Dynamic Deformation of Solid Primitives with Constraints

Dimitri Metaxas and **Demetri Terzopoulos**[1]
Department of Computer Science, University of Toronto, Toronto, Ontario, M5S 1A4

Keywords: *Physics-Based Modeling and Animation, Deformable Models, Constraints, Solid Primitives, Finite Elements*

Abstract: *This paper develops a systematic approach to deriving dynamic models from parametrically defined solid primitives, global geometric deformations and local finite-element deformations. Even though their kinematics is stylized by the particular solid primitive used, the models behave in a physically correct way with prescribed mass distributions and elasticities. We also propose efficient constraint methods for connecting these new dynamic primitives together to make articulated models. Our techniques make it possible to build and animate constrained, nonrigid, unibody or multibody objects in simulated physical environments at interactive rates.*

1 Introduction

The graphics literature is replete with solid object representations. Unfortunately, it is not particularly easy to synthesize realistic animation through direct application of the geometric representations of solid modeling [5], and the problems are exacerbated when animate objects can deform. Physics-based animation has begun to overcome some of the difficulties.

We propose a systematic approach for creating dynamic solid models capable of realistic physical behaviors starting from common solid primitives such as spheres, cylinders, cones, or superquadrics. Such primitives can "deform" kinematically in simple ways; for example, a cylinder deforms as its radius or length is changed. To gain additional modeling power we allow the primitives to undergo parameterized global deformations (bends, tapers, twists, shears, etc.) of the sort proposed in [2]. To further enhance the geometric flexibility, we permit local free-form deformations. Our local deformations are similar in spirit to the FFDs of [12], but rather than being ambient space warps [12, 10], they are incorporated directly into the solid primitive as finite element shape functions.

Through the application of Lagrangian mechanics and the finite element method our models inherit *generalized coordinates* that comprise the geometric parameters of the solid primitive, the global and local deformation parameters, and the six degrees of freedom of rigid-body motion. Lagrange equations govern the dynamics, dictating the evolution of the generalized coordinates in response to forces. Thus our models exhibit correct mechanical behaviors and their various geometric parameters assume well-defined physical meanings in relation to prescribed mass distributions, elasticities, and energy dissipation rates.

[1] Fellow, Canadian Institute for Advanced Research
E-mail addresses: dm@cs.toronto.edu; dt@cs.toronto.edu

Published in the Proceedings of SIGGRAPH 92 (Chicago, IL, July, 1992). In *Computer Graphics* Proceedings, Annual Conference Series, 1992, ACM SIGGRAPH, pp. 309–312.

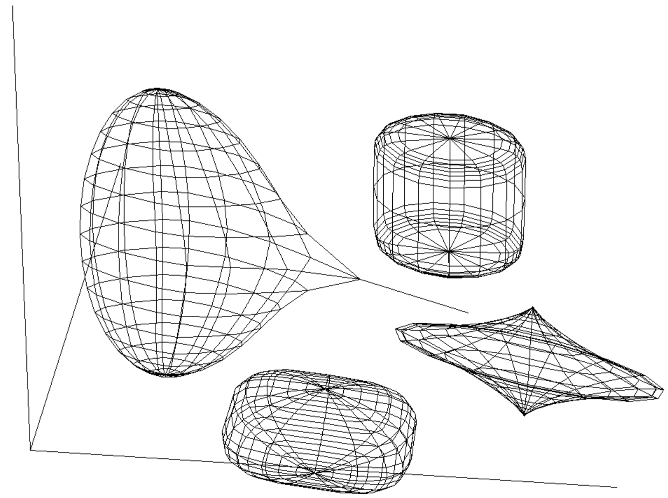


Figure 1: Interaction with deformable superquadrics.

For example, Fig. 1 shows several deformable superquadrics. A superellipsoid [1] is deforming in response to the traction from a linear spring attached to its surface and pulled interactively. In general, the models are abstract viscoelastic solids. It is possible, for instance, to mold a supersphere into any of the deformable superquadrics shown in the figure, not only through manual parameter adjustment but, more interestingly, by applying forces.

