Velocity-Aligned Discrete Oriented Polytopes for Dynamic Collision Detection

Daniel Coming and Oliver Staadt
presented by: Marek Kováčik

Physical-based Animations and Mathematical Modeling

27th of December 2010

Motivation

Collision detection and response of dynamic objects is a common requirement in many applications, such as:

- Virtual reality
- Computer animation
- Robotics
- Physics-based simulations

Collision detection and response can be thought of as a pipeline to visualization pipeline.

The pipeline starts with the description of objects, determines which pairs of intersecting objects are of interest and through multiple stages of filters, it prunes pairs of objects that do not intersect.

This process can be divided into two stages:

- Broad phase
- Narrow phase

Narrow phase

Narrow phase consists of precise object intersection tests. The three primary classifications of narrow phase are:

- Static collision detection - simple 3D intersection tests on non-moving objects. It is not applicable in interactive dynamic applications.
- Pseudo-dynamic collision detection - an extension of static method to account for moving objects is to consider objects non-moving for instantaneous moments and perform static collision detection in small time increments.
- Dynamic collision detection - detecting collisions in the time interval between consecutive frames. It requires solving parametrized equations involving both the position and the velocity of the object to determine the first time of intersection of the objects.

Broad phase

The collision-pair pruning is performed in the broad phase. The most frequent methods are:

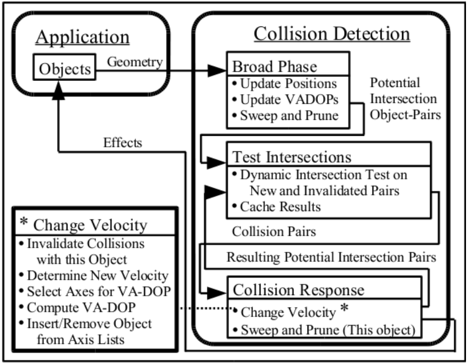
- Uniform space division
- Bounding volume intersection (AABB, OBB, k-DOP, . . .)
- Sweep and Prune

Contribution

The presented paper offers the following contributions for performing dynamic collision detection at interactive rates:

- Velocity-aligned discrete oriented polytopes (VADOP) - is a bounding volume based on k-DOPs. VADOPs offer faster update times than k-DOPs and are well adapted for dynamic collision detection and high object velocities.
- An extension and optimization of sweep and prune method to work with VADOPs in order to overcome the low-level velocity limitation of the original method.

Process overview



Process overview

How does it work?

- Initially, the application passes objects' geometry to the broad phase, which uses VADOP bounding volumes and sweep and prune to determine potential object-pair collisions.
- Next, the narrow phase involves dynamic collision tests for any pairs of objects that were not pruned during the broad phase. The colliding pairs are queued in time-order.
- Then, collision response is performed for each pair from the queue. The effects of the collisions are then sent back to the application.

Environment

- A linear motion is assumed for a short period of time (time from one rendered frame to the next).
- Temporal coherence - a condition in which the changes in the state of the application are small between time steps (4,2.beg).
- Only a slight change in object's velocity direction are assumed and large changes do not occur frequently (only when caused by collisions).

Collision detection tools

Separating axis theorem

Objects do not intersect if there exists a separating axis - a line for which the objects intervals of projection do not intersect.

SAT along with bounding volumes is used in the broad phase to simplify preliminary tests to reduce the number of expensive dynamic collision intersection tests.

Collision detection tools

Sweep and prune

Cohen et al. proposed a technique to optimize overlap tests in AABB (k-DOP) bounding volumes.

- For each axis of a bounding volume there is a pair of corresponding values for the minimum and maximum extrema of the bounding volume projected onto that axis.
- The extrema of the objects are put into the list and sorted by insertion sort.

Collision detection tools

Sweep and prune cont.

- Each time a swap is made, a value in a boolean matrix is toggled between true and false, representing whether the corresponding objects overlap along the axis associated with given list.

Object position updates exploit temporal coherence (low velocity - minor changes - nearly sorted - linear time).

Collision detection tools

Swept volumes

- When performing dynamic collision detection, we must consider the possibility of collisions over a continuous interval of time.
- Therefore, when performing broad phase detection, we must consider a bounding volume that is enclosing all the positions of the object over that time interval.
- If we assume a linear motion between consecutive frames, then a swept bounding volume is formed to bound the object at both the starting and the ending positions of the time interval.

