УДК 629-78

Д.А. Храмов

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ РАЗВЕРТЫВАНИЯ ВЕСОМОЙ НИТИ С ПОМОЩЬЮ ФИЗИЧЕСКИХ ДВИЖКОВ

Анотація. Фізичні движки — бібліотеки комп'ютерного моделювання руху систем тіл, що дозволяють додавати і видаляти тіла безпосередньо в ході розрахунків — являють собою зручний інструмент для дослідження динаміки систем, структура яких змінюється під час руху. У роботі запропоновано алгоритм розгортання вагомой нитки, складеної з шарнірно зв'язаних твердих тіл, що додаються по мірі розгортання. Моделювання виконано за допомогою движків Box2D і Bullet. Показано можливості верифікації результатів розрахунків.

Одной из тенденций современной космонавтики является использование на орбите протяженных систем. Так, в 2007 г. со спутника YES2 был развернут 30-километровый трос с закрепленной на его конце спускаемой капсулой. Подобные космические тросовые системы предполагается использовать для создания интерферометров с большой базой, солнечных электростанций, а также систем увода отработавших космических аппаратов [1, 2]. В связи с этим актуальна задача исследования динамики развертывания таких систем [3, 4].

Учет влияния космической среды на движение протяженного тела сам по себе является достаточно сложной проблемой. Поэтому на начальном этапе исследований целесообразно упростить исходную постановку задачи, рассмотрев поведение системы в более простых внешних условиях, а именно — исследовать развертывание весомой нити с закрепленным концом, находящейся в поле постоянной силы тяжести.

Трос можно представить системой связанных друг с другом точечных масс или твердых тел. Моделирование движения таких систем с помощью современных пакетов физического моделирования (например, SimMechanics или MapleSim) выполняется достаточно просто. Сложности возникают при решении задач, в которых структура системы изменяется в процессе движения. Добавление тел по мере

развертывания требует остановки расчета и внесения коррективов в схему системы.

Перспективным средством решения подобных задач представляются физические движки (physics engines) — библиотеки компьютерного моделирования движения систем частиц и тел [5], позволяющие добавлять и удалять тела непосредственно в ходе расчета. Появившись в связи с развитием индустрии компьютерных игр и 3D-моделирования, эти библиотеки затем нашли широкое применение в системах виртуальной реальности [6] и робототехнике. Обзор моделей, численных методов и алгоритмов, используемых в физических движках, содержится в [7]. Наиболее полный список подобных библиотек приведен на сайте [8].

Целью работы является обзор основных компонентов физических движков и создание на их основе модели развертывания весомой нити.

Принципы моделирования с помощью физических движков

В отличие от математических библиотек, реализующих те или иные численные методы, физический движок как библиотека дает возможность пользователю создавать такие объекты как «частица», «тело», «связь» и управлять их характеристиками.

Моделирование с помощью физического движка начинается с создания виртуального пространства — «мира», в который помещаются частицы и тела. Тела могут быть твердыми или «мягкими» (soft body), с заданными характеристиками движения и массой. Кроме того, каждому телу присваивается определенная форма (collision shape), которая потом используется при анализе столкновений.

Затем задаются действующие на тела силы и импульсы, добавляются связи между телами и выполняется расчет движения. Основными задачами физического движка на этом этапе являются обнаружение столкновений между телами и расчет на основе этой информации состояния тел в следующий момент времени. Полученные данные о состоянии тел передаются средству отображения (рендереру).

По окончании расчетов выполняется «уборка мусора»: удаляются созданные связи, тела и др. объекты.

Рассмотрим подробнее шаг расчета по времени. Прежде всего выполняется обнаружение столкновений тел (collision detection). Оно состоит из широкой и узкой фаз (broadphase и narrowphase). Во время

широкой фазы алгоритм обнаружения столкновений находит пары потенциально взаимодействующих объектов. При этом используется упрощенная геометрия системы, например, тела заменяются выровненными по осям ограничивающими параллелепипедами (axis aligned bounding box). Далее наступает черед узкой фазы, во время которой движок должен определить реальные точки контакта тел, нормали и глубины проникновения. Узкая фаза обычно значительно «дороже» в вычислительном плане, чем широкая, поэтому от эффективности выполнения широкой фазы во многом зависит производительность движка при большом количестве тел.

