

Subspace Self-Collision Culling

Jernej Barbič
University of Southern California

Doug L. James
Cornell University

Abstract

We show how to greatly accelerate self-collision detection (SCD) for reduced deformable models. Given a triangle mesh and a set of deformation modes, our method precomputes *Subspace Self-Collision Culling (SSCC) certificates* which, if satisfied, prove the absence of self-collisions for large parts of the model. At run-time, bounding volume hierarchies augmented with our certificates can aggressively cull overlap tests and reduce hierarchy updates. Our method supports both discrete and continuous SCD, can handle complex geometry, and makes no assumptions about geometric smoothness or normal bounds. It is particularly effective for simulations with modest subspace deformations, where it can often verify the absence of self-collisions in *constant time*. Our certificates enable low amortized costs, in time and across many objects in multi-body dynamics simulations. Finally, SSCC is effective enough to support self-collision tests at audio rates, which we demonstrate by producing the first sound simulations of clattering objects.

CR Categories: I.6.8 [Simulation and Modeling]: Types of Simulation—Animation, I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: self-collision detection, model reduction, kinetic data structures

1 Introduction

Reduced-order deformable models based on low-rank superpositions of displacement “modes” are widely used in computer graphics due to the significant accelerations possible for these simplified models. Such *subspace* models take the form $\mathbf{u} = \mathbf{U}\mathbf{q}$, where $\mathbf{u} \in \mathbb{R}^{3V}$ contains the $V = O(N)$ vertex displacements of an N triangle mesh, $\mathbf{U} \in \mathbb{R}^{3V \times r}$ is the basis matrix where each of the $r \ll V$ columns gives one deformation mode, and $\mathbf{q} \in \mathbb{R}^r$ are the reduced coordinates. Examples of subspace models include animation using linear modal analysis [Pentland and Williams 1989], nonlinear subspace deformations [Barbič and James 2005], freeform deformations [Faloutsos et al. 1997], multibody dynamics [Kaufman et al. 2008], data-driven character animation [Meyer and Anderson 2007], compressed animations [Alexa and Müller 2000], and nonlinear sound synthesis [Chadwick et al. 2009].

In this paper, we give an algorithm that accelerates self-collision detection (SCD) for subspace deformable models. We evaluate the algorithm using linear modal analysis, nonlinear subspace deformations, and modes that spatially vary as quadratic polynomials. Fast methods have been previously devised to reduce collision detection costs between separate reduced deformable objects [James and Pai 2004]. However, due to the unique challenges of SCD, these inter-object methods are not suited for efficient self-collision processing.

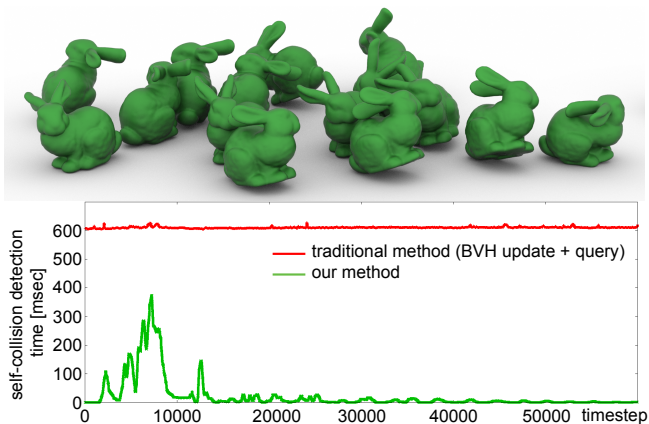


Figure 1: Fast self-collision detection using subspace culling resulted in a $29\times$ speedup in total self-collision costs for this 239,424-triangle, 16-bunny “ear drop” example (Top) with nonlinear many-body dynamics. (Bottom) Traditional self-collision costs (steady red curve) were roughly 600 msec/timestep, whereas our subspace culling approach (green curve) is non-negligible only during significant deformations, e.g., when bunnies fall on their ears.

Self-collisions occur with many deformable objects, and can be visually, dynamically and even acoustically important. Self-collision detection (SCD) often dominates inter-object collision costs, and is often cited as computationally expensive [Teschner et al. 2005], even when there are no self-collisions. The worst-case complexity of SCD is $O(N^2)$, encountered in degenerate cases where all triangles overlap. More importantly in practice, the lower-bound complexity of traditional bounding volume hierarchy SCD methods is $\Omega(N)$, because each triangle is in close proximity to neighboring triangles and must be checked for nearby collisions. SCD has therefore remained a significant complexity bottleneck for reduced deformable models, to the extent that many fast reduced-deformable simulations ignore SCD in practice, or restrict themselves to particular objects where self-collisions are unlikely, e.g., plastic chairs.

Our *Subspace Self-Collision Culling (SSCC)* method works by constructing simple scalar-valued inequalities (the *certificates*), which, if satisfied, prove that a subset of the model cannot be self-colliding in a given configuration \mathbf{q} . The certificates are precomputed by exploiting the known deformation structure (matrix \mathbf{U}), and can save a large number of hierarchy updates and close-proximity overlap tests. Our method is not intended for highly deformable objects such as cloth, and its worst-case complexity is still $O(N^2)$. However, by avoiding expensive tests in cases where self-collisions are limited or nonexistent (due to subspace deformation structure), we can dramatically reduce the lower bound from $\Omega(N)$ to $\Omega(r)$, for $r \ll N$ subspace dimensions. Our method therefore provides output-sensitive SCD at cost sublinear in the geometric complexity.

We designed our certificates so that they can work in tandem with traditional bounding volume hierarchies, which greatly accelerates SCD in multi-body dynamics (see Figure 1). Simulations typically exhibit periods of high and low SCD costs, e.g., due to large and small deformations and changing proximity. However, our time-averaged (amortized) cost is typically far lower than without using the certificates, even when including the certificate precomputation costs. The amortized costs of SSCC-based self-collision detection

are low enough for audio-rate SCD to enable sound simulations of self-colliding and thus clattering objects (see Figure 13). Finally, each certificate also provides a mechanism for culling *continuous* self-collision tests for entire parts of a triangle mesh.

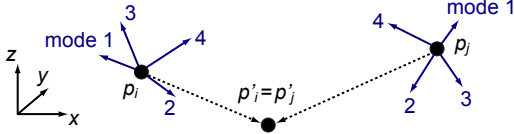


Figure 2: The min-norm collision for two points ($r = 4$).

The core technical insight of Subspace Self-Collision Culling is simple and easily illustrated by pairwise collisions between 3D vertices. Let vertex i 's undeformed position \mathbf{p}_i be displaced by a linear combination of r modal displacements (columns of $\mathbf{U}_i \in \mathbb{R}^{3 \times r}$):

$$\mathbf{p}'_i = \mathbf{p}_i + \mathbf{U}_i \mathbf{q} \quad \Leftrightarrow \quad \mathbf{p}'_i(t) = \mathbf{p}_i + \mathbf{U}_i \mathbf{q}(t). \quad (1)$$

Bringing two vertices, i and j , into contact ($\mathbf{p}'_i = \mathbf{p}'_j$) imposes 3 linear constraints, $(\mathbf{U}_i - \mathbf{U}_j)\mathbf{q} = (\mathbf{p}_j - \mathbf{p}_i)$, satisfied by infinitely many \mathbf{q} solutions for $r > 3$ (see Figure 2). There exists a solution \mathbf{q}^*_{ij} with the minimal $\|\mathbf{q}\|$ norm. Then, for a given deformation \mathbf{q} , vertices i and j cannot be colliding if $\|\mathbf{q}\| < \|\mathbf{q}^*_{ij}\| \equiv \mathcal{R}_{ij}$. Interestingly, \mathbf{q}^*_{ij} is the least-squares solution $\mathbf{q}^*_{ij} = (\mathbf{U}_i - \mathbf{U}_j)^\dagger (\mathbf{p}_j - \mathbf{p}_i)$, and can be computed easily using QR decomposition [Golub and Loan 1996]. Furthermore, by taking the minimum over all (i, j) pairs ($i \neq j$), we can compute a self-collision-free certificate valid whenever

$$\|\mathbf{q}\| < \mathcal{R} = \min_{i,j} \mathcal{R}_{ij}. \quad (2)$$

This same idea can be generalized from points to triangles, to arbitrary mesh subsets Ω and Ω' (see Figure 3), including self-collisions, $\Omega = \Omega'$, and to bounding volumes and hierarchies. Note that if a certificate is not satisfied, it is inconclusive, and SCD needs to recurse to finer geometric scales.

