МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Анализ алгоритмов сплайн-интерполяции и их применение при разработке мобильного приложения

Обучающийся И.А. Шатилов
(И.О. Фамилия) (личная подпись)

Руководитель М.А. Тренина
(ученая степень, звание, И.О. Фамилия)

Консультант к.п.н., доцент О.Н. Брега
(ученая степень, звание, И.О. Фамилия)

Тольятти 2023

Аннотация

Тема выпускной квалификационной работы – «Анализ алгоритмов сплайн-интерполяции и их применение при разработке мобильного приложения».

Работа посвящена анализу алгоритмов сплайн-интерполяции и их применение при разработке мобильного приложения. В данной работе рассматривается алгоритмы сплайн-интерполяции, основанных в реализации мобильного приложения.

Объект работы – алгоритмы сплайн-интерполяции.

Предмет работы – использование сплайн интерполяции в разработке мобильного приложения.

Цель работы – проанализировать алгоритмы сплайн-интерполяции и применить их в разработке мобильного приложения.

Данная работа состоит из ведения, трех глав, заключения и списка литературы.

Во введении определяются актуальность темы, цель и задачи, поставленные в работе.

В первой главе работы описываются различные формулы, алгоритмы сплайн-интерполяции, их преимущества и недостатки.

Во второй главе описывается мобильное приложение с использованием алгоритмов сплайн-интерполяции.

Третья глава посвящена оценке эффективности разработанного программного обеспечения.

Выпускная квалификационная работа содержит пояснительную записку объемом 56 страниц, 7 рисунков и список используемой литературы, состоящий из 25 источников.

Abstract

The topic of the final qualifying work is "Analysis of spline interpolation algorithms and their application in the development of useful applications".

The paper is devoted to the analysis of spline interpolation algorithms and their application in the development of mobile applications. In this paper, spline interpolation algorithms based on the implementation of mobile applications are used.

The object of the work is the algorithms of spline interpolation.

The subject of the work is the use of spline interpolation in the development of mobile applications.

The purpose of the work is to analyze the algorithms of spline interpolation and their application in the development of mobile applications.

This work consists of a study, three chapters, a review and a bibliography. Currently, the current topic, purpose and tasks set in the work are being determined.

The first paper presents various formulas, algorithms of spline interpolation, their advantages and disadvantages.

The second stage is an introduction to a mobile application using spline interpolation algorithms.

The third chapter is devoted to the analysis of the results of software calculations.

The final qualifying work contains an explanatory note of 56 pages, 7 figures and a list of references collected from 25 sources.

Оглавление

Введение

Алгоритмы сплайновой интерполяции являются фундаментальным инструментом для аппроксимации функций на основе набора разбросанных точек данных. Они используются в широком спектре приложений, включая обработку изображений, компьютерную графику и инженерное проектирование. В последние годы мобильные приложения стали неотъемлемой частью нашей повседневной жизни, и их развитие росло в геометрической прогрессии. Сплайновая интерполяция может быть использована для повышения точности и качества визуализации данных в мобильных приложениях. В данной работе проанализируем различные алгоритмы интерполяции сплайнов, включая кубические и квадратичные сплайны, и оценим их эффективность при разработке мобильного приложения. Сравним точность, вычислительную эффективность, гибкость этих алгоритмов и их влияние на общую производительность мобильного приложения. Кроме того, внедрим мобильное приложение, которое использует сплайновую интерполяцию для визуализации данных и демонстрации эффективности этих алгоритмов на практике. Целью данной работы является анализ алгоритмов сплайновой интерполяции и их применения при разработке мобильных приложений, дающий представление о потенциале этих методов для повышения качества и точности визуализации данных на мобильных платформах.

Бакалаврская работа будет разделена на несколько глав, начиная с анализа алгоритмов сплайновой интерполяции, их свойствах и приложениях. В следующей главе будет описана их реализация в мобильном приложение и методы, используемые для интеграции алгоритмов сплайновой интерполяции в процесс визуализации данных. Затем проанализируем производительность различных алгоритмов сплайновой интерполяции с точки зрения точности и вычислительной эффективности.

В следующей главе представим результаты нашей оценки алгоритмов сплайновой интерполяции и обсудим их сильные и слабые стороны. Также сравним производительность мобильного приложения, основанного на интерполяции сплайнов.

В заключительной главе обобщим наши результаты и сделаем выводы относительно эффективности алгоритмов сплайновой интерполяции при разработке мобильных приложений.

Целью данной работы является всесторонний анализ алгоритмов сплайновой интерполяции и их применения при разработке мобильных приложений. Результаты этой работы могут помочь повысить точность и качество визуализации данных в мобильных приложениях, что приведет к улучшению пользовательского опыта и повышению эффективности процессов принятия решений.

Задачей данной работы является анализ алгоритмов сплайн-интерполяции, применение этих алгоритмов при разработке мобильного приложения, сравнение данных алгоритмов и их тестирование в мобильном приложении.

Выпускная квалификационная работа содержит пояснительную записку объемом 56 страниц, 7 рисунков и список используемой литературы, состоящий из 25 источников.

Глава 1 Анализ алгоритмов сплайн-интерполяции

1.1 Определение сплайн-интерполяции, их преимущества и недостатки

Сплайн – функция, которая вместе с несколькими производными непрерывна на всем заданном отрезке $[a, b]$, а на каждом частичном отрезке $[x_i, x_{i+1}]$ в отдельности является некоторым алгебраическим многочленом [1].

Степенью сплайна называется максимальная по всем частичным отрезкам степень многочленов, а дефектом сплайна – разность между степенью сплайна и порядком наивысшей непрерывной на $[a, b]$ производной. Например, непрерывная ломанная является сплайном степени 1 с дефектом 1 (так как сама функция – непрерывна, а первая производная уже разрывна).

Интерполяция – способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений [2].

Сплайновая интерполяция – это метод построения гладкой кривой, проходящей через набор точек данных. В частности, учитывая набор из $n + 1$ точек данных $(x_0, y_0), (x_1, y_1) \dots, (x_n, y_n)$, сплайновая интерполяция включает в себя построение кусочно-определенной функции $S(x)$, которая удовлетворяет трем условиям [5].

Во-первых, $S(x)$ непрерывна на интервале $[x_0, x_n]$. Во-вторых, $S(x)$ дважды дифференцируемо, что означает, что оно имеет непрерывные первую и вторую производные на интервале $[x_0, x_n]$. В-третьих, $S(x)$ проходит через каждую из точек данных, что означает $S(x_i) = y_i$ для $i = 0, 1, \dots, n$.

Чтобы построить сплайн-функцию $S(x)$, интервал $[x_0, x_n]$ делится на n под-интервалов, и для каждого под-интервала определяется кубическая полиномиальная функция. Затем эти полиномиальные функции соединяются в точках данных для создания непрерывной и плавной кривой, проходящей через все точки данных [6].

Результирующая кривая называется кубическим сплайном, и она предоставляет способ интерполяции или оценки значений между заданными точками данных. Этот метод обычно используется во многих областях, включая математику, инженерное дело и компьютерную графику, для представления данных и создания плавных кривых.

Метод сплайновой интерполяции часто предпочтительнее других методов интерполяции, поскольку он создает плавную и непрерывную кривую, проходящую через все точки данных. Это облегчает анализ и интерпретацию данных, особенно когда данные зашумлены или содержат выбросы.

Существуют различные типы сплайновых функций, которые можно использовать для интерполяции, такие как линейные сплайны, квадратные сплайны и кубические сплайны. Однако кубические сплайны являются наиболее часто используемым типом, поскольку они обеспечивают хороший баланс между точностью и простотой.

Процесс построения кубического сплайна включает в себя решение системы линейных уравнений для определения коэффициентов кубических полиномиальных функций. Эта система уравнений может быть решена с использованием различных методов, таких как исключение Гаусса или инверсия матрицы.

Как только кубический сплайн построен, его можно вычислить в любой точке интервала $[x_0, x_n]$ для оценки значения функции. Точность оценки зависит от расстояния между заданными точками данных и оцениваемой точкой.

Таким образом, сплайновая интерполяция – это мощный метод построения гладкой кривой, проходящей через набор точек данных. Он широко используется в различных областях для представления данных, создания плавных кривых и оценки значений между заданными точками данных.

К преимуществам сплайн-интерполяции относятся.

Гладкость: сплайн-интерполяция создает гладкую кривую, проходящую через заданные контрольные точки. Это делает его идеальным для аппроксимации функций, которые имеют резкие изменения или разрывы.

Гибкость: сплайн-интерполяция может использоваться с различными типами сплайн-функций, такими как естественные, фиксированные и периодические сплайны, в зависимости от требований задачи.

Точность интерполяции: сплайн-интерполяция обычно обеспечивает более точную интерполяцию, чем другие методы интерполяции, такие как линейная или полиномиальная интерполяция.

Вычислительная эффективность: вычислительная стоимость сплайн-интерполяции относительно низка по сравнению с другими методами интерполяции, что делает ее практичной для использования в приложениях реального времени.

К недостаткам сплайновой интерполяции относятся.

Сложность: сплайн-интерполяция требует вычисления полиномиальных функций для каждого интервала между контрольными точками. Это может занять много времени и потребовать значительных ресурсов памяти для больших наборов данных.

Экстраполяция: сплайн-интерполяция не подходит для экстраполяции, так как она может создавать неограниченные или осциллирующие функции за пределами диапазона контрольных точек [9].

Чувствительность к выбору узлов: качество сплайновой интерполяции зависит от размещения контрольных точек или узлов. Неправильный выбор узлов может привести к переоснащению или недостатку обучения данных.

Не подходит для многомерных данных: сплайн-интерполяция может оказаться непригодной для многомерных данных, поскольку количество требуемых интервалов и полиномиальных функций экспоненциально увеличивается с размерностью данных.

1.2 Виды и формулы сплайн-интерполяции

Типы сплайновой интерполяции обычно не классифицируются как кубические, квадратичные или линейные. Вместо этого эти термины относятся к степени полиномиальных функций, используемых для определения сплайна.

Естественные сплайны – это тип кубического сплайна, который накладывает дополнительные ограничения на кривую, чтобы гарантировать, что она гладкая в конечных точках. Алгоритм использует набор естественных граничных условий для обеспечения того, чтобы вторые производные кривой были равны нулю в конечных точках. Это может быть полезно в случаях, когда кривая должна быть плавной и непрерывной по всему диапазону данных.

Интерполяция естественного кубического сплайна. Пусть x_i и x_{i+1} – две последовательные точки данных с соответствующими функциональными значениями y_i и y_{i+1}. Кубическая полиномиальная функция для интервала $[x_i, x_{i+1}]$ задается формулой:

$$f(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \quad (1)$$

Коэффициенты рассчитываются по следующим формулам:

$$a_i = y_i, \quad (2)$$

$$b_i = \frac{M_i}{h_{i-1}}, \quad (3)$$

$$c_i = \frac{3(x_{i+1} - x_i) - 2M_i - 1h_{i-1} - M_i h_{i-1}}{h_i^2}, \quad (4)$$

$$d_i = \frac{M_{i-1} + M_{i-2}(x_{i+1} - x_i)}{h_i^3}, \quad (5)$$

где M_i – вторая производная сплайновой функции в точке x_i.

