

Список литературы

1. Exact and Guaranteed Accuracy Solutions of Linear Programming Problems. by Distributed Computer Systems with MPI / *A. V. Panyukov, V. V. Gorbik* // Tambov University REPORTS: A Theoretical and Applied Scientific Journal. Series: Natural and Technical Sciences. Vol. 15, Issue 4, 2010. – P. 1392–1404. – URL: <http://vestnik.tsutmb.ru/old/index.php?module=subjects\&func=viewpage\&pageid=165>

2. Библиотека классов «Exact Computational» / Свидетельство о государственной регистрации программы для ЭВМ № 2009612777 от 29 мая 2009 г. / *А.В. Панюков, М.И. Германенко, В.В. Горбик* // Программы для ЭВМ, базы данных, топологии интегральных микросхем. Офиц. бюл. Российского агентства по патентам и товарным знакам № 3. – 2009. – С. 251.

Е.С. Городецкий

Нижегородский государственный университет им. Н.И. Лобачевского

ОРГАНИЗАЦИЯ МНОГОПОТОЧНЫХ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ В ЗАДАЧЕ РАСЧЁТА КИНЕМАТИКИ РЫЧАЖНЫХ МЕХАНИЗМОВ

Пространственные рычажные механизмы являются важной составляющей современной техники и производственных технологий [8]. Механические системы разрабатываются для выполнения достаточно серьёзных задач, требующих высокой надёжности и точности. При конструировании механизма обычно требуется многократно выполнять полный расчет его движения для постепенной оптимизации модели. Наибольший интерес для конструктора представляют технологии и инструменты для оптимального моделирования механических систем. Тем не менее такие возможности практически отсутствуют в современных общедоступных системах автоматизированного проектирования. Задача параметрической оптимизации модели рычажного

механизма была формализована и описана автором [6]. Дальнейшая работа ведётся в направлении автоматизации решения этой задачи с распараллеливанием векторных вычислений на графических картах.

Постановка задачи. Для эффективного решения задачи оптимизации кинематики механизма требуется применение максимально быстрых алгоритмов вычисления положений, реализованных под современные многоядерные архитектуры. Схема параллельных вычислений и структур данных для расчета положений пространственного механизма была разработана автором в общем виде для CPU и GPU [5]. В данной работе описывается организация многопоточных вычислений на графическом процессоре с использованием технологии DirectX 11 Compute Shader [2], используемые структуры данных и результаты вычислительного эксперимента.

Введение в задачу о положениях пространственного рычажного механизма. Под *механизмом* понимается совокупность взаимосвязанных твёрдых тел, предназначенная для преобразования движения входов на одном или нескольких твёрдых телах в движение на других твёрдых телах (выходах). Механизм может быть представлен своей структурной схемой, на которой выделяют *звенья* (твёрдые тела), соединяющиеся в подвижные *кинематические пары* с помощью различных типов *геометрических элементов* (вращательных, сферических, поступательных и др.). В основе методов конструирования механизмов и расчета кинематики лежит *принцип образования механизмов по Ассуру*, утверждающий что «*Всякий механизм представляет собою совокупность одного или нескольких двухзвенных (первичных) механизмов и одной или нескольких групп нулевой подвижности (структурных групп)*» [8]. В работе рассматривается класс пространственных рычажных механизмов, которые могут быть образованы согласно принципу Асура наложением одноконтурных структурных групп.

Прямая задача о положении для механизмов ставится следующим образом. Задано движение всех входных звеньев,

удовлетворяющее связям, которые накладывают кинематические пары, соответствующих первичных механизмов. Необходимо найти движение всех остальных звеньев механизма либо сообщить о невозможности реализации положения в тот или иной момент времени.

