# Particle Systems for Artistic Expression

David Tonnesen

FOAM - Starlab
Rue Englandstraat 555
B-1180 Brussels, Belgium
dave@starlab.net

## Abstract

Particle systems can be thought of as a general technique within the field of computer graphics for creating a wide range of effects. To illustrate the range of effects, this paper begins by quickly reviewing the existing research on particle systems in the field of computer graphics. It then discuss in more details two particular areas of particle systems research : first a technique for sculpting surfaces, and second a work in progress for an interactive art installation. A common goal of both projects is to provide flexible tools to aid in personal expression. The focus is to move away from the analytical and point and click style of interface, and towards a more humanistic interface which re-embodies the user in the physical world.

## Particle systems

A particle system is a collection of particles. The system state evolves over time and can exhibit complex dynamic behavior. In the simplest case, each particle is a point mass that reacts to applied forces according to the laws of Newtonian mechanics. Assuming a 2nd order dynamic system, each particle will exhibit the additional properties of velocity, momentum, and acceleration. These are but a few of the possible properties one could use.

In computer graphics, particle systems have been used to model visually complex natural phenomena, to emulate group and flocking behaviors, to model and reconstruct both surfaces and volumes, and to simulate the physics of deformable, elastic, viscous, and solid materials. To aid in the understanding of the different types of systems, we can categorize particle systems according to the interactions between particles. In *systems of independent particles*, the forces acting on each particle are independent of other particles in the system. In *systems with fixed connection*, particles interact with neighboring particles where the set of interactions is constant after the initial specification. In systems with *dynamically coupled particle interactions*, the interactions between particles evolve over time. The couplings are typically defined based on a distance metric.

## Previous work

Systems of independent particles have been used to model visually complex natural phenomena such as fire, smoke, foliage, and the spray of splashing water [9,5,10,15,16]. In these systems, the forces on each particle are independent of the other particles in the system. To create complex behavior, these techniques use large numbers of particles reacting to forces such as gravity, obstacles, wind fields, and turbulence. Particles are created and deleted from the system using rules based on the phenomena being modeled. Most of these approaches concentrate on creating a particular visual effect and make no attempt to define either an object volume or the corresponding surface.

The physically based deformable models [1,6,18] can be categorized as particle systems with fixed particle interactions. Typically these models can be thought of as a discretization of a volume or surface, which attempts to make a realistic model of the deformation of a given physical material. Through the use of plastic, elastic, and viscous coupling units between particles, these dynamic models can exhibit plasticity, elasticity, bending, and fracture. Shapes modeled with fixed couplings can be deformed, but ultimately are limited to the structure imposed by the given couplings.

In dynamically coupled particle systems, the interactions between particles evolve over time. That is, couplings between particles are automatically removed and new couplings are automatically created. Replacing the fixed set of interactions with couplings that dynamically evolve creates a flexible modeling paradigm in which can exhibit physical properties similar to those with fixed couplings, but with the advantage that geometric and topological changes can occur as the underlying structure of the system changes. Physically inspired models of deformable volumes [3,19], and fluids [8,12] use these advantages to allow for gross changes in geometry. In addition, dynamically coupled particle systems have been used to re-mesh polygonal models [21], to triangulate implicit surfaces [2,22], to reconstruct surfaces of arbitrary topology [17] from 3D data sets, to interactively model free-form surfaces [20], to distribute paint strokes for a painterly effect in rendering [7], and to grow cellular based textures [4]. Perhaps the most commonly

known application of dynamically coupled particle systems is in flocking algorithms [11]. Such algorithms are used extensively in the film industry to simulate flocks of birds, herds of animals, and crowds of people.

## Sculpting shapes

### Motivation

This section focuses on a particle system based tool for sculpting geometric shapes in an intuitive and natural manner. One can think of two basic approaches to the modeling of shapes. The first is to create a shape to meet a known set of specifications. Another approach, more exploratory in nature, is to evolve a shape from simple to complex, until it is aesthetically pleasing. In this case the focus is not on measurable analytic qualities but on subjective esthetic qualities. Spline based interpolation, extrusion, and constructive solid geometry, enforce analytical constraints on a shape, aiding in the objective design approach. The success of analytically based tools are in part because the constraints of each technique assist in defining a particular class of surfaces. However, if used to create shapes outside of the intended domain, the same constraints may hinder the creation process by imposing unnecessary limitations. For the exploratory approach, the design tools must be flexible in the sense that the constraints on the design process should be minimized.

### Sculpting metaphor

We created an interactive system [20] as a tool for developing shapes using an exploratory approach. The system is based on a sculpting metaphor to alleviate the user's need to think about the underlying representation or be limited by its choice [13]. Borrowing from traditional sculpting materials, we created synthetic shaping materials that can be manipulated in a variety of ways. The materials provide the user the ability to cut, merge, and join shapes; to deform shapes; and to shape and reshape the material with a basic set of tools. By including physics into the model, we create synthetic materials with one or more of the following properties: elasticity, stiffness, and fluidity. We can choose to conserve surface area or make infinitely elastic like materials. We can heat shapes make areas more malleable, similar to heating glass to shape it. We can minimize surface curvature through bending energies, similar to the smooth curve created by bending a thin metal spline.

