



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления  
КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

## КУРСОВАЯ РАБОТА

*НА ТЕМУ:*

*Визуализация ландшафтной сцены с облаками*

Студент

ИУ7-54Б

(группа)

(подпись, дата)

Т. А. Асадуллин

(И.О. Фамилия)

Руководитель курсового  
проекта

(подпись, дата)

К. А. Кивва

(И.О. Фамилия)

2024 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Аналитическая часть</b>	<b>5</b>
1.1 Формализация задачи и объектов	5
1.2 Алгоритмы генерации облаков	6
1.2.1 Жидкостная симуляция	6
1.2.2 Воксельная генерация	7
1.2.3 Генерация на основе обратной трассировки лучей	7
1.3 Модели освещения	8
1.3.1 Закон Бугера — Ламберта — Бера	8
1.3.2 Фазовая функция Хеньи — Гринстейна	8
<b>2 Конструкторская часть</b>	<b>10</b>
<b>3 Технологическая часть</b>	<b>11</b>
<b>4 Исследовательская часть</b>	<b>12</b>
<b>Заключение</b>	<b>13</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>14</b>

# ВВЕДЕНИЕ

**Компьютерная графика** – совокупность методов и средств преобразования в графическую форму и из графической формы с помощью ЭВМ [1]. Конечным продуктом компьютерной графики является изображение [2]. Ключевые моменты, которые компьютерная графика рассматривает – как [2]

- изображения представляются в компьютерной графике;
- изображения готовятся для визуализации;
- предварительно подготовленные изображения рисуются;
- осуществляется взаимодействие с изображением.

**Цель работы** – разработка программного обеспечения для визуализации динамической ландшафтной сцены с облаками.

для достижения поставленной цели требуется решить следующие задачи:

- изучить предметную область;
- спроектировать программное обеспечение;
- выбрать средства реализации программного обеспечения и создать его;
- провести исследование разработанного программного обеспечения.

# 1 Аналитическая часть

В аналитической части будут формализованы задачи и объекты сцены, определены геометрические и оптические характеристики объектов сцены. Также будут проанализированы и описаны алгоритмы, используемые для визуализации ландшафтной сцены с облаками. Будут установлены допустимые диапазоны и ограничения, накладываемые на входные данные.

## 1.1 Формализация задачи и объектов

Объектами сцены являются:

### 1) Облака (облачный пейзаж)

- Высота, на которой находятся облака;
- Скорость движения облаков по горизонту;
- Кучность: степень сжатия и плотности облаков, что влияет на их внешний вид и отбрасываемую тень.
- Плотность: определяет, сколько солнечного света облака могут заблокировать, что влияет на освещение ландшафта.

### 2) Ландшафт (ландшафтный пейзаж) –

- Рельеф: плоский равнинный.
- Материалы и текстуры: характеристики поверхности, такие как цвет и отражательная способность.
- Освещение от солнца и теней: ландшафт получает освещение, которое зависит от плотности облаков и положения солнца, а также отбрасываемых теней.

### 3) Бесконечно удаленный источник света (солнце) –

- Расположение: определяется положением на небесной сфере. Положение солнца влияет на длину и направление теней.
- Интенсивность: определяет, насколько ярко освещен ландшафт, также зависит от плотности облаков.

### 4) Наблюдатель (камера) –

- Расположение: координаты и угол обзора камеры, позволяющие наблюдать сцену с разных ракурсов.
- Поле зрения: угол обзора, влияющий на широту сцены.

Определение диапазонов и ограничений:

- **Положение солнца:** угол наклона от  $0^\circ$  до  $90^\circ$  над горизонтом и азимутальный угол от  $0^\circ$  до  $180^\circ$ .
- **Пространственное перемещение:** только для таких объектов, как камера и солнце.

## 1.2 Алгоритмы генерации облаков

Существует несколько подходов к реализации облаков [3–5]:

- **Геометрический:** облака представляют собой, например, набор треугольников, сфер или прямоугольников. Геометрический подход к созданию облаков имеет смысл в определенной стилистике изображения [3].
- **Двумерная текстура:** простой и малозатратный подход, но такая статичная картинка имеет смысл только как дальнеплановые статичные изображения, через которые, например, нельзя пролететь сквозь. К тому же такие облака не могут производить тени [3].
- **Объемные (*volumetric*):** динамические облака, с которыми можно взаимодействовать и которые способны производить тени [4, 6]. Именно поэтому такие облака будут реализованы в данной работе.

Исходя из требований к алгоритму, выдвигаемых в современной игровой индустрии [3–5], условия, которые будут использованы в данной работе:

- Облака должны быть объемные;
- Облака должны генерироваться процедурно;
- Должен быть быстродействующим.

### 1.2.1 Жидкостная симуляция

Использование жидкостной симуляции для создания объемных облаков: создать простые объекты (сферы, шары), вокселизировать их и рассматривать их как жидкость, получая похожие на объемные облака фигуру [4]. Современные физические модели облаков, основаны на решении уравнений Навье-Стокса [5], что влечет за собой следующие недостатки [5]:

- Алгоритм медленный;

- Сложность контроля генерации;
- Сложность реализации.
- Не используется на практике [3]

### 1.2.2 Воксельная генерация

Алгоритм заключается в генерации ограничивающего параллелепипеда (bounding box), состоящего из вокселей, хранящих информацию о цвете [4].  
Преимущества:

- Хорошо сочетается с алгоритмом построением теней

Недостатки:

- Высокие затраты памяти;
- Сложность обработки большого количества вокселей в реальном времени;
- Необходимость оптимизаций для обработки больших объемов.
- Не используется на практике [4]

### 1.2.3 Генерация на основе обратной трассировки лучей

Из точки наблюдателя для каждого пикселя грани высчитывается его итоговый цвет [5, 7]. Алгоритм также опирается на ограничивающий параллелепипед, но вместо этого визуализируются лишь видимые грани параллелепипеда. Вместо вычисления каждого вокселя, алгоритм ориентируется на пиксели, видимые пользователю, и рассчитывает итоговые цвета только для них.