Согласно закону Ассура прямая задача о положениях механизма сводится к последовательному решению ряда более простых подзадач расчета положений структурных групп механизма. На рис. 1, *а* приведён пример механизма выдвижения закрылков самолёта. Механизм состоит из одного первичного механизма и четырех структурных групп ВВВ (из трех вращательных пар), связанных друг с другом. Каждая группа имеет два входа, необходимых для расчета её положения. Исходя из структурной схемы механизма строится последовательность расчета структурных групп, показанная на рис. 1, *б* – вычисления производятся последовательно по уровням 0, 1, 2 и т.д., сверху вниз (стрелками показаны соединения групп друг с другом). Положения структурных групп ВВВ вычисляются аналитически. Произведя вычисления положений механизма для разных углов φ поворота входного звена, мы получим движение всех звеньев механизма.

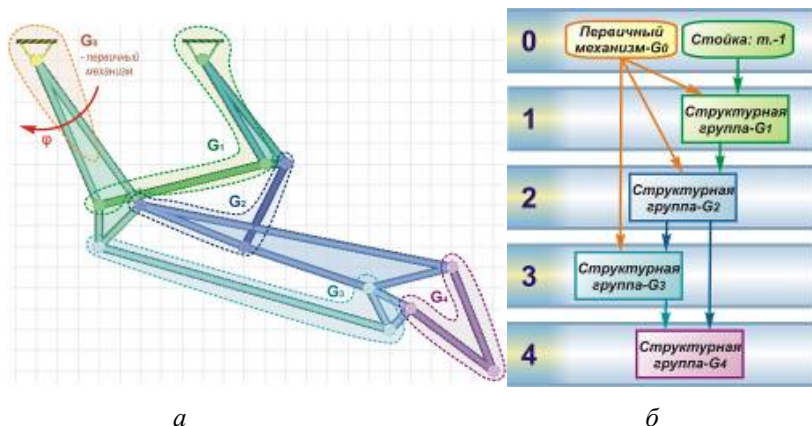


Рис. 1. Последовательность расчета структурных групп механизма

Важно отметить, что аналитический или численный расчет положения структурной группы механизма требует применения базовых векторных и матричных операций, а также широкого набора геометрических функций на их основе.

Моделирование структурной схемы механизма, разбиение механизма на группы и определение последовательности вычисления структурных групп реализовано в программной системе «Mechanics Studio.NET» [7]. Параллельный расчёт серии положений механизма на графическом процессоре выполняется по данным от системы моделирования.

Параллельный алгоритм вычисления серии N положений механизма. Вычисление серии $[1..N]$ положений пространственного рычажного механизма по заданному начальному нулевому положению и входным данным положений первичных механизмов может быть выполнено следующим образом.

Для каждого положения P механизма от 1 до N .

Последовательно для каждой структурной группы G от 1 до M :

- 1) Выполнить расчет положения группы G ;
- 2) Для каждого звена L механизма от 1 до K .

Если звену принадлежат точки или оси, положение P которых не вычислено, и имеются либо одна точка и две неколлинеарные оси, либо две точки и одна ось, либо три точки с вычисленным положением P , то выполнить пересчёт положения звена из его начального положения.

Ускорить вычисление положений рычажных механизмов можно за счёт применения аналитических методов расчета структурных групп [7], а также за счёт распараллеливания вычисления положений механизма, независимо для всех P . В связи с векторным характером вычислений и возможностью применения массового параллелизма по данным SIMD целесообразно использовать современные графические процессоры. Для реализации программы вычисления положений на графической карте была выбрана технология DirectX 11 Compute Shader [2], так как она ориентирована на работу с графическими картами разных

производителей (ATI и NVidia), достаточно близка к железу графических карт и позволяет эффективно задействовать возможности векторных вычислений языка HLSL.

