April 1993

Physics-Based Modeling, Analysis and Animation

Ioannis A. Kakadiaris
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-45.

Physics-Based Modeling, Analysis and Animation

Abstract

The idea of using physics-based models has received considerable interest in computer graphics and computer vision research the last ten years. The interest arises from the fact that simple geometric primitives cannot accurately represent natural objects. In computer graphics physics-based models are used to generate and visualize constrained shapes, motions of rigid and nonrigid objects and object interactions with the environment for the purposes of animation. On the other hand, in computer vision, the method applies to complex 3-D shape representation, shape reconstruction and motion estimation. In this paper we review two models that have been used in computer graphics and two models that apply to both areas. In the area of computer graphics, Miller [48] uses a mass-spring model to animate three forms of locomotion of snakes and worms. To overcome the problem of the multitude of degrees of freedom associated with the mass-spring lattices, Witkin and Welch [87] present a geometric method to model global deformations. To achieve the same result Pentland and Horowitz in [54] delineate the object motion into rigid and nonrigid deformation modes. To overcome problems of these two last approaches, Metaxas and Terzopoulos in [45] successfully combine local deformations with global ones. Modeling based on physical principles is a potent technique for computer graphics and computer vision. It is a rich and fruitful area for research in terms of both theory and applications. It is important, though, to develop concepts, methodologies, and techniques which will be widely applicable to many types of applications.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-45.

Physics-Based
Modeling Analysis and Animation

MS-CIS-93-45
GRASP LAB 346

Iaonnis A. Kakadiaris

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

April 1993

Physics-Based
Modeling, Analysis and Animation

SPECIAL AREA EXAM

Ioannis A. Kakadiaris
Advisor: Ruzena Bajcsy

GRASP Lab
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

APRIL 21, 1993

Physics-Based Modeling, Analysis and Animation

Ioannis A. Kakadiaris

Abstract

The idea of using physics-based models has received considerable interest in computer graphics and computer vision research the last ten years. The interest arises from the fact that simple geometric primitives cannot accurately represent natural objects. In computer graphics physics-based models are used to generate and visualize constrained shapes, motions of rigid and nonrigid objects and object interactions with the environment for the purposes of animation. On the other hand, in computer vision, the method applies to complex 3-D shape representation, shape recon- struction and motion estimation.

In this paper we review two models that have been used in computer graphics and two models that apply to both areas. In the area of computer graphics, Mill- er [48] uses a mass-spring model to animate three forms of locomo- tion of snakes and worms. To overcome the problem of the multi- tude of degrees of freedom associated with the mass-spring lattices, Witkin and Welch [87] present a geometric method to model global deformations. To achieve the same result Pentland and Horowitz in [54] delineate the object motion into rigid and nonri- gid deformation modes. To overcome problems of these two last approaches, Metaxas and Terzopoulos in [45] successfully combine local deformations with global ones.

Modeling based on physical principles is a potent technique for computer graphics and computer vision. It is a rich and fruitful area for research in terms of both theory and applications. It is important, though, to develop concepts, methodologies, and techniques which will be widely applicable to many types of applications.

Contents

1 Introduction

Physics-based modeling is an exciting paradigm which made its debut in computer graphics less than ten years ago. Existing geometric modeling techniques are not able to model the shape of natural shapes, like clouds and waves, and are not successful in generating aesthetically pleasing animations of their motion. The ultimate challenge for geometric methods is the animation of human movement. The human body is formed from a wide variety of complex shapes and performs an amazing number of complex motions as the result of the articulated skeleton and skin deformations.

Machine vision researchers also have recognized the importance of developing algorithms based on increasingly sophisticated physical models. Previous attempts at shape representation have met only with partial success in satisfying the often conflicting requirements of shape reconstruction and recognition. In addition the results from the previous studies of rigid motion cannot be extended to the analysis of nonrigid motion. We need new models that can accommodate deformation, non-convexity, non-planarity, inexact symmetry and local irregularities.

The new paradigm of physics-based modeling has become of special importance due to the advent of non-destructive sensing equipment (e.g LRF, CT, MRI and PET) that generates numerical sampling of real three-dimensional objects. This ability has been proven essential in numerous fields. It allows for the inspection of mechanical parts without damaging the product and the examination of a patient's organs without invasive surgery. Computer systems used by radiologists and physicians can incorporate model fitting methods to segment, analyze and visualize 3-D medical images. An example of such an application is the analysis of the nonrigid motion and the estimation of the motion parameters involved in the deformation of the heart, for the purpose of determining the fitness of an athlete. In applications such as teleconferencing, model-based image compression significantly reduces the size of information bandwidth when compared to the traditional statistical approaches. The wide range of applications using physics-based modeling that have been proposed and explored include range scanning, vehicle guidance, design automation and manufacturing automation, surveillance and remote sensing. Currently, research is also being conducted in the following areas: modeling and analysis of the motion of the heart, human face/head motion analysis and synthesis for model-based image compression, and evolution of coherent structures in fluid motions.

Several research questions arise. What computational methods allow us to represent objects which are more complex and which require greater representational accuracy than the ones we can provide today? What are the models suitable for modeling the motion of an object and what are the relevant mathematics? To what extent will the physics-based paradigm and solutions created for computer graphics and computer vision be useful for modeling efforts in other fields?

The principles and techniques reviewed in this paper represent efforts towards achieving the preceding research goals and questions. The goal of the paper is to present some of the physics-based methods applied in computer graphics and computer vision and evaluate their potential. Emphasis is placed on describing a strategy for designing and managing the complexity of physics-based models in order to increase understanding, generality, reusability and communication of the models.

The paper is organized as follows: In section 2, I expatiate on the term physics-based modeling and define its meaning on computer graphics and computer vision. In the area of computer vision, the term is used in two different contexts. I distinguish between the case in which exact physical models are taken into account to link the physics of image formation to the perceived image, and the case in which the principles of physics are used to drive the processes of both analysis and synthesis of an object's shape and motion. The problems of representing nonrigid articulated objects and animating their interaction with the rest of the world is discussed in section 3. Section 4 refers to the representation and reconstruction from noisy image data of complex nonrigid shapes and the estimation of their motion parameters. I review four papers from the areas of computer graphics and/or computer vision in section 5. These papers can be broadly classified into two categories. Miller and Witkin et al. present techniques applicable only to computer graphics, whereas the techniques proposed by Pentland et al. and by Metaxas et al. find applications in computer vision as well. Section 6 discusses the implications of using a mass-spring model versus a continuous medium to model a nonrigid object. I conclude in section 7 by summarizing the presented papers and my critique of them, and by exploring possible future applications of the paradigm both in computer graphics and computer vision.

2 Physics-Based Models

Physics-based modeling is a cross-disciplinary field, including elements of applied mathematics, numerical analysis, computational physics, computer graphics, computer vision, and software engineering. It has different ends than its parent fields, physics and applied mathematics, and it is somewhat different from its sibling fields, such as computational physics and classical computer graphics. Its long term goal is to develop methods enabling us to specify, design, build and control computational models of heterogeneous physical systems of objects.

The term *physics-based modeling* (or *physically-based modeling*) has become a catch-all term for a variety of techniques that all share the approach of defining physical principles of behavior of their models. A physics-based model is a mathematical representation of an object (or its behavior) which incorporates physical characteristics such as forces, torques and energies into the model, allowing numerical simulation of its behavior. In computer graphics, common elements are classical dynamics (motion based on forces, mass, inertia, etc.) with rigid or flexible bodies, inter-body interaction and constrained-based control.

In computer vision the prefix *physics-based* has been used to denote two approaches. In the first approach, the physics of image formation is taken into consideration to link the 3-D real world to the images which are the input to a vision system. Progress has been made in modeling the properties of surfaces [14, 68], illuminants [40], and sensors to exploit phenomena such as color [62, 29, 37], shading [31], highlights [41], polarization [89], and inter-reflection [42] for image interpretation. The physical models have led to new algorithms for segmenting images [6] and recovering properties of surfaces such as shape, spectral reflectance [49], and material. The second approach uses the principles of physics to form an abstraction of the world. The surface of the model is composed from simulated elastic materials that deform in response to applied forces. Constraints, whether derived from the image or specified by a human operator who builds the model, generate forces that mold the model to the desired form. The way in which the model responds to the applied forces depends on the desired properties of the object modelled. For example, the dynamic evolution of the model through time can be described in the form of differential equations, which can be solved numerically to estimate the shape and motion parameters of a moving object.

This new paradigm aims to create abstractions and mathematical representations of objects which move and their shape changes with time. Geometric constraint properties, mechanical properties of objects, the parameters representing the shape of an object, and the control of its motion are incorporated into the same conceptual framework.

3 Computer Graphics

As the computer graphics field matures, there is an increasing demand for complex, physics-based models. Previous models have often been ad hoc, special purpose, obscure and/or hard to extend. Little attention has been given to design methodologies. In addition, researchers want to be able to:

- Model nonrigid objects and their interaction with the physical world.
- Realistically animate the motion of articulated objects with possibly deformable parts.

In this section, we review work on nonrigid object modeling and animation, prior to the use of physics-based models, and mention some of the approaches that emerged from this framework.

Nonrigid object modeling: The major issues involved in object modeling include the effectiveness in modeling the desired properties, the implementation complexity of the model and its computational cost. Mathematical representations of solid objects are abundant in the computer graphics literature. Although these representations are particularly useful for modeling stationary, rigid objects whose shapes do not change over time, they are often inconvenient for modeling object motion and, moreover, the shape of natural objects. Even when spline patches [8, 26] are used to represent free form shapes, they have been treated as purely geometric entities deemphasizing their physical underpinnings. For example, McPheeters in [90] proposes animating soft objects as iso-surfaces in a 3-D scalar field enveloping control points. Since the control-point dynamics do not accurately describe the dynamics of the deformable materials, animations look contrived. Several researchers have used physics-based models [56, 61] to model waves [51], turbulence [91], clouds [59], terrain [36], cloths [81], skin [75], and deformable curves, surfaces and solid primitives [70], with elastic or inelastic behavior [71].

Motion animation: One of the most challenging aspects in computer animation is the control of object parameters like position, orientation and motion path. As the parameters change over time, the corresponding attributes of the object change to produce the animation. Animation techniques currently in use include [3]: *keyframing*, *parametric interpolation*, *kinematics*, *inverse kinematics*, *dynamic animation*, *constraints*, *simulation* and *scripting systems*.

A well-known method for specifying motion of a geometric object for computer animation is *keyframing* [18]. The keyframe technique is an extension of how traditional cell animation is done. In cell animation, the most talented artists draw the figures at *key* positions and other animators fill in the inbetween positions. The two-dimensional drawing of an object's boundary is transformed over time by interpolating points on the drawing between specified positions of key points in a sequence of keyframes. Whereas keyframing may be appropriate to model motions that are not too complex or need not bear too much resemblance to reality, it turns out that a very skillful animator is needed even to program reasonably realistic movements of nontrivial objects, let alone human or animal motion.

In *parametric interpolation* [63, 65], the user interactively specifies the values of object's parameters at certain instances of time. Next, using some interpolation rule, the object's shape and position are computed for the intermediate instances. Motion of an object along a given path, for example, can be achieved using parametric interpolation. Since the number of the required parameters can easily escalate to hundreds, the interaction between the parameters may become unmanageable.

Rather than specify key positions to be interpolated, the animator can specify a starting position and a function of time that specifies the change of the parameters. This method is called *kinematics* [76] since the motion of the object is controlled by functions of position, velocity or acceleration.

Inverse kinematics [4, 39] is used to handle the complexity of motion of humans and animals. Inverse kinematics refers to the positioning of a joined structure by defining the goal position for the end effector and computing the positions and orientations for the intermediate joints of the linkage.

To relief the burden of the animator, several researchers [82, 9, 70] advocate an alternative way of describing motion, namely *dynamic animation* based on Newton's laws. This approach uses time-dependent forces and moments to drive the motion of the center of gravity of the object and the motion of its components in order to produce physically correct motion.

While physics-based modeling has improved the realism of the animated objects, there remains much to be done with respect to controlling not only low level motion, but also the high level interactions of complex systems. *Constraints* [58, 50, 86, 84], in the form of relationships, boundary conditions, potential functions or springs, have been used to describe the structure of complex physical systems and to specify the goals of motion. Motion is the result of time-dependent constraint forces. These forces operate on hinge points between the components of the object in order to keep the components assembled. For example, the different parts of an articulated object are constrained not to separate. Force-based constraint methods enforce the constraints by adding external forces and impulses to physical systems [9, 73, 83]. [See Appendix A for a review of the methods used for control of the animation.]

Given a description of a process involving object interactions or parameter relationships that cannot pre-computed, a *simulation* approach is needed. The dynamic nature of the simulation [92] allows very general simulations to be modeled and animated.

A *scripting system* is a programming language where arbitrary changes to program variables can be invoked. The parameter changes are given times or temporal relationships and then posted to the event list for a simulation-style execution.

