Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2004)
R. Boulic, D. K. Pai (Editors)

Quasi-Rigid Objects in Contact†

Mark Pauly[1] Dinesh K. Pai[2] Leonidas J. Guibas[1]

[1] Computer Science Department, Stanford University, Stanford CA 94305, USA
[2] Computer and Information Sciences, Rutgers University, Piscataway, NJ 08854, USA

Abstract

We investigate techniques for modeling contact between quasi-rigid objects – solids that undergo modest deformation in the vicinity of a contact, while the overall object still preserves its basic shape. The quasi-rigid model combines the benefits of rigid body models for dynamic simulation and the benefits of deformable models for resolving contacts and producing visible deformations. We argue that point cloud surface representations are advantageous for modeling rapidly varying, wide area contacts. Using multi-level computations based on point primitives, we obtain a scalable system that efficiently handles complex contact configurations, even for high-resolution models obtained from laser range scans. Our method computes consistent and realistic contact surfacesand traction distributions, which are useful in many applications.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling: Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism: Animation

1. Introduction

Contact is ubiquitous in real world interactions and is one of the most difficult problems in 3D computer animation. Physical models used to address contact fall into two broad categories: Rigid body models and deformable body models.

Rigid body models are the most widely used today, despite the fact that they cannot reproduce visible deformations and surface tractions. This is due to their apparent efficiency and relative simplicity: A rigid body's configuration is characterized by only six variables, its velocity requires an additional six variables, leading to a small system of differential equations to integrate, if the forces on the body are known. We review this work in Section 2. This happy story changes dramatically when contact is involved, and some serious problems must be addressed by the "simple" rigid body model. Objects make contact at multiple points, and the rigid body model implies that these contacts are very sensitive to small changes in object state. The contact state, i.e., which contacts are active and which break, has to be determined very carefully since the resulting wrench (i.e., force and torque) on the object depends on it. Even though it is now well understood how to determine the contact state using LCP formulations (see Section 2), the fundamental sen-

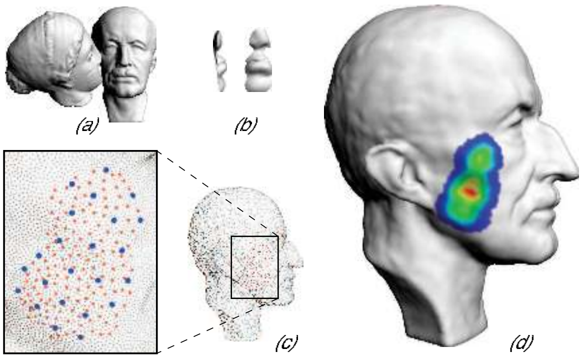


Figure 1: *A romance of two computer graphics classics. (a) the Igea and Max Planck models in contact, (b) the intersection volume, (c) sampling pattern with active nodes, (d) normal tractions on the deformed model*

† The authors wish to acknowledge support form NSF grants CARGO-0138456, ITR-0205671, IIS-0308157, EIA-0215887, as well ARO grant DAAD19-03-1-0331.

sitivity of the contact state remains as it is part of the physical model. This sensitivity problem becomes much worse when trying to simulate contact with complex scanned objects such as those we use in this paper – the contact state can wobble over the large number of surface points, like a stool with hundreds of legs.

To address this problem while retaining the advantages of rigid bodies, two alternatives have been explored: The first is to use *impulse-based methods* that treat even continuous contact as a type of impact and use a coefficient of restitution to determine the post-impact velocity of the object [Hah88, MW88, MC95, GBF03]. This approach is easier to formulate, but could require a large number of steps to resolve the resulting impact sequences. It has its own problems when trying to simulate resting contact with complex models and the wobbling problem remains. More fundamentally, the coefficient of restitution is physically meaningful only for particles, and it has been difficult to define a suitable quantity for rigid bodies [MC95]. Even for something as simple as a metal rod, the measured coefficient of restitution can vary over most of the feasible range [SH96]. The other alternative is to introduce virtual springs and dampers at the contact points (e.g., [GPS94]), to remove the contact state sensitivity. Optimization-based animation [MS01] has similar advantages. However, without an underlying model of elasticity, it is difficult to assign physically meaningful spring constants (or objective functions), and the effective compliance of the contact is again sensitive to the contact state. Rigid body contact turns out to be rather complex after all. This is not surprising, since the assumption of rigidity makes the contact problem highly singular, requiring discontinuous jumps in contact forces for small changes in position. In reality, all objects are deformable to a certain extent, and contact produces stresses that lead to deformations which may be small but significant.

This may suggest that the solution lies in using fully deformable models; such models have a long history in computer graphics (which we briefly review in Section 2). However, fully deformable models (including mass-spring particle systems) can be expensive, since the size of the system state is large all the time, even when the body is not undergoing visible deformations. This is unfortunate, since many real world objects appear to be rigid except at high stress regions near the contact, and the distribution of mass changes very little. Multiresolution and adaptive methods can address this problem to some extent [CGC*02, GKS02, JP03], but require meshes with good connectivity and relatively complex data structures. More subtly, deformable models in computer graphics have been used primarily to produce visual deformations, and not to solve contact problems. Contact has been treated by using user-specified boundary conditions (e.g., [JP99]) or by using penalty forces (e.g., [HFS*01]). True contact problems are much more difficult [KO88] because we do not know in advance whether a possible contact point on the surface will actually be in contact or break free from the surface; its contact state depends on the boundary conditions at all other possible contact points and we must determine *all the boundary conditions* simultaneously. Dynamically deformable objects such as mass-spring systems could, in principle, determine the contact boundary conditions by time stepping, but this requires sufficient temporal resolution to resolve the pressure waves traveling in the solid at the speed of sound.

Our Contributions

With the above motivation, we have developed a new type of model that we call **quasi-rigid**, which can combine the benefits of rigid body models for dynamic simulation and the benefits of deformable models for resolving contacts and producing visible deformations. By quasi-rigid we mean objects whose surfaces can undergo modest deformations in the vicinity of a contact, while the overall object still preserves its basic shape. There are a lot of objects that could be modeled in this way, including biological manipulators such as our hands and feet and everyday objects that appear rigid visually. We have developed a new set of representations and algorithms for modeling the geometry and physics of contacts between quasi-rigid objects. Our method is specifically designed to handle contact between complex geometric shapes obtained from laser scans, using sample-based surface representations to add detail to the deformable contact patch as needed. It can resolve the contact hierarchically to satisfy performance requirements.