A distinguishing feature of our approach is that it marries the parameterized and free-form modeling paradigms within a single physical model. Roughly speaking, we successfully combine locally deformable models [14, 9] with globally deformable models [8, 16]. More precisely, the method applies generally across all geometric primitives and deformations, so long as their equations are differentiable. The coupling of rigid-body and deformation dynamics is similar to that described in [15], but our formulation accommodates global deformations defined by fully nonlinear parametric equations. Hence, our models are more general than the restrictive, linearly deformable ones in [16] and quadratically deformable ones in [8, 10].

Our dynamic deformable models raise interesting challenges related to the application of constraints to construct composite models and control animation. We describe a method for computing generalized constraint forces between our models which is based on Baumgarte's constraint stabilization technique [4, 18]. As in [17, 3], our algorithm may be used to assemble complex objects satisfying constraints from initially mispositioned and misshaped parts, and it enables us to construct and animate articulated objects composed of rigid or nonrigid components.

The remainder of this paper describes our general formulation and presents some results. Space limitations preclude a complete elaboration of the mathematics. We refer the reader to [13, 7] for further details.

2 Geometry

Consider a solid model whose intrinsic (material) coordinates are $\mathrm{u} = (u, v, w)$. Referring to Fig. 2, $\mathbf{x}(\mathrm{u}, t) = (x_1(\mathrm{u}, t), x_2(\mathrm{u}, t), x_3(\mathrm{u}, t))^\top$, where $^\top$ denotes transposition, gives the positions of points on the model relative to the fixed reference frame Φ. We can write

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}, \tag{1}$$

where $\mathbf{p}(\mathrm{u}, t)$ denotes the positions of points relative to the noninertial, model-centered frame ϕ whose instantaneous position is $\mathbf{c}(t)$ and orientation relative to Φ is given by the rotation matrix $\mathbf{R}(t)$. We further express

$$\mathbf{p} = \mathbf{s} + \mathbf{d}, \tag{2}$$

the sum of a reference shape $\mathbf{s}(\mathrm{u}, t)$ and a displacement function $\mathbf{d}(\mathrm{u}, t)$. We define the reference shape as

$$\mathbf{s} = \mathbf{T}(\mathbf{e}(\mathrm{u}; a_1, a_2, \ldots); b_1, b_2, \ldots). \tag{3}$$

Here, a geometric primitive $\mathbf{e}$, defined parametrically in u and parameterized by the variables a_i, is subjected to the *global deformation* $\mathbf{T}$ which depends on the parameters b_i. Although generally nonlinear, $\mathbf{e}$ and $\mathbf{T}$ are assumed to be differentiable (so that we may compute the Jacobian of $\mathbf{s}$) and $\mathbf{T}$ may be a composite sequence of primitive deformation functions. We define the vector of global deformation parameters

$$\mathbf{q}_s = (a_1, a_2, \ldots, b_1, b_2, \ldots)^\top. \tag{4}$$

Next, we express the displacement as a linear combination of basis functions $\mathbf{b}_i(\mathrm{u})$. The basis functions can be local or global; however, finite element shape functions [6] are the natural choice for representing *local deformations*

$$\mathbf{d} = \mathbf{S}\mathbf{q}_d. \tag{5}$$

Here $\mathbf{S}$ is a shape matrix whose entries are the shape functions and

$$\mathbf{q}_d = (\ldots, \mathbf{d}_i^\top, \ldots)^\top \tag{6}$$

is the vector of local deformation parameters. Typically, finite elements have nodes at their vertices, and the parameter $\mathbf{q}_i$ denotes a displacement vector associated with node i of the model.

In [13, 7] we provide the formulas for a superquadric ellipsoid $\mathbf{e}$ with tapering, bending, shearing, and twisting deformations (see also [2]).

3 Kinematics and Dynamics

To convert the above geometric representation into a physical model that responds dynamically to forces, we first consider the kinematics implied by the geometry and then introduce mass, damping, and elasticity into the model to derive its mechanics.

