Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002)
D. Ebert, P. Brunet, I. Navazo (Editors)

Bag-of-Particles as a Deformable Model

D. Stahl, N. Ezquerra and G. Turk

College of Computing, Georgia Institute of Technology, Atlanta, GA USA

Abstract

We present an interactive, physically-based, elastically deformable model using a particle system to model surfaces with interior volumes that can be haptically felt. Oriented particles used in existing surface-only models, and unoriented particles used in volume-only simulations are combined to form a bag-of-particles. Multiple species of surface and volume particles, coupled with prede£ned interspecies parameters, determine the elastic properties of a bag. Starting with an object represented as a 3D voxel bitmap of connected components, the gradient of its distance map gives a vector £eld, or gradient map, that captures the static shape of an object and provides shape-maintaining forces. The gradient map enables the user to de£ne the geometry of the simulated objects, and provides feedback reaction forces allowing a user to feel the model. A bag-of-particles model can simulate several objects in the same scene, as well as objects composed of different materials, such as organs with multiple tissue types. We demonstrate the bag-of-particles approach using a number of different data sources, and apply it to modeling myocardium dynamics.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling: *Curve, surface, solid, and object representations, Physically based modeling*

1. Introduction

For computer models of physical objects to seem "real", both the visual and behavioral aspects of the model must be addressed: visual realism addressing how the model looks, behavioral realism addressing how it physically acts. In addition, if the model is to be manipulated, interactions provided by the haptic model should feel as realistic as possible. Furthermore, in the ideal approach, all three aspects would be integrated in a single, uni£ed model, and support real-time interaction. The computer graphics community has historically focused on these three areas separately and to varying degrees, with extensive research into rendering techniques, but scant research into haptic modeling. Over the last two decades, increasing attention has been directed toward physically-based deformable models, with several different fundamental approaches emerging, each with its strengths and weaknesses.

In particular, physically-based models using particles have employed these particles in a number of specialized ways. Particles have been used to create models of deformable surface shapes, but these surfaces are devoid of interior material. Alternatively, models representing amorphous, bulk interior matter have been developed, but in these models various properties of the particle ensemble are of interest, and surface shape is ignored. Particle system models are limited in other aspects, as well. Some models require particles to be £xed to polygon vertices, thus assuming a polygonal representation exists. Furthermore, particle systems in general have dealt with homogenous material only, and they have not addressed the issue of haptic modeling.

In this context, a modeling approach has been developed that speci£cally addresses several of the limitations in existing particle system modeling techniques. In the *bag-of-particles* approach, certain useful aspects of different, previous approaches are combined in a novel framework, resulting in a physically-based, elastically deformable model that uses particles to model both surfaces *and* their inner volumes. The framework permits objects with non-homogenous material to be modeled, and allows them to be haptically felt, and to undergo small elastic deformations, using direct manipulation. The approach uses a representation for shape that serves to integrate behavioral and interaction models, while allowing models to be constructed from several fundamentally different types of source data. The approach is applied to a modeling a human heart ventricle through a complete cardiac cycle.

2. Previous Work

The various approaches to physically-based deformable modeling can be grouped into four categories: *mass-spring* systems, *£nite-element* methods, *approximate continuum models*, and *particle system* methods. A brief review of these approaches follows.

A mass-spring model idealizes the physical composition and dynamic behavior of an object by treating it as a lattice of mass points connected by springs. Mass-spring systems have been successfully used, for example, in animation and soft tissue modeling [26],[13], and a surface can easily be visualized from a triangulation of the explicit mesh of mass points. However, this representation has several limitations. A small integration time-step is required for numerical stability when modeling hard objects, resulting in slow simulations. To deform a model an added computational cost must be incurred for collision-detection between the mesh and a deforming tool. Furthermore, desired material properties can only be approximated by tuning spring constants. More signi£cant, the basic mass-spring model represents linear dynamics only, precluding some desirable types of deformations such as those characterized by high degrees of ¤exibility, elasticity, or other non-linear behaviors.

The £nite-element method (FEM), on the other hand, is an accurate continuum model that considers properties of mass and energy distributed throughout a body, governed by continuous differential equations [11]. The FEM has been the subject of much interest over the past several years within the graphics domain. For instance, FEM has been used for shape £tting and recovery [6], to model the propagation of brittle fracture [18], for tissue modeling[28], and in surgery simulations[2]. The FEM approach more accurately models continuum behavior, but it is computationally more complex, and consequently is not interactive.

Some models use simpler mathematical formulations designed to produce the desired behavior at more interactive speeds [3],[12],[20],[24]. Though faster than full continuum models, these approximate continuum models have limitations. They might model surfaces or isotropic material only, or may require signi£cant off-line pre-processing, which precludes general topological changes or dynamic material fracture.

Models using particles treat matter similar to the lumped masses of mass-spring systems. Physically-based particle systems involve behavior expressed in terms of potential energy functions, force, and equilibrium. Some applications, such as cloth modeling, £x particles within a lattice structure [5], or constrain them to a surface [27]. The particle systems we are interested in, however, do not involve spring connections or hold particles in a £xed arrangement. Particle-based simulations of this type have been used in applications ranging from the scale of astrophysics, wherein particles capture the formation and evolution of galaxies, to the atomic level where they have modeled electron ¤ow in semiconductors and plasma ¤ow in magnetic £elds [10]. At intermediate scales, MD simulations employ aggregates of particles to investigate the behavior of condensed matter in various states [1],[16]. Both oriented [14] and unoriented particles [9] have been used for animation, while oriented particles have been used to form minimal-energy surfaces that can be interactively shaped [23]. As with other approaches, solving particle system equations of motion is parallelizable, and with the use of spatial partitioning, models with reasonably large particle populations are computationally viable. However, because surface shape is ignored in MD simulations and interior volumes are ignored in surface modeling systems, these types of particles systems are incomplete. In addition, objects composed of particle ensembles are not easily visualized using polygonal surfaces.