More generally, by explicitly computing the contact surface and distributed contact tractions, we open up new areas for physically-based computer graphics. Understanding the exact extent of the contact areas, the traction distributions over them, and their evolution over time is interesting and significant in many applications, ranging from dynamic simulations and computer animation to engineering and biomedicine.

2. Prior Work

2.1. Physical Models

Rigid body dynamics is widely used in computer graphics [Hah88, MW88, MC95, MS01, GBF03]. Baraff was the first to systematically treat contact problems [Bar89, Bar91, Bar94].

A fundamental feature of contact (and friction) is that it leads naturally to an inequality constrained problem. This can be formulated as a minimization of a quadratic functional subject to inequality constraints (e.g., [Bar89]). The necessary condition for optimality is a Linear Complementarity Problem (LCP) [Mur88, CPS92], and it is often simpler to solve it directly [Lot82, Bar89, Bar94]. Coulomb friction adds two complications. First, the classical friction cone leads to a quadratic constraint, but we can recover an

LCP by approximating the friction cone with a polyhedral cone [Lot82]. The second, more serious, complication is that the *dynamic LCP* formulated using acceleration and forces as the unknowns is not guaranteed to have a solution, or can have multiple solutions. This is a problem with the rigid body model itself [Pai95] and determining whether a solution exists is NP-complete [Bar91]. However, it was soon realized that by changing the problem to allow impulses (or equivalently, using forces and accelerations integrated over a time step) one can obtain solutions [Bar91, ST96]. We note that in Section 4.2 we formulate our contact problem as an LCP too, but ours is an *elastic LCP* which is always solvable.

Physically based deformable models for computer graphics were pioneered by [TPBF87], and this area has been active ever since. Early work is surveyed in [GM97]. The recent focus has been on interactive simulation [JP99, DSB99, DDCB01], and specialized methods for important cases such as cloth, biological tissues, and fracture [BW98, HFS*01, OBH02, BFA02]. Contact is treated in most of the papers cited above but, as we mentioned earlier, it is usually done by assuming contact boundary conditions. Contact problems have been addressed more carefully in engineering [Joh85, KO88], but their focus has been on obtaining accurate solutions for very simple and idealized geometries.

There have been a few methods suggested for combining the benefits of rigid bodies and deformable bodies. [TW88] considered how non-linearities in large displacement elasticity problems could be approximated by a rigid displacement plus a linear elastic model, but did not address the general contact problem. Cani (Gascuel) [GVP91, Gas93] — see also [SP95] — addressed contact and volume preservation, but used a simple elasticity model based on a scalar potential field. The model fits well with the implicit surface representation used, but is difficult to relate to the physics of 3D elastic deformation and to material properties. Contact surfaces were computed but did not account for the important coupling between deformations at different locations in a contact patch, as we do. In robotics, a model based on a rigid core with elastic deformation lumped at the contact point was introduced by Kumar and colleagues [WK94], and generalized to distributed contact in [SK03] using an LCP formulation similar to ours. However the local contact patch was represented differently, as a grid of lumped spring-dampers and was limited to simple surfaces. Baraff and Witkin [BW92] proposed a model also aimed at combining the advantages of rigid and deformable models, but based on global deformations; our work proceeds in the opposite direction, focusing on adding local deformation to a rigid body in the contact region.

2.2. Surface Models

The modeling of the geometry of contact surfaces does not easily fit into existing geometric modeling techniques. Consider triangle meshes, for example. Most commonly such meshes have irregular connectivity and thus it is difficult to get them to conform exactly in the contact area, as the two sides have incompatible connectivity graphs. One can try to compute a single contact mesh which is a common refinement of both sides of the contact, but as this process is repeated across deformation time steps, the elements in the contact area will become unreasonably small through the successive refinements. Even with resampled surfaces where the meshes can be made mostly regular, it may be difficult to get the nodes from the two sides to line up without causing excessive distortion to the triangles.

To overcome these problems we use a purely sample-based approach. Surfaces are represented by unstructured point clouds, i.e., sets of point primitives that sample the position and normal of the underlying surface. We use the moving least squares (MLS) surface model to define a smooth continuous surface from a set of point samples (see [Lev03] or [ABCO*01] for details). Given an unstructured point cloud P as input, the corresponding MLS surface S is defined as the stationary set of a projection operator $\Psi_P(\mathbf{x})$, i.e., $S = \{\mathbf{x} \in \mathbb{R}^3 | \Psi_P(\mathbf{x}) = \mathbf{x}\}$. The projection $\Psi_P(\mathbf{x})$ is evaluated by first fitting a local least squares plane that serves as a local parametrization domain. A second least squares optimization then yields a bivariate polynomial $g(u,v)$ that locally approximates the surface. The projection of $\mathbf{x}$ onto S is given as $\Psi_P(\mathbf{x}) = \mathbf{q} + g(0,0) \cdot \mathbf{n}$, where $\mathbf{q}$ is the origin and $\mathbf{n}$ the normal of the reference plane. Both least squares fits use a positive, monotonically decreasing weight function, typically a Gaussian or low degree polynomial, that adjusts the influence of the sample points according to the distance to $\mathbf{q}$. We use the linear approximation of the MLS method presented in [PGK02] that dynamically adapts the radius of the weight function to the local sample spacing and thus allows robust reconstruction for irregular point clouds with varying sampling density.

The main advantage of the MLS method for our purposes is that it is *meshless*. By avoiding global structural information such as an explicit connectivity graph or a parameterization, highly dynamic contact events can be modeled efficiently and a unique consistent contact surface can be defined without the need for re-triangulation or zippering (see also Section 4.4)

3. Modeling Quasi-Rigid Objects

Our goal is to model contact, possibly over a wide area, between quasi-rigid solids. Figure 2 shows a conceptual model of such an object. A point force acting on a quasi-rigid solids leads to a deformation that is restricted to a small, local area (active region) around the point of contact, while keeping the overall shape intact. This motivates the use of local, analytical models for linear elastic materials (Section 3.1). Since we are mainly interested in the contact region, we choose to work with boundary-based elasticity formulations. The resulting system response functions (Green's functions) are discretized using a collocation scheme. We will show in Section 3.2 how this approach can be integrated nicely with surfaces represented by point clouds, leading to an efficient implementation.

