Asian Journal of Information Technology 5 (3) : 324-338, 2006

Solid Deformation Modeling Techniques

Syaiful Nizam bin Yahya, Norhaida binti Mohd Suaib, Abdullah bin Bade
and Siti Mariyam Binti Hj Shamsuddin
Department of Computer Graphics and Multimedia, Faculty of Computer
Science and Information System, Universiti Teknologi Malaysia
81310 UTM Skudai, Johor, Malaysia

Abstract: Non-rigid solid object deformations techniques have been widely used in the computer graphics community to simulate and animate deformable objects. Both offline and real time applications have already benefited from deformation techniques evolutions. This paper discussed the most popular geometric deformation techniques used for both real time and offline applications such as virtual surgery and motion pictures. Deformations techniques are divided into two sub deformation type which is non-physical based deformations and physical based deformations. Computer Graphics, Physically Based Modeling, Animation and Virtual Reality.

Key words: Deformable object, physical based modeling, soft body, elastic, vicious

INTRODUCTION

Non-rigid objects modeling or deformable objects modeling have long been studied in the field of computer graphics. In 1987, Terzopaulos presents a deformable object modeling for virtual reality application in computer graphics. Since then, deformable object modeling have been actively studied producing various deformation modeling techniques for different types of deformable objects such as solid objects, fluid objects and gaseous objects. Different deformation features like fusion, brittle, fracture, cut, plasticity, cloth animation, fluid simulation and gaseous simulations further achieved by expanding deformation techniques, though are not fully covered by this discussion.

Chapter 2 describes the motivation for physical based deformation especially for solid volumetric objects. Chapter 3 discussed most popular deformation techniques which are divided into two subtopics, non-physical based deformation modeling and physical based deformation modeling.

MOTIVATIONS

In reality, no body is rigid, but for many bodies, the assumption of rigidity is a close approximation to the actual physical conditions. In some physical applications, the objects are considered to be deformable bodies, ones for which the rigid body analyses do not apply.

One of the holy grails of computer graphics is to accurately portray the real world based on physical principles using computer graphics technology. This includes the modeling of deformable object behaviors such as human tissue, gelatin, rubber, sea water, smoke and other non rigid materials. The problem of deforming various materials has been long studied in the field of computer graphics.

Expressing the physical behaviors of the deformable material is not trivial. First, one has to measure the elasticity of the real material. Achieving precise results is very hard. Even with 2 materials with the same mass and volume does not behave exactly the same. Based on this analysis, one has to express the continuum between the mass's atomic planes in order to achieve accurate results. Because of even a small mass consist of huge number of atoms, another approximation method to describe the expression is desired. This problem domain has been tackled in the field of physical engineering for quite some time now.

In computer graphics, the interest is to model the behavior of the deformable objects as a visual in the rendering device. Direct integration between physical principal of deformable material and computer graphics is a major problem due to the limitation of current computation capacity. The limited number of floating points available on the computer decreases the achievable accuracy. As the accuracy decreases, so is the stability and robustness of the systems. Certain application

Corresponding Author: Syaiful Nizam bin Yahya, Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia 81310 UTM Skudai, Johor, Malaysia

requires the systems to perform the simulation as a real time animation, thus limiting the available computation time for each frame. Since the computational time increases proportional to the increasing number of represented material elements, a direct solution would be to use much lower elements representation by simplifying the object. For some environment, for example virtual surgery, the systems cannot afford too much inaccuracy by object simplifying as the risk would be too high.

As technology matures, the demand for animation of complex behaviors grows. Deformable objects such as hair, cloth and gelatin are difficult to animate using traditional key framing as the nature of such system is very complex. The traditional key framing process is very tedious and does not guarantee accurate results. Physical based solution usually doesn't permit real time simulation as the computational cost is very high. Because of its high degree of accuracy, physical based solution often found its application in offline rendering application such as ray tracing packages where realistic results are desired. For real time animation, things are a bit different. In virtual reality field, user interactions are a requirement. The system must compute the material deformations, provide feedback to the user and perform other computations all in a limited time. Because of unpredictable nature of the simulation, assumption and pre-computation are not quite the solutions. Since the interest is real time simulation, the results usually behave a bit unrealistic as the system traded accuracy over speed.

For effective surgical simulation, things are even more difficult. Not only do we need real time interactive graphics, but the objects in the scene should also exhibit physically correct behaviors true to the behaviors of real human organ and tissues. The performance of a deformable modeling system often depends on multiple criteria. Tweaking and optimizing the deformation modeling system to achieve desired performance and accuracy provides a technical challenge in the field of computer graphics.

DEFORMATION MODELING

In this chapter, elementary theories and techniques that are relevant in volumetric object deformation are discussed. Literature coverage includes both non-physical based deformation modeling and physical based deformation modeling of deformable objects.

Non-physical based deformation modelling: 3d designer requires precise deformation tools which give them total deformation control. These tools usually come as purely geometric modification tools which doesn't have any physical justification in its deformation process. The output relies on the skill of the designer and how much control the deformation technique provides. Three most popular non-physical based modeling techniques are discussed which is global deformation, parametric representation and free form deformation.

Global deformations: In 1984, Barr introduced global deformation technique by extending the classical linear transformation operation[1]. The idea behind this method is to apply another transformation to existing transformation before it is applied to the object. The available deformations are tapering, twisting and bending. Given a formula definition for the transformations, $X = F_x(x)$, $Y = F_y$where (x, y, z) are vertex in undeformed state and (X, Y, Z) is the deformed vertex. To taper an object, choose a tapering axis and differentially scale the other two components setting up a tapering function along this axis. For example of tapering an object along its Z axis, X = rx, Y = ry, Z = z, Where is the tapering function either linear or non-linear. To globally twist the object, use differential rotation just as tapering is a differential scaling. To twist an object through an angle θ about the z-axis, we apply (X, Y, Z) = (x cosθ - Y sin θ, x sin θ + y cos θ, z) By varying the amount of rotation as a function of z, the object will become twisted. This Is done by setting θ = f (z) where f(z) specifies the rate of twist per unit length along the z axis. To bending an object along y axis, the deforming transformation is given by

$$X = x$$

$$Y = \begin{cases} -\sin\theta(z - k^{-1}) + y_0 & y_{\min} \le y \le y_{\max} \\ -\sin\theta(z - k^{-1}) + y_0 + \cos\theta(y - y_{\min}) & y < y_{\min} \\ -\sin\theta(z - k^{-1}) + y_0 + \cos\theta(y - y_{\max}) & y > y_{\max} \end{cases}$$

$$Z = \begin{cases} \cos\theta(z - k^{-1}) + k^{-1} & y_{\min} \le y \le y_{\max} \\ \cos\theta(z - k^{-1}) + k^{-1} + \sin\theta(y - y_{\min}) & y < y_{\min} \\ \cos\theta(z - k^{-1}) + k^{-1} + \sin\theta(y - y_{\max}) & y > y_{\max} \end{cases}$$

where $y_{min} \le y \le y_{max}$ is the bending region, K^{-1} is the radius of curvature of the bend, the center of the bend is at $y = y_0$, the bending angle is $\theta = K(y^1 - y_0)$ and

$$y' = \begin{cases} y_{\min} & y \le y_{\min} \\ y & y_{\min} < y < y_{\max} \\ y_{\max} & y \ge y_{\max} \end{cases}$$

Global deformation can be easily implemented into existing application since the deformation transformation

Fig. 1: Structures deforming global deformation example. Top, original cube and Utah teapot followed by tapering, twisting and bending deformations[2]

and classical transformation are similar in nature. The main set backs for this method is that the deformation is limited to the 3 previously mentioned particular types of deformation.

