



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

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:

Моделирование облаков

Студент ИУ7-526

(Группа)

Руководитель курсового проекта

_____ Мамаев М.В.

(Подпись, дата)

(И.О. Фамилия)

_____ Барышникова М.Ю.

(Подпись, дата)

(И.О. Фамилия)

Оглавление

ВВЕДЕНИЕ.....	4
1. Аналитический раздел	5
1.1. Анализ предметной области	5
1.2. Алгоритмы моделирования облаков	5
1.2.1. Использование заранее нарисованных текстур неба и интерполяция между ними	6
1.2.2. Модели мезомасштабных процессов	7
1.2.3. Генерация изображений облаков используя шум Перлина	8
1.2.4. Трёхмерное моделирование облаков.....	10
1.3. Алгоритмы визуализации облаков	11
1.3.1. Использование билбордов.....	12
1.3.2. Визуализация используя воксели	12
1.3.3. Перспективная проекция	14
1.4. Обоснование выбора метода моделирования и алгоритма визуализации	14
2. Конструкторский раздел.....	16
2.1. Алгоритм работы приложения.....	16
2.2. Разработка алгоритмов	16
2.2.1. Шум Перлина.....	16
2.2.2. Создание воксельной сетки	19
2.2.3. Преобразования камеры	19
3. Технологический раздел	21
3.1. Выбор средств реализации.....	21
3.2. Структура приложения.....	21
3.3. Интерфейс приложения	22

4. Исследовательский раздел.....	24
4.1. Постановка задачи анализа	24
4.2. Технические характеристики	24
4.3. Результаты анализа	24
4.4. Примеры визуализации программы	27
ЗАКЛЮЧЕНИЕ	28
СПИСОК ЛИТЕРАТУРЫ	29
ПРИЛОЖЕНИЕ А.....	30

ВВЕДЕНИЕ

С появлением систем виртуальной реальности, в которых наблюдатель погружается в мир модели, возникает потребность визуализации виртуальной среды, максимально приближенной к естественным условиям. Такие объекты как ландшафт, водная поверхность, растительность, небо и облака являются неотъемлемой частью практически любой естественной сцены. Моделирование облаков как природного феномена является одной из наиболее важных и трудных задач, решаемых при визуализации реалистичного изображения неба. Алгоритмы визуализации облаков используются во всех приложениях, где необходимо создать максимально естественную среду, а также в приложениях, в которых облака играют основную роль — в авиасимуляторах.

Целью данной работы является изучение различных алгоритмов визуализации облаков и реализация одного из них. При этом необходимо обеспечить возможность настройки параметров облака (размеров, плотности).

Для достижения этой цели поставлены следующие задачи.

1. Изучить предметную область.
2. Изучить существующие алгоритмы визуализации облаков и провести их сравнительный анализ.
3. Разработать программу на основе одного из существующих алгоритмов.
4. Исследовать зависимость времени работы алгоритма от сложности изображения)

Программа должна соответствовать техническому заданию и создавать изображения облаков.

1. Аналитический раздел

В данном разделе рассматриваются существующие алгоритмы создания облаков и их свойства, анализируются достоинства и недостатки каждого метода. Также выбраны методы реализации.

1.1. Анализ предметной области

Высококачественная облачная визуализация зарекомендовала себя как мощный инструмент повышения реалистичности виртуальных сред и решения проблемы визуализации больших открытых пространств.

Облака являются динамическими объектами, их структура сложна и неоднородна. Плотность капель воды, рассеянных в облаках, не одинакова ни на границах облаков, ни внутри них. Облака постоянно меняются, особенно при изменении погодных условий и выпадении осадков. При анализе и выборе алгоритма моделирования облаков следует иметь в виду, что основное внимание должно быть уделено частицам на поверхности облаков, так как плотность частиц внутри них можно считать постоянной, поскольку она влияет только на преломление и рассеяние света, и в визуализации это можно отразить, придав облакам белый цвет.

1.2. Алгоритмы моделирования облаков

Облака – это объекты, которые из-за своей структуры не могут быть непосредственно представлены простыми полигонами. В настоящее время известны три метода аппроксимации неба и облаков:

1. Использование заранее нарисованных текстур неба и интерполяция между ними.
2. Модели мезомасштабных процессов.
3. Генерация двумерных изображений облаков с помощью шума Перлина.
4. Представление облака как набора полигонов с текстурами, всегда обращенными к наблюдателю.

Рассмотрим каждый из способов отдельно.

1.2.1. Использование заранее нарисованных текстур неба и интерполяция между ними

Если камера находится достаточно близко к поверхности земли, то можно использовать заранее нарисованные текстуры для отображения неба. Тогда небесную поверхность можно представить как плоскость, с нанесенной на нее текстурой. Для достижения большей реалистичности следует представить небесную поверхность как небесную сферу, т.е. вместо плоскости использовать некую поверхность, представляющую собой часть сферы. Эта поверхность должна быть пологой, чтобы создавался эффект удаленности горизонта. За счет использования готовых текстур высокого качества можно добиться практически (высокой степени достоверности и реализма) фотореалистичного уровня изображения.

Для визуализации динамического неба можно использовать следующие подходы:

- Использовать несколько слоев текстур и сдвигать их относительно друг друга.
- Использовать интерполяцию между несколькими картинками неба.

Для визуализации смены дня и ночи можно применять различные цвета в зависимости от времени суток, а также использовать эффект послесвечения солнца или луны.

У данного метода есть существенные недостатки:

1. Изображение может быть сформировано только в том случае, если наблюдатель находится близко к поверхности. Если положение наблюдателя произвольно, то формирование изображения невозможно.
2. Для получения реалистичного изображения неба требуется большой набор текстур высокого разрешения для интерполяции. Это приводит к тому, что текстуры занимают много памяти.

Преимуществом этого метода является возможность получения высококачественных изображений.