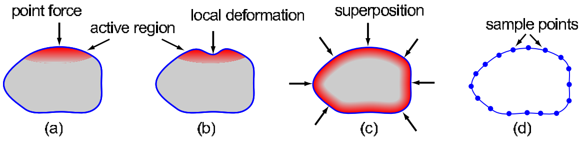


Figure 2: *Conceptual model of quasi-rigid objects. (a) point force acting on quasi-rigid body, (b) local deformation, (c) deformable layer, (d) point-based surface representation.*

3.1. Physical Model

Since we anticipate deformations to be small, our system is based on the physics of static linear elasticity. Our focus is on contacts between quasi-rigid objects, so we need to find the displacements and tractions that act on the *surfaces* of the objects. This could be done by solving a global boundary value problem to compute Green's functions as in [JP99], but here we seek a suitable approximation that can be efficiently evaluated locally. To keep the exposition simple, we only consider tractions p and displacements u that act normal to the surface. We will come back to the corresponding tangential quantities in Section 5.1, where we consider friction effects.

A widely used method in contact mechanics is the *Boussinesq approximation*, which models the surface around a point of contact as an elastic half-space [Joh85]. Suppose a normal traction $p(\mathbf{x})$ is applied at a point $\mathbf{x}$ on the surface. We define a local coordinate frame with $\mathbf{x}$ as the origin, the z-axis given by the surface normal $\mathbf{n}$ at $\mathbf{x}$ and x- and y-axes defined by two arbitrary (but fixed) vectors perpendicular to $\mathbf{n}$ and to each other. According to the Boussinesq formula, the displacement $u(\mathbf{y},\mathbf{x})$ at a point $\mathbf{y}$ on the surface in the normal direction due to a point load $p(\mathbf{x})$ is given as

$$u(\mathbf{y},\mathbf{x}) = \frac{1-\nu}{2\pi G}\frac{p(\mathbf{x})}{\parallel \mathbf{y}-\mathbf{x} \parallel}, \tag{1}$$

where ν is Poisson's ratio and G is the shear modulus. Since we assume linear elasticity, the total displacement $u(\mathbf{y})$ due to a distribution of pressure on the whole surface S can be found by superposition:

$$u(\mathbf{y}) = \frac{1-\nu}{2\pi G}\int_S \frac{p(\mathbf{x})}{\parallel \mathbf{y}-\mathbf{x} \parallel} d\mathbf{x}. \tag{2}$$

Figure 3 (a) shows the system response function $f(r) = \frac{1-\nu}{2\pi G}\frac{1}{r}$ of the Boussinesq approximation for a single point load, which exhibits a rapid fall-off with increasing distance $r = \parallel \mathbf{y}-\mathbf{x} \parallel$.

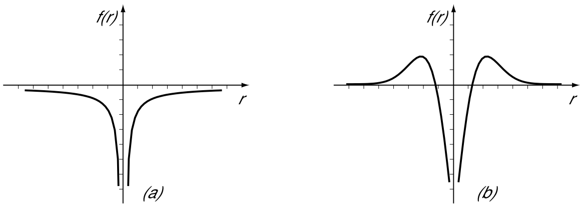


Figure 3: *System response functions. (a) Boussinesq approximation, (b) an empirical model that preserves volume.*

3.1.1. Volume Preservation

Since the Boussinesq model is based on the assumption of an infinite elastic half-space, it does not guarantee preservation of volume. However, many real-world materials, such as human tissue, are almost incompressible and thus effects due to a finite volume need to be considered for a realistic simulation. This can be achieved by ensuring that the deformation due to a point load does not lead to a change of volume, i.e., that $\int_S u(\mathbf{y},\mathbf{x})d\mathbf{y} = 0$. Then, due to linearity, volume preservation is ensured for any pressure distribution acting on the surface. Using integration in polar coordinates, volume preservation can be formulated as the following constraint on the response function $f(r)$:

$$\int_0^\infty f(r)rdr = 0. \tag{3}$$

Of course, not all response functions obeying this condition are physically valid. A plausible analytical function can be obtained by shifting the Boussinesq function in the positive z-direction and modulating it with a Gaussian as shown in Figure 3 (b). More realistic response functions could either be obtained through physical measurements or through off-line simulations using high-resolution finite element or mass-spring systems. Figure 4 illustrates volume preservation for quasi-rigid objects on a simple example of a ball in contact with a plane, using the transfer function of Figure 3 (b).

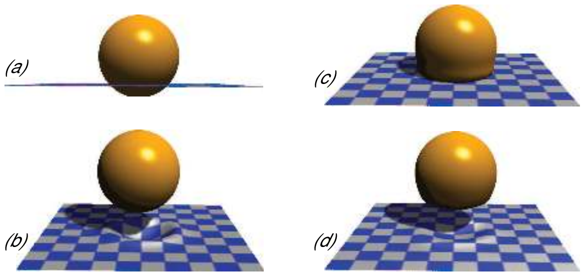


Figure 4: *Volume preservation. (a) original configuration prior to contact resolution, (b) plane deformable, sphere rigid, (c) sphere deformable, plane rigid, (d) both models deformable.*

3.2. Discretization

In a discrete setting, we need to find relations between the displacement u_i that node $\mathbf{q}_i$ experiences due to the tractions p_j acting on all other nodes. This can be expressed in matrix form as

$$\mathbf{u} = \mathbf{R}\mathbf{p}, \tag{4}$$

where $\mathbf{u} = [u_1 \dots u_N]^T$ is the vector of displacements, $\mathbf{p} = [p_1 \dots p_N]^T$ describes the corresponding tractions, and $\mathbf{R}$ is the system response matrix. To determine the entries of $\mathbf{R}$ we discretize the tractions using scattered data approximation as

$$p(\mathbf{x}) \approx \sum_{j=1}^{N} p_j \Phi_j(\mathbf{x}), \tag{5}$$

where Φ_j is the shape function

$$\Phi_j(\mathbf{x}) = \frac{\omega_j(\parallel \mathbf{x} - \mathbf{q}_j \parallel)}{\sum_{k=1}^{N} \omega_k(\parallel \mathbf{x} - \mathbf{q}_k \parallel)}, \tag{6}$$

for certain kernel functions ω_k. Then u_i can be expressed as

$$u_i = \frac{1-\nu}{2\pi G} \int_S \sum_j \frac{p_j \Phi_j(\mathbf{x})}{\parallel \mathbf{q}_i - \mathbf{x} \parallel} d\mathbf{x} = \frac{1-\nu}{2\pi G} \sum_j p_j \int_S \frac{\Phi_j(\mathbf{x})}{\parallel \mathbf{q}_i - \mathbf{x} \parallel} d\mathbf{x} \tag{7}$$

and it follows that

$$\mathbf{R}_{ij} = \frac{1-\nu}{2\pi G} \int_S \frac{\Phi_j(\mathbf{x})}{\parallel \mathbf{q}_i - \mathbf{x} \parallel} d\mathbf{x}. \tag{8}$$

As kernel functions ω_k we use truncated Gaussians. Their radius is chosen so that kernel functions of neighboring points overlap to achieve a hole-free coverage of the surface, similar to rendering algorithms that use Gaussian reconstruction kernels for surface splatting [ZPvG01]. Note that the singular integrals in $\mathbf{R}_{ii}$ are bounded [Joh85] and that all entries of $\mathbf{R}$ can be pre-computed for fixed sampling distributions. Due to the locality of the response function, the response matrix $\mathbf{R}$ will generally be sparse, contrary to global methods for general deformable objects that require transformations and thresholding to expose the sparsity [JP03].