Collision detection tools

Swept volumes

When choosing the right BV for swept BV we must take a few conditions into consideration.

- The OOBs would be most suitable for SBVs because the swept volume tends to be oriented in the direction of the objects motion, but we want to take the advantage of Sweep and Prune method, so we must choose between AABBs or k-DOPs.
- Faster update times of AABBs were outweighed by the increased size of SBVs of objects moving with swift velocities what lead to reduced ability of SaP to prune non-colliding objects.

Collision detection tools

Swept volumes theory

Consider an object i and its velocity v_i that is close to one of the axes. Let $\theta_{i,j}$ be the angle between the velocity and the axis a_j. Each time the object moves, its projection on the axis $p_{i,j}$ moves along the unit-length a_j by the amount:

$$\Delta p_{i,j} = v_{i,j} * a_j = |v_{i,j}||a_j| \cos\theta_{i,j} = |v_{i,j}| cos\theta_{i,j} \quad (1)$$

$0 \leq \theta_{i,j} \leq \pi/2$

We see that object projections move faster along axes nearly parallel to the object velocity and slower to axes almost orthogonal.

Collision detection tools

Swept volumes theory

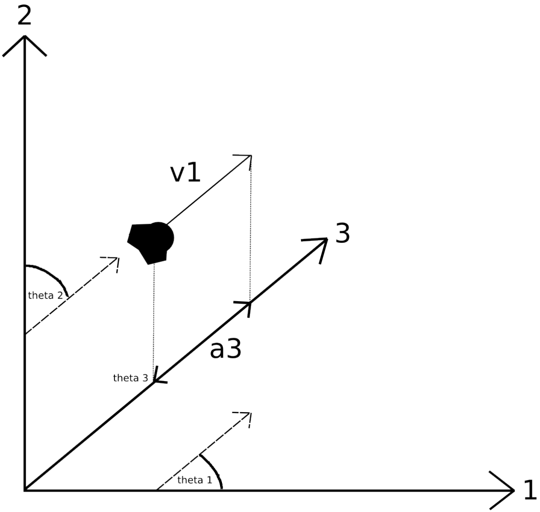
We would like to minimize $p_{i,j}$ and therefore to have lists in Sweep and Prune nearly sorted to optimize its performance. So we minimize:

$$\sum_{i=1}^{m} \cos \theta_{i,j} \tag{2}$$

The minimum for three orthogonal set of axes is equal to 1. $(\theta_1 = 0, \theta_2 = \pi/2, \theta_3 = \pi/2)$

The maximum for three orthogonal set of axes is equal to 1.73205. $(\theta_1 = \theta_2 = \theta_3 = 54.7356)$

Collision detection tools



Collision detection tools

Swept volumes theory conclusions

- Objects velocity determines the extent to which a list becomes unsorted (Equation 1).
- The length of the projected velocity vector $_{i,j}$ will decrease with increasing $\theta_{i,j}$.
- Smaller values of $_{i,j}$ imply that list items maintain a higher degree of coherence and, therefore, the list is less unsorted.

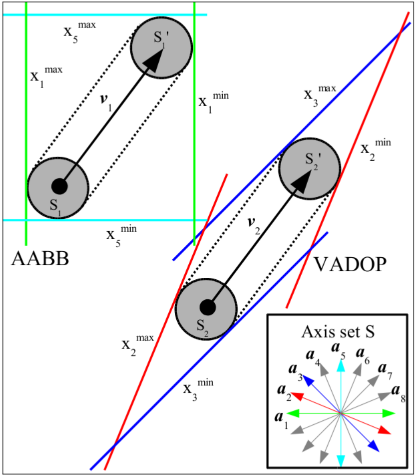
In terms of finding a separating axis to prune an unnecessary collision test, we argue that a vector that is orthogonal to the velocities of two objects is a good choice for dynamic collision detection.

VADOP

VADOP

- VADOP is a bounding volume whose description consists of number of axes, along with a pair of discrete values for each axis.
- Each pair of values bounds a discrete interval along the axis, representing the bounded objects maximum and minimum projection onto the axis.
- The axes used for an object's VADOP are selected from a common pool of axes. Specifically, the axes selected for a VADOP are the axes in the pool which are orthogonal to the corresponding object's velocity vector.
- Due to this selection of axes, the bounding volume tends to be long, narrow, and velocity- aligned.

