have the leaves rise up in surprise from a playful hovering configuration, and then fall "dead" to the ground. This sequence is depicted in figures 6-8. The hovering configuration (figure 6) was achieved by using a uniform field acting vertically to exactly cancel the normal rate of fall of the leaves under gravity. This field was linearly combined with a weak set of bounded fields used to keep the leaves hovering within a desired volume.

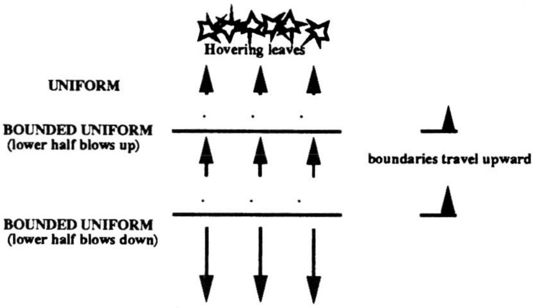


Figure 5: Field Combination Used in Surprise/Fear Scene

The surprise/fear sequence was designed to appear as a "shockwave" moving rapidly upward through the hovering leaves. This effect was achieved by successively moving the boundaries of two uniform fields up through the collection of leaves (see figure 5). The first field had a strong upward component causing the leaves to rise momentarily and to collect near its upper boundary. The second field contained a strong downward component designed to pull the leaves rapidly down to the ground. Figure 7 shows the collecting effect caused by the first field's boundary as well as a few leaves that have begun to fall as the result of the second. The combination of these fields causes successive layers of leaves to first rear up, then hurtle towards the ground. Once the leaves have all landed on the ground, a third uniform horizontal field causes the leaves to be blown towards the viewer. Figure 8 shows these leaves rolling along the ground (to escape the pursuit of the yard bin in the background).

As mentioned, it is through the linear combination of fields that complex motion paths can be set up for multiple objects. In the following example the addition of a vortex, sink and uniform flow were used to create a scene with a garbage bin inhaling some freely floating leaves. Figures 9, 10, and 11 show snapshots from this sequence. In the first shot (figure 9) the leaves have just begun to be inhaled by the garbage bin. For dramatic effect they were made to swirl around before being funneled in and down (figure 10). At the end of the scene the vortex is so strong that the main component of motion is downwards into the open mouth of the can (figure 11). The previewer was an essential ingredient in setting up the above motion, because the adjustment of the field strengths affects the overall movement of the leaves.

In several scenes the playground swings sway in the wind, further adding to the animated illusion. The swings were suspended in the environment by fixing the masses at the top of the ropes. A uniform field whose strength varied over time was applied to make them sway.

Rigid body simulation was not implemented in the simulator until the production was nearly complete, hence, most of the trash bin motion had to be keyframed. However, at the end of the chase sequence, the storyboard called for the bin to skid to a stop at the side of the garden shed. The end of the stop is punctuated by the bin rocking forward in response to the sudden stop. This rocking was produced by the rigid body simulation. The initial conditions (velocity and position) were matched with the the last frame of the keyframed motion; the forward momentum and ground friction caused the bin to rock out of balance.

5 Conclusions and Future Directions

Physically-based models can be a useful animation tool if they are provided in a general framework with adequate control mechanisms. The simulation environment presented here is general enough for rigid, flexible and brittle objects subjected to normal environmental affects including winds. The motion of wind affected objects can be controlled using fields. This coupled with a simulation preview facility enabled us to design the motion of hundreds of leaves blowing in the wind for scenes in the animation "Leaf Magic" (Norton 90).

In the future, we envision building higher levels of control into the simulator. Using techniques taken from control theory and robotics we intend to "sense" the current simulation state. This feedback, coupled with heuristic knowledge about the physical model at hand, can be used to generate environmental forces to direct the motion of the objects. In this way complex mechanisms can be controlled from higher levels of abstraction.

6 Acknowledgements

We wish to thank all the members of the Animation Systems Group, especially Jane Jung, without whom this work would not have been possible. Tim Kay, Greg Turk, John Snyder, John Hart and Mike Henderson provided invaluable software support and consulting. Thanks are also due to the reviewers for helpful comments and suggestions.

7 References

Amkraut S, (1989a) "Flock: A Behavioral Model for Computer Animation", Masters Thesis, Art Education, The Ohio State University.

