Chapter 2

Background

2.1 Surface Modeling

There exist a variety of techniques to solve the shape representation and manipulation problem for specific sets of shapes. For example, piecewise spline patches have gained wide spread acceptance in applications where controlling the degree of surface continuity is important. Constructive solid geometry is used in computer aided design and manufacturing (CAD/CAM) to create solid volume descriptions of shape, where the intrinsic properties of a solid object such as a continuous inside/outside boundary, a closed surface, and volume constraints are important. Surfaces of revolution and extruded two dimensional outlines create shapes with symmetry along a given dimension. Any faceted surface can be modeled using polygonal meshes, and subsequently smoothed using subdivision methods. Implicit surfaces are easily defined as an iso-value of a continuous scalar three dimensional field, and are easily manipulated when the field is a function of a skeleton or set of control points. Free-form deformations and parametric warping of space allow shapes to be deformed at global and local levels. This section reviews the most common shape representations, and discusses their advantages and limitations.

2.1.1 Polygonal Meshes

Polygonal meshes are perhaps the most widely used shape representation in computer graphics. Geometrically this is the simplest shape representation technique that allows the description of a wide variety of shapes and topologies. It is a natural description for surfaces, and with appropriate, constraints a description for solid volumes. Due to its simplicity it is included in virtually every modeling system, creating a lowest common denominator for shape representation.

However, simplicity comes at a cost. Polygonal meshes are limited to accurately describing surfaces composed of planar facets and thus are unable to accurately represent curved surfaces. Instead, curved surfaces are approximated by polygons which linearly interpolate between points on the original surface. At a fine resolution, the visual artifacts introduced by such approximations are negligible for rendering applications. At lower resolution, interpolating the vertex normals across the area of the polygon during the rendering phase results in the illusion of a smooth surface. For other more demanding applications, the exact representation of curves are required and the polyhedral model falls short.

One solution to generating smooth surfaces is to apply surface subdivision methods. Subdivision can be used to recursively approximate a polygonal mesh with finer polygonal meshes, which in the limit results in a smooth surface (Catmull and Clark, 1978; Peters and Reif, 1997) except at a number of extraordinary points (Doo and Sabin, 1978). As an alternative to approximating, one can use an interpolating subdivision scheme which in the limit interpolates a smooth surface between the original polyhedral vertices (Zorin, Schröder and Sweldens, 1996).

Still, to sculpt a shape, a designer must specify the location of each vertex, the edges joining each vertex, and the series of edges and the order of edges belonging to each polygon. In addition, some systems require that the outward facing polygons all have the same "sign", that is the sign of the plane normal defined by traversing edges around the polygon, either clockwise or counterclockwise, must be the same. Even with interactive tools to help specify the vertices and connections, this is a tedious and time consuming process, especially for models containing thousands of polygons.

2.1.2 Parametric Representations

In a parametric representation, a shape is defined by a set of parameterized functions, such as

$$x(u,v), \quad y(u,v), \quad z(u,v),$$

where u and v are parameters, and a surface point $\mathbf{x} = (x(u,v), y(u,v), z(u,v))$ is given by evaluating the parameterized functions. The parametric representation has two distinct advantages. First, an arbitrary number of surface points are easily generated by sweeping the parameters u and v through their domain, thus facilitating the rendering process. Second, different levels of curvature continuity can be controlled by careful selection of the underlying parametric equations.

Shapes can be defined using either global parametric or piecewise parametric patches. Quadrics and superquadrics (Barr, 1981) are examples of global parameterized primitives used to define complete surfaces such as spheres, ellipsoids, and tori. For a wider range of parametric shapes, one usually applies a piecewise surface construction approach.

Traditional spline techniques (Bartels, Beatty and Barsky, 1987; Farin, 1992) model an object's surface as a collection of piecewise-polynomial patches, with appropriate continuity constraints between the patches to achieve the desired degree of smoothness. Within a particular patch, a surface's shape can be expressed using a superposition of basis functions

$$\mathbf{s}(u_1,u_2) = \sum_i \mathbf{v}_i B_i(u_1,u_2), \tag{2.1}$$

where $\mathbf{s}(u_1,u_2)$ are the 3D coordinates of the surface as a function of the underlying parameters (u_1,u_2), $\mathbf{v}_i$ are the *control vertices*, and $B_i(u_1,u_2)$ are the piecewise polynomial *basis functions*. The surface shape can then be created by interactively positioning the control vertices or by directly manipulating points on an existing surface (Bartels and Beatty, 1989). Areas of a surface can be locally refined using a hierarchy of tensor-product B-splines (Forsey and Bartels, 1988).

