

АЛГОРИТМЫ ВЫЧИСЛЕНИЯ КОЛЛИЗИЙ В ВИРТУАЛЬНОМ ПРОСТРАНСТВЕ

Новиков П. А.

Кафедра информационных технологий автоматизированных систем,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: zincore36@gmail.com

В статье проводится исследование и сравнительный анализ различных видов алгоритмов вычисления коллизий в виртуальном пространстве для оптимизации производительности и достижения точности в симуляции физического взаимодействия.

ВВЕДЕНИЕ

Вычисление коллизий между объектами в виртуальном пространстве является фундаментальной задачей в области компьютерных игр, виртуальной реальности, симуляторов в области медицины, автомобилестроения и робототехники, и других приложений, где требуется создание реалистичных и интерактивных виртуальных сцен. Точное и эффективное обнаружение столкновений между объектами является ключевым аспектом, влияющим на качество и производительность таких приложений.

На данный момент существует множество различных алгоритмов, наиболее эффективных в решении определенных задач: взаимодействие сложных объектов, множества простых объектов, обработка огромных или динамических сцен, симуляции жидкостей и мягких тел, и других.

1. ОСНОВНЫЕ ГРУППЫ АЛГОРИТМОВ

1. Алгоритмы разделения осей (Separating Axis Theorem, SAT).

Алгоритмы разделения осей (SAT) являются одними из наиболее распространенных алгоритмов для вычисления коллизий. Они основаны на принципе разделения осей и позволяют эффективно определить, пересекаются ли два выпуклых объекта. Алгоритм SAT проверяет наличие коллизий, проецируя объекты на различные оси и определяя, существует ли разделительный вектор между ними [1]. Если такой вектор существует, то объекты не пересекаются.

Преимущества:

- точность;
- возможность обработки сложных геометрических форм.

Недостатки:

- вычислительная затратность при большом количестве объектов;
- большое количество проверок коллизий.

2. Алгоритмы иерархий ограничивающих объемов (Bounding Volume Hierarchies, BVH).

Алгоритмы иерархий ограничивающих объемов (BVH) используются для оптимизации вычислений коллизий. BVH организует объекты в древовидную структуру, где каждый узел пред-

ставляет собой ограничивающий объем, такой как ограничивающий параллелепипед или сфера. Это позволяет эффективно отсеивать пары объектов, которые не могут столкнуться, уменьшая количество необходимых проверок коллизий.

В зависимости от типа деления ограничивающих объемов, существуют алгоритмы BVH с делением по середине (Midpoint Split), по площади (Surface Area Heuristic), по средней точке (Mean Split) и по объему (Volume Split).

Преимущества:

- высокая производительность;
- работа с большим количеством объектов;
- быстрое обнаружение простых коллизий.

Недостатки:

- низкая точность по сравнению с SAT;
- выделение дополнительных ресурсов для построения и обновления иерархии.

3. Алгоритмы сеток разбиения пространства (Spatial Partitioning Grid Algorithms).

Алгоритмы сеток разбиения пространства используются для упрощения вычислений коллизий путем разбиения виртуального пространства на сетку ячеек. Каждый объект помещается в соответствующую ячейку сетки, и затем проверяются только объекты, находящиеся в той же ячейке или смежных ячейках. Это позволяет сократить количество проверок коллизий, так как большинство объектов находятся в далеких ячейках и не могут пересекаться.

Преимущества:

- простота и эффективность для больших сцен с разреженными объектами;
- быстрое обнаружение коллизий между рядом находящимися объектами.

Недостатки:

- низкая точность при наличии сложных геометрических форм;
- дополнительные усилия для обработки объектов на пересечении ячеек сетки.

4. Геометрические алгоритмы.

Геометрические алгоритмы основаны на анализе геометрических характеристик объектов, таких как точки, линии и поверхности, для определения коллизий. Эти алгоритмы обычно требуют более сложных вычислений, но могут обеспечить

высокую точность при обнаружении и решении коллизий.

Преимущества:

- обработка сложных форм;
- более точные вычисления коллизий.

Недостатки:

- наиболее вычислительно затратные;
- требуют наибольшего количества ресурсов среди всех алгоритмов.

