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 КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Разработка программного обеспечения для реалистичной визуализации плечевой одежды на примере футболки»

Студент: ИУ7-53Б _____ М. Д. Маслова
(группа) (подпись, дата) (И. О. Фамилия)

Руководитель: _____ А. А. Оленев
(подпись, дата) (И. О. Фамилия)

2021 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современные исследования в области компьютерной графики сосредоточены на моделировании и визуализации явлений реального мира с максимальной реалистичностью. Моделирование одежды и, как более общего случая, ткани играют не последнюю роль в детализации виртуальных сред [1]. Реалистичный вид одежды придает выразительности анимационным персонажам в компьютерных играх и мультипликации [2]; в фильмах помогает сделать неотличимыми реального человека, снятого на камеру, от цифрового дублера — виртуальной реалистичной копии, которая «выполняет» сложные и опасные для жизни трюки [3]. Также сегодня развивается идея виртуальной примерочной в интернет-магазинах [4]. Все это показывает практическую применимость моделирования одежды, а следовательно, необходимость разработки методов ее визуализации.

Ткань, основа одежды, является материалом с уникальными свойствами: гибкостью и изменением формы при небольшом воздействии [5]. Они вносят в рассматриваемые явления хаотичность, что замечается в реальной жизни: каждый раз, когда человек надевает тот или иной элемент одежды, многие детали выглядят по-разному [6]. Перечисленные свойства усложняют задачу моделирования тканных материалов по сравнению с моделированием твердых тел [7]. Стоит отметить также разные цели моделирования ткани. Так, в анимации акцент делается на внешний вид конечного результата, в то время как в инженерном сообществе, которое также работает с тканными материалами, ценится физическая точность [3]. Все выше перечисленное приводит к тому, что существует большое количество методов визуализации ткани, использующихся в каждой конкретной ситуации. В данной курсовой работе ставится цель получения изображения одежды и достижения его реалистичности.

Цель работы — разработать программное обеспечение для реалистичной визуализации плечевой одежды на примере футболки, предоставляющее возможность изменения положения камеры (перемещение, вращение, масштабирование).

Для достижения поставленной цели необходимо решить следующие задачи:

– формально описать модель ткани, как части одежды;

– проанализировать методы визуализации ткани и соединения ее частей для получения одежды;
– разработать и реализовать алгоритм визуализации футболки;
– оценить производительность реализованного алгоритма.

1 Аналитическая часть

1.1 Одежда, как объект физического мира

Любая одежда: футболки, брюки, носки, шарфы и куртки и т. д., — представляет собой одну или несколько соединенных между собой деталей из ткани. Ткань, в свою очередь, состоит из натуральных или искусственных волокон или нитей, которые производятся путем прядения различных материалов, таких как шерсть, хлопок, лен и т. п. Для соединения полученных волокон используют следующие техники:

- ткачество — изготовление ткани путем переплетения нитей под прямым углом (рисунок 1.1);
- вязание — соединение волокон между собой путем образования и протягивания петель (рисунок 1.2);
- макраме — закрепление нитей с помощью узлов;
- получение войлока — прессование волокон животных.

После применения одной из этих техник получается готовая ткань.

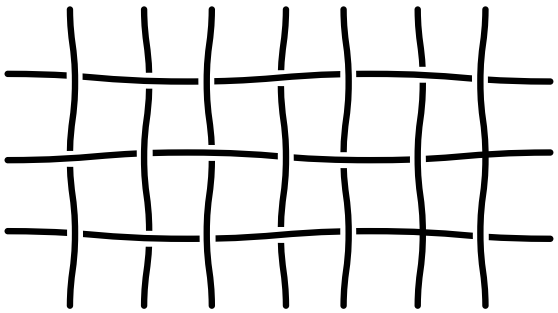

Рисунок 1.1 – Строение тканного полотна [8]