Parametric representations : By defining the object as parametric surfaces, users are given the ability to deform the surface by altering the functional description of the surface in the sense of displacing the control points. The first representational form or basis is due to Bézier, who was the originator of an early cad system, UNISURF used by Renault, a French car manufacturer.

Given a set of n + 1 control points P_0, P_1,..., P_n, the corresponding Bézier curve (or Bernstein-Bézier curve) is given by

$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t),$$

where is a Bernstein polynomial and. These functions are scaled or weighted by the network of control vertices, to form the surface patch. A cubic Bézier patch, an extension to the Bézier curve, is given by,

$$Q(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} P_{ij} B_i(u) B_j(v)$$

Bézier patch always passes through the first and last control points and lies within the convex hull of the control points. Undesirable properties of Bézier patch are their numerical instability for large numbers of control points and the fact that moving a single control point changes the global shape of the patch. The former is sometimes avoided by smoothly patching together low-order Bézier patch. The movements of the control points are constrained by continuity constraint between control points. These continuity constraints introduced two undesirable effects. First, undesirable plateau effect in the deformation is introduced if the deformation only displace the control points and not both control points and the continuity constraints. Second, it is impossible to achieve localize deformation since the continuity constraints may be propagate the patch further a field.

A generalization of the Bézier curve is the B-spline. As an improvement over the Bézier representation, B-spline are superior over the Bézier method within the context of deformation as B-Spline does not require continuity constraint and give the user the ability to perform localize deformation. Since the absence of continuity constraint, B-spline curve restricted the deformation by control points to only specific known region thus giving better control to the deformation made by the user.

Free form deformation: In 1986, Sederberg developed a technique that is more flexible than global deformation known as free form deformation[3]. This technique defines a free-form deformation of space by specifying a trivariate

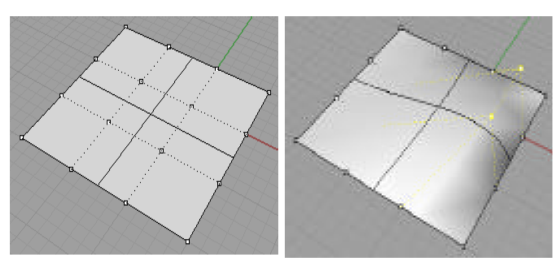

Fig. 2: Left; Original surface patch. Right; Deformed surface patch using Non-Uniform Rational B-Spline

Bézier solid, which acts on a parallelpiped region of space. Instead of deforming the object directly, this technique embeds the object in a defined space that is then deformed. The object is deformed according to the deformation the embedding space undergoes. The embedding space called FFD block, is hyperpatches connected together to form a piecewise Bézier volume. A single tricubic Bézier hyperpatch is defined as

$$q(u,v,w) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} p_{ijk} B_i(u) B_j(v) B_k(w)$$

where $B_i(\mu)$, $B_j(v)$ and $B_k(w)$ are the Bernstein polynomials of degree 3. The undeformed FFD block consist of a rectangular lattices of control points arranged along three mutual perpendicular axes. The end result is a parallelepiped with lattices as control points attached. To deform object using free form deformation method, first we must determine the positions of the vertices in lattice space. Then deform the FFD block by displacing the control points from the undeform lattice positions. Finally, determine the deformed positions of the vertices by finding the relevant hyperpatch within which the vertex is located and convert to the local coordinate system of the hyperpatch.

This method can be used to apply localized deformation or to deform the whole object. Multiple FFD

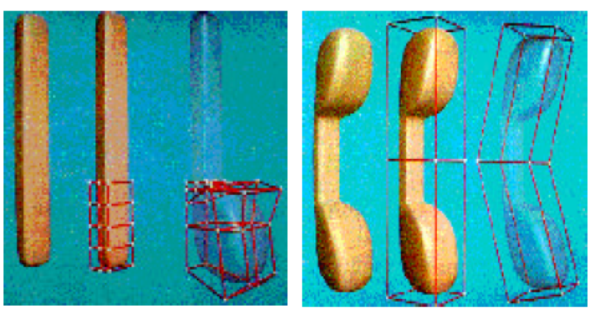

Fig. 3: Right, local free form deformation. Left, global free form deformation[3]

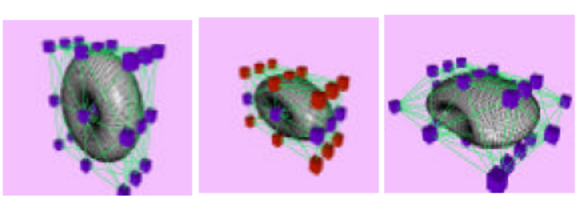

Fig. 4: Hirota's volume preserving method. Letf, original shape. Center, after free form deformation is applied. Right, unconstrained lattices are displaced to preserve original volume[5]

block can be define in piecewise manner to perform deformation that is impossible using just a single FFD. For modeling complex deformation and specific small region of deformation, careful placement of FFD block by the user is required. But when it does, the large number of FFD blocks would render the technique inefficient.

Unlike free form deformation by Sederberg, Coquillart's extended free form deformation does not define any specific FFD lattice space[4]. Coquilart states that parallelepiped shaped FFD block constraint the shape of the deformation and introduced nonparallelepiped lattices as the EFFD lattice space. To construct the EFFD block, the users are required to weld several elementary blocks, which is the classic FFD blocks, together. As with FFD, to perform deformation, EFFD lattices have to be displace. The deformation processing is very similar to that of previously discussed FFD except that unlike FFD, in EFFD, we cannot assume simple connection between the 2 adjacent spaces because lattice space of EFFD does not aligned with EFFD object space.

To preserve the total volume of solids undergoing free form deformation, Hirota uses discrete level-of detail representations[5]. Given the boundary representation of a solid and user-specified deformation, the algorithm computes the new node positions of the deformation lattice, while minimizing the elastic energy subject to the volume-preserving criterion. During iterations, a non-linear optimizer computes the volume deviation and its derivatives based on a triangular approximation, which requires a finely tessellated mesh to achieve the desired accuracy. To reduce the computational cost, Hirota exploit the multi-level representations of the boundary. This technique also provides interactive response by progressively refining the solution. Furthermore, it is generally applicable to lattice-based free-form deformation and its variants. This method is capable of large deformation, efficiently. It gives designers and engineers real-time visual feedback and an intuitive physical feel of free-form solids, during geometric design and shape modification.

Exact shape and point placement is difficult to achieve with traditional free form deformations. This is due to the free form deformation interface which permits the users to deform using only control points. Hsu *et al.*, introduced a free form deformation method that allows user to control a free form deformation of an object by manipulating the object directly instead of control points [6]. The method computes the necessary alteration to the control points of the free form deformation spline using least square approach that will induce the point's placements.