### Self organization

We developed a new shape representation model, based not on an underlying parametric representation, but rather on systems of self-organizing elements from which shape and structure emerge. In our sculpting system we use a dynamically coupled particle system, in which an orientation has been assigned to each particle. Thus each particle corresponds to a small surface sample (a ``surfel'') with both a surface normal direction and tangent plane. To encourage the particles to arrange into surfaces, we designed new particle interaction functions based on geometric and physical measures. The global structure of the shape evolves from the combination of direct incremental user manipulation via shaping tools and localized particle interactions.

### Implementation

We now briefly describe the computational process embedded in our sculpting system. At each step in the simulation there is a possibility of the user moving the sculpting tool. We compute the new position of the tool and test if any particles are affected by the tool. If so, we update those particles directly, or compute the appropriate forces, depending on the type of tool being used. For example, a displacement tool would geometrically displace the particle positions, while a heating tool might added heat to particles. Since the particles have the possibility of moving at each step, we must next compute the new particle couplings. A particle is coupled to each of its nearest neighbors. For a given particle, its nearest neighbors are defined as all the particles that lie within a fixed distance of the given particle. We use a spatial subdivision algorithm [14] to compute this efficiently.

We can now compute the physical forces on each particle due to particle interactions. The particle interaction functions are defined over all couplings. These functions result in forces being applied to the particles. After accumulating all of the forces, we then numerically integrate the forces over some small time interval according to Newtonian mechanics. The integration step results in both linear and angular accelerations, and thus changes in linear velocity, angular velocity (speed of rotation), position, and orientation of each particle. Next, we compute any heuristic functions, such as adding or deleting particles. If a continuous surface description is desired, we then can computing a triangulation with respect to the particle positions, using a modified version of the Delaunay triangulation. Finally, the scene is then rendered and the process repeated.

## Interactive performances

This section focuses on a current work-in-progress, that of a particle system being designed for an interactive art installation. An explicit goal of this project is to provide a new way for people to be able to easily express themselves through computer based technology. In this installation, people participate together through physical movement to interactively create both an auditory and visual environment. Each person's movement will generate streams of raw data, which are then interpreted by a computer process. This interpretation provides a mapping between the raw data and commands to a particle system. A dynamically coupled particle system is then used to generate the visual environment. The physical behavior of the particles will vary depending on each performers movements and the mappings active at any given time. Each particle will be responding to an individual performer movements, to group movements, or some combination of the two. Note, we will not be discussing the mapping process (interpretation) nor the generation of audio in this paper.

## Design objectives

We now discuss some of the design objectives of our system.

**Visuals corresponding to an individual performer:** We will be assigning to each performer a unique set of particles. Each set of particles will in turn respond, in some fashion, to the corresponding performer's movements. Particle behaviors and rendering will be applied uniformly to all particles within each unique set. Thus there are three ways one will be able to visually distinguish between particle sets: (1) by variations in rendering (e.g by changing the hue), (2) by variation in active behaviors, (e.g. one set has flocking behavior and another set does not), and (3) by results of the the performer's movements on the particle system.

**Visuals corresponding to group movement:** Because the particle system operates on sets of particles, we can create a new particle set; the group set, which is a union of the individual particle sets. Any operation that can be applied to a individual's particle set can also be applied to the group set. Thus we can use techniques similar to differentiating individual performers to differentiate an individual performer's visuals from the group based visuals.

**Interaction between individual and group visuals:** Multiple particle behaviors can usually be added together to generate more complex behaviors, and likewise a complex behavior can be broken down into independent components. As a very simple example, let's suppose that a individual's particle set behavior is flocking around the corresponding performer's current X-Y position and that the group behavior varies the speed of flocking. By separating a complex behavior (such as flocking) into orthogonal components (in this case speed and flocking) we can assign the different components into individual and group movements. As another example, we could say that individual behavior results in flocking about the person and that group behavior results in a wind field. Each flock of particles can be pushed by the wind (group behavior), while still acting as a flock (individual behavior).

**Smooth transitions between visuals:** A general solution to the transition problem is to define each behavior in terms of a scaling factor, say from zero to one. Given some behavior, at a scaling of zero there would be no noticeable effect and at a value of one there would be the maximum noticeable effect. An important constraint to smooth transitions is that the effect of the scaling factor, must be smooth and continuous for all values between zero and one. For example, we could slowly ramp up the strength of a wind field to a maximum value, and then after some time, ramp it back down to zero.