The velocity of points on the model is given by,

$$\dot{\mathbf{x}} = \dot{\mathbf{c}} + \dot{\mathbf{R}}\mathbf{p} + \mathbf{R}\dot{\mathbf{p}} = \dot{\mathbf{c}} + \mathbf{B}\dot{\boldsymbol{\theta}} + \mathbf{R}\dot{\mathbf{s}} + \mathbf{R}\mathbf{S}\dot{\mathbf{q}}_d, \tag{7}$$

where $\boldsymbol{\theta} = (\ldots, \theta_i, \ldots)^\top$ is a vector of rotational coordinates and $\mathbf{B} = [\ldots \partial(\mathbf{R}\mathbf{p})/\partial\theta_i \ldots]$. Now, $\dot{\mathbf{s}} = [\partial\mathbf{s}/\partial\mathbf{q}_s]\dot{\mathbf{q}}_s = \mathbf{J}\dot{\mathbf{q}}_s$, where $\mathbf{J}$ is the Jacobian of the reference shape with

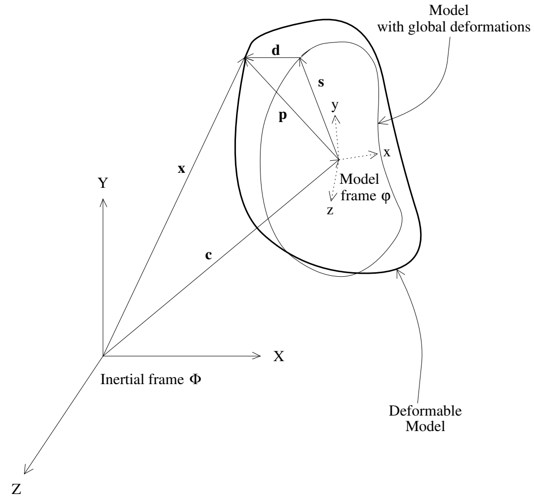


Figure 2: Geometry of deformable model.

respect to the global deformation parameter vector. We can therefore write the model kinematics compactly as

$$\mathbf{x} = \mathbf{c} + \mathbf{R}(\mathbf{s} + \mathbf{d}) = \mathbf{h}(\mathbf{q}), \tag{8}$$

$$\dot{\mathbf{x}} = [\mathbf{I}\ \mathbf{B}\ \mathbf{R}\mathbf{J}\ \mathbf{R}\mathbf{S}]\dot{\mathbf{q}} = \mathbf{L}\dot{\mathbf{q}}, \tag{9}$$

where

$$\mathbf{q} = (\mathbf{q}_c^\top, \mathbf{q}_\theta^\top, \mathbf{q}_s^\top, \mathbf{q}_d^\top)^\top, \tag{10}$$

(with $\mathbf{q}_c = \mathbf{c}$ and $\mathbf{q}_\theta = \boldsymbol{\theta}$) serves as the vector of generalized coordinates for the dynamic model.

To specify the dynamics, we introduce a mass distribution $\mu(\mathrm{u})$ over the model and assume that the material is subject to frictional damping. We also assume that the material may deform elastically or viscoelastically [14]. From Lagrangian mechanics we obtain second-order equations of motion which take the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_\mathbf{q} + \mathbf{f}_\mathbf{p}. \tag{11}$$

The mass matrix $\mathbf{M} = \int \mu \mathbf{L}^\top \mathbf{L}\, d\mathrm{u}$. The stiffness matrix $\mathbf{K}$ may be obtained from a deformation strain energy $(\mathbf{q}^\top \mathbf{K}\mathbf{q})/2$. The Raleigh damping matrix $\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K}$. The generalized inertial forces $\mathbf{g}_\mathbf{q} = -\int \mu \mathbf{L}^\top \dot{\mathbf{L}}\dot{\mathbf{q}}\, d\mathrm{u}$ include generalized centrifugal, Coriolis, and transverse forces due to the dynamic coupling between $\mathbf{q}_\theta$, $\mathbf{q}_s$, and $\mathbf{q}_d$. Finally, $\mathbf{f}_\mathbf{p} = \int \mathbf{L}^\top \mathbf{f}\, d\mathrm{u}$ are generalized external forces associated with the components of $\mathbf{q}$, where $\mathbf{f}(\mathrm{u}, t)$ is the force distribution applied to the model. See [13, 7] for explicit formulas for the above matrices and vectors.