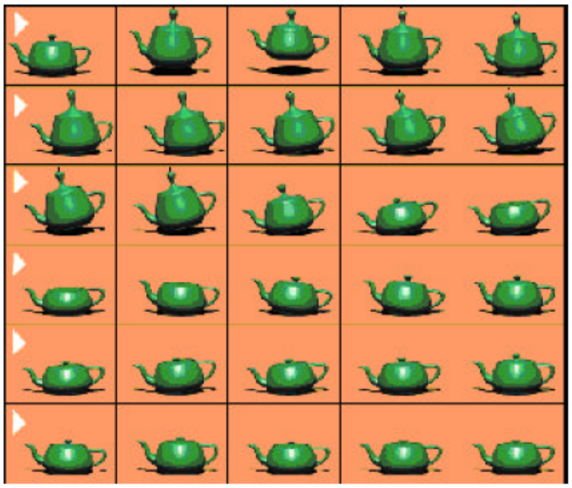

Fig. 5: Deformable teapot is animated using dynamic global free form deformation[7]

Faloutsos *et al.,* extends the use of free form deformation to a dynamic setting by coupling physical dynamics with free form deformation[7]. The method is based on parameterized hierarchical FFDs augmented with Lagrangian dynamics, provides an efficient way to animate and control the simulated characters. Objects are assigned mass distributions and elastic deformation properties, which allow them to translate, rotate, and deform according to internal and external forces. First, the dynamics generalization of conventional geometric free form deformation is formulated. The formulation employs deformation modes which are tailored by the user and are expressed in terms of free form deformations. Second, the formulation accommodates a hierarchy of dynamic free form deformations that can be used to model local as well as global deformations. Third, the deformation modes can be active, thereby producing locomotion.

Physical based deformation modelling: Physical base modeling uses physical principles to model realistic behavior of deformable models. This method uses more computational power than non-physical based method but the results is more convincing than the non-physical based method. Integration between physical principle and computer graphics for deformable object modeling was pioneered by Terzopoulos[8-10]. Two most common and well known physical based methods are finite element method and mass spring method. On the other hand, two of the most recently proposed methods for physical based modeling are known as mesh free method and gas based method. Here, basic physical based method for deformable object is discussed along with each method subsequent extension techniques.

Finite element method : The behavior of deforming objects is the topic of continuum mechanics, a branch of mathematics that tries to capture physical phenomena of continuous media in precise mathematical formulations. One branch of continuum mechanics, nonlinear elasticity, provides the mathematical description of how objects deform.

Continuum mechanics describes materials in terms of partial differential equations. The Finite Element Method (FEM) is a discretization method. It transforms a continuous, infinite-dimensional problem into systems of equations with a finite number of variables. For mechanical problems, the FEM discretizes the equations of motion; hence it delivers a system of ordinary differential equations, i.e., equations where time still has a role. There are two ways to deal with these systems: compute the evolution of the system, or try to find the final equilibrium solution directly. If the final state of the system is all that matters, a static method can be used. By assuming that velocity and acceleration are null, the system of differential equations is changed into a normal system of equations. For many mechanical problems, these equations can be stated in terms of finding minimum energy solutions. If transient effects do matter, then the evolution of the differential equations must be calculated using a time-integration method. Basically, the problems come from the simulation of soft tissue. Although simulating the full mechanical characteristics of soft tissue is not possible in an interactive setting, it is instructive to study exactly what kinds of characteristics are ignored in the simulations. It is not surprising when most implementation tends towards simplifications since the constraints of an interactive simulation do not allow for much sophistication.

To sum it up, the finite element method finds an approximation for a continuous function that satisfies an equilibrium condition which follows from the variation or weak formulation of the problem. The discretization of the problem consists of decomposing its domain into a mesh of carefully selected elements, joined at discrete nodes. The solution of the variational equation is expanded as a weighted sum of finite element basis or shape function on each element. Continuity across element boundaries is achieved by sharing discrete nodes and thus finite element weights. As a next step, the contributions of each element are assembled into a global system of equations which then can be solved for the shape function weights.

To analyze the stress in various elastic bodies, calculate the strain energy of the body in terms of nodal displacements and then minimize the strain energy with respect to these parameters-a technique known as the Rayleigh-Ritz. In fact, this leads to the same algebraic

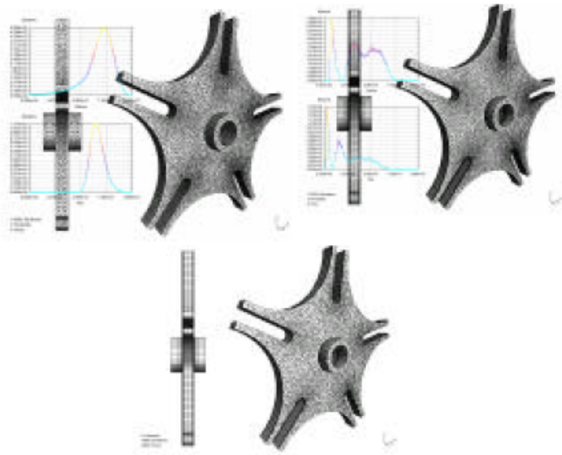

Fig. 6: Three type of geometry discretization using gmesh[11]

Fig. 7: Original happy buddha and its sliced tetraheralized version. Happy buddha is discretized using tetgen[12]

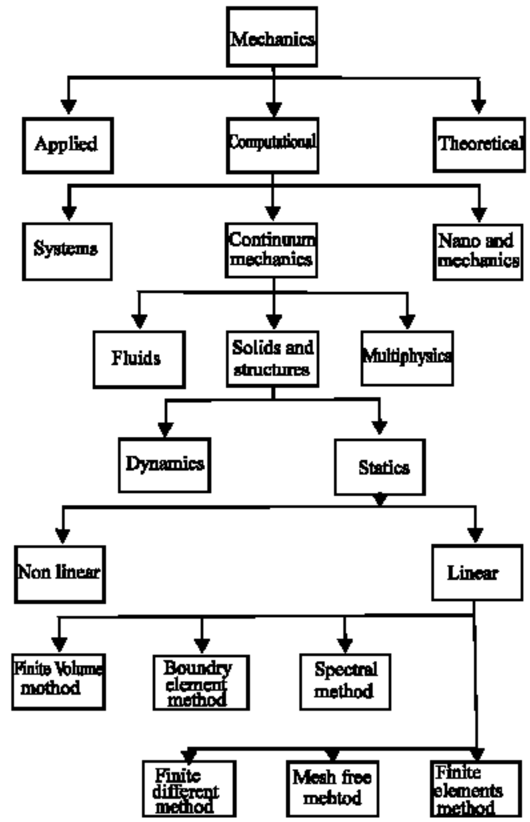

Fig. 8: Taxonomy for finite element method from mechanical physics view. [http://caswww.colorado.edu/courses.d/AFEM.d/Home.html]

equations as would be obtained by the Galerkin method but the physical assumptions made (in neglecting certain strain energy terms) are exposed more clearly in the Rayleigh-Ritz method.
In all cases, the finite elements steps are:

- Evaluate the components of strain in terms of nodal displacements.
- Evaluate the components of stress from strain using the elastic material constants.
- Evaluate the strain energy for each element by integrating the products of stress and strain components over the element volume.
- Evaluate the potential energy from the sum of total strain energy for all elements together with the work done by applied boundary forces.
- Apply the boundary conditions, e.g., by fixing nodal displacements.
- Minimize the potential energy with respect to the unconstrained nodal displacements.
- Solve the resulting system of equations for the unconstrained nodal displacements.
- Evaluate the stresses and strains using the nodal displacements and element basis functions.
- Evaluate the boundary reaction forces (or moments) at the nodes where displacement is constrained.