VADOP



Choosing the right axes

- To minimize the cost for updating and sorting, an object should only be projected onto those axes orthogonal to its velocity.
- In this case, updates would be unnecessary and the lists would not need to be re-sorted.
- Unfortunately this would only allow sweep and prune to prune collisions between pairs of objects for which one of the axes of projection is orthogonal to both objects' velocities.
- It follows that this would require $O(n^2)$ axes in the worst case.

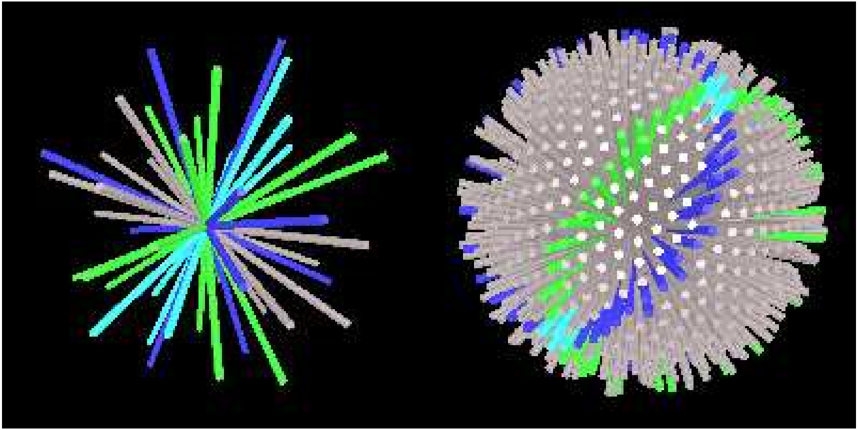
Choosing the right axes

- Instead of all $O(n^2)$ axes, we use a smaller set S of predetermined axes and project objects onto the set of axes that satisfies the following condition:
-
$$\theta_{i,j} \geq \frac{\pi}{2} - \phi. \tag{3}$$
- This allows us to prune collisions between pairs of objects for which the vector orthogonal to both objects' velocities is within ϕ of at least one axis in S.
- This also retains the property that the size of projections of moving objects along such an axis is largely unaffected by changes in magnitude of velocity when ϕ is small.

Choosing the right axes

- We consider zones (a zone is the surface area of a spherical segment) that are centered about the circumference of the sphere that is orthogonal to an axis.
- Given a set of uniformly distributed points on a unit sphere, each point corresponds to an axis of a VADOP, so the axes an object would use would be those corresponding to points on the sphere in the zone orthogonal to the object's velocity.
- It is clear that independently of the velocities of the two objects, their corresponding zones, if placed on the same unit sphere, will have some overlap, since the zones contain the largest circumference of the sphere.

Choosing the right axes



Choosing the number of the axes

- Work in the area of spherical coverings provides sets of ideal axes to use for VADOPs
- For performance reasons, k-DOPs and VADOPs require sets of axes which are uniformly distributed in a way that minimizes the maximum angle between any vector and an axis within the set.
- Previously, k-DOPs have been considered for k up to 26. Here it is useful to consider k up to 130 or even in excess of 78,000.

Choosing the number of the axes

- The optimal choice of k for a particular application depends on the velocity of objects in the scene.
- For scenes with lower velocities, using a smaller k will result in less elongated VADOPs, which is a tighter fit for a bounding volume that tries to encapsulate the movement of slow objects.
- For higher velocities, the optimal BV that encapsulates object's movement increases in size, therefore a larger k should be used.

Pruning collisions

During the broad phase, we update the positions of objects, their VADOPs, and perform sweep and prune.

- In order to quickly update object positions, we leave positions in a parametrized form, so that only a time value and a VADOP need change at each update.
- Updating VADOPs involves updating the object's projected position by its projected velocity multiplied with the elapsed time:

$$p_i = p_{i-1} + (t_i - t_{i-1}) * v_i. \quad (4)$$

Pruning collisions cont.

- The changes to the VADOPs are inserted subsequently into the lists used for sweep and prune.
- Then, sweep and prune is performed with the following modification: If two objects' VADOPs do not share a particular axis, they are presumed to have overlapping BVs.

Testing collisions

Testing collisions

- Objects that pass the broad phase are then filtered through narrow phase to determine exactly if and when they collide and where the intersection will occur.
- The results of positive collision tests will be valid until either of the tested objects changes its velocity.

Collision response

Collision response

Performing a collision response involves changing objects velocity and performing a modified version of sweep and prune on the colliding objects.

