3rd Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2006)
C. Mendoza, I. Navazo (Editors)

Efficient Updates of Bounding Sphere Hierarchies for Geometrically Deformable Models

J. Spillmann M. Becker M. Teschner

Computer Graphics, University of Freiburg, Germany

Abstract

We present a new approach for efficient collision handling of meshless objects undergoing geometric deformation. The presented technique is based on bounding sphere hierarchies. We show that information of the geometric deformation model can be used to significantly accelerate the hierarchy update. The cost of the presented hierarchy update depends on the number of primitives in close proximity, but not on the total number of primitives.
Further, the hierarchical collision detection is combined with a level-of-detail response scheme. Since the collision response can be performed on any level of the hierarchy, it allows for balancing accuracy and efficiency. Thus, the collision handling scheme is particularly useful for time-critical applications.

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

Keywords: physically-based modeling, collision detection, point-based models, shape matching, deformable objects

1. Introduction

Recent advances in interactive deformable modeling have renewed the interest in efficient deformable collision handling approaches. While the deformation of geometrically complex objects can be simulated efficiently, it is still a challenging problem to simulate interacting objects at interactive rates including deformation, collision detection and response.

Bounding volume hierarchies have shown to be very efficient for the detection of collisions among rigid objects. The hierarchy is pre-computed and can be updated efficiently if necessary. Unfortunately, in the context of deforming objects, the hierarchy has to be refitted in each simulation step. Traditional approaches require $\mathcal{O}(n)$ operations for this update, where n is the number of object primitives.

Various techniques have been proposed to accelerate the hierarchy update. In [LAM01], a combination of top-down and bottom-up updates has been presented. This approach accelerates the hierarchy update by a constant factor, but does not lower the complexity of $\mathcal{O}(n)$ operations.

In [OD99, RKC02], it has been shown that the hierarchy update for rigid bodies can be performed in an output-sensitive way, i. e. the complexity of the computation depends on the number of colliding primitives c which is commonly significantly smaller than the overall number of primitives.

In the context of deformable objects, first output-sensitive schemes have been proposed for specific deformation approaches. In [JP04], a very promising approach has been presented for reduced deformable models [JP02]. Further output-sensitive schemes for morphing and skeleton-based deformation have been presented in [LAM03] and [KZ05], respectively.

Our contribution

We present a new output-sensitive collision detection scheme for meshless object representations undergoing geometric deformation. Using the deformation model presented in [MHTG05], we show that the employed bounding sphere hierarchy can be efficiently updated. Its cost depends linearly on the number of pairs in close proximity. In typical scenarios, this number is limited due to spatial coherence. Thus, the expected cost to update the hierarchy is sublinear in the number of primitives. Each bounding sphere can be updated in constant time. In contrast to existing approaches

such as [JP04], our method is independent of the number of degrees-of-freedom of the underlying deformation model.

The collision detection algorithm can be combined with a standard collision response scheme such as [BFA02]. Further, the bounding sphere hierarchy can be employed as a multi-resolution data structure, allowing a level-of-detail collision response [PPG04].

The presented approach allows for time-critical collision handling of point-based and triangulated objects. In combination with the unconditionally stable deformation approach presented in [MHTG05], the dynamic interaction of geometrically complex deformable objects can be simulated at interactive rates.

2. Related Work

Collision detection with bounding volume hierarchies. Bounding volume hierarchies are efficient data structures for collision detection and a variety of bounding volumes have been investigated, e. g. spheres [Qui94, Hub95, BO02, KZ04], axis-aligned bounding boxes AABB [Ber97], oriented bounding boxes OBB [GLM96], k-DOPs [KHM*96], or convex hulls [EL01]. Collision detection based on these data structures is well-investigated and recent surveys can be found in [LG98, JTT01, Eri04, Ber04, TKH*05].

In the context of deforming geometries, bounding volume hierarchies have to be updated frequently and efficient update strategies are essential. In [LAM01], the traditional $\mathcal{O}(n)$ approach for the update of an AABB hierarchy is accelerated by a constant factor. In [GNRZ02], the update of wrapped sphere hierarchies is described. However, the approach is restricted to necklace-shaped geometries. Further approaches for polynomial deformable surfaces and cloth models have been presented in [HDLM96] and [MKE03], respectively. Other approaches exploit temporal coherence or lazy evaluation techniques. In [LAM05], an update strategy of an AABB hierarchy for deformable and breakable models has been presented that is linear in the number of primitives in most cases. In [BSB*01], a scheme is proposed that updates a sphere tree with a lazy evaluation scheme. However, this scheme is only fast if the deformations are kept small and local.

In order to further reduce the computational complexity, recent approaches to deformable collision detection exploit information provided by the deformation model. Output-sensitive hierarchy updates for models deformed by morphing and for skeletally deformable models are presented in [LAM03] and [KZ05], respectively. In [JP04], an output-sensitive hierarchy update and collision detection test for reduced deformable models [JP02] is discussed. However, in this approach, the update of each bounding volume is linear in the number of degrees of freedom of the deformation model.

In contrast to existing approaches, our output-sensitive collision detection scheme works for meshless object representations undergoing geometric deformation. In contrast to [JP04], the bounding volumes are updated in constant time.