Solid elements are three-dimensional finite elements that can model solid bodies and structures without any a priori geometric simplification. Finite element models of this type have the advantage of directness. Geometric and constitutive assumptions required to effect dimensionality reduction, for example to planar or axisymmetric behavior, are avoided. Boundary conditions can be more realistically treated.

Another attractive feature is that the finite element mesh visually looks like the physical system. This directness does not come for free. It is paid in terms of modeling, mesh preparation, computing and post-processing effort. To keep these within reasonable

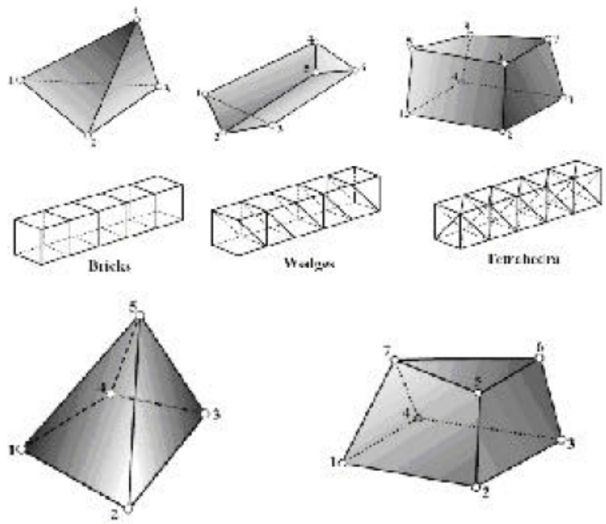

Fig. 9: Top,the three standard solid element geometries: tetrahedron (left), wedge (center) and brick (right). Only elements with corner nodes are shown. Middle, regular 3D meshes can be built with cube-like repeating mesh units. Meshes are built with bricks, wedges or tetrahedra. Bottom, two nonstandard solid element geometries: pyramid and wrick [w(edge)+(b)rick]. Four faces meet at corners 5 and 7, leading to a singular metric. [http://caswww.colorado.edu/courses.d/AFEM.d/Home.html

limits it may be necessary to use coarser meshes than with two dimensional models, which in turn may degrade accuracy. Thus finite element users should not automatically look upon solid elements as snake oil. Its use should be restricted to problems and analyses stages, such as verification, where the generality and flexibility of full 3D models is warranted.

Two dimensional (2D) finite elements have two standard geometries: quadrilateral and triangle. All other geometric configurations, such as polygons with five or more sides, are classified as nonstandard or special. Three dimensional (3D) finite elements offer more variety. There are three standard geometries: the tetrahedron, the wedge, and the hexahedron or brick. These have 4, 6 and 8 corners, respectively, with three faces meeting at each corner. These elements can be used to build topologically regular meshes. There are two nonstandard geometries that deserve consideration as they are occasionally useful to complete generated 3D meshes: the pyramid and the wrick. (The latter term is a contraction of wedge and brick) These have 5 and 7 corners, respectively. One of the corners is special in that four faces meet, which leads to a singular metric there. This singularity disqualifies these

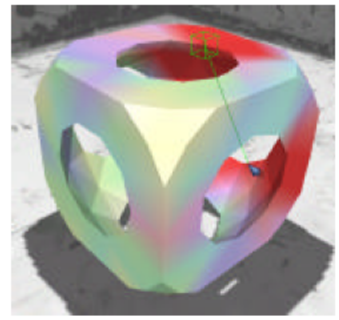

Fig. 10: A simple finite element method deformable object in action. Image is taken from project Xplodar[http://nesnausk.org/nearaz/projXplodar.html]

elements for use in stress analysis in highly stressed regions. However they may be acceptable away from such regions and in vibration analysis. Both standard and nonstandard elements can be refined with additional mid side nodes. These refined elements are of interest for more accurate stress analysis. Of course, the mid side nodes may be moved away from the midpoints to fit curved geometries better. The best choices of elements and interpolation functions depend on the object shape, convergence requirements, degree of freedom, and trade-offs between accuracy and computational requirements. In general, using elements that have more nodes and more complex interpolation functions require fewer elements for the same degree of accuracy.

Consider isoparametric solid elements with three translational degrees of freedom (DOF) per node. Much of the development of such elements can be carried out assuming an arbitrary number of nodes n. In fact a general template module can be written to form the element stiffness matrix and mass matrix. Nodal quantities will be identified by the node subscript. Thus $\{x_i , y_i , z_i \}$ denote the node coordinates of the i^{th} node, while $\{u_{xi} , u_{yi} , u_{zi} \}$ are the nodal displacement DOFs. The shape function for the ith node is denoted by N_i. These are expressed in term of natural coordinates which vary from element to element.

High contrast red denotes high stress area while bright white denotes less stress area. Even though the simulation is performed in real time manner, notice that the deformable object is low in polygons.

Forces must be numerically integrated over volume or surface at each timestep, requiring a lot of computation. This makes finite element method limited use in real time application despite the fact that finite element method provides better deformation accuracy. Because of its complexity nature, it is difficult to implements and optimize. Discretizing the object is also quite difficult. Discretization methods chose for real time applications are based on the ability of the discretizer to maintain high geometrical accuracy with less internal elements using single simple element type (usually tetrahedron). Large deformation and topological changes requires the system to recompute the large stiffness matrix. Finite element method requires less node points compared to mass spring systems to achieve similar degree of deformation accuracy. This results to a smaller linear system which can be solved in less time.

Terzopoulos used finite element modeling technique to discretize the deformable objects for its offline simulator[8]. The idea is to model deformable objects using differential equation analogous to the standard mass-spring-damper equation. Dynamics are computed from the potential energy stored in the elastically deformed body using finite difference discretization method. Later on, Terzopoulos extends the work to include simulation of inelastic object behaviour such as plasticity, fracture[9], heating and melting[10] .

Nielson and Cotin achieve real time finite element method deformation by lots of preprocessing and equation systems condensation[13]. By solving a smaller linear system, the implemented systems achieve 20 frames per second for models with 250 nodes on four Mips R4400 processor Silicon Graphics ONYX.

Although fast finite element models have been developed for medical applications[13], less attention has been paid to displaying time dependent deformations of large size finite elements models in real-time.[14] introduces two numerically fast techniques for real-time simulation of dynamically deformable (i.e. time dependent deformations) 3D objects modeled by FEM; modal analysis and spectral Lanczos Decomposition.

Existing techniques of deformable modeling for real time simulation have either used approximate methods that are not physically accurate or linear methods that do not produce reasonable global behavior. Nonlinear finite element methods (FEM) are globally accurate, but conventional FEM is not real time[15] apply nonlinear FEM using mass lumping to produce a diagonal mass matrix that allows real time computation. They propose a scheme for mesh adaptation based on an extension of the progressive mesh concept, called dynamic progressive meshes to minimize unnecessary computations.

Krysl et al. uses adaptive local finite element mesh refinement using wavelet theory to accelerate finite element deformation[16]. The refined mesh is nested in the refinement hierarchy, which simplifies the incorporation of multi-grid solvers. The method exploits refinement of basis functions rather than refinement of elements. It is in spirit much closer to some recent developments in the design of meshless methods. It is suitable in any number of spatial dimensions and for a much wider variety of finite element types than any standard mesh refinement algorithm.