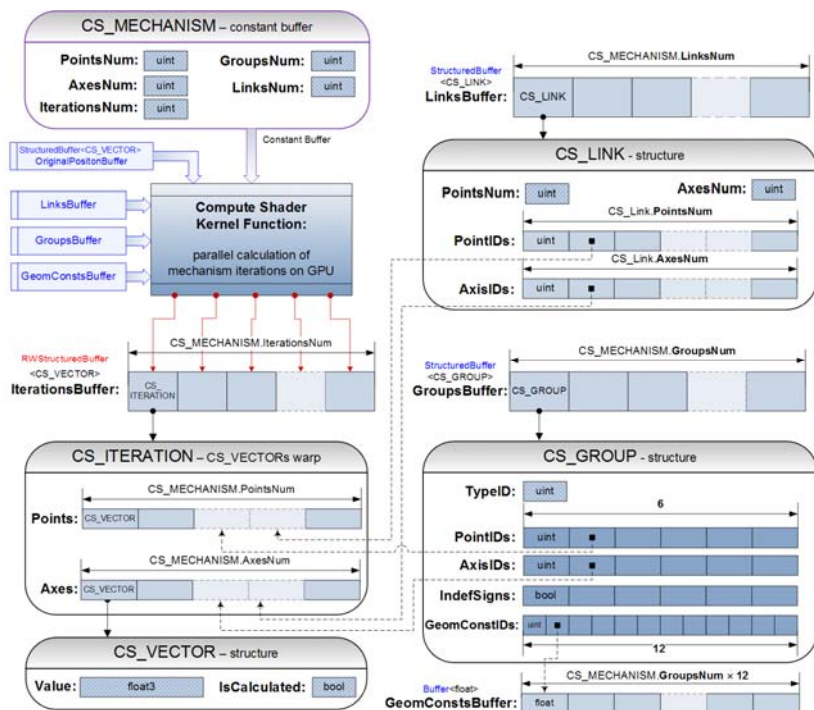


Рис. 2. Структуры данных для вычисления положений механизма на GPU

Структуры данных для расчёта серии положений механизма на GPU. Для расчёта положений механизма на графической карте данные упаковываются в структуры, представленные на рис. 2, передаются на графическую карту и используются программой вычислительного шейдера:

- CS_MECHANISM – константный буфер, хранящий размеры других буферов;
- LinksBuffer – буфер звеньев с индексами их точек и осей;

- *GroupsBuffer* – буфер групп с идентификатором типа, индексами точек и осей пар группы, знаками направления осей и индексами геометрических констант;
- *GeomConstsBuffer* – буфер геометрических констант для вычисления положений;
- *OriginalPositionBuffer* – буфер векторов точек и осей исходного положения механизма;
- *IterationsBuffer* – буфер для записи векторов точек и осей итераций механизма (свёртка элементов типа CS_ITERATION в одномерный массив векторов).

Программа вычислительного шейдера расчета положений механизма выполняется на графической карте на параллельных потоках, количество которых равно *числу итераций* – *IterationsNum*, т.е. числу положений механизма. Каждый поток шейдера независимо вычисляет вектора точек и осей *i*-го положения механизма и записывает их в *IterationsBuffer*. Алгоритм вычисления положений механизма описан блок-схемой на рис. 3. Алгоритм состоит в проходе по списку групп *GroupsBuffer*, последовательном решении задачи о положении для каждой структурной группы и расчёту положений тех звеньев из *LinksBuffer*, которые стало возможно посчитать по точкам и осям, полученным из положений пар группы. Расчёт положения структурной группы выполняется по определённым формулам для каждого типа групп [9] на основе данных из буфера *GroupsBuffer* и геометрических констант из *GeomConstsBuffer*. Расчёт положения звена выполняется путём преобразования его точек и осей из исходного положения *OriginalPositionBuffer*, путём умножения на матрицу поворота-переноса, полученную по изменению 3 известных векторов звена в текущем положении.

Результаты вычислительного эксперимента. Проведён эксперимент по сравнению времени вычисления серии N положений механизма для однопоточной реализации на C++/CLI.NET на CPU Intel Core 2 Quad Q6600 (2.4 GHz) и многопоточной реализации на Compute Shader 5.0 на GPU ATI Radeon HD 5850 (288 SIMD-массивов, 1440 скалярных ядер, 725 MHz). Для вы-

полнения вычислений на графической карте отдельно замерялось время *инициализации и передачи данных* на GPU, включающее упаковку, запись исходных данных в память GPU, загрузку результатов вычисления с GPU и их распаковку.