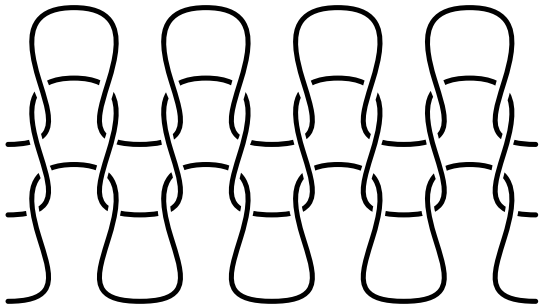

Рисунок 1.2 – Строение вязаного полотна [8]

Разные волоконные материалы и техники их соединения являются причиной различной степени проявления разными тканями основных механических свойств: растяжения, сдвига и изгиба. Отсутствие у тканных материалов упругих свойств приводит к образованию складок и легком драпировании на другие объекты. Таким образом, разное строение ткани в зависимости от типа волокон, большое количество узлов, из которых она состоит, разные степени проявления механических свойств, большое количество степеней свободы – все это следует учитывать при моделировании ткани. Вследствие отсутствия

возможности учесть все необходимые параметры ткани разрабатываются упрощенные модели и методы их реализации для решения конкретных задач: в одних целью является максимальная реалистичность изображения, в других — высокая скорость работы с наименьшей потерей реалистичности [9].

1.2 Методы визуализации одежды

Как уже было сказано выше, одежда является более сложной формой ткани, поэтому далее будут рассмотрены методы моделирования тканных материалов. Данные методы можно разделить на два основных типа:

- геометрические методы;
- физические методы.

1.2.1 Геометрические методы

Геометрические методы [7] не учитывают физические свойства ткани, они фокусируются на воспроизведении внешнего вида тканных материалов с помощью представления поверхности математическими функциями. Таким образом, в данных методах не требуется решение сложных систем уравнений, что дает им преимущество в виде большой скорости выполнения.

Хотя геометрические методы за короткое время могут с достаточной долей реалистичности визуализировать ткань, каждый из них либо решает достаточно специфическую задачу, например, воспроизведение висящей ткани или моделирование складок на рукаве, либо нуждается в активном содействии пользователя, что уменьшает количество сфер, в которых их можно применить.

1.2.1.1 Метод моделирования свисающей ткани

Метод моделирования свисающей ткани [10] предназначен для моделирования тканного материала, который закреплен на некотором количестве точек. Ткань считается прямоугольной и представляется в виде сетки, а моделирование выполняется в два этапа:

1) На первом этапе каждая пара заданных точек соединяется цепной кривой, представленной формулой (1.1):

$$y = a\cosh(\frac{x}{a}), \tag{1.1}$$

где a — коэффициент масштабирования.

Если при этом проекции двух кривых на плоскость XOZ пересекаются как показано на рисунке 1.3, то для устранения дополнительных вычислений, нижняя кривая удаляется. После завершения этого этапа модель представляет изогнутый каркас, форма которого только приближается к структуре ткани.

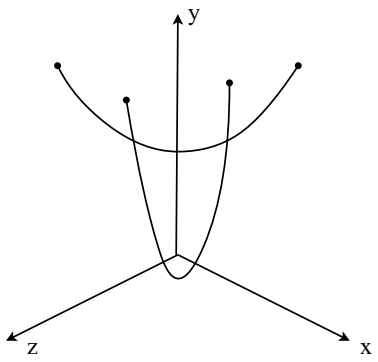

Рисунок 1.3 – Две пересекающиеся цепные кривые

2) Для получения более точной модели добавляются новые поверхности, которые создаются путем разделения треугольников, образованных цепными кривыми, вычисленных на предыдущем этапе. Новые треугольники также разделяются. Итерационный процесс продолжается до тех пор, пока максимальное смещение точек сетки за один проход не станет меньше заданного значения.

После завершения двух этапов полученная модель визуализируется с предварительным нанесением на нее сплайновых кривых для получения гладкого изображения ткани [7; 10].

1.2.1.2 Метод моделирования складок на рукаве

Метод моделирования складок на рукаве [7] ориентирован под конкретную задачу, а именно моделирование рукава на сгибающейся руке. Ткань представляется в виде полого цилиндра, состоящего из набора окружностей R_i (рисунок 1.4). Складки образуются в том случае, если модуль разности расстояний между точками двух соседних окружностей до (L_0) и после ($L_{i,j}$) деформации (рисунок 1.5) меньше некоторого заранее заданного порогового значения.