2.1.3 Constructive Solid Geometry

Many applications, such as engineering and product design, require the representation of solid volumes. In constructive solid geometry (CSG), a solid is represented as a set-theoretic Boolean expression of primitive solid objects (Hoffman, 1989). Standard primitives are parallelepipeds (blocks), triangular prisms, spheres, cylinders, cones, and tori. Through a combination of simple primitives, complex shapes are easily constructed. The advantage of the CSG representation is that valid volumes can always be guaranteed.

To construct a shape, a user begins by instantiating a generic primitive by specifying the parameters of the primitive, such as the length and width of a parallelepiped or the radius of a sphere. Once instantiated, these primitives are combined using rigid motions and regularized Boolean set operations. These operations are regularized union, regularized intersection, and regularized difference. They differ from the standard set-theoretic operations by operating on the interior of the two solids, thereby eliminating lower dimensional geometric primitives which do not bound the resulting volume. The volume resulting from a regularized Boolean operation can then be combined with other volumes, until the final shape is realized.

2.1.4 Implicit Representation

In the implicit representation, a shape is defined as the locus of points that satisfy an equation $f(\mathbf{x}) = 0$, where $\mathbf{x}$ is a three dimensional point. For points inside of the surface, $f(\mathbf{x}) > 0$, and for points outside of the surface, $f(\mathbf{x}) < 0$.

Implicit surfaces enjoy several benefits such as the ability to efficiently compute inside/outside tests and the ability to easily build up complex shapes (Bloomenthal, 1989). For example, the standard set operations of union and intersection are easily implemented. Surfaces can be defined directly from a function or constrained in term of other geometric primitives. *Skeletal* surfaces can be defined in terms of distance constraints from a geometric entity. For example, a sphere is defined as a fixed distance from a point and a rounded cylinder is defined as a fixed distance from a line segment. Alternatively an implicit surface can be constrained to be a fixed distance from another surface, creating an *offset* implicit surface. The implicit formulation also allows for the blending of surfaces at branch points, a difficult problem for piecewise parametric surfaces. Implicit surfaces are often defined by combining algebraic functions based on control points, thus allowing surfaces to be easily deformed by displacing the control points (Blinn, 1982; Wyvill, McPheeters and Wyvill, 1986a).

A considerable disadvantage of the implicit formulation is that, in general, surface points cannot be directly computed as in the parametric representation. Algebraic surfaces may be ray-traced or rendered using incremental scan line techniques, and though certain class of implicit surfaces can be ray-traced (Blinn, 1982; Tonnesen, 1989; Wyvill and Trotman, 1989), no similar incremental techniques exist for arbitrary implicit surfaces (Bloomenthal, 1989). As an alternative, the implicit surface can be converted to a polygonal representation which can then be rendered with a conventional polygon renderer (Bloomenthal, 1988; Bloomenthal, 1989; Velho, 1990; Witkin and Heckbert, 1994). These techniques are discussed in more detail in Chapter 5.

2.1.5 Geometric Deformations

Geometric deformations provide another method for modeling shape. Mapping from one R^3 space domain to another R^3 space provides a "warping of space" and the geometry within that space, creating smooth deformations. By applying such a mapping Barr (1984) has shown the ability to bend, twist, and taper geometric objects. The normal and tangent vectors of the deformed object, as well as the ratio of volume change, can be calculated from the Jacobian matrix of the point transformation function. Deformations can also be realized by embedding a geometric object within a lattice of trivariate polynomials (Sederberg and Parry, 1986; Coquillart, 1990). By moving the control points of the lattice, one changes the R^3 to R^3 mapping and the embedded geometry is deformed. With appropriate constraints, the deformations can be defined both globally and locally, in a piecewise manner, and layered into a hierarchy of deformations. For more intuitive control the user can directly move points on the original surface to the desired deformed position (Hsu, Hughes and Kaufman, 1992), and then a least squares minimization is used to calculate the new lattice control point positions, thereby completing the mapping. While such mappings are very powerful, they preserve the underlying structure and parametrization of the surface, and thus do not allow the user to change topologies.