3. Algorithm Overview

Our approach employs bounding sphere hierarchies to accelerate the collision detection among deformable objects. Therefore, a bounding sphere hierarchy is pre-computed for each object in its initial undeformed state. If the object deforms, the hierarchy is updated accordingly. Information on the geometric deformation model is used to efficiently refit the bounding spheres of the hierarchy (see Fig. 1).

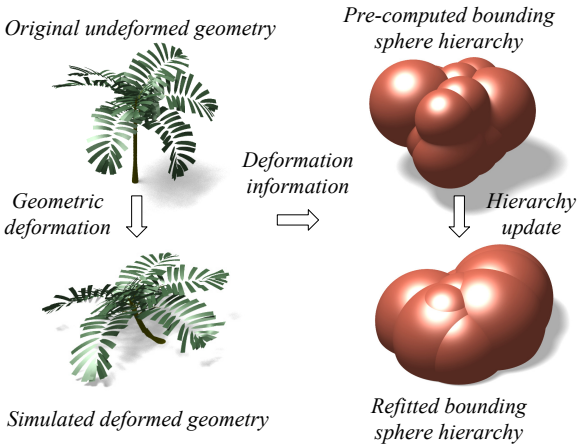

Figure 1: *Algorithm overview. A bounding sphere hierarchy is pre-computed for an undeformed object. If the object deforms, the hierarchy is updated employing information on the geometric deformation model.*

The level-of-detail collision response scheme computes contact forces at arbitrary levels of the bounding volume hierarchy and propagates the result to the enclosed object primitives. Alternatively, accurate contact forces are computed for object primitives if connectivity information is available.

The remainder of the paper is organized as follows. Section 4 describes the geometric deformation approach in order to illustrate the information that is provided to our collision detection approach. Section 5 discusses all aspects regarding the bounding volume hierarchy, i. e. generation, update, and query. In particular, it is shown that the expected update cost is sublinear in n and the cost to update a sphere is constant. Section 6 describes how the collision detection scheme can be combined with a level-of-detail response method. Section 7 analyzes the performance of the presented collision handling scheme and illustrates its application to interactive simulations.

4. Geometric Deformation

The geometric deformation approach [MHTG05] projects points representing a deformable object towards goal positions. To compute these goal positions, the deformation of the point cloud is estimated by a transformation matrix that minimizes the error between the deformed and the transformed initial positions. This registration process is called shape matching and represents the geometric deformation (see Fig. 2). The resulting transformation matrix is employed to accelerate the hierarchy update in our approach.

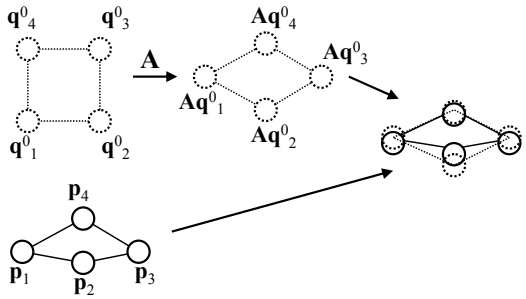

Figure 2: *The undeformed geometry* $\{\mathbf{q}^0_i\}$ *is registered with the deformed geometry* $\{\mathbf{p}_i\}$. *A transformation matrix* $\mathbf{A}$ *is computed that minimizes* $\sum_i \|\mathbf{Aq}^0_i - \mathbf{p}_i\|_2$.

The geometric deformation approach handles three transformation types. It considers points $\mathbf{x}^0_i \in \mathbb{R}^3$ of the undeformed object and points $\mathbf{x}_i \in \mathbb{R}^3$ of the deformed state. The centers of mass of the undeformed and the deformed geometry are $\mathbf{x}^0_{cm} \in \mathbb{R}^3$ and $\mathbf{x}_{cm} \in \mathbb{R}^3$, respectively. The vectors $\mathbf{q}^0_i = \mathbf{x}^0_i - \mathbf{x}^0_{cm}$ and $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}$ represent point positions relative to the center of mass.

Linear deformations. In this case, the deformation of a point cloud is estimated with a linear transformation matrix $\mathbf{A}$ that minimizes $\sum_i \|\mathbf{Aq}^0_i - \mathbf{p}_i\|^2_2$. Linear transformations are able to represent shear and stretch.

Quadratic deformations. To extend the range of deformations by twist and bending modes, quadratic transformations can be used. Having $\tilde{\mathbf{q}} = (q_x, q_y, q_z, q_x^2, q_y^2, q_z^2, q_xq_y, q_yq_z, q_zq_x)^T \in \mathbb{R}^9$, matrices $\mathbf{A}, \mathbf{Q}, \mathbf{M} \in \mathbb{R}^{3\times 3}$ are computed that minimize $\sum_i \|\tilde{\mathbf{A}}\tilde{\mathbf{q}}^0_i - \mathbf{p}_i\|^2_2$ with $\tilde{\mathbf{A}} = [\mathbf{AQM}] \in \mathbb{R}^{3\times 9}$. In this case $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{M}$ represent linear, quadratic and mixed terms, respectively.

Rigid objects. Shape matching for rigid objects is similar to the linear case. However, only the rotational part $\mathbf{R} \in \mathbb{R}^{3\times 3}$ of the matrix $\mathbf{A}$ is considered. $\mathbf{R}$ is computed from $\mathbf{A}$ using polar decomposition.