- Хорошо сочетается с алгоритмом построением теней;
- Меньшие затраты памяти;
- Сниженные вычислительные затраты благодаря обработке только видимых пикселей.
- Рекомендованы к использованию на практике [4, 5, 8]

Генерация на основе обратной трассировки лучей показывает преимущество перед воксельной и жидкостной генерациях, так как обрабатывает только видимые пиксели, что снижает вычислительные затраты и экономит память, что необходимо при формировании динамического изображения.

## 1.3 Модели освещения

### 1.3.1 Закон Бугера — Ламберта — Бера

Для облаков некоторая часть света рассеивается от направления распространения, а еще большее количество поглощается каплями воды и молекулами озона, но остается часть, которая продолжает движение без изменений [4, 5].

*Закон Бугера—Ламберта—Бера* определяет ослабление пучка света при поглощении средой.

$$I_l = I_0 e^{-k_\lambda l}, \quad (1.1)$$

где  $I_0$  — интенсивность света на входе в вещество,  $k_\lambda$  — показатель поглощения.

### 1.3.2 Фазовая функция Хеньи — Гринштейна

Облака представляют собой анизотропную среду (среда, где физические свойства: показатели преломления, скорость распространения и пр. — различаются в различных направлениях внутри этой среды) из-за того, что облака представляют собой капли жидкой воды и кристаллов ледяного льда. Для описания этого используют фазовую функцию (индикатриса) Хеньи — Гринштейна [4, 5]

*Фазовая функция Хеньи — Гринштейна* определяет угловое распределение интенсивности:

$$P(g, \theta) = \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (1.2)$$

где  $\theta$  — угол рассеяния, который определяется как угол между направлением распространения исходного и рассеянного света и  $g$  — параметр асимметрии, который описывает среднее значение косинуса угла рассеяния.

## Вывод

В аналитической части формализованы задачи и объекты сцены, определены геометрические и оптические характеристики объектов сцены. Так-

же проанализированы и описаны алгоритмы, используемые для визуализации ландшафтной сцены с облаками. Установлены допустимые диапазоны и ограничения, накладываемые на входные данные. Был выбран алгоритм использующий обратную трассировку лучей для генерации объемных облаков.



## **2 Конструкторская часть**

В конструкторской части будет спроектировано разрабатываемое программное обеспечение и формально описаны используемые алгоритмы.

### **Вывод**

### 3 Технологическая часть

В технологической части будут выбраны и описаны средства реализации программного обеспечения и представлены детали его реализации.

Для реализации программного обеспечения был выбран язык программирования *Rust* [9], так как он позволяет реализовать все алгоритмы, выбранные в результате проектирования и поддерживает все требуемые структуры данных.

Был выбран фреймворк *egui* [10] для реализации интерфейса программного обеспечения, так как в нём присутствуют инструменты для работы с изображениями и разработки интерфейса.

### Вывод

## **4 Исследовательская часть**

В исследовательской части будет проведено исследование разработанного программного обеспечения.

### **Вывод**

## Заключение

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. А.В. Куров. Конспект лекций по дисциплине «Компьютерная графика». 2024 год.
2. Роджерс Д. Алгоритмические основы машинной графики. Москва: Мир, 1989.
3. UNIGINE. Волюметрические облака в UNIGINE2. 2022. Дата доступа: 2024-11-27. URL: <https://doi.org/10.3390/sym10040125>.
4. Guerrilla Games. The Real-Time Volumetric Cloudscapes of Horizon Zero Dawn. 2023. Дата доступа: 2024-10-07. URL: <https://www.guerrilla-games.com/read/the-real-time-volumetric-cloudscapes-of-horizon-zero-dawn>.
5. Jiménez de Parga Carlos, Gómez Palomo Sebastián Rubén. Efficient Algorithms for Real-Time GPU Volumetric Cloud Rendering with Enhanced Geometry. 2022. Дата доступа: 2024-11-27. URL: <https://doi.org/10.3390/sym10040125>.
6. Efficient Cloud-Based Rendering of Real-Time Volumetric Clouds: Master's thesis. 2013.
7. Real-time Volumetric Rendering: Master's thesis. 2013. <http://patapom.com/topics/Revision2013/Revision%202013%20-%20Real-time%20Volumetric%20Rendering%20Course%20Notes.pdf>.
8. Haggstrom Fredrik. Real-Time Volumetric Cloud Rendering: Master's thesis: Linköping University, Department of Science and Technology. 2018. Дата доступа: 2024-10-07. URL: <https://www.diva-portal.org/smash/get/diva2:1223894/FULLTEXT01.pdf>.

9. The Rust Programming Language. 2024. Accessed: 2024-11-27. URL: <https://doc.rust-lang.org/>.
10. egui: A simple, fast, and portable GUI library for Rust. 2024. Accessed: 2024-11-27. URL: <https://github.com/emilk/egui>.