3. Particle model dynamics

Both oriented and unoriented particles are used in a bag-of-particles simulation. Both types have position p, but some are de£ned with an orientation, q. Oriented particles in this system are *surface* particles forming a thin exterior membrane, serving as a containing boundary to hold unoriented *volume* particles in an object's interior. The model uses four potentials that can be grouped into three categories, according to their effect:

(1) One potential producing volume forces that gather particles into a shapeless, space-£lling conglomeration,
(2) Two potentials producing surface forces and torques that arrange particles into locally planar surface con£gurations, and
(3) One potential producing shaping forces and torques that arranges these planar sections into a global object shape.

The overall effect of the various forces and torques produced by these potentials is to form a *bag of particles* - surface particles arranged in the surface shape of an object, that bound and contain space-£lling volume particles.

3.1. Volume particles

To produce the volume forces aggregating some particles into a space-£lling interior mass, we use a form of the Lennard-Jones (LJ) potential commonly used in MD simulations to model condensed matter:

$$\phi(r_{ij}) = 1.5\left[\left(\frac{1}{r_{ij}}\right)^3 - \left(\frac{1}{r_{ij}}\right)\right], \qquad (1)$$

where r_{ij} is the distance between two molecules i and j, and $\varepsilon = 1.5$ is the energy required to overcome the cohesion of molecules [10]. In this form the equilibrium particle spacing r_0 occurs at $r_{ij} = \sqrt{3}$ (Figure 1). For computational simplicity we assume particles have unity mass. To make the number of LJ force computations more tractable, particle interactions are ignored beyond a cutoff distance, $r_c = 3r_0$, where the LJ force is small.

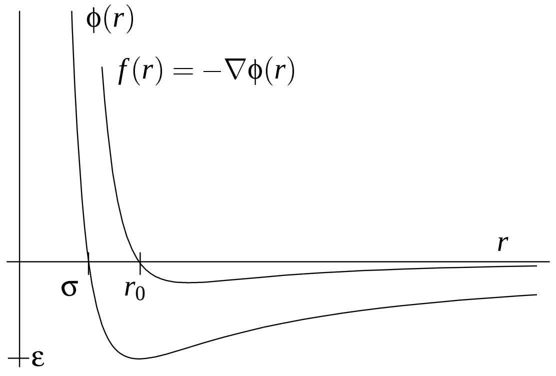


Figure 1: *Lennard-Jones potential, $\phi(r)$, and force, $f(r)$.*

3.2. Surface particles

The space-£lling particles described in section 3.1 are not suitable for creating surfaces. Instead, a mechanism is needed to form a kind of membrane or bounding surface around interior particles. For this purpose, we use the *co-normal* and *co-planar* potentials (ϕ^{CN}, ϕ^{CP}) and oriented particles, in addition to the LJ potential, as is done in the surface modelling system of [23]. Together, these three potentials produce surface forces and torques that act to arrange particles into locally planar surface sections. To form arbitraily shaped surfaces and not simply planes, however, the additional *shaping* forces and torques described next are needed.

As described in section 3.1, we can create amorphous aggregates of unoriented particles from volume forces. Alternatively, with co-normal, coplanar and LJ surface forces and torques only, we can form a planar surface of oriented particles. In neither case, however, can we create any arbitrary non-planar surface shape with its interior volume. To create an arbitrarily shaped closed surface, a force is needed that will arrange locally planar regions of particles into the desired global object shape, and a torque is also needed that will orient a surface particle to the object's surface normal. We obtain these shaping forces and torques with the following steps:

3.3. Shape from distance maps

1. Represent an object as a 3D voxel bitmap. This is illustrated in Figure 2(a). The bitmap is simply a binary labelling of connected components in a voxelization of the object, $V = V_0 \cup V_1$, where V_0 = non-object voxels, V_1 = object voxels.
2. Identify the object surface. *Surface* voxels separate *inside* voxels from *outside voxels*. We identify surface voxels as object voxels sharing common faces with (6-connected to) non-object voxels, $V_1 = V_{in} \cup V_{surf}$, $V_0 = V_{out}$. Integer labels distinguish outside, surface, and inside voxels, as shown in Figure 2(b).
3. Compute the distance map. Given the set $V_s \in \Re^3$ of points on a closed surface, the scalar, Euclidean distance of a point $\mathbf{x}$ to the nearest point on V_s is the *distance map* value at $\mathbf{x}$. The 2D, 8-point sequential Euclidean distance (8SED) algorithm [4] gives an approximation $d(v, v_s) = (|\Delta x|, |\Delta y|)$ of the distance metric $\sqrt{\sum(v - v_s)^2}$, using integer operations only, but the algorithm computes an unsigned distance, as it does not distinguish between inside and outside pixels. We extend the 8SED algorithm to 3D and modify it to compute the gradient of the *signed* distance (producing a *gradient map*, GM), storing for each voxel v the signed vector components of the distance $d(v, v_s) = (\Delta x, \Delta y, \Delta z)$ from the center of the voxel to the center of the nearest surface voxel v_s.