In summary, physics-based modeling facilitates the creation of complex shapes and realistic motions – once the sole province of highly trained modelers and animators. In addition, it adds new levels of representation of objects; embodies physical laws which make them responsive to one another and the simulated physical world; and synthesizes complex motions automatically, to produce the desired animation.

4 Computer Vision

In computer vision, the need for physics-based models stems from the desire to:

- Represent complex nonrigid shapes
- Reconstruct them from noisy image data.
- Estimate and track the motion of nonrigid multi-part objects.

Shape representation and reconstruction: The visual processing system must recover the complete three-dimensional description of objects in space, from the intensity changes occurring on a two-dimensional image. Although humans can understand and communicate a wide variety of shapes almost effortlessly, finding a useful and general method for machine representation of shape has proven difficult.

First, the chosen shape must satisfy the often conflicting requirements of shape reconstruction and shape recognition. A representation scheme should afford enough flexibility to describe complex curved objects, and yet provide compact object descriptions capable of supporting recognition. Most existing techniques are limited to rigid objects with simple shapes; natural shapes cannot be represented accurately. There is a need for new models that can accommodate deformation, nonconvexity, nonplanarity, inexact symmetry and local irregularities.

Several different approaches have been used to bridge the gap between raw data and high level representations. Binford [13] introduced the *generalized cylinders* that represent a volume by sweeping a 2-D closed contour along a 3-D space curve which forms the axis of a cylinder. They are suitable for axisymmetric objects, where the axis is clear but they are not suitable for blob-like objects with no obvious axis of symmetry. Since some amount of asymmetry is evident in many synthetic and most natural objects, the use of generalized cylinders may result in the loss of crucial information.

Kass, Witkin and Terzopoulos [35] have developed *snakes* which model the contours of an image by minimizing the energy associated with a spline. The energy of a snake configuration is based upon the image and its first and second derivative, the curvature of the components in the image, and the first and second derivative of the spline. Terzopoulos at al. [74] extended the concept of snakes into symmetry-seeking models that derive a three dimensional shape from a two dimensional image by employing an axisymmetric elastic skin spread over a flexible spline. Although the model is capable of representing natural objects with asymmetries and fine detail, the generalized spline components of the model do not explicitly provide a representation with few parameters. Currently, a priori information about the configuration and orientation of the object being modeled is required.

Solina et al. [64] have used superquadric models with global deformations as volumetric primitives to segment dense range data from complex 3-D scenes into their constituent parts. They define an energy or cost function whose value depends on the distance of object points from the model's surface and on the overall size of the model. Model recovery is formulated as a least squares minimization of the cost function for all range points belonging to the same part. For the case of objects with more than one part, the model can actively search for a better fit by compressing or expanding itself. Gupta [28] has developed an integrated framework for the recovery of structured descriptions of complex objects without a priori domain knowledge. To recover shape descriptions he uses bi-quadric models for surface representation and superquadric models for volumetric representation.

Solid modeling systems could use geometric models created automatically by a vision system. The design time would be reduced, especially when designing sculptured free-form surfaces, because such task is a very time consuming process and typically requires extensive knowledge about the modeling primitives, for instance, spline functions. Since there is no single representation that would be the most appropriate in all situations, Koivunen [38] employs multiple representations, e.g., NURBS surfaces and superellipsoids, to build *procedural* CAD models from range data. Procedural models can represent overall geometric properties useful in analysis and process planning in addition to low level geometric data.

Visual reconstruction as a data fitting problem has received considerable interest in the context of the surface reconstruction problem [15]. Surface reconstruction techniques based on generalized splines [60] have attracted interest in the vision community for several years

[69, 15, 17, 66, 67] (see also the survey [16]). Despite the large body of work on 3-D surface reconstruction, the ability to extract accurate, quantitative shape models has not kept up with the ability to produce the actual images. A promising approach that can be applied to surface reconstruction problems is the use of physics-motivated deformable models. The dynamic model fitting approach is being pursued by several researchers [72, 53, 54], For example, Wang [78] presents a 3D surface reconstruction technique that is based on elastic, deformable models. The basic structure used is an imaginary elastic grid which is made of a membranal, thin plate type material. Shape reconstruction is guided by a set of imaginary springs, derived from the image data, that enforce consistency in the position, orientation, and curvature measurements of the elastic grid and the desired shape. The dynamics of the reconstruction is guided by the principle of least action.

Nonrigid motion Estimation: Motion analysis, the process of inferencing the 3-D motion parameters of an object based on 2-D images of the object taken during two or more time instances, has been an important research topic for several decades. Most of this work has focused on rigid motion, that is, the motion of objects whose shape does not change over time. The results of this work, in general, cannot be extended to the analysis of nonrigid motion. In addition, in previous approaches, motion was conceptualized as being mostly unstructured. What one could say is how a point or a patch is moving, which requires three unknowns per point, and as consequence the problem becomes under-constrained. In that spirit, Ullman's *incremental rigidity scheme* [77] finds the structure and motion of an object that are most consistent with the given noisy data by assuming a small change in the rigidity of this object between frames. It works best for rubbery objects which are almost rigid.

Recently, there has been an increasing trend towards research in nonrigid motion analysis [79, 24, 22, 30, 34] due to its potential applications, which include: remote sensing, range scanning, medical scanning, vehicle guidance, surveillance, design automation and manufacturing automation. Nonrigid motion analysis has proven a rich and fruitful area for research. Applications of the analysis of nonrigid motion include the tracking of cloud movement for weather studies [1], tracking cell mobility [43] and quantifying the motion of the heart [19, 23].

Analysis of the restricted motions of articulated objects has been done by numerous researchers [58, 80, 2, 12]. For example, Webb and Aggarwal [80] studied the case in which the rotation axis can be assumed to be fixed in direction throughout the observed sequence. Chen and Penna in [21, 20] carried out a more general investigation of elastic motion and proposed several approaches. However, motion parameters can be obtained. only under restricted assumptions such as isometry - an isometric motion preserves lengths of curves and angles between intersecting curves. Goldgof and Huang in [27] used 3-D curvature to analyze nonrigid motion. Specifically, they used the mean and the Gaussian curvature to segment out rigid parts of an articulated object, and to distinguish among rigid, isometric, homothetic, conformal and general nonrigid motions. In another deformable matching technique, Bajcsy and Kovačič [5] used a multiresolution approach to elastically deform a known brain atlas to match the image of a scanned brain. This approach decreases the resolution of the data set, then deforms the brain atlas so that the outer edge and ventricles match the data. The resolution is then increased and the deformation process is repeated. The drawback in using an atlas to fit an object is that a sufficiently precise atlas is required for every object to be modeled.

Jasinchi and Yuille in [33] study the problem of recovering the structure from the motion of figures that are permitted to perform a controlled (motion that preserves the Gaussian curvature) nonrigid motion. They use Regge calculus to approximate a general surface by a net of triangles. The nonrigid flexing motion that they examine corresponds to the triangles remaining rigid and the bending occurring only at the joints between the triangles. The depth information of the vertices of the triangles can be obtained using a modified version of the incremental rigidity scheme. In cases in which the motion of the figure displays fundamentally different views at each frame presentation, the algorithm works well, not only for strictly rigid motion but also for a limited amount of bending deformation. The results obtained in the case of two triangles performing rigid global rotation followed by a local bending deformation (along the common edge of the faces) shows that the algorithm performs well when the axis of global rotation is close to the parallel position with respect to the image plane, and when the angle of rotation lies in the range between 30 and 60 degrees. The bending angle also has to be small.

5 Critical review

In the following, I elaborate on the physics-based modeling paradigm by reviewing four relevant papers. I have structured my review of the physical models to have four sections: The *conceptual model*, the *mathematical model*, the *addressed problems* and the *implementation*. These sections can be thought as answers to the following questions:

1. **What is the model trying to do?** The conceptual model is a description of the properties, features and characteristics of the entity being modeled. This abstract description rarely captures all the aspects of the entity; instead it focuses on those aspects that are relevant to one's purpose in creating the model and excludes those that are irrelevant. For example, it may contain information such as mass or momentum that enter into the mathematical description of motion, or surface color and specularity that are useful for the rendering in the case of computer graphics.

2. **What are the underlying equations?** The mathematical model is a collection of mathematical equations that describe the behavior of the model. These mathematical equations state the relationships between the entities in the model and are complete without the definition of the conceptual model; they are context free.

3. **What are the knowns and unknowns?** The third step of a physics-based modeling, once we have defined the conceptual and the mathematical model, is the

collection of the mathematical problems, which are the statements of the conceptual tasks pursued. In other words, in the addressed problems section we examine which of the terms of the mathematical model have known values and which terms are unknown. In addition, we determine the order in which the equations from the mathematical model come into play.

4. **What are the solution techniques?** Sometimes the mathematical problems may be solved analytically, but, in the case of physics-based models, the problems are solved numerically. The implementation section contains details such as which numerical solvers are used for the specific problem, and which numerical parameter settings produce acceptable results.

For the rest of the paper, I will use this division to understand, analyze, and present the reviewed techniques. Additionally, posing and answering these questions during the modeling and development phase can help us build models that will be robust, reliable and extensible.

5.1 The Motion Dynamics of Snakes and Worms

Miller in [48] addresses the question of modeling biological forms for the purposes of animation. Animating three specific modes of locomotion of legless creatures, such as snakes and worms, is attained by using mass-spring models.

Conceptual model Animals can be classified into two categories based on their structure. Animals that have rigid skeletons and parts connected together by joints and animals such as worms and snakes that can be thought as a tube which can be bend and stretched. The complex topology of the skeleton of animals with arms and legs requires the consideration of the dynamics of hierarchical rigid structures. The authors argue that the kinematic and dynamic models that have been presented in the literature for the modeling of legless figures [32], although successful at animating the skeletal structure, have not addressed problems concerning the skin and muscles. Modeling the skin and the muscles is not the only challenge: worms and snakes change shape during the locomotion. They move on the ground in a way that depends on their grip to it. Also, they deform elastically when they come into contact with other creatures or objects in their environment.

The author is not trying to model accurately snakes and worms, but rather to contrive a dynamic model to meet his goal for realistic animation of locomotion. Worms are modeled as tubes formed from elastically deformable materials. Although snakes do have a skeleton, their shape was approximated as a tube as well. Each segment of the creature is modeled as a cube with its mass m distributed at the vertex points having springs of length L along each edge and across the diagonals of each face. The edge and diagonal springs together control the Young's modulus and the diagonal springs affect the shear and twisting moduli. To make the model realistic (since in reality the muscles of worms bulge out during contraction) the circumferential spring lengths are increased as a function of the axial compression. To keep the total volume constant, the springs around the circumference of the worm are scaled using the scale factor $1/L$.

To achieve locomotion, the author incorporates to the model forces, that act on the environment in a direction opposite to the motion. He draws the analogy from walking systems that use isotropic friction to propel themselves forward by alternating pressure from one foot to the other. However, since snakes and worms are constantly in touch with the ground, they utilize different mechanisms. The interaction forces can be categorized into *macroscopic friction* and *microscopic friction*. Macroscopic friction occurs when an object such as a snake interacts with large objects, pushing against them by reconfiguring its body shape. Microscopic friction occurs when the surface micro structure of the object interacts with the small scale features of the surface.

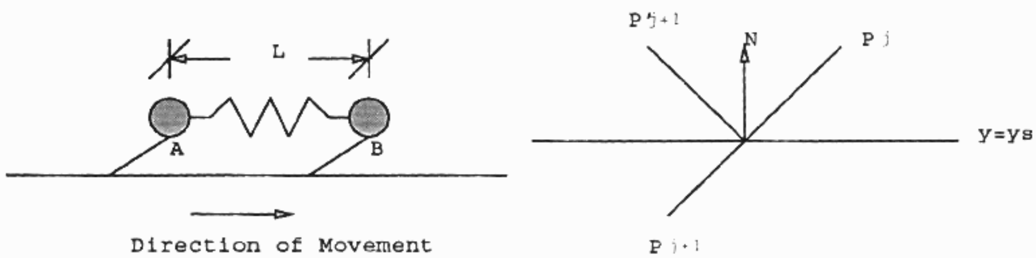

Figure 1: (a) A two mass, one spring worm. (b) Point-plane constraint intersection.

Scientists have observed three basic patterns of locomotion in snakes and worms. In *horizontal undulatory progression* the snake moves forward by throwing out lateral undulations of the body and pushing them against any irregularity in the surface. On surfaces with no macroscopic features to push on, snakes move forward by scuting forward and backward over their ribs. This movement where the snake moves so that every part of its body follows approximately a straight-line path is called *rectilinear progression*. When a body segment moves forward, the scales slide relatively easily over the ground with minimal friction. On the contrary, when the body segment slides backward the scales dig in and the frictional forces are large. Fig. (1a) shows a geometric model of this form of microscopic directional friction. When the spring expands, scale B will slide over the ground and scale A will grip. When the spring contracts, scale B will dig and scale A will slide. A third form of snake locomotion, used on smooth or yielding surfaces, is *sidewinding*. In cases of poor surface grip, such as sand, one way to increase the frictional force is by reducing the contact area with the surface resulting on an increase of the effective pressure. In essence, the snake anchors a portion of the body in the substratum and lifts the rest out laterally to a new position. When the lifted part in anchored again, the portions of the body left behind are lifted to a new position. By constantly lifting and anchoring alternate parts the snake moves in a lateral direction.

