# Kinetic Collision Detection for Balls Rolling on Plane

Mohammad Ali Abam* Mark de Berg* Sheung-Hung Poon* Bettina Speckmann*


## Abstract

This abstract presents a first step towards kinetic collision detection in 3 dimensions. In particular, we design a compact and responsive kinetic data structure (KDS) for detecting collisions between $n$ balls of arbitrary sizes rolling on a plane. The KDS has size $O(n \log n)$ and can handle events in $O(\log n)$ time. The structure processes $O(n^2)$ events in the worst case, assuming that the objects follow low-degree algebraic trajectories. The full paper [1] presents additional results for convex fat 3-dimensional objects that are free-flying in $\mathbb{R}^3$.


## 1 Introduction

Collision detection is a basic computational problem arising in all areas of computer science involving objects in motion—motion planning, animated figure articulation, computer simulated environments, or virtual prototyping, to name a few. Very often the problem of detecting collisions is broken down into two phases: a *broad phase* and a *narrow phase*. The broad phase determines pairs of objects that might possibly collide, frequently using (hierarchies of) bounding volumes to speed up the process. The narrow phase then uses specialized techniques to test each candidate pair, often by tracking closest features of the objects in question, a process that can be sped up significantly by exploiting spatial and temporal coherence. See [13] for a detailed overview of algorithms for such collision and proximity queries.

Algorithms that deal with objects in motion traditionally discretize the time axis and compute or update their structures based on the position of the objects at every time step. But since collisions tend to occur rather irregularly it is nearly impossible to choose the perfect time-step: too large an interval between sampled times will result in missed collisions, too small an interval will result in unnecessary computations (and still there is no guarantee that no collisions are missed). Event-driven methods, on the other hand, compute the event times of significant changes to a system of moving objects, store those in a priority queue sorted by time, and advance the system to the event at the front of the queue. The kinetic data structure (KDS) framework initially introduced by Basch et al. [4] presents a systematic way to design and analyze event-driven data structures for moving objects (see [7] and [8] for surveys on kinetic data structures).

A kinetic data structure is designed to maintain or monitor a discrete attribute of a set of moving objects, where each object has a known motion trajectory or *flight plan*. A KDS contains a set of *certificates* that constitutes a proof of the property of interest. These certificates are inserted in a priority queue (*event queue*) based on their time of expiration. The KDS then performs an event-driven simulation of the motion of the objects, updating the structure whenever a certificate fails. A KDS for collision detection finds a set of geometric tests (elementary certificates) that together provide a proof that the input objects are disjoint.

Kinetic data structures and their accompanying maintenance algorithms can be evaluated and compared with respect to four desired characteristics. A good KDS is *compact* if it uses little space in addition to the input, *responsive* if the data structure invariants can be restored quickly after the failure of a certificate, *local* if it can be updated easily when the flight plan for an object changes, and *efficient* if the worst-case number of events handled by the data structure for a given motion is small compared to some worst-case number of "external events" that must be handled for that motion.

**Kinetic data structures for collision detection.** One of the first papers on kinetic collision detection was published by Basch et al. [3], who designed a KDS for collision detection between two simple polygons in the plane. Their work was extended to an arbitrary number of polygons by Agarwal et al. [2]. Kirkpatrick et al. [11] and Kirkpatrick and Speckmann [12] also described KDS's for kinetic collision detection between multiple polygons in the plane. These solutions all maintain a decomposition of the free space between the polygons into "easy" pieces (usually pseudo-triangles). Unfortunately it seems quite hard to define a suitable decomposition of the free space for objects in 3D, let alone maintain it while the objects move—the main problem being, that

*Department of Mathematics and Computer Science, TU Eindhoven, m.a.abam@tue.nl, m.t.d.berg@tue.nl, spoon@win.tue.nl, speckman@win.tue.nl. M.A. and S.-H.P. were supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 612.065.307. M.d.B. was supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 639.023.301.

all standard decomposition schemes in 3D can have quadratic complexity. Hence, even though collision detection is the obvious application for kinetic data structures, there has hardly been any work on kinetic collision detection in 3D.