4 Constrained Nonrigid Motion

We can extend (11) to account for the motions of composite models with interconnected deformable parts. Shabana [11] describes the well-known Lagrange multiplier method for multibody objects. We form a composite generalized coordinate vector $\mathbf{q}$ and force vectors $\mathbf{g}_\mathbf{q}$ and $\mathbf{f}_\mathbf{p}$ for an n-part model by concatenating the $\mathbf{q}_i$, $\mathbf{g}_{\mathbf{q}_i}$, and $\mathbf{f}_{\mathbf{p}_i}$ associated with each part $i = 1, \ldots, n$. Similarly, the composite matrices $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ for the n-part model are block diagonal matrices with submatrices $\mathbf{M}_i$, $\mathbf{D}_i$, and $\mathbf{K}_i$, respectively, for each part i. The method solves the composite equations of motion $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g}_\mathbf{q} + \mathbf{f}_\mathbf{p} - \mathbf{C}_\mathbf{q}^\top\boldsymbol{\lambda}$. The generalized constraint forces $\mathbf{f}_{g_c} = -\mathbf{C}_\mathbf{q}^\top\boldsymbol{\lambda}$ acting on the parts stem from the holonomic constraint equations

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0}; \tag{12}$$

i.e., $\mathbf{C} = [\mathbf{C}_1^\top, \mathbf{C}_2^\top, \ldots, \mathbf{C}_k^\top]^\top$ expresses k constraints among the n parts of the model. The term $\mathbf{C}_\mathbf{q}^\top$ is the transpose of the constraint Jacobian matrix and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_n^\top)^\top$ is a vector of Lagrange multipliers that must be determined.

To obtain an equal number of equations and unknowns, we differentiate (12) twice with respect to time, yielding

$$\boldsymbol{\gamma} = \mathbf{C}_\mathbf{q}\ddot{\mathbf{q}} = -\mathbf{C}_{tt} - (\mathbf{C}_\mathbf{q}\dot{\mathbf{q}})_\mathbf{q}\dot{\mathbf{q}} - 2\mathbf{C}_{\mathbf{q}_t}\dot{\mathbf{q}}. \quad (13)$$

We obtain the augmented equations of motion

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_\mathbf{q}^\top \\ \mathbf{C}_\mathbf{q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{D}\dot{\mathbf{q}} - \mathbf{K}\mathbf{q} + \mathbf{g}_\mathbf{q} + \mathbf{f}_\mathbf{p} \\ \boldsymbol{\gamma} \end{bmatrix}. \quad (14)$$

In principle, these equations may be integrated from initial conditions $\mathbf{q}(0)$ and $\dot{\mathbf{q}}(0)$ satisfying $\mathbf{C}(\mathbf{q}(0), 0) = \mathbf{0}$ and $\dot{\mathbf{C}}(\mathbf{q}(0), 0) = \mathbf{0}$. At each time step, we solve (14) for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ with known $\mathbf{q}$ and $\dot{\mathbf{q}}$, and then we integrate $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ from t to $t + \Delta t$ to obtain $\dot{\mathbf{q}}$ and $\mathbf{q}$ (e.g., using the simple Euler method $\dot{\mathbf{q}}^{(t+\Delta t)} = \dot{\mathbf{q}}^{(t)} + \Delta t\, \ddot{\mathbf{q}}^{(t)}$; $\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \Delta t\, \dot{\mathbf{q}}^{(t+\Delta t)}$).

There are two practical problems in applying (14) to animation. First, the various parameters of the parts may not be set initially so that the constraints are satisfied (i.e., $\mathbf{C}(\mathbf{q}, 0) \neq \mathbf{0}$). Second, even if the constraints may be satisfied at a given time step of the animation (i.e., $\mathbf{C}(\mathbf{q}, t) = \mathbf{0}$), they may not be satisfied at the next time step (i.e., $\mathbf{C}(\mathbf{q}, t + \Delta t) \neq \mathbf{0}$) because of numerical errors, etc.

The constraint stabilization method of Baumgarte [4, 18] remedies these problems. The constraint equation $\mathbf{C} = \mathbf{0}$ is replaced in (14) by the damped second-order equation $\ddot{\mathbf{C}} + 2\alpha\dot{\mathbf{C}} + \beta^2\mathbf{C} = \mathbf{0}$, where α and β are stabilization factors. This replaces the lower entry of the vector on the rhs of (14) to $\boldsymbol{\gamma} - 2\alpha\dot{\mathbf{C}} - \beta^2\mathbf{C}$. Fast stabilization means choosing $\beta = \alpha$ to obtain the critically damped solution $\mathbf{C}(\mathbf{q}, 0)e^{-\alpha t}$ which, for given α, has the quickest asymptotic decay towards constraint satisfaction $\mathbf{C} = \mathbf{0}$. Baumgarte constraint stabilization and its variations are also applied in [3, 9, 16].