Естественная интерполяция кубического сплайна требует, чтобы вторая производная была равна нулю в обеих конечных точках.

Интерполяция с зажатым кубическим сплайном. Сжатая интерполяция кубического сплайна аналогична естественной интерполяции кубического сплайна, но она ограничивает первые производные в конечных точках. Пусть x_i и x_{i+1} – две последовательные точки данных с соответствующими функциональными значениями y_i и y_{i+1}. Кубическая полиномиальная функция для интервала $[x_i, x_{i+1}]$ задается формулой:

$$f(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \quad (6)$$

Коэффициенты рассчитываются по следующим формулам:

$$a_i = y_i, \quad (7)$$
$$b_i = \frac{M_i}{h_{i-1}}, \quad (8)$$
$$c_i = \frac{3(x_{i+1} - x_i) - 2M_i - 1h_{i-1} - M_i h_{i-1}}{h_i^2}, \quad (9)$$
$$d_i = \frac{M_{i-1} + M_{i-2}(x_{i+1} - x_i)}{h_i^3}, \quad (10)$$

где M_i – вторая производная сплайновой функции в точке x_i.

Однако первые производные в обеих конечных точках ограничены некоторыми заданными значениями, $f'(x_1) = s_1$ и $f'(x_n) = s_n$.

Сплайны Akima – это тип кубического сплайна, который предназначен для обработки данных со значительным шумом или выбросами. Алгоритм использует локальный метод для оценки наклона кривой в каждой точке данных, что делает его более устойчивым к зашумленным данным. Результирующая кривая может быть не такой гладкой, как стандартный кубический сплайн, но она может быть более точной в случаях, когда данные зашумлены.

Интерполяция сплайнов Akima. Сплайн-интерполяция Akima использует кусочно-определенную кубическую полиномиальную функцию с измененными коэффициентами для обеспечения непрерывности и дифференцируемости. Пусть x_i и x_{i+1} – две последовательные точки данных с соответствующими функциональными значениями y_i и y_{i+1}. Сплайн-функция Akima для интервала $[x_i, x_{i+1}]$ задается формулой:

$$f(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \tag{11}$$

Коэффициенты рассчитываются по следующим формулам:

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i(c_{i+1} + 2c_i)}{3}, \tag{12}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \tag{13}$$

$$c_i = \frac{S_i(y_{i+1} - y_i)}{h_i^2} + S_i + \frac{1(y_{i+2} - y_{i+1})}{h_{i+1}^2}, \tag{14}$$

$$S_i = \frac{y_{i+1} - y_i}{h_i}\text{, для } i = 1, \dots, n - 1 \text{ и } i + 1, \dots, n. \tag{15}$$

B-сплайновая интерполяция – это обобщение алгоритма кубического сплайна, который позволяет более гибко управлять формой кривой. B-сплайны определяются набором контрольных точек и параметром степени, который определяет порядок полиномов, используемых для построения кривой. B-сплайны широко используются в компьютерной графике и приложениях CAD / CAM, поскольку они позволяют легко манипулировать формой кривой [3].

Интерполяция B-сплайна использует кусочно-определенную полиномиальную функцию степени k с $k + 1$ контрольными точками, которые определяют форму кривой. Базисная функция B-сплайна для интервала $[x_i, x_{i+1}]$ задается формулой:

$$N_0, k(x) = 1,\ \text{для } x_i \leq x < x_{i+1}, \tag{16}$$

$$N_i, k(x) = \frac{x - x_i}{x_{i+k-i+1} - x_i} * N_{i-1}, k - 1(x) +$$

$$+ \frac{x_{i+k-i+2} + x}{x_{i+k-i+2} - x_{i+1}} * N_{i-1}, k - 1(x)\text{, для } i = 1, \dots, k - 1. \tag{17}$$

Функция B-сплайна для всей области затем определяется как линейная комбинация этих базисных функций и соответствующих им контрольных точек [11].

Интерполяция сплайнов Catmull-Rom – это тип интерполяции сплайнов Эрмита, который требует дополнительных контрольных точек для определения формы кривой. Пусть x_{i-1}, x_i, x_{i+1} и x_{i+2} – четыре последовательные точки данных с соответствующими функциональными значениями y_{i-1}, y_i, y_{i+1} и y_{i+2}. Функция сплайна Catmull-Rom для интервала $[x_i, x_{i+1}]$ задается формулой:

$$f(x) = a + bx + cx^2 + dx^3. \tag{18}$$

Коэффициенты рассчитываются по следующим формулам:

$$a = 0.5 * ((2y_i - 2y_{i+1}) + (y_{i-1} - y_{i+2})), \tag{19}$$
$$b = 0.5 * ((y_{i+1} - y_{i-1}) + 2y_i), \tag{20}$$
$$c = 0.5 * ((y_{i+1} - y_i) - (y_{i-1} - y_{i+2})), \tag{21}$$
$$d = 0.5 * ((y_{i-1} - y_{i+1}) + (2y_{i+1} - y_i)). \tag{22}$$

Сплайны Эрмита используют как значения функции, так и производные функции в каждой точке данных для построения гладкой кривой. Это позволяет им обрабатывать данные с разрывами или острыми углами более эффективно, чем другие типы сплайновых алгоритмов. Сплайны Эрмита широко используются в компьютерной графике и научных вычислениях.

Интерполяция сплайна Эрмита использует кусочно-определенную полиномиальную функцию степени 3 с двумя контрольными точками и их соответствующими производными, которые определяют форму кривой. Пусть x_i и x_{i+1} – две последовательные точки данных с соответствующими функциональными значениями y_i и y_{i+1}, и пусть f_i и f_{i+1}– их соответствующие производные. Сплайн-функция Эрмита для интервала $[x_i$, $x_{i+1}]$ задается формулой [8]:

$$f(x) = a + bx + cx^2 + dx^3. \quad (23)$$

Коэффициенты рассчитываются по следующим формулам:

$$a = y_i, \quad (24)$$
$$b = f_i, \quad (25)$$
$$c = 3(y_{i+1} - y_i) - 2f_i - f_{i+1}, \quad (26)$$
$$d = 2(y_i - y_{i+1}) + f_i + f_{i+1}. \quad (27)$$

Интерполяция сплайна Эрмита также может быть расширена до более высоких степеней, с большим количеством контрольных точек и их соответствующих производных.

Интерполяция сплайна Безье использует кусочно-определенную полиномиальную функцию степени n с $n + 1$ контрольными точками, которые определяют форму кривой. Пусть x_i и x_{i+1} – две последовательные точки данных с соответствующими функциональными значениями y_i и y_{i+1}, и пусть $P_0, P_1, \ldots, P_n$ – их соответствующие контрольные точки. Сплайн-функция Безье для интервала $[x_i, x_{i+1}]$ задается формулой:

$$B(t) = (1-t)^3 * P_0 + 3 * (1-t)^2 * t * P_1 +$$
$$+ 3 * (1-t) * t^2 * P_2 + t^3 * P_3. \quad (28)$$

где $B(t) -$ координаты точки на кривой Безье в момент времени t;

t – параметр, который изменяется от 0 до 1 и указывает положение точки на кривой Безье;

P_0, P_1, P_2, P_3 – точки управления, заданные пользователем.

Интерполяция сплайнов Безье может быть расширена до более высоких измерений с отдельными контрольными точками для каждого измерения.

Каждый из этих алгоритмов сплайновой интерполяции имеет свои собственные сильные и слабые стороны, и выбор того, какой алгоритм использовать, будет зависеть от конкретных потребностей приложения [7].

1.3 Алгоритмы сплайн-интерполяции

Естественная сплайновая интерполяция – это тип кубической сплайновой интерполяции, при которой вторые производные в конечных точках интервала устанавливаются равными нулю. Это приводит к получению более гладкой и естественной кривой, чем обычная интерполяция кубического сплайна.

Алгоритм построения натурального кубического сплайна из набора точек данных выглядит следующим образом:

учитывая набор из n точек данных $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, переменные $x_0 < x_1 < \dots < x_n$, проводим вычисление первых разностей значений y:

$$h_i = x_{i+1} - x_i, \quad (29)$$

$$\Delta_i = \frac{(y_{i+1} - y_i)}{h_i}, \text{ для } i = 0,1, \dots, n-1. \quad (30)$$

Проводим вычисление вторых разностей значений y:

$$d_i = \frac{2 * (\Delta_{i+1} - \Delta_i)}{h_{i+1} + h_i}, \text{ для } i = 0,1, \dots, n-2. \quad (31)$$

Далее происходит решение систем линейных уравнений, чтобы получить вторые производные в конечных и внутренних точках:

$$A_i * m_{i-1} + B_i * m_i + C_i + m_{i+1} = d_i, \text{ для } i = 1,2, \dots, n-2. \quad (32)$$

Коэффициенты рассчитываются по следующим формулам:

$$A_i = h_{i-1}, \quad (33)$$

$$B_i = 2 * (h_{i-1} + h_i), \quad (34)$$

$$C_i = h_i, \quad (35)$$

$$m_0 = m_n = 0. \quad (36)$$

Результирующая система линейных уравнений представляет собой трехдиагональную матрицу, которая может быть эффективно решена с помощью специализированных алгоритмов [4].

Проводим вычисление коэффициентов кубических многочленов для каждого интервала:

$$a_i = y_i, \quad (37)$$
$$b_i = \frac{\Delta_i - h_i * (2 * m_i + m_{i+1})}{6}, \quad (38)$$
$$c_i = \frac{m_i}{2}, \quad (39)$$
$$d_i = \frac{(m_{i+1} - m_i)}{6 * h_i}. \quad (40)$$

Результирующая кубическая сплайновая функция определяется кусочно на каждом интервале $[x_i, x_{i+1}]$ как:

$$S_i(x) = a_i + b_i * (x - x_i) + c_i * (x - x_i)^2 + \\ + d_i * (x - x_i)^3, \text{ для } x_i \le x \le x_{i+1}. \quad (41)$$

Преимущество этого алгоритма интерполяции естественного кубического сплайна состоит в том, что он создает плавную и естественно выглядящую кривую, проходящую через точки данных, избегая при этом проблемы превышения, которая может возникнуть при интерполяции ненатурального кубического сплайна.

Зажатый кубический сплайн – это кусочно-полиномиальная функция, которая проходит через заданный набор точек данных и имеет непрерывные первую и вторую производные. Алгоритм интерполяции с фиксированным кубическим сплайном включает в себя нахождение коэффициентов кубических полиномов, которые интерполируются между точками данных, с учетом дополнительных ограничений, согласно которым первые производные в конечных точках равны заданным значениям.

Для набора из $n+1$ точек данных $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, переменные $x_0 < x_1 < \dots < x_n$, и значений первых производных на концах: $f'(x_0) = m_0$ и $f'(x_n) = m_n$.

Выполним разделение интервала $[x_0, x_n]$ на n подинтервалов $[x_{i-1}, x_i]$, для $i = 1,2, \dots, n$.