Этот метод может быть использован в системах, где наблюдатель находится близко к земле, и система не предъявляет высоких требований к динамичности неба и облаков. [4]

1.2.2. Модели мезомасштабных процессов

Мезомасштабные процессы охватывают масштабы от нескольких километров до нескольких сотен километров. Исследование и моделирование таких процессов требует более сложных подходов, чем при рассмотрении масштабов, например, мезо- или микрослоя атмосферы. Для моделирования мезомасштабных процессов в облаках используют различные численные модели и методы.[9]

Существующие методы и модели:

- 1) Мезомасштабные численные модели атмосферы (MM5, WRF): это численные модели, которые позволяют учитывать мезомасштабные явления в атмосфере, такие как конвекция, вихри и турбулентность. Они обычно включают уравнения гидродинамики и термодинамики, а также параметры облаков.
- 2) Параметризация облаков: Мезомасштабные модели часто включают параметризации облаков, чтобы описать их влияние на радиацию, теплообмен и другие факторы. Эти параметризации учитывают процессы формирования облаков, осадков и теплообмена с поверхностью.
- 3) Модели микрофизики облаков: для более детального моделирования образования и эволюции облаков используются модели микрофизики. Они учитывают процессы конденсации, кристаллизации и коагуляции частиц в облаках.
- 4) Радиационные модели: Мезомасштабные процессы в облаках сильно влияют на радиационный баланс Земли. Радиационные модели учитывают взаимодействие солнечного излучения и излучения поверхности Земли с облаками.

1.2.3. Генерация изображений облаков используя шум Перлина

Использование алгоритма шума Перлина для симуляции облачной атмосферы является наиболее простым и реалистичным подходом. Этот метод широко применяется при создании визуальных эффектов, включая генерацию тумана, облаков, пламени и других природных явлений. Идея заключается в представлении небесной поверхности как части большой сферы, а облаков — в виде двумерной текстуры.

Поверхность облаков неоднородна, но далека от полного хаоса. В некоторых областях наблюдается сгущение облаков, тогда как в других — их разреженность. Визуализацию облачных форм можно проводить с использованием хаотичной функции, ядро которой подчиняется определенному закону. Один из наиболее эффективных подходов — применение функций Perlin Noise. [5]

Функции шума Перлина представляют собой синтез шумовых и интерполяционных функций, которые обеспечивают эффективное взаимодействие. Целочисленные значения служат аргументами для шумовой функции.

Для реализации эффекта Перлина предлагается построить ряд функций, в которых основными параметрами будут частота и амплитуда. Амплитуда здесь представляет значения локальных экстремумов, в то время как частота обратно пропорциональна длине волны. Начинаем с построения функции с минимальной выбранной частотой и максимальной амплитудой. Далее, для каждой последующей функции постепенно увеличиваем частоту и уменьшаем амплитуду. Последняя функция в последовательности будет иметь максимальную частоту и минимальную амплитуду. Каждая из этих функций называется октавой. Затем проводим операцию сложения всех полученных функций. Таким образом, создается функция, обладающая гладкостью, при этом сохраняя достаточную степень хаотичности [6].

Алгоритм работы показан на рисунке 1.1

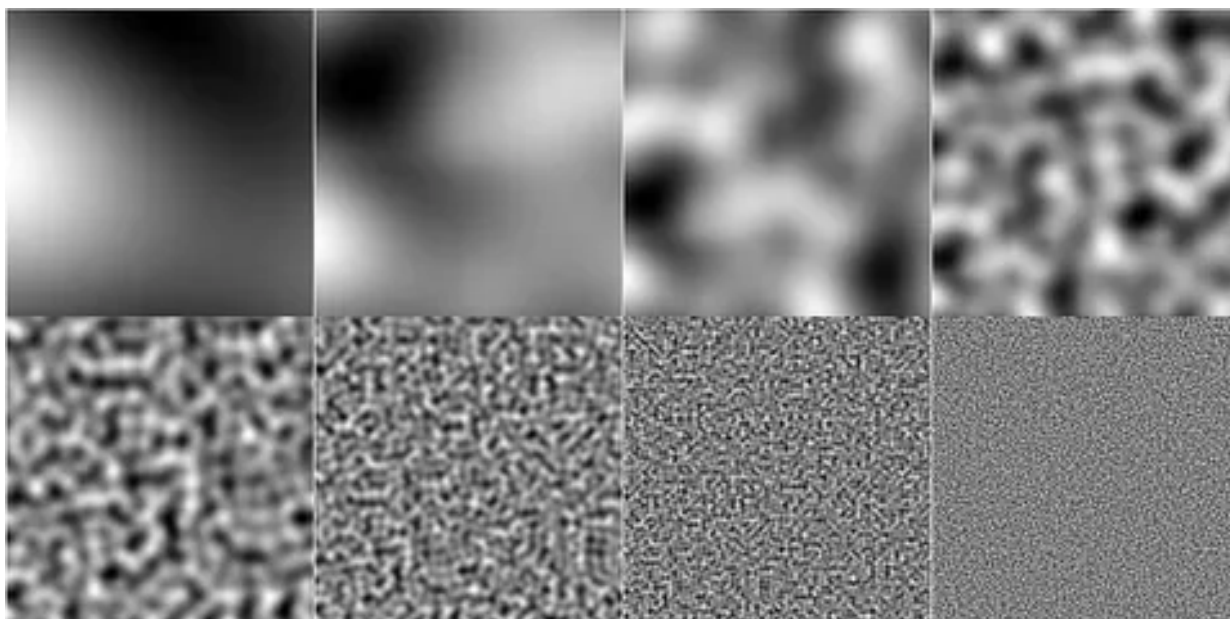


Рисунок 1.1. Алгоритм работы шума Перлина для текстуры облака

Также возможно реализовать алгоритм, который, используя вышеописанные функции, генерирует изображение, соответствующее окончательной функции. С применением шума Перлина можно формировать двумерные карты облаков. Для контроля плотности облаков достаточно установить все пиксели, значения которых меньше определенного порога, в цвет фона. Иллюстрацию этого процесса можно увидеть на рисунке 1.2, где продемонстрирован результат применения данного отсечения.

Преимущество данного подхода перед предыдущим заключается в отсутствии необходимости выделять дополнительную память для текстур. Реалистичность изображения зависит от количества использованных октав.