There are only a few papers that deal directly with (specialized versions of) kinetic 3D collision detection. Guibas et al. [9], extending work by Erickson et al. [6] in the plane, show how to certify the separation of two convex polyhedra moving rigidly in 3D using certain outer hierarchies. Basch et al. [5] describe a structure for collision detection among multiple convex *fat* objects that have almost the same size. The structure of Basch et al. uses $O(n \log^2 n)$ storage and events can be processed in $O(\log^3 n)$ time. If all objects are spheres of related sizes Kim et al. [10] present an event-driven approach that subdivides space into cells and processes events whenever a sphere enters or leaves a cell. Unfortunately there is only experimental evidence for the performance of this structure. Finally, Guibas et al. [9] use the power diagram of a set of arbitrary balls in 3D to kinetically maintain the closest pair among them. The worst-case complexity of this structure is quadratic and it might undergo more than cubically many changes.

**Results.** In this abstract we describe a compact and responsive kinetic data structure for detecting collisions between $n$ balls of arbitrary sizes rolling on a plane. The KDS has size $O(n \log n)$ and can handle events in $O(\log n)$ time. It processes $O(n^2)$ events in the worst case, assuming that the objects follow low-degree algebraic trajectories.

## 2 Balls rolling on a plane

Assume that we are given a set $\mathcal{B}$ of $n$ 3-dimensional balls which are rolling on a 2-dimensional plane $T$, that is, the balls in $\mathcal{B}$ move continuously while remaining tangent to $T$. In this section we describe a responsive and compact KDS that detects collisions between the balls in $\mathcal{B}$.

The basic idea behind our KDS is to construct a *collision tree* recursively as follows:

- If $|\mathcal{B}| = 1$, then there are obviously no collisions and the collision tree is just a single leaf.
- If $|\mathcal{B}| > 1$, then we partition $\mathcal{B}$ into two subsets, $\mathcal{B}_S$ and $\mathcal{B}_L$. The subset $\mathcal{B}_S$ contains the $\lfloor n/2 \rfloor$ smallest balls and the subset $\mathcal{B}_L$ contains the $\lceil n/2 \rceil$ largest balls from $\mathcal{B}$, where ties are broken arbitrarily. The collision tree now consists of a root node that has an associated structure to detect collisions between any ball from $\mathcal{B}_S$ and any ball from $\mathcal{B}_L$, and two subtrees that are collision trees for the sets $\mathcal{B}_S$ and $\mathcal{B}_L$, respectively.

To detect all collisions between the balls in $\mathcal{B}$ it suffices to detect collisions between the two subsets maintained at every node of the collision tree. Let $\mathcal{B}_S$ and $\mathcal{B}_L$ denote the two subsets maintained at a particular node. The remainder of this section focusses on detecting collisions between the balls contained in $\mathcal{B}_S$ and $\mathcal{B}_L$. In particular, we describe a KDS of size $O(|\mathcal{B}_S| + |\mathcal{B}_L|)$ that can handle events in $O(1)$ time—see Theorem 5. The structure processes $O((|\mathcal{B}_S| + |\mathcal{B}_L|)^2)$ events in the worst case, assuming that the balls follow low-degree algebraic trajectories. Since the same event can occur simultaneously at $O(\log n)$ nodes of the collision tree, we obtain the following theorem:

**Theorem 1** *For any set $\mathcal{B}$ of $n$ 3-dimensional balls that roll on a plane, there is a KDS for collision detection that uses $O(n \log n)$ space and processes $O(n^2)$ events in the worst case, assuming that the balls follow low-degree algebraic trajectories. Each event can be handled in $O(\log n)$ time.*

### 2.1 Detecting collisions between small and large balls

As mentioned above, we can restrict ourselves to detecting collisions between balls from two disjoint sets $\mathcal{B}_S$ and $\mathcal{B}_L$ where the balls in $\mathcal{B}_L$ are at least as large as the balls in $\mathcal{B}_S$. Recall that all balls are rolling on a plane $T$. Our basic strategy is the following: we associate a region $D_i$ on $T$ with each $B_i \in \mathcal{B}_L$ such that if the point of tangency of a ball $B_j \in \mathcal{B}_S$ and $T$ is not contained in $D_i$, then $B_j$ can not collide with $B_i$. The regions associated with the balls in $\mathcal{B}_L$ need to have two important properties: ($i$) each point in $T$ is contained in a constant number of regions and ($ii$) we can efficiently detect whenever a region starts or stops to contain a tangency point when the balls in $\mathcal{B}_L$ and $\mathcal{B}_S$ move. We first deal with the first requirement, that is, we consider $\mathcal{B}_L$ to be static. For a ball $B_i$ let $r_i$ denote its radius and let $t_i$ be the point of tangency of $B_i$ and $T$.