2.1.6 Variational Surfaces

By specifying "character lines", one can outline the shape of a surface and use either finite element (Celniker and Gossard, 1991) or variational techniques (Moreton and Séquin, 1992; Welch and Witkin, 1992; Welch and Witkin, 1994) to fit smooth surfaces between the character lines. The finite element approach provides physically realistic surfaces by minimizing an energy functional that describes the resistance to stretching and bending. Control of the final surface shape is achieved by first parameterizing the shape, then applying loads and geometric constraints. Terzopoulos and Qin (1994; 1996) use finite element techniques to solve a physics based generalization of non-uniform rational B-splines. Their model allows designers to sculpt shapes by applying forces and shape constraints, in addition to the traditional method of adjusting control points. The variational approach also uses geometric constraints and optimizes a constrained surface functional to create smooth surfaces. By stitching curves together the user can construct smooth shapes of arbitrary topology. Moreton and Séquin (1992) perform a non-linear optimization to minimize a fairness functional of the squared magnitude of the variation in principal curvatures. The result is a patchwork of Bézier patches which form a G^1 continuous surface. The variational surfaces of Welch and Witkin (1992) differ from the finite element and physically based models in that they "do not require the surface to respond in an intuitive or natural way to direct control-point manipulation". The variational surfaces of Welch and Witkin (1994) minimize the squared magnitude of the principal curvatures and are similar to implicit surfaces in that they allow the construction of arbitrary topology surfaces and, in general, cannot be explicitly computed.

2.1.7 Discussion

Existing surface representation techniques allow users to specify any geometric shape and any topology. However the specification of the shapes may be both time-consuming and tedious. Polygon meshes can be used to represent any faceted shape, and linearly approximate any curved surface. Piecewise parametric functions, such as spline surface patches, can be used to represent any curved surface with tangent plane or higher continuity conditions. Since polygons can be written as a parameterized equation, both polygonal meshes and parameterized surfaces are easily rendered by incrementally varying the parameters of the respective equations, thereby generating a stream of surface points. Implicit surfaces also allow the generation of geometric shapes of arbitrary topology, but do not possess a surface parametrization making it more difficult to render. Solutions to rendering are found through ray-casting techniques or by approximating with polygonal meshes or particles. However implicit surfaces have other valuable properties such as the guarantee of a closed surface, the ability to easily perform inside/outside tests, and the automatic blending of surfaces.

From a designer's point of view, perhaps the most serious drawback is the difficulty one has in creating and manipulating these complex shapes. The use of CSG allows the designer to add and subtract shapes with the knowledge that the final shape will always represent a closed solid volume. The use of free-form deformations allows shapes of static topology to be deformed both locally and globally. Finite element and variational techniques solve portions of the shape construction and manipulation problem by allowing the user to outline the shape as a set of character lines which the user then "skins" with a surface, which may then be locally deformed. These techniques address the construction of a valid geometric shape, but do not address the need to be able to manipulate the final shape at both the local and global level. While in some applications the specification of a shape is all that is needed, other applications such as animation require the ability to continually change the basic shape and in some cases even the topology, with minimal user intervention. The technique presented in this dissertation provides users a means to interactively shape surfaces, modify the surfaces locally and globally, including topological changes, with minimal user interaction.

2.2 Surface Reconstruction

Many vision researchers have investigated the reconstruction of $2\,^1\!/_2$-D viewer-centered surface representations (Terzopoulos, 1984; Boult and Kender, 1986). These representations are typically based on parametric spline models with internal strain energies. Equally intense effort has gone into the development of 3-D object-centered surface representations. These include generalized cylinders (Agin and Binford, 1973; Nevatia and Binford, 1977), superquadrics (Pentland, 1986; Solina and Bajcsy, 1990), and triangular meshes (Boissonnat, 1984), as well as their physics-based generalizations, dynamic deformable cylinders (Terzopoulos, Witkin and Kass, 1988), spheres (Miller et al., 1991; McInerney and Terzopoulos, 1993), superquadrics (Terzopoulos and Metaxas, 1991), and meshes (Vasilescu and Terzopoulos, 1992). Physically based models incorporate internal deformation energies and can be fitted through external forces to visual data such as 2-D images or 3-D range points. The $2\,^1\!/_2$-D viewer centered and 3-D object centered representations both assume a given topology, usually parameterized over a planar or spherical domain.