In order to further extend the range of deformations, the geometry can be decomposed into **clusters**. In that case, shape matching is performed for each cluster. Typical objects consist of one up to ten clusters. Refer to [MHTG05] for details.

5. Bounding sphere hierarchy

This section discusses the construction, update and query of the bounding sphere hierarchy. In particular, it is shown how information on the geometric deformation can be employed to accelerate the hierarchy update.

5.1. Construction of the hierarchy

The bounding sphere hierarchy is pre-computed for an object in its undeformed state. Since the construction of bounding volume hierarchies has already been investigated [OD99, BO02], we employ a technique similar to [OD99]. This guarantees that all successors of a bounding sphere contain the same number of points. Thus, the resulting tree is well-balanced. Following [GNRZ02], we use a wrapped hierarchy, i. e. a parent sphere does not enclose all child spheres, but only the associated geometry. This yields smaller spheres and improves the performance of the collision query. The minimum enclosing sphere of a point set is computed using [FG03].

5.2. Update of the hierarchy

If an object deforms, its bounding sphere hierarchy has to be updated. Instead of using information on all points enclosed by a sphere, we use information on the geometric deformation to refit spheres of the hierarchy.

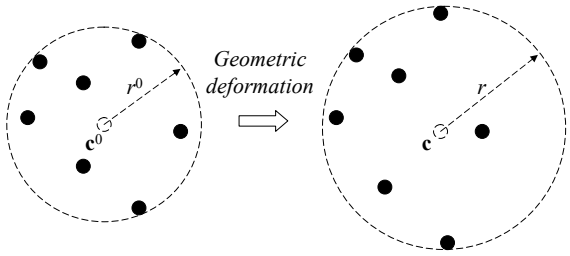

Figure 3: *Update of a bounding sphere hierarchy upon deformation. Left: In the undeformed state, a sphere given by* $\mathbf{c}^0$ *and* r^0 *encloses a point set* $\mathfrak{P}^0$. *Right: After a geometric deformation of the point set, a new center* $\mathbf{c}$ *and a new radius r have to be computed such that all deformed points are enclosed by the sphere.*

Let $\mathbf{x}^0_{cm}$ be the center of mass of the undeformed object and $\mathfrak{P}^0 = \{\mathbf{x}^0_i | i = 1, \ldots, n\}$ a subset of the undeformed object. Further, $\mathbf{p}^0_i = \mathbf{q}^0_i = \mathbf{x}^0_i - \mathbf{x}^0_{cm}$ are the relative positions of the subset with respect to the center to mass. We consider a sphere S^0 of the bounding volume hierarchy with center $\mathbf{c}^0$ and radius r^0 which encloses all points of $\mathfrak{P}^0$, i. e.

$$\|\mathbf{x}^0_i - \mathbf{c}^0\| \leq r^0 \;\; \forall \mathbf{x}^0_i \in \mathfrak{P}^0. \tag{1}$$

Fig. 3 illustrates the setting.

5.2.1. Update rule for linear deformations

Now we assume our object to be deformed. We call the new center of mass $\mathbf{x}_{cm}$, the new point positions $\mathbf{x}_i$, and the new relative positions $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}$. Let $\mathbf{q}_i = \mathbf{A}\mathbf{q}_i^0 = \mathbf{A}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0)$ be the transformed initial positions that result from the shape matching procedure. The deviation between the transformed initial positions and the deformed positions is

$$\mathbf{d}_i := \mathbf{q}_i - \mathbf{p}_i = \mathbf{A}\,\mathbf{q}_i^0 - \mathbf{p}_i. \tag{2}$$

For a sphere update, we consider the 2-norm of a square matrix $\mathbf{A}$ that is defined as the square root of the absolute value of the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. In our implementation, a polar decomposition is performed on $\mathbf{A}^T\mathbf{A}$ in order to find the largest eigenvalue. Geometrically, the 2-norm is the maximum amount the length of a vector can change under the transformation $\mathbf{A}$.

Lemma 1 (Linear case) If the center of the bounding sphere S^0 is updated with

$$\mathbf{c} \leftarrow \mathbf{A}\,(\mathbf{c}^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm} \tag{3}$$

and the radius with

$$r \leftarrow \|\mathbf{A}\|_2\, r^0 + d \tag{4}$$

then the updated sphere encloses all transformed points, i. e.

$$\|\mathbf{x}_i - \mathbf{c}\| \le r \;\; \forall i = 1,\ldots,n \tag{5}$$

The value $d := \max_j \left\|\mathbf{d}_j\right\|_2$ is computed from all colliding points $\mathbf{p}_j$. For non-colliding points $\mathbf{p}_i$, $\mathbf{d}_i$ is negligible since the positions of these points coincide with the computed goal positions, and the bounding spheres are proven to enclose the latter.