Складки моделируются путем преобразования окружностей в многоугольники и изменения положения их вершин [7].

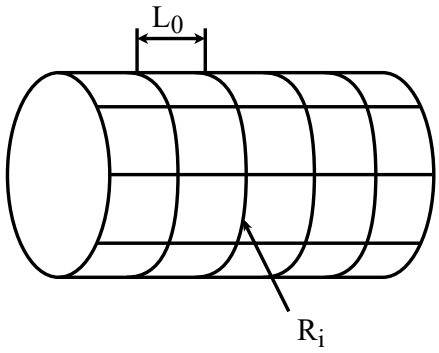

Рисунок 1.4 – Модель рукава: полый цилиндр из набора окружностей

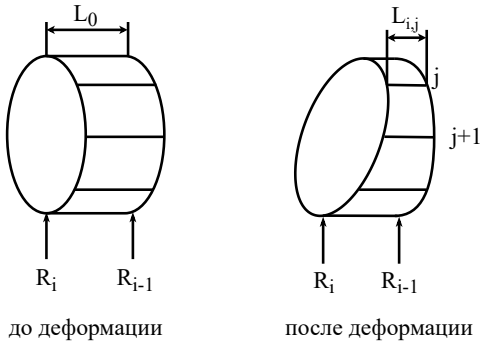

Рисунок 1.5 – Одна секция рукава до и после деформации

1.2.1.3 Методы со значительной степенью вмешательства пользователя

Методы со значительной степенью вмешательства пользователя [11; 12] разрабатывались специально под графические редакторы. Основная их идея состоит в том, чтобы изначально представить одежду как полностью прилегающую к телу или расположенную на небольшом расстоянии с повторением контуров тела, а далее предоставить пользователю интерфейс для редактирования положения ткани. Так как в данных методах реалистичность итогового изображения во многом зависит от пользователя, а в этой работе ставится задача получения реалистичного изображения без его вмешательства, более подробное описание этих методов здесь не приводится.

1.2.2 Физические методы

В физических методах [7] модель ткани представляют в виде треугольных или прямоугольных сеток с точечными массами в узлах. Взаимодействие между этими массами описываются различными способами в зависимости от метода. В моделях, основанных на энергии, положение точки определяется энергетическим состоянием системы, а именно: ищется такое состояние ткани, в котором энергия системы минимальна. В других моделях силы взаимодействия между точечными массами описываются дифференциальными уравнениями, решение которых производится с помощью численного интегрирования, в результате чего получают координаты точки.

Так как в физических методах проводится большое количество вычислений: решение системы дифференциальных уравнений или перебор состояний системы для поиска минимумов энергии, — скорость их выполнения ниже, чем у геометрических методов. Однако физические методы предоставляют большую свободу: мы можем создать реалистичное изображение без привлечения пользователя, а также можем смоделировать разные виды ткани, изменяя физические характеристики (например, увеличение значения массы в узлах приведет к утяжелению ткани), что также позволяет сделать модель более правдоподобной.

1.2.2.1 Модель сплошной среды

В модели сплошной среды ткань рассматривается в виде сплошной, однородной структуры. Ее поведение моделируется с помощью теории упругости, из физически обоснованных выражений которой выводится большая система обыкновенных дифференциальных уравнений. Решение этой системы производится численно. Такой подход позволяет модели естественным образом реагировать на приложенные силы, окружающую среду и другие объекты, однако требует дорогих вычислительных затрат [13; 14].

1.2.2.2 Энергетическая модель системы частиц

В методе энергетической модели системы частиц [15] точки пересечения нитей ткани рассматриваются как точечные массы или частицы, а моделирование состоит из двух этапов. На первом этапе учитывается гравитация и определяются любые столкновения с объектами или землей. Позиции частиц определяются с помощью уравнения, описывающегося формулой (1.2):

$$m\vec{a} + c\vec{v} = m\vec{g}, \tag{1.2}$$

где m — масса частицы,

$\vec{a}$ — вектор ускорения,

c — сопротивление воздуха,

$\vec{v}$ — вектор скорости,

$\vec{g}$ — вектор ускорения свободного падения.