4. Contact Resolution

Given the model for quasi-rigid objects defined above, our goal is to determine the contact surface S of two solids A and B and compute the forces that act on this surface. The bodies A and B are represented by two point clouds P_A and P_B that define two corresponding MLS surfaces S_A and S_B. During a simulation, the two bodies might collide and interpenetrate as illustrated in Figure 7. To resolve this collision we need to find a local deformation on each body, such that both objects touch without interpenetration. We determine this deformation and the resulting contact surface as follows: First we check if the two bodies intersect. If a collision is detected, we compute two sets of *active nodes* Q_A and Q_B, i.e., those surface points of each model that potentially experience a displacement due to the deformation that resolves the interpenetration (see Section 4.1). For the sets of active nodes we assemble the system response matrices as described in Section 3.2. Then we set up linear complementarity constraints that define the contact surface and solve for the tractions that act on this surface. From these tractions we compute the corresponding displacements and the total wrench that acts on the quasi-rigid bodies (see Section 4.2).

4.1. Collision Detection

To determine the set of active nodes Q_A (analogously for Q_B) we first find all points of P_A that are inside the volume bounded by S_B. For this purpose we use the inside/outside classification algorithm presented in [PKKG03], which approximates the signed distance function induced by a point-sampled surface. To classify a point $\mathbf{r}_A \in P_A$ with respect to the surface S_B, we find the projected point $\mathbf{x}_B$ of $\mathbf{r}_A$ on S_B. The approximate signed distance is then given as $d(\mathbf{r}_A, S_B) = (\mathbf{r}_A - \mathbf{x}_B) \cdot \mathbf{n}_B$, where $\mathbf{n}_B$ is the outward pointing normal at $\mathbf{x}_B$. If $d(\mathbf{r}_A, S_B) < 0$ we classify $\mathbf{r}_A$ as intersecting.

The set of intersecting points is not sufficient to define the active nodes for contact resolution, since tractions on these points can lead to displacements outside of the intersecting region. To determine the required additional points we make use of the fact that the system response function drops quickly with increasing distance (see Section 3.1). Contrary to fully deformable objects where all points potentially experience a significant displacement, we can thus confine the active region to a local neighborhood around the penetrating samples. The additional points can then be found with a simple range query[†].

As described in [PKKG03], finding the point $\mathbf{x}_B \in S_B$ for inside/outside classification requires a costly MLS projection. We can speed up these computations significantly by observing that we are not required to compute the intersection region precisely. As long as we get a conservative estimate (that might include some points that are not penetrating, but close to the intersection region), we are guaranteed to find all points that potentially experience a displacement. Thus for classification we search for the closest point $\mathbf{r}_B \in P_B$ instead of $\mathbf{x}_B \in S_B$. This amounts to a piecewise linear approximation of the signed distance function of S_B, requiring only a closest point query instead of a full MLS projection. To further improve performance, we use a bounding sphere hierarchy for fast intersection culling. This spatial data structure is built using the clustering method described in [PGK02]. The point set P_A is divided into disjoint clusters and for each of these clusters a tightly fitting bounding sphere is computed. A nested hierarchy can be obtained by repeating this process recursively.

[†] When using a volume preserving response function a slightly different approach needs to be applied (see the discussion in Section 6.1).

4.2. Linear Complementarity Formulation

Given the sets of active nodes Q_A and Q_B, we need to find tractions on these points such that the corresponding displacements lead to deformed surfaces S'_A and S'_B that are in contact, but do not interpenetrate. Let $S = S'_A \cap S'_B$ be the (as yet unknown) contact surface between the deformed models. If the two surfaces touch at a point $\mathbf{x} \in S$, the corresponding normal tractions at this point on each of the two surfaces are equal, i.e., $p_A(\mathbf{x}) = p_B(\mathbf{x})$ (we are only considering the equilibrium state). Note that we use the convention that normal tractions are pointing inwards, i.e., $p_A(\mathbf{x})$ and $p_B(\mathbf{x})$ are defined with respect to different local coordinate systems. Let $\mathbf{q}_A \in Q_A$ be an active node of model A and assume for now that after deformation this point will end up on the final contact surface. Then there exists a *corresponding point* $\mathbf{x}_B \in S_B$ such that its position in the final configuration coincides with the deformed position of $\mathbf{q}_A$ and the normal tractions at both points are equal. Let Q'_B be the set of all corresponding points of the active nodes of model A and let Q'_A be defined analogously. Furthermore let $\tilde{Q}_A = Q_A \cup Q'_A = \{\tilde{\mathbf{q}}^1_A, \ldots, \tilde{\mathbf{q}}^N_A\}$ and $\tilde{Q}_B = Q_B \cup Q'_B = \{\tilde{\mathbf{q}}^1_B, \ldots, \tilde{\mathbf{q}}^N_B\}$ be two sets that are ordered such that $\tilde{\mathbf{q}}^i_A$ and $\tilde{\mathbf{q}}^i_B$ are a pair of corresponding points for all i, where $N = |\tilde{Q}_A| = |\tilde{Q}_B|$.

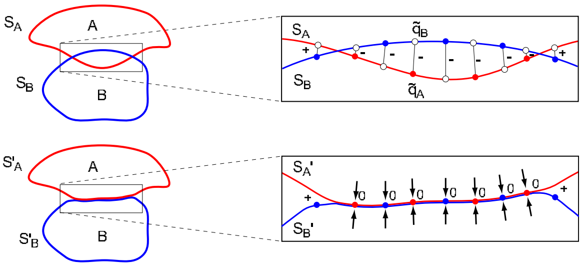


Figure 5: *Contact resolution in 2D. The top image shows the active nodes prior to displacement, where the signs indicate the initial separation of corresponding points. After contact resolution, all separations are non-negative and points on the contact surface experience non-zero tractions, indicated by two opposing arrows.*

In the following all quantities that are defined for model A have their analog for model B. We define a vector of tractions on $\tilde{Q}_A$ as $\mathbf{p}_A = [p^1_A \ldots p^N_A]^T$, such that p^i_A is the normal traction at node $\tilde{\mathbf{q}}^i_A$. These tractions will cause displacements, which will be denoted as $\mathbf{u}_A = [u^1_A \ldots u^N_A]^T$. Using the system response matrix $\mathbf{R}_A$ for $\tilde{Q}_A$ as defined in Section 3.2, the displacements $\mathbf{u}_A$ are related to the tractions $\mathbf{p}_A$ by $\mathbf{u}_A = \mathbf{R}_A \mathbf{p}_A$.