Для каждого под-интервала $[x_{i-1}, x_i]$ находим кубический многочлен $S_i(x)$, который удовлетворяет следующим условиям:

$$S_i'(x_{i-1}) = y_{i-1}, S_i(x_i) = y_i, \quad (42)$$

$$S_i'(x_{i-1}) = S_{i-1}'(x_{i-1}), S_i'(x_i) = S_{i+1}'(x_i), \text{ для } i = 1,2, \dots, n-1, \quad (43)$$

$$S_0'(x_0) = m_0, S_n'(x_n) = m_n. \quad (44)$$

Первые два условия гарантируют, что кубический многочлен проходит через точки данных, а третье условие гарантирует, что первые производные непрерывны на под-интервалах.

Кубические полиномы для каждого под-интервала можно записать в следующем виде:

$$S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3. \quad (45)$$

Коэффициенты рассчитываются по следующим формулам:

$$a_i = y_{i-1}, b_i = S_i'(x_{i-1}), \quad (46)$$

а коэффициенты c_i и d_i находятся путем решения системы линейных уравнений.

Как только коэффициенты найдены для каждого под-интервала, ограниченная функция кубического сплайна $S(x)$ может быть определена как:

$$S(x) = S_i(x),\ \text{для}\ x_{i-1} \leq x \leq\ x_i, \text{для}\ i = 1,2, \dots, n. \tag{47}$$

Алгоритм интерполяции с фиксированным кубическим сплайном обеспечивает точную и плавную аппроксимацию точек данных, а дополнительное ограничение на первые производные в конечных точках делает его полезным для приложений, где значение функции и ее наклон известны на границах, например, в механике или инжиниринг [10].

Алгоритм сплайна Акима – это тип метода кусочно-кубической интерполяции, который строит непрерывную и гладкую функцию из набора точек данных. Алгоритм работает, вычисляя наклоны в каждой точке данных, используя средневзвешенное значение наклонов соседних точек. Наклоны затем используются для построения кусочно-кубических полиномов между соседними точками данных.

Общую идею алгоритма можно описать следующим образом:

учитывая набор из n точек данных (x_i, y_i), где $i = 0,1, \dots, n-1$, алгоритм вычисляет наклоны m_i в каждой точке, используя средневзвешенное значение наклонов соседних точек.

Веса, используемые для вычисления наклонов, определяются следующим образом:

$$w_i = \frac{|x_{i+1}-x_{i-1}|}{|x_{i+1}-x_i|+|x_i-x_{i-1}|}, \tag{48}$$

где $|x_{i+1} - x_i|$ и $|x_i - x_{i-1}|$ – расстояния между соседними точками данных.

Наклоны затем используются для построения кусочно-кубических полиномов между соседними точками данных.

В частности, для каждого интервала x_i, x_{i+1} кубический полином определяется как:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad (49)$$

где $a_i, b_i, c_i,$ и d_i – коэффициенты, которые определяются значениями наклона в точках x_i и x_{i+1} и значениями функции в точках y_i и y_{i+1}.

Коэффициенты можно найти, решив систему линейных уравнений, которая включает значения функции и ее первой производной в конечных точках интервала, а также наклоны в точках x_i и x_{i+1}.

Наконец, результирующая кусочно-кубическая функция строится путем объединения кубических многочленов, определенных на каждом интервале.

Сплайн-алгоритм Akima обеспечивает плавную и непрерывную интерполяцию данных, распределенных неравномерно. Он широко используется в различных областях, таких как компьютерная графика, обработка сигналов и численный анализ, благодаря своей точности и надежности.

Алгоритм интерполяции B-сплайном:

на вход подается набор точек данных $(x_i, y_i), i = 1,2,\ldots,n$, и параметр степени k.

На выход: гладкая функция $f(x)$, которая проходит через точки данных.

Проводим вычисление векторов узлов $t = \{t_0, t_1, \ldots, t_m\}$, где $m = n + k + 1$,

а значения узлов t_i определяются следующим образом:

$t_i = x_i$, если $i \leq k$, $t_i = x_n$, если $i > n$, $t_i = t_{i-1} + 1$, для $i = k + 1, k + + 2, \ldots, n$.

Определим базисные функции B-сплайна $B_i(x)$ степени k и вектора узла t следующим образом:

$$B_0(x) = 1 \text{ для } t_i \leq x < t_{i+1}, \quad (50)$$

$$B_k(x) = \frac{(x - t_i)}{\frac{\left(t_{i+k-1-t_i}\right) * B_{k-1}(x) + (t_{i+k} - x)}{(t_{i+k} - t_{i+1}) * B_{k-1}(x+1)}}, \text{для } k \geq 1 \text{ и } t_i \leq x < t_{i+k+1}. \quad (51)$$

Вычислим матрицу M и вектор D следующим образом:

для $i = 1,2,\ldots,n$, и $j = 1,2,\ldots,n$, $M_{ij} = B_i(x_j)$, $D_i = y_i$.

Проводим решение линейной системы уравнений $M * Y = D$, чтобы получить коэффициенты Y.

Функция $f(x)$ задается линейной комбинацией базисных функций B-сплайна с использованием коэффициентов Y:

$$f(x) = \sum_{i=1}^{n} Y_i * B_i(x). \tag{52}$$

На выходе получаем $f(x)$.

Этот алгоритм может быть реализован с использованием стандартных численных методов решения линейных систем уравнений, таких как исключение Гаусса или LU-разложение. Сложность алгоритма составляет $O(n^2)$ для вычисления матрицы M и $O(n^3)$ для решения линейной системы уравнений, но его можно оптимизировать с помощью эффективных алгоритмов и структур данных для разреженных матриц.

Алгоритм сплайн-интерполяции Catmull-Rom – это метод, используемый для интерполяции между набором точек для получения гладкой кривой. Он был разработан Эдвином Кэтмаллом и Рафаэлем Ромом в 1974 году и с тех пор стал популярным алгоритмом, используемым в компьютерной графике и анимации.

Алгоритм работает путем подгонки кривой к набору точек с использованием кусочно-полиномиальной функции. Полученная кривая проходит через каждую из входных точек и является гладкой в точках пересечения. Это делает его особенно полезным для создания анимаций с плавно меняющимися траекториями движения.

Чтобы выполнить расчет кривой по набору точек $P_0, P_1, P_2, \ldots, P_{n-1}$ вычислим контрольные точки $Q_{1,}\ Q_2, \ldots, Q_{n-2}$.

Для каждой точки P_i определим две контрольные точки Q_i и Q_{i+1} следующим образом:

$$Q_i = P_{i-1} + \left(\frac{1}{2}\right) * (P_i - P_{i-2}), \quad (53)$$

$$Q_{i+1} = P_i + \left(\frac{1}{2}\right) * (P_{i+1} - P_{i-1}). \quad (54)$$

Затем определим кривую между P_i и P_{i+1} как кубическую полиномиальную функцию:

$$C_i(t) = at^3 + bt^2 + ct + d. \quad (55)$$

Коэффициенты рассчитываются по следующим формулам:

$$a = -0{,}5 * Q_{j-1} + 1{,}5 * Q_j - 1{,}5 * Q_{j+1} + 0{,}5 * Q_{j+2}, \quad (56)$$

$$b = Q_{j-1} - 2{,}5 * Q_j + 2 * Q_{j+1} - 0{,}5 * Q_{j+2}, \quad (57)$$

$$c = 0{,}5 * Q_{j+1} - 0{,}5 * Q_{j-1}, \quad (58)$$

$$d = Q_j. \quad (59)$$

Затем интерполируем между каждой парой точек P_i и P_{i+1}, оценивая $C_i(t)$ при различных значениях t от 0 до 1.

Например, если хотим интерполировать между P_1 и P_2, то происходит оценка $C_1(t)$ для t между 0 и 1.

Наконец, соединим кривые для каждой пары точек, чтобы получить гладкую общую кривую.

В целом, алгоритм сплайновой интерполяции Catmull-Rom обеспечивает простой и эффективный способ интерполяции между набором точек и создания плавной кривой. Его универсальность и простота реализации сделали его популярным выбором для многих приложений в компьютерной графике и анимации.

Сплайны Эрмита – это тип кривой, которую можно использовать для интерполяции точек данных. Они полезны, когда точки данных имеют как значение, так и производную в каждой точке. Алгоритм сплайнов Эрмита используется для вычисления коэффициентов этих кривых.

Алгоритм сплайнов Эрмита работает путем деления точек данных на меньшие интервалы и построения кубического многочлена, который проходит через каждую точку и имеет ту же производную, что и соседние интервалы в этой точке. Это гарантирует, что кривая будет гладкой и непрерывной.

Для расчета коэффициентов сплайна Эрмита можно предпринять следующие шаги:

разделение точек данных на меньшие интервалы.

Для каждого интервала происходит расчет наклона между соседними точками данных, используя следующую формулу:

$$c = \frac{(y_2 - y_1)}{(x_2 - x_1)}, \qquad (60)$$

где s – наклон, y_2 и y_1 – значения соседних точек данных, а x_2 и x_1 – их соответствующие положения.

Построение кубического многочлена для каждого интервала, используя следующую формулу:

$$f(x) = a + b(x - x_i) + c(x - x_i)^2 + d(x - x_i)^3, \qquad (61)$$

где $f(x)$ – значение сплайна Эрмита в позиции x, x_i – положение первой точки данных в интервале, а a, b, c и d – коэффициенты многочлена.

Для расчета коэффициентов решим следующую систему уравнений:

$$f(x_i) = y_1, \qquad (62)$$
$$f(x_{i+1}) = y_2, \qquad (63)$$
$$f'(x_i) = s_1, \qquad (64)$$
$$f'(x_{i+1}) = s_2, \qquad (65)$$

где y_1 и y_2 – значения соседних точек данных, s_1 и s_2 – наклоны между соседними точками данных, а x_i и x_{i+1} – их соответствующие положения.

Построение кубического многочлена и решение системы уравнений для каждого интервала, чтобы вычислить коэффициенты сплайна Эрмита.

После расчета коэффициентов сплайн Эрмита можно использовать для интерполяции точек данных в пределах диапазона входных данных. Результирующая кривая будет гладкой и непрерывной и будет проходить через каждую точку данных с той же производной, что и соседние интервалы.

Алгоритм интерполяции сплайнов Безье – это математический метод, используемый для построения гладкой кривой через набор точек. Он обычно используется в компьютерной графике, анимации и программном обеспечении CAD для создания кривых и форм.

Алгоритм работает путем построения серии кривых Безье, которые проходят через каждую из входных точек. Кривая Безье определяется набором контрольных точек, а сама кривая является функцией одного параметра t, который находится в диапазоне от 0 до 1.

Первым шагом в алгоритме сплайновой интерполяции Безье является выбор степени для кривых Безье. Часто используется степень 3 (т. е. кубическая), поскольку она обеспечивает хороший баланс между гладкостью и сложностью.

Далее алгоритм строит набор кривых Безье, проходящих через входные точки. Каждая кривая определяется набором контрольных точек, которые рассчитываются следующим образом: первая и последняя контрольные точки устанавливаются в соответствующие входные точки.