После первого этапа получается грубая модель ткани. Для получения реалистичного изображения вводят второй этап, на котором минимизируется энергия системы частиц, где энергия каждой частицы U_i представляется, как сумма энергий основных взаимодействий отталкивания U_{repel_i}, растяжения $U_{stretch_i}$, сдвига U_{shear_i}, изгиба U_{bend_i} и гравитации $U_{gravity_i}$, что описано формулой (1.3) [7]:

$$U_i = U_{repel_i} + U_{stretch_i} + U_{shear_i} + U_{bend_i} + U_{gravity_i}. \tag{1.3}$$

С помощью этого метода достигается реалистичность итогового изображения, однако для поиска минимума энергии требуются большие временные затраты [9].

1.2.2.3 Массо-пружинная модель

В массо-пружинной модели ткань также представляется в виде сетки с точечными массами в узлах, но частицы между собой связаны пружинами, которые отвечают за упругое поведение материала (рисунок 1.6). Такое представления ткани практично, так как криволинейные поверхности часто представляются полигональными сетками, вершины которых можно рассматривать, как точечные массы, а ребра — как пружины [14]. Каждая пружина в

этой модели относится к одному из типов [16]:

- пружины структуры ткани соединяют частицы с индексами $[i, j]$ и $[i + 1, j]$, а также $[i, j]$ и $[i, j + 1]$;
- пружины сдвига ткани соединяют частицы с индексами $[i, j]$ и $[i + 1, j + 1]$, а также $[i + 1, j]$ и $[i, j + 1]$;
- пружины изгиба ткани соединяют частицы с индексами $[i, j]$ и $[i+2, j]$, а также $[i, j]$ и $[i, j + 2]$.

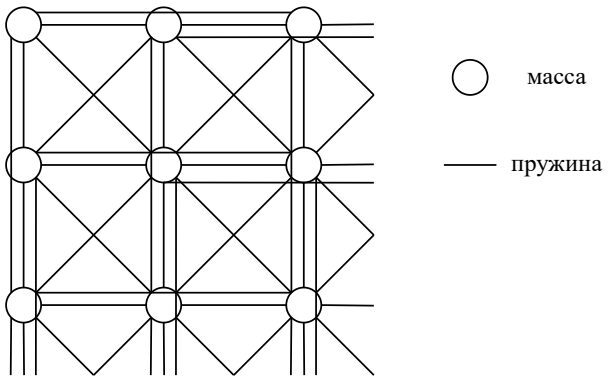

Рисунок 1.6 – Сетка масс и пружин

После представления ткани в виде сетки для каждой точки вычисляется результирующая сила, вычисляемая путем сложения внутренних и внешних сил, а положение точки определяется путем явного интегрирования полученных уравнений [9]. Так как данный метод предназначен для анимации ткани, скорость вычислений в нем выше, чем в ранее описанных методах [16].

Вывод

На основе поставленных целей и задач для визуализации одежды была выбраны критерии оценки описанных методов (универсальность, необходимость вмешательства пользователя и скорость вычислений) и составлена таблица 1.1 сравнения, после анализа которой выбрана массо-пружинная модель ткани. Этот метод, как физический, имеет преимущество перед геометрическими, так как он универсален, а реалистичность итогового изображения не зависит от действий пользователя. Среди физических эта модель выделяется скоростью работы, которая требуется для создания динамического изображения и взаимодействия с пользователем. Также эта модель является интуитивно понятной в силу базовых физических законов, лежащий в ее основе, что является преиму-

ществом как для реализации, так и для построения понятного для пользователя интерфейса.