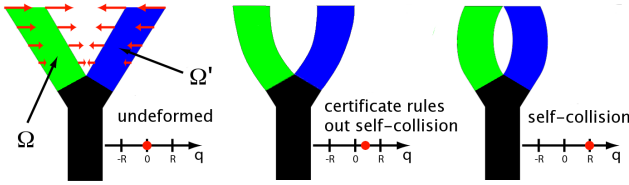


Figure 3: Self-collision detection with a certificate: (Left) undeformed configuration and the deformation mode ($r = 1$). (Middle) certificate guarantees the absence of collisions between Ω and Ω' in $O(r)$ time. (Right) certificate becomes violated at a self-collision.

In the rest of the paper we show how subsets Ω can be selected, processed in a hierarchical manner, and the computation organized so that certificate precomputation is both efficient and useful for subsequent SCD. We show how to easily improve culling in BVH-based SCD using collision-free subspace certificates (§4). We describe efficient methods for precomputing subspace bounds (§5) between triangles, between bounding volumes, and for hierarchical BVH construction. Mode scaling is described for improved culling (§6), and our experimental results are given in §7.

2 Related Work

Deformable collision detection is widely studied in computer graphics, and we refer the reader to a good recent survey [Teschner et al. 2005]. We focus here on self-collision detection (SCD), making distinctions between SCD methods for general deformations, and those that exploit deformation structure for speed.

Perhaps the most popular approach to SCD is to build a bounding volume hierarchy (BVH) on the object, and test the BVH against itself. Such approaches are widely used for cloth simulation [Volino and Magnenat-Thalmann 1994; Bridson et al. 2002]. Common bounding volume choices include spheres [Hubbard 1995], axis-aligned bounding boxes (AABBs) [van den Bergen 1997], oriented bounding boxes (OBBs) [Gottschalk et al. 1996], and discrete oriented polytopes (k -DOPs) [Klosowski et al. 1998]. Various inherent tradeoffs in bound tightness and hierarchical update speeds have been established [Gottschalk et al. 1996], with AABB/ k -DOPs performing well in practice [van den Bergen 1997; Teschner et al. 2005]. Continuous SCD methods are needed for processing thin objects such as cloth, shells, or rods [Bridson et al. 2002], and our certificates support continuous tests. Spatial partitioning techniques, such as uniform subdivision, support general SCD and have been optimized, e.g., for tetrahedral models [Teschner et al. 2003]. Since these methods must partition and process N triangles at each timestep, they have an inherent $\Omega(N)$ cost.

Such SCD approaches have two $\Omega(N)$ per-timestep costs that we aim to reduce: $[C_{\text{Update}}]$ the BVH must be updated to refit the N deformed triangles, and worse $[C_{\text{Query}}]$ the BVH is traversed to identify and perform triangle-level overlap tests, which typically involves testing each of the N triangles against non-adjacent neighbors. Various strategies have been proposed for reducing both SCD costs, C_{Update} and C_{Query} . To reduce C_{Update} , faster methods for updating BVHs have been proposed [Larsson and Akenine-Möller 2001]. Also, lazy BVH updates have been proposed for inter-object CD that exploit deformation structure [Larsson and Akenine-Möller 2003]. Bounded Deformation Trees (BD-Trees) [James and Pai 2004] exploit the same reduced-order deformation model (1) to update bounding volumes (BVs) in an output-sensitive manner, but can not alone accelerate SCD because SCD usually requires updating and testing all BVs. Also, looser BVs can yield poorer C_{Query} performance. In experiments, we observed roughly $1.5 \times$ slower SCD for AABB-based BD-Trees than traditional AABB-Trees. In contrast, we augment BVHs (including BD-Trees) with SSCC certificates to greatly reduce both C_{Update} and C_{Query} .

To reduce overlap tests in BVH traversal (and thus C_{Query}), complementary strategies have been proposed including chromatic decompositions [Govindaraju et al. 2005a] and representative triangles [Curtis et al. 2008]. Curvature tests [Volino and Magnenat-Thalmann 1994] and normal bounds [Provot 1997; Grinspun and Schröder 2001] have been proposed to avoid overlap tests in smooth regions, and recently extended to continuous collision detection [Tang et al. 2009]. “Bounded Normal Trees” [Schvartzman et al. 2009] exploit certain reduced-order deformations based on convex combinations of linear transformations (but not modal deformations (1)) to intelligently update normal bounds hierarchically, with good performance obtained in areas of smooth geometry and deformations. In contrast, our method can efficiently cull self-collision tests even in areas of non-smooth geometry (see Figure 4, and our direct experimental comparison in Section 7). Deformation structure has also been exploited for SCD with polynomial spline patches [Hughes et al. 1996] and adaptive multigrid-based BVH updates [Otaduy et al. 2007]. For skeleton-driven deformation, Capell et al. [2007] exploited preprocessing for character deformations to reduce both C_{Update} and C_{Query} costs.

Our certificates are related to Kinetic Data Structures (KDS) for maintaining proximity information. Efficient updating of certificates is challenging [Guibas 2004]. Related KDS advances include $O(n \log n)$ SCD methods for bead-like “necklaces” [Guibas et al. 2002], and sparse kinetic $(1 + \epsilon)$ -spanner graphs for deforming point clouds [Gao et al. 2006]. Our paper provides self-collision-free certificates useful for moderate subspace deformations, with $O(r)$ evaluation cost. Self-collision processing has also been sched-



Figure 4: Subspace culling even for non-smooth surfaces: A smooth bunny (Left) is deformed (Middle) to its root bound’s limit along the first deformation mode. In contrast, a noisy bunny (Right) can achieve a similar magnitude, indicating similar subspace culling even for nonsmooth geometry.

uled in the context of asynchronous contact mechanics [Harmon et al. 2009], with an emphasis on dynamical correctness and not efficient SCD. Our certificates correspond to closest triangle pairs in the r -dimensional subspace distance metric. Methods for tracking and updating closest features are widely used for inter-object collision detection [Lin and Canny 1991; Mirtich 1998].

Recently there has been increased interest in exploiting GPUs for SCD. Examples include z -buffer image-based techniques [Heidelberger et al. 2004], hardware-accelerated visibility queries to quickly find potentially colliding primitives [Govindaraju et al. 2005b], and rasterization and Voronoi methods that greatly reduce extraneous overlap tests [Sud et al. 2006]. These GPU-based methods apply to general deformations, however subspace bounds could help them exploit deformation structure to avoid $\Omega(N)$ costs.

3 Background: BVH-based SCD

A triangulated deformable model is *self-colliding*, by definition, if there exist two *non-adjacent* triangles that are colliding. Two triangles are considered non-adjacent if they do not share any vertices. If needed, the SCD definition could be modified, without altering our method, to also include collisions of non-neighboring features (vertex-face and edge-edge) of two neighboring triangles. Note that self-collisions are unaffected by rigid motion. Therefore, for subspace simulations, we do not include rigid motion in the basis \mathbf{U} , but resolve self-collisions in the local object frame of reference.

Brute-force SCD checks all triangle pairs, an $O(N^2)$ operation. The most common approach to accelerate SCD is to fit a bounding volume hierarchy (BVH) to the triangle mesh, and then query the BVH against itself (see Algorithm 1, without lines 3, 4, 17, 18). The code is invoked by `Intra-collision_test(root)`. Even in the best possible case, however, the time complexity is going to be $\Omega(N)$, because the algorithm has to recurse to pairwise triangle tests everywhere on the model, to resolve geometry that is both non-neighboring and geometrically close. Our collision certificates are designed to integrate with this standard SCD mechanism. When a certificate is valid, it can rule out self-contact for an entire BVH subtree at the cost of merely comparing two scalar floating-point values, leading to large computational savings. When invalid, the algorithm defaults to the standard BVH traversal, ensuring a correct result.

The BVH can be constructed using any of the standard approaches. We recursively partition the mesh using k -means clustering, combining geometric position and the modes [Huang et al. 2009]. Such clustering strengthens our certificates, because it tends to assign topologically distant mesh regions to different nodes of the BVH, even when geometrically close (see Figure 5). Every BVH node n is responsible for bounding a subset of the triangles, T_n . Our partition is disjoint: if n is non-leaf, then T_n is a disjoint union of $T_{n'}$,

where n' runs over all children of n (see Figure 5). We terminate the recursion when the size of T_n drops below a specified threshold.

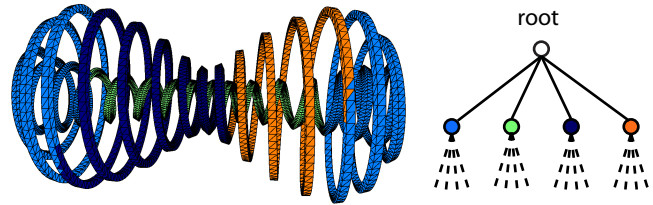


Figure 5: The hierarchy: (Left) Disjoint sets of triangles belonging to the four nodes at level 1 of the spring model. (Right) The bounding volume hierarchy.