Учетом обнаруженных столкновений (resolve collisions) занимается подсистема движка, называемая «решателем» (solver). В задачу решателя входит вычисление скоростей тел после столкновения, и коррекция их положений во избежание проникновения друг в друга.

На завершающем этапе происходит вычисление новых координат и скоростей тел. Для этого используются традиционные методы численного решения обыкновенных дифференциальных уравнений, в частности, методы Эйлера-Кромера, Верле и Рунге-Кутты.

Более подробно внутреннее устройство и принципы работы физических движков описаны в [9–11].

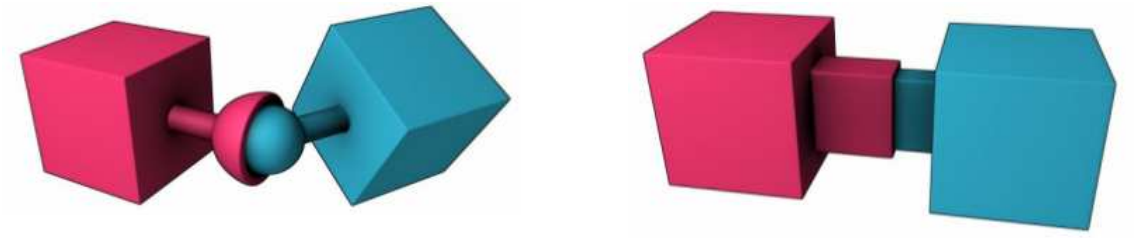
Схема нити

Пусть нить состоит из шарнирно связанных твердых тел. В физических движках используются следующие три вида твердых тел:

- динамические тела (dynamic rigidbodies) — обычные твердые тела, способные двигаться и сталкиваться с другими телами;
- статические тела (static rigidbodies) — неподвижные тела, с которыми могут сталкиваться другие тела;
- кинематические тела (kinematic rigidbodies) — подвижные тела, движение которых задается пользователем и столкновение с другими телами не может его изменить.

В нашем случае понадобятся динамические тела, составляющие нить, и статическое тело, к которому нить будет крепиться.

Для соединения тел понадобится сферический шарнир (revolute joint) (рис. 1а), допускающий одну (в плоском случае) или три (в пространственном) вращательные степени свободы, а также призматическое соединение (prismatic или sliding joint), имеющее одну поступательную степень свободы (рис. 1б).



а) б)

Рисунок 1 — Соединения между телами и их условные обозначения: а) сферический шарнир; б) призматическое соединение [12]

Кроме того, в соединениях можно задать моторы (joint motor), то есть силы, регулирующие взаимное движение соединенных тел, а также ограничители (joint limits) взаимного перемещения/вращения тел. Для моделирования развертывания необходимо задать мотор в призматическом соединении, позволяющий раздвигать тела с заданной скоростью.

Схема развернутой нити представлена на рис. 2.

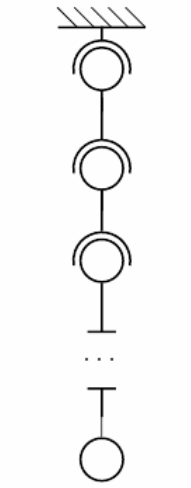


Рисунок 2 — Схема развернутой нити

Алгоритм развертывания

Алгоритм развертывания нити состоит из двух основных этапов: инициализации (рис. 3) и шага расчета.

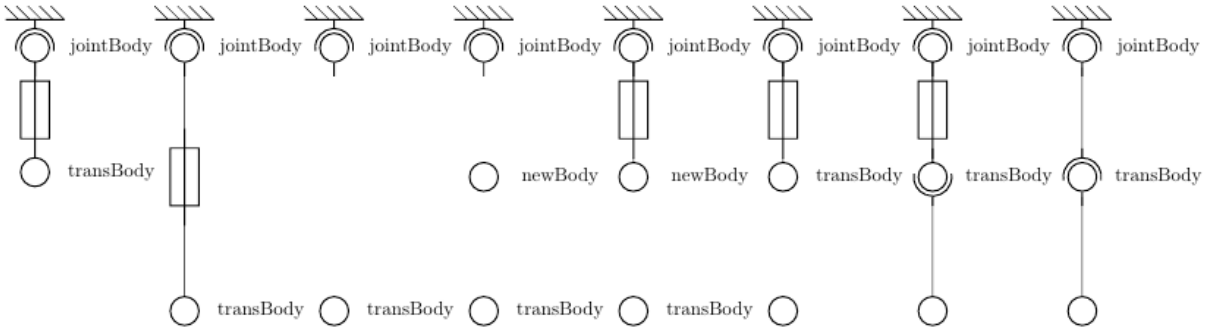
```
Создаем «мир» и задаем гравитацию
Создаем статическое тело ground
Создаем динамическое тело jointBody и помещаем его в точку крепления нити
Соединяем ground и jointBody сферическим шарниром
Создаем динамическое тело transBody
Соединяем jointBody и transBody призматическим шарниром, задаем мотор
```