Proof Applying the definitions of $\mathbf{q}_i^0$ and $\mathbf{p}_i$ to (2), we get

$$\begin{aligned} \mathbf{A}\,(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) &= \mathbf{x}_i - \mathbf{x}_{cm} + \mathbf{d}_i \\ \Longleftrightarrow \mathbf{x}_i &= \mathbf{A}\,\mathbf{x}_i^0 - \mathbf{A}\,\mathbf{x}_{cm}^0 + \mathbf{x}_{cm} - \mathbf{d}_i \end{aligned} \tag{6}$$

Using (3) and (6) to describe the distance between a new position and the new center, we get

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{c}\|_2 &= \|\mathbf{A}\,\mathbf{x}_i^0 - \mathbf{A}\,\mathbf{x}_{cm}^0 + \mathbf{x}_{cm} - \mathbf{d}_i \\ &\quad -(\mathbf{A}\,(\mathbf{c}^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm})\|_2 \\ &= \|\mathbf{A}\,(\mathbf{x}_i^0 - \mathbf{c}^0) - \mathbf{d}_i\|_2 \end{aligned} \tag{7}$$

Applying the triangle inequality and the submultiplicative property of the matrix norm $\|\cdot\|_2$, we get

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \;\le\; \|\mathbf{A}\|_2\, \|\mathbf{x}_i^0 - \mathbf{c}^0\|_2 + \|\mathbf{d}_i\|_2 \tag{8}$$

Using (1),

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \;\le\; \|\mathbf{A}\|_2\, r^0 + \|\mathbf{d}_i\|_2 \tag{9}$$

Using (4) and the definition of d, we end up with

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \le r \tag{10}$$

□

5.2.2. Update rule for quadratic deformations

As in the linear case, $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}$ are the relative positions of the deformed points of $\mathfrak{P}^0$. In contrast to the linear case, the transformation of the relative point positions $\mathbf{q}_i^0$ is now defined as:

$$\mathbf{q}_i \;:=\; [\mathbf{A}\,\mathbf{Q}\,\mathbf{M}]\tilde{\mathbf{q}}_i^{\,0} \tag{11}$$

with $\tilde{\mathbf{q}}_i^{\,0} := (q_{i,x}^0, q_{i,y}^0, q_{i,z}^0, (q_{i,x}^0)^2, (q_{i,y}^0)^2, (q_{i,z}^0)^2, q_{i,x}^0 q_{i,y}^0, q_{i,y}^0 q_{i,z}^0, q_{i,z}^0 q_{i,x}^0)^T$. Matrices $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{M} \in \mathbb{R}^{3\times 3}$ are provided by the shape matching procedure. Now, the difference between a deformed point and its transformed initial position is

$$\mathbf{d}_i := \mathbf{q}_i - \mathbf{p}_i = [\mathbf{A}\,\mathbf{Q}\,\mathbf{M}]\tilde{\mathbf{q}}_i^0 - \mathbf{p}_i \tag{12}$$

As in the linear case, the center and radius of the sphere S^0 are transformed such that all deformed points of the set $\mathfrak{P}^0$ are within the new sphere. Therefore, we define

$$q_{max} \;:=\; \max_{i=1,\ldots,n} \left\| \begin{pmatrix} (x_{i,x}^0)^2 - (c_x^0)^2 \\ (x_{i,y}^0)^2 - (c_y^0)^2 \\ (x_{i,z}^0)^2 - (c_z^0)^2 \end{pmatrix} \right\|_2 \tag{13}$$

$$m_{max} \;:=\; \max_{i=1,\ldots,n} \left\| \begin{pmatrix} x_{i,x}^0 x_{i,y}^0 - c_x^0 c_y^0 \\ x_{i,y}^0 x_{i,z}^0 - c_y^0 c_z^0 \\ x_{i,z}^0 x_{i,x}^0 - c_z^0 c_x^0 \end{pmatrix} \right\|_2 \tag{14}$$

Lemma 2 (Quadratic case) If the center of the bounding sphere S^0 is updated with

$$\mathbf{c} \leftarrow [\mathbf{A}\,\mathbf{Q}\,\mathbf{M}]\tilde{\mathbf{c}}_{rel}^0 + \mathbf{x}_{cm} \tag{15}$$

and the radius with

$$r \leftarrow \|\mathbf{A}\|_2\, r^0 + \|\mathbf{Q}\|_2\, q_{max} + \|\mathbf{M}\|_2\, m_{max} + d \tag{16}$$

then all deformed points are in the new sphere, i. e.

$$\|\mathbf{x}_i - \mathbf{c}\| \le r \;\; \forall i = 1,\ldots,n \tag{17}$$

($d := \max_j \|\mathbf{d}_j\|_2$ for all colliding points $\mathbf{p}_j$ and $\mathbf{c}_{rel}^0 := \mathbf{c}^0 - \mathbf{x}_{cm}^0$.)

Proof Using (15) and the definition of $\mathbf{p}_i$, the distance between a new point position and the new sphere center can be written as

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \;=\; \|\mathbf{p}_i + \mathbf{x}_{cm} - ([\mathbf{A}\,\mathbf{Q}\,\mathbf{M}]\tilde{\mathbf{c}}_{rel}^0 + \mathbf{x}_{cm})\|_2 \tag{18}$$

Assume without loss of generality $\mathbf{x}_{cm}^0 = 0$. Using (12) we get

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \;=\; \|[\mathbf{A}\,\mathbf{Q}\,\mathbf{M}](\tilde{\mathbf{q}}_i^0 - \tilde{\mathbf{c}}_{rel}^0) - \mathbf{d}_i\|_2 \tag{19}$$