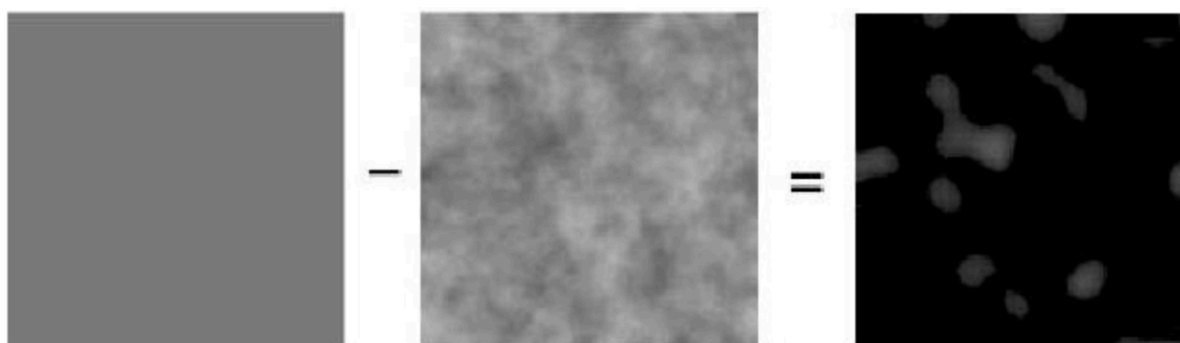


Рисунок 1.2. Контроль над плотностью облаков путем удаления определенной части пикселей, у которых значения минимальны.

Облака могут динамично двигаться относительно наблюдателя, проходить процессы исчезновения и появления. Тем не менее, в данном случае также используется двумерная текстура, и, подобно предыдущему методу, невозможно создать трехмерное изображение облака с возможностью его рассмотрения с разных сторон.

1.2.4. Трёхмерное моделирование облаков

Трёхмерное моделирование облаков предоставляет более реалистичное и физически точное изображение, способное визуализировать широкий спектр явлений, происходящих в облаках. В отличие от полигональных моделей, структура облаков не может быть представлена непосредственно, что создает трудности в их трехмерном отображении. Альтернативой является представление облаков как набора частиц с текстурами прозрачности. Такой подход сохраняет физическую структуру облака, определяемую перекрытием полупрозрачных частиц.

Модель облачной поверхности, охватывающей небесное пространство в определенном диапазоне, представляет собой трехмерную решетку с вокселями¹ в узлах. Создание частиц облака может быть реализовано с использованием шума Перлина, где результаты трехмерного шума служат основой для формирования объемных облаков. Вариант реализации на основе двумерной карты высот, полученной из двумерного шума Перлина, также представляет интерес, где трехмерная структура облака строится на основе перемещения по карте высот, обеспечивая динамичные эффекты.

Шум Перлина не ограничен размерностью, что дает возможность использовать его результаты для формирования объемных облаков с управляемой плотностью, клубящимися эффектами и динамичным появлением. Однако данному методу присущи проблемы в затратах ресурсов

¹ Воксел - элемент объёмного изображения, содержащий значение элемента раstra в трёхмерном пространстве. Воксели являются аналогами двумерных пикселей для трёхмерного пространства.

на вычисления и отображение облаков, особенно при необходимости динамической картины неба в реальном времени. [4]

Важным моментом является альтернативный метод, основанный на двумерной карте высот, где хранение трехмерной решетки не требуется, и моделирование облаков происходит с использованием только карты высот. Этот подход требует тщательного создания карты высот, поскольку она определяет форму и структуру облачности, влияя на реалистичность изображения. Однако, преимуществом является возможность реализации эффектов, таких как изменение формы облака и воздействие ветра, за счет сдвига заполненных вокселей. [5]

Оба метода имеют свои преимущества и недостатки. Трехмерное моделирование более универсально, но требует больших вычислительных ресурсов, тогда как метод на основе карты высот более эффективен в плане ресурсов, но требует более тщательной предварительной подготовки карты высот. [3]

1.3.Алгоритмы визуализации облаков

Из предыдущего раздела видно, что подходы к моделированию облаков существенно отличаются от тех, которые применяются для создания других объектов. В техниках, основанных на поверхностном представлении объектов, начально формируется промежуточная модель, используя плоские треугольники, перед тем как производится их визуализация. В отличие от этого, методы, основанные на воксельном представлении объемов, создают трехмерное изображение объекта непосредственно из объемных данных. Объемно-ориентированный подход к визуализации существенно отличается от традиционной растровой полигональной графики. В растровой графике объекты определяются поверхностями, представленными полигонами, применяемыми к проволочной структуре модели. Это достаточно для приложений в игровой индустрии, анимационных и синтетических объектов,

но является недостаточным для отображения внутренней структуры естественных объектов или явлений. [3]

1.3.1. Использование билбордов

Билборд – это полигон, ориентированный постоянно в направлении наблюдателя. В данном методе облачная поверхность представляет собой трехмерную сетку с билбордами в ее узлах. С использованием шума Перлина и объединения октав шума формируется трехмерная карта облаков. Позиции билбордов определяются на основе этой карты, а затем к каждому билборду применяется текстура, соответствующая определенной части облака. Для правильного отображения облачной поверхности необходимо осуществлять сортировку билбордов от самого дальнего к самому ближнему относительно положения камеры. Также возможны оптимизации, такие как разделение облачной поверхности на сектора и проверка, какие из них попадают в поле зрения камеры. Видимые сектора сортируются от дальнего к ближнему, а невидимые отсекаются. Необходимо учитывать полностью скрытые сектора за другими. Для всех видимых билбордов в секторах рассчитывается цвет с учетом источников света, плотности и рассеяния. Такой метод обеспечивает высокую степень реализма и позволяет работать в реальном времени. [1]

1.3.2. Визуализация используя воксели

Отличие этого подхода от предыдущего заключается в замене билбордов на воксели. Таким образом, небо представляется трехмерной сеткой, в узлах которой размещены воксели. С использованием шума Перлина формируется трехмерная карта облаков, на основе которой определяются позиции вокселей. При использовании трассировки лучей света рассчитывается цвет каждого вокселя с учетом источников света, плотности облаков и рассеяния света. Этот метод обеспечивает более детализированный результат по сравнению с билбордами, однако требует больше ресурсов и сложнее в оптимизации. [1]

Вводятся три битовых параметра для каждого вокселя, отражающих его физическую природу как часть облака:

- *hum* (влажность) – 1 указывает, что в вокселе достаточно пара для формирования облака;
- *act* (активность) – 1 указывает, что в вокселе началась фаза преобразования из пара в воду, то есть в облако;
- *cld* (облако) – 1 указывает на наличие облака в вокселе.

Работая с этими параметрами, можно изменять структуры облачной поверхности, реализуя эффекты, такие как зарождение, угасание, перемещение и изменение структуры облаков [2,5]. Графическое представление данного метода представлено на рисунке 1.3.

