

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМ. Н. Э. БАУМАНА (национальный  
исследовательский университет)

УДК \_\_\_\_\_

№ госрегистрации \_\_\_\_\_

Инв. № \_\_\_\_\_

УТВЕРЖДАЮ

\_\_\_\_\_  
головой исполнитель НИР

\_\_\_\_\_  
« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

ОТЧЁТ  
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по теме:

Программа моделирования движения воды с использованием вокселей  
(промежуточный)

Руководитель темы

\_\_\_\_\_ А. С. Кострицкий

Москва 2019

## РЕФЕРАТ

Отчет содержит 18 стр., 2 рис., 17 источн., 2 прил.

Это пример каркаса расчётно-пояснительной записки, желательный к использованию в РПЗ проекта по курсу РСОИ .

## СОДЕРЖАНИЕ

Введение .....	4
1 Аналитический раздел .....	5
1.1 Физическая модель .....	5
1.2 Существующие подходы к симуляции жидкостей .....	5
1.3 Анализ методов визуализации жидкостей .....	6
1.3.1 Трассировка лучей .....	7
1.3.2 Реконструкция поверхности .....	8
1.4 Оптимизации воксельного способа решения задачи объёмного рендеринга .....	8
1.5 Вывод .....	9
2 Конструкторский раздел .....	10
2.1 Архитектура приложения .....	10
2.2 Вывод .....	10
3 Технологический раздел .....	11
3.1 Требования к программному обеспечению .....	11
3.2 Используемые технологии .....	11
3.3 Листинги кода .....	12
3.4 Вывод .....	12
4 Исследовательский раздел .....	13
4.1 Примеры использования .....	13
4.2 Выводы .....	13
Заключение .....	14
Список использованных источников .....	15
Приложение А Картинки .....	17
Приложение Б Еще картинки .....	18

## ВВЕДЕНИЕ

Визуализация различных явлений становится всё более важной во множестве инженерных областей знаний. Задача объёмного рендеринга имеет большое значение, например, в визуализации данных компьютерной и магнитно-резонансной томографии[8]. Интерактивная 3D симуляция позволяет учёным ясно воспринимать и оценивать результаты собственных исследований.

В настоящее время для этого используются различные вычислительные техники обработки и графического представления экспериментальных данных[5].

В данной работе рассматривается 3D симуляция воды. Симуляция жидкостей, в целом, является примером того, что получаемые о них сведения без соответствующего 3D изображения довольно сложны для человеческого восприятия.

Целью работы является разработка программного продукта для моделирования движения воды с использованием воксельной графики. Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать существующие методы моделирования движения жидкостей и методы рендеринга с помощью вокселей;
- спроектировать программное обеспечение, симулирующее поведение воды;
- реализовать программу и проверить её работоспособность.

## 1 Аналитический раздел

В данном разделе производится анализ методов вычислений характеристик жидкостей и их преобразований в графический формат.

Далее рассматриваются идеи применения данных методов к симуляции жидкостей и существующие решения в этой области.

### 1.1 Физическая модель

Жидкости моделируются как векторное поле скорости жидкости и скалярное поле плотности. Движение задаётся уравнениями Навье-Стокса [1].

Далее рассматривается только движение воды (несжимаемой жидкости) в условиях постоянной температуры.

Тогда уравнения Навье-Стокса в векторной форме принимают следующий вид:

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = \vec{F} - \frac{1}{\rho} \nabla p + \eta \Delta \vec{v}. \quad (1.1)$$

В уравнении 1.1  $\vec{v}$  - скорость частицы воды,  $t$  - время,  $\vec{F}$  - внешняя удельная сила,  $p$  - давление,  $\eta = \frac{\mu}{\rho}$  - кинематический коэффициент вязкости,  $\nabla$  - оператор Гамильтона,  $\Delta$  - оператор Лапласа.

Данная физическая модель лежит в основе многих подходов симуляции жидкостей[8]. Их обзор приведён далее.