**The threshold disk.** We define the distance of a point $q$ in the plane to a ball $B_i$ as follows. Imagine that we place a ball $B(q)$ of initial radius 0 at point $q$. We then inflate $B(q)$ while keeping it tangent to $T$ at $q$, until it collides with $B_i$. The radius of $B(q)$ equals the distance of $q$ and $B_i$ which we denote by $\mathrm{dist}(q, B_i)$. More precisely, $\mathrm{dist}(q, B_i)$ is the radius of the unique ball that is tangent to $T$ at $q$ and tangent to $B_i$. It is easy to show that $\mathrm{dist}(q, B_i) = d(q, t_i)^2/4r_i$ where $d(q, t_i)$ denotes the Euclidean distance between $q$ and $t_i$.

Since we have to detect collisions only with balls from $\mathcal{B}_S$ we can stop inflating when $B(q)$ is as large as the smallest ball in $\mathcal{B}_L$. Based on this, we define the threshold disk $D_i$ of a ball $B_i \in \mathcal{B}_L$ as follows: a point $q \in T$ belongs to $D_i$ if and only if $\operatorname{dist}(q, B_i) \leq r_{\min}$ where $r_{\min}$ is the radius of the smallest ball in $\mathcal{B}_L$. It is straightforward to show that $D_i$ is a disk whose radius is $2\sqrt{r_i \cdot r_{\min}}$ and whose center is $t_i$.

Clearly a ball $B_j \in \mathcal{B}_S$ can not collide with a ball $B_i \in \mathcal{B}_L$ as long as $t_j$ is outside $D_i$. In following, we prove that a point $q \in T$ can be contained in at most a constant number of threshold disks. For a given constant $c \geq 0$ let us denote with $c \cdot D_i$ a disk with radius $c \cdot \operatorname{radius}(D_i)$ and center $t_i$.

**Lemma 2** *The number of disks $D_j$ that are at least as large as a given disk $D_i$ and for which $c \cdot D_i \cap c \cdot D_j \neq \emptyset$, is at most $(8\,c^2 + 2\,c + 1)^2 + 1$.*

**Proof.** Let $\mathcal{D}(i)$ be the set of all disks $D_j$ that are at least as large as $D_i$ and for which $c \cdot D_i \cap c \cdot D_j \neq \emptyset$. First we prove that there are no two balls $B_j$ and $B_k$ such that $r_k \geq r_j > 16\,c^2\,r_i$ and $D_j, D_k \in \mathcal{D}(i)$. Assume, for contradiction, that there are two balls $B_j$ and $B_k$ such that $r_k \geq r_j > 16\,c^2\,r_i$ and $D_j, D_k \in \mathcal{D}(i)$. Since $B_j$ and $B_k$ are disjoint, we have

$$d(t_j, t_k) \geq 2\sqrt{r_j \cdot r_k} > 8\,c\sqrt{r_k \cdot r_i}\,.$$

On the other hand, we know that

$$\begin{aligned} d(t_j, t_k) &\leq d(t_j, t_i) + d(t_i, t_k) \\ &\leq (2\,c\sqrt{r_i \cdot r_{\min}} + 2\,c\sqrt{r_j \cdot r_{\min}}) + \\ &\quad\ (2\,c\sqrt{r_i \cdot r_{\min}} + 2\,c\sqrt{r_k \cdot r_{\min}}) \\ &< 8\,c\sqrt{r_k \cdot r_i} \end{aligned}$$

which is a contradiction. Hence, there is at most one ball $B_j$ such that $r_j > 16\,c^2\,r_i$ and $D_j \in \mathcal{D}(i)$.