We use axis-aligned bounding boxes (AABBs) as our bounding volumes (BVs), which are a common choice for SCD [van den Bergen 1997]. For a standard self-collision query, the BVs needs to be updated at every timestep to ensure that each BV still covers its triangles. The BVH update cost is often not negligible. In our examples, it is typically less, but close to the cost of the subsequent SCD query, e.g., the ratio of the costs is about 1:2 in the bunny example (see Figure 1). We adopt the BD-Tree approach to lazily update AABBs (at $O(r)$ cost) during the BVH-BVH traversal [James and Pai 2004]. The update is performed upon the first encounter of a BV during the traversal, on line 19 of Algorithm 1. With lazy updates, both the BV updates and overlap tests in a subtree are avoided when a certificate culls a subtree. Alternatively, our method could be used with standard AABB updating techniques that update the entire BVH before the query, in which case it only reduces C_{Query} .

4 Collision-Free Certificates

Our collision bounds are *certificates*, which if satisfied, guarantee a self-collision-free configuration for a subtree of the BVH, or between two subtrees. For example, given two tree nodes n and m , our *inter-node* certificate test can guarantee that no triangle from T_n collides with a non-neighboring triangle from T_m provided that $\|\mathbf{q}\|_{\mathbf{M}} < \mathcal{R}_{nm}$, whereas, if false, it is inconclusive. Here $\mathbf{q} = \mathbf{q}(t)$ are the current reduced coordinates, and we use the \mathbf{M} -weighted 2-norm, $\|\mathbf{q}\|_{\mathbf{M}} \equiv \sqrt{\mathbf{q}^T \mathbf{M} \mathbf{q}}$ for some positive definite (typically diagonal) constant matrix, \mathbf{M} , so that the collision-free certificate corresponds to an ellipsoidal region in \mathbf{q} space (see Figure 6). The pre-computed certificates $\mathcal{R}_{nm} \geq 0$ correspond to the subspace distance to the nearest self-contact, and are strictly non-decreasing as one descends the hierarchy—in practice they grow large very quickly (see Figure 7). *Intra-node* certificates for self-collisions within a node n can be seen as a special case with $m = n$; we adopt the notation that $\mathcal{R}_n = \mathcal{R}_{nn}$. The matrix \mathbf{M} can be chosen to maximize the utility of the bound given prior knowledge of subspace deformations; we use diagonal matrices \mathbf{M} for ease of computation, with the particular construction described later in §6.

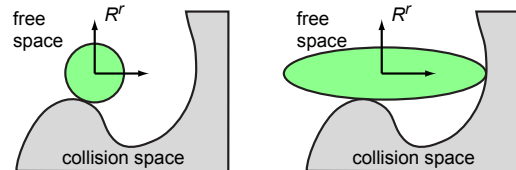


Figure 6: Collision-free certificates correspond to ellipsoids in the \mathbf{q} -space: (Left) $\mathbf{M} = \mathbf{I}$ produce spherical bounds, whereas we use (Right) diagonal $\mathbf{M} \neq \mathbf{I}$, giving axis-aligned ellipsoids.

We can easily accelerate traditional BVH-BVH SCD traversals by adding only **two additional comparisons** (in red) (see Algorithm 1). Given the configuration \mathbf{q} , the algorithm first evaluates and caches

Algorithm 1: BVH-based self-collision detection with SSCC

```

1 Procedure Intra-collision_test( $n$ )
  Input: BVH node  $n$ ,  $\|\mathbf{q}\|_{\mathbf{M}}$ 
  Output: Colliding non-neighborng triangle pairs from  $T_n$ 
2 begin
3   if  $\|\mathbf{q}\|_{\mathbf{M}} < \mathcal{R}_n$  then
4     return // certificate culls collision-free node
5   if  $n$  is leaf then
6     for all non-neighborng triangles  $\tau_1, \tau_2 \in T_n$  do
7       if triangles  $\tau_1$  and  $\tau_2$  collide then
8         Report ( $\tau_1, \tau_2$ )
9   else
10    for all children  $n'$  of  $n$  do
11      Intra-collision_test( $n'$ )
12    for all distinct children pairs  $n', m'$  of  $n$  do
13      Inter-collision_test( $n', m'$ )
14 end

15 Procedure Inter-collision_test( $n, m$ )
  Input: BVH nodes  $n, m$ ,  $\|\mathbf{q}\|_{\mathbf{M}}$ 
  Output: Colliding non-neighborng triangle pairs  $\tau_1 \in T_n, \tau_2 \in T_m$ 
16 begin
17   if  $\mathcal{R}_{nm}$  is cached and  $\|\mathbf{q}\|_{\mathbf{M}} < \mathcal{R}_{nm}$  then
18     return // certificate culls collision-free nodes
19   if BVs of  $n$  and  $m$  do not intersect then
20     return
21   if both  $n$  and  $m$  are leaves then
22     for all non-neighborng  $\tau_1 \in T_n, \tau_2 \in T_m$  do
23       if triangles  $\tau_1$  and  $\tau_2$  collide then
24         Report ( $\tau_1, \tau_2$ )
25   else
26      $N \leftarrow (n \text{ or } m \text{ is leaf}) ? \text{the non-leaf} : \text{higher node } (n \text{ if tie})$ 
27      $M \leftarrow \text{the other node}$ 
28     for all children  $n'$  of  $N$  do
29       Inter-collision_test( $n', M$ )
30 end

```

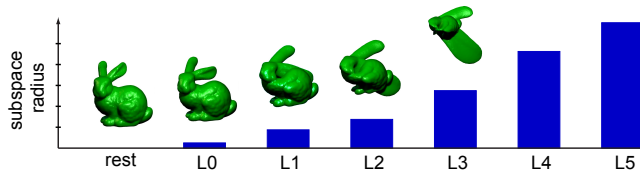


Figure 7: Certificate subspace radii, \mathcal{R}_n , increase at finer geometric scales as shown here by the plot of the median radius per tree level (bunny, $r = 30$). Consequently our self-collision culling abilities improve at finer scales. The bunny images show the magnitude of the median-radius deformation, along the first modal direction. The L0 radius corresponds to the root bound (see Figure 4). L1 radii (and higher) are severe deformations in this example.

$\|\mathbf{q}\|_{\mathbf{M}}$, which is an $O(r)$ operation (for \mathbf{M} diagonal) that only needs to be performed once for each \mathbf{q} . To process a node n , the algorithm first checks the subspace certificate: if $\|\mathbf{q}\|_{\mathbf{M}} < \mathcal{R}_n$ there can be no self-collision in the subtree of n , so there is no need for further recursion (lines 3,4 in Algorithm 1). Similarly, for an inter-node query between nodes $n \neq m$, the algorithm first determines if the inter-node certificate \mathcal{R}_{nm} has been precomputed, then (if so) if $\|\mathbf{q}\|_{\mathbf{M}} < \mathcal{R}_{nm}$ the nodes are collision-free and recursion is terminated (lines 17,18 in Algorithm 1). In practice, augmenting traditional

BVH self-collision traversals with our simple $\|\mathbf{q}\|_{\mathbf{M}}$ tests provides cheap and effective collision culling. Note that the certificates are useful even if the model is (nearly) self-colliding in the rest configuration, in which case the root certificate will be (nearly) zero. However, parts of the model that are far from self-colliding at rest will be culled by large certificates at their respective BVH nodes.

Storing all intra-node bounds \mathcal{R}_n requires only $O(N)$ storage, whereas storing all \mathcal{R}_{nm} would require $O(N^2)$ storage. Intra-node certificates can avoid the otherwise inevitable self-collision query recursion to all pairs of neighboring triangles in a subtree. In general, they have stronger culling performance than the inter-node certificates. This is because a pair of BVs can often cull collisions between two distinct tree nodes efficiently, without needing the certificate. Therefore, we compute and store all intra-node certificates \mathcal{R}_n , but only compute and store inter-node certificates \mathcal{R}_{nm} , where n and m are children of the root node, for a total of $n + 6$ certificates (our branching factor is 4). It would also be possible to hash inter-node bounds where n and m are either children of the same node (not necessarily root), or have neighboring triangles.

5 Certificate Precomputation

We now describe how we precompute the intra- and inter-collision certificates \mathcal{R}_n and \mathcal{R}_{nm} , respectively. Given the scaling matrix \mathbf{M} and any two subsets of the triangle model Ω and Ω' , we define the collision bound $R(\Omega, \Omega') \in [0, \infty]$ as the smallest value of $\|\mathbf{q}\|_{\mathbf{M}}^2 = \mathbf{q}^T \mathbf{M} \mathbf{q}$ over all configurations \mathbf{q} in which Ω and Ω' self-collide. Note that $\mathcal{R}_n = \sqrt{R(T_n, T_n)}$ and $\mathcal{R}_{nm} = \sqrt{R(T_n, T_m)}$.