II. МЕТОДЫ СОКРАЩЕНИЯ ПРОВЕРКИ КОЛЛИЗИЙ (COLLISION CULLING TECHNIQUES)

Методы сокращения проверок коллизий используются для исключения ненужных проверок между парами объектов, которые могут быть гарантированно отсеяны без учета всех деталей геометрии. Некоторые из методов сокращения проверок коллизий включают:

1. Ограничивающие объемы (Bounding Volumes). Вместо проверки коллизий на основе полной геометрии объектов, используются ограничивающие объемы, такие как ограничивающие параллелепипеды или сферы, чтобы быстро определить, пересекаются ли объекты на основе их объемов. Виды ограничивающих объемов:

- Axis Aligned Bounding Boxes (AABB) – прямоугольный параллелепипед, который охватывает объект целиком, и определяется минимальным и максимальным значениями по каждой оси [1]. Применяет алгоритм Sweep and prune для нахождения столкновений [2].
- Bounding Spheres – сферические объемы, которые окружают объекты.
- Oriented Bounding Boxes (OBB) – ограничивающие параллелепипеды, имеющие произвольное направление и ориентацию в пространстве.

2. Временная согласованность (Time Coherence). Если движение объектов непрерывно, то можно использовать информацию о предыдущем состоянии объектов для прогнозирования и отсеивания некоторых пар объектов без дополнительной проверки коллизий [3]. Виды алгоритмов временной согласованности:

- Сохранение и восстановление (Save and Restore) – сохраняет промежуточные результаты вычислений на одном кадре и восстанавливает их на следующем кадре.
- Прогнозирование (Prediction) – предсказывает будущее состояние объектов на основе их предыдущего состояния и параметров движения.
- Кэширование (Caching) – кэширует результаты вычислений для объектов на предыдущих кадрах и повторно использует их на следующих кадрах.

- Упорядочение обновления (Update Ordering) – оптимизирует порядок обновления объектов или элементов сцены на последующих кадрах или временных шагах, на основе изменений и зависимостей между ними.

3. Пространственная разбивка (Spatial Partitioning). Использование методов разбиения пространства, таких как сетки разбиения пространства или иерархии ограничивающих объектов, позволяет эффективно отсеивать пары объектов, находящиеся далеко друг от друга [4]. Виды разбивки:

- Дерево октантов (Octree) – рекурсивно разделяет пространство на восемь равных кубических ячеек - октант.
- Сетка (Grid). Каждая ячейка сетки содержит информацию о объектах, находящихся внутри нее, что позволяет быстро определить, с какими объектами взаимодействует данная сущность.
- KD-дерево (KD-Tree) – разделяет пространство на оси-выделители (splitter axes) путем альтернативного разделения объектов по координатам X, Y и Z.
- R-дерево (R-Tree) – иерархическое дерево, которое разделяет пространство на ограничивающие прямоугольники. Каждый узел дерева содержит информацию о прямоугольниках и ссылки на поддеревья.

III. ЗАКЛЮЧЕНИЕ

Были рассмотрены основные алгоритмы вычисления коллизий объектов в виртуальном пространстве. Каждый из этих алгоритмов имеет свои преимущества и ограничения, и выбор конкретного алгоритма зависит от требований приложения и характеристик сцены.

Алгоритмы разделения осей (SAT) обеспечивают точность, но могут быть вычислительно затратными. Алгоритмы иерархий ограничивающих объемов (BVH) обеспечивают высокую производительность, но могут потерять в точности. Алгоритмы сеток разбиения пространства обеспечивают эффективность для разреженных сцен, но могут быть менее точными. Геометрические алгоритмы обеспечивают высокую точность, но требуют большего вычислительного ресурса.

СПИСОК ЛИТЕРАТУРЫ

1. Real-time Collision Detection. / Ericson, C. // Elsevier. – 2005. – № 1. – p. 632.
2. Collision detection between geometric models: a survey. / Ming C. Lin, Gottschalk S. // Proc. of IMA conference on mathematics of surfaces. – 1998. – Vol. 1. – pp. 602–608.
3. Physics for Game Developers. / Bourg M.D., Bywalec B. // O'Reilly Media. – 2013. – № 2. – p. 575.
4. Collision Detection in Interactive 3D Environments / Gino van den Bergen // CRC Press. – 2003. – № 1. – p. 277.