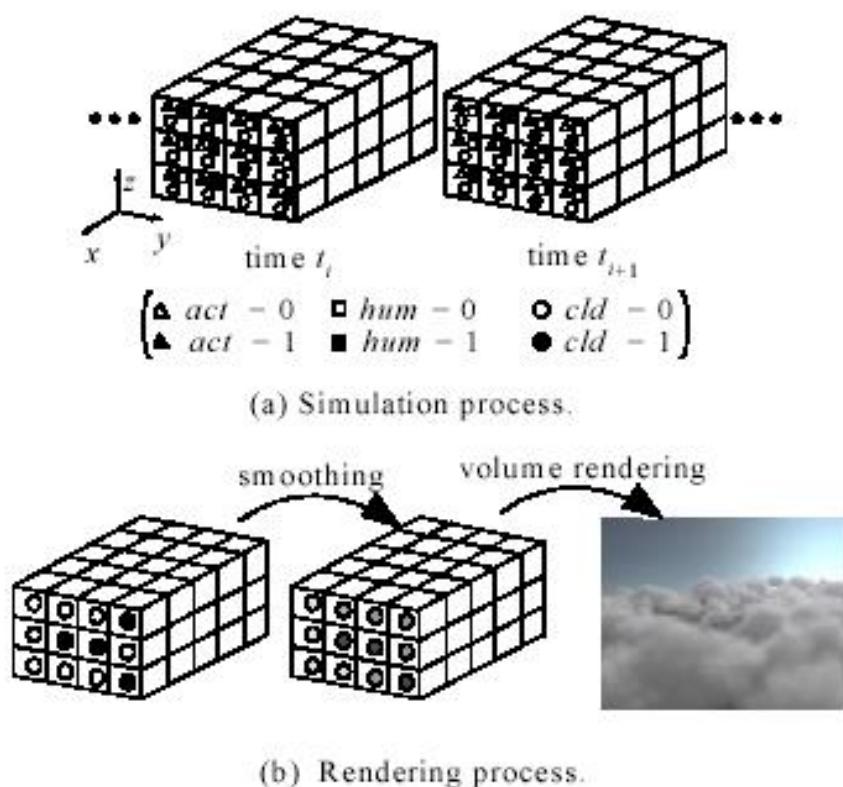


Рисунок 1.3. Визуализация облаков на основе вокселей и трёх битовых полей.

1.3.3. Перспективная проекция

Значимым компонентом в композиции сцены является точка обзора, или камера. Камера играет ключевую роль в создании реалистичного изображения объектов с использованием принципов проецирования. Важными характеристиками для настройки камеры являются точка обзора, угол обзора, размеры окна просмотра и расстояние до окна просмотра. С использованием этих параметров можно получить перспективное изображение, максимально приближенное к реальности. Так, объекты, находящиеся в дали, визуальнo представлены меньшими, в то время как близкие объекты кажутся большими. Пример проецирования точки на экран камеры представлен на рисунке 1.4.

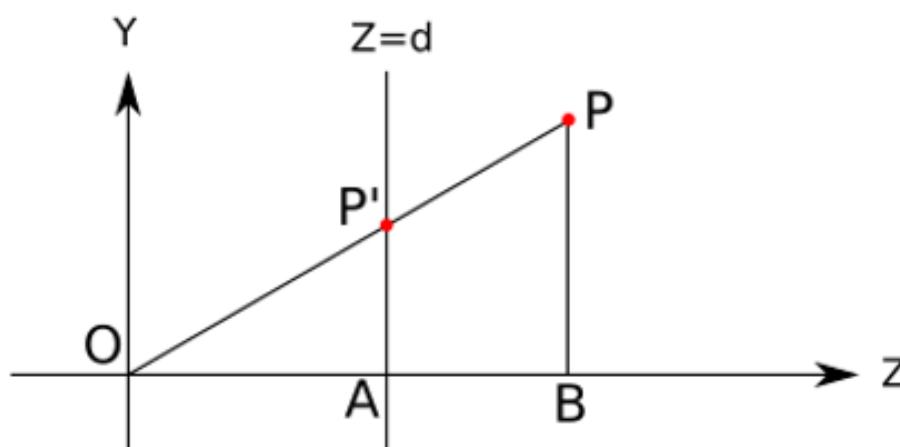


Рисунок 1.4. Проецирование точки в окно просмотра

Этот метод применения перспективы также применим к изображению облаков. Таким образом, можно демонстрировать приближение к облакам и даже реализовывать эффект пролета сквозь них. Точки, находящиеся вне области видимости камеры (за её пределами или не проецирующиеся в окно просмотра), отсекаются, что создаёт возможность наблюдать «внутренние» части облачного слоя.

1.4.Обоснование выбора метода моделирования и алгоритма визуализации

Для реализации данного проекта был выбран метод визуализации облаков, основанный на трехмерной решетке вокселей. Этот подход

обеспечивает высококачественные и реалистичные изображения, позволяя рассматривать облака со всех сторон и воплощать эффект пролета сквозь них, а также другие визуальные эффекты. Этот метод может быть успешно применен в различных приложениях.

Методы, основанные на 2D текстурах, не обеспечивают достаточно реалистичного трехмерного изображения, что делает их неприменимыми.

Для создания таблицы вокселей будет использован трехмерный шум Перлина, поскольку ручное задание значений вокселей требует слишком много усилий. Таким образом, облака будут генерироваться процедурно, а использование шума Перлина позволит настраивать их плотность.

Для визуализации будет применен метод билбордов, где каждая частица облака будет представлена как небольшая текстура, всегда обращенная к наблюдателю. Кроме того, будет реализована камера с использованием перспективы для создания наиболее реалистичной сцены и возможности пролета сквозь облака.

2. Конструкторский раздел

В данном разделе представлен алгоритм для решения поставленной задачи, а также представлены типы и структуры данных, которые используются, и разработана структура программы.

2.1. Алгоритм работы приложения

- 1) Ввод пользовательских данных
- 2) Создание сцены
- 3) Создание трёхмерной сетки на основе шума Перлина
- 4) Применение пользовательских настроек
- 5) Преобразования сцены: поворот, перемещение и масштабирование
- 6) Преобразование к координатам камеры
- 7) Визуализация полученной сцены

2.2. Разработка алгоритмов

2.2.1. Шум Перлина

Шум Перлина — это градиентный шум, состоящий из набора псевдослучайных единичных векторов (направлений градиента), расположенных в определенных точках пространства и интерполированных функцией сглаживания между этими точками. Для генерации шума Перлина в одномерном пространстве необходимо для каждой точки этого пространства вычислить значение шумовой функции, используя направление градиента (или наклон) в указанной точке.