Mathematical model The author calculates the forces exerted at the ends of the spring at each time interval. The force f along the spring direction is given by the formula:

$$f = k(L - l) - D\frac{\partial l}{\partial t}$$

where k is the spring constant, L is the minimum energy spring length, l is the current length of the spring and D is the damping constant. The total force, including the force of gravity and forces imposed to the system externally is divided by the mass to give the acceleration. By integrating the acceleration twice with respect to time, the new position $\mathbf{x}_p$ of point p is computed as $\mathbf{x}_p = \frac{1}{m_p}\int\int \mathbf{f}_t dt\ dt$, where m_p is the mass of the point and $\mathbf{f}_t$ is the total force acting on the point.

The effects of the directional friction were studied using a geometric model. The local forward spine unit vector $\mathbf{s}$ was computed from the center of the next and last segments:

$$\mathbf{s} = \frac{\mathbf{x}_{p+1} - \mathbf{x}_{p-1}}{||\mathbf{x}_{p+1} - \mathbf{x}_{p-1}||}$$

The velocity $\mathbf{v}$ was then modified as follows: $if\ \ (\mathbf{s}\cdot\mathbf{v} < 0.0)\ \ \mathbf{v} = \mathbf{v} - \mathbf{s}(\mathbf{s}\cdot\mathbf{v})$, which results in preventing any backward sliding of the creature.

As a point mass moves from one point to another, collision with another point mass or surface is possible. For computer animations, two specific issues are at hand, collision detection and collision response. Detection is primarily a kinematic problem, which involves the relative positions of two objects, while response is a dynamic problem which involves predicting behavior according to laws of physics. The author presents a geometric method to detect if the motion of a point intersects with a surface described by a constraint equation $y = y_s$ and compute the coordinates of the new position in the case of collision. For a point mass traveling from P_j to P_{j+1} (Fig. 1b) the intersection position (x_i, y_i) will be given by $y_i = y_s$ and $x_i = x_{j+1} + (y_s - y_{j+1})\frac{(x_j - x_{j+1})}{(y_j - y_{j+1})}$. The new position (x'_{j+1}, y'_{j+1}) is given by: $y'_{j+1} = y_i + r_n(y_i - y_{j+1})$ and $x'_{j+1} = x_i + r_t(x_i - x_{j+1})$, where r_n is the coefficient of normal reflection, and r_t is the coefficient of tangent reflection, assuming an inelastic collision.

Addressed Problems Using the above mathematical model, the author seeks to animate the three natural modes of locomotion. To achieve horizontal undulatory progression external forces are applied as waves that are send down to the mass-spring system. To achieve the effect of bending the snake the springs on the left hand side are 180 degrees out of phase with those on the right hand side. Rectilinear progression is attained by oscillating the spring length as a function of time. The function chosen is a sine wave, since using a compression square wave results in peculiar distortions of the shape of the worm, due to the coarse nature of the mass-spring approximation. For the sidewinding motion a vertical sinusoidal flexing of the snake, which is 90 degrees out of phase with respect to the horizontal undulations, is used.

Implementation Notes To compute the position $\mathbf{x}$ and the velocity $\mathbf{v}$ of each point-mass the Euler integration method was employed:

$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{v}^{(t)}\ \delta t + \frac{1}{2}\mathbf{a}^{(t)}\ \delta t^2$ and $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + \mathbf{a}^{(t)}\ \delta t$ where $\mathbf{a}^{(t)}$ is the acceleration of a point at time t. The author chose the Euler integration, although it is slower to converge than higher order methods such as the Runge-Kutta [10]. The reason is that the higher order methods are based on the assumption that the forces vary smoothly as a function of time, which does not hold in the present model which has similar behavior as if collision detection was implemented as impulse based forces - infinite forces applied instantaneously.

Comments The use of a mass-spring model, introduces excessive implementation detail at the conceptual level and results to discretization artifacts hardwired to the model. Specifically, the disadvantages are the following:

- Changing the solution parameters requires modifying the higher level model.
- The interaction with other objects is restricted to occur at the grid points of the model. In collision detection for example, tests must be performed in order for small objects not to penetrate through the empty space between the masses.
- The user needs to select initially the number of point-masses suitable for the simulation.

The primary source of complexity at the given model is the number of the point-masses. Since we are only considering one snake, the conceptual model is simple. The mathematical model, on the other hand, is presented collections of equations that describe the individual components of the model but there is no mathematical expression for the model as a whole. In that respect the presentation of the mathematical model is not complete. There is no indication of the amount of mass assigned to each segment neither are the dimensions of the springs specified, nor the coordinate system is presented. In addition, the choice of the model parameters are arbitrary and based on trial and error.

The geometric model for collision detection of the snake with a surface has the following advantages:

- It is easy to compute.
- It is done in the static coordinate frame of the constraint equation that describes the surface so that we can into account the movement of the surface.
- The constraint is always met, and therefore the points are guaranteed not to penetrate the surface.

Nevertheless, in case of surfaces with sharp discontinuities, extensive collision detection tests have to be performed, which increase the computational load of the algorithm.

One possible extension would be to consider environmental obstacles also to use a steer-to-avoid algorithm. Reynolds in [57] presents a detailed steer-to-avoid algorithm based on a perception model of simulated vision.

After a collision is detected the issue of collision response comes into play. In the current scheme there is no efficient method for response that would account for the stress distributions due to deformations across the surface. Moreover, the current model permits the snakes to inter-penetrate themselves when they coil up.

A more realistic model will involve more than one snakes or worms moving at an environment with static and dynamic obstacles. This would require to direct the snakes to specific actions giving the creatures goals and path planning abilities. Having different goals though, implies being able to mingle various behavior modes into a single model, whose overall behavior will change as a function of time.

5.2 Fast Animation and Control of Nonrigid Structures

Systems that use mass-spring lattices possess many local degrees of freedom, and they tend to lead to stiff equations which are expensive to solve. To overcome these problems, Witkin and Welch present in [87] a fast method for approximating the shape of objects composed from nonrigid components, and for controlling their motion during simulation. The goals of the model include the ability to:

- Represent nonrigid objects.
- Simulate nonrigid behavior in a rapid way.
- Produce animation by directly specifying the goals, and by dynamically calculating the motion required to satisfy them.
- Develop a vocabulary of simple goal-directed behaviors that can be chained to produce complex actions.

Conceptual model In order to create simplified dynamic models, Witkin and Welch model nonrigid objects using global deformations, with relatively few degrees of freedom. A global deformation is a mathematical function that maps space to itself by assigning new, deformed coordinates to each point of the undeformed space. They have augmented global deformations in two ways. First, mass is embedded in the space where the deformation acts; second, an energy term is added, inducing either elastic or volume preserving behavior.

Mathematical model Global deformations that are linear functions of the state of the system can be written as $x_i = R_{ij}p_j$, where $\mathbf{x}$ is a world-space point, the components of matrix $\mathbf{R}$ are the generalized coordinates, and $\mathbf{p}$ is a function of the coordinates of the undeformed point, but not of $\mathbf{R}$ or of time. The index i denotes the (x_1, x_2, x_3) coordinates of a point. Notice that while deformations of this form are linear in the state $\mathbf{R}$, $\mathbf{p}$ may depend nonlinearly on the undeformed coordinates, as in $p(x, y, z) = (1, x, y, z, xy, xz, yz, x^2, y^2, z^2)$ which is second order.

A particularly simple linear deformation that can describe the deformation of a body that undergoes affine transformations - translation, rotation, stretch and shear - has $p(x,y,z) = [x,y,z,1]^{\mathsf{T}}$ and

$$\mathbf{R} = \begin{bmatrix} n_{11} & n_{12} & n_{13} & t_1 \\ n_{21} & n_{22} & n_{23} & t_2 \\ n_{31} & n_{32} & n_{33} & t_3 \end{bmatrix}$$

The 3×3 submatrix $\mathbf{N}$ is an ordinary 3-D transformation matrix, and $\mathbf{T}$ is a translation vector. If we imagine a cloud of α fixed points, each one with mass m, all subjected to global deformation, changing the deformation parameters will result in movement of the deformed points. Thus, we can associate a mass displacement (change in the kinetic energy) with a parameter change. Given the velocity of a point $\dot{x}_i = \dot{R}_{ij}p_j$, the kinetic energy of the model is

$$\mathcal{T} = \frac{1}{2}m\dot{x}_i\dot{x}_i = \frac{1}{2}\dot{R}_{ij}\dot{R}_{ik}M_{jk}$$

where $\mathbf{M}$ is a constant symmetric matrix defined by[1] $M_{jk} = mp_jp_k$.

To attain elastic behavior, we define an energy function $\mathcal{V}_e = \mathcal{K}_e|n_{ij}n_{ik} - \delta_{jk}|^2$, where $\mathcal{K}_e$ is a stiffness constant, whose minimum lies at the undeformed state and the δ's are *Kronecker deltas*, defined by $\delta_{ij} = 1$ if $i = j$ and zero otherwise. The validity of the statement can be verified by the following observations. The squared magnitude of a transformed vector x_j is $n_{ij}n_{ik}x_jx_k$, which is equal to the squared magnitude of x for all x exactly when N is orthogonal. An affine deformable body is in its undeformed state exactly when the submatrix N is an orthogonal matrix, that is $n_{ij}n_{ik} = \delta_{jk}$.

In order to attain volume-preserving behavior, in which when a body is stretched along one dimension, should squash along the others, the authors define the energy function $\mathcal{V}_c = \mathcal{K}_c|det(\mathrm{N}) - 1|^2$, where $\mathcal{K}_c$ is a stiffness constant. This is true because an affine transformation is volume-preserving exactly when $det(\mathrm{N}) = 1$. The forces associated with these energy terms are given by the gradients of the defined functions.

The authors omit the potential energy $\mathcal{V}$, noting that the force due to $\mathcal{V}$ is $-(\frac{\partial \mathcal{V}}{\partial q})$, which may be subsumed in the generalized force Q, therefore the Lagrangian $\mathcal{L} = \mathcal{T}$. The generalized coordinates that describe the geometric degrees of freedom of the system are the components of the matrix $\mathbf{R}$. To obtain the Lagrange equations of motion we observe that

$$\frac{\partial \mathcal{L}}{\partial \dot{R}_{rs}} = \frac{1}{2}(\delta_{ir}\delta_{js}\dot{R}_{ik} + \delta_{ir}\delta_{ks}\dot{R}_{ij})M_{jk}$$

Using the identity $a_{ij}\delta_{ij} = a_j$ and also the symmetry of $\mathbf{M}$, we obtain $\frac{\partial \mathcal{L}}{\partial \dot{R}_{rs}} = \dot{R}_{rk}M_{ks}$ from which it follows that $\frac{d}{dt}(\frac{\partial \mathcal{L}}{\partial \dot{R}_{rs}}) = \ddot{R}_{rk}M_{ks}$ and also that $\frac{\partial \mathcal{L}}{\partial R_{rs}} = 0$. Combining the above, the equations that govern the change in the value of the generalized coordinates as a result of the application of the external forces are: $\ddot{R}_{ij}M_{jk} - Q_{ik} = 0$ and since $\mathbf{M}$ is constant its

[1] In the paper is erroneously stated as: $M_{jk} = \sum(mp_jp_k)$ with summation performed over all the mass points in the body.

inverse $\mathbf{W}$ may be precomputed giving:

$$\ddot{R}_{ij} = Q_{ik} W_{kj} \tag{1}$$

where the generalized force $\mathbf{Q}$ due to a force $\mathbf{f}$ applied at a world-space point $\mathbf{x}$ is $Q_{ik} = f_r \frac{\partial x_r}{\partial R_{ik}} = f_r \delta_{ri} \delta_{jk} p_j = f_i p_k$.

Equation (1) refers to the entire system. In a system consisting of more that one objects, the global state vector is formed by concatenating the state vectors of each object. In the case of objects with parts, the joints are abstracted using attachment constraints according to the method described in [85], which is related to [9] and [55]. Given a physical system whose state is described by the vector $\mathbf{q}$, we can implicitly impose a holonomic constraint, where we define as states consistent with the constraint the ones that satisfy the equation $c(q,t) = 0$. If multiple constraints are to be met simultaneously then $\mathbf{c}$ is a vector of constraints The basic assumption is that if the system begins in a legal state with $\mathbf{c}(\mathbf{q},t) = \mathbf{0}$, and $\dot{\mathbf{c}}(\mathbf{q},t) = \mathbf{0}$ then requiring $\ddot{\mathbf{c}}(\mathbf{q},t) = \mathbf{0}$ suffices, at least in principle, to hold the constraints in force. The differential equations governing motion of the constrained system are:

$$\ddot{q}_j = W_{jk}(\mathcal{F}_k + Q_k) \tag{2}$$

where $\mathbf{Q}$ is the known applied force and $\mathbf{F}$ is the constrained force. Because $\ddot{\mathbf{q}}$ depends on the force, the problem is to calculate a constrained force $\mathbf{F}$ that projects the acceleration into the legal subspace. The authors' method is similar to the constraint stabilization method.