The gradient map is subsequently used to ensure that surface and volume particles occupy surface and volume voxels, respectively. Speci£cally, the pre-computed gradient map vectors are used as shown in Figure 3: (a) for a surface particle occupying an outside or an inside voxel v, a gradient map force $f^{GM} = d(v, v_s)$ will push the particle into the nearest surface voxel v_s, (b) within a surface voxel, the force is $f^{GM} = r_s - r$, pushing the surface particle at position r towards the voxel center, r_s, and (c) by treating the gradient map vector as a surface normal, $f^{GM} = n_s$, the torque $\tau^{GM} = n_i \times n_s$ will orient a surface particle normal n_i to the surface normal. Using oriented particle forces and torques, and gradient map vectors as described above, locally planar regions of particles conform to an object's surface shape. It is important to note that gradient map forces and oriented particle forces are *both* required for forming a surface from particles. Without oriented particle forces, surface particles

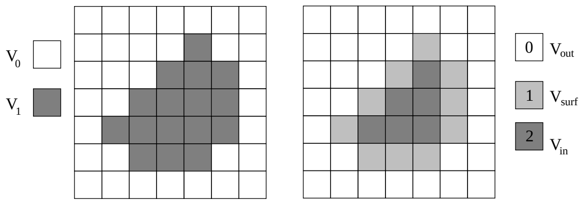


Figure 2: *Object as a labelled bitmap.*

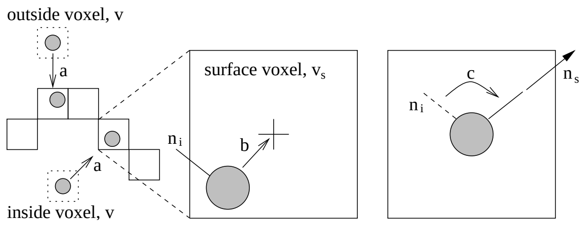


Figure 3: *Oriented particle forces and torques.*

will not distribute themselves evenly over the surface; rather, groups of particles near a common surface voxel will tend to clump together as the gradient map force attracts them to that voxel.

It is possible for a surface particle to leave a surface voxel while seeking equilibrium after an external force is applied. The gradient map force acts, however, to return the particle to the nearest surface voxel. Likewise, it is possible for an interior particle within an expanding surface to stray into a surface voxel (or even into an outside volume voxel), but the gradient map force is applied to these particles to return them to the inner volume. Referring to Figure 4: (a) when an inside volume particle is in an outside voxel, the gradient map force $f^{GM} = d(v, v_s)$ will push the particle into the nearest surface voxel, (b) when an inside volume particle is in a surface voxel, application of $-f^{GM}$ will push the volume particle towards the nearest inside voxel. The expected behavior of the two different types of particles is established in this manner: surface particles will occupy surface voxels, and volume particles will occupy volume voxels, at equilibrium.

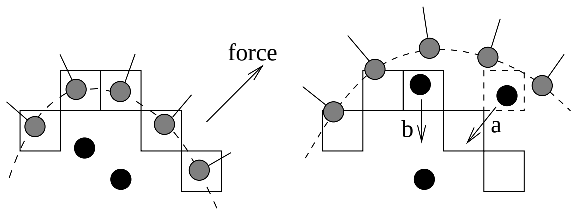


Figure 4: *Gradient map force applied to volume particles.*

3.4. Bag of particles

To form an elastic bag of particles, some particles must arrange themselves into surfaces, others must £ll the enclosed volume, and the desired elastic behavior of the entire system must be captured in the particle dynamics. In this model, the elastic behavior of a soft, squishy bag (as opposed to a more rigid, £rmer bag) is not based on mechanical properties such as stress and strain, but results from a combination of the various forces and torques acting on the different types of particles.

In the simplest case, a bag is composed of two different particle types: space-£lling (interior), unoriented volume particles that clump together under the in¤uence of LJ forces, and oriented surface particles forming an enclosing surface, which are subject to LJ forces, co-normal and co-planar forces and torques, and gradient map forces and torques. The gradient map force acts to keep surface (volume) particles in surface (volume) voxels, and the gradient map torque maintains surface particles oriented to the surface normal.

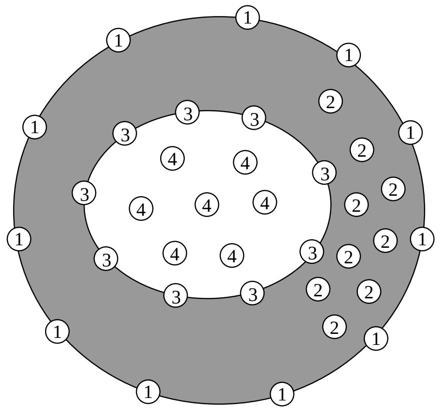


Figure 5: *Particles in a bag within a bag.*

A more complex example is illustrated in Figure 5, where one bag contains a second bag with different elastic behavior. To model bags with non-homogenous material such as this we de£ne an additional particle attribute, its species. This example requires four particle species: two composing an outer bag (surface species-1, volume species-2) that surround an inner bag composed of two other species (surface species-3, volume species-4). We would like the species-1 surface to contain only species-1 particles, except perhaps while coming to equilibrium after an external force is applied, and similarly for the species-3 surface. If we only labelled voxels as outside, surface, or inside, as can be done for a single bag, it is possible for gradient map forces to push a species-1 surface particle onto the species-3 surface (and vice versa). To prevent this, a voxel is labelled with both the particle species that should occupy it at equilibrium, and the species of the nearest surface. By ordering voxel species numbers in increasing order from outer to inner and applying a negative gradient map force in some situations, particles will always move towards the nearest surface labelled with its species number, i.e., they will also order themselves by increasing species number from outer to inner, as in Figure 5.