Функция шума Перлина является результатом сложения нескольких шумовых функций с разными значениями амплитуды и частоты. Для построения данных функций введены следующие величины:

- seed – случайная величина, используемая в шумовой функции
- persistence – величина, обеспечивающая зависимость амплитуды от частоты
- octaves – количество октав шума

Октава – каждая выполненная функция шума. Количество октав, также, является основной величиной в данном процессе и влияет на качество изображения.

Эти величины соотносятся между собой по следующим формулам:

$$frequency^2 = 2^i, (2.1)$$

$$amplitude^3 = persistence^i, (2.2)$$

где i – номер генерируемой функции.

Основой для данной функции является шумовая функция, параметры которой определяются значением *seed*. На вход этой функции поступают целые числа x, y, z , а на выходе формируется значение в пределах от -1 до 1.

После этого применяется интерполяция. На этом этапе осуществляется сглаживание функции, где значение функции вычисляется с использованием следующей формулы:

$$y = a * \left(1 - \frac{1 - \cos(x * \Pi)}{2}\right) + b * \frac{1 - \cos(x * \Pi)}{2}, (2.3)$$

где y – искомое значение функции; a – начало отрезка; b – конец отрезка; x – точка, в которой необходимо найти значение.

Эти шаги повторяются для каждой октавы, после чего полученные результаты суммируются для формирования итоговой функции шума Перлина.

² Frequency - частота

³ Amplitude - амплитуда

Схема алгоритма шума Перлина показана на рисунке 2.1.

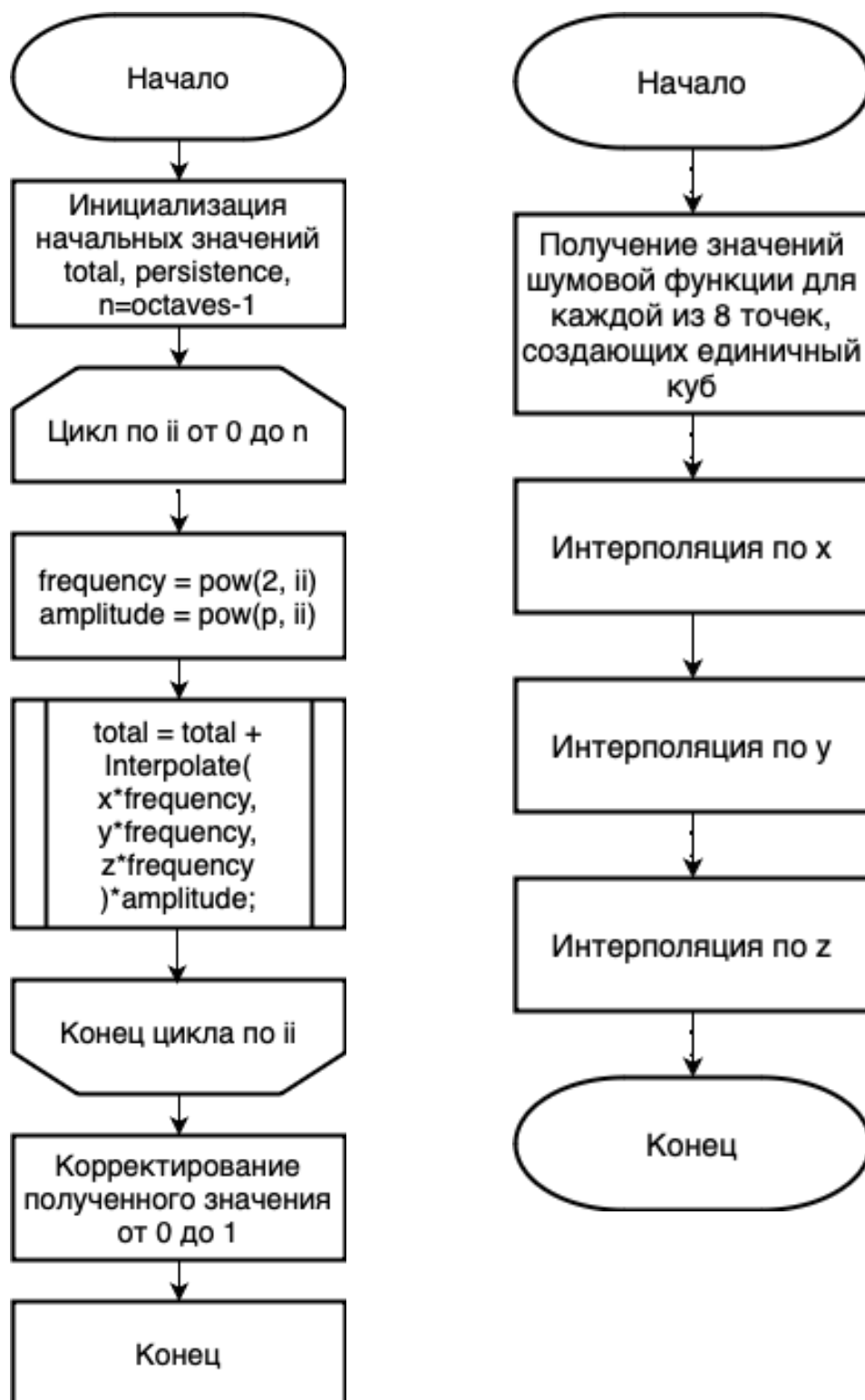


Рисунок 2.1 Схема получения шума Перлина

2.2.2. Создание воксельной сетки

Используя функцию Перлина можно создать трёхмерную сетку. Для каждого узла сетки требуется вычислить значение функции шума Перлина. Это значение будет использоваться для определения прозрачности соответствующего вокселя.

Для заполнения сетки достаточно обойти все её элементы и присвоить каждому вокселю новое значение прозрачности, полученное с использованием функции Перлина. Однако при такой реализации многие воксели получают значения близкие или равные нулю, что может замедлить работу системы при визуализации сцены. Для повышения производительности предлагается использовать кеш, в котором будут храниться все значимые точки и их цвета. Стандартный цвет каждого вокселя установлен как белый. Во время записи в кеш также можно применить пользовательские настройки и отсеять точки, не соответствующие установленному значению плотности.