First, they express the vector $\ddot{\mathbf{c}}$ as a function of $\ddot{\mathbf{q}}$ as

$$\ddot{c}_i = \frac{\partial c_i}{\partial q_j}\ddot{q}_j + \frac{\partial \dot{c}_i}{\partial q_j}\dot{q}_j + \frac{\partial^2 c_i}{\partial^2 t} \tag{3}$$

where $\frac{\partial \dot{c}_i}{\partial q_j} = \frac{\partial^2 c_i}{\partial q_j \partial q_k}\dot{q}_k$. The constraint force should not add or remove energy from the system therefore according to the principle of the virtual work $\mathcal{F}_j = \lambda_i \frac{\partial c_i}{\partial q_j}$, where λ's are known as *Lagrange multipliers*. Substituting (2) in (3) requiring $\ddot{\mathbf{c}} = \mathbf{0}$, we obtain

$$-\left[\frac{\partial c_i}{\partial q_j} W_{jk} \frac{\partial c_r}{\partial q_j}\right] \lambda_r = \frac{\partial c_i}{\partial q_j} W_{jk} Q_k + \frac{\partial \dot{c}_i}{\partial q_j}\dot{q}_j + \frac{\partial^2 c_i}{\partial^2 t} \tag{4}$$

The constraints are enforced by solving (4) for λ and then using λ to compute $\mathbf{F}$. In practice, $\mathbf{F}$ plus an additional feedback term, are added to the applied forces to compute the legal accelerations. The feedback term $(\alpha c_i + \beta \dot{c}_i)\frac{\partial c_i}{\partial q_j}$, where α and β are constants, inhibits drift and brings the system initially to a legal state.

To handle collisions the authors allow the prescribed velocity of a point to undergo a discontinuous change, using impulses $\mathcal{I}$. Impulse is the product of the average value of a force with the time during which it acts and equal to the change in momentum produced by the force in this time interval. For discontinuities at the velocity, due to very large forces that act for a very short period of time, researchers treat the duration of the event as zero and describe the behavior of the system in terms of the integral over the short time interval. The equation $\ddot{q}_i = W_{ij}Q_{ij}$ becomes $\Delta\dot{q}_i = W_{ij}\mathcal{I}_j$ where $\Delta\dot{q}$ is the change in velocity. Assuming that impulse forces are a linear combination of the constraint gradients the discontinuity at the velocity appears in the direct derivative of the constraint with respect to time.

$$-\left[\frac{\partial c_i}{\partial q_j} W_{jk} \frac{\partial c_r}{\partial q_j}\right] \lambda_r = \Delta \frac{\partial c_i}{\partial t}$$

Once λ is obtained, the constrained impulse is $\mathcal{I}_j = \lambda_i \frac{\partial c_i}{\partial q_j}$.

Addressed Problems To achieve their first goal, being able to represent complex nonrigid objects, the authors used point-to-point constraints. The joint between the parts, around which the bodies may move freely, are abstracted as a point-to-point constraint which requires that two points (one from each object) to coincide in space.

Their mathematical model provides the machinery required to animate a collection of objects, by moving arbitrary points of the object as a function of time. The user can select points on arbitrary frames and specify the trajectories that they are allowed to follow. Given the position and the velocity at the beginning of the frame, the model is able to accurately and stably follow any piecewise twice differentiable trajectory, specified by a point-to-path constraint of the form $R_{ij}p_j - w_i(t) = 0$ ($w(t)$ is a twice-differentiable function of time), at interactive speed. As the control points move along the specified path, the rest of the body points move with passive dynamics. The real advantage of this method is that fewer degrees of control than degrees of freedom are employed since the rest of the motion is determined by the laws of physics.

One important aspect of their technique for controlling the animation is the ability to freely turn constraints on and off during the animation. Turning a constraint off is easy; it involves eliminating the relevant blocks from the constraint matrix $\frac{\partial c_i}{\partial q_j} W_{jk} \frac{\partial c_r}{\partial q_k}$ thus eliminating the restoring forces. On the contrary, turning on a constraint, during an ongoing motion, raises several technical issues. The reason being that, when the constraint is initiated the position and the velocity of the control point, which up to that point were in accordance with the specified splined trajectory, may not initially fulfill the constraint. To handle this problem the authors propose the method of *constraint preroll*. To bring the point smoothly from the uncontrolled state to the required initial state, they compute a spline segment that joins the the two states and is activated shortly before the nominal activation of the constraint.

The authors have placed emphasis on the development of a vocabulary of goal-directed behaviors that can be combined to attain complex behavior. One example of atomic behavior is to move a body point to a specified position and velocity, over a determined time interval. Specifying point trajectories has the disadvantage that it cannot account for the dynamic changes in the environment. For example, in the case of the computation of the motion required for a hand to grasp an object, if the object's position is changed "the hand will happily grab the empty space where the object used to be" [p. 249]. To solve this problem they directly specify the goals of the actions, and dynamically calculate the motion required to satisfy them. In the case of the atomic behavior of chasing a target, if the target is stationary, a spline segment is constructed where the chaser's initial position and velocity are the initial conditions, and the velocity and position at the target are the final conditions. If the target is moving, its position and velocity at the time of contact are calculated based on the current values and a spline to the estimated point is constructed and updated only when things change.

An additional feature of the proposed method for the control of an animation refers to the ability to provide a graceful way to start and stop animation using impulses. The authors not only use an impulse to install the initial control point velocities and start an animation but also to bring the control points to a well-behaved halt.

Comments The use of global deformations offers the advantage of reducing the dimensionality of the equations and eliminating the high-frequency components that lead to stiffness. For deformations that are linear functions of state, the matrix $\mathbf{M}$ is constant which in a true dynamic system is not constant. Pre-inverting this matrix yields considerable benefits of performance and allows reasonably complex systems to be manipulated at interactive speed.

The constrained matrix remains constant except when constraints are added or deleted, therefore interactive speed is attained. If $\mathbf{x}$ is a point on a linearly deformable body, then $\frac{\partial x_i}{\partial R_{rs}} = \frac{\partial(R_{ij}p_j)}{\partial R_{rs}} = \delta_{ir}p_s$, which is a constant. Since each constraint is a linear function of one or more points, the derivative of any constraint with respect to a point $\mathbf{x}$ is constant as well. Their technique for constraint satisfaction offers the additional advantage, over the constraint stabilization technique, that it can handle contradictory constraints with grace. If there are any contradictory constraint forces in the system, they cancel out through the Lagrange multiplier method. The relative strengths of the feedback term determine the relative importance of each constraint and therefore the final state of the system. However, since these feedback terms interact, it is difficult to control the exact motion towards the constraints.

Being able to freely turn control points off and on during the animation is important, since we need not control the movement of all control points at all times. If, for example, they wanted to animate walking it would not be necessary to control heel, toe and knees, all at the same time. The heel position must be accurately controlled before and during the support phase, but during the swing it can just follow the toe.

One important aspect of the authors' research is the development of a vocabulary of goal-directed behaviors that can be combined to attain complex behavior. Specifying goals of the actions for the animation and dynamically calculating the motion required to satisfy them has the advantage that dynamic changes in the environment are taken into account during the animation.

On the other hand, representing only global deformations that are linear in state, they give up the ability to represent a number of natural objects. The individual components of the mathematical model are described as collections of equations, having no mathematical description of the relationships between these collections. Therefore, the presentation of the mathematical model is incomplete.

5.3 Recovery of Nonrigid Motion and Structure

Pentland and Horowitz [54] take advantage of the constraints that real materials impose on the types of nonrigid motion, to allow over-constrained estimates of 3-D nonrigid motion from optical flow data. Their method is inspired by modal analysis, a technique for analyzing the vibrations of linear mechanical systems under periodic forcing conditions, and they discard high frequency modes, to reduce the dimensionality and the stiffness of the models.

Conceptual model The authors use the *force-and-process* metaphor of modeling clay: the shape of the an object is considered to be the result of pushing, pinching and pulling on a lump of elastic material such as clay. For the representation, they employ parametric solid models described as implicit functions. To stress the importance of global deformations they mention the following quote from Grimson "An elastic motion, including that of a walking man with his gestures and facial expressions, could be analyzed into a set of rigid motions of elementary particles if one wished to do so, but it is better thought of in terms of components like bending, flexing, stretching, skewing, expanding and bulging"[p. 730].

Mathematical model The authors' intention being to use deformable models, they use formulations for relating the forces on the surface and within the body to its deformations. Finite Element Method (FEM), which is one of the numerical procedures for solving systems of differential equations for engineering analysis, provides a very efficient formulation for this kind of application. The primary steps of the FEM [10] are to:

- Idealize the system into a form that can be analyzed - discretize.
- Formulate the governing equilibrium equations of the idealized system.
- Solve these equilibrium equations.
- Interpret the results to the entire system - utilize continuum of elements to obtain solutions for the whole system.

The formulation of the displacement-based FEM is based on the fact that when a part of an object is allowed to be displaced, it is possible to calculate the applied forces on it. First, we idealize the structure as an assemblage of elements that are interconnected, and identify the unknown displacements, which are the result of the applied forces. For the equilibrium of the body it is required that for any compatible, small virtual displacements, which satisfy the essential boundary conditions imposed on the body, the total virtual work is equal to the total external virtual work. Energy equations are formulated in terms of the displacements of the nodes of the elements along with the forces corresponding to the unknown node displacements. The equations governing the linear dynamic response of the system of finite elements are[2]:

$$\mathbf{M\ddot{U}} + \mathbf{C\dot{U}} + \mathbf{KU} = \mathbf{R} \tag{5}$$

where $\mathbf{U}$ is a $3n \times 1$ vector of the $(\Delta x, \Delta y, \Delta z)$ displacements of the n nodal points relative to the object's center of mass, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are the $3n \times 3n$ matrices describing the mass, damping and material stiffness between each point within the body and $\mathbf{R}$ is a $3n \times 1$ vector describing the (x, y, z) components of the forces acting on the nodes. Equation (5) is a system of linear differential equations of second order. The solution can be obtained by standard methods for solving differential equations. However, these methods are computationally expensive; hence a few effective techniques are required to reduce the computational cost. Two of these techniques are discussed below.

Direct integration methods: In direct integration the equation is integrated using an iterative technique. By direct, it is meant that prior to numerical integration, the equation need not be transformed. These techniques are aimed at satisfying the equation only at discrete time intervals Δt apart, rather than at any time. The variations in displacement, velocities and accelerations within a time step are assumed and it is this assumption that determines the accuracy, stability and cost efficiency of the solution. Therefore, direct integration techniques are good for simulations with short duration.

Change of basis to generalized displacements: To reduce the computational cost, we can transform the original coordinate system for nodal displacements to one whose basis vectors are the columns of a matrix $\mathbf{P}$ for which $\mathbf{U} = \mathbf{P\tilde{U}}$, where $\mathbf{P}$ is a square transformation matrix and $\mathbf{\tilde{U}}(t)$ is a time-dependent vector of *generalized displacements*. By substituting this transformation into equation (5) and pre-multiplying by $\mathbf{P}^{\top}$, the governing equation is transformed into the equation:

$$\mathbf{\tilde{M}\ddot{\tilde{U}}} + \mathbf{\tilde{C}\dot{\tilde{U}}} + \mathbf{\tilde{K}\tilde{U}} = \mathbf{\tilde{R}} \tag{6}$$

(where $\mathbf{\tilde{M}} = \mathbf{P}^{\top}\mathbf{MP}$, $\mathbf{\tilde{C}} = \mathbf{P}^{\top}\mathbf{CP}$, $\mathbf{\tilde{K}} = \mathbf{P}^{\top}\mathbf{KP}$ and $\mathbf{\tilde{R}} = \mathbf{P}^{\top}\mathbf{RP}$) for the coordinate system defined by the basis $\mathbf{P}$. The new mass, stiffness and damping matrices have smaller bandwidth than the ones of the original system. The optimal basis $\mathbf{\Phi}$ to diagonalize the system of equations, which uncouples the degrees of freedom and provides the ability to find closed form solutions, has as columns the eigenvectors of $\mathbf{M}^{-1}\mathbf{K}$ [10]. These eigenvectors are called the *free vibration modes*. Using this transformation matrix we get

$$\mathbf{\ddot{\tilde{U}}} + \mathbf{\Phi}^{\top}\mathbf{C\Phi}\mathbf{\dot{\tilde{U}}} + \mathbf{\Omega}^2\mathbf{\tilde{U}} = \mathbf{\Phi}^{\top}\mathbf{R}(t)$$

where $\mathbf{\Omega}^2$ has as diagonal elements the eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$.