Spherical scaling: For notational convenience, we can introduce a change of variables that transforms our ellipsoid bounds into spherical bounds. Let $\hat{\mathbf{M}}$ be the square root of the positive-definite matrix \mathbf{M} , $\hat{\mathbf{M}}^2 = \mathbf{M}$, $\hat{\mathbf{M}} > 0$. Then, the transformations

$$\hat{\mathbf{U}} = \mathbf{U} \hat{\mathbf{M}}^{-1}, \quad \hat{\mathbf{q}} = \hat{\mathbf{M}} \mathbf{q} \quad (3)$$

transform $\|\mathbf{q}\|_{\mathbf{M}}^2$ into an unweighted norm $\|\hat{\mathbf{q}}\|^2$. In the remainder of Section 5, we shall assume that this transformation has been made, and drop the hat superscript on \mathbf{U} and \mathbf{q} .

5.1 Collision Bound for Two Triangles

We now describe how we compute the collision bound $R(\tau_1, \tau_2)$ (denoted by R hereafter) for two non-neighborng triangles, τ_1 and τ_2 . These bounds will be used as leaf primitives for BVH certificates in Section 5.2. Each vertex of the two triangles deforms according to its $3 \times r$ submatrix of \mathbf{U} . The deformation modes anywhere on each triangle are defined using barycentric interpolation.

Constrained optimization formulation of R : Denote the rest positions of vertices of τ_i by $\mathbf{p}_i^1, \mathbf{p}_i^2, \mathbf{p}_i^3 \in \mathbb{R}^3$, and the modes at those vertices by $\mathbf{U}_i^1, \mathbf{U}_i^2, \mathbf{U}_i^3 \in \mathbb{R}^{3 \times r}$, for $i = 1, 2$. Then, $R(\tau_1, \tau_2)$ equals the minimum of the following optimization problem:

$$\text{minimize } \|\mathbf{q}\|^2, \quad \text{over all } \mathbf{q}, \alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, \quad (4)$$

$$\text{subject to } \alpha_i + \beta_i + \gamma_i = 1, \alpha_i \geq 0, \beta_i \geq 0, \gamma_i \geq 0, i = 1, 2, \quad (5)$$

$$\begin{aligned} \alpha_1 \mathbf{p}_1^1 + \beta_1 \mathbf{p}_1^2 + \gamma_1 \mathbf{p}_1^3 + (\alpha_1 \mathbf{U}_1^1 + \beta_1 \mathbf{U}_1^2 + \gamma_1 \mathbf{U}_1^3) \mathbf{q} = \\ \alpha_2 \mathbf{p}_2^1 + \beta_2 \mathbf{p}_2^2 + \gamma_2 \mathbf{p}_2^3 + (\alpha_2 \mathbf{U}_2^1 + \beta_2 \mathbf{U}_2^2 + \gamma_2 \mathbf{U}_2^3) \mathbf{q}. \end{aligned} \quad (6)$$

Formally, this is a non-convex quadratically constrained quadratic program, which are NP-hard and difficult to solve in general [Boyd and Vandenberghe 2004]. Our problem can be solved exactly by breaking it into 6 vertex-triangle and 9 edge-edge subproblems and applying Lagrange multipliers, resulting in 6×6 nonlinear cubic algebraic systems. We tried solving these systems using Gröbner bases in Mathematica, but the computation was slow: it typically took several seconds for a single pair of triangles with $r = 10$ modes.

A fast constant-mode approximation: We now give a fast approximation algorithm to the optimization problem of Equations 4–6, with a provable bound on approximation quality. Our algorithm produces a *conservative* estimate \hat{R} , satisfying $(1 - \varepsilon)R < \hat{R} \leq R$, where R is the exact minimum of (4, 5, 6) and ε can be made arbitrarily small at the cost of more computation (we use $\varepsilon=0.1$). We derive the approximation by observing that the optimization problem (4, 5, 6) becomes linearly constrained if one assumes *constant modes* across each triangle, i.e., if one replaces Equation 6 with

$$\alpha_1 \mathbf{p}_1^1 + \beta_1 \mathbf{p}_1^2 + \gamma_1 \mathbf{p}_1^3 + \mathbf{U}_1^c \mathbf{q} = \alpha_2 \mathbf{p}_2^1 + \beta_2 \mathbf{p}_2^2 + \gamma_2 \mathbf{p}_2^3 + \mathbf{U}_2^c \mathbf{q}, \quad (7)$$

where $\mathbf{U}_i^c = (\mathbf{U}_i^1 + \mathbf{U}_i^2 + \mathbf{U}_i^3)/3$, for $i = 1, 2$. The quadratic problem (4, 5, 7) has linear constraints and can be solved analytically as follows. The min-norm solution to (4, 5, 7) will always involve (a) a vertex of one triangle vs the other triangle, or (b) an edge of one triangle vs an edge of the other triangle (edges include endpoints). Namely, if a collision does not satisfy (a) and (b), one can perturb \mathbf{q} in all possible directions and still remain in collision; therefore, one can shrink $\|\mathbf{q}\|$ until either in case (a) or (b). It is therefore sufficient to compute the min-norms for all vertex-face and edge-edge pairs, and take their minimum. For vertex-face pairs, let $\mathbf{p}^0, \mathbf{p}^j, j = 1, 2, 3$, denote the rest positions of the vertex and triangle vertices, respectively. Let \mathbf{V} be the modes at the vertex, \mathbf{U}^c the triangle constant modes, and α, β, γ the barycentric coordinates in the triangle. Then, a short derivation using Lagrange multipliers shows that for each α, β , the optimal objective $Q = \|\mathbf{q}\|^2$ in (4) is a *quadratic* in α, β :

$$Q(\alpha, \beta) = c_{11}\alpha^2 + 2c_{12}\alpha\beta + c_{22}\beta^2 - 2c_{01}\alpha - 2c_{02}\beta + c_{00}, \quad (8)$$

$$\text{where } c_{ij} = (\mathbf{p}^i - \mathbf{p}^3)^T ((\mathbf{U}^c - \mathbf{V})(\mathbf{U}^c - \mathbf{V})^T)^{-1} (\mathbf{p}^j - \mathbf{p}^3). \quad (9)$$

The constant-mode vertex-triangle problem can therefore be solved by minimizing the quadratic $Q(\alpha, \beta)$ over the triangle. Edge-edge collisions have the same structure and are handled analogously.

Bounding the constant-mode approximation error: We now derive tight bounds $R^{\text{lo}} \leq R \leq R^{\text{hi}}$. If the modes vary little across the triangle, e.g., for small triangles, the constant-mode solution will be close to R , and the gap between R^{lo} and R^{hi} will be small. Otherwise, we resolve R more tightly using triangle subdivision.

Our upper bound is derived as follows. Let $\alpha_i, \beta_i, \gamma_i, i = 1, 2$, be the coordinates that minimize the constant-mode problem (4, 5, 7), and let $\mathbf{p}_i^c = \alpha_i \mathbf{p}_i^1 + \beta_i \mathbf{p}_i^2 + \gamma_i \mathbf{p}_i^3$ be the minimizing points. Our upper bound R^{hi} is the min-norm solution to the point-vs-point problem between \mathbf{p}_1^c and \mathbf{p}_2^c , where we use the true (i.e., barycentrically interpolated) modes for \mathbf{p}_i^c . Any such bound is greater than the minimum of (4, 5, 6), hence $R \leq R^{\text{hi}}$. Points \mathbf{p}_i^c are a good representative choice because they minimize the constant-mode problem, which converges to the original problem with decreasing triangle size.

Our lower bound follows by observing that the original problem (4, 5, 6) can be seen as obtained by perturbing the modes (but not geometry) of the constant-mode problem (4, 5, 7). We derived the following tight perturbation bounds (proofs in Appendix A, B):

$$R^{\text{lo}} \equiv \frac{1}{1 + \frac{\|\Delta\mathbf{U}_1\|_2 + \|\Delta\mathbf{U}_2\|_2}{\sigma_{\min}(\mathbf{U}_2^c - \mathbf{U}_1^c)}} R_c \leq R, \quad (10)$$

$$\|\Delta\mathbf{U}_i\|_2 \leq \frac{1}{3} \left(\max_{j=1,2,3} (\|\mathbf{U}_i^c - \mathbf{U}_i^j\|_2) + \sum_{j=1}^3 \|\mathbf{U}_i^c - \mathbf{U}_i^j\|_2 \right), \quad (11)$$

where R_c is the minimum to (4, 5, 7), $\sigma_{\min}(\cdot)$ is the smallest matrix singular value, and $\|\Delta\mathbf{U}_i\|_2$ is the maximum matrix 2-norm difference between the constant modes and the barycentrically interpolated modes, $\|\mathbf{U}_i^c - \mathbf{U}_i(X)\|_2$, over all triangle points $X \in \tau_i$. Note that $\|A\|_2 = \sigma_{\max}(A)$. We compute $\sigma_{\min}, \sigma_{\max}$ using SVD.

