

УДК 621.865:004.021

Алгоритм определения пересечения произвольных полигональных моделей для нахождения коллектора контакта, нормалей и глубины проникновения при планировании и моделировании захвата деформируемого объекта

© Феоктистова Елена Владимировна, Лесков Алексей Григорьевич

Россия, г. Москва, Московский государственный технический университет имени Н.Э. Баумана

agleskov@rambler.ru, feoktistovaev@mail.ru

Рассмотрен алгоритм определения пересечений полигональных моделей, методы фаз которого выбраны на основании анализа и сравнения существующих алгоритмов определения пересечений. Методом широкой фазы является модифицированный алгоритм ограничивающих параллелепипедов типа ОБВ. Алгоритм узкой фазы, решающий задачу анализа взаимоположения множества вершин полигональной модели одного тела, множества проекций этих вершин на поверхность другого тела вдоль вектора мгновенной скорости первого тела относительно второго и направления вектора мгновенной скорости, сводится к задаче поиска ближайших соседей между вершинами полигональных сеток взаимодействующих объектов.

Ключевые слова: алгоритм определения пересечения полигональных моделей, пересечение многогранников, ограничивающий параллелепипед типа ОБВ.

При планировании и моделировании захвата деформируемых объектов возникает необходимость разрешения задачи обнаружения пересечений, что по сути является вычислительной проблемой обнаружения геометрического пересечения между собой звеньев захватного устройства (ЗУ) и объекта манипулирования (ОМ). Причем необходимо не просто указать сам факт пересечения, но и определить конкретные точки, множество которых называется коллектором контакта, нормали контактов, а также глубину проникновения для вычисления качественных параметров захвата.

Сложность в детектировании пересечений объектов заключается в необходимости постоянного расчета расстояния между поверхностями объектов на каждом шаге моделирования. С увеличением точности расчетов повышаются и требования к вычислительной мощности.

В связи с характером моделируемого процесса — захвата объекта — считают, что в начальный момент времени контакт между ЗУ и ОМ отсутствует. В процессе смыкания пальцев захватного устройства расстояние между элементами схвата и объектом уменьшается, а затем происходит пересечение поверхностей ЗУ и ОМ — непосредственно захват объекта. Исходя из этого, компромисс между приемлемой сложностью вычислений и достаточной точностью заключается в следующем подходе: на начальных шагах моделирования использовать менее точные методы обнаружения пересечения, а при уменьшении расстояния между фалангами ЗУ и ОМ до значения ошибки неточного метода, умноженного на некоторый коэффициент запаса,

использовать более точные методы до детектирования пересечения и получения искомым характеристикам контакта, таких как коллектор контакта, нормали и глубина проникновения. Описанные этапы получили названия широкой (быстрая и грубая оценка) и узкой (более медленная и точная) фаз обнаружения пересечений [1].

Наиболее часто используемыми алгоритмами широкой фазы обнаружения пересечений являются метод разделяющей плоскости [2] и методы ограничивающих объемов (методы ограничивающих сферы, параллелепипеда типа AABV и параллелепипеда типа OBB) [3]. Метод разделяющей плоскости имеет ряд недостатков, что ограничивает его применение. Ограничивающая сфера — самый простой и быстрый способ определения наличия или отсутствия пересечения между объектами, но он не применим из-за грубости оценки. Обычно параллелепипед точнее, чем сфера, повторяет форму объектов. Решение о выборе между параллелепипедами типов AABV и OBB зависит от условий конкретной задачи: если проверка на каждом шаге моделирования осуществляется между одними и теми же объектами, то выбирают более точный метод — параллелепипеды типа OBB, а если между разными объектами — то более экономичный — параллелепипеды типа AABV.

Исходя из этого, в качестве алгоритма широкой фазы обнаружения пересечений выбран метод ограничивающего параллелепипеда типа OBB с выделением областей потенциального пересечения и определения вершин полигональной модели, которые попадают в эту область и будут использоваться в узкой фазе определения пересечения (рис. 1), что приводит к уменьшению времени моделирования.