[2]In the paper the equation is erroneously stated as: $\mathbf{MU} + \mathbf{C\ddot{U}} + \mathbf{K\dot{U}} = \mathbf{R}$

Note that the eigenvector ϕ_i, which is called the ith mode's *shape vector*, describes how the coordinates of each nodal point $(x_i, y_i, z_i)^\top$ change as a function of $\tilde{u}_i$, the ith mode's amplitude:

$$\phi_i = (\frac{dx_1}{d\tilde{u}_i}, \frac{dy_1}{d\tilde{u}_i}, \frac{dz_1}{d\tilde{u}_i}, \ldots, \frac{dx_n}{d\tilde{u}_i}, \frac{dy_n}{d\tilde{u}_i}, \frac{dz_n}{d\tilde{u}_i})^\top$$

Letting $\mathbf{V} = (\dot{x}_1, \dot{y}_1, \dot{z}_1, \ldots, \dot{x}_n, \dot{y}_n, \dot{z}_n)^\top$ be the 3-D velocity of each node, we then have $\mathbf{V} = \mathbf{\Phi}\frac{d\tilde{\mathbf{U}}}{dt} = \mathbf{\Phi}\dot{\tilde{\mathbf{U}}}$, and given the 3-D motions of each node, the modal velocities can be computed as follows: $\dot{\tilde{\mathbf{U}}} = \mathbf{\Phi}^{-1}\mathbf{V}$.

Addressed Problems For shape estimation, which is described in [54], sensor measurements are used to define virtual forces which deform an object to fit the data points. The authors solve the equilibrium equation $\mathbf{KU} = \mathbf{R}$ to obtain the displacements $\mathbf{U}$. To overcome the difficulty posed by the large dimensionality of the matrix $\mathbf{K}$ and to obtain a closed form solution, this equation is converted to the modal coordinate system: $\mathbf{\Phi}^\top\mathbf{K}\mathbf{\Phi}\tilde{\mathbf{U}} = \mathbf{\Phi}^\top\mathbf{R}$.

In the case of motion estimation, the problem is to "find the rigid and nonrigid 3-D motions $\frac{d\mathbf{U}}{dt}$ that best account for the observed 2-D image velocities"[p. 733]. The major difficulty in finding a solution for this case is that there are $3n$ unknown degrees of freedom in the model and at most $2n$ degrees of freedom in the observations. To reduce the number of unknowns, the authors discard the high frequency modes. Therefore the problem in the modal coordinate system becomes to "find the set of 3-D *mode velocities* $\frac{\partial\tilde{\mathbf{U}}}{\partial t}$ that best account for the observed 2-D image velocities"[p. 733]. By choosing the m ($m \leq 2n$) lowest frequency modes, the problem can always be made over-constrained. To estimate the nonrigid 3-D deformation at each subsequent time t given noisy estimates of 2-D optical flow data at m image points, they first allocate each of the optic flow vectors to the nodal point whose image projection is closer to the flow's vector position. This produces estimates of the projected 2-D nodal velocities $\mathbf{V}_p$. The authors construct matrix $\mathbf{\Phi}_p$ by removing the rows of $\mathbf{\Phi}$ that correspond to z-axis displacements, the rows that correspond to the x and y displacement of nodes without nearby optical flow, along with the columns that correspond to modes that cannot be observed under orthographic projection, like translation, scaling and shearing along z-axis. Corresponding rows and columns are removed from the matrix $\tilde{\mathbf{U}}$ also, yielding $\tilde{\mathbf{U}}_p$. To over-constrain the problem, they also discard a sufficient number of the low-amplitude high frequency modes. Therefore, the estimate of the object's 3-D shape $\tilde{\mathbf{U}}_p^{(t)}$ based on the optic flow data is: [3]

$$\tilde{\mathbf{U}}_p^{(t)} = {\mathbf{\Phi}_p}^{-1}\mathbf{V}_p^{(t)}\Delta t$$

Whenever the 3-D velocities of the individual nodes are required, we can convert $\tilde{\mathbf{U}}_p$ back to the original space coordinates by multiplying by $\mathbf{\Phi}_p$.

Until now, we have considered kinematic equations where velocity at only one instant is taken into account. For time sequences however, we need to consider the dynamic properties

[3]In the paper is erroneously stated as: $\tilde{\mathbf{U}}^{(t)} = {\mathbf{\Phi}_p}^{-1}\mathbf{V}_p^{(t)}\Delta t$

of the body and data measurements. The *Kalman filter* [25] is a standard technique for obtaining estimates of the state vectors of dynamic models, and for predicting the state vectors at some later time. The authors use a linear Kalman filter to estimate position and velocity for the finite element modal parameters. Recall that, the *state transition equation* is $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w}$, where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}$ is the driving noise that accounts for the possible inadequacies of the state space model; the *measurement equation* is $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$, where $\mathbf{H}$ is the measurement matrix that describes the linear combinations of the state variables and $\mathbf{v}$ is the noise associated with the measurements $\mathbf{z}$. Then, the optimal estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ is given by the following equation: $\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}_f(\mathbf{z} - \mathbf{F}\hat{\mathbf{x}})$, where $\mathbf{K}_f$ is the Kalman gain matrix. The authors choose as state variables the modal amplitudes $\tilde{\mathbf{U}}$ and their velocities $\tilde{\mathbf{V}} = \dot{\tilde{\mathbf{U}}}$. In state space notation, the system of equations is

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \dot{\tilde{\mathbf{V}}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{V}} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \alpha \tag{7}$$

where α is a noise vector due to nodal accelerations. The observed variable will the $m \times 1$ vector of the 2-D nodal velocities[4], $\mathbf{V}_p = \frac{\mathbf{\Phi}_p}{\Delta t}\tilde{\mathbf{U}}_p + \mathbf{v}$, where $\mathbf{v}$ is the observation noise. The Kalman filter is therefore:[5]:

$$\begin{bmatrix} \dot{\hat{\tilde{\mathbf{U}}}} \\ \dot{\hat{\tilde{\mathbf{V}}}} \end{bmatrix} = \begin{bmatrix} \dot{\hat{\tilde{\mathbf{U}}}} + (\frac{2\alpha}{v})^{1/2}\mathbf{\Phi}_p{}^{-1}\Delta t(\mathbf{V}_p - (\mathbf{\Phi}_p\hat{\tilde{\mathbf{U}}}/\Delta t)) \\ (\frac{\alpha}{v})^{1/2}\mathbf{\Phi}_p{}^{-1}\Delta t(\mathbf{V}_p - (\mathbf{\Phi}_p\hat{\tilde{\mathbf{U}}}/\Delta t)) \end{bmatrix}$$

where v and α are the standard deviations of the modeling and measurement noise respectively. Using these equations, we can formulate the displacement prediction at time $t + \Delta t$.

For the case of two objects attached to each other, the authors assume a virtual spring between a point on each object's surface, which exerts equal and opposite attractive forces on the two points of attachment. Given a priori knowledge of the constraints that the spring imposes, the authors compensate for the contribution of that constraint to the state equations and then estimate motion as previously. The Kalman state equation (7) becomes

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \dot{\tilde{\mathbf{V}}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{V}} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} (\mathbf{R}^c + \alpha)$$

where $\mathbf{R}^c$ is a vector describing the load exerted on each nodal point by all active constraints.

[4]In the paper is erroneously stated as: $\mathbf{V}_p = \frac{\mathbf{\Phi}_p}{\Delta t}\tilde{\mathbf{U}} + \mathbf{v}$

[5]In all Pentland's papers the equation is erroneously stated as:

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \dot{\tilde{\mathbf{V}}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{U}}} + (\frac{2\alpha}{v})^{1/2}\mathbf{\Phi}_p{}^{-1}\Delta t(\mathbf{V}_p - \mathbf{\Phi}_p/\Delta t\tilde{\mathbf{U}}) \\ (\frac{\alpha}{v})^{1/2}\mathbf{\Phi}_p{}^{-1}\Delta t(\mathbf{V}_p - \mathbf{\Phi}_p/\Delta t\tilde{\mathbf{U}}) \end{bmatrix}$$

Implementation Notes Using either mode superposition or direct integration procedure the solution is obtained by numerical integration. However, as the periods of vibration, $T_i = \frac{2\pi}{\omega_i}, i = 1, 2, \ldots, n$ are known, in the numerical integration of equations (6) an appropriate time step can be chosen that ensures a required level of accuracy. On the other hand, if all the n equations are integrated using the same time step, then the mode superposition is equivalent to direct integration. The essence of the mode superposition solution is that frequently only a small fraction of the total number of decoupled equations need to be considered, in order to attain a good approximation. That means that only p eigenvalues and eigenvectors need to be found. The reason for only the lowest modes being considered lies with the fact that when monotonic convergence conditions are not satisfied, the finite element analysis approximates the lowest frequencies, and little or no accuracy can be expected in approximating the higher frequencies.

Comments To solve the shape recovery problem in isolation from segmentation, the authors assume that the object part segmentation is given in advance. Although segmentation and shape representation appear to be distinct problems and are treated as such in most computer systems, Bajcsy et al [7] have presented arguments that these two problems are related and have to be treated simultaneously. If any of the two problems is solved first, the other one becomes easier. For example, if the image is correctly divided into parts, the subsequent shape description of those parts becomes easier. The opposite is also true: when the shapes of parts are known, the partitioning of the image becomes simpler.

The modal representation provides a natural multi-scale representation for three dimensional object shapes in much the same manner as Fourier transform provides a multi-scale image representation for images. Although using deformation modes is efficient for the recovery of smooth, symmetrically deformed objects, there are cases for simple objects, such as rods, for which the error increases severely. In these cases there is not enough data to distinguish between various modes of deformation. For example, in the case of a rod, rigid-body rotation cannot be distinguished from lengthwise contraction. To avoid these problems, the authors limit inter-frame motions to small rotations (less than 10 degrees) and deformations (less than 10% of the object size). In addition, global deformation modes lack parameters with an obvious physical meaning. Such parameters would give more intuitive ability in understanding the parameterized shape.

While the authors present the theory for finding the modes, in fact they do not find the real modes of an object but they assume that the object's shape is a linear combination of different modes of vibration of an undeformed superquadric. Pentland in [52] states that using only the low order modes to describe object deformations is valid since the low order modes change very slowly as a function of object's shape. Consequently, he assumes that the same vibration modes can be used to describe the shape of a range of different - but similar - undeformed shapes without incurring substantial error. Since we know that when the shape changes, the eigenvectors and eigenvalues change, a quantitative analysis should be provided to back the assumption above. The authors make the additional claim that by describing the object behavior using a truncated series of vibration/deformation modes, one can obtain the best r.m.s. error description possible for a given number of parameters. This claim, though, is not supported.

Moreover, their modeling primitives are not fully dynamic in that the underlying superquadric parameters do not respond to forces and are not fitted to data through force interactions. The authors point out that another limitation of the method arises from the fact that the input is the optical flow rather than feature points. The use of optical flow requires the integration of object motion over time in order to determine the object's current position and shape. There is no way to connect the estimates of position and shape to current observations.

The notation at the development of the Kalman filter in the paper is problematic with typographical errors, which we already have pointed out, and conceptual errors. Specifically, by assuming that the velocity is constant their system is no longer dynamic. In addition, they assume that the observations are linear combinations of the generalized coordinates, which is not true. Also, the assumption that the error covariance matrix is constant limits the ability of an estimator to recover accurate parameters from the data.

The use of springs to model hard point-to-point constraints is problematic since springs are appropriate for modeling weak constraints only. The reason is that springs do not cancel the force components violating the constraints and stiffening of springs yields ill-conditioned equations.

5.4 Constrained Deformable Superquadrics and Nonrigid Motion

To overcome the restrictions of the linearly deformable models in [87] and quadratically deformable ones in [53], Metaxas and Terzopoulos in [45] successfully combine local deformations with global deformations. The authors present a physics-based framework for shape and motion estimation with the following specific goals:

- Recovery and representation of closed surfaces with complex shapes.
- Estimation of nonrigid 3D motion.
- Fast computation of the point-to-point constraints.
- Tracking of articulated objects with deformable parts.

Conceptual model The authors use as models abstract viscoelastic solids and imbue them with mass and damping densities to make them dynamic - the positions of material points become a time-dependent function. Using Lagrangian dynamics, the energies yield forces and when the forces equilibrate, the model becomes static.