Finite elements method benefits from a solid background and established technique, books and vast literature. For computer applications, there are a lot of libraries for solving finite elements. Applications to discretize geometry object into sets of elements are also widely available. Compared to mass spring method, integrating actual tissue properties are easier with finite element method. Solutions for large linear or non linear systems using numerical techniques already exist. With constraint, some assumption and optimization, real time computation is possible with current mainstream hardware. Finite element method allows parallel computing techniques for its simulation; enabling scalable simulations.

Finite element method is not without it drawbacks. Simulation time is slow even for linear elasticity deformation. For non linear deformation, it is even slower. To permit real time performance, multiple accelerating strategies should be implemented. For medical application, some real time accelerating strategies are not applicable due to limited allowable deformations and inaccuracy introduced. Finite element system is very complex and it is not that easy to implement.

Mass spring method: Mass spring method is one of the physical based methods that have been extensively used in the field of real time deformable object modeling. The surface or volume is discretize into a set of mass points. Each mass point is linked to its neighbors by one dimensional spring. Deformation is computed by finding equilibrium state between interconnected points after external force is applied. The spring is often linear, but non-linear elasticity can be simulated by applying multi-varied stiffness springs. Mass spring systems can also modeled as either static or dynamic (where time has influence) system.

There are multiple ways to construct the mass spring lattices. One can construct the springs manually or discretize the object into sets of tetrahedrons[17,18]or cubes. Acquired geometry topology (tetrahedrons or cubes) are represented as configuration of point masses connected by springs.

Basically, as spring experience external forces, the spring is either compress or extends to the direction of the force and creates a repulsive force oppose to the direction of the force. The created force is described mathematically by

$$F = -k * \Delta x$$
$$\Delta x = x_c - x_i$$

where F is the resultant force, k is the spring coefficient, and the distance between the two points (x_c = current distanceΔx and x_i = distance at the inertial position). Inertial position is the distance between two separated points. No force will be generated if the points are not displaced. If the spring is compressed, then will be negative, generating a positive force (expansion). If the spring is expanded, then will be positive, generating a negative force (compression). Elasticity coefficient is represented by k. Also known as Young's modulus, one dimensional deformation coefficient weights the spring final force. Stiffer spring have bigger k as it creates a larger force from its inertial state. Conversely, a spring with a smaller k is more flexible because it creates a smaller force from its inertial state.

To compute the distance between two points, one an use Pythagoras' theorem. Then, multiply with k coefficient and finally use the inverse of this value to compute the force. Spring force alone does not enough for most simulation. Other forces can be applied into the system such as damping force. This is to simulate the energy loss experience by the springs. This results into an extended equation

$$F = -kx - bv$$

where b is the coefficient of damping and v is the relative velocity between the two connected points.

For a networked configuration of mass spring lattices, when a spring is displaced, the resultant force propagates throughout the entire network. This results into deformable object behaviors. Based on this phenomena,

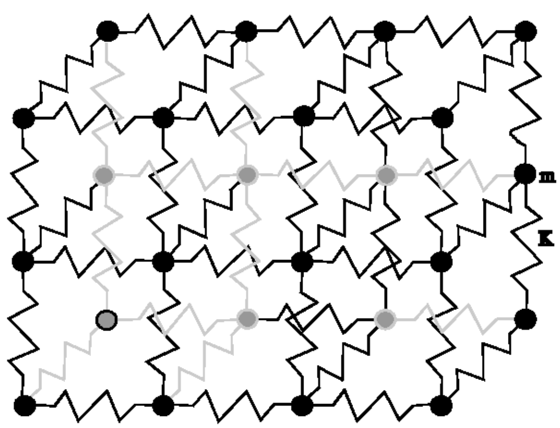

Fig. 11: An example of mass spring model. Connected spring exerted forces on neighboring points, displacing the points from its rest position[19]

mass spring are used in modeling string, cloth, jelly, face, human tissue and various other. The difference between these applications is the initial spring configurations.

In a dynamic three dimensional deformable system, where time is integrated into the system, the mass m_j at position $x_i \in \Re^3$at time t are governed by Newton's second law of motion

$$m_i \ddot{x}_i(t) + \gamma_i \dot{x}_i(t) + f_i^{int}(t) = -f_i^{ext}(t)$$

where y denotes a damping factor, $f_i^{int}(t)$ refers to the internal forces resulting from spring interconnection and $f_i^{ext}(t)$ represents the sum of external forces applied by the user or due to gravity or collision. The equations of motion for the entire system result from assembling the equations of all masses m_i in the lattice. Writing the positions of all m masses component-wise into a position vector x of size 3n, we can state a matrix equation for the entire mass-spring system as

$$M\ddot{x} + D\dot{x} + Kx = -f$$

where M, D and K are 3n 3n matrices representing mass, damping and stiffness, respectively. Although possibly large, these matrices are very sparse. M and D are diagonal, where K in a regular lattice is banded according to adjacency between masses. The equation is reduced in to two coupled systems of first order differential equations to numerically integrated through time as

$$\dot{x} = v$$

$$\dot{v} = M^{-1}(-Dv - Kx - f)$$

The problem of solving large and complex networked configuration of mass spring lattices calls for numerical integrators. There are lots of numerical integrator techniques available, but 4 most popular integrators are Euler, Midpoint, Runge-Kutta and Verlet. These integrators vary in its accuracy and computational cost. The fastest one but with less accurate are Euler integrator and the most accurate integrator but slow to compute is Runge-Kutta. Verlet integrator, on the other hand, is both fast and accurate integrator compared to other integrators. Accuracy is important to maintain simulation robustness. Although, all integrators accumulate errors at each time-step, the highest accuracy integrators will maintain the simulation stable for a longer time. Inaccuracy also leads to instability, where the simulation will explode and turn to chaos.

Chadwick *et al.,* coupled multi layered mass spring system with free form deformation for its computer animation system[20]. The method allows for global and local deformation of articulated character. Teschner *et al.,* approximate the object's shape into uniform tetrahedral meshes of free form deformation constraint[17]. Physical based deformation is applied to the tetrahedral meshes using mass spring techniques where the mass spring system will deform the free form deformation control points. Deformed free form deformation control points will then deform the underlying vertices. To preserve volume undergoing deformation, volume and surface preserving coefficient is introduced to the mass spring system. This two fold deformation method which coupled mass spring system and free form deformation allows for high geometry deformation as the rendering geometry and deformation geometry are independent of each other. Other hybrid method of mass spring systems is by Christensen *et al.,*[2] where the deformable object is approximately wrapped with simple mass spring lattice configuration. Then physical based deformation is applied to the mass spring where the lattice configuration will act as free form deformation constraint to the actual object geometry. This method is used for animating characters in 3d animation. Cotin et al. combined finite element method and mass spring system for virtual surgery application[22]. Finite element method is used to model tissue deformation using pre-computed deformations allowing large deformation. To enable volume cutting and topological changes to the tissue, a mass spring model variant called tensor mass model is applied into the system.