Рисунок 3 — Алгоритм развертывания: инициализация (псевдокод)

 ISSN 1562-9945

По окончании инициализации система приобретает вид, показанный на рис. 4а. Нить состоит из единственного звена, включающего в себя два тела (jointBody и transBody), соединенных призматическим шарниром.

В ходе развертывания эти тела раздвигаются с заданной скоростью. Когда расстояние между ними превысит некоторую фиксированную длину length (рис. 4б), звено трансформируется: удаляется связывающий тела шарнир (рис. 4в) и создается новое тело newBody (рис. 4г). Если нить еще не развернута на полную длину, то newBody соединяется с jointBody призматическим шарниром (рис. 4д) и процесс развертывания продолжается уже в звене jointBody–newBody. Теперь newBody играет роль transBody (рис. 4е). Если же нить развернута, то newBody и jointBody соединяются сферическим шарниром и система приобретет вид, показанный на рис. 4з.



а) б) в) г) д) е) ж) з)

Рисунок 4 — Схема развертывания нити

Шаг расчета алгоритма развертывания нити представлен на рис. 5.

```
Вычисляем длину звена
Если длина звена больше length
    удаляем призматический шарнир
    создаем динамическое тело newBody
    Если нить не развернута полностью
        соединяем newBody и jointBody призматическим шарниром
    иначе
        соединяем newBody и jointBody сферическим шарниром
    соединяем newBody и transBody сферическим шарниром
    transBody = newBody
```

Рисунок 5 — Алгоритм развертывания: шаг расчета (псевдокод)

Моделирование развертывания

Рассмотрим реализацию алгоритма развертывания на примере двух физических движков — Box2D [13] и Bullet [14]. Программный код реализации находится на сайте [15].

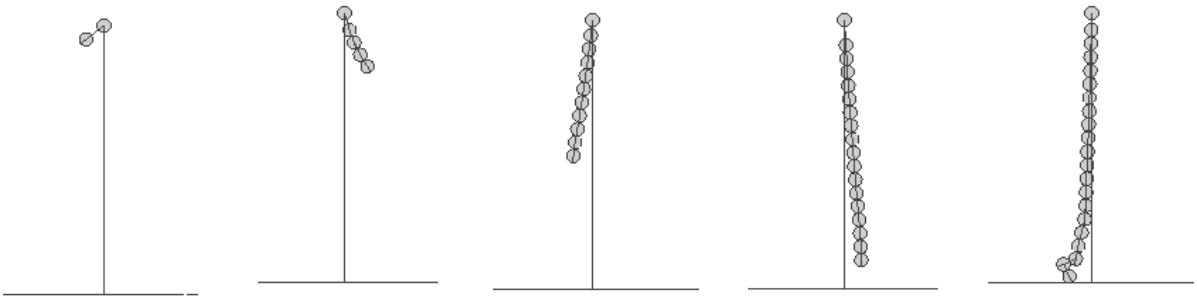
Box2

Box2D — свободный двумерный физический движок, разработанный Эрином Катто (Erin Catto). Простой в освоении, Box2D широко используется в компьютерных играх (в частности, в Angry Birds) и в учебных приложениях.

Box2D предназначен для моделирования динамики систем твердых тел. Тела могут быть связаны шарнирами различных видов (всего их более десятка), снабженными моторами и ограничителями, подвергаться действию разных сил, в частности, сил гравитации, трения и удара. Движок может моделировать столкновения тел, составленных из выпуклых многоугольников, окружностей и линий. Работа с Box2D подробно описана в [16].