Our goal is to describe the deformed surfaces in contact without interpenetration (this is called the Signorini Problem [KO88]). It follows immediately that $\mathbf{p}_A = \mathbf{p}_B$. Additionally, we can formulate conditions on the displacements. We define the separation of two corresponding nodes $\tilde{\mathbf{q}}^i_A$ and $\tilde{\mathbf{q}}^i_B$ as

$$u^i_A + u^i_B + \sigma_i \parallel \tilde{\mathbf{q}}^i_A - \tilde{\mathbf{q}}^i_B \parallel, \tag{9}$$

where $\sigma_i = -1$, if $\mathbf{q}^i_A$ intersects the volume bounded by S_B, and $\sigma_i = 1$, if no interpenetration occurs. Note also that u^i_A and u^i_B are defined with respect to opposing normals. Non-penetration implies $s^i \geq 0$. If we assume that the objects can only push and not pull on each other, then the traction $p^i = p^i_A = p^i_B$ has to satisfy $p^i \geq 0$ (this is commonly known as the "no velcro" assumption). Finally, we observe that the separation s^i and the traction p^i are *complementary* – at each point, at least one of them is zero; to be able to have non-zero traction, we must have zero separation, and to have non-zero separation, we must have zero traction. Therefore, we can combine these *complementarity conditions* to obtain

$$\begin{aligned} \mathbf{s} &= \mathbf{R}\mathbf{p} + \mathbf{q} \\ \mathbf{s} &\geq 0 \\ \mathbf{p} &\geq 0 \\ \mathbf{s}^T\mathbf{p} &= 0, \end{aligned} \tag{10}$$

where $\mathbf{s} = [s^1 \ldots s^N]^T$, $\mathbf{R} = \mathbf{R}_A + \mathbf{R}_B$, and $\mathbf{q} = [\sigma_1 \parallel \tilde{\mathbf{q}}^1_A - \tilde{\mathbf{q}}^1_B \parallel \ldots \sigma_N \parallel \tilde{\mathbf{q}}^N_A - \tilde{\mathbf{q}}^N_B \parallel]^T$ (see Figure 5). This is a linear complementarity problem (LCP) and techniques for solving such problems have been extensively studied [Mur88, CPS92]. In general, an LCP may not have a solution or its solution may not be unique. However, in our case, because the matrix $\mathbf{R}$ is co-positive (in the Boussinesq case it is easy to see that it is positive), a solution can always be found by Lemke's method.

Note that the point correspondences are used both for computing the initial separation $\parallel \mathbf{q}^i_A - \mathbf{q}^i_B \parallel$ and for assembling the system response matrix $\mathbf{R}$. The crucial question is how can we actually find the corresponding points, since this seems to require knowledge of the unknown final contact surface. We found that, instead of trying to solve Equation 10 directly, we can drastically simplify the computations by splitting the combined LCP into two LCPs, each defined separately on the active nodes of each object alone. Thus we only need to assemble a response matrix for each of the sets of active nodes Q_A and Q_B instead of for the extended sets $\tilde{Q}_A$ and $\tilde{Q}_B$. The underlying assumption is that the distance metrics on both models in the active region are similar and thus the separate response matrices will describe a similar physical behavior. To estimate the initial separation we compute the minimum distance of an active node of A to the surface of B (and vice versa). Thus we avoid having to find explicit point correspondences altogether. However, with these simplifications we are not guaranteed to resolve the interpenetration completely. The displacements will consistently be too small, since we use a lower bound on the initial separation. To find the correct solution of Equation 10 we iterate the above scheme until the penetration is resolved (up to some epsilon separation). We found that for all practical purposes, three iterations were sufficient. The fast convergence can be explained by the fact that the closer the two surfaces get the more accurate will both the approximation of the system response matrices and the estimate for the initial separation be. So instead of solving a single LCP of size $|Q_A| + |Q_B|$, we iteratively solve two LCPs of size $|Q_A|$ and $|Q_B|$. Since Lemke's method has at least quadratic complexity [Mur88], this iterative scheme can even be faster than solving the single larger system.

4.3. Hierarchical Representation

Even with the above optimizations, the cost of computing the contact surface and corresponding normal tractions can be significant for densely sampled models or large contact areas, as it depends critically on the number of active nodes. To improve performance and obtain a scalable algorithm suitable for interactive applications, we extend the single-level computations described above to a multi-level scheme based on a spatial clustering hierarchy. Each level of this hierarchy represents the model surface at a different resolution, similar to mesh-based multi-resolution representations used in surface editing, e.g., [ZSS97]. To build the hierarchy of surface approximations we use the same spatial data structure we described for accelerating the collision queries (see Section 4.1). For each cluster at a certain level, we choose the center of the bounding sphere of the cluster's sample points as a representative sample (see also Figure 1). The radius of the bounding sphere determines the radius of the kernel function used to compute the system response matrix (Section 3.2). The computations of Section 4.2 are then performed at the coarsest level first and the resulting tractions are propagated to the next finer level using interpolation. As soon as the desired accuracy is obtained this recursive scheme is terminated, allowing us to trade off speed versus accuracy, which is particularly important in realtime applications.

4.4. Contact Surface Model

As described above, the active region comprises all surface points that experience a displacement due to the contact resolution forces. The parts of the active region that are actually in contact are specified by all active nodes that have zero final separation (or, due to complementarity, non-zero traction). The fact that we have point-wise compliance does not guarantee that both surfaces conform exactly in the region of contact, however. Assume, for example, that both surfaces were represented as triangle meshes. Even though the distance between both meshes might be zero at all the mesh vertices, there could still be intersections or gaps between the two surfaces (the same holds for two separate MLS surfaces). To obtain a single conforming contact surface, i.e., to ensure that the surfaces of both models agree exactly in the region of contact, we make use of the meshless, implicit surface definition of the MLS projection. At any time instance, the surface of model A is represented by its original points P_A plus all active nodes of B that lie on the contact surface (analogously for model B) as shown in Figure 6. In the interior of the contact region both surfaces will thus coincide exactly, since the same points are used in the MLS optimization, as indicated by the neighborhood region N_2. Closer to the boundary of the contact area these neighborhoods will be different (indicated by N_1 in Figure 6) and hence the surfaces will diverge. Even though we have no formal proof at this time, extensive experimentation confirms that no interpenetrations occur in these regions between the MLS surfaces. Note that the increased sampling density in the contact region is easily handled by the adaptive MLS approximation that dynamically chooses the neighborhood radius according to the local sample spacing (see Section 2.2).