- Anytime the object changes its velocity we must invalidate collisions involving this object.
- Calculate new velocity.
- Given this velocity, we then select new axes for a VADOP and construct it by using these axes.
- Finally, we determine which axes of the object's VADOP are no longer used and remove them and the object from the SaP lists associated with that axis.

Testing environment

Testing environment

- The test environment is a simulation of a large number of spheres in a box, with random velocities and with a constant density, run at 30 frames/second.
- To ensure validity of comparison, we have set up the simulations so that the paths of objects are the same for all experiments.
- The motivation for using spheres is that a bounding sphere can be placed centered about the center of mass of an object, allowing the object to rotate freely without affecting the bounding volume.

Testing environment

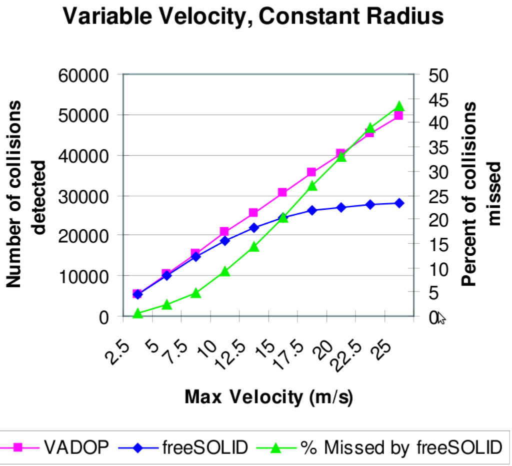
Testing environment

- VADOP have been compared with a pseudo-dynamic collision detection package called freeSOLID (an extension to I-COLLIDE).
- VADOP method yields asymptotically similiar performance to freeSOLID, within a small constant factor.

Accuracy

- The results of testing all possible pairs of objects for collisions were exactly the results that VADOP method produced.
- The VADOP method detected all of the collisions that were missed by freeSOLID.
- The number of collisions detected by both methods increases as objects move faster.
- Pseudo-dynamic methods tend to miss some of the collisions but rarely returns false positives.

Choosing the right axes

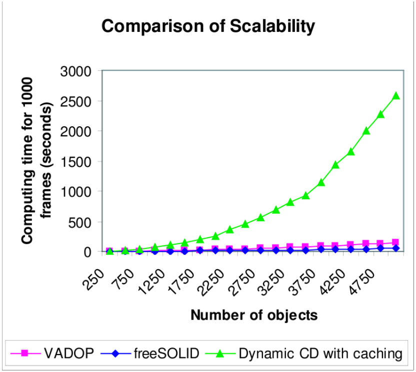


Accuracy

Accuracy

- While dynamic collision detection guarantees accuracy, this comes at increased cost of runtime.
- The factor is constant and VADOP method is asymptotically comparable with pseudo-dynamic methods.
- It is not expected for a dynamic method to be faster than a pseudo-dynamic method, because to account for motion, the bounding volumes are necessarily larger and the individual intersection tests are much more time consuming.
- VADOP method does represent significant speedup for dynamic collision detection.

Runtime performance



$$c_{total} = c_u + c_s + c_t + c_r.$$

Memmory performance

Time costs are expected to be linear, this comes with a cost in a memory performance.

- Sweep and prune method requires a n by n matrix to store (i,j) ojbect overlaps.
- In addition, SaP requires $O(n)$ space for each list.
- Thus the total memory cost of VADOP algorithm is: $O(n^2 + mn)$ (where n is the number of objects and m is the number of axes in the commom axes pool).

Expected time performance

The expected time performance is:
$O(mn + c' + c \log c + (c + d)mn)$, where:

- n is the number of objects, m is the number of axes.
- c is the number of collisions detected, c' collisions that pass the broad-phase
- d is the number the object changes its direction.
- mn - time for sorting
- $c \log c$ - collision response processing
- $(c + d)mn$ - time for changing directions

Worst case time performance

- Worst case time performance occurs only when many objects move very swiftly.
- Then the performance degrades to: $O(mn^2 + c \log c)$.
- The limiting factor of the performance is the insertion sort of $O(n^2)$ performed on m lists.

Conclusions

- Velocity-aligned discrete oriented polytopes offer faster update times.
- VADOPs are more well-adapted for dynamic collision detection.
- VADOPs are well-adapted for objects moving in higher velocities.

Drawbacks:

- VADOPs have the limitation of assuming linear motion.
- However, axes selection with use of sphere coverings is good, it is not optimal.