Остальные контрольные точки вычисляются с помощью рекурсивного алгоритма, зависящего от степени кривой. Для кубической кривой вторая и предпоследняя контрольные точки вычисляются следующим образом:

$$P_1 = P_0 + \frac{P_2 - P_0}{3}, \tag{66}$$

$$P_2 = P_3 - \frac{P_3 - P_1}{3}, \tag{67}$$

где P_0, P_1, P_2 и P_3 – первая, вторая, третья и четвертая входные точки соответственно.

Затем алгоритм строит кривую Безье, используя набор контрольных точек для каждого сегмента. Наконец, алгоритм соединяет кривые Безье встык, чтобы сформировать гладкую кривую, проходящую через все входные точки.

Таким образом, алгоритм сплайновой интерполяции Безье строит гладкую кривую через набор точек, строя серию кривых Безье, проходящих через каждую точку. Алгоритм является гибким и может использоваться для создания кривых различной степени гладкости и сложности.

1.4 Сравнительный анализ алгоритмов, их производительность, точность и эффективность

Сравнение производительности, точности и действенности алгоритмов сплайновой интерполяции может быть затруднено, поскольку каждый алгоритм имеет свои собственные сильные и слабые стороны. Однако вот некоторые общие соображения, которые могут помочь при сравнении:

Точность алгоритма сплайновой интерполяции зависит от того, насколько хорошо он аппроксимирует исходные данные. Как правило, сплайны более высокой степени (например, кубический или B-сплайн) обеспечивают лучшую точность, чем сплайны более низкой степени (например, линейные или квадратичные). Однако сплайны более высокой степени также могут быть более чувствительными к шуму или выбросам в данных.

Эффективность алгоритма сплайновой интерполяции зависит от того, насколько быстро он может вычислить интерполирующую кривую. Как правило, сплайны более низкой степени вычисляются быстрее, чем сплайны более высокой степени. Однако некоторые алгоритмы (например, Akima spline) используют локальные методы для оценки наклона кривой, что может сделать их более быстрыми и эффективными, чем другие типы сплайнов.

Плавность алгоритма сплайновой интерполяции зависит от того, насколько хорошо он интерполирует точки данных, не внося резких изменений или колебаний в кривую. В общем, сплайны более высокой степени, как правило, более гладкие, чем сплайны более низкой степени. Однако некоторые алгоритмы (например, сплайн Эрмита) могут обрабатывать данные с разрывами или острыми углами более эффективно, чем другие типы сплайнов.

Надежность алгоритма сплайновой интерполяции зависит от того, насколько хорошо он обрабатывает зашумленные или посторонние данные. В целом, некоторые алгоритмы (например, Akima spline) более устойчивы к зашумленным данным, чем другие типы сплайнов.

Выбор граничных условий также может повлиять на точность и плавность алгоритма сплайновой интерполяции. Некоторые алгоритмы (например, естественный сплайн) накладывают определенные граничные условия для обеспечения того, чтобы интерполирующая кривая была гладкой и непрерывной в конечных точках. Другие алгоритмы (например, периодический сплайн) могут лучше подходить для данных, которые периодически повторяются.

Реализация алгоритма сплайновой интерполяции также может повлиять на его производительность и действенность. Некоторые алгоритмы могут требовать более сложных или дорогостоящих в вычислительном отношении процедур, в то время как другие могут быть проще и быстрее в вычислении. Выбор реализации может зависеть от конкретной используемой аппаратной или программной платформы.

Характеристики интерполируемых данных также могут влиять на производительность и точность алгоритмов сплайновой интерполяции. Например, для данных с большим количеством острых углов или разрывов может потребоваться алгоритм сплайна, который может обрабатывать такие особенности. Аналогично, для данных с большим количеством зашумленных или выделяющихся точек данных может потребоваться более надежный алгоритм сплайна.

В целом, интерполяция кубического сплайна является широко используемым и хорошо зарекомендовавшим себя методом, который обеспечивает хороший баланс точности, эффективности и плавности для многих приложений. Однако другие алгоритмы сплайновой интерполяции могут быть более подходящими в зависимости от конкретных потребностей приложения, такие как B-сплайны для компьютерной графики или сплайны Эрмита для научных вычислений. При выборе алгоритма сплайновой интерполяции важно тщательно учитывать характеристики данных и конкретные требования приложения.

Таким образом, не существует единого алгоритма сплайновой интерполяции, который был бы наилучшим для всех приложений. Выбор алгоритма зависит от конкретных требований приложения, таких как желаемая точность, плавность и эффективность интерполирующей кривой, а также характеристик интерполируемых данных. Важно тщательно оценить сильные и слабые стороны различных алгоритмов сплайновой интерполяции и выбрать тот, который наилучшим образом соответствует потребностям приложения.

1.5 Примеры реального применения алгоритмов сплайн-интерполяции

Алгоритмы сплайновой интерполяции имеют широкий спектр применений в различных областях, включая компьютерную графику, научные вычисления и инженерию. Вот несколько примеров реального применения алгоритмов сплайновой интерполяции.

Компьютерная графика: в компьютерной графике алгоритмы сплайновой интерполяции обычно используются для создания плавных кривых для рендеринга 2D и 3D-объектов. Интерполяция Catmull-Rom особенно полезна для этого приложения, поскольку она может обрабатывать сложные формы и обеспечивать плавную кривую, проходящую через набор контрольных точек.

Географические информационные системы (ГИС): в ГИС алгоритмы сплайновой интерполяции используются для создания гладких поверхностей из разрозненных точек данных, таких как данные о высоте. Затем эти поверхности можно использовать для различных применений, таких как моделирование рельефа и анализ окружающей среды. Одним из примеров алгоритма сплайновой интерполяции, используемого в ГИС, является кригинг, который использует пространственную статистическую модель для интерполяции значений данных.

Обработка изображений: при обработке изображений алгоритмы сплайновой интерполяции используются для повторной выборки изображений с различными разрешениями или для выполнения деформации и морфинга изображений. Интерполяция B-сплайнов часто используется для этих приложений, поскольку она может обрабатывать неоднородные сетки выборки и получать гладкое изображение.

Обработка сигналов: при обработке сигналов алгоритмы сплайновой интерполяции используются для восстановления сигналов из дискретных выборок или для сглаживания зашумленных сигналов. Интерполяция кубического сплайна обычно используется для этого приложения, поскольку она обеспечивает хороший баланс точности и плавности.

Научные вычисления: в научных вычислениях алгоритмы сплайновой интерполяции используются для подгонки экспериментальных данных и решения дифференциальных уравнений. Интерполяция сплайнов Эрмита часто используется для этого приложения, поскольку она может обрабатывать данные с разрывами и обеспечивать плавную кривую, проходящую через набор точек данных.

Инженерия: в инженерном деле алгоритмы сплайновой интерполяции используются для проектирования и анализа различных конструкций, таких как крылья самолетов и автомобильные рамы. Интерполяция сплайнов особенно полезна для проектирования плавных кривых и поверхностей, которые отвечают определенным критериям проектирования, таким как минимизация нагрузок или максимизация аэродинамических характеристик.

Финансовое моделирование: в финансовом моделировании алгоритмы сплайновой интерполяции используются для оценки срочной структуры процентных ставок. Это включает в себя подгонку кривой, проходящей через набор наблюдаемых цен на облигации и доходностей. Интерполяция кубического сплайна часто используется для этого приложения, поскольку она может обеспечить плавную кривую, которая отражает временную структуру процентных ставок.

Медицинская визуализация: в медицинской визуализации алгоритмы сплайновой интерполяции используются для восстановления 3D-изображений из 2D-сканирований или для интерполяции между фрагментами 3D-изображения. Интерполяция B-сплайнов часто используется для этого приложения, поскольку она может обрабатывать неоднородные сетки выборки и обеспечивать плавное изображение.

Физическое моделирование: в физическом моделировании алгоритмы сплайновой интерполяции используются для аппроксимации поведения физических систем. Это включает в себя подгонку кривой, которая удовлетворяет определенным физическим ограничениям, таким как сохранение энергии или импульса. Интерполяция кубического сплайна часто используется для этого приложения, поскольку она может обеспечить плавную кривую, удовлетворяющую этим ограничениям.

В целом, алгоритмы сплайновой интерполяции имеют широкий спектр применений в различных областях и являются важным инструментом для создания плавных кривых и поверхностей из разрозненных точек данных.

Это всего лишь несколько примеров из множества реальных применений алгоритмов сплайновой интерполяции. Гибкость и точность сплайновой интерполяции делают ее универсальным инструментом, который может быть применен к широкому кругу задач.

Вывод по первой главе

В первой части главы было определено, что такое сплайн-интерполяция, а также ее преимущества и недостатки. Рассмотрены различные виды и формулы сплайн-интерполяции. Представлены алгоритмы сплайн-интерполяции, которые используются для построения кривых и поверхностей.

Проведен сравнительный анализ алгоритмов, где были рассмотрены их производительность, точность и эффективность. Это позволило выбрать наиболее подходящий алгоритм для конкретной задачи. В последнем разделе главы, были приведены примеры реального применения алгоритмов сплайн-интерполяции.

В целом, глава представляет собой важный обзор алгоритмов сплайн-интерполяции, их различных формул, а также преимуществ и недостатков. Кроме того, глава предоставляет информацию о сравнительном анализе алгоритмов и примерах их реального применения. Так как в компьютерной графике чаще всего используется алгоритмы сплайн-интерполяции Catmull-Rom и кубическая кривая Безье, дальнейшая работа будет именно с данными алгоритмами.

Глава 2 Реализация сплайн-интерполяции при разработке мобильного приложения

2.1 Введение в разработку мобильных приложений

Разработка мобильных приложений – это процесс проектирования, разработки и развертывания программных приложений для мобильных устройств, таких как смартфоны, планшеты и носимые устройства. С быстрым ростом мобильной индустрии разработка мобильных приложений стала важной областью внимания как для разработчиков программного обеспечения, так и для бизнеса.

Разработка мобильного приложения требует различных навыков, включая разработку программного обеспечения, дизайн пользовательского интерфейса и программирование мобильных устройств.

Процесс разработки мобильных приложений включает в себя несколько этапов, включая создание идеи, дизайн, разработку, тестирование и развертывание. На стадии разработки идеи разработчик определяет назначение приложения и целевую аудиторию. На стадии проектирования разработчик создает каркасы и разрабатывает пользовательский интерфейс. На стадии разработки происходит собственно кодирование и запрограммированность. На этапе тестирования разработчик проверяет наличие ошибок и гарантирует, что приложение функционирует должным образом. Наконец, на этапе развертывания приложение становится общедоступным.

Мобильные приложения могут служить множеству целей, таких как развлечение, повышение производительности, коммуникация и образование. Их можно загрузить и установить из магазинов приложений, таких как Apple App Store или Google Play Store [12].

Разработка мобильных приложений может осуществляться с использованием различных языков программирования, фреймворков и инструментов. Unity – популярный игровой движок, используемый для разработки мобильных игр для различных платформ, таких как iOS, Android и других. Unity известна своей простотой использования и кроссплатформенными возможностями, которые делают ее популярным выбором для разработчиков игр.