В статистической физике модель поведения частиц жидкости описывается кинетическим уравнением Больцмана. Данная модель применима для систем, где есть ограничения на малую скорость частиц[9].

### 1.2 Существующие подходы к симуляции жидкостей

В вычислениях поведения жидкости необходимо представить физическую модель в дискретном виде. Данную проблему решает вычислительная гидродинамика - совокупность теоретических, экспериментальных

и численных методов, предназначенных для моделирования потоковых процессов.

Наиболее распространёнными методами описания характеристик жидкости в вычислительной гидродинамике являются:

- сеточные методы Эйлера;
- метод гидродинамики сглаженных частиц;
- методы, основанные на турбулентности;
- метод решёточных уравнений Больцмана.

Сеточные методы Эйлера являются наиболее простым решением симуляции жидкостей. Они заключаются в поиске решения задачи Коши для функций, заданных таблично. Для каждого узла функции уровня жидкости требуется вычисление значений разложений Тейлора в их окрестностях. Решение задачи Коши в данном случае аппроксимирует решение уравнения Навье-Стокса[11].

Метод гидродинамики сглаженных частиц и методы, основанные на турбулентности, заключаются в выборе размера частицы ("длины сглаживания"), на котором их свойства "сглаживаются" посредством функции ядра или интерполяции, и решения уравнений Навье-Стокса с учётом вязкости и плотности. Это позволяет эффективно моделировать поведение жидкостей, газов и даже использовать в астрофизике[12].

Метод решёточных уравнений Больцмана основан на кинетическом уравнении Больцмана, упомянутом ранее. Этот метод поддерживает многофазные жидкости, наличие теплопроводности и граничные условия на макроскопическом уровне[10].

### 1.3 Анализ методов визуализации жидкостей

Основными требованиями к методам симуляции жидкостей со стороны компьютерной графики являются визуальная правдоподобность и скорость анимации.

Исходя из этих условий, в компьютерной графике используются специально модифицированные и оптимизированные методы, основанные на указанных ранее. При этом решается задача объёмного рендеринга - на каждый кадр требуется найти проекцию трёхмерного дискретного набора данных о жидкости[13].

В настоящее время существует всего 2 принципиально различных техники объёмного рендеринга: трассировка лучей и реконструкция поверхности[8].

### 1.3.1 Трассировка лучей

Данная техника состоит в том, чтобы создать виртуальный экран и далее проследить путь луча до тех пор, пока он проходит сквозь анимируемые объекты. В итоге луч позволяет определить цвет каждого отдельного пикселя. Настоящий алгоритм трассировки лучей имеет значительную вычислительную сложность, поэтому его часто заменяют на правдоподобные аппроксимации[8].

Одним из способов аппроксимации является трассировки лучей путём замены полигонов на воксели. Это позволяет существенно повышать производительность анимации за счёт возможностей параллелизма и снижения вычислительной сложности. С другой стороны для хранения вокселей требуется большее количество памяти, чем для хранения полигонов.

Другим популярным методом является метод срезов объёмных текстур так же, как и воксельный, является математическим упрощением алгоритма трассировки лучей. Данные сцены представляются в виде 3D текстуры. Рендеринг производится с самого дальнего по отношению к наблюдателю слоя текстуры. В итоге видимыми являются ближайшие из частей слоя. В качестве составляющих срезов могут быть использованы как воксели, так и полигоны. Этот способ оптимизирован аппаратно для некоторых графических процессоров и в этом случае имеет преимущество перед воксельным методом по времени работы.

### 1.3.2 Реконструкция поверхности

Реконструкцией поверхности, или трёхмерной реконструкцией, называют процесс получения облика реальных объектов.

В качестве входных данных получают множество точек, характеризующее объект. Далее выбирают точки на поверхности объекта так, что получают полигоны, которые имеют необходимую конфигурацию для аппроксимации формы поверхности в её видимой части. Набор результирующих полигонов - реконструированная модель.