2.2.3. Преобразования камеры

Для того чтобы иметь возможность наблюдать сцену с различных ракурсов, необходимо осуществить изменения в параметрах камеры, такие как масштабирование, поворот и перемещение. Для проведения матричных преобразований перейдем к использованию однородных координат $[x, y, z, 1]$. В этой форме точки можно модифицировать, умножая их на матрицу преобразования размером 4×4 .

$$\begin{aligned} \text{Трёхмерный перенос реализуется матрицей } T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{bmatrix} \\ \text{Частичные изменения реализуются матрицей } S &= \begin{bmatrix} kx & 0 & 0 & 0 \\ 0 & ky & 0 & 0 \\ 0 & 0 & kz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Однако умножение на эту матрицу дает масштабирование только относительно начала координат. Более полное масштабирование реализуется с помощью формул:

$$x = Cx + kx * (x - Cx), (2.4)$$

$$y = Cy + ky * (y - Cy), (2.5)$$

$$z = Cz + kz * (z - Cz), (2.6)$$

где $C(Cx, Cy, Cz)$ — центр масштабирования.

Поворот вокруг оси Z описывается матрицей

$$Rz = \begin{bmatrix} \cos(a) & \sin(a) & 0 & 0 \\ -\sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Поворот вокруг оси X описывается матрицей

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & \sin(a) & 0 \\ 0 & -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Поворот вокруг оси Y описывается матрицей

$$Ry = \begin{bmatrix} \cos(a) & 0 & -\sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Применяя эти формулы, можно двигать, поворачивать и масштабировать изображение.

3. Технологический раздел

В данном разделе будут представлены средства разработки программного обеспечения, интерфейс программного обеспечения и примеры работы

3.1.Выбор средств реализации

Для написания данного курсового проекта был выбран язык C++.

Выбор языка программирования обосновывается следующим образом:

1. Преимущества объектно-ориентированной модели разработки, которая обеспечивает структурирование программы и эффективное создание высококачественного программного обеспечения.

2. Эффективное использование ресурсов системы благодаря обширному набору функций и классов из стандартной библиотеки, что способствует оптимальной работе программ.

3. Высокие показатели вычислительной производительности, особенно важные для задач генерации реалистичных изображений, делают язык программирования C++ неотъемлемым инструментом.

Для разработки использовалась среда Qt Creator, обладающая всем необходимым функционалом для написания, профилирования и отладки программ, а также создания графического пользовательского интерфейса. Эта среда поставляется с фреймворком Qt, который включает все необходимые средства для работы с пикселями изображения. Для упрощения процесса сборки проекта программного обеспечения использовалась утилита qmake.

3.2.Структура приложения

Реализованы следующие классы:

- 1) MainWindow – класс главного окна сцены
- 2) Cloud – класс для генерации облака
- 3) Scene – класс управления объектами сцены
- 4) Camera – класс камеры
- 5) Nois – класс для подсчёта функции шума Перлина
- 6) Voxel – структура вокселя и его преобразование
- 7) Voxelgrid – воксельная трёхмерная решётка

8) Vector3 – класс векторов трёхмерного пространства и его преобразования

9) Matrix – класс матриц

3.3.Интерфейс приложения

На рисунке 3.1 представлен уникальный пользовательский интерфейс, предлагающий возможность ввода индивидуальных параметров сетки, установки плотности и запуска различных примеров. Структура окна включает в себя: основной экран для отображения изображения, боковую панель с настройками справа и выпадающее меню в верхней части интерфейса.

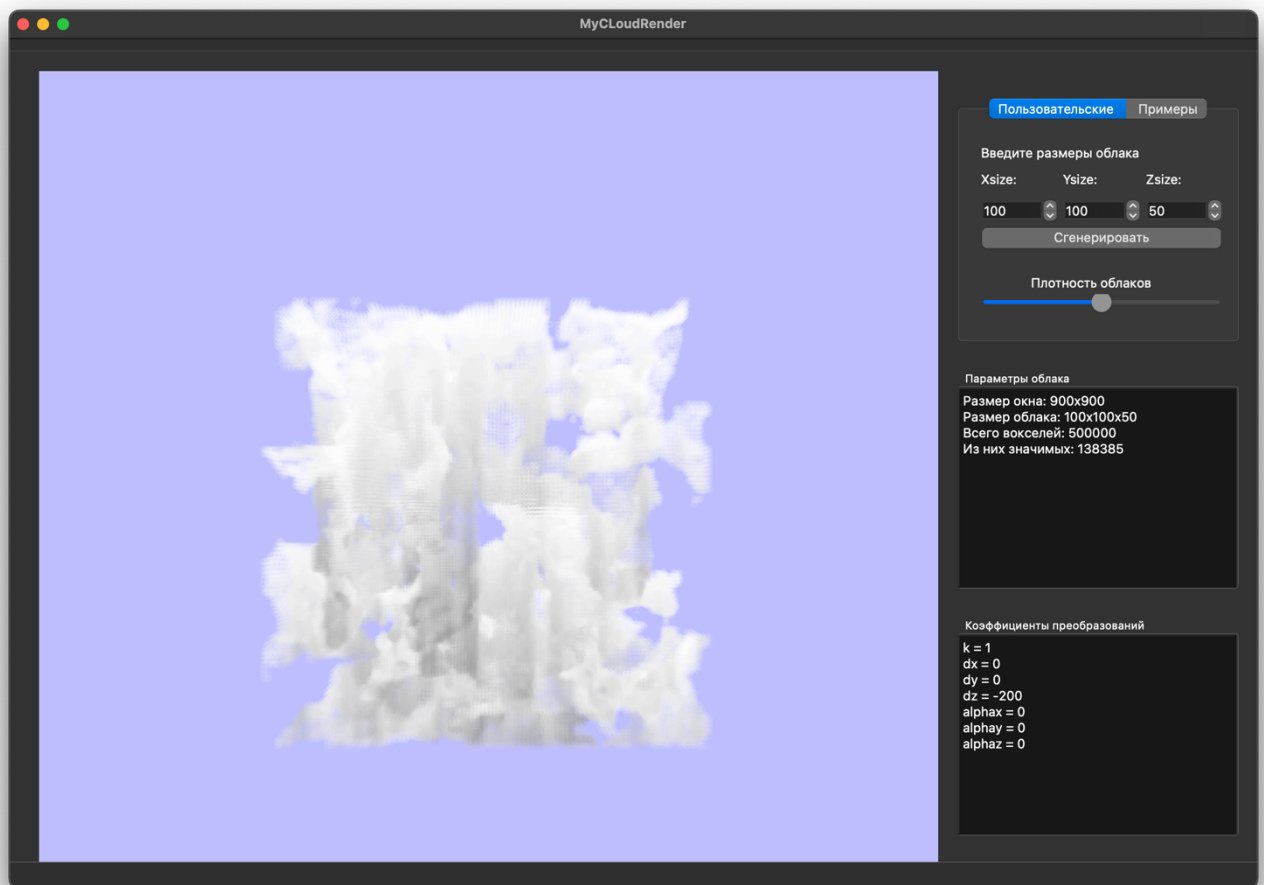


Рисунок 3.1 Интерфейс пользователя

Значения Xsize, Ysize, Zsize могут быть в диапазоне от 1 до 200. Значения плотности задаются с помощью ползунка.