Таблица 1.1 – Сравнение методов моделирования ткани

Метод	**Универсальность**	**Вмешательство пользователя**	**Скорость**
Свисающей ткани	-	не требуется	высокая
Складок на рукаве	-	не требуется	высокая
С вмешательством пользователя	+	требуется	низкая
Сплошной среды	+	не требуется	низкая
Энергетический	+	не требуется	низкая
Массо-пружинная модель	+	не требуется	средняя

1.3 Описание реализуемого метода

В качестве реализуемого метода была выбрана массо-пружинная модель [16], в которой ткань представляется в виде сетки точечных масс, соединенных пружинами так, как показано на рисунке 1.7.

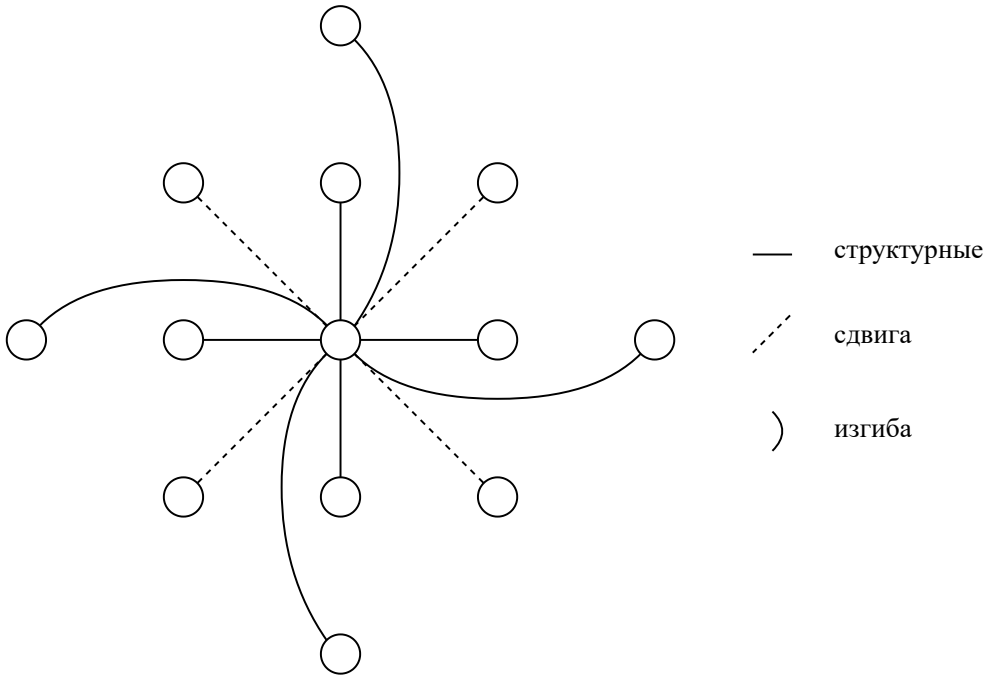

Рисунок 1.7 – Пружины точечной массы

Моделирование ткани происходит с шагом по времени Δt. На каждом шаге определяется новое положение каждой точечной массы на основе предыдущего положения точки и шага Δt.

На каждую точечную массу действуют внутренние и внешние:

– сила упругости со стороны каждой из связанных с данной точечной массой пружин, определяющаяся формулой (1.4):

$$\vec{F}_{\text{упр}} = k \cdot [\vec{l} - l_0 \frac{\vec{l}}{||\vec{l}||}], \tag{1.4}$$

где k — коэффициент жесткости пружины;

$\vec{l}$ — вектор от данной точечной массы к той, с которой ее соединяет пружины (его модуль определяет текущую длину пружины);

l_0 — длина пружины в недеформированном состоянии.

– сила сопротивления воздуха, вычисляемая по формуле (1.5):

$$\vec{F}_{\text{сопр}} = -C_{\text{сопр}}\vec{v}, \tag{1.5}$$

где $C_{\text{сопр}}$ — коэффициент сопротивления;

$\vec{v}$ — текущая скорость точечной массы.

– сила тяжести (формула (1.6)):

$$\vec{F}_{\text{тяж}} = m\vec{g}, \tag{1.6}$$

где m — масса точки;

$\vec{g}$ — ускорение свободного падения.