Box2D написан на C++ и может работать на любой платформе, на которой присутствует компилятор C++. Кроме того, он портирован на многие языки программирования и программные среды, включая Java, C#, Adobe Flash, JavaScript и Delphi.

Результаты моделирования развертывания при помощи Box2D представлены на рис. 6.



а) б) в) г) д)

Рисунок 6 — Кадры анимации развертывания нити, смоделированного при помощи Box2D

Bullet

Bullet — свободный трехмерный физический движок реального времени, разработанный Эрвином Кумансом (Erwin Coumans). Помимо твердых тел, движок позволяет моделировать движение мягких

 ISSN 1562-9945

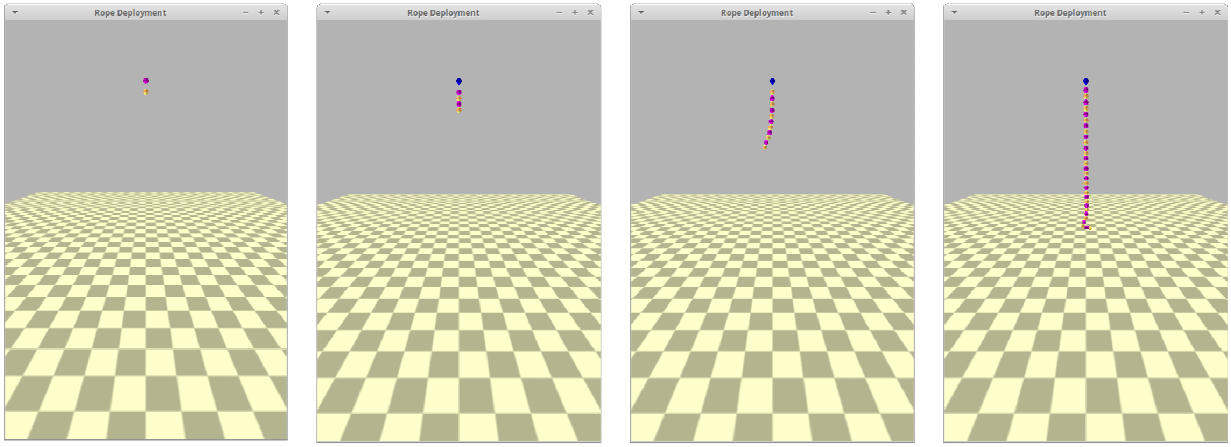
тел, реализует методы гидродинамики сглаженных частиц (smoothed particle hydrodynamics), поддерживает технологии CUDA и OpenCL.

Bullet активно используется в компьютерных играх (GTA 4, 5), фильмах (Shrek 4), программах 3D-моделирования (Blender).

На рис. 7 представлены результаты моделирования развертывания нити с помощью Bullet.

Возможности верификации результатов

Рассмотренные выше библиотеки являются свободными и распространяются с открытым исходным кодом, что позволяет специалисту самостоятельно оценить то, как реализован тот или иной компонент библиотеки и, при необходимости, исправить ошибки или добавить новые компоненты.



а) б) в) г)

Рисунок 7 — Кадры анимации развертывания нити, смоделированного при помощи Bullet

Кроме того, можно сравнить между собой результаты, полученные с помощью разных движков. Для этого существуют формат COLLADA Physics и API Physics Abstraction Layer (PAL).

COLLADA [17] — это стандартный формат обмена данными между 3D-приложениями, основанный на XML. Стандарт COLLADA состоит из нескольких частей, одна из которых — COLLADA Physics — отвечает за передачу данных о физических объектах. Файл COLLADA представляет собой мгновенный «снимок» виртуального пространства, содержащий данные обо всех находящихся в нем телах и связях между ними. Стандарт поддерживается наиболее известными физическими движками, в частности, Bullet. Это дает возмож-

ность, импортируя файл COLLADA, выполнить аналогичную симуляцию на другом движке.

Иной подход к обмену данными предлагается в PAL [18]. PAL представляет собой интерфейс к абстрактному физическому движку, с помощью которого разработчик может использовать одновременно несколько движков или менять их между собой, в том числе непосредственно в ходе расчетов.