В окне «Параметры облака» выводятся сведения о размере окна, размере сетки, количестве вокселей и количестве значимых точек. В окне

«Коэффициенты преобразований» содержится информация о текущих преобразованиях, совершённых над объектом.

Управление используя клавиатуру:

Клавиши W/S предназначены для вращения облака вокруг оси OX.

Клавиши A/D предназначены для вращения облака вокруг оси OY.

Клавиши Z/E предназначены для вращения облака вокруг оси OZ.

Клавиши H/K предназначены для перемещения облака по оси OX.

Клавиши J/U предназначены для перемещения облака по оси OY.

Клавиши N/I предназначены для перемещения облака по оси OZ.

Клавиши +/- предназначены для масштабирования облака относительно начала координат.

4. Исследовательский раздел

В данном разделе приведены технические характеристики устройства, на котором проводилось измерение времени работы программного обеспечения, а также результаты замеров времени.

4.1. Постановка задачи анализа

Целью является анализ скорости работы алгоритма генерации изображения.

4.2. Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование.

- Процессор: Apple M1 Pro 3.2 ГГц.
- Количество ядер: 10 CPU и 14 GPU.
- Оперативная память: 16 Гбайт.
- Операционная система: macOS Sonoma 14.1.1.

Во время тестирования устройство было нагружено только встроенными приложениями окружения, а также непосредственно системой тестирования.

4.3. Результаты анализа

Для анализа зависимости времени генерации изображения от количества узлов сетки и времени визуализации сцены от количества значимых точек, использовались сетки кубические сетки размером от 10x10x10 до 150x150x150 с шагом 10.

Результаты анализа представлены в таблице 1.

Таблица 1 – результаты анализа по замеру времени

Количество узлов сетки	Время генерации облака	Количество значимых точек	Время визуализации сцены
1000	976	134	697
8000	7069	1659	2189
27000	23937	6054	6878

64000	56066	12700	16411
125000	110788	41416	51468
216000	195200	43715	65859
343000	316762	111129	151495
512000	451136	135730	182845
729000	675646	225540	330993
1000000	884671	257572	384726
1331000	1247122	359026	510818
1728000	1553268	494453	723323
2197000	2118391	620860	905963
2744000	2558966	802007	1139844

На рисунке 4.1 представлен график зависимости времени генерации облака от размера сетки.



Рисунок 4.1 График зависимости времени генерации облака от количества узлов сетки

Видно, что образуется линейная зависимость. Это означает, что чем больше размер сетки, тем дольше будет генерироваться облако.

На рисунке 4.2 представлен график зависимости времени отображения сцены от количества значимых точек.

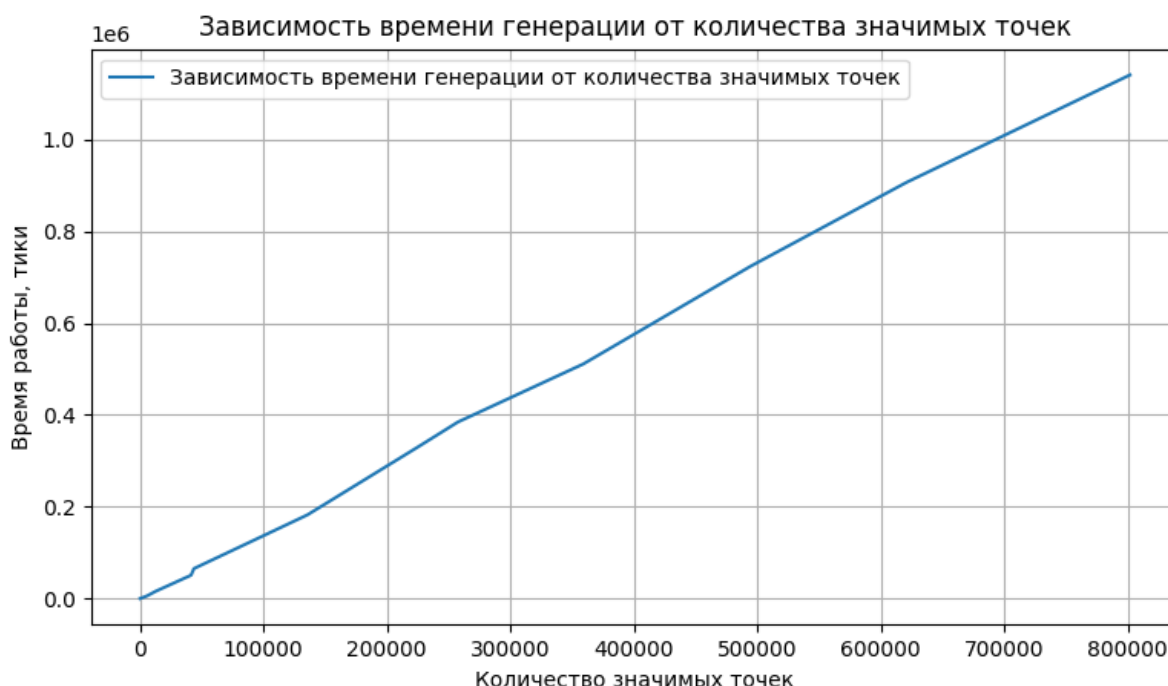


Рисунок 4.2 График зависимости отображения сцены от количества значимых точек

Этот график тоже линеен.