– сила ветра, каждая из трех составляющих вектора которой представляется комбинацией тригонометрических функций, зависящих от координат точечной массы и времени.

Результирующая сила $\vec{F}$, действующая на точечную массу, определена как векторная сумма всех действующих сил.

По второму закону Ньютона определяет ускорение данной точки (формула (1.7)), которая является второй производной координаты по времени.

$$\vec{a} = \frac{\vec{F}}{m}, \tag{1.7}$$

где $\vec{a}$ — ускорение точечной массы;

m — масса точки.

Новое положение точки определяется путем численного интегрирования ускорения. В качестве метода численного интегрирования может использоваться метод Эйлера в данном случае представляющий уравнения движения, описывающиеся формулой (1.8).

$$\begin{cases} \vec{a}_{t+\Delta t} = \frac{\vec{F}_t}{m} \\ \vec{v}_{t+\Delta t} = \vec{v}_t + \Delta t \vec{a}_{t+\Delta t} \\ \vec{P}_{t+\Delta t} = \vec{P}_t + \Delta t \vec{v}_{t+\Delta t} \end{cases}, \tag{1.8}$$

где $\vec{a}_{t+\Delta t}, \vec{v}_{t+\Delta t}, \vec{P}_{t+\Delta t}$ — ускорение, скорость и радиус-вектор точечной массы на текущем шаге;

$\vec{F}_t, \vec{v}_t, \vec{P}_t$ — сила, скорость и радиус-вектор точечной массы на предыдущем шаге.

Для повышения стабильности системы в данной работе используется метод численного интегрирования Верле [1]. Получение формулы для определения нового положения методом Верле из уравнения движения определяется последовательностью преобразований, описывающихся формулами (1.9)-(1.11). Таким образом, значение координаты точечной массы на каждом шаге определяется по формуле (1.12).

$$\vec{P}_{t+\Delta t} = \vec{P}_t + \Delta t \vec{v}_t + \frac{\vec{a}_{t+\Delta t} \Delta t^2}{2}, \tag{1.9}$$

$$\vec{v}_t = \frac{\vec{P}_t - \vec{P}_{t-\Delta t}}{\Delta t}, \tag{1.10}$$

где $\vec{P}_{t-\Delta t}$ — радиус-вектор точечной массы на два шага ранее.

$$\vec{P}_{t+\Delta t} = \vec{P}_t + \vec{P}_t - \vec{P}_{t-\Delta t} + \frac{\vec{a}_{t+\Delta t} \Delta t^2}{2}, \tag{1.11}$$

$$\vec{P}_{t+\Delta t} = 2\vec{P}_t - \vec{P}_{t-\Delta t} + \frac{\vec{a}_{t+\Delta t} \Delta t^2}{2}. \tag{1.12}$$

1.4 Формализация модели

Основным элементом сцены является футболка, которая, как и в реальной жизни, состоит из нескольких частей — выкроек. Для упрощения моделирования футболка представляется в виде двух одинаковых частей, каждая из которых является сеткой из масс и пружин. В силу возможности первоначального расположения масс по квадратной сетки, форма выкроек задается матрицей, в которой единица соответствует наличию массы, ноль — ее отсутствию. Соединение частей происходит с помощью введения дополнительных пружин у частиц расположенных по местам швов.

Свойства тканного материала могут быть описаны массой узлов сетки и жесткостью пружин, а также коэффициентами рассеянного и зеркального отражения света от точечного источника света.

Таким образом, модель футболки описывается следующими параметрами:

- форма частей — матрица из нулей и единиц;
- масса узла сетки m — вещественное число от 0.001 до 1 (граммы);
- жесткость пружины k — целое число от 100 до 25000 (ньютон/метр);
- коэффициент рассеянного отражения — вещественное число от 0 до 1;
- коэффициент зеркального отражения — вещественное число от 0 до 1.

1.5 Методы рендеринга изображения

Рендеринг (rendering, визуализация) [17] — в компьютерной графике, процесс преобразования цифровых моделей в визуализируемое представление — изображение.