Unity поддерживает широкий спектр языков программирования, включая C#, JavaScript и Boo. C# является рекомендуемым языком для разработки Unity mobile. Unity позволяет разрабатывать игры как для 2D, так и для 3D платформ [14].

Чтобы начать разработку Unity mobile, вам нужно будет загрузить и установить редактор Unity. Редактор Unity – это мощный инструмент, который позволяет вам создавать, тестировать и развертывать ваши игры на различных платформах. После того, как вы установили редактор Unity, вы можете приступить к созданию своей игры.

Первым шагом в создании мобильной игры с Unity является создание нового проекта. Вы можете сделать это, выбрав "Новый проект" в меню "Файл". Далее вам нужно будет выбрать платформу, для которой вы хотите разрабатывать. Unity поддерживает широкий спектр платформ, включая iOS, Android, Windows, Mac и другие.

После того, как вы выбрали платформу, вы можете приступить к разработке своей игры. Unity предоставляет широкий спектр инструментов для проектирования и создания игр, включая мощный редактор, физический движок и скриптовый движок.

В Unity также есть большое сообщество разработчиков, которые делятся советами и хитростями, оказывают поддержку и предлагают учебные пособия по использованию платформы. Это сообщество – отличный ресурс для разработчиков, которые только начинают свою деятельность.

После того как вы разработаете и соберете свою игру, вы сможете протестировать ее на своем мобильном устройстве. Unity предоставляет ряд инструментов для тестирования вашей игры, включая мобильный эмулятор и тестирование на удаленном устройстве.

Наконец, как только вы будете готовы к развертыванию своей игры, Unity предоставит ряд инструментов для создания и упаковки вашей игры для различных платформ. Вы можете создать свою игру для iOS, Android, Windows, Mac и других платформ, и Unity автоматически упакует вашу игру для распространения.

В заключение можно сказать, что Unity – это мощный инструмент для разработки мобильных игр. Он предоставляет ряд инструментов для проектирования, сборки, тестирования и развертывания вашей игры на различных платформах. Благодаря простоте использования и кроссплатформенным возможностям Unity является популярным выбором для разработчиков игр [25].

2.2 Варианты использования сплайн-интерполяции в разработке мобильных приложений

Интерполяция сплайнов может быть полезным методом для разработчиков мобильных приложений в различных вариантах использования.

Изменение размера изображения: при изменении размера изображения в мобильном приложении можно использовать сплайновую интерполяцию для сглаживания краев и предотвращения неровностей. Интерполяция B-сплайнов часто используется для этого приложения, поскольку она может обеспечить плавную кривую, проходящую через пиксели изображения.

Обработка аудио и видео: при обработке аудио или видеоданных в мобильном приложении для интерполяции недостающих точек данных и создания более плавного сигнала можно использовать сплайновую интерполяцию. Интерполяция кубического сплайна часто используется для этого приложения, поскольку она может обеспечить плавную кривую, которая отражает динамику сигнала.

Распознавание жестов: при обнаружении и отслеживании жестов пользователя в мобильном приложении можно использовать сплайновую интерполяцию для создания плавной траектории движений пользователя. Это может повысить точность алгоритмов распознавания жестов и создать более естественный пользовательский интерфейс.

Анимация: при анимации объектов в мобильном приложении для создания плавных и естественных движений можно использовать сплайновую интерполяцию. Кривые Безье и сплайны Catmull-Rom часто используются для этого приложения, поскольку они могут создавать сложные кривые и анимацию.

Отображение и навигация: при отображении карт и навигационных маршрутов в мобильном приложении сплайновая интерполяция может использоваться для интерполяции недостающих точек данных и создания плавного маршрута. Интерполяция кубического сплайна часто используется для этого приложения, поскольку она может обеспечить плавную кривую, которая отражает форму и направление маршрута.

Визуализация данных: при отображении диаграмм и графиков в мобильном приложении можно использовать сплайновую интерполяцию для создания плавной и непрерывной кривой между точками данных. Это может улучшить удобочитаемость и эстетичность визуализации, а также облегчить пользователям интерпретацию данных. Естественная интерполяция кубического сплайна часто используется для этого приложения, поскольку она может создать плавную кривую, которая проходит через точки данных и сохраняет монотонность и вогнутость данных.

Обработка данных датчиков: при обработке данных датчиков с акселерометра, гироскопа или других датчиков в мобильном приложении можно использовать сплайновую интерполяцию для интерполяции недостающих точек данных и создания более плавного сигнала. Это может повысить точность алгоритмов обработки данных датчиков и создать более надежный пользовательский интерфейс. Интерполяция кубического сплайна часто используется для этого приложения, поскольку она может обеспечить плавную кривую, которая отражает динамику данных датчика.

Игры: при создании игр в мобильном приложении сплайновая интерполяция может использоваться для создания плавных и реалистичных анимаций и движений. Кривые Безье и сплайны Catmull-Rom часто используются для этого приложения, поскольку они могут создавать сложные кривые и анимацию, которые реагируют на вводимые пользователем данные и создают увлекательный игровой процесс [15].

Дополненная реальность: при создании опыта дополненной реальности (AR) в мобильном приложении сплайновая интерполяция может использоваться для создания плавных и естественных движений виртуальных объектов в реальном мире. Кривые Безье и сплайны Catmull-Rom часто используются для этого приложения, поскольку они могут создавать сложные кривые и анимацию, которые следуют за движением и ориентацией пользователя.

В заключение, сплайновая интерполяция – это универсальный метод, который может быть использован многими различными способами при разработке мобильных приложений. Сглаживая точки данных и создавая плавные кривые, сплайновая интерполяция может повысить точность, естественность и удобство использования мобильных приложений в различных областях.

Таким образом, сплайновая интерполяция может быть ценным инструментом для разработчиков мобильных приложений во многих различных случаях использования, начиная от обработки данных и визуализации и заканчивая играми и дополненной реальностью. Создавая плавные и непрерывные кривые между точками данных, сплайновая интерполяция может повысить точность, естественность и вовлеченность мобильных приложений в различных областях.

2.3 Реализация сплайн-интерполяции в мобильном приложении

Для реализации сплайновой интерполяции в мобильном приложении Unity, выбрали два метода сплайн-интерполяции: Catmull-Rom и кривая Безье. Для начала реализуем алгоритм Catmull-Rom [24].

Объявим переменные cPnt (массив точек контроля), nPntPerSgmnt (количество точек на каждом сегменте) и points (массив вычисленных точек для сплайна) [20].

Определяем метод UpdatePoints(), который вычисляет точки для сплайна.

Вычисляем количество сегментов сплайна, вычитая 3 из длины массива cPnt.

Итерируем через каждый сегмент сплайна и для каждого сегмента генерируем nPntPerSgmnt точек [13].

Для каждой точки используем параметр t, который изменяется от 0 до 1 в равных интервалах.

Вычисляем точку на сплайне с помощью метода CalculatePoint() и сохраняем ее в массив points.

Определяем метод CalculatePoint(), который вычисляет точку на сплайне для заданного значения параметра t и набора точек контроля p1,p2,p3,p4 [22].

Блок-схема алгоритма предоставлена на рисунке 1.

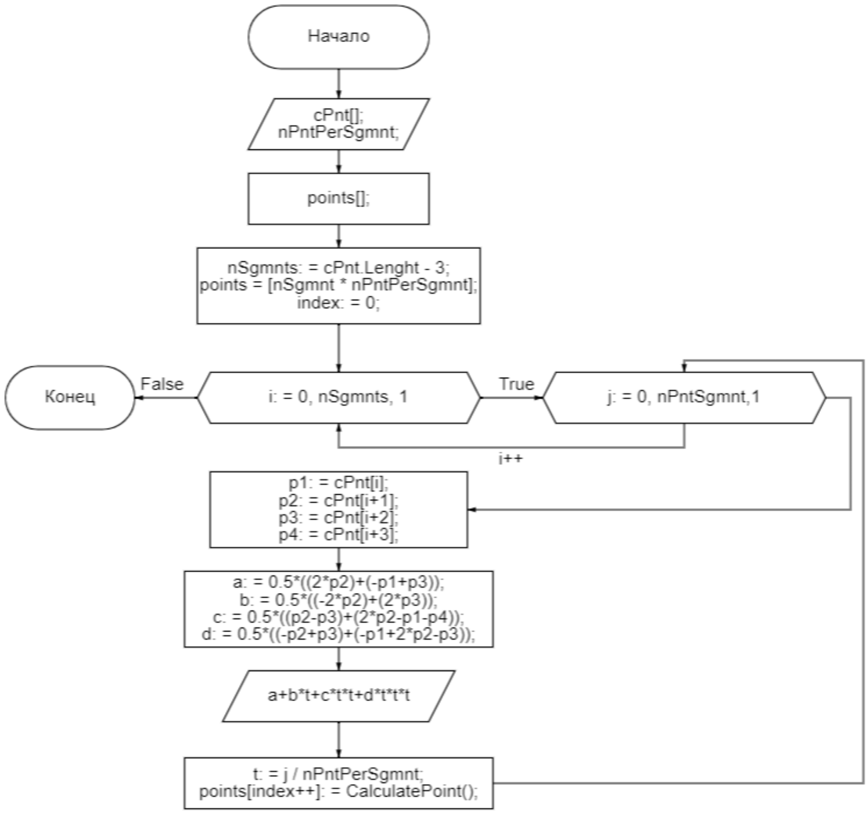

Рисунок 1 – Блок-схема с использованием сплайн-интерполяции Catmull-Rom

Данный код можно использовать для создания плавных сплайнов в Unity, задавая точки управления через массив Transform и настраивая параметры сплайна по своему усмотрению, как показано на рисунке 2 [16].

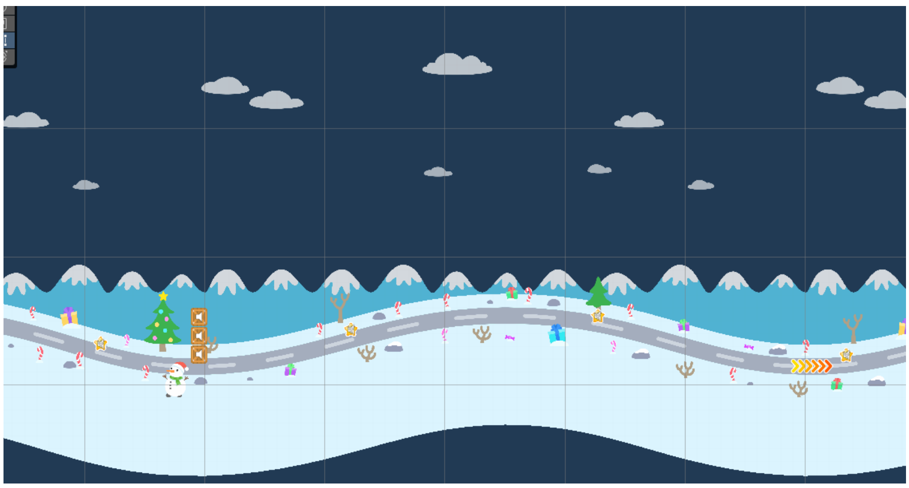

Рисунок 2 – Результат с использованием сплайн-интерполяции Catmull-Rom

В дополнение к интерполяции Catmull-Rom, существуют другие алгоритмы сплайнов, которые можно использовать, такие как кривые Безье и сплайны Эрмита. Выбор алгоритма будет зависеть от конкретных потребностей вашего приложения и желаемых свойств интерполируемой кривой. Далее мы рассмотрим алгоритм сплайн-интерполяции кривая Безье предоставленный на рисунке 3 [17].