3.5. Behavior of a particle species

The dynamic behavior of a species is determined by the balance between gradient map and oriented particle forces and torques. It is set such that gradient map forces and torques (f^{GM}, τ^{GM}) have a greater effect than oriented particle forces and torques (f^{OP}, τ^{OP}), causing surface particles which would otherwise form small planar areas to form the global object shape:

$$f_i = \alpha_f^{OP} f_i^{OP} + \alpha_f^{GM} f_i^{GM}, \quad (2)$$

$$\tau_i = \alpha_\tau^{OP} \tau_i^{OP} + \alpha_\tau^{GM} \tau_i^{GM}. \quad (3)$$

The oriented particle force (torque) on a species-i particle is the sum of Lennard-Jones, co-normal and coplanar forces (torques):

$$f_i^{OP} = f_i^{LJ} + f_i^{CN} + f_i^{CP}, \quad (4)$$
$$\tau_i^{OP} = \tau_i^{CN} + \tau_i^{CP}, \quad (5)$$

where the oriented particle force (torque) from a potential is itself a weighted sum over all pairs of interacting species. For example, the oriented particle forces for a species-i particle are weighted sums of the pairwise forces between that particle and all other species (1, 2, ...) with which it interacts:

$$f_i^{LJ} = \alpha_{i1}^{LJ} \sum_1 f_{i1}^{LJ} + \alpha_{i2}^{LJ} \sum_2 f_{i2}^{LJ} + \dots, \quad (6)$$
$$f_i^{CN} = \alpha_{i1}^{CN} \sum_1 f_{i1}^{CN} + \alpha_{i2}^{CN} \sum_2 f_{i2}^{CN} + \dots, \quad (7)$$
$$f_i^{CP} = \alpha_{i1}^{CP} \sum_1 f_{i1}^{CP} + \alpha_{i2}^{CP} \sum_2 f_{i2}^{CP} + \dots. \quad (8)$$

where the summations are over all particles in the indicated species. The coef£cients α_{ij}^{ϕ} are a set of (possibly zero) interspecies weights, which determine the strength of interactions between all species pairs i and j. Typically, pairwise interactions are not symmetric ($\alpha_{ij}^{\phi} \neq \alpha_{ji}^{\phi}$).

Increasing the weights α_{ij}^{CP} increases the magnitude of coplanar forces and has the effect of ¤attening regions of surface particles. Using a global value for all species-*i* surface particles is acceptable if the surface curvature does not vary widely. For surfaces where curvature does vary signi£cantly, however, a value that is adequate for ¤at regions is too large for areas with high curvature, with particles tending to ¤atten rather than conform to the highly curved regions. To avoid this, α_{ij}^{CP} is weighted according to the surface curvature of the static object, computed from the 3D voxel bitmap [25].

The particle behavior resulting from interspecies weights establishes the elasticity of the surface and the bulk 'matter' it contains. Applying these weights to forces and torques obtained by integrating the particle system equations of motion determines the equilibrium shape of a bag of particles, and its time-varying shape when external forces are applied.

3.6. Constructing a model

A bag is initially populated with particles using birth and death processes that maintain approximately constant particle density. The equilibrium, lowest energy con£guration for a group of particles subject to an isotropic potential such as the LJ potential is hexagonal close-packed. Particles are added via birth in regions of low density, and removed via death if they are too closely spaced.

Using a simple menu and button interface and a few input device clicks, a bag can be fully populated in the following manner. The bag surface is created £rst. The simulation volume is seeded with one particle of each surface species using a mouse to select the species from a menu, then clicking anywhere in the simulation volume with a 3D input device, to add a particle of the selected species. After gradient map and oriented particle forces (torques) have attracted particles to their proper surface, birth is enabled for surface species. In a similar manner, bags are £lled with volume particles by seeding a volume voxel of each species with a particle of the same species, and enabling birth for those volume species. Figure 6 shows several bags within bags during creation of the surfaces using this process.

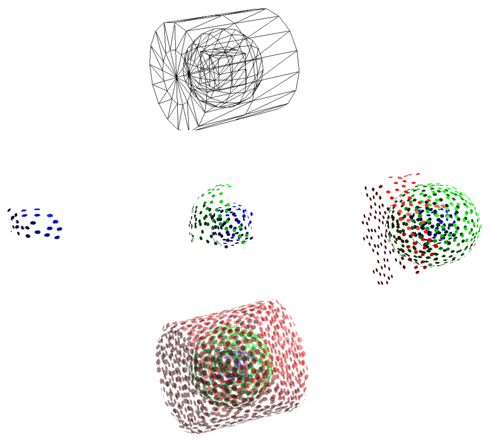


Figure 6: *Creating multiple surfaces.*

4. Implementation

The following sections discuss speci£c implementation details, including the types of data used to create a bag of particles model, how a model is visualized and deformed by the user, the integration method used in the particle simulation, and measures taken to reduce computational complexity.

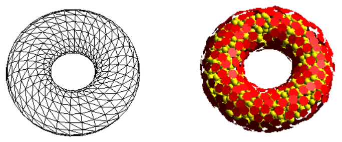
4.1. Visualization

To visualize a bag of particles model, a surface triangulation computed from particle positions can be rendered with shaded or textured polygons, giving a snapshot of the system at a speci£c time step. However, since particles can move freely over the surface in response to forces, visualizing a dynamically changing surface shape with this technique would require a new surface triangulation at each time step, and would not be interactive. For this reason, we visualize a bag of particles using a single shaded polygon for each surface particle, and shaded spheres for particles contained in the interior. Because the surface is visualized only where particles are located, and not covered with a complete polygonal tesselation, the gaps between particles are sometimes noticeable as a hole in the surface.

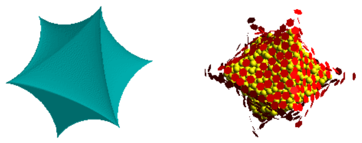
4.2. Data sources

The common representation for all bag-of-particles models is the 3D voxel bitmap. A labelled gradient map derived from this bitmap is then used to derive the shape forces arranging particles into an object's speci£c shape. Models created from several fundamentally different types of source data are shown in Figure 7. The birth process used to populate these models, described in section 3.6, proved to be robust over this wide range of shapes and data sets.