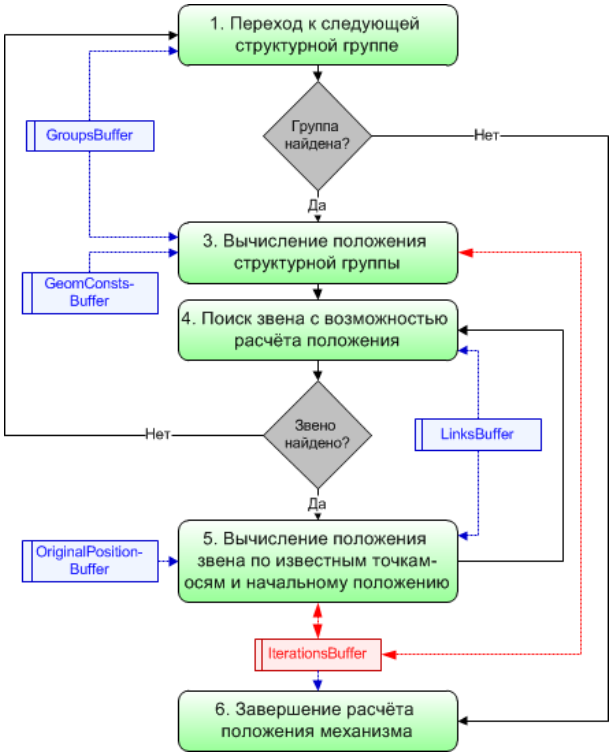


Рис. 3 Алгоритм вычисления положения рычажного механизма на GPU

Результат эксперимента (рис. 4) показывает ускорение вычислений в 100 раз при распараллеливании вычисления положений механизма на GPU, не учитывая инициализацию и передачу данных. С учётом времени инициализации данных на GPU выигрыш по сравнению с последовательными вычислениями на CPU незначителен. Это говорит о необходимости уменьшения объёма пере-

даваемых данных на GPU, возможного за счёт синтеза механизма непосредственно на GPU по параметрам.

Следует отметить что ранее представленные результаты экспериментов в публикациях [1, 3] содержали ошибочные данные о времени вычислений на GPU. Это связано с недокументированной особенностью работы DirectX: при вызове функции выполнения вычислений Dispatch происходит только добавление этой команды в очередь, а сами вычисления на графической карте выполняются асинхронно, в тот момент, когда требуется их непосредственный результат. Поэтому обыкновенный замер времени вызова Dispatch показывал только постановку этой команды в очередь. Эта проблема была решена за счёт синхронизации GPU-вычислений с CPU при помощи механизма событий DirectX.

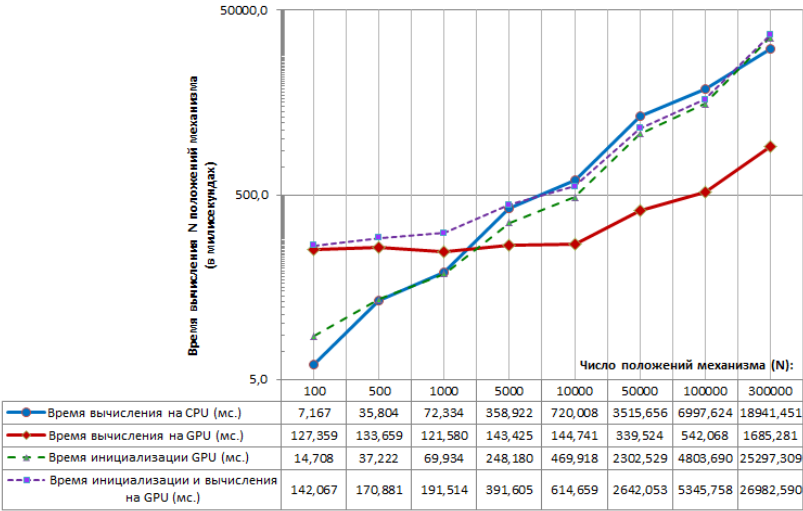


Рис. 4. Время вычисления серии N положений механизма на CPU и GPU

Разработанные структуры данных и схема вычислений описывают параллельный алгоритм расчёта серии положений механизма, в котором каждое положение вычисляется независимо. Алгоритм реализован с использованием технологии вы-