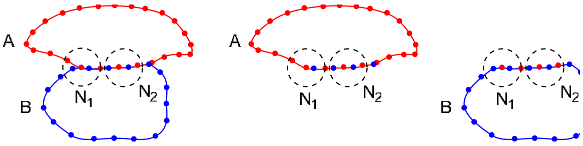


Figure 6: *Contact surface model. Both objects share active nodes that have zero separation.*

5. Simulation Framework

To test our contact resolution method in a dynamic simulation, we have implemented a prototype of a quasi-rigid body simulator as illustrated in Figure 7. Our method replaces the contact handling code of standard rigid body simulators, where dynamic contacts are typically resolved by applying a finite impulse at the moment of impact. The quasi-rigid scheme avoids such singularities by integrating the forces that act on the contact surface. Whenever a collision between two objects is detected, we retrace the simulation to the time instance of first contact and proceed with a smaller time step. Note that the first contact time does not have to be isolated as precisely as with rigid body contact, due to the compliance that is built into the model. Even though we used explicit integration in our experiments, implicit integration can be done efficiently with quasi-rigid bodies since the Jacobian of the force function needed for implicit integration [BW98] is nothing but the stiffness matrix of the contact, and can be computed directly from the response matrix $\mathbf{R}$ restricted to the nodes in contact.

In each step, we resolve the interpenetration, compute the forces acting on the contact surface and update the total wrenches, i.e., the forces and torques, acting on the bodies. During this dynamic simulation we make extensive use of the multi-level approach of Section 4.3. Since we are mostly interested in the total wrench, we can solve for the contact forces on a coarse scale. We found that for a typical configuration, e.g., the one shown in Figure 11, 40 to 100 active nodes per model were sufficient.

Note that, in accordance with the quasi-rigid model, we do not have to take the dynamic effects of body deformations into account, e.g., we use a fixed inertia tensor for each body during the whole simulation. Since the deformations are generally small this does not lead to any perceivable artifacts. Also, we assume that bodies resume their original, undeformed shape after breaking contact.

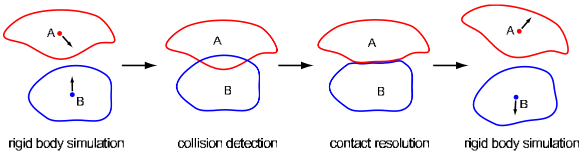


Figure 7: *Contact resolution in quasi-rigid body simulation.*

5.1. Friction

A major benefit of explicitly modeling the contact surface is that we can accurately simulate effects that are primarily dictated by contact, such as friction. So far we have only considered tractions that act normal to the surface, i.e., the frictionless case. Friction could be added to the elastic LCP (Eq. 10) in essentially the same way as it is added to dynamic LCPs [ST96], using a polyhedral approximation to the friction cone. But this increases the size of the LCP significantly by adding $m+1$ pairs of complementary variables at each contact point, where m is the number of facets of the friction cone. However, if we make the simplifying assumption that tangential tractions do not cause significant deformations, we can decouple tangential and normal quantities during contact resolution. This greatly simplifies the computations. As a proof of concept, we implemented a simple dynamic Coulomb friction model, where the tangential friction force at a point of contact is proportional to the normal traction that acts at this point [Bar91]. Our experiments show that this assumption is reasonable, yet a more quantitative analysis remains to be done. Figure 8 shows how angular momentum of a spinning ball is transformed into linear motion due to friction.

6. Results

We tested our contact resolution method both for dynamic and static contacts on a variety of models of different complexity. All computations were performed on an Intel Pentium IV with 2.8 GHz and 1 Gb main memory.

Figure 9 shows an example of a large contact between a rigid ground plane and a quasi-rigid foot. The latter is a laser-range scan of a human foot, consisting of 55,061 sample points of which 11,317 experience a displacement due to the contact resolution. The contact computations take 952ms using a three level hierarchy. Observe the bulging due to volume preservation (Section 3.1.1). Also note that this point cloud is unprocessed scanner data, still containing holes and a fair amount of measurement noise. This example demonstrates that our method can be applied to a wide range of input models and is not limited to synthetic or highly preprocessed data. We also tried to validate our model by comparing the computed tractions with physical measurements of the real foot as shown in (e) and (f). Measurements were performed using the X2 FootSensor System (manufactured by XSensor Corp.) with a resolution of 37 by 13 measurement points, with 1.94 sensors/cm^2. Even though our model does not account for non-linearities in human tissue and the articulation of toes and bones in the foot, we still get very plausible results.

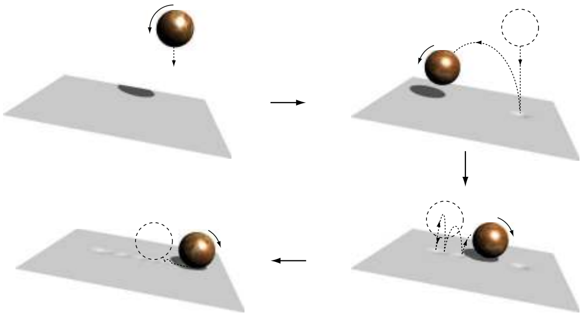


Figure 8: *A spinning ball is dropped onto an inclined plane. The dotted line shows the trajectory of the ball, the curved arcs illustrate the angular velocity. Both objects are quasi-rigid and for illustration purposes the deformations of the plane are maintained.*

An illustration for a medical application is given in Figure 10. The femur ball joint (44,513 points) is brought into contact with a hip bone (141,221 points), which leads to a complex conforming contact surface, illustrating the robustness of our method in difficult contact configurations. This type of analysis is useful in the design of artificial joints or material parts, where the distribution of tractions can be used to predict material wear.

Figure 11 shows three frames of a dynamic sequence simulated with our quasi-rigid body simulator. The head of Max Planck (52,809 points) moves around in a cubic room and frequently comes into contact with the non-planar walls. Both models deform on impact and the resulting forces cause the head model to bounce off the walls. Total contact resolution time varies between 20 and 200ms depending on the configuration. This includes collision detection using a four level hierarchy and typically involves 40 to 100 active nodes. Even though the timings are not yet suitable for realtime simulations, we strongly believe that further optimizations will greatly improve the performance of our method (see Section 6.1).