## Real-time performance

To guarantee real-time performance we need to use efficient algorithms for the potentially expensive operations: computing particle interactions, computing the new neighbors, and numerical integration. To reduce the cost of finding the new neighbors at each step, we will use spatial subdivision search techniques [14]. We also need to choose the behaviors and physical forces to be complex enough to be interesting, but simple enough so that they can be computed quickly. For example, when computing ``flocking'' behavior, each particle looks at it's nearest neighbors to determine where it should go. We may need to look at only the four nearest neighbors instead of twenty. The more neighbors a particle interacts with, the longer it will take to compute the interactions. When the interactions are based on physics, we need to be careful to avoid ``stiff'' equations. Stiff equations are related to large oscillating forces as are commonly found in stiff physical materials. Numerical integration costs can also be computationally expensive. Because the structure of a dynamically coupled particle system is continually changing, they do not lend themselves well to implicit integration schemes, and explicit schemes are preferred.

## Building blocks

We will develop the particle system software in a modular approach, so that the system remains flexible and we can extend the system in the future. Some of the concepts that we will be building into the system include: particle generators, heuristic based particle behaviors, physics based particle behaviors, neighboring finding techniques, and

numerical integration techniques. An example of a particle generator, is an object that say creates particle behind the performer, much like a shadow following oneself, or as if the performer were generating ``calligraphic" strokes. Heuristic based particle behaviors include follow, flocking, hunting (particle of set A "hunts" for particle of set B), eating (A eats B), life, death, and spawning. Physically based particle behaviors include the physical forces of gravity, wind fields (e.g. vortices), viscous forces, attraction, repulsion, and noise.

## References


[1] D. E. Breen, D. H. House, and M. J. Wozny. Predicting the drape of woven cloth using interacting particles. *Computer Graphics (Proceedings SIGGRAPH'94)*, 28(3):365-372, July 1994.

[2] P. Crossno and E. Angel. Isosurface extraction using particle systems**.** In *Proceedings of the conference on Visualization '97*.

[3] M. Desbrun and M. Gascuel. Animating soft substances with implicit surfaces. *Computer Graphics (Proceedings SIGGRAPH'95)*, pages 287-290, August 1995.

[4] K. Fleischer, D. Laidlaw, D. Currin, and A. Barr. Cellular texture generation. *Computer Graphics (Proceedings SIGGRAPH'95)*, pages 239-248, August 1995.

[5] M. E. Goss. A real time particle system for display of ship wakes. *IEEE Computer Graphics and Applications*, 10(3):30-35, May 1990.

[6] D. Haumann, J. Wejchert, K. Arya, B. Bacon, A. Khorasani, A. Norton, and P. Sweeney. An application of motion design and control for physically-based animation. In *Proceedings of Graphics Interface '91*, pages 279-286, June 1991.

[7] B. Meier. Painterly rendering for animation. *Computer Graphics (SIGGRAPH'96)*, pages 477-484, August 1996.

[8] G. Miller and A. Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *In SIGGRAPH '89, Course 30 notes: Topics in Physically-based Modeling*, pages R1-R23. SIGGRAPH, August 1989.

[9] J. F. O'Brien and J. K. Hodgins. Dynamic simulation of splashing fluids. Technical Report GIT-GVU-94-32, Georgia Institute of Technology, 1994.

[10] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics*, 19(3):313-322, July 1985.

[11] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH'87)*, 21(4):25-34, July 1987.

[12] T. M. Roy. Physically based fluid model using smoothed particle hydrodynamics. Master's thesis, University of Illinois at Chicago, 1995.

[13] E. Sachs, A. Roberts, and D. Stoops. 3-Draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, 11(6):18-26, November 1991.

[14] H. Samet. The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading, Massachusetts, 1989.

[15] K. Sims. Particle animation and rendering using data parallel computation. *Computer Graphics (SIGGRAPH'90)*, 24(4):405-413, August 1990.

[16] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. *Computer Graphics (Proceedings of SIGGRAPH'93)*, pages 369-376, August 1993.

[17] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Curvature and continuity control in particle-based surface models. *In SPIE Vol. 2031 Geometric Methods in Computer Vision II*, pages 172-181, San Diego, July 1993. Society of Photo-Optical Instrumentation Engineers.

[18] D. Terzopoulos and K. Fleischer. Modeling inelastic deformations: Visoelasticity, plasticity, fracture. *Computer Graphics (SIGGRAPH'88)*, 22(4):269-278, August 1988.

[19] D. Tonnesen. Modeling liquids and solids using thermal particles. In *Proceedings Graphics Interface*, June 1991, pages 255-262.

[20] D. Tonnesen. Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction, and Animation. PhD thesis, Dept. of Computer Science, University of Toronto, Toronto, Canada, 1998.

[21] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics (Proceedings SIGGRAPH'92)*, 26(2):55-64, July 1992.

[22] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. *Computer Graphics (Proceedings SIGGRAPH'94)*, 28(3):247-256, July 1994.