Since $\mathbf{x}_{cm}^0 = 0$, we have $\mathbf{q}_i^0 = \mathbf{x}_i^0$, $\mathbf{c}_{rel}^0 = \mathbf{c}^0$. Therefore, the distance is

$$\|\mathbf{x}_i - \mathbf{c}\|_2 = \|[\mathbf{A}\,\mathbf{Q}\,\mathbf{M}](\tilde{\mathbf{x}}_i^0 - \tilde{\mathbf{c}}^0) - \mathbf{d}_i\|_2 \tag{20}$$

Applying the triangle inequality we get

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \le \|[\mathbf{A}\,\mathbf{Q}\,\mathbf{M}](\tilde{\mathbf{x}}_i^0 - \tilde{\mathbf{c}}^0)\|_2 + \|\mathbf{d}_i\|_2 \tag{21}$$

Decomposing the right hand side into linear, quadratic and

mixed terms, using the triangle inequality and the submultiplicative property of the matrix norm $\|\cdot\|_2$ on the right hand side, we get

$$\|\mathbf{x}_i-\mathbf{c}\|_2 \leq \|\mathbf{A}\|_2\|(\mathbf{x}_i^0-\mathbf{c}^0)\|_2+\|\mathbf{Q}\|_2\left\|\begin{pmatrix}(x_{i,x}^0)^2-(c_x^0)^2\\(x_{i,y}^0)^2-(c_y^0)^2\\(x_{i,z}^0)^2-(c_z^0)^2\end{pmatrix}\right\|_2 + \|\mathbf{M}\|_2\left\|\begin{pmatrix}x_{i,x}^0x_{i,y}^0-c_x^0c_y^0\\x_{i,y}^0x_{i,z}^0-c_y^0c_z^0\\x_{i,z}^0x_{i,x}^0-c_z^0c_x^0\end{pmatrix}\right\|_2+\|\mathbf{d}_i\|_2 \quad (22)$$

Using (1), (13) and (14), we get

$$\|\mathbf{x}_i-\mathbf{c}\|_2 \leq \|\mathbf{A}\|_2 r^0+\|\mathbf{Q}\|_2 q_{max}+ +\|\mathbf{M}\|_2 m_{max}+\|\mathbf{d}_i\|_2 \quad (23)$$

Using the definition of d and the updated radius r, we end up with

$$\|\mathbf{x}_i-\mathbf{c}\|_2 \leq r \quad (24)$$

□

Fig. 4 illustrates the update of the sphere hierarchy for a deforming object.

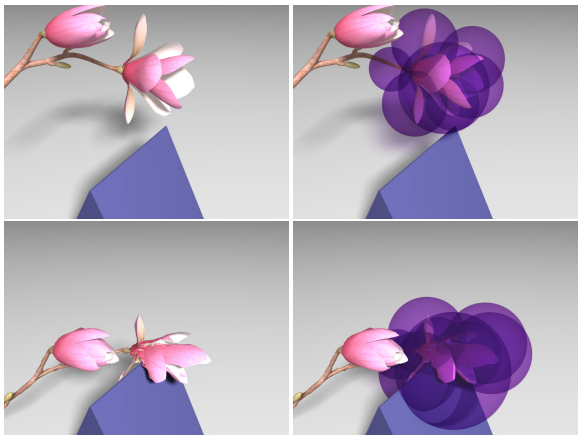

Figure 4: *A quadratically deformable flower collides with a rigid object. The bounding sphere hierarchy is updated according to the description in Section 5.2.2. For clarity, only a subset of the bounding spheres is shown.*

5.2.3. Update rule for rigid objects

For rigid objects, the transformation matrix $\mathbf{R}$ is orthogonal. Thus, the eigenvalues of $\mathbf{R}^T\mathbf{R}=\mathbf{R}^{-1}\mathbf{R}=\mathbf{I}$ are one and $\|\mathbf{R}\|_2=1$. The sphere centers and radii are updated as in the linear case.

5.2.4. Clustered objects

According to the deformation model, all clusters of an object are of the same type, i. e. rigid, linear or quadratic. If an object consists of more than one cluster, the shape matching process computes a transformation matrix for each cluster. In the linear and the quadratic case, the norm of the transformation matrix has to be computed for each cluster.

5.3. Collision query

In order to test two objects for collision, the root nodes of the hierarchy are tested for interference. If they interfere, both hierarchies are traversed recursively. Therefore, only children of colliding spheres need to be updated. At leaf nodes, primitives of one object are tested against enclosing spheres of the other hierarchy. Hence, point-based objects can be handled. If connectivity information is available, traditional interference tests among primitives can be performed. For experiments, we have implemented collision queries for point-based objects and for objects with triangulated surfaces.

5.4. Performance

In most cases, the cost for updating the hierarchy is linear in c where c is the number of pairs of primitives in close proximity. Assume that c leaf nodes in the hierarchy have to be tested. Due to spatial coherence, they will share about $\frac{c}{2}$ parent nodes which themselves have $\frac{c}{4}$ parent nodes. Thus, in most cases $2c$ nodes have to be updated. However, an exact analysis is difficult since the cost function depends on the distribution of the c leaf nodes.

If the object is clustered, a separate branch in the hierarchy is computed for each cluster. Thus, each bounding sphere belongs to only one cluster and can be updated in constant time. The cost to update the root and those leaves that are within the cluster overlap regions is linear in the number of clusters they belong to. However, neither the number of clusters nor the percentage of cluster overlap depend on the number of primitives. Further, the number of clusters is significantly smaller than the number of primitives.