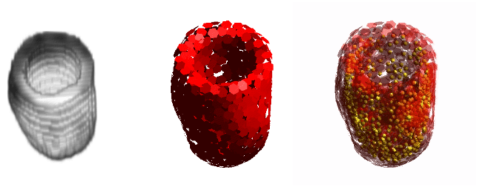
The torus bitmap was created from a polygonal model scan-converted into voxels. The bitmap for the star-like model came directly from the implicit function for a superellipsoid. The sampled data bitmap was created by segmenting PET image slices.



(a) Polygonal mesh



(b) Implicit function



(c) Acquired data (PET images)

Figure 7: *Example source data and bag of particles models.*

4.3. Interaction

In the present system, a user can deform models by touching them with a probe whose endpoint is modeled as a single LJ type particle. The form of the LJ force function used for this particle is

$$f_i^{probe}(r_i) = k_1 f_i^{LJ}\left(\frac{r_i}{k_0}\right), \tag{9}$$

where r_i is the spacing between the probe and particle i. The effect of the two coef£cients k_0 and k_1 is a probe particle with an adjustable equilibrium spacing, and the ability to strengthen or weaken the effect of the probe on other particles. Higher values of k_0, for example, result in a probe that interacts at longer ranges (i.e., a larger probe), while higher values of k_1 result in stronger probe forces.

A PHANToM 3D haptic interface device is used to position the probe particle in the simulation volume. The PHANToM provides six degrees-of-freedom position sensing and three degrees-of-freedom force feedback for a stylus-like end-effector [15]. With a probe particle £xed to the tip of the stylus, the user can perform small elastic deformations to a bag of particles model (Figures 8 and 9). The GHOST haptic

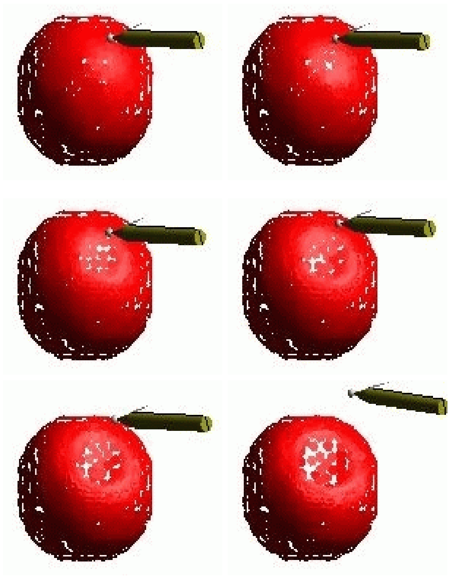


Figure 8: *Deforming a bag with a probe modeled as a single particle.*

programming API [7] used to interface the PHANToM device with the bag-of-particles simulation is also used for force-feedback modeling. By treating gradient map vectors as a £eld of force vectors, reaction forces applied to the PHANToM can be felt when a bag is deformed. The modeled effect is £ngertip interaction with elastic material. An object's surface can be felt when touched, but other perceptions such as weight and texture have not been modeled.

4.4. Time integration

The forces and torques acting on a particle include the oriented particle and gradient map forces and torques described previously, as well as external user-applied forces, f_{ext}, applied by the interaction probe, and velocity dependent viscous damping:

$$f_i = \alpha_f^{OP} f_i^{OP} + \alpha_f^{GM} f_i^{GM} + f_{ext} - \beta_v v_i \tag{10}$$

$$\tau_i = \alpha_\tau^{OP} \tau_i^{OP} + \alpha_\tau^{GM} \tau_i^{GM} - \beta_\tau \omega_i \tag{11}$$

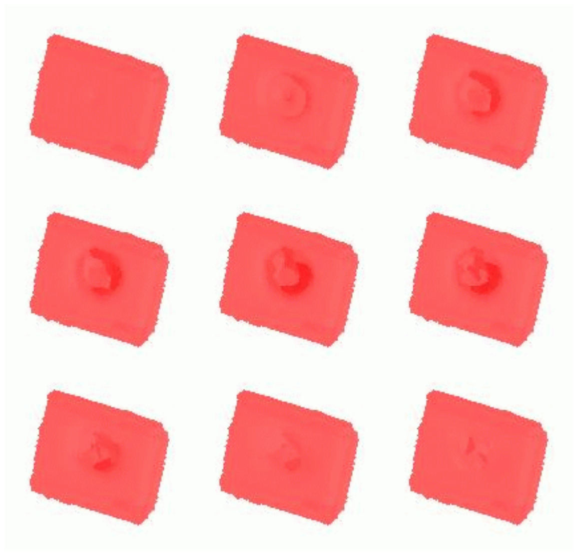


Figure 9: *Sequence showing a planar sheet being deformed (the single point probe particle is not shown).*

Translational ($\beta_v v_i$) and rotational ($\beta_\tau \omega_i$) damping act as drags opposing particle motion and improves numerical stability. Given these expressions for force and torque, we can solve the equations of motion governing particle system dynamics, yielding an updated position and orientation for each particle, at each integration time step.

We determine particle position and orientation using classic Newtonian mechanics:

$$\begin{array}{ll} a = F/m & \alpha = I^{-1}\tau \\ \dot{v} = a & \dot{\omega} = \alpha \\ \dot{x} = v & \dot{q} = \frac{1}{2}[0,\omega]q \end{array} \qquad (12)$$

The left column expresses Newton's second law for rigid body translational motion as a set of coupled £rst order ordinary differential equations, and analagously in the right column for rotational motion, where the inertia tensor, I, relates a torque, τ, exerted on a particle, to rotational motion expressed by angular acceleration, α. The relationship between a particle's orientation, represented by a unit quaternion q, and its angular velocity, ω, expressed as the quaternion $[0,\omega]$, is given by [21].