Baraff et al introduced implicit integration for its mass spring cloth simulations[1]. By using implicit integration, the system is much more stable and independent from number of particles used. Fuhrmann et al. describe and algorithm which replaces the internal cloth forces by several constraints and therefore can easily take large time steps[23]. Instability, inaccuracy and speed problem for numerical integration can be minimized by using Verlet integrator. Jacobsen uses velocity less Verlet integration for its real time physic systems[24]. Teschner et al. have perform a little experiment on various integrators to find the fastest integrator and have proved that Verlet integrator is the best numerical integrator suitable for mass-spring systems period[12].

Mass spring systems are easier to implement than finite element method. Computation cost for mass spring systems are much lower compared to finite element method thus mass spring systems have much wider appeal for real time applications. Non linear deformable object can also be modeled by mass spring systems. In addition, mass spring systems are suitable for parallel processing allowing a scalable simulation platform.

Since mass spring systems rely on numerical integrators, the systems are vulnerable to convergence and instability. The principle of mass spring systems defined that force travels according to the spring's links, not by continuum. This physical approximation is too coarse to be applicable for some critical applications. Certain applications requirement such as specific constraint and materials properties cannot be modeled with mass spring systems. Behavior of incompressible materials and thin object are unpredictable if modeled using mass spring systems. It is hard to model material stiffness by setting spring coefficient parameter. Sometime the deformation acts differently than desired behavior. Even after successfully tuning the spring coefficient, other coefficient, for example gravity, when changed, the spring coefficient have to be tuned all over again.

Gas pressure method: Matyka and Ollila proposed a novel technique for modeling elastic soft body object[25]. Soft body is described as three dimensional deformable meshes which always keep constant volume. The method is based on simple thermodynamics laws and uses the Lausius-Clapeyron state equation for pressure calculation. The pressure force is accumulated into a force accumulator of a 3d mesh object by using mass spring technique. Behavior of soft body is obtained after the integration of Newton's second law of motion with fixed or non-fixed air pressure inside of it. Simply put, the idea is to create a closed mass spring cloth represented as manifold mesh object and put air pressure inside it.

To enable simple pressure formulation, Matyka uses ideal gas approximation which is defined as one in which all collisions between atoms or molecules are perfectly elastic and in which there are no intermolecular attractive forces. One can visualize it as a collection of perfectly hard spheres which collide but which otherwise do not

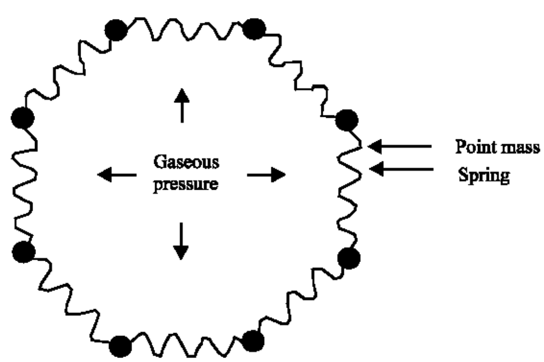

Fig. 12: Example of gaseous pressure method for simple two dimensional meshes. The mesh must be manifold, represented as wrapped cloth which will have ideal gas pressure inside[25]

interact with each other. In such a gas, all the internal energy is in the form of kinetic energy and any change in internal energy is accompanied by a change in temperature.

An ideal gas can be characterized by three state variables: absolute pressure P, volume V, and absolute temperature T. The relationship between them may be deduced from kinetic theory and represented by

$$P = \frac{nRT}{V}$$

where n is the number of moles and R is universal gas constant . To calculate pressure for the point of the shape, the expression used is

$$\vec{P} = P \cdot \hat{n} \left[\frac{N}{m^2} \right]$$

Next, the volume of the deformed body has to be recalculated to measure the gas pressure inside the object. Matyka uses simple bounding geometry such as sphere, box and ellipses to approximate the current volume. A better volume computation method is presented by Owen using Gauss's Theorem. Gauss's Theorem relates the divergence of a vector field within a volume to the flux of a vector field through a closed surface by the following where the surface s encloses the volume v. Detail theory and implementation are available at[20].

Deformation based on ideal gas pressure method does proved to be fast(able to perform real time deformation with coupled thousand of vertices)[25]. The method is simple to implement and requires no extensive geometry discretization preprocessing (unlike finite element method). Since its volume dynamic is represented as simple ideal gas equation, it does not exhibit complex internal volume structure like volumetric mass spring method and finite element method to compute internal dynamics. Finite element method and volumetric mass spring stored invisible internal geometry topology data for dynamics processing while gas pressure method only store visible surface geometry topology data which means less memory footprint.

Albeit all gas pressure method strengths, it's not without weaknesses. It is very hard to define the deformation coefficient (Young's modulus and pressure coefficient) to model desired material. Deformation behavior looks like a balloon filled with water placed underwater. From the available demo, it doesn't look like a balloon filled with gas at all. Since it uses mass spring technique which consist of numerical integration, gas pressure method inherit mass spring drawback which

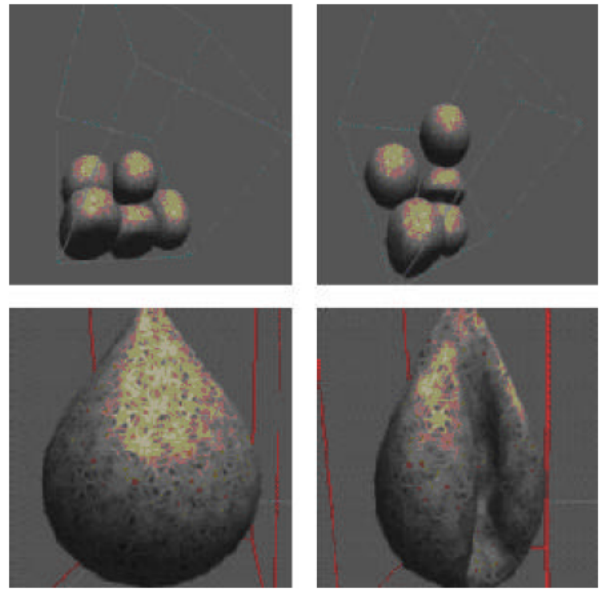

Fig. 13: Screen shot of of gas pressure method for three dimensional volumetric deformable objects[25]. The simulation is fast enough to be performed in real time

is numerical integration accuracy and stability. The deformation is prone to explode if it undergoes huge deformations.

Mesh free method: Numerical methods like Finite Elements, Finite Volumes and Finite Differences are already very well developed. However, there are limitations to these methods. First of all the time an engineer spends on solving a problem, goes mainly into the meshing of his solution domain. Secondly, the mesh is sensitive to large deformations, which can cause accuracy deterioration. To circumvent the meshing as a whole and make the problem more flexible, the so-called mesh free methods are invented.

To give some applications of this method, first the differences between the mesh free methods and the other methods should be clear. Instead of using a pre-defined mesh, mesh free methods only use node generation (giving the points without the need to prescribe the relationship between the nodes) and for each node a shape function is created. Since the mesh less method does not describe point topology explicitly, neighbor search is fundamental in finding the equilibrium state of the deformed object. The lack of topology structure and the ability of the system to self organize provides a system that is able to simulate a wider range of deformable material compared to commonly used deformation technique. The next step is to form a system of equations and solve this system.