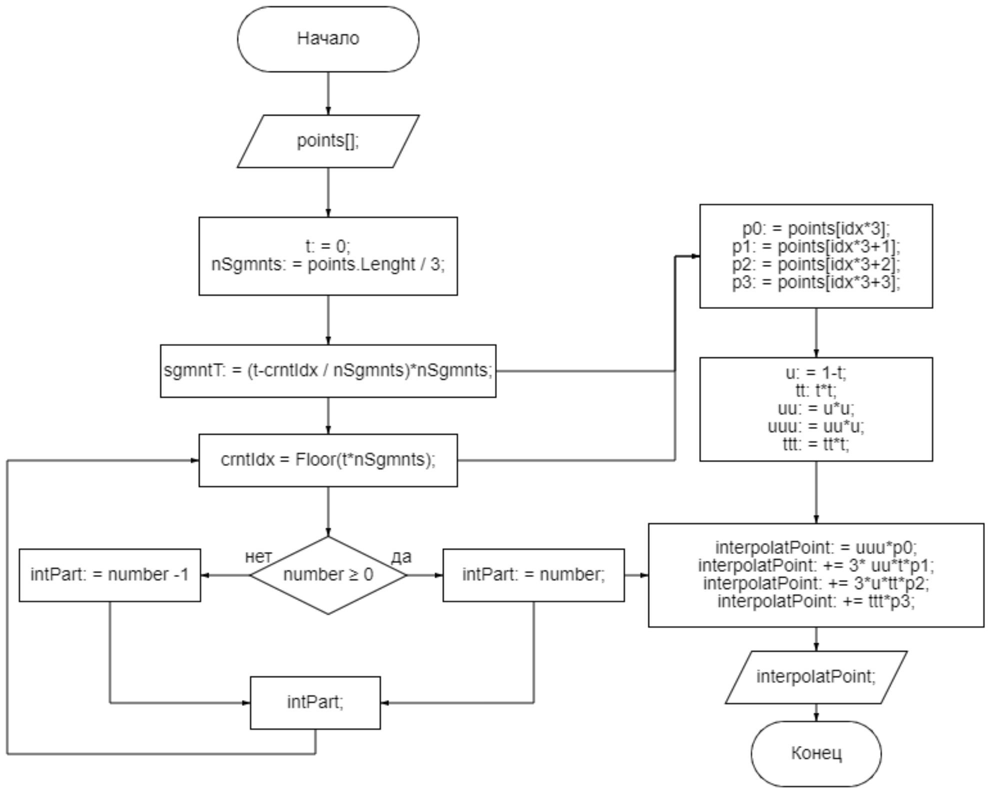

Рисунок 3 – Блок-схема с использованием сплайн-интерполяции кривой Безье

Объявим переменную points (массив точек кривой Безье) [23].

Метод InterPolate() выполняет интерполяцию кривой Безье в заданной точке t.

Вычисляем количество сегментов кривой Безье путем деления длины массива points на 3 (каждый сегмент содержит 3 точки).

Вычисляем текущий индекс сегмента, в котором находится точка t.

Вычисляем локальный параметр сегмента segmentT, который представляет долю пути внутри текущего сегмента.

Вызываем метод CalculateSegment(), чтобы вычислить точку на кривой Безье для заданного сегмента и параметра segmentT.

Данный метод вычисляет точку на сегменте кривой Безье.

Извлекаются четыре точки контроля сегмента из массива points.

Вычисляются вспомогательные значения для интерполяции на кривой Безье.

Вычисляются интерполированная точка на сегменте с использованием формул кривой Безье [21].

Возвращается интерполированная точка.

Метод Floor, используется для округления числа number вниз до ближайшего целого числа.

Итоговый график показан на рисунке 4.

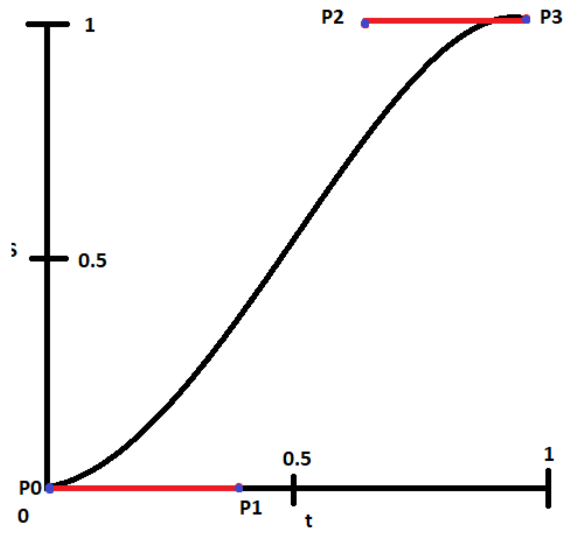

Рисунок 4 – Итоговый график вычисления сплайн-интерполяции кривой Безье

Стоит отметить, что сплайновая интерполяция может использоваться не только для позиционирования объектов. Он также может быть использован для интерполяции цветов, масштабов, поворотов и других свойств игровых объектов. Например, вы могли бы использовать сплайновую интерполяцию для плавного перехода цвета игрового объекта с одного цвета на другой.

Наконец, важно иметь в виду последствия использования сплайновой интерполяции для производительности, особенно на мобильных устройствах с ограниченными ресурсами. Интерполяция большого количества точек данных за длительный период времени может потребовать больших вычислительных затрат и может повлиять на производительность вашего приложения. Чтобы смягчить это, вы можете рассмотреть возможность уменьшения количества точек данных, использования более простых алгоритмов интерполяции или реализации оптимизаций, таких как предварительное вычисление сплайновой кривой.

Еще одним соображением при использовании сплайновой интерполяции является качество входных данных. Точность и разрешение точек данных, используемых для интерполяции, могут сильно повлиять на плавность и точность интерполируемой кривой. Например, если точки данных расположены слишком далеко друг от друга или данные зашумлены, интерполированная кривая может выглядеть неровной или неустойчивой. С другой стороны, если точки данных расположены слишком близко друг к другу, интерполированная кривая может показаться чрезмерно сглаженной или нереалистичной.

Чтобы решить эти проблемы, важно тщательно выбирать и предварительно обрабатывать входные данные. Это может включать фильтрацию, сглаживание или повторную выборку данных для обеспечения их согласованности и точности. Вы также можете рассмотреть возможность использования адаптивного или динамического подходов для настройки разрешения точек данных в зависимости от сложности кривой или желаемого уровня детализации [19].

В целом, сплайновая интерполяция может стать мощным инструментом для создания плавных и реалистичных анимаций и переходов в мобильных приложениях Unity. Тщательно отбирая и предварительно обрабатывая свои данные, выбирая подходящие алгоритмы и оптимизируя производительность, вы можете добиться высококачественного и привлекательного пользовательского интерфейса, который повысит общую привлекательность вашего приложения [18].

Вывод по второй главе

Глава посвящена использованию сплайн-интерполяции в разработке мобильных приложений. В начале главы рассмотрены основы разработки мобильных приложений. Затем описаны различные варианты использования сплайн-интерполяции в мобильных приложениях, такие как анимация, интерполяция движения, интерполяция цвета и т.д. Описан процесс реализации сплайн-интерполяции в мобильном приложении.

В целом, глава предоставляет полезную информацию о том, как можно использовать сплайн-интерполяцию в мобильных приложениях для создания более плавного и привлекательного пользовательского интерфейса. Рассмотренные примеры использования сплайн-интерполяции могут помочь разработчикам выбрать наиболее подходящий метод для своего проекта.

Таким образом, глава 2 представляет важную информацию для разработчиков мобильных приложений, которые хотят улучшить пользовательский опыт с помощью сплайн-интерполяции.

Глава 3 Тестирование работы мобильного приложения с применением сплайн-интерполяции

3.1 Производительность с использованием сплайн-интерполяции

Производительность сплайн-интерполяции Catmull-Rom зависит от количества точек, которые необходимо интерполировать, а также от метода, используемого для расчета коэффициентов интерполяции. В общем случае при большом количестве точек это может занять значительное время, но при малом количестве точек расчеты выполняются быстро, данные представлены на рисунке 5.

Сплайн-интерполяция Catmull-Rom также может быть реализована на графическом процессоре, что позволяет значительно ускорить процесс расчета, особенно при обработке больших объемов данных. В этом случае параллельные вычисления позволяют быстрее вычислять значения для каждого пикселя графика.

Хотя сплайн-интерполяция Catmull-Rom имеет хорошую производительность и может использоваться для создания плавных кривых, ее также следует использовать с осторожностью. Это связано с тем, что при наличии большого количества точек для интерполяции интерполяция может привести к нежелательному искажению данных и потере информации. Поэтому перед использованием сплайн-интерполяции Catmull-Rom необходимо тщательно оценить требования к точности и количество точек, которые будут использоваться в процессе интерполяции.

Производительность сплайн-интерполяции кривой Безье также зависит от количества контрольных точек, которые необходимо интерполировать. Однако вычислительная сложность интерполяции кривой Безье обычно ниже, чем у интерполяции Catmull-Rom, что может сделать ее быстрее для большего количества контрольных точек.

Сплайн-интерполяция кривой Безье также может быть реализована на графическом процессоре, что может еще больше ускорить процесс расчета, особенно при обработке больших объемов данных. В этом случае можно использовать параллельные вычисления для ускорения вычисления значений для каждого пикселя графика.

Однако следует отметить, что сплайн-интерполяция кривой Безье не так гибка, как интерполяция Catmull-Rom. Кривые Безье ограничены контрольными точками, что означает, что они могут неточно отражать базовые данные для некоторых графиков. Кроме того, пользователь должен вручную выбирать контрольные точки, что может занять много времени и требует тщательного рассмотрения, чтобы убедиться, что результирующая кривая точно представляет данные.

В целом, сплайн-интерполяция кривой Безье может быть эффективным методом создания плавных кривых для графиков, особенно при работе с большим количеством контрольных точек. Однако его следует использовать с осторожностью и с тщательным рассмотрением исходных данных и выбора контрольных точек, данные представлены на рисунке 5.

Рисунок 5 – График времени обработки алгоритмов с использованием 100 точек

3.2 Сравнение сплайн-интерполяции

Одно из основных различий между сплайн-интерполяцией Catmull-Rom и интерполяцией кривой Безье заключается в том, как они интерполируют значения между соседними точками или контрольными точками. Интерполяция Catmull-Rom использует кубические полиномы, а интерполяция кривой Безье использует полиномы Бернштейна. Оба метода обеспечивают непрерывность первой производной, но кривые Безье также обеспечивают непрерывность второй производной. Это означает, что кривые Безье обычно более гладкие и содержат меньше артефактов, чем сплайны Catmull-Rom.