Из проведенных замеров можно сделать вывод о линейности работы алгоритмов. Скорость формирования облака непосредственно зависит от размера используемой воксельной сетки. Время отображения сцены, в свою очередь, зависит от количества значимых точек, изменяющегося в зависимости от плотности облака. Таким образом, большие облака с небольшим числом значимых точек могут быть визуализированы быстрее, чем маленькие облака с большим количеством таких точек.

4.4.Примеры визуализации программы

На рисунках 4.3 – 4.4 представлены примеры облаков с разной ПЛОТНОСТЬЮ.

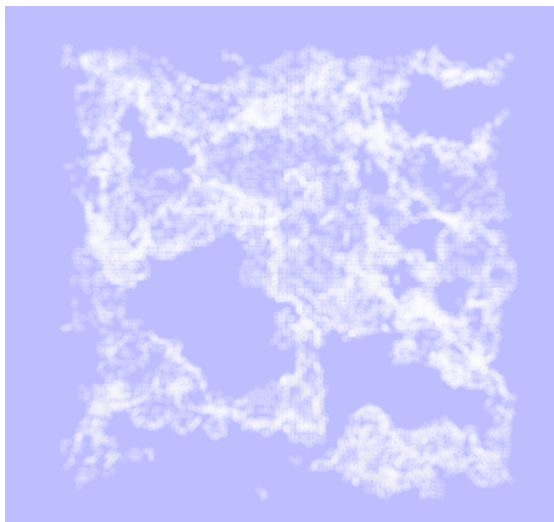


Рисунок 4.6 облако малой
ПЛОТНОСТИ

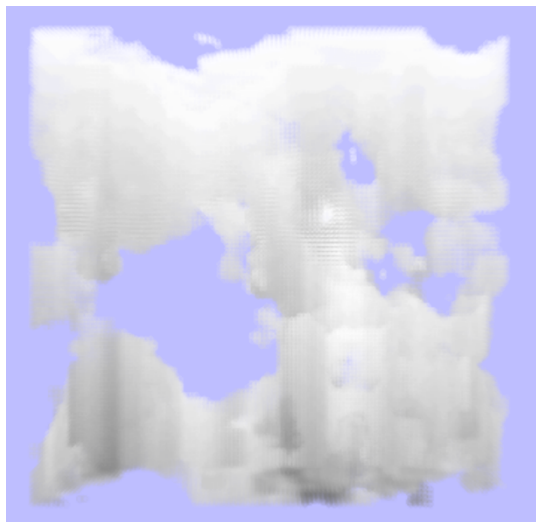


Рисунок 4.7 облако высокой
ПЛОТНОСТИ

ЗАКЛЮЧЕНИЕ

В рамках выполнения курсового проекта было разработано программное обеспечение, предназначенное для визуализации реалистичных изображений облаков. Для этого была применена трехмерная сетка, на которой размещены билборды – объекты, всегда ориентированные к пользователю. Для достижения естественного облачного эффекта использовался трехмерный шум Перлина.

В процессе исследования были рассмотрены методы процедурной генерации облаков, а также различные подходы к их визуализации и алгоритмы отрисовки сцены. Разработанное программное обеспечение позволяет генерировать облака с разной плотностью, а также осуществлять поворот, перемещение и масштабирование облачного слоя. Кроме того, предусмотрена возможность пролета сквозь облака.

В ходе исследования были сделаны следующие выводы:

1. Методы генерации облаков существенно отличаются от традиционных методов растровой полигональной графики, что делает неприменимыми многие алгоритмы, применяемые к таким объектам.

2. Применение трехмерной модели представления облаков с использованием небольших частиц позволяет реализовывать различные естественные эффекты, включая пролет сквозь облака. Это особенно актуально для авиасимуляторов, где реалистичный пролет через облака является важным элементом.

3. Время генерации облачного слоя линейно зависит от количества узлов сетки. Тем не менее, время визуализации сцены зависит от количества значимых точек, и может оставаться невеликим даже для обширных сеток.

СПИСОК ЛИТЕРАТУРЫ

1. Агафонов, Н.А. МЕТОДЫ ВИЗУАЛИЗАЦИИ ПОГОДНЫХ ЯВЛЕНИЙ В ИМИТАЦИОННЫХ СИСТЕМАХ / Н. А. Агафонов, А. М. Гиацинтов, А. В. Родителей // Федеральный научный центр Научно-исследовательский институт системных исследований РАН. Вестник кибернетики. - 2018.
2. Аксенов, Андрей. Компьютерная графика.
URL: <http://algolist.ru/graphics/3dfaq/index.php>
3. Вяткин, С.И. ВИЗУАЛИЗАЦИЯ ПОЛУПРОЗРАЧНЫХ ОБЪЕКТОВ НА БАЗЕ ФУНКЦИЙ ВОЗМУЩЕНИЯ И ПРОЗРАЧНОСТИ / С.И. Вяткин, Б.С. Долговесов // Российская академия наук, Сибирское отделение, Автометрия. - Новосибирск, 2005. - Том 41, №3.
4. Елыков, Н.А. Практическая модель динамических атмосферных явлений при визуализации открытых пространств в системах визуализации реального времени / Н.А.Елыков, И.В.Белаго, С.М.Козлов С.А.Кузиковский, М.М.Лаврентьев // Лаборатория программных систем машинной графики ИАиЭ СО РАН. - Новосибирск, Россия.
5. Мальковский, Александр. Визуализация неба и облаков.//Введение в компьютерную графику. Полугодовой курс ВМиК МГУ, 2002.
URL:https://www.graphicon.ru/oldgr/courses/cg02b/assigns/hw-5/hw5_cld.htm
6. Слеповичев, Сергей Олегович. ШУМ ПЕРЛИНА КАК СПОСОБ ПОЛУЧЕНИЯ КОМПЬЮТЕРНЫХ СПЕЦЭФФЕКТОВ ПРИРОДНЫХ ЯВЛЕНИЙ // Инновационное развитие ИП Сигитов Т.М. - Пермь, 2017.
7. Dobashi, Yoshinori. A Simple, Efficient Method for Realistic Animation of Clouds / Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, Tomoyuki Nishita // Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA — 2000. - Pages 19-28.
8. Wang, Niniane. Realistic and Fast Cloud Rendering / Niniane Wang, Microsoft Corporation - November 11, 2003.
9. И.А. Ботыгин, В.С Шерстнёв, А.Н. Шерстнёва. Программные средства построения мезомасштабных атмосферных моделей.

ПРИЛОЖЕНИЕ А