If $R^{\text{hi}} - R^{\text{lo}} < \varepsilon R^{\text{hi}}$, the ε -approximation criterion is satisfied, and we can conservatively approximate $\hat{R} := R^{\text{lo}}$. Otherwise, we connect edge midpoints to subdivide each triangle into four subtriangles, and invoke the algorithm recursively on 16 triangle pairs.

5.2 Certificates for Bounding Volumes

We now use the triangle-triangle collision bounds to precompute BV certificates \mathcal{R}_{nm} for all the required (sibling or neighboring) tree nodes (n, m) described in §4. The method is similar to a standard BVH query to determine the two nearest non-neighboring features in a triangle mesh, except it operates with distances in the reduced space as opposed to Euclidean 3D distances. If the bounds $R(\tau_1, \tau_2)$ were known for all pairs of triangles, we could simply set

$$\mathcal{R}_{nm}^2 = \min \{ R(\tau_1, \tau_2) \mid \tau_1 \in T_n, \tau_2 \in T_m, \tau_1, \tau_2 \text{ not neighbors} \}. \quad (12)$$

Clearly, processing all pairs of triangles for every tree node pair n, m of interest is not practical. We now give an algorithm that computes the same values \mathcal{R}_{nm} much more quickly, using a BVH.

Leaf-Leaf \mathcal{R}_{nm} Case: If both n and m are leaves, \mathcal{R}_{nm} can be computed via (12). This requires computing the tri-tri bounds $R(\tau_1, \tau_2)$, which can be done using the approximation algorithm of Section 5.1. While each such tri-tri computation is fast, many pairs need to be processed. The same values \mathcal{R}_{nm} can be computed faster (about $3 \times$ in our examples) as follows. Instead of computing $R(\tau_1, \tau_2)$ for all τ_1, τ_2 , we first compute a *fast* conservative approximation $\tilde{R}(\tau_1, \tau_2) \leq R(\tau_1, \tau_2)$. We then sort the \tilde{R} values, and compute $R(\tau_1, \tau_2)$ in order of ascending \tilde{R} . We can conservatively terminate the computation as soon as the smallest encountered bound R drops below the current \tilde{R} , thereby avoiding the unnecessarily accurate computation of large bounds for distant triangles.

The bound $\tilde{R}(\tau_1, \tau_2)$ is computed using a subspace analogue of traditional 1D space-time bounds. We fit the smallest enclosing ball to each triangle, and project the collision problem to the line joining the centers of the two triangles (reduction to 1D; see Figure 8, Left). Spheres project to line segments. For each mode, the 3-dim modal vectors at triangle vertices project to (signed) scalars. We bound the largest possible relative motion of the two line segments, u_{\max} , by computing the maximum absolute value difference between each scalar of the first triangle and each scalar of the second triangle (9 pairs). Our conservative bound $\tilde{R} \leq R$ is then $\tilde{R} = (d/u_{\max})^2$, where d is the 1D distance between the two intervals.

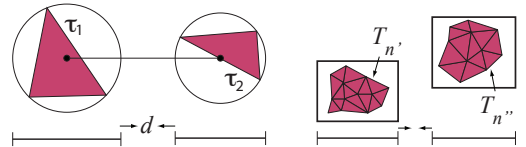


Figure 8: Geometry for conservative min-norm bounds via 1D projection: (Left) two triangles, (Right) two BVs.

Nonleaf \mathcal{R}_n Case: To compute \mathcal{R}_n for an interior node n , we recursively compute $\mathcal{R}_{n'}$, for all children nodes n' of n , and $\mathcal{R}_{n'n''}$ for all distinct pairs of children of n , and set \mathcal{R}_n to the minimum of these values. We employ three accelerations to this computation.

First, we maintain the smallest bound R_{\min} encountered in the subtree of n up to the current point in the algorithm, and pass it to all recursive computations of $\mathcal{R}_{n'}$ and $\mathcal{R}_{n'n''}$. These computations can then terminate early if they cannot produce a smaller bound.

Second, before starting the precomputation, we determine a global largest bound of interest, R_{global} , and then start every computation of \mathcal{R}_n with $R_{\min} = R_{\text{global}}$. We decided to add this optimization because we noticed that in practice, the bounds for small triangle

patches, deep in the tree, can be huge. Most often, the deformations encountered during the natural course of a simulation are not large enough to violate these bounds, so it is sufficient to resolve them up to the threshold R_{global} . We set $R_{\text{global}} = \|\mathbf{q}_1\|^2$, where \mathbf{q}_1 is the deformation, aligned with the lowest (first) mode, that causes some vertex to deform at least the distance of a geometric radius of the mesh. Such (adjustable) thresholding accelerated our precomputations by $5\times$, with a negligible reduction in runtime culling.

Third, when processing two distinct nodes n' and n'' , we first compute a fast conservative estimate $\tilde{R}(n', n'') \leq R(n', n'')$. If $\tilde{R}(n', n'') \geq R_{\text{min}}$, the pair n', n'' cannot contribute an improved min-norm bound for \mathcal{R}_n , so its processing is terminated. Such culling quickly localizes the computation to self-collision hotspots. Our estimate $\tilde{R}(n', n'')$ is similar to the 1D tri-tri estimate, but extended to the AABB-AABB case. It gives a conservative bound for the min-norm deformation under which the *bounding volumes* of n' and n'' enter contact (see Figure 8, Right); it is therefore also conservative for the covered geometry $T_{n'}$, $T_{n''}$. The BD-Tree method [James and Pai 2004] gives an explicit formula for a conservative AABB in configuration \mathbf{q} . Specifically, the AABB’s center tracks the average deformation of the covered triangles, while the size of the AABB, along the coordinate axis $D = 0, 1, 2$, grows according to a linear model, $\sum_{i=1}^r a_{i,D} |q_i|$, where q_i is the i -th component of \mathbf{q} , and $a_{i,D} \geq 0$ are properly precomputed BD-Tree scalar coefficients. We use this explicit model to bound the min-norm subspace distance to the AABB-AABB collision. We project both AABBs along each world-coordinate axis to obtain two intervals (see Figure 8, Right). For each mode i and each coordinate axis, the BD-Tree model gives the displacement of the center of the interval, and the growth of the size of the interval. One can then easily solve for the smallest $|q_i|$ under which the two intervals collide, and determine the mode $i = i_{\text{min}}$ where $|q_i|$ is the smallest. Finally, we set $\tilde{R}(n', n'')$ to the square of the largest $|q_{i_{\text{min}}}|$ among the three coordinate axes.

Nonleaf \mathcal{R}_{nm} Case: To compute \mathcal{R}_{nm} for a distinct pair $n \neq m$ where at least one node is non-leaf, we pick one of n, m for recursion (say, n), and then take the minimum of $\mathcal{R}_{n'm}$, over all children n' of n . We recurse on the non-leaf if one node is a leaf, and on the node higher in the hierarchy otherwise. Similar to \mathcal{R}_n , we employ the accelerations R_{min} and \tilde{R} described previously.

6 Scaling Matrices for Improved Culling

The scaling of the columns of \mathbf{U} can greatly influence the \mathcal{R} computation and the definition of minimal $\|\mathbf{q}\|_{\mathbf{M}}$ values. For example, if one column \mathbf{U}_k is scaled by 10^{-6} , then for the same displacement $\mathbf{u} = \mathbf{U}\mathbf{q}$ we would have registered a 10^6 larger value for its corresponding q_k , which would greatly alter the min-norm \mathbf{q} solution. Our method must therefore be invariant with respect to input scaling. We now describe how we select diagonal scaling matrices, \mathbf{M} , which, intuitively, weight individual modes so that $\|\mathbf{q}\|_{\mathbf{M}}$ will provide practical bounds for expected deformation ranges.