Реализуемым методом моделирования одежды выбрана массо-пружинная модель, в которой используется большое количество узлов. Отрисовка модели одежды должна происходить с такой скоростью, чтобы обеспечить непрерывное движение, а также взаимодействие с пользователем. Стандартной графической библиотеки с программной реализацией алгоритмов отрисовки не сможет обеспечить необходимую скорость работы, поэтому необходимо выбрать прикладной программный интерфейс (API), который позволит получить необходимую скорость синтеза изображения. Далее рассматриваются такие API, как OpenGL, Vulkan и DirectX.

1.5.1 OpenGL

OpenGL (Open Graphics Library — «Открытая Графическая Библиотека») [18] — это прикладной программный интерфейс для разработки приложений в области двумерной и трехмерной графики. OpenGL имеет продуманную внутреннюю структуру и процедурный интерфейс, с помощью которых можно создавать сложные и мощные программные комплексы, затрачивая при этом минимальное время по сравнению с другими графическими библиотеками. Рендеринг изображения базируется на конвейерной обработке, то есть графические данные прежде, чем стать конечным изображением, проходят несколько этапов обработки.

Преимущества:

– кроссплатформенность;
– открытый код;
– высокая производительность;
– поддержка многими языками программирования и аппаратными графическими устройствами.

Недостатки:

– сложность работы с новыми поколениями GPU;
– отсутствие обработки устройств ввода-вывода и поддержки менеджера окон.

1.5.2 Vulkan

Vulkan [18] — это первый открытый низкоуровневый API, который предоставляет разработчикам прямой доступ к GPU для полного контроля над его работой. Как и OpenGL, Vulkan — это бесплатный стандарт с открытым кодом, доступный для любой платформы.

Преимущества:

– кроссплатформенность;
– открытый код;
– высокая производительность.

Недостатки:

– работа напрямую с GPU (сложно для неподготовленного программи-

ста);

– отсутствие поддержка многими языками программирования.

1.5.3 DirectX

DirectX [18] — это набор API, разработанных для решения задач компьютерной графики под управлением Microsoft Windows. Наиболее широко используется при написании компьютерных игр.

Преимущества:

– высокая производительность;

– высокое качество изображения.

Недостатки:

– работа только под управлением Microsoft Windows;

– отсутствие поддержки старого оборудования новыми версиями.

Вывод

С учетом представленных описаний программных интерфейсов выбраны критерии их оценки (кроссплатформенность, открытый код, поддержка языком Python) и составлена таблица 1.2 сравнение, после анализа которой для визуализации модели одежды был выбран OpenGL, он является кроссплатформенным, что дает ему преимущество перед DirectX, и поддерживается многими языками программирования, что дает ему преимущество перед Vulkan.

Таблица 1.2 – Сравнение графических API

API	**Кроссплатформенность**	**Открытый код**	**Поддержка Python**
OpenGL	+	+	+
Vulkan	+	+	-
DirectX	-	-	+

1.6 Существующие программные обеспечения

Реалистичная визуализация одежды является востребованной в современном мире. Так, она используется в профессиональных коммерческих про-

граммных обеспечениях для индустрии моды, таких как Clo [19] (рисунок 1.8), Browzwear [20] (рисунок 1.9).

Рисунок 1.8 – Пример работы программы Clo

Рисунок 1.9 – Пример работы программы Browzwear

Не являются исключением и бесплатные кроссплатформенные приложения для работы с 3D графикой, таких как Blender. Данное приложение предоставляет большой функционал для моделирования, симуляции, рендеринга, монтажа, записи видео и создания игр. Дополнения к Blender упрощают моделирование тех, или иных объектов. В том числе существует дополнение для моделирования ткани и одежды, пример которого представлен на рисунке 1.10.

Рисунок 1.10 – Пример визуализации одежды в Blender

Также существует аппаратная реализация визуализации одежды PhysX Clothing [21], пример работы которой представлен на рисунке 1.11.