The equations of motion are integrated at each time step using the Velocity Verlet method [22]. Other integration methods such as Runge-Kutta or predictor-corrector algorithms offer greater accuracy and permit larger time steps at the cost of implementation complexity [17], but the chosen method is easily implemented, numerically stable, and amenable to quaternion representation for orientation [19].

Because force and torque calculations dominate in the overall simulation, several techniques are used to reduce computational complexity.

4.5. Complexity

In a naive particle system implementation, calculating pairwise interparticle forces is $O(N^2)$ in the number of particles, N. However, spatial subdivision can be employed to reduce complexity. In our implementation, the simulation volume is partitioned into cells with size equal to the LJ force cutoff distance r_c, each cell containing a list of particles within that cell subvolume.

Cell lists are updated at each time step when new particle positions have been determined. For each particle p_i in a cell list, pairwise force calculations are computed at the next time step only for other particles p_j in that cell and in (at most) the adjacent 26 cells, if the particle separation $\|r_{ij}\|$ is less than the force cutoff distance. Assuming particle density is constant - a reasonable assumption for particles subject to a LJ potential - such a spatial subdivision reduces complexity to $O(N)$ in the number of particles.

The number of particles needed to fully populate a model, and therefore the number of pairwise force calculations to be performed, can be adjusted by varying the ratio of the gradient map resolution to the force scale (voxel size$/r_0$). At a one-to-one ratio where each gradient map voxel is one equilibrium particle spacing wide, a fully populated object occupying N voxels requires about N particles. Increasing the length of the equilibrium spacing with respect to gradient map voxel size reduces the number of particles required to £ll a volume or cover a surface. Larger spacing between surfaces particles is equivalent to an undersampling of the surface, however, which results in a loss of surface detail.

To improve simulation interactivity, the rendering, interaction and dynamics tasks are performed separately on two hosts communicating via sockets. Force calculations and state updates are performed on an SGI Origin 2000 multiprocessor with up to sixteen 195 MHz MIPS R10000 CPU/MIPS R10010 FPU and 256 Mb main memory per CPU available. Visualization is done on an SGI Indigo2 195 MHz MIPS R10000 CPU/MIPS R10010 FPU with 96 Mb main memory and Maximum Impact graphics. Interaction is served by a process on the visualization host, driving the PHANToM in a sensing loop cycle of about 1000 Hz. At each time step the PHANToM position and the state of all particles are passed across the network between the dynamics and graphics hosts in UDP packets.

5. Results

Simulation runs were conducted using several bag-of-particles models to assess its performance as an elastic deformable model. In these runs, the simulation used a single 3D voxel bitmap for each model, and deformations were made interactively using the probe particle discussed in section 4.3. A £nal simulation was performed using a *series* of bitmaps to model a human heart left ventricle through a full cardiac cycle. In this run, the model was not deformed interactively, but by cycling through a sequence of gradient maps. In this scheme, the set of shape forces changes from gradient map to gradient map, causing surface particles to conform to the changing shape of the endocardial surface as it varies from bitmap to bitmap. The importance of this latter technique is demonstrated by obtaining a medically important, *dynamic* measure such as blood pressure through the cardiac cycle, from *static* image data.

5.1. Single Bitmap Models

Table 1 gives performance results for the models described in Section 4.2. Values were recorded during routine traf£c load on the network connecting the graphics and dynamics hosts. No attempt was made to optimize the code, however, or to balance the load across CPUs on the dynamics host. As a result, although computations are distributed across CPUs by assigning simulation subvolumes to speci£c processors, the speed of the dynamics simulation is bottlenecked by the CPU handling the subvolume with the largest number of particles. Furthermore, the frame rate values in Table 1 do not re¤ect a possible optimization. Since pair potentials such as the LJ potential produce symmetric forces, the LJ force needs to be computed only once for each pair of particles. With this optimization in place, frame rates are expected to approximately double.

	Torus		Star		Ventricle
$\frac{\text{voxel size}}{r_0}$	2	1.4	2	1.4	2
Surface	191	386	158	302	780
Volume	169	614	242	698	651
Total	360	1000	400	1000	1431
fps	15.2	8.4	14.0	8.8	6.5

Table 1: *Simulation performance for 12 CPU, for the models shown in Figure 7.*

5.2. Application: Myocardium Dynamics

The bag-of-particles models described in section 5.1 are derived from a single 3D voxel bitmap, and the behavior of surface particles in these models is determined by the forces and torques described in sections 3.2 and 3.3. In this section an application of the bag-of-particles approach is described that uses an additional mechanism for dynamically changing the shape of an object.

The images in Figure 10 are 3D texture volume renderings of the left ventricle (LV) of a human myocardium (heart muscle), obtained from a sequence of positron emission tomography (PET) image slices, acquired at eight evenly spaced temporal intervals (gated PET) during one complete cardiac cycle. The image slices in each frame were segmented to extract the inner (endocardial) surface of the ventricle, and a 3D voxel bitmap was created from each set of these segmented slices. Distance gradient maps were then created from each of these bitmaps, as described in section 3.3.