числительных шейдеров для выполнения на современных графических процессорах с сотнями векторных вычислительных ядер, что позволяет добиться ускорения вычислений в сотни раз по сравнению с CPU. Высокопроизводительное вычисление положений механизма разработано для дальнейшего применения в решении задачи параметрической оптимизации рычажных механизмов, где для вычисления одного значения целевой функции требуется вычислить серию из десятков или сотен положений конфигурации механизма [6].

Работа выполнена в рамках гранта Президента Российской Федерации для государственной поддержки молодых российских ученых – кандидатов наук (МК-3473.2010.1), а также госконтракта 02.740.11.0839.

Список литературы

1. Городецкий Е.С. Результаты эксперимента по распараллеливанию вычисления серии положений пространственного рычажного механизма на графическом процессоре с помощью DirectX 11 Compute Shader // Технологии Microsoft в теории и практике программирования: материалы конф. – Н. Новгород: Изд-во Нижегород. гос. ун-та, 2010. – URL: http://www.itlab.unn.ru/archive/MSConf10/msconf-2010_book.pdf. – С. 85–88.

2. Городецкий Е.С. Применение технологии DirectX 11 Compute Shader для вычислений общего назначения на графическом процессоре // Технологии Microsoft в теории и практике программирования: материалы конф. – Н. Новгород: Изд-во Нижегород. гос. ун-та. – 2010. – С. 81–84.

3. Городецкий Е.С., Турлапов В.Е. Вычисление функций критериев и ограничений в задаче оптимизации пространственного рычажного механизма с распараллеливанием на графическом процессоре // Параллельные вычислительные технологии (ПаВТ'2010): тр. междунар. науч. конф. – Челябинск: ЮУрГУ, 2010. – 206

URL: <http://omega.sp.susu.ac.ru/books/conference/PaVT2010>. – С. 454–461.

4. Городецкий Е.С. Реализация параллельного вычисления серии положений пространственного рычажного механизма на графическом процессоре с помощью DirectX 11 Compute Shader // Высокопроизводительные параллельные вычисления на кластерных системах: материалы XI Междунар. конф.-семинара. – Владимир: Изд-во Владимир. гос. ун-та, 2009. – С. 131–135.

5. Городецкий Е.С. Алгоритм параллельного вычисления серии положений пространственного рычажного механизма // Технологии Microsoft в теории и практике программирования: материалы конф. – Н. Новгород: Изд-во Нижегород. гос. ун-та, 2009. – С. 102–108.

6. Городецкий Е.С. Задача параметрической оптимизации пространственных механизмов с параллельными вычислениями кинематики на графической карте // Высокопроизводительные параллельные вычисления на кластерных системах: материалы Седьмой Междунар. конф.-семинара. – Н.Новгород: Изд-во Нижегород. гос. ун-та, 2007. – С. 117–124.

7. Городецкий Е.С. Система проектирования кинематики пространственных механизмов Mechanics Studio .NET на основе Managed DirectX // Технологии Microsoft в теории и практике программирования: материалы конф. – Н. Новгород: Изд-во Нижегород. гос. ун-та, 2007. – С. 48–52.

8. Заблонский К.И., Белоконев И.М., Щекин Б.М. Теория механизмов и машин: учеб. для вузов. – Киев: Выща шк., 1989. – 376 с.

9. Турлапов В.Е. Задача о положениях и классификация одноконтурных структурных групп пространственных рычажных механизмов // Прикладная геометрия. – 2000. Вып. 2. – № 2. – С. 1–22.