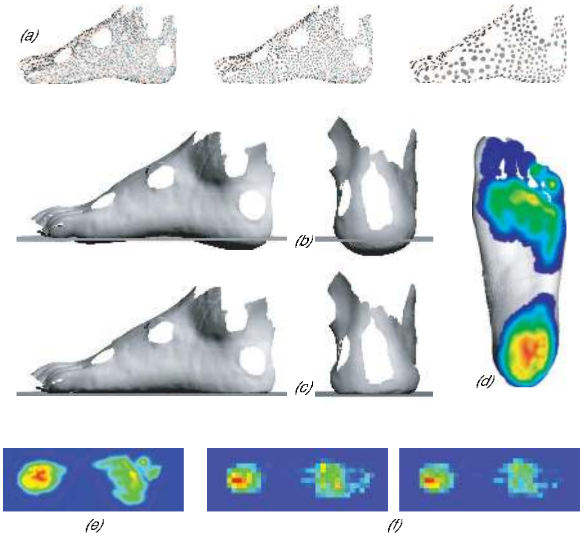


Figure 9: *Contact of a deformable human foot with a rigid ground plane. (a) sampling patterns of the hierarchical representation used for collision detection and multi-level computations, (b) original, undeformed configuration, (c) final deformed surface, (d) normal tractions on foot, (e) normal tractions on ground plane, (f) measured tractions. Blue denotes zero traction and red maximum traction.*

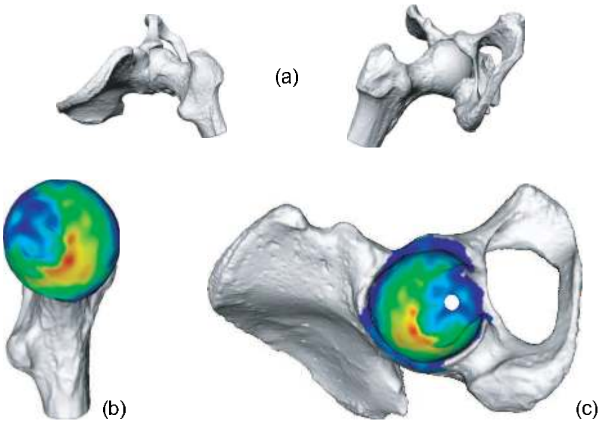


Figure 10: *Complex conforming contact. (a) original configuration, (b) tractions on balljoint, (c) tractions on hip bone.*

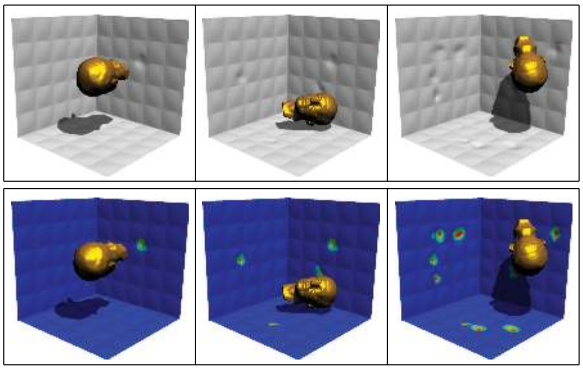
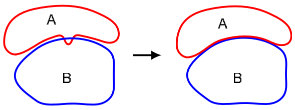


Figure 11: *Max Headroom. The top shows the maximum displacements on the walls, the bottom the corresponding tractions.*

6.1. Discussion

When using system response functions that guarantee volume preservation (see Section 3.2), the size of the resulting contact surface can be significantly bigger than the actual interpenetration region (see illustration).



In such cases our collision detection method (Section 4.1) will fail to produce sufficiently large sets of active nodes. However, we can easily detect such a situation by checking for nodes on the boundary of the active region that have zero separation after contact resolution, i.e., lie on the contact surface. In this case we can successively enlarge the active region until the boundary points experience zero displacement.

Crucial to achieving high performance during dynamic simulation is temporal coherence. So far we only exploit temporal coherence during collision detection and computation of the active nodes. It has been previously observed, however, that significant performance gains can be achieved in the LCP algorithm (Lemke's method) itself [Bar93]. Since this part dominates the computational overhead, we expect substantial speedups by reusing parts of the computations over multiple time steps.

We were surprised about the robustness of our method for complex conforming contacts such as the example of Figure 10. However, regions of extreme curvature such as spikes are difficult to handle with our method. The reason is that we assume that interpenetrations can be resolved by normal displacements only. To avoid self-intersections the deformable layer must be very thin in regions of high curvature, which in the limit reduces such areas to completely rigid bodies. To address this problem and make the system more widely applicable we propose to combine our surface-based method with a coarse volumetric representation such as a low resolution FEM mesh. This allows us to support large scale deformations at low cost, while still preserving a highly detailed surface representation for accurate contact modeling.

The geometry and physics at contact surfaces can have a profound influence on how a simulation evolves and significantly affect other visible parts of the overall behavior. For example, an uneven distribution of frictional tractions over a contact area can create a torque that can cause an object to flip around – a common cause of highway accidents.

Determining the traction distributions over the contact area can be important in computer animation of walking and running [HWBO95] and biomedical visualization. We believe that understanding the exact extent and temporal evolution of contact areas and the forces acting on them is an interesting and significant aspect of many simulation and animation applications.

7. Conclusions and Future Work

We have introduced a new model of quasi-rigid bodies and presented an efficient method for handling contacts between such objects, bridging the gap between rigid body models and fully deformable FEM or mass-spring models. Using hierarchical representations and multi-level computations our method is both applicable in complex, high resolution contact configurations as well as dynamic quasi-rigid body simulations. We found that point primitives can be used effectively to represent wide-area conforming contact surfaces. This allows an efficient analysis of the evolution of contact surfaces over time.

We believe that our work opens up many possible directions for future research. More advanced physical models could replace the Boussinesq approximation, e.g., physically acquired system responses can easily be integrated into our system. Many aspects of dynamic simulations, such as resting contact vs. dynamic contact or advanced friction models, can now be analyzed and improved using our contact resolution method as a framework. The evaluation of contact forces could also improve haptic feedback during interactive applications such as surgery simulation.