A common way to cope with unknown topological structure is to resort to a "patchwork" surface representation (Sander and Zucker, 1990) which abandons a global representation and describes the surface only locally in terms of planar, quadric, or cubic patches. A drawback of such local surface representations compared to globally parameterized geometric models is that they do not facilitate common surface analysis tasks such as area, curvature, and enclosed volume computations. More serious difficulties arise in the dynamic analysis of objects. Possible scenarios include the incremental reconstruction of surfaces from sequential views around objects, or the reconstruction, tracking, and motion estimation of dynamic non-rigid objects such as a beating heart. A globally consistent surface model can provide powerful constraints for solving these dynamic estimation problems.

Another approach to inferring topological structure is to construct a graph over the sample points which reflects spatially adjacent points (Hoppe et al., 1992; Edelsbrunner and Mücke, 1994; Guo, Menon and Willette, 1997). If the data are not sampled isotropically, that is with the same density in each dimension, then the correct surface may not be realized (Hoppe et al., 1992). The alpha shapes of Edelsbrunner and Mücke (1994) encode the spatial proximity relationship of the point set as a *simplicial complex*[1] overcoming the isotropic sampling restriction. In general, the result is not a connected surface interpolating the sample points, but a collection of points, lines, and surface patches with which the user is left to infer the shape. In a post-processing phase, the exterior faces of an alpha shape can be converted into a 2D-manifold surface (Guo, Menon and Willette, 1997), though due to the chosen alpha parameter the exterior faces may not adequately represent the full set of sample points, resulting in a loss of detail. To extract surfaces from medical images, *T-surfaces* (McInerney and Terzopoulos, 1997) overcome the topology restriction by continually recomputing a deformable surface based on an inside-outside classification

[1] A three-dimensional simplicial complex is a collection of k-simplices, $0 \leq k \leq 3$: A 0-simplex is a point, a 1-simplex is a line connecting two points, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron.

of a grid of voxels, analogous to the polygonization of implicit surfaces. A summary of deformable models applied to medical image analysis can be found in (McInerney and Terzopoulos, 1996).

Existing surface representations have limitations— viewer-centered methods make no attempt to represent non-visible portions of object surfaces, while object-centered methods make strong assumptions about object topology, and graph-theoretic methods make strong assumptions about the sampling density. This dissertation proposes a new approach to surface modeling which overcomes these limitations. The approach leads to flexible reconstruction algorithms which are able to compute detailed geometric descriptions that are not only inherently viewpoint invariant, but more importantly, are sufficiently powerful to represent surfaces of arbitrary topology. The algorithms can interpolate regular or scattered 3-D data acquired from an imaged object, without any a priori knowledge of the object topology.

2.3 Physically Based Modeling

2.3.1 Deformable models

Physically based modeling provides the ability to generate animations of physical phenomena through simulation. Starting with a parametric representation for the surface $\mathbf{s}(u_1, u_2)$ and adding a physical level of abstraction, Terzopoulos *et al.* (1987) create elastically deformable surfaces. To define the dynamics of the surface, they use weighted combinations of different tensor (stretching and bending) measures to define a deformation energy which controls the elastic restoring forces for the surface. Additional forces to model gravity, external spring constraints, viscous drag, and collisions with impenetrable objects can then be added. To simulate the movement of a deformable surface, these analytic equations are discretized using either finite element or finite difference methods. This results in a set of coupled differential equations governing the temporal evolution of the set of control points. Physically-based surface models can be thought of as adding temporal dynamics and elastic forces to an otherwise inert geometric spline model.

Physically-based surface models have been used to model a wide variety of materials, including cloth (Breen, House and Getto, 1991; Terzopoulos and Fleischer, 1988b), membranes (Terzopoulos et al., 1987), and paper (Terzopoulos and Fleischer, 1988b). Viscoelasticity, plasticity, and fracture have been incorporated to widen the range of modeled phenomena (Terzopoulos and Fleischer, 1988b). By adding muscles and skin to an otherwise inert model, the movement of characters, such as worms (Miller, 1988), fish (Tu and Terzopoulos, 1994), and human faces (Waters and Terzopoulos, 1990; Lee, Terzopoulos and Waters, 1995), can be automatically generated.