Еще одним отличием этих методов является их гибкость. Сплайны Catmull-Rom более гибкие, потому что они интерполируют значения между соседними точками, что означает, что они могут точно представлять более сложные кривые с большим количеством точек перегиба, как показано на рисунке 6. Напротив, кривые Безье ограничены контрольными точками, что означает, что они могут неточно отражать основные данные для некоторых графиков.

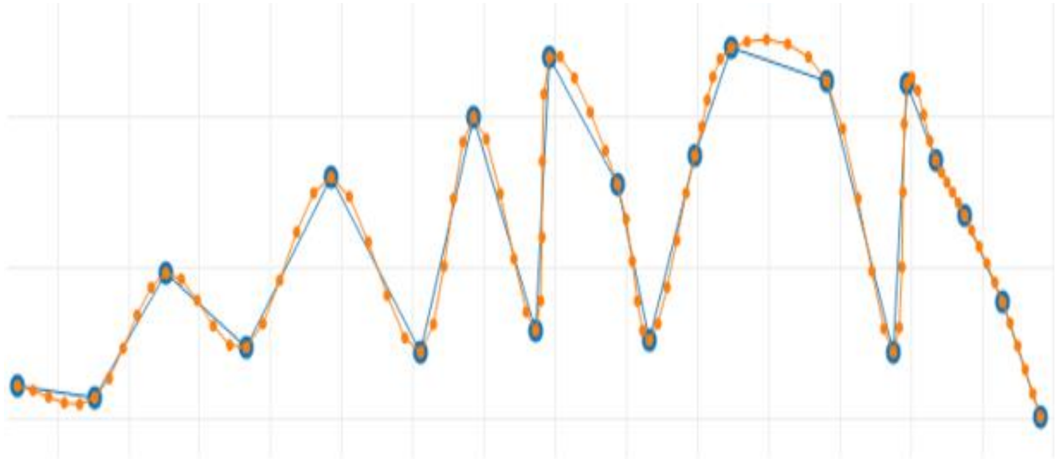

Рисунок 6 – График Catmull-Rom с использованием 100 точек

Выбор контрольных точек для кривых Безье также может занять много времени и требует тщательного рассмотрения, чтобы убедиться, что полученная кривая точно представляет данные, как показано на рисунке 7. Напротив, сплайны Catmull-Rom требуют только набора точек, и их легче вычислить, чем кривые Безье.

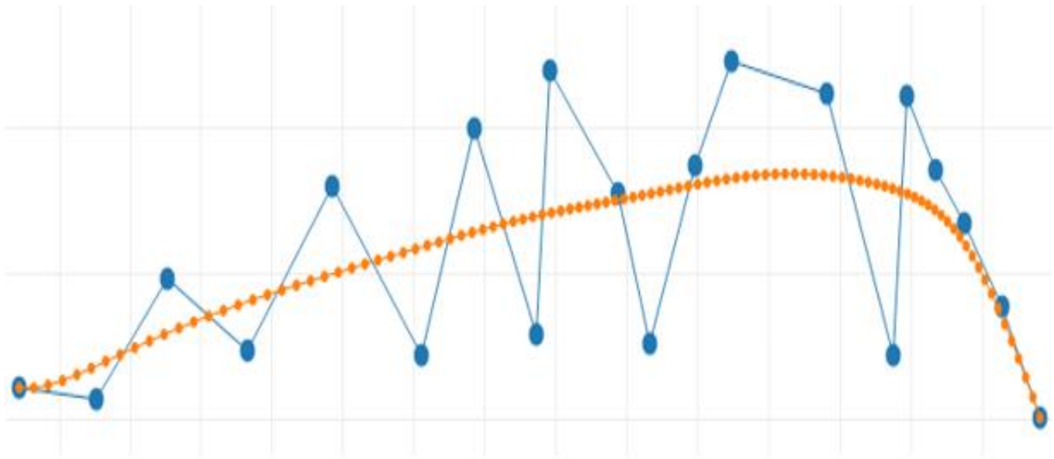

Рисунок 7 – График кривой Безье с использованием 100 точек

С точки зрения производительности сплайны Catmull-Rom могут быть медленнее кривых Безье для большого количества контрольных точек. Однако оба метода могут быть реализованы на GPU для ускорения процесса вычислений, особенно при обработке больших объемов данных.

Таким образом, сплайн-интерполяция Catmull-Rom и интерполяция кривой Безье имеют свои сильные и слабые стороны. Сплайны Catmull-Rom более гибкие и их легче вычислять, но они могут быть не такими гладкими, как кривые Безье. Кривые Безье более гладкие и содержат меньше артефактов, но требуют тщательного выбора контрольных точек и могут быть медленнее при большем количестве контрольных точек. Выбор между этими методами в конечном итоге зависит от конкретных требований приложения и характеристик, представляемых данных.

3.3 Последствия для будущей разработки мобильных приложений с применением алгоритмов сплайн-интерполяции

Будущие разработки мобильных приложений с использованием алгоритмов сплайновой интерполяции могут извлечь выгоду из достижений в области исследований сплайновой интерполяции. Использование алгоритмов сплайновой интерполяции при разработке мобильных приложений уже широко распространено и, как ожидается, будет расти по мере того, как мобильные устройства становятся более мощными и возрастает спрос на продвинутую графику и визуализации.

Одним из последствий для будущей разработки мобильных приложений является необходимость разработки более эффективных и адаптивных алгоритмов сплайновой интерполяции, которые могут более эффективно обрабатывать большие наборы данных и многомерные данные. Это может повысить производительность и масштабируемость мобильных приложений, использующих сплайновую интерполяцию, особенно тех, которые связаны с обработкой данных и визуализацией в реальном времени.

Другим следствием является необходимость включения оценок неопределенности в алгоритмы сплайновой интерполяции. Мобильные приложения, которые включают обработку данных датчиков, геопространственный анализ или научные вычисления, могут извлечь выгоду из возможности предоставления оценки неопределенности или ошибки в интерполированных значениях. Это может повысить точность и надежность мобильных приложений, что может иметь решающее значение во многих приложениях.

Кроме того, разработчики мобильных приложений могут изучить использование алгоритмов сплайновой интерполяции в новых областях, таких как машинное обучение, робототехника и компьютерное зрение. Эти области могут извлечь выгоду из способности алгоритмов сплайновой интерполяции обрабатывать данные с неравномерным интервалом и возможности распространения сплайновой интерполяции на неевклидовы пространства.

Наконец, разработчики мобильных приложений также могут исследовать оптимизацию алгоритмов сплайновой интерполяции для мобильных устройств. Поскольку мобильные устройства имеют ограниченную память и вычислительную мощность, оптимизация алгоритмов сплайн-интерполяции может повысить производительность и действенность мобильных приложений, использующих сплайн-интерполяцию.

Кроме того, будущая разработка мобильных приложений также может извлечь выгоду из использования сплайновой интерполяции в сочетании с другими методами, такими как машинное обучение и визуализация данных. Интегрируя сплайновую интерполяцию с алгоритмами машинного обучения, мобильные приложения могут делать точные прогнозы и классификации на основе неравномерно разнесенных данных. Это может быть полезно в таких приложениях, как прогнозирующее обслуживание, обнаружение мошенничества и аномалий.

Более того, мобильные приложения, использующие сплайновую интерполяцию, также могут извлечь выгоду из интеграции методов визуализации данных, таких как точечные диаграммы, контурные графики и тепловые карты. Визуализируя интерполированные данные, мобильные приложения могут предоставить пользователям лучшее понимание базовых моделей данных и тенденций. Это может быть полезно в таких приложениях, как прогнозирование погоды, прогнозирование трафика и финансовый анализ.

Еще одним следствием будущей разработки мобильных приложений с использованием алгоритмов сплайновой интерполяции является необходимость обеспечения конфиденциальности и безопасности пользовательских данных. Мобильным приложениям, использующим алгоритмы сплайновой интерполяции, часто требуется доступ к пользовательским данным, таким как данные датчиков, данные о местоположении или личная информация. Таким образом, разработчики мобильных приложений должны обеспечить защиту пользовательских данных от несанкционированного доступа или утечек данных. Этого можно достичь путем внедрения надежных методов шифрования, контроля доступа и механизмов получения согласия пользователя.

Наконец, использование алгоритмов сплайновой интерполяции при разработке мобильных приложений также имеет значение для дизайна пользовательского интерфейса. Разработчики мобильных приложений должны убедиться, что пользовательский интерфейс интуитивно понятен и удобен для пользователя, а интерполированные данные представлены в четкой и понятной форме. Этого можно достичь, используя соответствующие методы визуализации данных, предоставляя четкие инструкции и обратную связь, а также проводя тестирование пользователей для сбора отзывов и улучшения пользовательского опыта.

В заключение, будущая разработка мобильных приложений с использованием алгоритмов сплайн-интерполяции может выиграть от интеграции других методов, таких как машинное обучение и визуализация данных, обеспечения конфиденциальности и безопасности пользовательских данных и оптимизации дизайна пользовательского интерфейса. Учитывая эти последствия, разработчики мобильных приложений могут создавать мощные и эффективные мобильные приложения, которые используют преимущества алгоритмов сплайновой интерполяции.

3.4 Оценка применения алгоритмов сплайн-интерполяции на практике

Применение алгоритмов сплайновой интерполяции на практике получило широкое распространение в различных областях, таких как компьютерная графика, инженерия, геофизика, финансы и многие другие. Алгоритмы сплайновой интерполяции особенно полезны при работе с неравномерно расположенными данными или данными с высокой степенью шума. Они обеспечивают плавное и непрерывное представление данных, которыми можно легко манипулировать, анализировать и визуализировать.

Одним из главных преимуществ алгоритмов сплайновой интерполяции является их способность обрабатывать большие и сложные наборы данных. Это делает их особенно полезными в таких приложениях, как анализ данных, геопространственное моделирование и научные вычисления. Например, в геофизике алгоритмы сплайновой интерполяции используются для интерполяции сейсмических данных и создания трехмерных моделей недр. В финансах они используются для интерполяции финансовых данных, таких как цены на акции и процентные ставки, для оценки будущей стоимости и принятия обоснованных инвестиционных решений.

Более того, алгоритмы сплайновой интерполяции также широко используются в компьютерной графике, где они используются для создания плавных и реалистичных кривых и поверхностей. Это можно увидеть в таких приложениях, как анимация, видеоигры и виртуальная реальность, где алгоритмы сплайновой интерполяции используются для создания плавных движений камеры, анимации персонажей и моделирования окружающей среды.

Однако существуют также некоторые ограничения на применение алгоритмов сплайновой интерполяции на практике. Одним из главных ограничений является их подверженность переоснащению и недооснащению, особенно при работе с зашумленными данными. Это может привести к неточным результатам и ненадежным прогнозам, что может быть проблематично во многих приложениях. Чтобы решить эту проблему, важно тщательно выбрать подходящий алгоритм сплайновой интерполяции и настроить его параметры для достижения желаемого уровня точности и обобщения.