Mathematical model For the case of a solid whose material coordinates are $\mathrm{u} = (u, v, w)$, the position of the points on the model can be written as $\mathbf{x}(\mathrm{u}, t) = \mathbf{c}(t) + \mathbf{R}(t)\mathbf{p}(\mathrm{u}, t)$ where

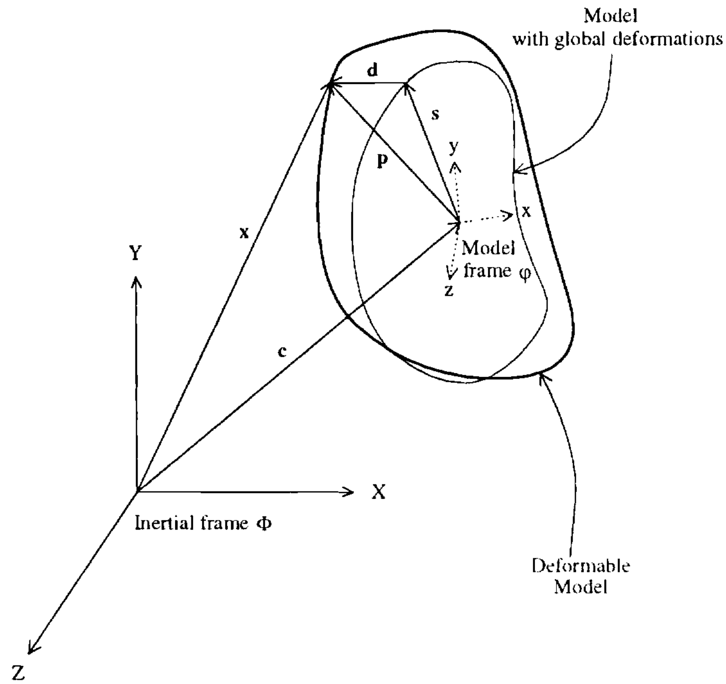

Figure 2: Geometry of deformable model

$\mathbf{x}$ gives the positions of points on the model with respect to the fixed reference frame Φ and $\mathbf{c}$ is the instantaneous position of the non-inertial model-centered frame ϕ, whose orientation relative to Φ is $\mathbf{R}$. The positions $\mathbf{p}$ of points relative to ϕ can be expressed as $\mathbf{p}(\mathrm{u}, t) = \mathbf{s}(\mathrm{u}, t) + \mathbf{d}(\mathrm{u}, t)$ where $\mathbf{s}$ is the reference shape and $\mathbf{d}$ a displacement function.

Any geometric primitive described as a differentiable parameterized function of u - $\mathbf{e}(\mathrm{u}; a_1, a_2, \ldots)$ - can be used as a reference shape. To gain additional modeling power the authors extend the reference shape to include parameterized global deformations $\mathbf{s} = \mathbf{T}(\mathbf{e}(\mathrm{u}; a_1, a_2, \ldots); b_1, b_2, \ldots)$, where $\mathbf{T}$ is a sequence of primitive deformation functions (like tapering, bending and shearing) with parameters b_i. The global deformation parameters form the vector $\mathbf{q}_s = (a_1, a_2, \ldots, b_1, b_2, \ldots)^\top$.

To further enhance flexibility, local free-form deformations are incorporated directly into the geometric primitive as finite element shape functions. The authors employ the finite element method to discretize the deformable surface models into a set of connected element domains. By collecting the displacement vectors $\mathbf{q}_i$, associated with each node at the corners of each finite element, they construct the vector $\mathbf{q}_d = (\ldots, \mathbf{q}_i, \ldots)^\top$. The displacement which describes the local deformations is $\mathbf{d} = \mathbf{S}\mathbf{q}_d$, where $\mathbf{S}$ is a shape matrix whose entries are the finite element basis functions.

The velocity of a point is $\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{L}\dot{\mathbf{q}}$, where $\mathbf{q}$ is the vector of generalized coordinates for the dynamic model and $\mathbf{L}$ is the Jacobian that maps q-space to 3-space. The generalized coordinates are the geometric parameters of the solid primitive, the global and local deformation parameters, and the six degrees of freedom of rigid body motion.

To make their models dynamic, the authors assume that the model has a mass distribution $\mu(\mathrm{u})$, that the material is subject to frictional damping and that the material deforms elastically or viscoelastically. The dynamic behavior of the model is governed by

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_q + \mathbf{f}_q.$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{K}$ are the mass, damping and stiffness matrices respectively, $\mathbf{g}_q$ are the generalized inertial forces arising from the dynamic coupling between local and global degrees of freedom and $\mathbf{f}_q(\mathrm{u}, t)$ are the generalized external forces.

The elastic properties of the model determine the stiffness matrix $\mathbf{K}$. The energy $\mathcal{E}_{s_i}$ associated with each of the global parameters is assumed Hookean; therefore $\mathcal{E}_{s_i} = \frac{1}{2}\mathcal{K}_{s_i}(\mathbf{q}_{s_i} - \mathbf{q}_{s_{i0}})^2$, where $\mathcal{K}_{s_i}$ is the stiffness associated with the global parameter $\mathbf{q}_{s_i}$ and $\mathbf{q}_{s_{i0}}$ is the natural rest value of this parameter. One of two models is used depending on the desired continuity of the surface of the deformable model. The *loaded membrane* is suitable for $\mathbf{C}^0$ continuous surfaces and the *thin plate under tension* model is used for $\mathbf{C}^1$ continuous surfaces.

To connect these new dynamic primitives together, the authors use point-to-point constraints between two objects or parts of an object. They compute the generalized forces between the models, using a stabilized Lagrange multiplier technique which is based on Baumgarte's constraint stabilization technique [11]. Although the Lagrange multiplier method is very general, it is potentially expensive for the deformable models. As an alternative, a fast specialized method to compute the unknown generalized constraint forces $\mathbf{f}_{g_c}$ associated with the point-to-point constraints is proposed. For the case of a single point-to-point constraint, let $\mathbf{f}_c$ be the constraint force between the first and the second object or part of an object. The constraint equation can be written as $\mathbf{N}\mathbf{f}_c + \mathbf{v} = 0$, where the 3×3 matrix $\mathbf{N}$ and the 3×1 vector $\mathbf{v}$ must be determined. After computing $\mathbf{N}$ and $\mathbf{v}$, then $\mathbf{f}_c = -\mathbf{N}^{-1}\mathbf{v}$. In the case of multiple point-to-point constraints, they define an object's constraint force vector $\mathbf{f}_c = (\mathbf{f}_{c_1}, \mathbf{f}_{c_2}, \ldots, \mathbf{f}_{c_k})$, where $\mathbf{f}_{c_i}$ is the constraint force for constraint i, assemble a system of equations $\mathbf{N}\mathbf{f}_c + \mathbf{v} = \mathbf{0}$, where $\mathbf{N}$ is a $3k \times 3k$ matrix, $\mathbf{v}$ a vector with length $3k$ and solve it through LU decomposition of $\mathbf{N}$.

Addressed Problems The physics-based framework that is described above can be used in computer vision for both shape estimation and motion estimation. Dynamic fitting is achieved by integrating Lagrangian equations of motion through time to adjust the deformation degrees of freedom of the models. The objectives of the method are the following: first, to fit the data as fast as possible and second the model to come to rest as soon as the data dependent forces vanish. A more tractable subset of the Lagrange equations, while preserving the useful dynamics, is attained by setting the mass density to zero: $\mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q$ (since $\mathbf{M}$ and $\mathbf{g}_q$ vanish). Therefore, the model has no inertia and comes to rest as soon as all the applied forces vanish or equilibrate. These applied forces are internal forces, which describe elastic properties of the surface, and external forces which are produced from the salient image features. External forces can subsequently be divided into *short range* and *long range* forces. The image $I(x,y)$ is converted into a force field using the gradients of image potentials $P(x,y) = \| \nabla(G_\sigma \star I) \|$, where G_σ denotes a Gaussian smoothing filter of characteristic width σ. The *short range* forces $\mathbf{f} = \beta \nabla P$ act on the model, and deform it to become consistent with the image data. On the other hand, the *long range* forces are based on the distance between the data points and the model's surface. The goal is to adjust the translational, rotational and deformational degrees of freedom of the model to be consistent with the data.

Background knowledge about the image formation and the shape of the objects can be incorporated in the form of constraints. For example, these constraints can help to retain the parts of an articulated object in the correct configuration. Constraints can also be helpful in deriving the shape and motion of an occluded part.

In the case of moving objects, the authors have developed a nonlinear Kalman filter for recursively estimating the shape and nonrigid motion. The nonlinear Kalman filter employs as a model the motion equations and transforms the discrepancy between current observations and the positions of the model into forces that attract the model. The Kalman filter equations for the dynamic model take the form:

$$\dot{\mathbf{q}} = -\mathbf{D}^{-1}\mathbf{K}\mathbf{q} + \mathbf{D}^{-1}\mathbf{f}_{q_c} + \mathbf{w}$$

$$\mathbf{z} = \mathbf{H}(\mathbf{q}) + \mathbf{v}$$

where $\mathbf{H}$ is the matrix that relates the time varying measurements $\mathbf{z}(t)$ to the model's state vector $\mathbf{q}(t)$, while $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are modeling and measurement errors. The state estimation equation is:

$$\dot{\hat{\mathbf{q}}} = -\mathbf{D}^{-1}\mathbf{K}\hat{\mathbf{q}} + \mathbf{D}^{-1}\mathbf{f}_{q_c} + \mathbf{G}(\mathbf{z} - \mathbf{H}\hat{\mathbf{q}})$$

where $\mathbf{G}(t)$ is the Kalman gain matrix, which depends on the system's dynamics and noise processes, and $\hat{\mathbf{q}}$ is the estimated state of the model.

Experimental results suggest that the model is useful for shape reconstruction of objects or part of objects with irregular shape from regular or sparse data and for tracking the motions of articulated objects consisting of rigid and nonrigid parts.

The authors have used the same framework to address challenges in computer graphics and visualization also [46]. As in [83, 9], constraints can be used to assemble complex objects from initially mis-positioned parts. The framework is used to synthesize articulated objects with deformable parts, nonrigid motions and object interactions with the physical world for the purposes of animation. Animations depict flexible multibody objects in gravitational fields, including elastic collisions with obstacles and friction effects.

Comments General purpose shape reconstruction requires models with the ability to assume a wide range of shapes. The models must extract meaningful information from noisy sensor data while, at the same time, making the weakest possible assumptions about observed shapes. The proposed physics-based framework allows the systematic creation of dynamic models from parameterized solid primitives, global geometric deformations and local deformations. Global deformations contribute to the efficiency and accuracy of estimation by coarsely approximating the true shape of the object, which subsequently allows accurate shape recovery by using local deformations. The coupling of rigid-body and deformation dynamics is similar to that described in [74], but the new formulation accommodates global deformations defined by fully nonlinear parametric equations. Therefore, the proposed framework can potentially satisfy the often conflicting requirements of shape reconstruction and shape recognition.

An important feature of the approach is that the method applies across all geometric primitives and deformations. Depending on the application, the user can specify the geometric model and the method automatically converts the geometric degrees of freedom of the model to physical degrees of freedom through the Lagrange equations of motion. Moreover, the various geometric parameters assume well-defined physical meanings in relation to prescribed mass distributions, elasticities, and energy dissipation rates.

The models are dynamic and their behavior is governed by the Lagrange equations of motion. The equations of motion make the models responsive to forces derived from the image data or constraints. These forces cause the models to conform to the projected shape of the object in the image space.

The authors address the challenges related to the application of constraints to construction and control of articulated models by developing efficient computation methods for the point-to-point constraints. The method involves the solution of a linear system whose size is equal to the number of constraints, which is usually small.

To speedup computations, they do not assemble and factorize a finite element stiffness matrix, as is common practice in finite element analysis, but instead they compute $\mathbf{Kq}_d$ efficiently in an element-by-element fashion. The element-by-element computation makes the model-fitting process easily parallelizable, which is especially useful for multi-processor architectures.

In contrast to other papers in physics-based methods, the mathematical model is presented in a complete form. The systematic approach to Lagrangian dynamics and finite element method adds to the clarity of presentation and effectiveness in the communication of the model.

Despite the strengths of the framework, the goal being to solve the shape recovery problem in isolation from segmentation, knowledge of the object part segmentation is required in advance. The problem of selecting an initial position for the model becomes simpler when one considers a sequence of images. For example, in the case of tracking, only the very first frame needs initialization, which can always be solved by permitting user interaction. For the following frames, the model found in the preceding frame can be used as a good initial position for the analysis of the current frame.

Many of the experiments use synthetic data, and several quantitative experiments are presented along with an error analysis. The performance of the technique, though, should be measured not only by the quality of the final fitted model but also by the amount of time required for the fit. Among the factors that influence these measures are the following:

a priori information for the object, the magnitude and variation of local deformations, the force gain parameters, and the time step used in the Euler method. The initial shape and size (or location) does affect the fitting time, since the closer the initial models are to the shape and size of the shape to be estimated, the less time is required. The performance of the model is also greatly affected, by the values of the magnitude and variation of the local deformations that control the internal forces, and by the weights of the external forces. These weights have to be chosen such that the external forces have comparable magnitude with the internal ones. If the internal forces are dominant, the model will not interpolate the data and it becomes too smooth. On the contrary, if external forces are dominant, the model tends to be insufficiently smooth and fits the noise. Currently, these values are set manually on a trial and error basis. As for the time step, it must be chosen carefully, since the Euler method, although simple and fast, has a limited range of stability. Small time steps can slow down the computation while large time steps result in numerical instabilities. From this discussion, it is clear that the model fitting process is greatly influenced by several model parameters. Guidelines for setting these parameters, including parameter bounds, need to be more fully developed. A next step would also be to develop an extensive error analysis for real data.

The method, in the current stage of development, cannot deal with highlights and texture. However, it can serve as an umbrella for the integration of different qualitative and quantitative modalities. Addressing the problems of integration of segmentation and physics-based fitting techniques to estimate shape of parameterized objects from noisy data under orthographic, perspective and stereo projections is a topic of future research[47, 44]. In principle, using the same framework, researchers may be able also to determine experimentally the stiffness of a material. Based on this information, we could classify the material from which the object is made [52].