Основным алгоритмом, реализующим этот метод, является Marching cubes. Он широко используется в компьютерной и магнитно-резонансной томографии[14]. Суть алгоритма в том, что для кубов, находящихся на поверхности объекта, известно, какие точки лежат внутри объекта и какие снаружи. Это позволяет выбрать приближающий полигон так, чтобы его вершины находились на отрезках, соединяющих вершины куба, в примерных точках реального пересечения с объектом. Таким образом, однажды применив алгоритм, можно использовать объект в анимации, если он не изменяет форму поверхности. Для предметов, теряющих форму, например, жидкостей требуется повторное вычисление реконструированной поверхности. Подобные вычисления имеют значительную сложность, поэтому на практике не используются[8].

### 1.4 Оптимизации воксельного способа решения задачи объёмного рендеринга

Существуют случаи, в которых система объёмного рендеринга получает на вход трёхмерные данные, в которых есть области, не требующие отрисовки. В подобных ситуациях следует пропускать вычисление результирующего изображения для пустого пространства[2].

Для хранения вокселей можно использовать различные структуры данных. Наиболее простой является трёхмерный массив вокселей. При хранении вокселей в виде массива может требоваться значительное количество памяти, даже если исходная информация достаточно однородная.



Поэтому для оптимизации памяти применяют такие структуры данных, как октодерево и binary sparse partitioning дерево[2].

Использование иерархических структур данных имеет следующий недостаток - они статичны, и их использование для моделей, теряющих форму, требует перестроение всего дерева[2].

Поскольку вода теряет свою форму, были предложены оптимизации не для хранения вокселей, а для обработки лучей и самих данных симуляции. Данные оптимизации основаны на принципах обработки чисел с плавающей запятой на аппаратном обеспечении, а также на возможности дальнейшего параллелизма вычислений[8].

## 1.5 Вывод

Исходя из полученных сведений о достоинствах и недостатках каждого из методов объёмного рендеринга в дальнейшем рассматривается реализация с помощью вокселей. Она обеспечивает возможность изменения формы жидкости и наиболее быструю из предложенных техник трассировки лучей.

## 2 Конструкторский раздел

В данном разделе описано проектирование метода рендеринга воды с помощью вокселей с указанием соответствующих схем алгоритмов.

### 2.1 Архитектура приложения

### 2.2 Вывод

### 3 Технологический раздел

В данном разделе описаны требуемые средства и подходы к реализации ПО по ранее указанным методам.

#### 3.1 Требования к программному обеспечению

Разработанное ПО должно моделировать движение воды с использованием вокселей.

Моделирование движения должно осуществляться с использованием операций переноса, масштабирования и поворота.

#### 3.2 Используемые технологии

Для реализации ПО выбран язык Clojure. Данный язык программирования является компилируемым и, одновременно с этим, динамическим. Clojure - преимущественно язык функционального программирования, который поддерживает нативный доступ к Java фреймворкам [15].

Данные свойства языка позволяют вести разработку приложений с помощью REPL (Read-Eval-Print Loop), модифицируя их во время выполнения. Это может быть полезно для добавления новых функциональностей в программу без её повторной сборки [16].

Для создания пользовательского интерфейса используется Java библиотека Swing. Swing предоставляет широкий набор компонентов для создания графического пользовательского интерфейса. Данные компоненты полностью реализованы на Java, поэтому их внешний вид не зависит от платформы.

Для организации проекта на Clojure и для автоматизации сборки используется Leiningen. Все конфигурации для сборки приложения, запуска REPL и описания зависимостей управляются с помощью этого модуля[17].