The main drawback of both splines and deformable surface models is that the rough shape of the object must be known or specified in advance (Terzopoulos, Witkin and Kass, 1987). For spline models, this means discretizing the surface into a collection of patches with appropriate continuity conditions, which is generally a difficult problem (Loop and DeRose, 1990). For deformable surface models, we can bypass the patch formation stage by specifying the location and interconnectivity of the point masses in the finite element approximation. In either case, defining the model topology in advance remains a tedious process. Furthermore, it severely limits the flexibility of a given surface model.

2.3.2 Fluid Models

The complex nature of liquids is extremely fascinating and poses a number of problems for research in computer graphics. Liquids exhibit a wide range of phenomena, such as conforming to the shape of containers, wave propagation, cresting and breaking as exhibited by ocean waves, splashing, sheeting, foam, and bubbles. Capturing all of these in one model is difficult and hence a variety of techniques have been proposed, each modeling a subset of the phenomena. Trigonometric functions have been used to model ocean waves (Peachy, 1986; Fournier and Reeves, 1986). Individual particles have been used to model the spray of water from boat wakes (Goss, 1990), the spray of cresting waves (Peachy, 1986), splashes (O'Brien and Hodgins, 1995), and waterfalls (Sims, 1989). Three approaches to modeling water as height fields have been proposed: as a linearized approximation to the shallow water equation (Kass and Miller, 1990), as columns of fluid where the volumes of fluid transferred between columns is conserved (O'Brien and Hodgins, 1995; Mould and Yang, 1997), and a solution of Navier-Stokes equations in two dimensions that is then mapped to a 3D height field (Chen and Lobo, 1995). To model rapid changes in topology of viscous liquids, coupled particle systems with attractive-repulsive forces have been used (Miller and Pearce, 1989; Terzopoulos, Platt and Fleischer, 1989; Tonnesen, 1991; Desbrun and Gascuel, 1995; Reynolds, 1997) as well as smoothed particle hydrodynamics (Roy, 1995; Desbrun and Gascuel, 1996) (discussed in Section 2.4.1). The particle based models are discussed further in Section 2.4.2. Recent work has solved the Navier-Stokes equations over a low resolution regular grid using a 3D finite-difference approximation (Foster and Metaxas, 1996; Foster and Metaxas, 1997a; Foster and Metaxas, 1997b).

2.4 Particle Systems

A particle system is a collection of point masses with associated forces whose movement is governed by the laws of physics. To describe each particle, a set of attributes, such as mass, position, velocity, and acceleration, are assigned to the particle. Potentials are commonly used to generate forces acting on the particles, and the movement of the particles is given by the laws of Newtonian physics,

$$\frac{\mathbf{f}_i(t)}{m_i} = \frac{\mathrm{d}\mathbf{v}_i(t)}{\mathrm{d}t}; \qquad \mathbf{v}_i(t) = \frac{\mathrm{d}\mathbf{x}_i(t)}{\mathrm{d}t},$$

where $\mathbf{f}_i$, $\mathbf{v}_i$, $\mathbf{x}_i$, and m_i are the force acting on, the velocity, the position, and the mass of particle i. Given initial conditions, these systems can be simulated over time by integrating the equations of motion. Depending on the forces applied, such systems can model a variety of complex and time dependent behavior. Rather than modeling as an Eulerian dynamical system, where the system state is defined at fixed grid samples, particle systems use a Lagrangian approach, where the samples of state follow the movement of the system.

2.4.1 Particle Systems in the Physical Sciences

In the physical sciences, particle systems have been used by to model a variety of phenomena including the evolution of galaxies, plasma, the properties of semiconductors, magnetic fields, compressible gas flows, and the phase changes in matter (Hockney and Eastwood, 1988; Heyes, 1998; Monaghan, 1992). Typically, each particle in the system models a primitive element, such as a star or molecule, of the phenomena under study. To predict the correct dynamic behavior, such simulations require the computation of complex interactions with high numerical accuracy. This problem is aggravated by the number of elements in the systems under consideration. For example, on a microscopic scale, the number of molecules in an ounce of water is on the order of 10^{25}. And on a cosmological scale, the number of stars in a galaxy[2] is on the order of 10^{10} to 10^{12} (Hockney and Eastwood, 1981).