6 Discussion

This section contrasts the two approaches presented for modeling a deformable object. In the first approach, Miller has refined the conceptual model to a mass-spring system from which he directly obtained the discrete equations. On the contrary, in the second approach, expressed by the last three papers, the conceptual model is a deformable object made from a continuous medium and the mathematical representation is discretized to find a solution. There are several disadvantages to the use of mass-point-spring model, as opposed to using a continuous medium:

- Changing the solution parameters requires modifying the higher level model. In the second approach, since the method of solution is independent from the higher levels of the model, the efficiency of various numerical solvers can be investigated without affecting the higher level of the program.
- The interaction with other objects in the model is restricted to grid points. On the contrary in the second approach, collision detection tests can be performed between

surfaces without worrying if small objects can penetrate through the cracks.

- The user needs to select from the beginning a grid density suitable for the simulation. In the second approach, the grid density can be chosen automatically by the numerical solver, based on the specific domain or can vary as a function of time or as a function of space.

The choice of the first approach for the modeling introduces excessive implementation detail at the conceptual level and results to discretization artifacts hardwired to the model. This approach is mostly used because it can be implemented on the top of rigid body simulation systems and there are no partial differential equations involved.

The model in the second approach is clean and robust. Moreover, the use of adaptive solvers will allow us to simulate more efficiently systems that require fine sampling for only a few extreme configurations. In addition, the body can be rendered without regard to the numerical discretization.

7 Future Open Problems and Conclusions

I have reviewed four papers from the emerging field of physics-based methods for computer graphics and/or computer vision. Although these methods are not by any means exhaustive of the research done at the area, they give us an intuition about the physics-based methods and a basis for a comparative study of the various approaches.

The papers were presented in such a way as to answer the following questions:

- *What is the model trying to do ?*
- *What are the underlying equations ?*
- *What are the knowns and unknowns in these equations ?*
- *What are the solution techniques ?*

All these questions are important not only in analyzing and understanding a model but also in developing reliable and extensible models.

In computer graphics the issues are object modeling and animation. Miller models biological forms using the mass-spring model. He demonstrates the animation of three specific modes of locomotion of creatures like snakes and worms despite the inherent disadvantages of the model. Specifically, the use of a mass-spring model introduces implementation detail at the conceptual level and results in discretization artifacts hardwired to the model. The geometric models of collision detection and response are not adequate to handle a wide variety of cases. In a more demanding scenario of animation, which would require more accurate interaction with the environment, the weaknesses of the model will become a limiting factor.

Witkin and Welch present a geometric model for linear global deformations, to overcome the problem that the many degrees of freedom of mass-spring systems result in stiff numerical equations. By representing only global deformations that are linear in state, though, they give up the ability to represent a number of natural objects. However, one of the strengths of their model is the ability to directly specify the goal of the animation and to dynamically calculate the motion required to satisfy them. They can handle contradictory constraints with grace, but it is difficult to control the exact motion towards the constraints.

In computer vision, the concern lies with shape and motion estimation. Pentland and Horowitz parameterize whole-body motion using a linear combination of the different modes of vibration of an undeformed superquadric. The formulation allows them to discard the high-frequency low amplitude components without excessive error. Nevertheless, using deformation modes is efficient only for the recovery of smooth, symmetrically deformed objects. Using optic flow data they have been able to obtain over-constrained estimates of rigid and nonrigid motion and track the objects over time, only when they limit inter-frame motion to small rotations and deformations.

Metaxas and Terzopoulos, by successfully combining local with global deformations, overcome the problems of the previous approaches. The proposed framework applies across all geometric primitives and deformations. Using a simulated force technique they are able to fit their dynamic models to sparse, noise-corrupted 2-D and 3-D visual data and track the motions of an object over time. Currently, though, a number of parameters must be set manually on a trial and error basis for the experiments.

The two methods described above estimate shape and attempt to extract global properties - the parameters of the shape. For reliable parameter estimation, though, it is required that all the points used for the estimation belong to the same object, the object for which the parameters are estimated. However, these techniques assume that the 2-D or 3-D vision data have been segmented in a preprocessing step. Segmenting data into different objects, though, is a research problem on its own.

Physics-based models tend to require large amounts of computation. With the advent of more powerful computers and the development of more efficient algorithms, this will tend to be less and less of a problem.

The new paradigm of physics-based models opens new opportunities both in the areas of computer graphics and computer vision. In computer graphics, we can build drawing tools for shape description and develop visualization tools to attain pleasing animations. In addition, we can more efficiently handle interactions between multibody systems and use dynamics to simulate tasks such as riding a bicycle or skiing, which are considered hard with the existing methods. By incorporating dynamics, we can determine the feasibility of certain tasks by measuring forces and torques during simulation.

Physics-based modeling is also applicable, perhaps most importantly, to biomedical applications. Such applications will include modeling and simulation of the physical properties of tissues and organs and shape reconstruction of internal organs or external parts.

Modeling based on physical principles is establishing itself as a potent technique in computer graphics and computer vision. It is a rich and fruitful area for research in terms of both theory and applications. It is important, though, to develop concepts, methodologies, and techniques which will be widely applicable to many types of applications. Physics-based models, while computationally more complex than many traditional models, offer unsurpassed realism in the modeling of natural phenomena.

A Appendix: Constraint-Based Modeling

Several researchers have proposed physics-based constraint methods for controlling animations of rigid and non-rigid bodies. A powerful way to control models is to specify constraints on the geometric configuration of the bodies. Barzel and Barr [9] have introduced three types of constraints on the locations of body points: *point-to-nail*, *point-to-path* and *point-to-point*. The point-to-nail constraint requires that a body point be in a constant location in space. A point-to-path constraint requires that a body follows a pre-specified kinematic path. For the assembly of complex objects a point-to-point constraint may be used where two points are required to stay attached, although the bodies that they belong may move freely. There are two outlooks for the physical interpretation of the constraints:

- *Abstraction of the mechanical mechanisms:* A point-to-point constraint can be thought of as an idealized ball-and-socket joint. Although we don't need to model the exact details of the joint, the forces resulting from a point-to-point constraint result in the same net force and torque that a frictionless physical joint would exert.
- *Exploration:* Sometimes we don't know the underlying physical mechanism. For example, we assume the kinematic path that a body point would follow in order to determine the dynamic response of the rest of the system.

I will describe three methods previously used to constrain physic-based models: the *penalty method*, the *constraint stabilization* and the *dynamic constraints* method. The two latter ones are examples of the generic *Lagrangian constraints method.*

Given a physical system whose state is described by the vector $\mathbf{q}$, we can implicitly impose a holonomic constraint, which defines as states consistent with the constraint the ones that satisfy the equation $C(q,t) = 0$. If multiple constraints are to be met simultaneously, then $\mathbf{C}$ is a vector of constraints.

The Penalty method: Researchers [73, 83] have used this method extensively in the past. This method converts a constrained problem to an unconstrained problem in which deviation from the constraint is penalized. It is equivalent to adding a rubber band to the mechanical system. The penalty force is $\mathbf{F}_{penalty} = -2k\mathbf{C}(\mathbf{q})\frac{\partial \mathbf{C}}{\partial \mathbf{q}}$, where k is the strength of the rubber band and $\mathbf{F}_{penalty}$ points toward the manifold $\mathbf{C}(q) = 0$, when $\frac{\partial \mathbf{C}}{\partial \mathbf{q}}$ is not zero. The constraint force, as a function of the system state, causes the system to find a particular state that balances the forces. This has the disadvantage that the constraint is not guaranteed to be fulfilled. In addition, as penalty strengths increase, the equations of the physical system become stiff, therefore the numerical differential equation solver takes very small steps, consuming computing time without significant progress. The main attraction of the method is that it offers a very simple way to convert a constrained problem into an unconstrained one.

Lagrangian constraints: The penalty method involves a formulation of the problem in which constraints are only approximately satisfied. A completely different viewpoint is taken when the constraints are to be fulfilled exactly. In Lagrangian constraints method, forces that would cause the system to violate the constraint are cancelled independently of the system state, and they are substituted by a force that gradually makes the system to fulfill the constraints. There are two lagrangian constraint methods that have been used: *constraint stabilization* and *dynamic constraints.*

Constraint stabilization: Constrained stabilization has been used by mechanical engineers to correct numerical inaccuracies in systems that are initialized with the constraints fulfilled. If the system begins in a legal state, with $\mathbf{C}(q,t) = \mathbf{0}$ and $\dot{\mathbf{C}}(q,t) = \mathbf{0}$, then requiring $\ddot{\mathbf{C}}(q,t) = \mathbf{0}$ suffices in principle to hold the constraints in force. In practice an additional feedback term must be added to inhibit drift and bring the system to a legal state initially. Baumgarte in [11] suggested the damped second order differential equations

$$\ddot{\mathbf{C}} + \alpha\dot{\mathbf{C}} + \beta\mathbf{C} = \mathbf{0} \tag{8}$$

where α and β are stabilization factors, so that if a system departs from a constraint, it will get pushed back.

To enforce the constraints exactly, forces are added to the mechanical system, which are computed via the Lagrange multipliers. The augmented equations of motion take the form:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_q + \mathbf{f}_q - \mathbf{f}_{g_c} \tag{9}$$

Lagrangian physics states that constraint forces must be in the direction of the gradient of the constraint function, in order to obey the principle of the virtual work [88], therefore $\mathbf{f}_{g_c} = -\mathbf{C}_q^\top\lambda$, where $\mathbf{C}_q^\top$ is the transpose of the constraint Jacobian matrix and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_n^\top)^\top$ is the vector of Lagrange multipliers that must be determined. Combining (8) and (9) we obtain:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_\mathbf{q}^\top \\ \mathbf{C}_\mathbf{q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{D}\dot{\mathbf{q}} - \mathbf{K}\mathbf{q} + \mathbf{g}_q + \mathbf{f}_q \\ \boldsymbol{\gamma} - \alpha\dot{\mathbf{C}} - \beta\mathbf{C} \end{bmatrix}.$$

Notice that the penalty method also creates forces that are in the gradient of the function C(q) but approximates the lagrangian multipliers by $-k\mathbf{C}$. The force becomes zero when the constraint is fulfilled and any external force will pull the system out of the constraint surface.

Constraint stabilization offers the following advantages:

- Regardless of the forces applied to the physical system all the constraints are fulfilled exactly.
- If a physical system starts away from a constraint surface $\mathbf{C}(\mathbf{q}) = \mathbf{0}$, the system can return to the constraint surface.

Using the parameters α and β, we can control the rate by which the constraints are fulfilled. Additionally, we can construct complex objects initially apart or enforce constraints in the middle of animation without sudden jumps. The price to pay for the above flexibility is that the rates α and β are non-intuitive and the system may oscillate around the state in which the constraints are fulfilled, which is unpleasant for the animation.

Dynamic constraints: Barzel and Barr [9] were among the first to use the idea of stabilized constraints [55] for computer graphics modeling and animation with a variation of the constraint stabilization method called *dynamic constraints.* Often the animation starts with the constraints being violated. They assume that a physical system is always very close to fulfilling the constraints. To control the speed of the constraint fulfillment they have suggested using the parameters $\alpha = \frac{2}{t}$ and $\beta = \frac{1}{t^2}$, t being the time constant for the critically dumped motion and $\frac{2}{t}$ being the speed of fulfillment. But they model their constraint forces as $\mathbf{F}_{constraint} = C_k G_{ik}$, where C_k is the strength and G_{ik} is the direction of the constraint force. Therefore it is possible to choose a direction for a constraint force that it is not in the gradient direction, therefore violating the principle of virtual work.

In summary, constraints allow the animators to control the animation. Constraint forces are applied in the direction of the gradient of the constraint violation functions. Using constraint forces that are collinear with the gradient direction assures that the Lagrangian physics is still obeyed. The magnitude of these constraint forces are computed so that the physical simulation approaches the constraint surface with critically damped motion.

References

[1] J. K. Aggarwal and R. O. Duda. Computer Analysis of Moving Polygonal Images. *IEEE Transactions on Computers*, C(24):966–976, 1975.

[2] M. Asada, M. Yachida, and S. Tsuji. Understanding of 3-D Motions in Blocks World. *Pattern Recognition*, 17(1):57–84, 1984.

[3] N. I. Badler. *Computer Graphics I: CIS 560.* University of Pennsylvania, 1992.

[4] N. I. Badler, J. O'Rourke, and B. Kaufman. Special Problems in Human Movement Simulation. *Computer Graphics*, 14(3):189–197, July 1980.

[5] R. Bajcsy and S. Kovačič. Multiresolution Elastic Matching. *Computer Vision, Graphics, and Image Processing*, 46(1):1–21, 1989.

[6] R. Bajcsy, S.W. Lee, and A. Leonardis. Color Image Segmentation with Detection of Highlights and Local Illumination Induced by Inter-reflections. In *Proc. 10th International Conf. on Pattern Recognition*, Atlantic City, NJ, June 1990.