References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *Proceedings of Visualization '01* (2001). 3

[Bar89] BARAFF D.: Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proceedings of SIGGRAPH 1989* (1989), pp. 223–232. 2

[Bar91] BARAFF D.: Coping with friction for non-penetrating rigid body simulation. *Computer Graphics 25*, 4 (1991), 31–40. 2, 3, 8

[Bar93] BARAFF D.: Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica 10* (1993), 292–352. 9

[Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of SIGGRAPH 1994* (1994), pp. 23–34. 2

[BFA02] BRIDSON R., FEDKIW R. P., ANDERSON J.: Robust treatment of collisions, contact, and friction for cloth animation. *ACM Transactions on Graphics 21*, 3 (2002), 594–603. 3

[BW92] BARAFF D., WITKIN A.: Dynamic simulation of non-penetrating flexible bodies. In *Proceedings of SIGGRAPH 1992* (1992), pp. 303–308. 3

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of SIGGRAPH 1998* (1998), pp. 43–54. 3, 7

[CGC*02] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: A multiresolution framework for dynamic deformations. In *ACM SIGGRAPH Symposium on Computer Animation* (2002), pp. 41–48. 2

[CPS92] COTTLE R., PANG J., STONE R.: *The Linear Complementarity Problem*. Academic Press, Inc., 1992. 2, 6

[DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of SIGGRAPH 2001* (2001), pp. 31–36. 3

[DSB99] DESBRUN M., SCHRÖDER P., BARR A.: Interactive animation of structured deformable objects. In *Graphics Interface '99* (1999), pp. 1–8. 3

[Gas93] GASCUEL M.-P.: An implicit formulation for precise contact modeling between flexible solids. In *Proceedings of SIGGRAPH 93* (Aug. 1993), Computer Graphics Proceedings, Annual Conference Series, pp. 313–320. 3

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Transactions on Graphics 22*, 3 (2003), 871–878. 2

[GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: CHARMS: A simple framework for adaptive simulation. *ACM Transactions on Graphics 21*, 3 (2002), 281–290. 2

[GM97] GIBSON S. F., MIRTICH B.: *A Survey of Deformable Models in Computer Graphics*. Tech. Rep. TR-97-19, MERL, Cambridge, MA, 1997. 3

[GPS94] GOYAL S., PINSON E. N., SINDEN F. W.: Simulation of dynamics of interacting rigid bodies including friction. *Engineering with Computers 10* (1994), 162–195. 2

[GVP91] GASCUEL M.-P., VERROUST A., PUECH C.: Animation and collisions between complex deformable bodies. In *Graphics Interface '91* (June 1991), pp. 263–270. 3

[Hah88] HAHN J. K.: Realistic animation of rigid bodies. *Computer Graphics 22*, 4 (1988), 299–308. 2

[HFS*01] HIROTA G., FISHER S., STATE A., LEE C., FUCHS H.: An implicit finite element method for elastic solids in contact. In *Proceedings of Computer Animation 2001* (Seoul, 2001). 2, 3

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of SIGGRAPH 1995* (1995), pp. 71–78. 10

[Joh85] JOHNSON K. L.: *Contact Mechanics*. Cambridge University Press, Cambridge, 1985. 3, 4, 5

[JP99] JAMES D. L., PAI D. K.: Artdefo: accurate real time deformable objects. In *Proceedings of SIGGRAPH 1999* (1999), pp. 65–72. 2, 3, 4

[JP03] JAMES D., PAI D. K.: Multiresolution green's function methods for interactive simulation of large-scale elastostatic objects. *ACM Transactions on Graphics 22*, 1 (2003), 47–82. 2, 5

[KO88] KIKUCHI N., ODEN J.: *Contact problems in elasticity : a study of variational inequalities and finite element methods*. SIAM, Philadelphia, 1988. 2, 3, 6

[Lev03] LEVIN D.: Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization* (2003). 3

[Lot82] LOTSTEDT P.: Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal on Applied Mathematics* (1982). 2, 3

[MC95] MIRTICH B., CANNY J.: Impulse-based simulation of rigid bodies. In *1995 Symposium on Interactive 3D Graphics* (1995), pp. 181–188. 2

[MS01] MILENKOVIC V. J., SCHMIDL H.: Optimization-based animation. In *Proceedings of SIGGRAPH 2001* (2001), pp. 37–46. 2

[Mur88] MURTY K. G.: *Linear complementarity, linear and nonlinear programming*. Heldermann Verlag, Berlin, 1988. 2, 6, 7

[MW88] MOORE M., WILHELMS J.: Collision detection and response for computer animation. In *Proceedings of SIGGRAPH 1988* (1988), pp. 289–298. 2

[OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics 21*, 3 (2002), 291–294. 3

[Pai95] PAINLEVE P.: Sur les lois du frottement de glissement. *Comptes Rendus de l'Academie des Sciences 121* (1895). 3

[PGK02] PAULY M., GROSS M., KOBBELT L. P.: Efficient simplification of point-sampled surfaces. In *Proceedings of Visualization '02* (2002), pp. 163–170. 3, 5

[PKKG03] PAULY M., KEISER R., KOBBELT L. P., GROSS M.: Shape modeling with point-sampled geometry. *ACM Transactions on Graphics) 22*, 3 (2003), 641–650. 5

[SH96] STOIANOVICI D., HURMUZLU Y.: A critical study of the applicability of rigid-body collision theory. *ASME Journal of Applied Mechanics 63* (1996), 307–316. 2

[SK03] SONG P., KUMAR V.: Distributed compliant model for efficient dynamic simulation of systems with frictional contacts. In *The 2003 ASME Design Engineering Technical Conferences* (Chicago, Illinois, Sept. 2003). 3

[SP95] SINGH K., PARENT R.: Implicit function based deformations of polyhedral objects. In *Implicit Surfaces'95* (Grenoble, France, Apr. 1995), pp. 113–128. Proceedings of the first international workshop on Implicit Surfaces. 3

[ST96] STEWART D. E., TRINKLE J. C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Internat. J. Numer. Methods Engineering* (1996). 3, 8

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Proceedings of SIGGRAPH 1987* (1987), 205–214. 3

[TW88] TERZOPOULOS D., WITKIN A.: Physically based models with rigid and deformable components. *IEEE Computer Graphics & Applications 8*, 6 (Nov. 1988), 41–51. 3

[WK94] WANG Y.-T., KUMAR V.: imulation of mechanical systems with unilateral constraints. *ASME Journal of Mechanical Design 116*, 2 (1994), 571–580. 3

[ZPvG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. In *Proceedings of SIGGRAPH 2001* (2001), pp. 371–378. 5

[ZSS97] ZORIN D., SCHRÖDER P., SWELDENS W.: Interactive multiresolution mesh editing. In *Proceedings of SIGGRAPH 1997* (1997), pp. 259–268. 7