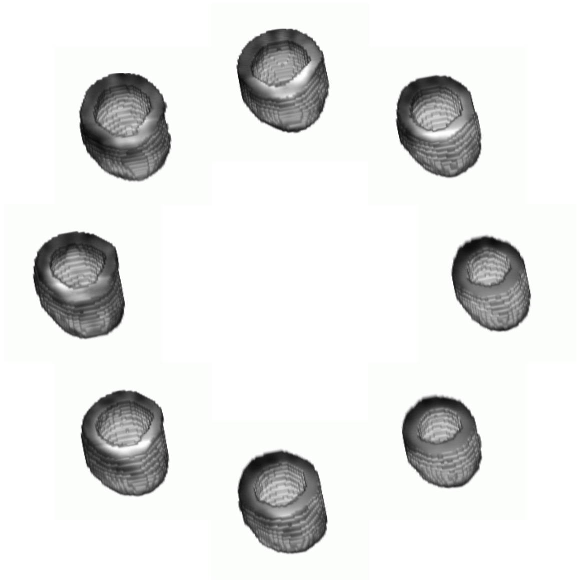


Figure 10: *Eight frames of a gated PET acquisition of a human heart left ventricle during one cardiac cycle, volume rendered using 3D texture.*

By looping through this set of gradient maps as the simulation progresses, in the same manner as key frames in an animation sequence, surface particles comprising the ventricle surface conform to the changing shape of the endocardium, compressing the interior blood volume particles during systole. Figure 11 depicts this LV bag-of-particles model.

The expected blood pressure response of a beating ventricle can be quanti£ed in a bag-of-particles model, as shown in the pressure vs. time plot of Figure 12. Pressure is de£ned in terms of the average temperature, T, of the N blood particles, and the virial function[8],

$$T = \frac{1}{3} N k_b \sum_i m_i v_i^2, \tag{13}$$

where temperature is a function of particle mass and velocity (m_i, v_i) and the Boltzman constant (k_b), and the virial function is given by

$$\mathcal{V} = \sum_{i<j} r_{ij} \cdot f_{ij}. \tag{14}$$

These expressions simplify by using unity particle mass

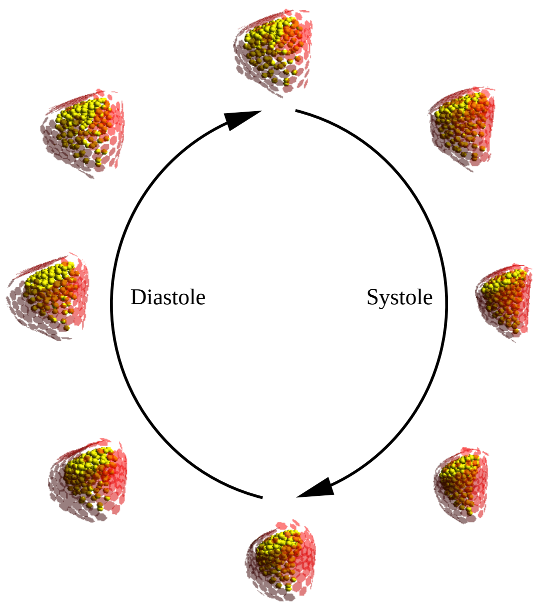


Figure 11: *Left ventricle bag-of-particles model through one cardiac cycle.*

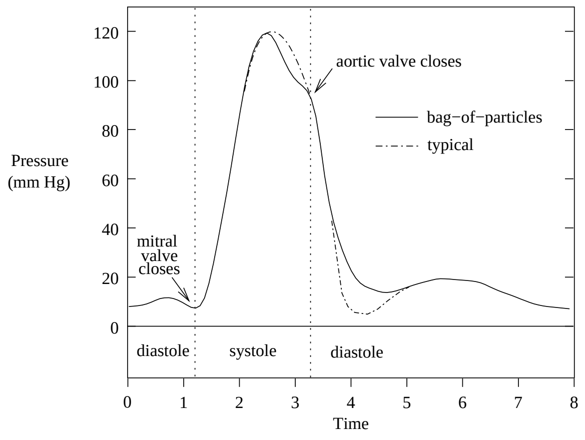


Figure 12: *Shape of ventricle pressure vs. time through one cardiac cycle, for a bag-of-particles model and a typical left ventricle (scaled to give peak systolic pressure of 120 mm Hg).*

($m_i = 1$), and by scaling temperature with an assumed Boltzman constant value, $k_b = 1$, giving

$$P = \frac{1}{3V}\left(\sum_i v_i^2 + \frac{1}{N}\sum_{i<j} r_{ij}\cdot f_{ij}\right). \qquad (15)$$

Volume, V, is approximated by the number of volume voxels in the currently selected gradient map.

Pressure values calculated using Equation 15 were averaged over ten simulation time step intervals and recorded. To obtain the plot in Figure 12, a curve was £tted to the averaged pressure data using cubic spline interpolation, then scaled to give a peak pressure equal to nominal peak systolic pressure of 120 mm Hg. Figure 12 .

Superimposed on the bag-of-particles pressure curve is a plot representing the typical shape of the pressure expected in a LV. The difference in the shape of the two curves is due to anatomic inaccuracy in the bag-of-particles model. The LV PET data does not include an aortic valve. Hence, in the bag-of-particles model, blood particles remain within the LV endocardium, preventing pressure from dropping further as would occur if blood was ejected from the LV through the aortic valve.

6. Conclusions

A physically-based, elastically deformable particle system model with several advantages over other particle system modeling techniques has been described. Unlike systems that model surfaces only, or use a single type of particle, this method models both surfaces and interior volumes using multiple particle species in a *bag of particles*. Since volume particle dynamics is governed by the inter-molecule potentials employed in MD simulations, inner material can exhibit the behavior of gaseous, liquid, or semi-solid condensed matter. A distance gradient map that captures object shape provides forces and torques which elastically restore the shape after deformation, but also can be used as reaction forces to external, user applied deformations, providing a simple means of haptic feedback. In addition, species labels associated with the gradient map, and a set of interspecies parameters allows bags-within-bags to represent objects composed of non-homogenous material, where each bag models a different elastic behavior. By coupling a gradient map that captures shape, with a physically-based particle system that models behavior, any shape that can be represented as a bitmap of connected components can become an elastically deformable bag of particles.