The Lagrange multiplier method is very general, but it is potentially expensive for our deformable models since the matrix in (14) is large. We have devised a fast specialized method to solve for the unknown constraint forces $\mathbf{f}_{g_c}$ arising from point-to-point constraints. The method involves the solution of a linear system whose size is on the order of the number of constraints, which is usually small. In this sense, it is similar to the dynamic constraint technique of [3], but it is suitable for nonrigid parts. The details are in [7].

5 Animation Examples

The partitioning of complex nonrigid behavior into rigid-body motions, global deformations, and local deformations leads to numerically well-conditioned discrete equations and stable simulation algorithms based on explicit numerical integration methods. We represent the rotation component of the models using quaternions, which facilitates the integration of $\mathbf{q}_\theta$. We do not assemble and factorize a finite element stiffness matrix as is common practice in finite element analysis, but instead compute $\mathbf{K}\mathbf{q}_d$ efficiently in an element-by-element fashion. For greater efficiency, we can lump masses to obtain a diagonal $\mathbf{M}$, and we may assume mass-proportional damping, i.e. $\mathbf{D} = \nu\mathbf{M}$ where ν is the damping coefficient [6].

The following animation examples involve deformable superquadric primitives that interact with one another and their simulated physical environments through point-to-point constraints, collisions, gravity, and friction against impenetrable surfaces. To attain animation rates of several frames per second (on an SGI 4D-35TG), we have implemented dynamic "shells" where the material domain is restricted to a membrane surface $\mathrm{u} = (u, v, 1)$ and the interior mass density $\mu(\mathrm{u}) = 0$ for $0 \leq w < 1$. We triangulate the surface of the model into linear elastic elements (see [13] for the details).

Fig. 3 shows several frames from an animation of multiple "balloon pendulums" suspended in gravity by inextensible strings. The simulation starts from the initial configuration shown in Fig. 3(a), and the balloons swing, collide, and deform until the kinetic energy is dissipated. The inelastic collisions are implemented using reaction constraints [9] between multiple deformable bodies.

Fig. 4 shows the construction and animation of a minimalist dragonfly from its deformable body parts (Fig. 4(a)) and 4 point-to-point constraints. The dragonfly self-assembles [17, 3] and it "works," inasmuch as internal forces open and flap the wings (Fig. 4(b)). An impenetrable plane appears out of nowhere and swats the dragonfly in the rear (Fig. 4(c)). The body parts deform in response to the blow, but they remain joined by the constraints (Fig. 4(d)).

Fig. 5 illustrates the self-assembly and animation of a snowman. Twelve point-to-point constraints assemble and hold the deformable superquadric body parts together (Fig. 5(a–b)). When gravity is turned on, the assembled snowman drops to the impenetrable floor and locomotes along a prespecified path through controlled bouncing (Fig. 5(b–d)).

Fig. 6 illustrates a simple "circus stunt," in which two deformable superquadric balls interact with a pivoting springboard mounted on immobile planes. The simulation starts from Fig. 6(a) with the yellow ball dropping downward. Fig. 6(b) shows the motion tracks.

The figures have demonstrated a rather sophisticated family of parameterized models—deformable superquadrics with global twisting, bending, tapering, and shearing deformations, and local membrane deformations. We emphasize in closing, however, that our approach is generally applicable to a wide variety of parameterized models, global deformations, and finite element basis functions.

References

[1] Barr, A., (1981) "Superquadrics and angle preserving transformations," *IEEE Computer Graphics and Applications*, **1**(1), 11-23.

[2] Barr, A., (1984) "Global and local deformations of solid primitives," *Computer Graphics*, **18**(3), 21–30.

[3] Barzel, R., and Barr, A., (1988) "A modeling system based on dynamic constraints," *Computer Graphics*, **22**(4), 179-188.

[4] Baumgarte, J., (1972) "Stabilization of constraints and integrals of motion in dynamical systems," *Comp. Meth. in Appl. Mech. and Eng.*, **1**, 1–16.