Amkraut S, Girard, M, (1989b) "EURHYTHMY" Siggraph Video Review, Issue 52 (SIGGRAPH '89 Film and Video Show), selection 8.

Barzel R, Barr A, (1988) "A Modeling System based on Dynamic Constraints", Computer Graphics (SIGGRAPH 88 Proceedings) 22 (4) 179.

Feynman R, Leighton R, Sands M (1965) "The Feynman Lectures on Physics", Addison Wesley Publishers.

Fournier A, Reeves W, (1986) "A Simple Model of Ocean Waves", Computer Graphics (SIGGRAPH '86 Proceedings) 20 (4) 75.

Girard M, Maciejewski A, (1986) "Computational Modeling for the Computer Animation of Legged Figures", Computer Graphics (SIGGRAPH 86 Proceedings) 19 (3) 263.

Hahn J., (1988) "Realistic Animation of Rigid Bodies", Computer Graphics (SIGGRAPH 88 Proceedings) 22 (4) 299

Haumann D, Parent R, (1988) "The Behavioral Test-Bed: Obtaining Complex Behavior from Simple Rules" The Visual Computer 4 (6) 332.

Issacs P, Cohen M, (1987) "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics", Computer Graphics (SIGGRAPH 87 Proceedings) 21 (4) 215.

Kass M, Miller G, (1990) "Rapid, Stable Fluid Dynamics for Computer Graphics" Computer Graphics (SIGGRAPH 90 Proceedings) 24 (4) 49.

Miller G, (1988) "The Motion Dynamics of Snakes and Worms" Computer Graphics (SIGGRAPH 88 Proceedings) 22 (4) 169.

Miller G, Pearce A, (1989) "Globular Dynamics: A Connected Particle System for Animating Viscous Fluids", Siggraph '89 Course Notes, Topics in Physically-Based Modeling.

Norton A, et. al., (1990) "Leaf Magic", Computer Generated Film, IBM Research, Yorktown, N.Y.

Norton A, Turk G, Bacon R, (1991) "Animation and Fracture by Physical Modeling", The Visual Computer (to appear). See also RC 15371 (#68412) 1/11/90 (IBM Computer Science Research Report).

Pintado X, Fiume E, (1989) "Grafields: Field-Directed Dynamic Splines for Interactive Motion Control", Computers and Graphics 13 (1) 77.

Platt J, Barr A, (1988) "Constraint Methods for Flexible Models", Computer Graphics (SIGGRAPH 88 Proceedings) 22 (4) 279.

Reeves W, (1983) "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects" Computer Graphics (SIGGRAPH 83 Proceedings) 17 (3) 359.

Reynolds C, (1987) "Flocks, Herds, and Schools: A Distributed Behavioral Model", Computer Graphics (SIGGRAPH 87 Proceedings) 21 (4) 25.

Sims K, (1990) "Particle Animation and Rendering Using Data Parallel Computation", Computer Graphics (SIGGRAPH '90 Proceedings) 24 (4) 405.

Terzopoulos D, Platt, J, Barr, A, Fleischer, K, (1987) "Elastically Deformable Models" Computer Graphics (SIGGRAPH '87 Proceedings) 21 (4) 205.

Terzopoulos D, Fleischer K, (1988) "Deformable Models" The Visual Computer (1988) 4 306.

Witkin A, Kass M, (1988) "Spacetime Constraints", Computer Graphics (SIGGRAPH 88 Proceedings) 22 (4) 159.

Wilhelms J, (1987) "Using Dynamic Analysis for Realistic Animation of Articulated Bodies", IEEE Computer Graphics and Applications, 7 (6) 12.

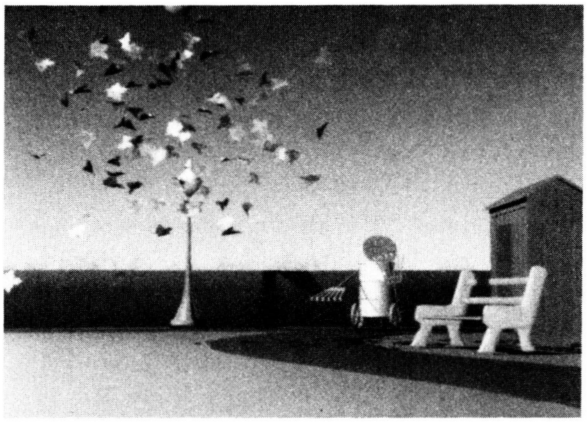


Figure 6: An upward vertical field keeps leaves hovering; "lilting" motion is due to their shape.