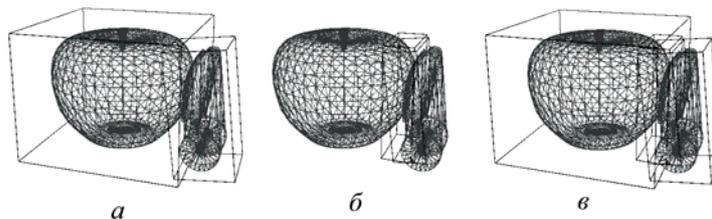


Рис. 1. Результат широкой фазы определения пересечения полигональных моделей:

a — определение пересечения OBB; *б* — выделение областей потенциального пересечения; *в* — соотношение OBB и областей потенциального пересечения

В качестве алгоритмов узкой фазы обнаружения пересечения полигональных моделей чаще всего используются базирующиеся на методе Gilbert-Johnson-Keerthi (GJK) алгоритмы [4], которые определяют пересечение выпуклых тел путем анализа их разницы Минковского, и алгоритмы, опирающиеся на взаимное положение областей Вороного фрагментов полигональных моделей — алгоритм Lin-Canny [5] и алгоритм V-Clip [6] — улучшенный в некоторых аспектах алгоритм Лин-Кенни.

В [6] приведено сравнение быстродействий алгоритмов GJK, Lin-Canny и V-Clip, согласно которому вычислительная скорость определения пересечения между объектами последнего значительно выше скорости вычислений других алгоритмов. Алгоритм V-Clip используется во многих физических движках — компьютерных программах, производящих моделирование физических законов реального мира в виртуальной среде с той или иной степенью аппроксимации, в частности в Bullet Physics Library и в Open Dynamics Engine, которые занимают лидирующие по популярности позиции среди свободных физических движков. Но, несмотря на явные до-

стоинства, этот метод сложен в реализации, требует предварительного расчета областей Вороного, и результат его работы требует дополнительной проверки и фильтрации (рис. 2). Кроме того, существенным недостатком рассматриваемых алгоритмов является их применимость только к выпуклым телам.



Рис. 2. Моделирование пересечения полигональных моделей звена ЗУ и ОМ с использованием Bullet Physics Library:
 а — определение ложной области контакта вне поверхности модели; б — определение неполного коллектора контакта

Исходя из этого, задачу узкой фазы определения пересечения предлагается формулировать не в ее классической интерпретации как задачу определения взаимопenetрации многоугольников, а как задачу анализа взаимоположения множества вершин полигональной модели $P = p_1, \dots, p_n$ тела A , множества проекций этих вершин $P' = p'_1, \dots, p'_n$ на поверхность другого тела B вдоль вектора мгновенной скорости v первого тела относительно второго и направления вектора v (рис. 3). При этом, если вектор $p_i p'_i$ сонаправлен с вектором v , то вершина p_i тела A находится в контакте с телом B , глубина проникновения для нее равна длине вектора $p_i p'_i$, а нормаль контакта противоположно направлена по отношению к нормали полигона, которому принадлежит проекция p'_i .

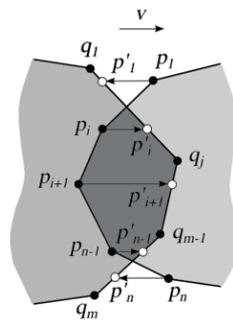


Рис. 3. Отыскание проекций вершин $P' = p'_1, \dots, p'_n$ полигональной модели тела A на поверхность полигональной модели тела B вдоль вектора мгновенной скорости v первого тела относительно второго

Полигон, на котором будет располагаться проекция p'_i , в качестве одной из своей вершин будет иметь вершину q_j , которая будет являться самой близкой среди множества вершин полигональной модели $Q = q_1, \dots, q_m$ тела B к вершине p_i тела A . То есть, найдя пару ближайших вершин p_i и q_j , можно избежать перебора всех полигонов тела B в поисках проекции p'_i , что сократит время моделирования, и осуществлять поиск только среди полигонов, в которых q_j является одной из вершин. Таким образом, необходимо решить задачу «поиска ближайшего соседа», которая заключается в отыскании среди множества элементов, расположенных в метрическом пространстве, элементов близких к заданному. Полигональная модель в этом случае может рассматриваться как замкнутое метрическое пространство.

В качестве алгоритмов «поиска ближайших соседей» были рассмотрены линейный поиск [7], метод редких точек [8] и алгоритмы разбиения пространства — диаграмма Вороного [7], поиск в kD-деревьях [9], поиск в BSP-деревьях [10], поиск в VP-деревьях [11]. Учитывая их скорости, достоинства и недостатки, был выбран последний. Поиск при этом осуществляется среди вершин полигональной модели, попавших в область потенциального взаимодействия.