### 3.3 Листинги кода

В листинге 3.3 указана конфигурация сборки проекта на языке Clojure.

```
1 (defproject computer-graphics-coursework-backend "0.1.0-SNAPSHOT"
2   :description "Pudov's computer graphics coursework"
3   :url "https://github.com/DPudov"
4   :license {:name "Eclipse Public License"
5             :url "http://www.eclipse.org/legal/epl-v10.html"}
6   :dependencies [[org.clojure/clojure "1.10.0"]
7                 [seesaw "1.5.0"]]
8   :repl-options {:init-ns computer_graphics_coursework_backend.core}
9   :main computer_graphics_coursework_backend.core
10  :profiles {:uberjar {:aot :all}}
11  :java-cmd "/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java")
```

### 3.4 ВЫВОД

## 4 Исследовательский раздел

В данном разделе проводится апробация разработанной программы.

### 4.1 Примеры использования

### 4.2 Выводы

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы стало ясно, что ничего не ясно...

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тирский Г. А. Большая российская энциклопедия. Том 21. — Большая российская энциклопедия, 2012.
2. Bautembach Dennis. Animated Sparse Voxel Octrees // UNIVERSITY OF HAMBURG Department of Informatics. — 2011.
3. Zadick Johanne, Kenwright Benjamin, Mitchell Kenny. Integrating Real-Time Fluid Simulation with a Voxel Engine // The Computer Games Journal. — 2016.
4. Premoze S., Ashikhmin M. Rendering Natural Waters. — Hong Kong, 2000. — P. 22–30.
5. Magnetic Resonance Imaging: Physical Principles and Sequence Design / Robert W. Brown, Y.-C. Norman Cheng, E. Mark Haacke et al. — John Wiley Sons, 2014. — P. 976.
6. VanderHart Luke, Neufeld Ryan. Clojure Cookbook: Recipes for Functional Programming. — O'Reilly Media, Inc., 2014.
7. Lombard Yann. Realistic Natural Effect Rendering: Water I. — 2004. — Access mode: <https://www.gamedev.net/articles/programming/graphics/realistic-natural-effect-rendering-water-i-r2138/>.
8. Ash Michael. Simulation and Visualization of a 3D Fluid : Master's thesis / Michael Ash ; Université d'Orléans. — 2005.
9. Мякишев Г. Я. Кинетическое уравнение Больцмана. — Режим доступа: <https://www.booksite.ru/fulltext/1/001/008/061/212.htm>.
10. Баранов Василий. Моделирование гидродинамики: Lattice Boltzmann Method. — 2013. — Режим доступа: <https://habr.com/ru/post/190552/>.

11. Асламова В. С., Колмогоров А. Г., Сумарокова Н. Н. Вычислительная математика. Часть вторая: Учебное пособие для студентов дневного и заочного обучения технических и химико-технологических специальностей. — Ангарская государственная техническая академия, 2005. — С. 94.
12. Bate Matthew R., Bonnell Ian A., Bromm Volker. The Formation of Stars and Brown Dwarfs and the Truncation of Protoplanetary Discs in a Star Cluster. — 2002. — Access mode: <https://www.ukaff.ac.uk/starcluster/>.
13. Lacroute Philippe, Levoy Marc. Fast Volume Rendering Using a Shear-Warp Factorization of Viewing Transformation // Computer Graphics Proceedings, Annual Conference Series. — 1994.
14. Anderson Ben. An Implementation of the Marching Cubes Algorithm. — 2005. — Access mode: [http://www.cs.carleton.edu/cs\\_comps/0405/shape/marching\\_cubes.html](http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html).
15. Hickey Rich. The Clojure Programming Language. — Access mode: <https://clojure.org/>.
16. Hickey Rich. Programming at the REPL: Introduction. — Access mode: <https://clojure.org/guides/repl/introduction>.
17. Hagelberg Phil. Leiningen. — Access mode: <https://leiningen.org/>.



## ПРИЛОЖЕНИЕ А

### КАРТИНКИ

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Рисунок А.1 — Картинка в приложении. Страшная и ужасная.

## ПРИЛОЖЕНИЕ Б

### ЕЩЕ КАРТИНКИ

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Рисунок Б.1 — Еще одна картинка, ничем не лучше предыдущей. Но надо же как-то заполнить место.