For models based on linear modal analysis, we scale each mode so that $\|\mathbf{q}\|_{\mathbf{M}}^2$ equals the linear elastic strain energy in configuration \mathbf{q} . The scaling for mode i equals $\omega_i^2/2$, where ω_i is the i -th natural model angular frequency. Our certificates then correspond to minimum strain energy levels that are self-collision free. Alternatively, for non-modal-analysis \mathbf{U} matrices, we first compute the reduced stiffness matrix, $\tilde{\mathbf{K}} = \mathbf{U}^T \mathbf{K} \mathbf{U}$, where \mathbf{K} is the model stiffness matrix at $\mathbf{q} = 0$. We then perform diagonalization, $\tilde{\mathbf{K}} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$, and replace \mathbf{U} by $\mathbf{U}' = \mathbf{U} \mathbf{Q}$. The matrix \mathbf{U}' spans exactly the same subspace as \mathbf{U} , but has the property that the reduced stiffness matrix, $\mathbf{\Lambda}$, is diagonal at $\mathbf{q} = 0$. We can then use diagonal weights $\mathbf{M} = \mathbf{\Lambda}$. Often, some frequencies in $\mathbf{\Lambda}$ will be very high, which could worsen the certificate culling ability, so we filter the elements of $\mathbf{\Lambda}$ using

the function $\lambda \mapsto (\frac{1}{\lambda} + \frac{1}{\ell \lambda_{\text{min}}})^{-1}$ (for $\ell = 100$). Note that we use the physical stiffness matrix and elastic strain energy to compute the weights, but other energy-like models could be used instead, including geometrical ones. Our scaling matrices \mathbf{M} are diagonal and performed well in our experiments. It would also be possible to use dense $r \times r$ matrices, without significant overhead (runtime or precomputation). Such matrices could rotate and scale the axes of the collision-free ellipsoids to further improve culling.

Simpler diagonal matrices \mathbf{M} include normalizing the modes by their largest vertex displacement, or by how much mass (or volume) is displaced. With such scalings, certificates essentially measure the least amount of mass displaced to reach contact. It is then optimal to cause self-collisions by bridging any narrow gaps with a localized deformation. We noticed that this becomes increasingly possible as r grows, because the modes can reconstruct a delta function increasingly better. For a small number of modes (e.g., 10-20), the results were similar to energy-based scalings. In order to support larger ranks, we decided to use aforementioned energy-based scalings, which scale better to higher rank, r (see Figure 9).

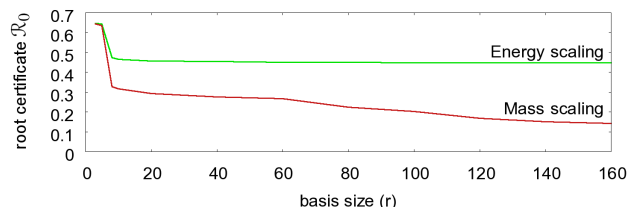


Figure 9: Energy-scaled certificates support high-rank deformations: In this experiment (spring example), the first mode’s weight is 1.0 for both methods, so the root certificate then equals the largest deformation along mode 1 that is self-collision free.

7 Results

Simulation statistics are available in Table 1. All results were run on an Apple Mac Pro (2×2.66 GHz Quad-Core Intel Xeon processor, 16 GB RAM) with GPU-accelerated rendering of deformations. Least-squares problems were solved robustly using QR decomposition. To perform comparisons, we use a traditional BVH-based self-collision method using AABBs (Algorithm 1, without the [lines 3, 4, 17, 18](#)). Our implementation is optimized for best performance (e.g., unrolled bounding volume traversal tree (BVTT) loops), and can perform SCD on a 15000-triangle Stanford bunny model in rest configuration at 33 Hz. In our examples, we combined SCD with a penalty-based volumetric self-contact resolution method, to demonstrate that our certificates can provide large speedups for self-contacting simulations. Specifically, after performing SCD and identifying self-colliding pairs of triangles, penalty forces push colliding triangles out of the volume of the other object using material distances [Hirota et al. 2000]. Our self-collision detection algorithm, however, is not specific to the penalty contact model, or any other model to compute contact forces once the colliding triangles have been identified. All collisions (inter-object and self-collisions) are resolved down to the individual triangles. Larger speedups would result from approximate inter-object collisions.

SSCC can greatly accelerate SCD for reduced-order multi-body dynamics. A $29\times$ SCD speedup was observed when simulating 16 falling bunnies (see Figure 1 and 12) each with ear-related self-collision hotspots that very visibly self-penetrate without SCD. Large-deformation dynamics were simulated using reduced-order FEM models [Barbič and James 2005], and inter-object collisions use AABB BD-Trees. Although our certificates stop becoming useful under large deformations, most objects are not deformed to large deformations at any given moment of time, so the certificate

Example	tri	r	#	dynamics	precomp	sim	inter-obj	self	SSCC	speedup _{SCD}	speedup _{SIM}
spring (sound)	16,000	20	1	linear	29 sec	4.3 min	0 msec	30 msec	0.32 msec	127.1×	79.1×
bunny (16)	14,964	30	16	nonlinear	113 sec	59 min	20 msec	640 msec	22 msec	29.1×	10.9×
bunny (128)	14,964	30	128	nonlinear	113 sec	11.7 hours	305 msec	3797 msec	255 msec	14.9×	6.4×
dragon (nonlinear)	77,203	30	1	nonlinear	75 min	31 min	0 msec	443 msec	28 msec	15.6×	15.1×
dragon (polynomial)	77,203	24	1	linear	89 min	7.5 min	0 msec	381 msec	3.7 msec	102.9×	101.7×

Table 1: Simulation statistics for #triangles (*tri*), subspace rank (*r*), #simulation objects (#), precomputation (*precomp*) and total simulation time (*sim*); average per-frame times for inter-object collisions (*inter-obj*), and SCD without (*self*), and with our certificates (*SSCC*); and SCD speedup (*speedup_{SCD}*) and overall simulation speedup (*speedup_{SIM}*). Dynamics and ground contact times are small (1-10 msec).

pre-processing costs amortize in time and across many objects. A larger 128-bunny simulation (~2 million triangles) produced more energetic inter-object and self collisions, larger deformations, and a somewhat smaller $15\times$ SCD speedup. However, both SSCC-based speedups were sufficient to remove SCD bottlenecks from collision processing. A complex dragon mesh (77k triangles) demonstrates further triangle-level scalability of the method (see Figure 10).

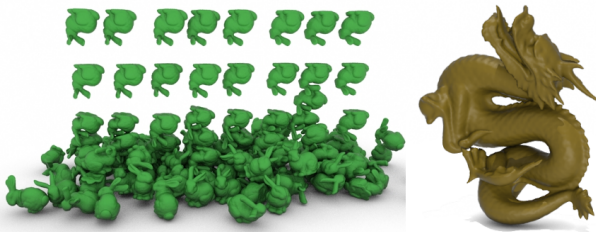
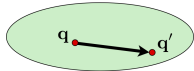


Figure 10: Complex self-collision detection simulations can be computed an order of magnitude faster due to the accelerations offered by our certificates: (Left) 128 bunnies (1,915,392 triangles); (Right) the dragon (77k triangles) with multiple self-collision sites.

Our method can also accelerate continuous SCD. Each collision-free certificate bounds a convex region in the reduced-coordinate space ($\|\mathbf{q}\|_M < \mathcal{R}_{nm}$). If the bound is satisfied for $\mathbf{q}(t)$ and $\mathbf{q}(t + \Delta t)$, then, assuming a straight-line trajectory, it is also satisfied at all intermediate times, $t \in [t, t + \Delta t]$, thereby guaranteeing a collision-free trajectory between the two configurations. Continuous tests between bounding volumes, and leaf-level tests between triangles can then be performed using standard continuous methods (see [Bridson et al. 2002; Teschner et al. 2005]). We implemented such continuous SCD, and found that our certificates offer a speedup similar to the speedup in discrete SCD (see Table 2).



	traditional (AABB)	our method (SSCC)	speedup
discrete SCD	640 msec	22 msec	29.1×
continuous SCD	2559 msec	81 msec	31.6×

Table 2: Continuous SCD can be greatly accelerated by SSCC: Average SCD costs (per timestep) of the 16-bunny simulation (see Figure 1) demonstrate large continuous SCD speedups using SSCC.

We also implemented a standard normal-cone culling technique [Volino and Magnenat-Thalmann 1994], in which a subtree is culled if all triangle normals give a positive dot product with at least one of the 14 trial directions (all corners and face centers of a (unit) cube centered at the origin). Our method performed SCD $15.3\times$ and $128\times$ faster than the normal cones in the 16-bunny and spring examples, respectively. Note that the normal cone test is only approximate unless one also performs the ‘‘contour test,’’ which is often omitted for speed. In contrast, our certificates provide an unconditional mathematical proof of the absence of self-collisions, and have no preferred directions or normal orientations.

The root certificate provides a self-collision-free ellipsoid $\mathcal{E}_{\text{root}}$ centered around $\mathbf{q} = 0$. In this region, SCD can be performed without any BVH traversal, at a minimal additional cost to the simulation ($O(r)$ flops, Figure 11). In our examples, we used volumetric penalty forces (in the full geometric space) for collision response, but low-cost real-time simulations might trade speed for accuracy by constraining \mathbf{q} to the interior of $\mathcal{E}_{\text{root}}$, using contact responses modeled in the reduced space. Linear simulations can then be kept self-collision free and limited to modest deformations.