In the linear and rigid cases, a sphere center is updated with two vector additions and one matrix-vector multiplication. In the quadratic case, two vector additions and three matrix-vector multiplications are required. The radius is updated with one multiplication and one addition in the linear case, with one addition in the rigid case, and with three multiplications and four additions in the quadratic case.

Prior to a hierarchy traversal, the following values have to be computed once per object cluster. In the linear case, the matrix norm of $\mathbf{A}$ and the value d are computed. The norm of $\mathbf{A}$ can be computed in constant time and the computation of d is linear in the number of colliding points c. In the quadratic case, the matrix norms $\mathbf{A}$, $\mathbf{Q}$ and $\mathbf{M}$ can be computed in constant time and d is computed as described for the linear case.

The center of mass of the undeformed model $\mathbf{x}_{cm}^0$, the center of mass of the deformed model $\mathbf{x}_{cm}$, and the deformation

matrices $\mathbf{A}$ and $[\mathbf{A}\,\mathbf{Q}\,\mathbf{M}]$ are provided by the geometric deformation approach. Further, the radii r^0 and the values q_{max}, m_{max} in the quadratic case can be pre-computed. They only depend on the initial undeformed state.

6. Level-of-detail collision response

Bounding volume hierarchies are dedicated to level-of-detail collision response. The collision query can be aborted at an arbitrary level of the hierarchy. The collided bounding volumes at this layer are an approximation of the collided object parts. Collision response is then performed on those collided volumes rather than on the original point cloud.

Level-of-detail collision handling can be used to implement time-critical simulations. Depending on the time available, the hierarchies are tested for collisions down to a certain layer and the collision response is performed on this layer.

6.1. Generic point cloud approximation

We propose a generic level-of-detail collision response scheme similar to [PPG04]. As preprocessing step, approximations of the point cloud are computed. The approximation of a point cloud at a given layer coincides with the centers of the bounding spheres at this layer in the hierarchy. If topology information is available, a surface normal is computed that is updated upon deformation.

6.2. Collision Response

The collision query returns a set of pairs of colliding spheres or a set of sphere / point pairs. Since the sphere centers are considered approximations of the enclosed point set, collision response can be consistently computed on pairs of points. A contact normal is computed for each contact pair, where the plausibility of the collision response depends on the accuracy of the contact normal.

To compute the contact force, a standard response scheme such as [OD99, GO05] is employed. If topology information is available, e. g. for triangulated surface, a more accurate scheme [BFA02] is applied. Thus, even stacking problems can be handled as illustrated in Fig. 5.

7. Results

In this section, three experiments are discussed to illustrate the efficiency of the proposed approach. All measurements were taken on an Intel Xeon, 3.8 GHz CPU.

Collision query. We propose to update the hierarchy of a deforming object in an output-sensitive manner, i. e. only potentially colliding bounding spheres are updated. This is achieved by considering the deformation matrices obtained from the underlying deformable model. Since (16) is a conservative estimation, the updated spheres have larger

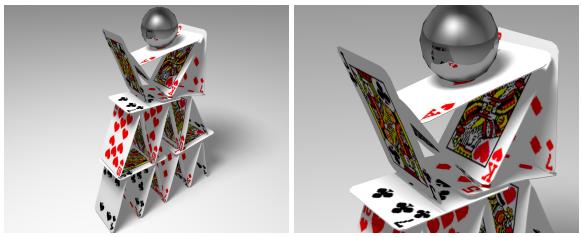

Figure 5: *By employing the response scheme presented in [BFA02] we can handle stacking problems with deformable objects.*

radii compared to minimally enclosing spheres. As a consequence, the bounding spheres are not as tight-fitting as possible and a collision query generally traverses more nodes as would be required in case of minimally enclosing spheres. However, the following experiment shows that this fact has only a minor influence on the collision query performance. On the other hand, computing the minimum enclosing sphere of a set of points is an expensive operation that is hardly feasible in interactive simulations.

For the measurements we have taken two objects with varying resolutions ranging from 8k to 160k points for both objects. The two objects move towards each other, collide, and deform under the influence of the contact forces. Thus, the sphere hierarchies have to be updated in each time step. Fig. 6 shows the test scenario and Fig. 7 provides the measurements. Since the same dynamic simulation is computed for all measurements, the number of updated spheres is varying according to the resolution of the objects. For optimally refitted hierarchies, the maximum number of sphere updates per simulation step ranges from 5.2k to 93k, depending on the object resolution. For hierarchies that are updated with our approach, the maximum number of sphere updates ranges from 5.4k to 96k.

Time-critical collision handing. In order to show the benefit of the presented approach for time-critical applications, we have simulated 300 deformable objects with an overall number of 131k points as depicted in Fig. 8. Table 1 shows the performance of the collision handling scheme computed at various levels of the hierarchy ranging from 2 to 6. As described in Sec. 6.2, the number of colliding spheres varies according to the hierarchy level. Further, the computing time for the collision query and the collision response depends on the hierarchy level. If the collision response is computed at a lower level, it is less accurate compared to the response at a higher level. Thus, accuracy and efficiency can be balanced. Table 1 shows that the presented collision handling scheme can handle 131k points at interactive rates.