Другим ограничением является вычислительная сложность некоторых алгоритмов сплайновой интерполяции, которая может быть узким местом при работе с большими наборами данных. Это может привести к снижению производительности и КПД, особенно в приложениях реального времени. Чтобы преодолеть это, были разработаны различные методы оптимизации и аппроксимации, позволяющие снизить вычислительные затраты на алгоритмы сплайновой интерполяции.

Более того, эффективность алгоритмов сплайновой интерполяции на практике также зависит от качества и характеристик входных данных. Например, если входные данные сильно нерегулярны или содержат значительные отклонения, алгоритмы сплайновой интерполяции могут не давать точных или надежных результатов. Поэтому важно тщательно обработать и очистить данные перед применением алгоритмов сплайновой интерполяции.

Другим важным фактором, который следует учитывать при использовании алгоритмов сплайновой интерполяции на практике, является компромисс между точностью и вычислительными затратами. Некоторые алгоритмы сплайновой интерполяции могут давать более точные результаты, но требуют больше вычислительных ресурсов, в то время как другие могут давать менее точные результаты, но требуют меньше вычислительных ресурсов. Поэтому важно тщательно оценить компромисс между точностью и вычислительными затратами, исходя из конкретных требований приложения.

Кроме того, применение алгоритмов сплайновой интерполяции на практике также требует тщательного рассмотрения ограничений и допущений выбранного алгоритма. Например, некоторые алгоритмы сплайновой интерполяции могут предполагать определенный уровень гладкости или непрерывности данных, что может быть применимо не ко всем типам данных. Поэтому важно тщательно оценить допущения и ограничения выбранного алгоритма, основываясь на конкретных характеристиках данных.

Наконец, эффективность алгоритмов сплайновой интерполяции на практике также зависит от знаний и опыта пользователя. Алгоритмы сплайновой интерполяции могут давать сложные и нюансированные результаты, которые могут потребовать глубокого понимания основополагающих принципов и математики. Поэтому важно, чтобы пользователи имели глубокие знания в области математики, статистики и информатики, а также опыт работы со сплайновыми алгоритмами интерполяции.

В заключение, применение алгоритмов сплайновой интерполяции на практике может быть весьма эффективным и полезным, но требует тщательного рассмотрения различных факторов, включая качество и характеристики входных данных, компромисс между точностью и вычислительной стоимостью, ограничения и допущения выбранного алгоритма, а также знания и опыт пользователя. Тщательно оценивая эти факторы и выбирая соответствующий алгоритм сплайновой интерполяции, пользователи могут получать точную, надежную и ценную информацию и решения сложных проблем с данными.

Вывод по третьей главе

Глава описывает результаты тестирования работы мобильного приложения с применением сплайн-интерполяции. Оценивается производительность алгоритмов сплайн-интерполяции в мобильном приложении, а также проводится сравнение различных алгоритмов. Рассматриваются возможные последствия для будущей разработки мобильных приложений с применением алгоритмов сплайн-интерполяции, оценивается применение этих алгоритмов на практике.

Таким образом, глава позволяет сделать выводы о том, что использование алгоритмов сплайн-интерполяции в мобильных приложениях может существенно улучшить качество и производительность приложений, особенно в задачах, связанных с интерполяцией данных. Однако, необходимо учитывать особенности различных алгоритмов и выбирать наиболее подходящий в каждой конкретной задаче. Также важно тестировать работу приложений с применением сплайн-интерполяции для обеспечения их эффективности и стабильности.

Заключение

В заключение, сплайн-алгоритмы являются мощными инструментами интерполяции, и их применение при разработке мобильных приложений огромно. Они эффективно аппроксимируют кривые и поверхности с высокой точностью, что делает их подходящими для различных приложений, таких как моделирование, симуляция и визуализация. Алгоритмы сплайнов широко используются в компьютерной графике, играх и анимации, среди прочих областей, для создания плавных и реалистичных визуальных эффектов.

В контексте разработки мобильных приложений сплайн-алгоритмы имеют широкий спектр применения. Их можно использовать для создания плавных анимаций, интерполяции точек данных и аппроксимации кривых и поверхностей. В последние годы наблюдается значительный рост разработки мобильных приложений, и использование сплайн-алгоритмов стало более распространенным из-за их эффективности в улучшении пользовательского опыта и повышении общей производительности приложений.

В целом, анализ сплайн-алгоритмов интерполяции и их применение при разработке мобильных приложений – увлекательная тема. Очевидно, что сплайн-алгоритмы обладают значительным потенциалом и могут играть жизненно важную роль в различных областях, включая разработку мобильных приложений. Поскольку технология продолжает развиваться, вполне вероятно, что использование сплайновых алгоритмов в мобильных приложениях станет более распространенным, что приведет к дальнейшим инновациям в этой области.

Разработано и протестировано приложение, реализующее алгоритмы сплайн-интерполяции Catmull-Rom и кривой Безье. Показан результат работы в самом приложении. Также проанализирована производительность и гибкость данных алгоритмов.

Список используемой литературы и используемых источников

1. Алберг Дж. Теория сплайнов и ее приложение / Дж. Алберг, Э. Нильсон, Дж. Уолш; пер. с англ. Ю. Н. Субботина; под ред. С. Б. Стечкина; с доб. С. Б. Стечкина, Ю. Н. Субботина. – Москва: Мир, 1972. – 315 с. : ил. – Библиогр. : с. 267-269. – Предм. указ.: с. 310-311. – Имен указ.: с. 312-313 URL: https://reallib.org/reader?file=466782
2. Карпова А.П., Силаева М.Н. Интерполяция алгебраическими многочленами. Сплайн-интерполяция: учебное пособие / Министерство науки и высшего образования РФ, Федеральное государственное бюджетное образовательное учреждение высшего образования "Воронежский государственный университет": Издательский дом ВГУ, 2022. – 64 с. URL: https://studfile.net/preview/16724010/
3. Б. В. Соболь, Б. Ч. Месхи, И. М. Пешхоев. Практикум по вычислительной математике. – Ростов-на-Дону: Феникс, 2008 – 208 с. URL: https://djvu.online/file/vKTcIzCSLJyqI
4. Де Бур К. Практическое руководство по сплайнам. - М.: Финансы и статистика, 1984, - 306 с. URL: https://reallib.org/reader?file=792215
5. Пулькин С. П. Вычислительная математика : учеб. пособие для вузов / С. П. Пулькин, Л. Н. Никольская, А. С. Дьячков. - Москва : Просвещение, 1980. - 176 с. URL: https://www.nehudlit.ru/books/detail1184968.html
6. Вычислительная математика [Электронный ресурс] : учеб. пособие в двух частях. Ч. 1 / В. Н. Варапаев [и др.]. - Москва : МГСУ : Ай Пи Эр Медиа, 2017. - 88 с. - (Прикладная математика). - ISBN 978-5-7264-1455-3 URL: https://search.rsl.ru/ru/record/01009642508
7. Завьялов, Ю.С. Методы сплайн-функций / Ю.С. Завьялов, Б.И. Квасов, В.Л. Мирошниченко – М.: Наука, 1980, - 113с. URL: http://old.math.nsc.ru/conference/msf11/msf11_abstracts.pdf

8. Стечкин, С.Б. Сплайны в вычислительной математике / С.Б. Стечкин, Ю.Н. Субботин – М.: Наука, 1976, - 248 с. URL: https://forkettle.ru/biblioteka/estestvoznanie/matematika/159-vychislitelnaya-matematika/1578-splajny-v-vychislitelnoj-matematike

9. Квасов, Б.И. Методы изогеометрической аппроксимации сплайнами / Б.И. Квасов – М.: Физматлит, 2006, - 306 с. URL: https://istina.msu.ru/publications/book/63685192/

10. Корнейчук Н. П. Сплайны в теории приближения / Н. П. Корнейчук. - Москва : Наука, 1984. - 352 с. - Библиогр.: с. 343-349. - Предм. указ.: с. 350-351 URL: https://edu.tltsu.ru/elib/find.php

11. Макаров В. Л. Сплайн-аппроксимация функций : учеб. пособие для вузов / В. Л. Макаров, В. В. Хлобыстов. - Москва : Высш. шк., 1983. - 80 с. - Библиогр.: с. 80. URL: https://edu.tltsu.ru/elib/find.php

12. Хокинг Дж. Unity в действии : Мультиплатформенная разработка на C# / Дж. Хокинг. - Санкт-Петербург : Питер, 2016. - 334 с. : ил. - Прил.: с. 321-334. - ISBN 978-5-496-01960-6: 864-00. URL: https://edu.tltsu.ru/elib/find.php

13. C# для чайников. Мюллер Дж. М.: 2019. – 608 с. URL: https://www.at.alleng.org/d/comp/comp612.htm

14. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов. Марк Дж. Прайс. 2018. – 640 с. URL: https://sd.blackball.lv/books/17527?mode=read

15. Unity и C#. Геймдев от идеи до реализации. Джереми Гибсон Бонд. 2019. – 928 с. URL: pdfdrive.com/unity-и-с-Геймдев-от-идеи-до-реализации-e187417665.html

16. Троелсен и Джепикс. Язык программирования C# 7 и платформы .NET и .NET Core. 2018. – 1330 с. URL: https://sd.blackball.lv/books/17526?mode=read

17. Албахри Дж. C# 7 карманный справочник. 2017 – 226 с. URL: https://library.kre.dp.ua/Books/2-

4%20kurs/Програмування%20%2B%20мови%20програмування/C%23/Албахари_Дж_C%23%207.0_карманный_справочник_2017.pdf

18. Джозеф Хокинг. Unity в действии. Мультиплатформенная разработка на C#, 2019. – 352 с. URL: https://ru.pdfdrive.com/unity-в-действии-Мультиплатформенная-разработка-на-с-e187707045.html

19. Мэннинг Д., Батфилд-Эддисон П.: Unity для разработчика. Мобильные мультиплатформенные игры, 2018. – 352 с. URL: https://ru.pdfdrive.com/unity-для-разработчика-Мобильные-мультиплатформенные-игры-e183885794.html

20. C# 7.0. Справочник. Полное описание языка | Албахари Бен, Албахари Джозеф, 2000. – 1024 с. URL: https://www.goodreads.com/book/show/51834428-c-7-0

21. Larry L. Schumaker. Spline Functions: Basic Theory. Wiley-Interscience, New York, 1981. URL: https://archive.org/details/splinemodelsforo0000wahb

22. Carl R de Boor. A Practical Guide to Spline, 1978. – 341 с. URL: https://www.researchgate.net/publication/200744645_A_Practical_Guide_to_Spline

23. Harrison Ferrone. Learning C# by Developing Games with Unity – Seventh Edition, 2022. – 458 с. URL: https://www.packtpub.com/product/learning-c-by-developing-games-with-unity-seventh-edition/9781837636877

24. Robert Nystrom. Game Programming Patterns, 2014. – 354 с. URL: https://freecomputerbooks.com/Game-Programming-Patterns.html

25. Simon Jackson. Mastering Unity 2D Game Development, 2014. – 474 с. URL:https://www.packtpub.com/product/mastering-unity-2d-game-development/9781849697347