[5] Hoffmann, C.M., (1989) *Geometric and solid modeling*, Morgan-Kaufmann, Palo Alto.

[6] Kardestuncer, H., (ed.), (1987) *Finite element handbook*, McGraw–Hill, New York.

[7] Metaxas, D., and Terzopoulos, D., (1992) "Shape and nonrigid motion estimation from synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

[8] Pentland, A., and Williams, J., (1989) "Good vibrations: Modal dynamics for graphics and animation," *Computer Graphics*, **23**(3), 215–222.

[9] Platt, J., (1989) "Constraint methods for neural networks and computer graphics," PhD Thesis, Dept. of Computer Science, California Institute of Technology, Pasadena, CA (Caltech-CS-TR-89-07).

[10] Sclaroff, S., and Pentland, A., (1991) "Generalized implicit functions for computer graphics," *Computer Graphics*, **25**(4), 247–250.

[11] Shabana, A., (1989) *Dynamics of multibody systems*, Wiley, New York.

[12] Sederberg, T.W., and Parry, S.R., (1989) "Free-form deformation of solid geometric primitives," *Computer Graphics*, **20**(4), 151–160.

[13] Terzopoulos, D., and Metaxas, D., (1991) "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(7), 703–713.

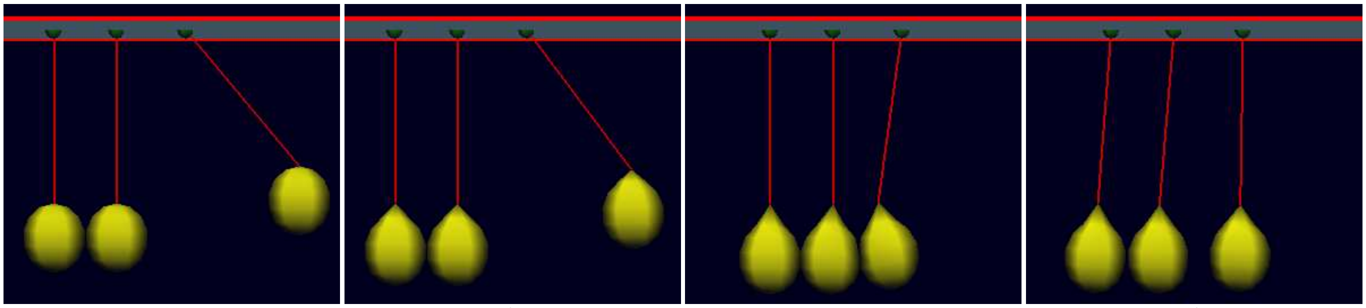
[14] Terzopoulos, D., and Fleischer, K., (1988) "Deformable models," *The Visual Computer,* **4**(6), 306–331.

[15] Terzopoulos, D., and Witkin, A., (1988) "Physically-based models with rigid and deformable components," *IEEE Computer Graphics and Applications,* **8**(6), 41–51.

[16] Witkin, A., and Welch, W., (1990) "Fast animation and control of nonrigid structures," *Computer Graphics*, **24**(3), 243–252.

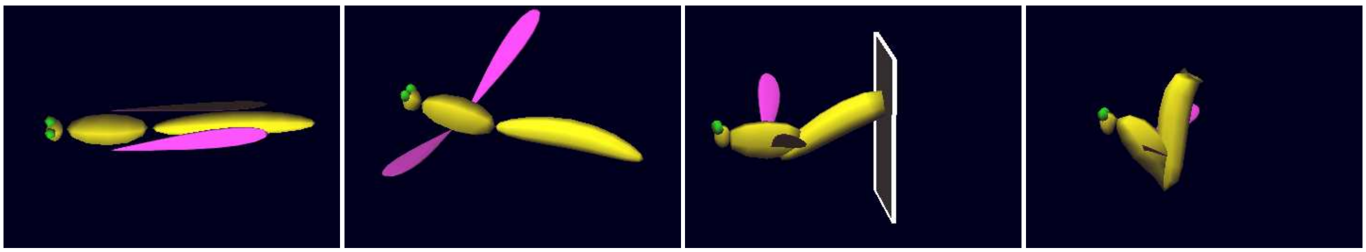
[17] Witkin, A., Fleischer, K., and Barr, A., (1987) "Energy constraints on parameterized models," *Computer Graphics*, **21**(4), 225–232.