Interactive simulation of deformable objects. The third experiment depicted in Fig. 9 shows that the presented approach can be integrated into interactive simulations. In the experiment, three objects with an overall number of 19k

Figure 6: *Simulation of two geometrically complex models. For the performance measurements illustrated in Fig. 7, the overall number of points in the scene varies from 8k to 160k. The number of sphere updates ranges from 5.2k to 93k for optimally refitted hierarchies, and from 5.4k to 96k for hierarchies that are updated with our approach.*

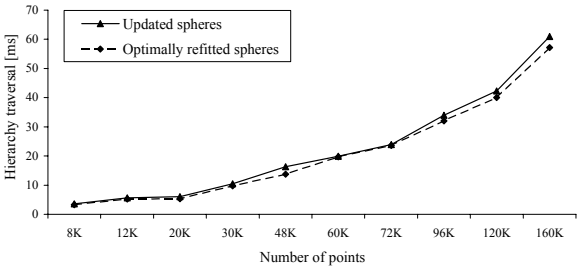

Figure 7: *Comparison of the collision query times of an optimally refitted bounding sphere hierarchy versus the updated sphere hierarchy using our approach. The optimally refitted spheres have smaller radii than the conservatively updated spheres. Since in the latter case more spheres have to be visited and updated, the collision query is less efficient. However, the diagram shows that the difference is almost negligible.*

points are simulated. The average collision detection time is 4.39 ms and collision response takes 10.66 ms. The overall time of the geometric deformation is 24.33 ms, and the visualization takes 7.8 ms. Thus, the total computing time per frame is 47.19 ms and the animation runs at 21 frames per second. The time for detecting and resolving the collisions is less than 15 ms. It is no longer the bottleneck of the simulation.

8. Conclusion and Future Work

We have presented an output-sensitive collision detection scheme for meshless and mesh-based objects undergoing

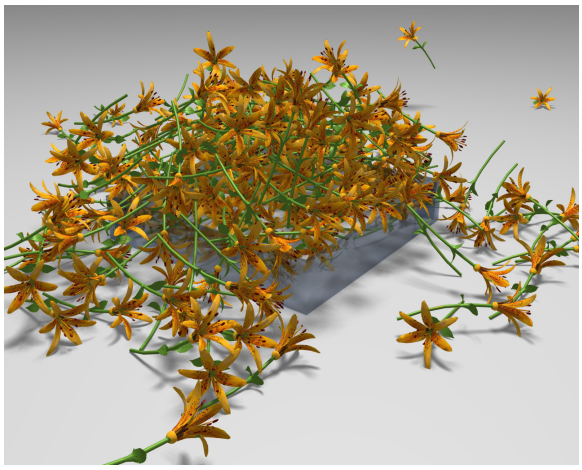

Figure 8: *Simulation of 300 deformable objects with an overall number of 131k points. Level-of-detail collision handling can be computed at interactive rates as shown in Tab. 1.*

Level	# Contacts	Det. [ms]	Resp. [ms]	Total [ms]
2	1031	18.41	24.11	42.52
3	1650	22.19	27.01	49.20
4	7511	30.24	32.03	62.27
5	27195	58.61	45.35	103.96
6	42635	93.97	52.88	146.85

Table 1: *The collision response can be computed efficiently at a lower level. At higher levels, the response is more accurate. Fig. 8 illustrates the test scenario.*

geometric deformation. The cost to update the hierarchy under deformation depends on the number of primitives in close proximity, and not on the total number of primitives. It takes constant time to update a bounding sphere. In particular, the update does not depend on the number of enclosed particles or on the number of degrees of freedom of the underlying deformation model. The collision detection algorithm is combined with a level-of-detail collision response. The resulting collision handling scheme is fast and provides

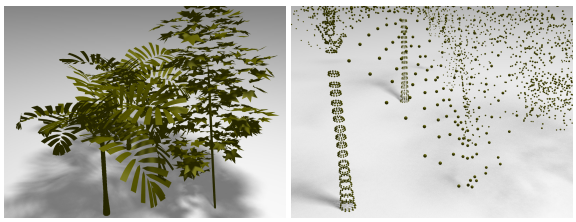

Figure 9: *Interactive animation of three deformable objects. The scene consists of 19k points and can be computed at 21 frames per second. The right-hand side image illustrates the simulated point cloud.*

a physically plausible behavior. It can be used in interactive simulations that require time-critical collision handling. In combination with the unconditionally stable deformation model presented in [MHTG05], the approach is particularly interesting for interactive applications such as games.

Limitations. In its present form, the scheme applies only to objects that are deformed using the approach in [MHTG05]. However, based on the presented approach, similar schemes can be developed for other deformable models whose deformations are represented by matrices. In its present form, the method does not detect self intersections which will be addressed by future research.

9. Acknowledgements

The flower models have been taken from the Toucan Virtual Museum, Toucan Corporation, Japan.

References

[Ber97] BERGEN G.: Efficient Collision Detection of Complex Deformable Models Using AABB Trees. *Journal of Graphics Tools 2*, 4 (1997), 1–13.

[Ber04] BERGEN G.: *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann Publishers, ISBN 1-55860-801-X, 2004.