Astrophysicists model the evolution of star systems based on the density of bodies and gravitational fields (Hockney and Eastwood, 1988; Heggie, 1987). To maintain accuracy, several approaches have been suggested. When modeling small clusters of stars, on the order of a 1000, the system can be directly modeled as a particle system with each particle representing a star, and forces computed directly. This requires $O(N^2)$ operations for N particles. For coulombic interaction[3], the computations can be reduced to $O(N)$ by using particle-mesh methods where short range forces are computed directly between particles, and long range forces are computed over a mesh (Greengard, 1988; Zhao, 1987). For larger systems, each particle models the mean properties of density and gravitional fields for clusters of approximately 10^6 stars, thus allowing a system of 10^4 particles to model a galaxy.

Molecular dynamicists have used particle systems to study solids, liquids, gases and the phase changes between these states (Barton, 1974; Christy and Pytte, 1965; Temperley, 1978; Trevena, 1975; Heyes, 1998). The concept of a pairwise intermolecular potential energy function has proven valuable in describing inter-molecular interactions in a quantitative fashion. The short range repulsive forces can be modeled by a potential energy function represented as an exponential expression $\phi_R(r) \propto r^{-n}$, where r is the distance between the two molecules. The long range attractive forces, to a reasonable approximation, can be treated together as a single expression, and modeled by the potential function $\phi_A(r) \propto r^{-m}$. Higher body interactions (e.g. 3-body interactions) can be computed for molecular systems, though the effects of these interactions are usually incorporated into the model by modifying the values of a pairwise potential (Heyes, 1998). In addition to modeling the phase transitions, particle systems have also modeled the macroscopic properties of matter, such as

[2] The number of galaxies in the observable universe is estimated on the order of 10^9.

[3] Coulombic potentials are inversely proportional to distance, that is r^{-1}. Coulombic forces are proportional to r^{-2}.

temperature, volume, and local geometry. Similar to modeling galaxies, modeling large molecular systems with particles is computationally expensive. Simulations which ignore distant particle forces have been used to reduce the computational cost, although this approach has been found to contribute to sensitivities in the computed phase diagram, especially close to critical points (Heyes, 1998). In modeling the phase changes between states, physicists are in effect modeling changes in *structure*. It is the ability to provide such fluid changes in structure that we wish to capture in our approach to the volumetric modeling of deformable materials.

Smoothed particle hydrodynamics (SPH) is a Lagrangian method used to study fluid flow in astrophysics (Monaghan, 1982; Monaghan, 1985; Monaghan, 1988; Monaghan, 1992), in particular compressible fluids such as stellar gases. SPH is based on the mathematical identity

$$A(r) = \int A(s)\delta(r-s)ds$$

given here in one dimension, where δ is the Dirac delta function defined to be zero for all non-zero values of $(r-s)$. Any field $A(r)$ can be approximated by replacing the Dirac function with an interpolating kernel $w(u,h)$ of compact support, such that in the limit as the "smoothing length" h goes to zero, the kernel equals the delta function. In three dimensions a discrete approximation of the continuous integral is given by representing the volume as particles, and summing the kernel weighted contributions from each particle. This formulation allows the density of the fluid at any point in space to be approximated as a weighted sum of particle masses. Forces on particles result from solving the gradients in pressure as functions of the density at each particle. Over a set of uniformly ordered particles, the SPH method is equivalent to finite-difference schemes, with the particular scheme dependent on the choice of interpolating kernel. Error is minimal when the particles are equally spaced and increases as the particles become disordered.

2.4.2 Particle Systems in Computer Graphics

In computer graphics, particle systems have been used to model visually complex natural phenomena such as fire, foliage, and waterfalls; to model and reconstruct both surfaces and volumes; and to emulate the physics of deformable, elastic, viscous, and solid materials. To aid in the review of related work, we categorize particle systems according to the interactions between particles. In *systems of independent particles*, the forces on each particle are independent of other particles in the system. *Particle systems with fixed connections* interact with neighboring particles where the set of interactions is constant after the initial specification. In particle systems with *spatially coupled particle interactions*, the interactions between particles evolve over time due to their relative spatial state. This results in the ability to model both varying geometry and topology as will be shown in the thesis.

Systems of independent particles have been used to model *visually complex* natural phenomena such as fire, smoke, foliage, and the spray of splashing water (Reeves, 1983b; Reeves and Blau, 1985; Sims, 1990; Stam and Fiume, 1993; Stam and Fiume, 1995; O'Brien and Hodgins, 1994; Goss, 1990; Sims, 1992). In these systems, 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[4] or the corresponding surface.