Определение «ближайшего соседа» с помощью поиска по VP-дереву в узкой фазе рассматриваемого алгоритма иллюстрирует рис. 4. Результат работы описанного алгоритма представлен на рис. 5.

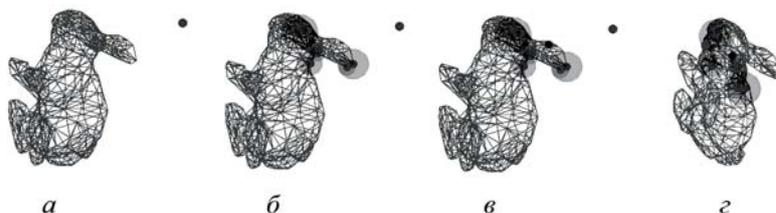


Рис. 4. Результат поиска «ближайшего соседа» для заданной точки среди вершин невыпуклой замкнутой полигональной метки:
a — задание начальных данных; *b* — построение бинарного VP-дерева; *v*, *z* — определение «ближайшего соседа»

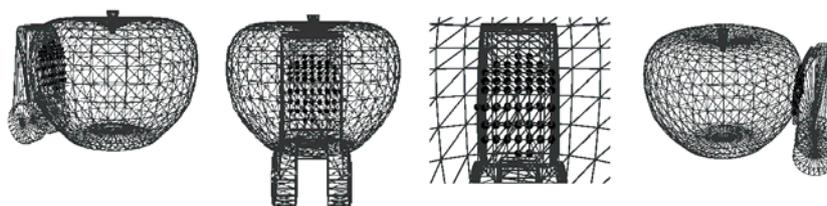


Рис. 5. Результат работы алгоритма: полностью определенный коллектор контакта, точки которого располагаются на поверхности модели

Таким образом, описанный алгоритм определяет с более высокой точностью по сравнению с существующими методами коллектор контакта и параметры, характеризующие его отдельные точки. Эти данные могут быть использованы для определения качества захвата и моделирования деформации ОМ в процессе захвата.

Литература

- [1] Christer E. *Real-time Collision Detection*. Elsevier, 2005, 633 p.
- [2] *Game Developer's Links Collection. Collision Detection FAQ*. URL: <http://gdlinks.hut.ru/cdfaq/> (дата обращения 18.12.2014).
- [3] Плеханов Н.С. *И вот они столкнулись*. URL: <http://www.vr-online.ru/content/i-vot-oni-stolknulis%E2%80%A6-634> (дата обращения 24.12.2014).
- [4] *Реализация алгоритма Гилберта — Джонса — Керми*. URL: <http://100byte.ru/100btwrks/gjk/gjk.html> (дата обращения 20.12.2014).
- [5] Lin M.C. *Efficient Collision Detection for Animation and Robotics*. University of California, 1993. URL: <http://dl.acm.org/citation.cfm?id=920962> (дата обращения 19.12.2014).
- [6] Mirtich B. *V-Clip: Fast and Robust Polyhedral Collision Detection*. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.3275> (дата обращения 18.12.2014).
- [7] Препарата Ф. *Вычислительная геометрия: Введение*. Москва, Мир, 1989, 478 с.
- [8] Tao Y., Sheng C. Fast Nearest Neighbor Search with Keywords. URL: <http://www.cse.cuhk.edu.hk/~taoyf/paper/tkde13-spakey.pdf> (дата обращения 23.09.2013).
- [9] Panigrahy R. *An Improved Algorithm Finding Nearest Neighbor Using Kd-trees*. Microsoft Research, Mountain View CA, USA, 2008. URL: <http://theory.stanford.edu/~rinap/papers/kdtreelatin.pdf> (дата обращения 23.09.2013).
- [10] Maneewongvatana S., Mount D.M. *An Empirical Study of a New Approach to Nearest Neighbor Searching*. Department of Computer Science, University of Maryland, Maryland. URL: <http://www.cs.umd.edu/~mount/Papers/alnex01-empir.pdf> (дата обращения 23.09.2013).
- [11] Yianilos P.N. *Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces*. URL: <http://www.cs.iastate.edu/~hnavar/nndatastructures.pdf> (дата обращения 23.09.2013).