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Transactions on Graphics 21*, 3 (2002), 594–603.

[BO02] BRADSHAW G., O'SULLIVAN C.: Sphere-Tree Construction Using Dynamic Medial Axis Approximation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 33–40.

[BSB*01] BROWN J., SORKIN S., BRUYNS C., LATOMBE J.-C., MONTGOMERY K., STEPHANIDES M.: Real-time simulation of deformable objects: Tools and application. In *Computer Animation 2001* (2001), pp. 228–236.

[EL01] EHMANN S. A., LIN M. C.: Accurate and Fast Proximity Queries Between Polyhedra Using Convex Surface Decomposition. In *Eurographics* (2001), pp. 500–510.

[Eri04] ERICSON C.: *Real-Time Collision Detection*. Morgan Kaufmann Publishers, ISBN 1-55860-732-3, 2004.

[FG03] FISCHER K., GARTNER B.: The Smallest Enclosing Ball of Balls: Combinatorial Structure and Algorithms. In *Symposium on Computational Geometry* (2003), pp. 292–301.

[GLM96] GOTTSCHALK S., LIN M., MANOCHA D.: OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *ACM SIGGRAPH* (1996), pp. 171–180.

[GNRZ02] GUIBAS L., NGUYEN A., RUSSEL D., ZHANG L.: Collision Detection for Deforming Necklaces. In *Symposium on Computational geometry* (2002), pp. 33–42.

[GO05] GIANG T., O'SULLIVAN C.: Closest Feature Maps for Time-Critical Collision Handling. In *Workshop on Virtual Reality Interaction and Physical Simulation* (2005).

[HDLM96] HUGHES M., DIMATTIA C., LIN M. C., MANOCHA D.: Efficient and Accurate Interference Detection for Polynomial Deformation. In *Computer Animation* (1996), p. 155.

[Hub95] HUBBARD P. M.: Collision Detection for Interactive Graphics Applications. *IEEE Transactions on Visualization and Computer Graphics 1*, 3 (1995), 218–230.

[JP02] JAMES D. L., PAI D. K.: DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware. In *ACM Siggraph* (2002), pp. 582–585.

[JP04] JAMES D. L., PAI D. K.: BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models. *ACM Transactions on Graphics 23*, 3 (2004), 393–398.

[JTT01] JIMÉNEZ P., THOMAS F., TORRAS C.: 3D Collision Detection: A Survey. *Computers and Graphics 25*, 2 (2001), 269–285.

[KHM*96] KLOSOWSKI J., HELD M., MITCHELL J., SOWIZRAL H., ZIKAN K.: Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics 4*, 1 (1996), 21–36.

[KZ04] KLEIN J., ZACHMANN G.: Point Cloud Collision Detection. *Computer Graphics Forum (Proceedings of Eurographics 2004) 23*, 3 (2004), 567–576.

[KZ05] KAVAN L., ZARA J.: Fast Collision Detection for Skeletally Deformable Models. *Computer Graphics Forum (Proceedings of Eurographics 2005) 24*, 3 (2005), 363–372.

[LAM01] LARSSON T., AKENINE-MÖLLER T.: Collision Detection for Continuously Deforming Bodies. In *Eurographics* (2001), pp. 325–333.

[LAM03] LARSSON T., AKENINE-MÖLLER T.: Efficient Collision Detection for Models Deformed by Morphing. *The Visual Computer 19*, 2-3 (2003), 164–174.

[LAM05] LARSSON T., AKENINE-MÖLLER T.: A Dynamic Bounding Volume Hierarchy for Generalized Collision Detection. In *Workshop on Virtual Reality Interactions and Physical Simulations* (2005), pp. 91–100.

[LG98] LIN M., GOTTSCHALK S.: Collision Detection Between Geometric Models: A Survey. In *IMA Conference on Mathematics of Surfaces* (1998), pp. 37–56.

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless Deformations Based on Shape Matching. *ACM Transactions on Graphics 24*, 3 (2005), 471–478.

[MKE03] MEZGER J., KIMMERLE S., ETZMUSS O.: Hierarchical Techniques in Collision Detection for Cloth Animation. In *Journal of WSCG* (2003), vol. 11, pp. 322–329.

[OD99] O'SULLIVAN C., DINGLIANA J.: Real-time Collision Detection and Response Using Sphere-Trees. In *Spring Conference on Computer Graphics* (1999), pp. 83–92.

[PPG04] PAULY M., PAI D. K., GUIBAS L. J.: Quasi-Rigid Objects in Contact. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 109–119.

[Qui94] QUINLAN S.: Efficient Distance Computation between Non-Convex Objects. In *IEEE International Conference on Robotics and Automation* (1994), pp. 3324–3329.

[RKC02] REDON S., KHEDDAR A., COQUILLART S.: Fast Continuous Collision Detection between Rigid Bodies. *Computer Graphics Forum (Proceedings of Eurographics 2002) 21*, 3 (2002), 279–279.

[TKH*05] TESCHNER M., KIMMERLE S., HEIDELBERGER B., ZACHMANN G., RAGHUPATHI L., FUHRMANN A., CANI M., FAURE F., MAGNENAT-THALMANN N., STRASSER W.: Collision Detection for Deformable Objects. *Computer Graphics Forum 24*, 1 (2005), 61–81.