The previously mentioned deformable models of (Terzopoulos and Fleischer, 1988b; Haumann et al., 1991; Breen, House and Getto, 1991; Breen, House and Wozny, 1994) and the surface reconstruction model of (Miller et al., 1991) can be categorized as particle systems with fixed particle interactions. The forces felt by a particle are in part due to the fixed inter-particle connections and in part due to external forces. Shapes modeled by a system with fixed particle interactions can be deformed to change the surface geometry, but are limited to the structure imposed by the original connections.

In particle systems with spatially coupled particle interactions, the interactions between particles evolve over time — connections are automatically broken and *new* connections are automatically created. Replacing the fixed set of interactions with interactions that dynamically evolve creates a flexible modeling paradigm in which geometric and topological changes can occur as the underlying structure of the system changes. We briefly mention work cited early to place them in context of particle systems. The physically-based deformable volumes and fluids of (Miller and Pearce, 1989; Terzopoulos, Platt and Fleischer, 1989; Desbrun and Gascuel, 1995; Reynolds, 1997) are spatially coupled particle systems. The related volume models of Roy (Roy, 1995) and Desbrun and Gascuel (Desbrun and Gascuel, 1996) have applied Monaghan's SPH model (Monaghan, Thompson and Hourigan, 1994) of nearly incompressible fluids. To approximate surfaces, spatially coupled particle systems have been used to re-mesh polygonal models (Turk, 1992), to triangulate implicit surfaces (Witkin and Heckbert, 1994; Crossno and Angel, 1997) and variational surfaces (Welch and Witkin, 1994). Spatially coupled particle systems have also been used to distribute paint strokes for a painterly effect in rendering applications (Meier, 1996), and to grow cellular based textures (Fleischer et al., 1995).

2.4.3 Discussion

Sculpting with particle systems does not fall cleanly into previous modeling paradigms. It is neither a parametric representation, an implicit representation, nor a solid modeling primitive. However, we can construct such representations from our particle-based model. For the oriented particle system, a natural continuous surface representation is a triangulation interpolating the particle positions. From this, a subdivision surface (Loop, 1987) or smooth piecewise parametric representation (Loop, 1994) can be constructed. For the un-oriented particle system, an implicit surface is a natural

[4]An exception is (Stam and Fiume, 1993; Stam and Fiume, 1995) who use "blobs" to define a volume.

surface representation. As a sculpting metaphor, our work (Szeliski and Tonnesen, 1992) shares similarities to the recent work of (Welch and Witkin, 1994) and (Witkin and Heckbert, 1994) in that both techniques relying on a triangulation of a particle system to render the surface, though the origin of the surface is distinctly different. In (Welch and Witkin, 1994), particles are used not to define the surface, but to quickly approximate a variational surface. Instead of using a particle system as a basis for continual changes in topology, as this dissertation does, they add "just enough structure to the particle system to unambiguously fix its topology". Since their surfaces cannot be explicitly computed, they approximate them via a triangulation, using a particle system as a step in the triangulation process. In (Witkin and Heckbert, 1994) they also present a coupling between particles and an smooth surface, but in this case they use particles for both sampling and as control points of an implicit surface. The sampling of the surface using repulsion forces is similar to (Welch and Witkin, 1994) and (Turk, 1992). The computational expense of using particles as implicit surface control points requires over $O(N^3)$ time, where N is the number of control particles. We also allow particles to act as control points, but we have designed our system to execute in $O(N \log N)$ in the number of particles.

Lombardo and Puech (Lombardo and Puech, 1995) extended our oriented particles to endow objects with memory of their original shape. In this case, particles prefer a rest state matching originally specified curvature measures instead of a default zero curvature measure as our method does. This allows them to create oriented particle skeletons for modeling implicit surfaces with "shape memory".

In surface reconstruction, it is the dynamic nature of our model that is important to extracting the surface structure from a set of 3D data. By allowing our surfaces to extend out from known surface samples we can extend the range of surface reconstruction. For example, this will allow us to overcome some problems associated with the anisotropic sampling or random under-sampling in regions. In areas of high sampling density, the samples will be sufficient to reconstruct the correct surface, while the growth of new particles in low density sampled regions will approximate the under-sampled area. As adjacent surface patches meet the particles, patches will automatically join together, completing the interpolation. The energy based nature of the particle system allows us to optimally fit surfaces to data based on minimizing flatness and curvature functions. Our particles can be thought of as providing a finite difference solution to a minimization problem, similar to deformable models. The difference between this and an actual finite difference scheme is that our grid is not fixed, but dynamic in nature.