The applicability of this approach for objects of widely different shapes, with different source data representations was demonstrated. A simple, point-probe interaction mechanism was also illustrated. Because physical behavior is based on interacting particles, a separate collision-detection mechanism for tool-object interactions is not needed.

The approach was applied to modeling a human heart left ventricle through a complete cardiac cycle, demonstrating the ability to measure time-varying quantities of medical interest, such as blood pressure and cardiac ejection fraction, using only static medical images.

Areas for further work are suggested. Since the gradient map is precomputed, a bag of particles is presently an elastic model only. Also, the speci£c elastic dynamics of a model is not based on mechanical material properties, but results from values for the interspecies parameters that must be determined empirically. With severe deformations where material fracture is expected, the model breaks down, allowing volume particles to escape outside the surface.

References

1. M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*. Oxford University Press, 1987. 2
2. M. Bro-Nielsen and S. Cotin, Real Time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation, *EUROGRAPHICS '96*, **15**(3):C57–C66, 1996. 2
3. S. Cover, N. Ezquerra, J. O'Brien, R. Rowe, T. Gadacz and E. Palm, Interactively Deformable Models for Surgery Simulation, *IEEE Computer Graphics and Applications*, **13**(6):68–75, 1993. 2
4. P. Danielsson, Euclidean Distance Mapping, *Computer Graphics and Image Processing*, **14** 227–248, 1980. 3
5. B. Eberhardt, A. Weber and W. Strasser, A Fast, Flexible, Particle System Model for Cloth Draping, *IEEE Computer Graphics and Applications*, **16**(5):52–59, 1996. 2
6. I. Essa, S. Sclaroff and A. Pentland, A Uni£ed Approach for Physical and Geometric Modeling for Graphics and Animation, *EUROGRAPHICS '92*, **11**(3):C129–C138, 1992. 2
7. *General Haptic Open Software Toolkit (GHOST)*, SensAble Technologies, Inc., 215 First Street, Cambridge, MA, 02142. 6
8. J. Hansen and I. McDonald, *Theory of Simple Liquids*. Academic Press, 1986. 8
9. T. Hilton and P. Egbert, Vector Fields: an Interactive Tool for Animation, Modelling and Simulation with Physically Based 3D Particle Systems and Soft Objects, *EUROGRAPHICS '94*, **13**(3):C329–C338, 1994. 2
10. R. Hockney and J. Eastwood, *Computer Simulation Using Particles*. Institute of Physics Publishing, 1988. 2
11. K. Huebner, *The Finite Element Method for Engineers*. John Wiley and Sons, 1975. 2
12. M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active Contour Models, *Proc. First International Conference on Computer Vision*, 259–268, 1987. 2
13. Y. Lee, D. Terzopoulos and K. Waters, Realistic Modeling for Facial Animation, *Proc. ACM SIGGRAPH*, 55–62, 1995. 2
14. J. Lombardo, C. Puech, Oriented Particles: A Tool for Shape Memory Objects Modelling, *Graphics Interface 95*, 255–262, 1995. 2
15. T. Massie and J. K. Salisbury, The PHANTOM Haptic Interface: A Device for Probing Virtual Objects, *Proc. ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1994. 6
16. G. Miller and A. Pearce, Globular Dynamics: A Connected Particle System For Animating Viscous Fluids, *Computers and Graphics*, **13**(3):305–309, 1989. 2
17. S. Nakamura, *Applied Numerical Recipes in C*. Prentice Hall, 1993. 7
18. J. O'Brien and J. Hodgins, Graphical Modeling and Animation of Brittle Fracture, *Proc. ACM SIGGRAPH*, 137–146, 1999. 2
19. I. Omelyan, On the numerical integration of motion for rigid polyatomics: the modi£ed quaternion approach, *Computers in Physics*, **12**(1):97–103, 1998. 7
20. A. Pentland and J. Williams, Good Vibrations: Modal Dynamics for Graphics and Animation, *Proc. ACM SIGGRAPH*, 215–222, 1989. 2
21. J. Powles, W. Evans, E. McGrath, K. Gubbins and S. Murad, A computer simulation for a simple model of liquid hydrogen chloride, *Molecular Physics*, **38**(3):893–908, 1979. 7
22. W. Swope, H. Andersen, P. Berens and K. Wilson, A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters, *Journal of Chemical Physics*, **76**(1):637–649, 1982. 7
23. R. Szeliski, and D. Tonnessen, Surface Modeling with Oriented Particle Systems, *Proc. ACM SIGGRAPH*, 185–194, 1992. 2, 3
24. D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, Elastically Deformable Models, *Proc. ACM SIGGRAPH*, 205–214, 1987. 2
25. J. Thirion, and A. Gourdon, Computing the Differential Characteristics of Isointensity Surfaces, *Computer Vision and Image Understanding*, **61**(2):190–202, 1995. 5
26. X. Tu and D. Terzopoulos, Arti£cial Fishes: Physics, Locomotion, Perception, Behavior, *Proc. ACM SIGGRAPH*, 43–50, 1994. 2
27. A. Witkin, P. Heckbert, Using Particles to Sample and Control Implicit Surfaces. *Proc. ACM SIGGRAPH*, 269–277, 1994. 2
28. Q. Zhu, Y. Chen, and A. Kaufman, Real-time Biomechanically-based Muscle Volume Deformations using FEM, *EUROGRAPHICS '98*, **17**(3):C275–C284, 1998. 2