Fig. 14: Rendering techniques for particle based surface; axes, discs, wireframe triangularion and flat shaded triangulation[2]

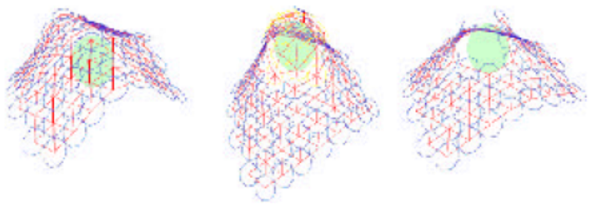

Fig. 15: Left, deforming. Center, deforming and surface restructuring by adding new points. Right, deforming and tearing.[27]

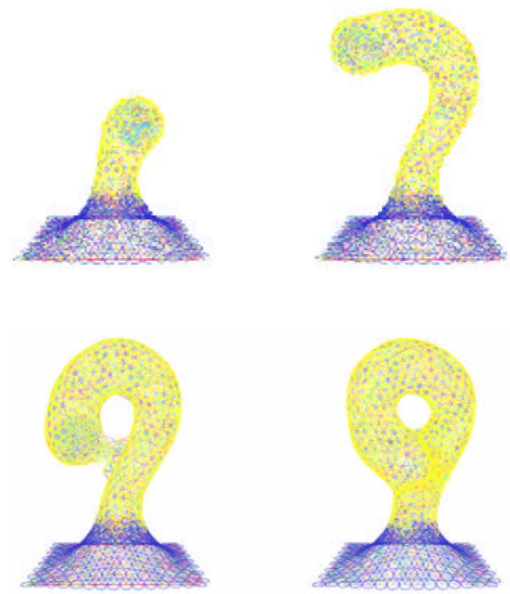

Fig. 16: Fusioning deformable objects[27]

Common geometric representations approximate the body by a mesh of nodes of fixed topology which are not adapted to the animation of substances undergoing large inelastic deformations. In this case, the use of mesh less method for object representation and dynamic representation is more appropriate. These systems are unstructured in the sense that interactions between point masses do not depend on a specified graph of connections, but on distance. The need to simulate various complex deformation types such as melting, solidifying, splitting and fusion motivated the use of mesh less method in modeling deformable objects in the field of computer graphics.

To derive Inter-point forces, Tonnesen used the pair-wise Lennard-Jones potential energy functions as a dynamics system solution[27]. To enable stretching and growing, Tonnesen introduced orientation to the point's properties. Under large deformation, Tonnesen proposed a kd-tree hierarchical data structuring approach to compute forces and torques at reduced number of points. By spatially subdivide the object space within some radius (natural inter-points spacing), all to be deform neighbor points can be efficiently found. To further reduce the computation, this operation is occasionally performed and cache list of neighbors were used for intermediate time steps. New points were added when neighboring points have large enough space between them and still under maximum number of allowable points between the ranges.

Each point is given state variables of position and mass for the system to interact with the dynamics. For more complex systems, additional state variables combined with simple heuristics were formed to create application specific behaviors. The surface is rendered as iso surface which yield an implicit coating of the point which handles topological changes such as splitting and merging by construction.

The Lennard-Jones potential is well known in molecular dynamics for modeling the interaction potential between pairs of atoms. It creates long-range attractive and short-range repulsive forces, yielding particles arranged into hexagonally ordered 2D layers in absence of external forces. Increasing the dissociation energy (magnitude of the potential energy) increases the stiffness of the model, while the width of the potential energy can be varied. Therefore, large dissociation energy and high potential energy exponents yield rigid and brittle material, while low dissociation energy and small potential energy exponents result in soft and elastic behavior of the object. This allows the modeling of a wide variety of physical properties ranging from stiff to fluid-like behavior. By coupling the dissociation energy with thermal energy such that the total system energy is conserved, objects can be melted and frozen. Furthermore, thermal expansion and contraction can be simulated by adapting the equilibrium separation distance to the temperature.

Desbrun and Cani[28-30] use smoothed particle hydrodynamics approach used by physicists for cosmological fluid simulation as its deformable dynamics basis. The Smoothed Particle Hydrodynamics (SPH) formalism was introduced by physicists for accurate simulation of fluid dynamics. Simulating a fluid consists in computing the variations of continuous functions such

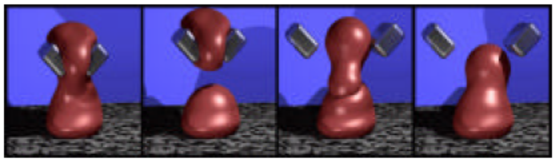

Fig. 17: Deformable object are splitted and then fused together[29]

Fig. 18: Target morph using point based method[31]

as mass density, speed, pressure, or temperature over space and time. Standard finite element techniques in hydrodynamics use an Eulerian approach: they consist of dividing space into a fixed grid of voxels, and then studying what flows in or out of each voxel. However, this kind of approach requires the division of huge empty volumes and is not intuitive for flows.

SPH belongs to an alternative approach, called the Lagrangian approach that consists of following the evolution of selected fluid elements over space and time. The particles can be viewed either as matter elements or sample points scattered in a soft substance. Each of them represents a small volume of inelastic material that moves over time. In practice, smoothed particles are used to approximate the values and derivatives of continuous physical quantities, such as local mass density or pressure that need to be computed during the simulation. Smoothed particles ensure valid and stable simulation of a state equation describing the physical behaviors of the material. It is also used for deforming the surface of the substance in a coherent way using the level sets of the mass density function. To reduce computation time, adaptive time steps for integration is used according to a local stability criterion along with efficient data structure for neighbor search. Desbrun further the research for rendering the point particles using implicit surface rendering method[29].

Using mesh less method, dubbed point based method; Keiser et al. were able to simulate wide range of material properties such as stiff elastic to highly plastic using a single application framework[31]. By using points for both volume and surface representation, arbitrarily large deviations from the original shape can be simulated. In contrast to previous mesh less based elasticity in computer graphics, the physical model is derived from continuum mechanics, which allows the specification of common material properties such as Young's Modulus and Poisson's Ratio.

In each step, spatial derivatives of the discrete displacement field were computer using a Moving Least Squares (MLS) procedure. It is from these derivatives that strains, stresses and elastic forces at each simulated points were obtain. Equations of motion for these forces were solved using both implicit and explicit integration. Point sampled surface were rendered dynamically adaptive for scalable and faster performance. Although material anisotropy can be simulated, only linear elasticity are implemented in the dynamic system. MLS only works if there are at least 3 neighboring points within non-degenerate locations. This makes it only suitable for volumetric objects, not two dimensional or one dimensional object. The nature of the system is close proximity points always interact with each other. This makes it difficult to model fracture and brittle materials. Even with stiff coefficient, hard edges are difficult to achieve.

Deformable object ranging from stiff elastic to highly inelastic objects can be modeled efficiently using mesh less method due to its natural properties of not having topological properties explicitly. Surfaces are easy to shape, extend, fusion and split. Material properties such as stretching, bending or variation in curvature can be controlled by adjusting strength of various potential energy functions. Input model doesn't have to be discretize into elements which is a requirement for finite element method.

Mesh free method application in computer graphics deformable object simulation is quite new. The first idea implementation was seen in 1995[28]. With this method, objects are easy to deform and new deformed shape are easy to construct for the purpose of rendering (no topology needed). Material stiffness and other properties such as resistance to stretching, bending can be controlled by adjusting strength of various potential energy functions.