The nature of our particle system has several advantages for animation. The physical properties embedded in our model allow the animator to mimic a variety of common real-world materials, reducing the amount of animator effort required to create a sequence. Since the shape will be continually changing, we do not want to impose unnecessary constraints on the animator. We would also like to point out that animating a shape is closely related to sculpting a shape. In fact, animation could be considered a continuous sculpting exercise, with each new image representing a complete sculpture. We feel the flexibility our model provides for sculpting will be directly applicable to animation.

We apply particle systems to model volumes as do (Miller and Pearce, 1989; Terzopoulos, Platt and Fleischer, 1989), extending their attractive-repulsive inter-particle force model, based on the Lennard-Jones function, to create a thermoelastic model in which the stiffness varies as a function of thermal energy. This provides a mechanism by which the model can mimic the “melting” and “freezing” of objects. Two recent papers suggest alternative attractive-repulsive forces. Lombardo and Puech suggest a “cohesion” force which has a similar shape to the Lennard-Jones force that we use, and assert this force reduces oscillations, allowing the system to reach a rest state sooner. Reynolds (Reynolds, 1997) suggests using Boscovich’s law of force which has multiple minimal energy states, which he states is better for modeling inelastic deformations.

The recent work of Roy (1995) and Desbrun and Gascuel (1996) model fluids using the smoothed particle hydrodynamics model (Section 2.4.1). In the SPH model, forces between particle pairs are a result of gradients in pressure over the volume. As particles approach, the density and hence pressure increase, resulting in repulsive forces. As particles separate, the density and pressure becomes lower than the surrounding areas, resulting in attractive forces which equalize pressure. In particular, one must be careful that particles do not become too close with respect to the smoothing length; otherwise the particles tend to “clump” together in an unrealistic fashion. This is an artifact of the gradient of the kernel and is discouraged by

- adding a velocity-based damping term (Monaghan, Thompson and Hourigan, 1994) which is analogous to the ideal viscous unit (3.18) that this thesis employs for modeling visco-elastic materials,
- lowering the smoothing length (Roy, 1995),
- or defining a cusp-shaped smoothing kernel (Desbrun and Gascuel, 1996) such that the magnitude of the derivative increases rather than decreases when approaching an inter-particle separation distance of zero.

A cusp-shaped kernel results in a force curve that is similar in shape to the Lennard-Jones force curve (Desbrun and Gascuel, 1996)[Figure 3]. It differs in that the curve converges to a constant slope for small separations, like an ideal spring does, rather than to infinity as the Lennard-Jones model does. Thus the fluids SPH models are more compressible than fluids modeled using the Lennard-Jones function.

All of the spatially coupled particle volume models share features in common:

- Each particle represents a small volume element.
- The equations defined over pairs of particles result in attractive and repulsive forces in the direction of the vector separating neighboring particles.
- They can model elastic and visco-elastic materials.
- They are inadequate at *accurately* modeling incompressible materials, such as liquids, but they can model *nearly incompressible* fluids.

- Stiffer, less compressible materials require smaller time steps than more compressible materials.

In summary, our model is based on previous particle systems work in computer graphics and inspired by the physical sciences. The particles interact according to pairwise potential energy functions. These potential energies, inspired by physics and differential geometry, share similarities to the energy functions used for deformable models and variational surfaces. The self-organizing nature of our system is distinctly different than the traditional modeling techniques where one manually specifies the connectivity of surface patches, as is done in spline, polygonal, and variational based surface modeling. It allows for the ability to join and separate objects as do implicit surface modeling and CSG methods. While our model is a point based sampling description rather than a continuous description, we can generate full surface descriptions. The surfaces generated are implicit surfaces for volumetric samplings and triangulated polygonal models for surface samplings. The polygonal models can then be converted to smooth surfaces using either surface subdivision techniques or triangular based splines. Our model, like deformable models, allows us to create animations of visco-elastic materials, e.g. cloth. In addition our synthetic materials can be stretched, ripped, and joined back together automatically. Our model shares similarities to previous particle based volume models and recent fluid based particle models. We can also construct viewpoint invariant 3D surfaces that interpolate sparse point data and fit optimal surfaces to 3D volumetric data. Unlike object centered methods, the surfaces can be of arbitrary topology and genus without requiring prior assumptions of the surface structure.