Figure 7: Successive uniform bounded fields moving up through the leaves gives the appearance of a "shock wave" moving up through the leaves, spreading them out as they descend.

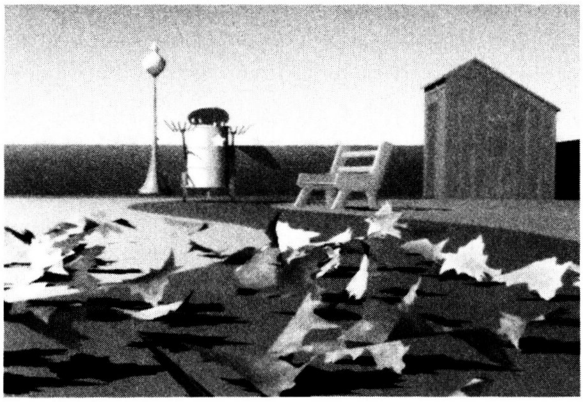


Figure 8: A horizontal field tumbles leaves along the ground (towards viewer) to escape the pursuing garbage bin (background).

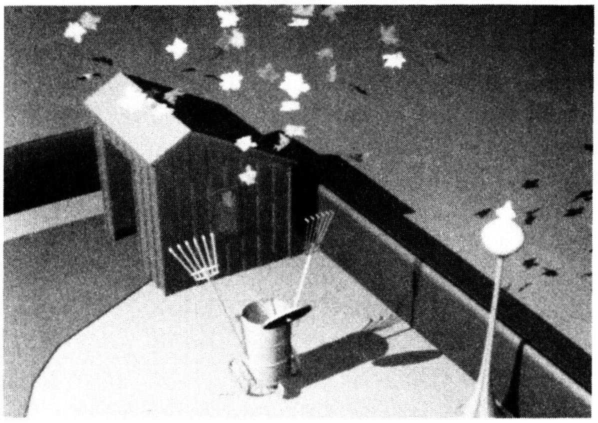


Figure 9: The addition of vortex, sink and uniform flows simulates leaves being inhaled by the bin.

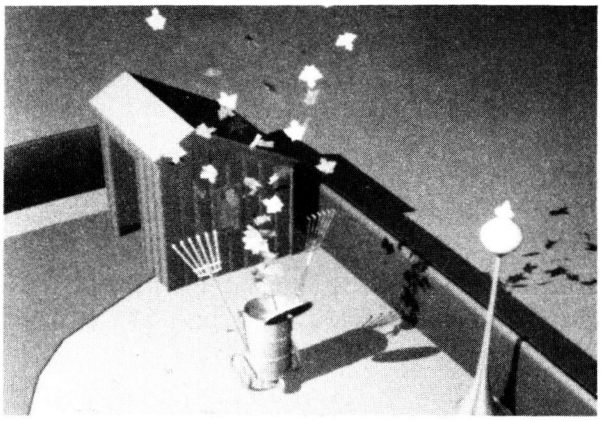


Figure 10: Inhalation proceeds; the swirling funnel shape caused by the vortex can be clearly seen.

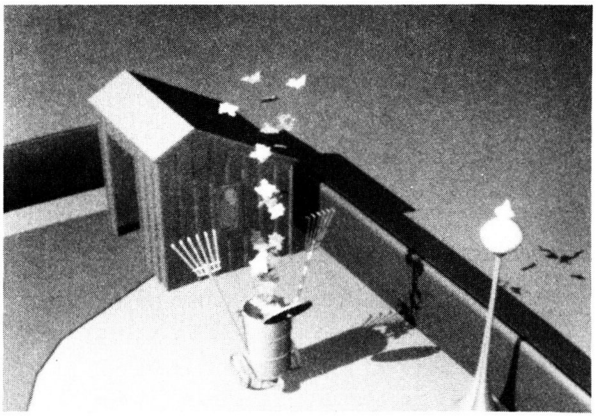


Figure 11: Inhalation finale; the leaves converge into the bin.