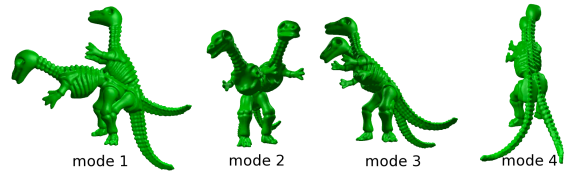


Figure 11: Extremal deformations of the root-node certificate ellipsoid, along the first 4 modal directions, in the positive and negative directions (56,192 triangles, $r = 18$, 9.8 min precomputation).

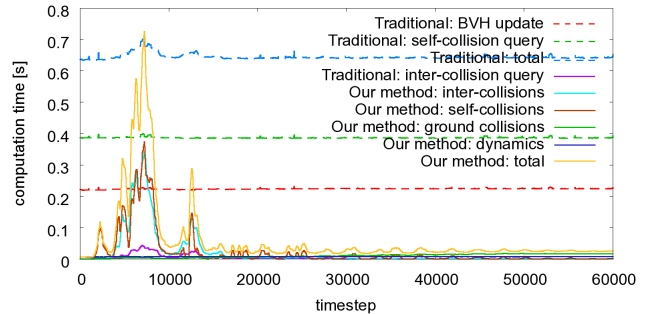


Figure 12: Our bounds amortize over time and over many objects, as seen in this timing diagram, corresponding to the 16-bunny sequence (see Figure 1). The traditional method (update AABB hierarchy + query) has to recurse to all pairwise neighboring triangle tests, leading to a near-constant computational cost.

Our certificates are especially useful with simulations running at high update rates, where certificates save a large amount of computation time relative to the precomputation costs. For example, our certificates enable efficient simulations of clattering sounds of self-contacting objects (see Figure 13). To illustrate this point, we time-stepped a linear spring model ($r = 20$) with SCD and self-contact resolution at audio rates (44,100 Hz). Ground and self-contact forces drive a 200-mode modal sound simulation. While SCD was a major bottleneck for self-collision sound synthesis, our experiment observed an $79\times$ speedup in SCD, demonstrating that SSCC can greatly alleviate such bottlenecks (see Table 3).

Geometrically based modal deformations can be constructed using quadratic polynomials of vertex rest positions [Witkin and Welch 1990]. Unlike linear modes (or nonlinear extensions), such modes vary smoothly, even across small geometric gaps. Consequently, such models are usually less prone to self collisions, which manifests as larger collision bounds (see Figure 14 and Table 1).

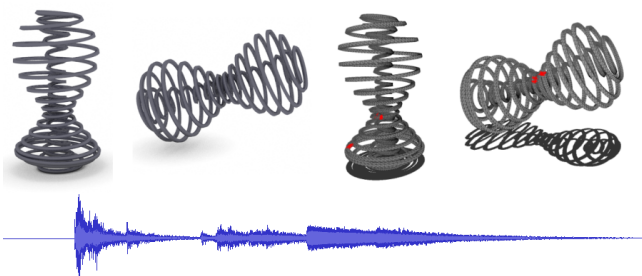


Figure 13: Self-collision sounds are produced by this clattering double-helix spring. Without SSCC, SCD would be 127× more expensive. Self-contacts are shown in red (on Right).

	SCD	modalDyn	groundCD	penalty	total
traditional	98%	1%	<1%	<0.01%	486 min
our method	42%	51%	6%	1%	6.1 min

Table 3: Cost breakdown for clattering sounds: SSCC overcomes the SCD bottleneck in traditional simulation, to make SCD costs comparable to modal dynamics (incl. 200-mode simulation); an 79× speedup in total simulation time is observed. Additional costs include ground CD, and penalty force response.

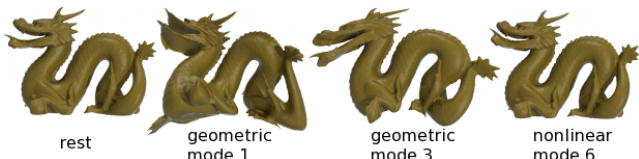


Figure 14: Our method supports geometrically defined modes (Middle), which usually exhibit larger bounds than modes from vibration analysis (Right, “mouth-closing” mode). Images show extremal collision-free deformations aligned with selected modes.

8 Conclusion

Subspace Self-Collision Culling (SSCC) can greatly accelerate SCD for mode-based deformable models. The method integrates with standard collision detection methods based on bounding volume hierarchies. It works by precomputing certificates which guarantee the absence of self-collisions. Our method is not sensitive to highly curved geometry, and is effective whenever detailed geometry is deformed with sufficiently smooth deformations. Our experiments demonstrate that SSCC certificates can provide large speedups in multi-body dynamics and sound simulations with deformable objects in self-contact and close proximity scenarios.

Limitations and Future Work: The most significant limitation is that our method does not support general deformations, however future work might generalize the technique by exploiting measures of deformation smoothness. Future work should investigate other reduced-order deformable models, including skeleton-based, multi-resolution, lattice-based, and embedded deformer. Certificate pre-computation is not interactive, and can not currently be used to exploit low-rank motion in general simulations. Our method assumes modal matrices with a full rank (three) variation across the mesh; any degenerate cases could be detected and handled separately. We have considered the magnitude of the deformation from $\mathbf{q} = \mathbf{0}$, i.e., $\|\mathbf{q}\|_M$, however bounds can also be computed about a nonzero reference configuration \mathbf{q}_0 by making the transformation $\mathbf{q} \rightarrow (\mathbf{q} - \mathbf{q}_0)$. We use k -means clustering to create our hierarchies, however hierarchies that explicitly separate opposing-gap triangles into different tree nodes might improve culling. Higher-rank deformations can have localized responses which can lead to weaker bounds. Improving the certificates, e.g., by designing multi-resolution \mathbf{M} weighting schemes or using local \mathbf{q} coordinates, is important future work.

Acknowledgements: This research was sponsored in part by the James H. Zumberge Research and Innovation Fund at the University of Southern California, the National Science Foundation (CAREER-0430528, HCC-0905506), the Alfred P. Sloan Foundation, the Boeing Company, Pixar, NVIDIA, Intel, and Autodesk.

A Proof of perturbation inequality (Eq. 10)

Given points p_1 and p_2 , with modal matrices $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{3 \times r}$, the unperturbed problem is to find the min-norm solution to $p_1 + \mathbf{U}_1 \mathbf{q} = p_2 + \mathbf{U}_2 \mathbf{q}$, that is, $\mathbf{U} \mathbf{q} = p$, for $\mathbf{U} = \mathbf{U}_2 - \mathbf{U}_1$ and $p = p_2 - p_1$. In the perturbed problem, one changes \mathbf{U} to $\mathbf{U} + \Delta \mathbf{U}$. Denote the min-norm solutions of the unperturbed and perturbed problems by \mathbf{q} and $\bar{\mathbf{q}}$, respectively. We then have $\mathbf{U} \bar{\mathbf{q}} = p - \Delta \mathbf{U} \bar{\mathbf{q}}$. The image of the unit ball in \mathbb{R}^r under \mathbf{U} (assuming \mathbf{U} has full rank, i.e., 3) contains the ball in \mathbb{R}^3 with radius $\sigma_{\min}(\mathbf{U})$. Therefore, there exists \mathbf{q}' such that $\mathbf{U} \mathbf{q}' = \Delta \mathbf{U} \bar{\mathbf{q}}$ and $\|\mathbf{q}'\| < \|\Delta \mathbf{U}\| \|\bar{\mathbf{q}}\| / \sigma_{\min}(\mathbf{U})$. Then, $\mathbf{U}(\bar{\mathbf{q}} + \mathbf{q}') = p$. Therefore, $\bar{\mathbf{q}} + \mathbf{q}'$ is a solution to the unperturbed problem, so we have $\|\bar{\mathbf{q}} + \mathbf{q}'\| \geq \|\mathbf{q}\|$. It follows that

$$\|\bar{\mathbf{q}}\| \geq \|\mathbf{q}\| - \frac{\|\Delta \mathbf{U}\| \|\bar{\mathbf{q}}\|}{\sigma_{\min}(\mathbf{U})}, \quad \text{and} \quad \frac{\|\mathbf{q}\|}{1 + \frac{\|\Delta \mathbf{U}\|}{\sigma_{\min}(\mathbf{U})}} \leq \|\bar{\mathbf{q}}\|. \quad (13)$$

B Proof of inequality for $\|\Delta \mathbf{U}\|_2$ (Eq. 11)