[18] Wittenburg, J., (1977) *Dynamics of systems of rigid bodies*, Tubner, Stuttgart.



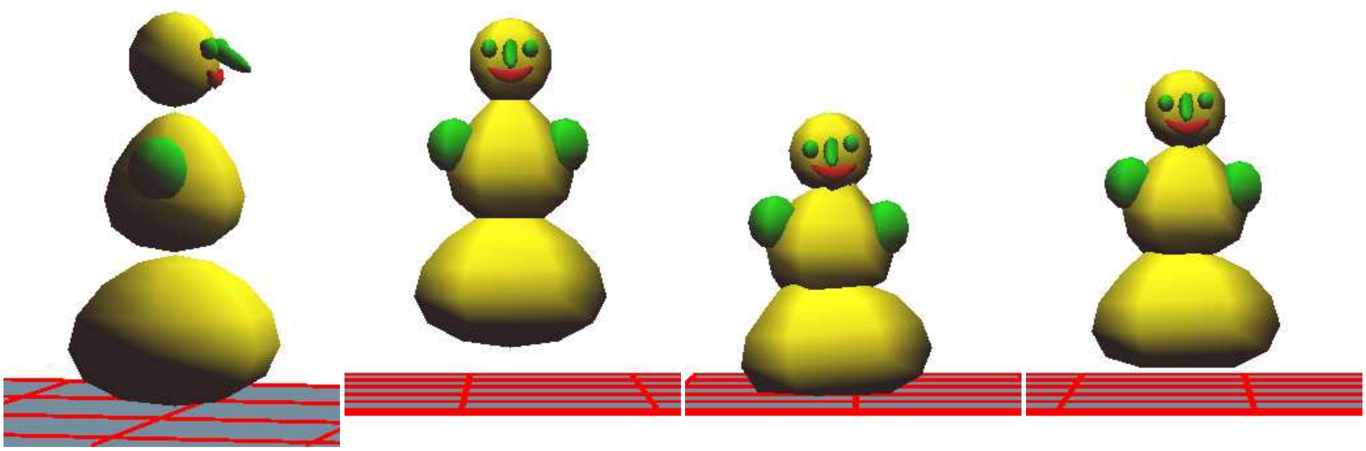
(a) (b) (c) (d)

Figure 3: Balloon multi-pendulum. Initial state (a). Swinging and colliding (b–d).



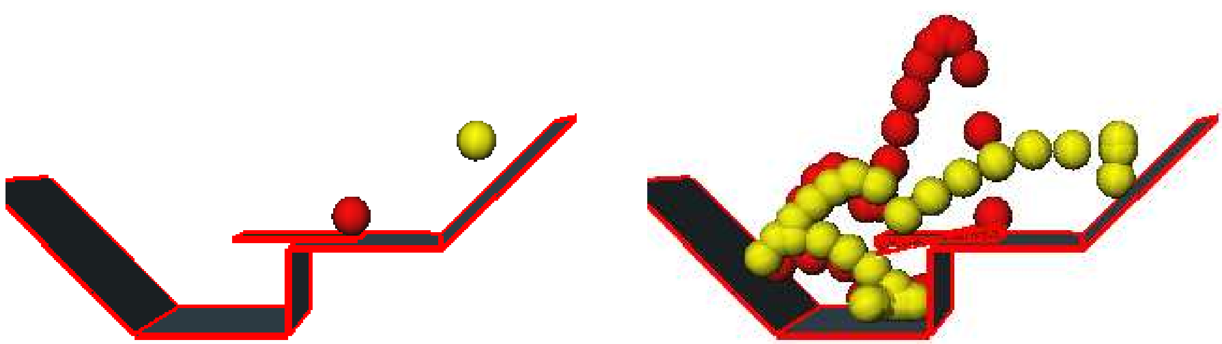
(a) (b) (c) (d)

Figure 4: Dragonfly. Self-assembly (a). Flight (b). Swatting (c). Swatted fly (b).



(a) (b) (c) (d)

Figure 5: Yellow snowman. Self-assembly (a). Hopping (b–d).



(a) (b)

Figure 6: Balls and springboard. Initial state (a). Motion tracks (b).