One problem of mesh free method is that the surface is not explicitly defined thus poses a problem rendering the points. The points cannot be rendered using trivial geometry rendering technique. It is harder to achieve exact control of the shape. Usually, sampled points are shape approximation of the original object shape. Hard edges are also hard to preserve during point sampling of the object. Accurate dynamic computation is expensive. To enable real time performance, implementation must include heavy optimization. The lack of precise control and shape degeneration due to point sampling makes it unsuitable for engineering purpose.

CONCLUSIONS

The main strength in parametric representation based surface deformation is it maintains object smoothness under any deformation complexity. Users are given total deformation control up to the control point complexity level. Because of this features, parametric based surface deformation are widely used in computer aided design and model editing application.

Parametric based surface deformation is not without its limitation. Since the object representations are defined as sets of parametric surfaces, the deformation detail level depends on the quantity of the control points. It is impossible to apply localized deformation in between control points. Remeshing the parametric surfaces introduced aliasing that may not accurately reflect the deformation the user intended to make because of continuity of the constraints. It is difficult to represent object parametrically especially for objects possessing complicated topology. It is impossible to deform volumetric object while at the same time preserved its volume as objects represented as parametric surfaces hold no volume information whatsoever. Eventually, simple deformation requires the adjustments of multiple control points or reconstructing the control points altogether which is very tedious.

Global deformation, FFD and EFFD provide higher level control than deformation based on parametric surfaces. While global deformation only provides limited sets of deformations, FFD allow user to manipulate its deformation constraint anyway they like. FFD is not without its setbacks. The first two techniques are limited in permitted deformations as the techniques constraints the deformation with its static deformation constraint but the latter provides a powerful tool as it gives the user the ability to construct the deformation constraint.

Physically based deformable models have seen wide application in many fields of computer graphics. The ability to simulate real world various material behaviors does prove to be useful in the field of medical and engineering. Physical based model limits the direct user controls of the deformation process. Deformations are computed using approximations of physical dynamics. Sometimes deformation behavior is unpredictable due to gross approximation of dynamics. This can be seen when tuning mass spring system spring stiffness for specific materials. Unlike most non-physical based deformation technique, deformation parameters for physical based technique are much more complicated to configure. With limited computing power, computing complex dynamics is very expensive. For finite element method, internal geometry structures are required for dynamics computation. Gas pressure method on the other hand, does not have this internal geometry structure for its dynamics computation thus making it less memory footprint requirements. Physical based method does not appeal to some computer graphics application especially in the field of object modeling and editing because of it gives user limited control of deformations.

REFERENCES

1. Baraff, D. and A. Witkin, 1998. Large steps in cloth simulation. In proceedings of siggraph.
2. Alan, W. and W. Mark, 1992. Advanced Animation And Rendering Techniques:Theory And Practice Acm Press, Addison-Wesley.
3. Thomas, W., S. Sederberg and R. Parry, 1986. Free-Form Deformation Of Solid Geometric Models Siggraph.
4. Sabine C., 1990. Extended Free-Form Deformation : A Sculpturing Tool For 3d Geometric Modeling Siggraph.
5. Gentaro H., M. Renee and C.L. Ming, 1999. Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization Appeared In Acm Solid Modeling '99 Paper Session.
6. William M.H., F.H. John and K. Henry, 1992. Direct manipulation of free form deformations computer graphics.
7. Petros, F., V.D.P. Michiel and T. Demetri, 1997. Dynamic free-form deformations ieee transactions on visualization and computer graphics.
8. Terzopoulos, D., J. Platt, A. Barr and K. Fleischer, 1987. Elastically Deformable Models. In Siggraph.
9. Terzopoulos, D. and A. Witkin, 1988. Physically based models with rigid and deformable components. Ieee computer graphics and applications.
10. Terzopoulos, D., J. Platt, K. Fleischer, 1989. Heating and melting deformable models from goop to glop. In graphics interface.
11. Christophe, G. and F.R. Jean Gmsh, 2005.A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities version 1.60, 14 www.Geuz.Org/Gmsh/Index.Htm.
12. Hang, S.T., 2005. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator.research group of numerical mathematics and scientific computing, weierstrass institute for applied analysis and stochastics mohrenstr. Version 1.3.4, Http://Tetgen.Berlios.De/
13. Bro-Nielsen, M. and S. Cotin, 1996. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. Computer graphics forum.

14. Basdogan, C., 2001, Real-time simulation of dynamically deformable finite element models using modal analysis and spectral lanczos decomposition methods, proceedings of the medicine meets virtual reality mmvr conference, Irvine, Ca.
15. Xunlei, W., S.D. Michael, G. Tolga and T. Frank, 2001. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes eurographics 2001 / a. Chalmers and t.-m. Rhyne guest editors.
16. Petr, K., G. Eitan and S. Peter, 2003. Oder natural hierarchical refinement for finite element methods. Intl. J. Numerical methods in engin.
17. Teschner, M., B. Heidelberger, M. Müller and M. Gross, 2004. A versatile and robust model for geometrically complex deformable solids. In Proceedings of Computer Graphics International cgi.
18. Mollemans, W., F. Schutyser, J.V. Cleynenbreugel and P. Suetens, 2003. Tetrahedal mass spring model for fast soft tissue deformation IS4TM, pp: 145-154.
19. Gibson, S.F. and B. Mirtich, 1997. A survey of deformable models in computer graphics. Technical Report tr-97-19, Merl, Cambridge.
20. John, E., D.R. Chadwick, R. Haumann and E. Parent, 1989. Layered construction for deformable animated characters computer graphics.
21. Christensen, J., J. Marks and J.T. Ngo, 1997. \Automatic motion synthesis for 3D mass-spring models, The Visual Computer, pp: 20-28.
22. Cotin, S., H. Delingette and N. Ayache, 2000. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. The visual computer.
23. Fuhrmann, A., C. Groß and V. Luckas, 2003. Interactive animation of cloth including self collision detection. In Proceedings of Winter School of Computer Graphics WSCG.
24. Jacobsen, T., 2003. Advanced Character Physics. http://www.gamasutra.com/resource_guide/20030121/jacobson_pfv.htm
25. Maciej, M. and O. Mark, 2003. Pressure model of soft body simulation sigrad the annual sigrad conference. Special Theme–Real-Time simulations, Umeå University, Umeå, Sweden.
26. Kyle, O., 2005. Area and volume calculations http://www.gamedev.net/reference/articles/article2247.Asp Kwoxrf@Umr.
27. Szeliski, R. and D. Tonnesen, 1992. Surface modeling with oriented particle systems. Computer Graphics 26: 2.
28. Desbrun, M. and M.P. Cani, 1995. Animating soft substances with implicit surfaces. In Computer Graphics Proceedings, Acm Siggraph.
29. Desbrun, M. and M.P. Cani, 1996. Smoothed particles: a new paradigm for animating highly deformable bodies. In 6th eurographics Workshop on Computer Animation and Simulation '96.
30. Desbrun, M. and M.P. Cani, 1999. Space-time adaptive simulation of highly deformable substances. Tech. Rep., Inria nr. 3829.
31. Müller, M., R. Keiser, A. Nealen, M. Pauly, M. Gross and M. Alexa, 2004. Point based animation of elastic, plastic and melting objects. In Proceedings of the 2004 Acm siggraph/Eurographics Symposium on Computer Animation.