[7] R. Bajcsy, F. Solina, and A. Gupta. Segmentation versus Object Representation - are they separable? In Jain, R. and Jain, A., editors, *Analysis and Interpretation of Range Images.* Springer-Verlag, 1990.

[8] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann Publishers, Los Altos, California, 1987.

[9] R. Barzel and A. Barr. A Modeling System Based on Dynamic Constraints. *Computer Graphics*, 22(4):179–188, 1988. Proc. ACM SIGGRAPH'88.

[10] K.-J. Bathe and E. L. Wilson. *Numerical Methods in Finite Element Analysis.* Prentice–Hall, Englewood Cliffs, NJ, 1976.

[11] J. Baumgarte. Stabilization of Constraints and Integrals of Motion in Dynamical Systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[12] B. M. Bennett and D. D. Hoffman. The Computation of Structure from Fixed Axis Motion: Nonrigid Structures. *Biol. Cybernetics*, 51:293–300, 1985.

[13] T. O. Binford. Visual Perception by Computer. In *IEEE Conference on Systems and Control*, 1971.

[14] A. Blake. Specular Stereo. In *Proc. of 9th Int. Joint Conf. Artif. Intell.*, pages 973–976, Los Angeles, CA, 1985.

[15] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, MA, 1987.

[16] R. M. Bolle and B. C. Vemuri. On Three-Dimensional Surface Reconstruction Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13, January 1991.

[17] T. E. Boult and A. D. Gross. Recovery of Superquadrics from 3-D Information. In *Proc. SPIE Intelligent Robots and Computer Vision*, pages 358–365, November 1987.

[18] N. Burtnyk and W. Wein. Computer Generated Key Frame Animation. *Journal of the SMPTE 80*, pages 149–153, 1971.

[19] C. W. Chen and T. S. Huang. Epicardial Motion and Deformation Estimation from Coronary Artery Bifurcation Points. In *IEEE Proc. Third International Conference on Computer Vision (ICCV'90), Osaka, Japan, Dec. 1990*, pages 456–459, 1990.

[20] S. S. Chen and M. Penna. Recognizing Deformations of Nonrigid Bodies. In *IEEE Computer Vision and Pattern Recognition Conference*, pages 175–207, Miami, FL, 1986. IEEE.

[21] S. S. Chen and M. Penna. Shape and Motion of Nonrigid Structures. *Computer Vision, Graphics, and Image Processing*, 36:175–207, 1986.

[22] I. Cohen, N. Ayache, and P. Sulger. Tracking Points on Deformable Objects Using Curvature Information. In *Proceedings of the Second European Conference on Computer Vision 1992*, pages 458–466, Santa Margherita Ligure, Italy, May 1992.

[23] I. Cohen, L. D. Cohen, and N. Ayache. Using Deformable Surfaces to Segment 3-D Images and Infer Differential Structures. *Computer Vision, Graphics, and Image Processing: Image Undestanding*, 56(2):242–263, 1992.

[24] J. S. Duncan, R. L. Owen, and P. Anandan. Measurement of Nonrigid Motion Using Contour Shape Descriptors. In *IEEE Computer Vision and Pattern Recognition Conference*, pages 318–324, Hawaii, 1991.

[25] A. Gelb (Editor). *Applied Optimal Estimation.* MIT Press, 1989.

[26] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, 1990.

[27] D. B. Goldgof, H. Lee, and T. S. Huang. Motion Analysis of Nonrigid Structures. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR'88)*, pages 375–380, 1988.

[28] A. Gupta. *Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Models.* PhD dissertation, University Of Pennsylvania, Department of Computer and Information Science, June 1991.

[29] G. H. Healey and T. O. Binford. Using Color for Geometry-Insensitive Segmentation. *Journal of the Optical Society of America*, 6, 1989.

[30] I.L. Herlin and N. Ayache. Features Extraction and Analysis Methods for Sequences of Ultrasound Images. In *Proceedings of the Second European Conference on Computer Vision 1992*, pages 43–57, Santa Margherita Ligure, Italy, May 1992.

[31] B. K. P. Horn. Understanding Image Intensities. *Artificial Intelligence*, 8:201–231, 1977.

[32] P. Isaacs and M. Cohen. Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics. *Computer Graphics*, 21(4):215–224, July 1987.

[33] R. Jasinchi and A. Yuille. Nonrigid Motion and Regge Calculus. *Journal of the Optical Society of America*, 6(7):1088–1095, July 1989.

[34] C. Kambhamettu and D. B. Goldgof. Point Correspondence Recovery in Nonrigid Motion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE, IEEE Computer Society Press, 1992.

[35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[36] A. D. Kelley, M.C. Malin, and G. M. Nielson. Terrain Simulation Using a Model of Stream Erosion. *Computer Graphics*, pages 263–268, 1988. Proc. SIGGRAPH.

[37] G.J. Klinker, S.A. Shafer, and T. Kanade. A Physical Approach to Color Image Understanding. *Intern. Journal of Computer Vision*, 4, 1990.

[38] V. Koivunen and R. Bajcsy. Building Procedural CAD Models using 3-D Machine Vision. In *9th International CAD/CAM, Robotics and Factories of the Future Conference*, Newark, NJ USA, August 18-20 1993.

[39] J. U. Korein and N. I. Badler. Techniques for Goal Directed Animation. *IEEE Computer Graphics and Applications*, 2(9):71–81, November 1982.

[40] H.-C. Lee, E. J. Breneman, and C. P. Schulte. Modeling Light Reflection in Computer Vision. *IEEE Trans. PAMI*, 12:402–409, 1990.

[41] S. W. Lee. *Understanding of Surface Reflections in Computer Vision by Color and Multiple Views.* PhD thesis, University of Pennsylvania, 1991.

[42] S.W. Lee and R. Bajcsy. Detection of Specularity Using Color and Multiple Views. *Image and Vision Computing*, 10:643–653, 1992.

[43] F. Leymarie. Tracking and Describing Deformable Objects Using Active Contour Models. Technical Report TR-CIM-90-9, McGill University, Montreal, Quebec, February 1990.

[44] D. Metaxas and S. J. Dickinson. Integration of Quantitative and Qualitative Techniques for Deformable Model Fitting from Orthographic, Perspective, and Stereo Projections. In *International Conference on Computer Vision*, 1993.

[45] D. Metaxas and D. Terzopoulos. Constrained Deformable Superquadrics and Nonrigid Motion Tracking. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR'91), Hawaii*, pages 337–343, 1991.

[46] D. Metaxas and D. Terzopoulos. Dynamic Deformations of Solid Primitives with Constraints. *Computer Graphics*, 26(2):309–312, 1992. Proc. ACM SIGGRAPH '92.

[47] D. Metaxas and D. Terzopoulos. Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1992, to appear.

[48] G. Miller. The Motion Dynamics of Snakes and Worms. *Computer Graphics*, 22(4):169–178, August 1988.

[49] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface Reflection: Physical and Geometrical Perspective. *IEEE Trans. PAMI*, 13:611–634, 1991.

[50] G. Nelson. Juno: A Constraint-Based Graphics System. *Computer Graphics*, 19(3):235–243, 1985.

[51] D. R. Peachey. Solid Texturing of Complex Surfaces. *Computer Graphics*, pages 279–286, 1985. Proc. SIGGRAPH.

[52] A. Pentland. Automatic Extraction of Deformable Part Models. Technical Report Vision-Sciences No 104, MIT Media Lab, 1988.

[53] A. Pentland and B. Horowitz. Recovery of Non-rigid Motion and Structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.

[54] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically-Based Modeling and Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):715–729, 1991.

[55] J. Platt. *Constraint Methods for Neural Networks and Computer Graphics.* PhD thesis, 1989.

[56] W. Reeves. Particle Systems - A Technique for Modeling a Class of Fuzzy Objects. *ACM Transactions on Graphics*, 2:91–108, April 1983.

[57] C. W. Reynolds. Not Bumping into Things. In *SIGGRAPH Course 27 Notes: Developments in Physically-Based Modeling*, pages G1–G13. 1988.

[58] J. O' Rourke and N. Badler. Model-based Image Analysis of Human Motion Using Constraint Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:522–536, 1980.

[59] H. Rushmeier and K. Torrance. The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium. *Computer Graphics*, pages 293–302, 1987. Proc. SIGGRAPH.

[60] L. L. Schumaker. Fitting Surfaces to Scattered Data. In C. K. Chui G. G. Lorentz and L. L. Shumaker, editors, *Approximation Theory II*, pages 203–267. Academic Press, 1976.

[61] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Primitives. *Computer Graphics*, 20(4):151–160, 1986.

[62] S. A. Shafer. Using Color to Separate Reflection Components. *COLOR Research and Application*, 10:210–218, 1985.

[63] K. L. Shelley and D. P. Greenberg. Path Specification and Path Coherence. *Computer Graphics*, 16(3):157–166, July 1982.

[64] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(2):131–146, 1990.

[65] S. Steketee and N. I. Badler. Parametric Keyframe Interpolation Incorporating Kinetic Adjustment and Phrasing Control. *Computer Graphics*, 19(3):255–262, 1985.

[66] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision.* Kluwer Academic Publishers, Boston, MA, 1989.

[67] R. Szeliski and D. Terzopoulos. Physically-Based and Probabilistic Modeling for Computer Vision. In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, pages 140–152, San Diego, July 1991. Society of Photo-Optical Instrumentation Engineers.

[68] H.D. Tagare and R. J. deFigueiredo. Photometric Stereo for Diffuse Non-Lambertian Surface. *IEEE Trans. PAMI*, 13, 1991.

[69] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[70] D. Terzopoulos and K. Fleischer. Deformable Models. *The Visual Computer*, 4(6):306–331, 1988.

[71] D. Terzopoulos and K. Fleischer. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *Computer Graphics*, 22(4):269–278, August 1988.

[72] D. Terzopoulos and K. Waters. Analysis of Dynamic Facial Images using Physical and Anatomical Models. In *IEEE Third International Conference on Computer Vision (ICCV'90), Osaka, Japan, Dec.*, pages 727–732, 1990.

[73] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-Seeking Models and 3D Object Reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1987.

[74] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[75] D. Thalmann, N. M. Thalmann, and J. Gourret. Simulation of Object and Human Skin Deformations in a Grasping Task. *Computer Graphics*, 23(3):21–30, July 1989. Proc. SIGGRAPH '89.

[76] N. Thalmann and D. Thalmann. *Computer Animation: Theory and Practice.* Springer-Verlag, New York, 1985.

[77] S. Ullman. Maximizing Rigidity: The Incremental Recovery of 3-D Structure from Rigid Rubbery Motion. AI Memo 721, MIT, Cambridge, MA, 1983.

[78] Y. Wang and J. Wang. Surface Reconstruction using Deformable Models with Interior and Boundary Constraint. In *Proc. of IEEE Conference on Computer Vision*, pages 300–303, 1990.

[79] A. M. Waxman and K. Wohn. Contour Evolution, Neighborhood Deformation and Image Flow: Textured Surfaces in Motion. In W. Richards and S. Ullman, editors, *Image Understanding.* Ablex, 1984.

[80] J. A. Webb and J. K. Aggarwal. Structure from Motion of Rigid Joined Objects. *Artificial Intelligence*, 19(1):107–130, 1983.

[81] J. Weil. The Synthesis of Cloth Objects. *Computer Graphics*, 20(4):49–54, 1986. Proc. SIGGRAPH.

[82] J. Wilhelms. Using Dynamic Analysis for Realistic Animation of Articulated Bodies. *IEEE Computer Graphics and Applications*, 7(6):12–27, 1987.

[83] A. Witkin, K. Fleischer, and A. Barr. Energy Based Constraints on Parameterized Models. *Computer Graphics*, 21(4):225–232, 1987.

[84] A. Witkin, K. Fleisher, and A. Barr. Energy Constraints on Parameterized Models. *Computer Graphics*, 21(3):225–232, 1987.

[85] A. Witkin, M. Gleicher, and W. Welch. Interactive Dynamics. *Computer Graphics*, 24(2):11–22, March 1990.

[86] A. Witkin and M. Kass. Spacetime Constraints. *Computer Graphics*, 22(4):159–168, 1988.

[87] A. Witkin and W. Welch. Fast Animation and Control of Nonrigid Structures. Technical Report CMU-CS-90-103, School of Computer Science, Carnegie Mellon University, 1990.

[88] J. Wittenberg. *Dynamics of Systems of Rigid Bodies.* Tubner, Stuttgart, 1977.

[89] L. B. Wolff. *Polarization Methods in Computer Vision.* PhD thesis, Department of Computer Science, Columbia University, 1991.

[90] B. Wyvill, C. McPheeters, and Wyvill G. Animating Soft Objects. *The Visual Computer*, 2:235–242, 1986.

[91] L. Yeager and C. Upson. Combining Physical and Visual Simulation - Creation of the Planet Jupiter for the Film 2010. pages 85–93, 1986. Proc. SIGGRAPH.

[92] D. Zeltzer, S. Pieper, and D. Sturman. An Integrated Graphics Simulation Platform. In *Proc. Graphics Interface '89*, 1989.