It remains to show that the number of balls $B_j$ whose radii are not greater than $16\,c^2\,r_i$ and whose disks $D_j$ belong to $\mathcal{D}(i)$ is at most $(8\,c^2 + 2\,c + 1)^2$. Let $B_j$ be one of these balls and let $x$ be a point in $c \cdot D_j \cap c \cdot D_i$. Since

$$\begin{aligned} d(t_i, t_j) &\leq d(t_i, x) + d(t_j, x) \\ &\leq 2\,c\sqrt{r_i \cdot r_{\min}} + 2\,c\sqrt{r_j \cdot r_{\min}} \\ &\leq (2\,c + 8\,c^2)\,r_i \end{aligned}$$

$t_j$ must lie in a disk whose center is $t_i$ and whose radius is $(2\,c + 8\,c^2)\,r_i$. We also know that for any two such balls $B_j$ and $B_k$, $d(t_j, t_k) \geq 2\sqrt{r_j \cdot r_k} \geq 2\,r_i$ holds. Thus the set $\mathcal{D}'(i)$ of disks centered at $t_j$ with radius $r_i$ for all $D_j \in \mathcal{D}(i)$ are disjoint. Note that any disk in $\mathcal{D}'(i)$ lies inside the disk centered at $t_i$ with radius $((2\,c + 8\,c^2) + 1)\,r_i$. Thus $|\mathcal{D}(i)| = |\mathcal{D}'(i)| \leq \pi\,((2\,c + 8\,c^2 + 1)\,r_i)^2/\pi r_i^2 = (2\,c + 8\,c^2 + 1)^2$. □

**Lemma 3** *Each point $q \in T$ is contained in at most a constant number of threshold disks.*

**Proof.** Let $D_i$ be the smallest threshold disk containing $q$. Lemma 2 with $c = 1$ implies that the number of disks which are not smaller than $D_i$ and which intersect $D_i$ is constant. Hence the number of threshold disks containing $q$ is constant. □

The threshold disks have the important property that each point in $T$ is contained in a constant number of disks. But unfortunately, as the balls in $\mathcal{B}_L$ and $\mathcal{B}_S$ move, it is difficult to detect efficiently whenever a tangency point enters or leaves a threshold disk. Hence we replace each threshold disk by its axis-aligned bounding box. The bounding box of a threshold disk $D_i$ associated with a $B_i \in \mathcal{B}_L$ is called a *threshold box* and is denoted by $\mathrm{TB}(B_i)$. In the following we prove that the threshold boxes retain the crucial property of the threshold disk, namely, that each point $q \in T$ is contained in at most a constant number of threshold boxes.

**Lemma 4** *Each point $q \in T$ is contained in at most a constant number of threshold boxes.*

**Proof.** Instead of considering the threshold boxes directly, we consider the disks defined by the circumcircles $D(\mathrm{TB}(B_j))$ of each threshold box $\mathrm{TB}(B_j)$ with $B_j \in \mathcal{B}_L$. Clearly we have $\operatorname{radius}(D(\mathrm{TB}(B_j))) = \sqrt{2} \cdot \operatorname{radius}(D_j)$ for all $B_j \in \mathcal{B}_L$. Let $\mathrm{TB}(B_i)$ be the smallest box containing $q$. Lemma 2 with $c = \sqrt{2}$ implies that the number of circumcircle disks which are at least as large as $D(\mathrm{TB}(B_i))$ and which intersect $D(\mathrm{TB}(B_i))$ is constant. Hence the number of threshold boxes which are not smaller than $\mathrm{TB}(B_i)$ and intersect $\mathrm{TB}(B_i)$ is constant and so is the number of threshold boxes containing $q$. □

**Kinetic maintenance.** Recall that to detect collisions between $\mathcal{B}_S$ and $\mathcal{B}_L$, for each ball $B_j \in \mathcal{B}_S$ we determine which threshold boxes contain the tangency point $t_j$. Note that according to Lemma 4, $t_j$ is contained in a constant number of threshold boxes. For each $B_j \in \mathcal{B}_S$ we maintain the set of threshold boxes that contain $t_j$ and certificates that guarantees disjointness of $B_j$ and the balls from $\mathcal{B}_L$ whose threshold boxes contain $t_j$.