Выводы

Использование физических движков позволяет разработчику расчетных программ оперировать объектами более близкими к предметной области — динамике систем твердых тел — по сравнению с подходом, основанным на использовании языков программирования общего назначения совместно с математическими библиотеками.

Физические движки позволяют без значительных усилий разработать программу развертывания нити и получать результаты расчетов в реальном времени. Следует отметить, что для большинства физических движков скорость выполнения расчетов важнее их точности. Поэтому, в частности, во многих движках используются методы численного интегрирования не выше 2-го порядка. Однако, для свободных движков этот недостаток легко исправить подключением дополнительных математических библиотек.

Отметим, что ряд физических движков позволяет рассчитывать динамику деформируемых (мягких) тел. Однако методы, на которых основываются эти расчеты, весьма скудно описаны в руководствах и требуют дальнейшего изучения. Поэтому в работе предложен алгоритм моделирования развертывания нити, составленной из шарнирно связанных твердых тел.

ЛИТЕРАТУРА

1. Cartmell M. P. A review of space tether research / M. P. Cartmell, D. J. McKenzie // Progress in Aerospace Sciences. — 2008. — V. 44, N 1. — P. 1–21.
2. Волошенюк О. Л. Космические тросовые системы — перспективное направление космической техники и технологии / О. Л. Волошенюк, А. В. Пироженко, Д. А. Храмов // Космічна наука і технологія. — 2011. — Т. 17, № 2. — С. 32–44.
3. Dynamics of variable-length tethers with application to tethered satellite deployment / J. L. Tang, G. X. Ren, W. D. Zhu, H. Ren // Communications

in Nonlinear Science and Numerical Simulation. — 2011. — V. 16, N 8. — P. 3411–3424.

4. Dynamical simulation of tether in orbit deployment / N. N. Smirnov, Yu. A. Demyanov, A. V. Zvyaguin, A. A. Malashin, A. A. Luzhin // Acta Astronautica. — 2010. — V. 67, N 3--4. — P. 324–332.
5. Jones M. T. Open source physics engines. Building believable worlds with open source [Электронный ресурс]. — Режим доступа : http://www.ibm.com/developerworks/opensource/library/os-physicsengines/os-physicsengines-pdf.pdf
6. An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations / J. Hummel, R. Wolff, T. Stein, A. Gerndt, T. Kuhlen // Advances in Visual Computing. 8th International Symposium, ISVC 2012, Rethymnon, Crete, Greece, July 16–18, 2012, Revised Selected Papers, Part II. — 2012. — P. 346–357.
7. Interactive simulation of rigid body dynamics in computer graphics / J. Bender, K. Erleben, J. Trinkle, E. Coumans // EUROGRAPHICS 2012 State of the Art Reports. — Cagliari, Sardinia, Italy : Eurographics Association, 2012.
8. Physics Engines List [Электронный ресурс]. — Режим доступа : http://www.digitalrune.com/Support/Blog/tabid/719/EntryId/30/Physics-Engines-List.aspx
9. Millington I. Game Physics Engine Development / I. Millington. — New York : Morgan Kaufmann Publishers, 2010. — 553 p.
10. Bourg D. M. Physics for Game Developers / D. M. Bourg, B. Bywalec. — Cambridge : O'Reilly, 2013. — 551 p.
11. Eberly D. H. 3D Game Engine Architecture / D. H. Eberly. — New York : Morgan Kaufmann Publishers, 2005. — 736 p.
12. Bullet 2.82 Physics SDK Manual [Электронный ресурс].- Режим доступа: http://bullet.googlecode.com/svn/trunk/Bullet_User_Manual.pdf
13. Box2D [Электронный ресурс]. — Режим доступа : http://www.box2d.org/
14. Bullet Physics Library [Электронный ресурс]. — Режим доступа: http://bulletphysics.org/
15. Компьютерное моделирование развертывания весомой нити с помощью физических движков [Электронный ресурс]. — Режим доступа: http://dkhramov.dp.ua/index.php?n=Sci.SciSoft
16. Box2D C++ tutorials [Электронный ресурс]. — Режим доступа: http://www.iforce2d.net/b2dtut/
17. COLLADA [Электронный ресурс]. — Режим доступа: http://collada.org/
18. PAL [Электронный ресурс]. - Режим доступа: http://www.adrianboeing.com/pal/index.html