Given barycentric coordinates $\alpha_1, \alpha_2, \alpha_3$ of a point on a triangle,

$$\Delta \mathbf{U} = \mathbf{U}^c - \sum_{j=1}^3 \alpha_j \mathbf{U}^j = \sum_{j=1}^3 (\alpha_j - \frac{1}{3}) (\mathbf{U}^c - \mathbf{U}^j), \quad (14)$$

$$\Rightarrow \|\Delta \mathbf{U}\|_2 \leq \sum_{j=1}^3 |\alpha_j - \frac{1}{3}| k_j \stackrel{\text{def}}{=} F(\alpha_1, \alpha_2, \alpha_3), \quad (15)$$

where $0 \leq k_j \stackrel{\text{def}}{=} \|\mathbf{U}^c - \mathbf{U}^j\|_2$. Without loss of generality, assume that $k_1 = \max\{k_1, k_2, k_3\}$. If $\alpha_1 < \frac{1}{3}$, we can increase α_1 to $\frac{1}{3}$ while simultaneously decreasing α_2, α_3 . Because k_1 is maximal, this operation does not decrease $F(\alpha_1, \alpha_2, \alpha_3)$. Similarly, if $\alpha_1 \geq \frac{1}{3}$, we can use the inequality $|x - \frac{1}{3}| - \frac{1}{3} \leq x$, valid for all $x \geq 0$, to increase α_1 to 1 and decrease α_2, α_3 to 0, while not decreasing F . Therefore $\alpha_1 = 1, \alpha_2 = \alpha_3 = 0$ maximizes F , which gives Inequality 11.

References

- ALEXA, M., AND MÜLLER, W. 2000. Representing Animations by Principal Components. *Comp. Graphics Forum* 19, 3, 411–418.
- BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. on Graphics* 24, 3, 982–990.
- BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.
- BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust Treatment of Collisions, Contact, and Friction for Cloth Animation. *ACM Trans. on Graphics* 21, 3, 594–603.
- CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2007. Physically based rigging for deformable characters. *Graphical Models* 69, 1, 71–87.
- CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic Shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Transactions on Graphics* 28, 5, 1–10.

- CURTIS, S., TAMSTORF, R., AND MANOCHA, D. 2008. Fast collision detection for deformable models using representative-triangles. In *Proc. ACM Symp. Interactive 3D Graphics and Games*, 61–69.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 1997. Dynamic Free-Form Deformations for Animation Synthesis. *IEEE Trans. on Vis. and Comp. Graphics* 3, 3, 201–214.
- GAO, J., GUIBAS, L., AND NGUYEN, A. 2006. Deformable spanners and applications. *Computational Geometry: Theory and Appl.* 35, 1-2, 2–19.
- GOLUB, G. H., AND LOAN, C. F. V. 1996. *Matrix Computations*, 3rd ed. Johns Hopkins University Press.
- GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 1996. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *Proc. of ACM SIGGRAPH 96*, 171–180.
- GOVINDARAJU, N., KNOTT, D., JAIN, N., KABUL, I., TAMSTORF, R., GAYLE, R., LIN, M., AND MANOCHA, D. 2005. Interactive collision detection between deformable models using chromatic decomposition. *ACM Trans. on Graphics* 24, 3, 991–999.
- GOVINDARAJU, N., LIN, M., AND MANOCHA, D. 2005. Quick-CULLIDE: Fast inter-and intra-object collision culling using graphics hardware. In *Proc. IEEE Virtual Reality*, 59–66.
- GRINSPUN, E., AND SCHRÖDER, P. 2001. Normal bounds for subdivision-surface interference detection. In *IEEE Visualization 2001*, 333–340.
- GUIBAS, L., NGUYEN, A., RUSSEL, D., AND ZHANG, L. 2002. Collision Detection for Deforming Necklaces. In *Proc. of the ACM Symp. on Computational Geometry*, 33–42.
- GUIBAS, L. 2004. Kinetic Data Structures. In *Handbook of Data Structures and Applications*. Chapman and Hall/CRC.
- HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous Contact Mechanics. *ACM Transactions on Graphics* 28, 3, 87:1–87:12.
- HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2004. Detection of collisions and self-collisions using image-space techniques. *Journal of WSCG* 12, 3, 145–152.
- HIROTA, G., FISHER, S., AND LIN, M. C. 2000. Simulation of Non-penetrating Elastic Bodies Using Distance Fields. Tech. rep., University of North Carolina at Chapel Hill.
- HUANG, Q., WICKE, M., ADAMS, B., AND GUIBAS, L. 2009. Shape decomposition using modal analysis. *Computer Graphics Forum* 28, 2, 407–416.
- HUBBARD, P. M. 1995. *Collision Detection for Interactive Graphics Applications*. PhD thesis, Department of Comp. Science, Brown University.
- HUGHES, M., DIMATTIA, C., LIN, M., AND MANOCHA, D. 1996. Efficient and accurate interference detection for polynomial deformation. In *Proc. Computer Animation '96*, 155–166.
- JAMES, D. L., AND PAI, D. K. 2004. BD-Tree: Output-Sensitive Collision Detection for Reduced Deformable Models. *ACM Trans. on Graphics* 23, 3, 393–398.
- KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Transactions on Graphics* 27, 5, 164:1–164:11.
- KLOSOWSKI, J. T., HELD, M., MITCHELL, J. S. B., SOWIZRAL, H., AND ZIKAN, K. 1998. Efficient Collision Detection Using Bounding Volume Hierarchies of k -DOPs. *IEEE Trans. Vis. & Comp. Graphics* 4, 1, 21–36.
- LARSSON, T., AND AKENINE-MÖLLER, T. 2001. Collision detection for continuously deforming bodies. *Eurographics 2001*, 325–333.
- LARSSON, T., AND AKENINE-MÖLLER, T. 2003. Efficient collision detection for models deformed by morphing. *The Visual Computer* 19, 2, 164–174.
- LIN, M., AND CANNY, J. 1991. A fast algorithm for incremental distance calculation. In *IEEE ICRA*, vol. 2, 1008–1014.
- MEYER, M., AND ANDERSON, J. 2007. Key Point Subspace Acceleration and Soft Caching. *ACM Trans. on Graphics* 26, 3, 74:1–74:8.
- MIRTICH, B. 1998. V-Clip: Fast and Robust Polyhedral Collision Detection. *ACM Trans. on Graphics* 17, 3, 177–208.
- OTADUY, M. A., GERMANN, D., REDON, S., AND GROSS, M. 2007. Adaptive Deformations with Fast Tight Bounds. In *Symp. on Computer Animation (SCA)*, 181–190.
- PENTLAND, A., AND WILLIAMS, J. 1989. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics (Proc. of ACM SIGGRAPH 89)* 23, 3, 215–222.
- PROVOT, X. 1997. Collision and Self-Collision Handling in Cloth Model Dedicated to Design Garments. In *Graphics Interface*, 177–189.
- SCHVARTZMAN, S. C., GASCÓN, J., AND OTADUY, M. A. 2009. Bounded normal trees for reduced deformations of triangulated surfaces. In *Symp. on Computer Animation (SCA)*, 75–82.
- SUD, A., GOVINDARAJU, N., GAYLE, R., KABUL, I., AND MANOCHA, D. 2006. Fast Proximity Computation Among Deformable Models using Discrete Voronoi Diagrams. *ACM Trans. on Graphics* 25, 3, 1144 – 1153.
- TANG, M., CURTIS, S., YOON, S.-E., AND MANOCHA, D. 2009. Interactive continuous collision detection between deformable models using connectivity-based culling. *IEEE Trans. on Visualization and Computer Graphics* 15, 544–557.
- TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., POMERANETS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. In *Proc. Vision, Modeling, and Visualization Conference*, 47–54.
- TESCHNER, M., KIMMERLE, S., HEIDELBERGER, B., ZACHMANN, G., RAGHUPATHI, L., FUHRMANN, A., CANI, M., FAURE, F., MAGNENAT-THALMANN, N., STRASSER, W., AND VOLINO, P. 2005. Collision Detection for Deformable Objects. *Computer Graphics Forum* 24, 1, 61–81.
- VAN DEN BERGEN, G. 1997. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *J. of Graphics Tools* 2, 4, 1–14.
- VOLINO, P., AND MAGNENAT-THALMANN, N. 1994. Efficient Self-Collision Detection on Smoothly Discretized Surface Animations using Geometrical Shape Regularity. *Comp. Graphics Forum* 13, 3, 155–166.
- WITKIN, A., AND WELCH, W. 1990. Fast Animation and Control of Nonrigid Structures. *Computer Graphics (Proc. of ACM SIGGRAPH 90)* 24, 4, 243–252.