To maintain our structure we only need to detect when a tangency point $t_j$ enters or leaves a threshold box. To do so, we maintain two sorted lists on the $x$- and $y$-coordinates of the tangency points of $\mathcal{B}_S$ and the extremal points of the threshold boxes associated with the balls in $\mathcal{B}_L$. Clearly the number of events processed by our structure is quadratic in the size of of $\mathcal{B}_S$ and $\mathcal{B}_L$ and each event can be processed in constant time. Unfortunately this structure is not local—a ball $B_i \in \mathcal{B}_L$ might be involved in a number of certificates that is linear in the size of $\mathcal{B}_S$.

**Theorem 5** *Let $\mathcal{B}_S$ and $\mathcal{B}_L$ be two disjoint sets of balls that roll on a plane where the balls in $\mathcal{B}_L$ are* *at least as large as the balls in $\mathcal{B}_S$. There is a KDS for collision detection between balls of $\mathcal{B}_S$ and balls of $\mathcal{B}_L$ that uses $O(|\mathcal{B}_S| + |\mathcal{B}_L|)$ space and processes $O((|\mathcal{B}_S| + |\mathcal{B}_L|)^2)$ events in the worst case if the balls follow low-degree algebraic trajectories. Each event can be handled in $O(1)$ time.*

## 3 Conclusions

This abstract describes a first step towards kinetic collision detection in 3 dimensions: a compact and responsive kinetic data structure for detecting collisions between $n$ balls of arbitrary sizes rolling on a plane. The full paper [1] presents additional results for convex fat 3-dimensional objects of constant complexity that are free-flying in $\mathbb{R}^3$. In that case we can detect collisions with a KDS of $O(n \log^6 n)$ size that can handle events in $O(\log^6 n)$ time. The structure processes $O(n^2)$ events in the worst case, assuming that the objects follow low-degree algebraic trajectories. If the objects have similar sizes then the size of the KDS becomes $O(n)$ and events can be handled in $O(1)$ time.

**Acknowledgements.** The last author would like to thank David Kirkpatrick for valuable discussions on the presented subject.

## References

[1] M. A. Abam, M. de Berg, S.-H. Poon, and B. Speckmann. Kinetic collision detection for convex fat objects. Submitted, 2005.

[2] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tilings for kinetic collision detection. *International Journal of Robotics Research*, 21:179–197, 2002.

[3] J. Basch, J. Erickson, L. J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection for two simple polygons. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 102–111, 1999.

[4] J. Basch, L. Guibas, and J. Hershberger. Data structures for mobile data. *Journal of Algorithms*, 31:1–28, 1999.

[5] J. Basch, L. Guibas, and L. Zhang. Proximity problems on moving points. In *Proc. 13th Symposium on Computational Geometry*, pages 344–351, 1997.

[6] J. Erickson, L. Guibas, J. Stolfi, and L. Zhang. Separation-sensitive collision detection for convex objects. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 327–336, 1999.

[7] L. Guibas. Kinetic data structures: A state of the art report. In *Proc. 3rd Workshop on Algorithmic Foundations of Robotics*, pages 191–209, 1998.

[8] L. Guibas. Motion. In J. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 1117–1134. CRC Press, 2nd edition, 2004.

[9] L. Guibas, F. Xie, and L. Zhang. Kineitc collision detection: Algorithms and experiments. In *Proc. International Conference on Robotics and Automation*, pages 2903–2910, 2001.

[10] D. Kim, L. Guibas, and S. Shin. Fast collision detection among multiple moving spheres. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):230–242, 1998.

[11] D. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. *International Journal of Computational Geometry and Applications*, 12(1&2):3–27, 2002.

[12] D. Kirkpatrick and B. Speckmann. Kinetic maintenance of context-sensitive hierarchical representations for disjoint simple polygons. In *Proc. 18th ACM Symposium on Computational Geometry*, pages 179–188, 2002.

[13] M. Lin and D. Manocha. Collision and proximity queries. In J. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 787–807. CRC Press, 2nd edition, 2004.