Рисунок 1.11 – Пример визуализации одежды PhysX Clothing

Вывод

В данном разделе были описаны методы визуализации ткани, как части одежды, проанализированы методы рендеринга изображения, формально описана модель футболки.

В качестве реализуемого метода выбрана и описана массо-пружинная модуль, в качестве методы рендеринга изображения выбран OpenGL.

2 Конструкторская часть

2.1 Требования к программному обеспечению

Разрабатываемое программное обеспечение должно предоставлять следующие функциональности:

– изменение параметров одежды (массы узлов, жесткости пружин, цвета, коэффициентов рассеянного и зеркального отражения) в интерактивном режиме;

– изменение параметров окружающей среды (гравитации, ветра);

– вращение, перемещение и масштабирование камеры.

При этом программа должна удовлетворять следующим требованиям:

– время отклика программы должно быть не более 1 секунды для корректной работы в интерактивном режиме;

– программа должна корректно реагировать на любые действия пользователя.

2.2 Разработка алгоритмов

В данном подразделе представлены схемы выбранных алгоритмов.

2.2.1 Алгоритм моделирования ткани

На рисунке 2.1 представлена схема алгоритма вычисления нового положения точечной массы. На рисунке 2.2 представлена схема алгоритма вычисления результирующей силы, действующей на точечную массу.

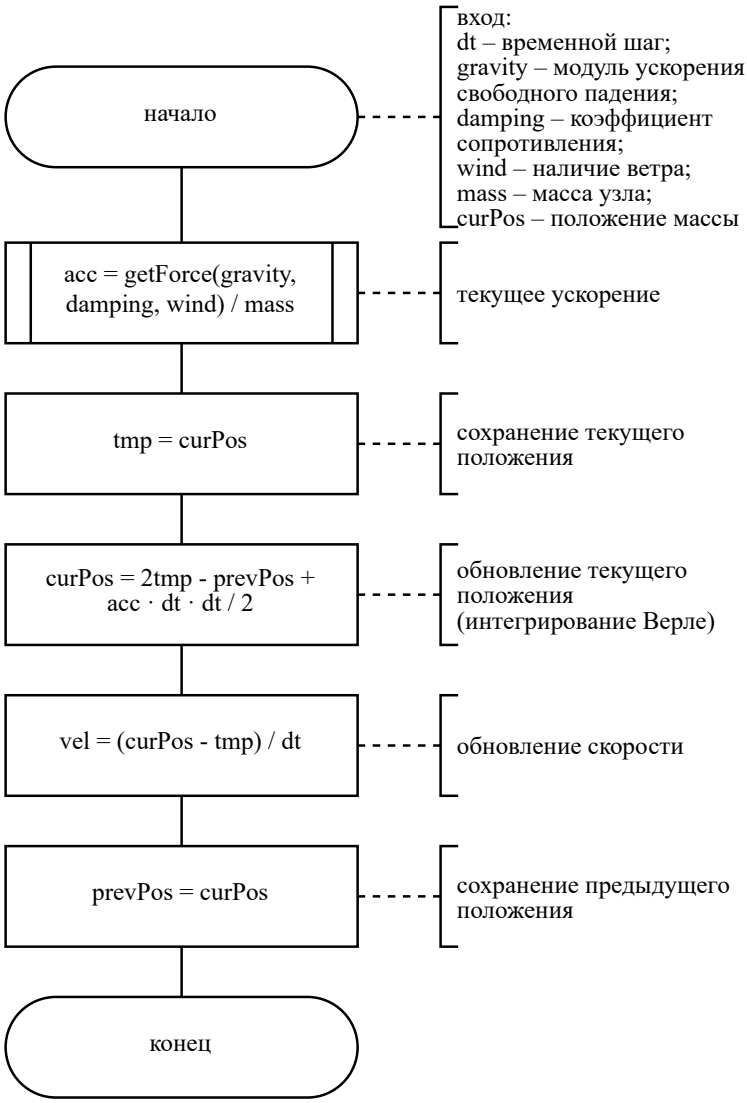

Рисунок 2.1 – Схема алгоритма вычисления нового положения точечной массы

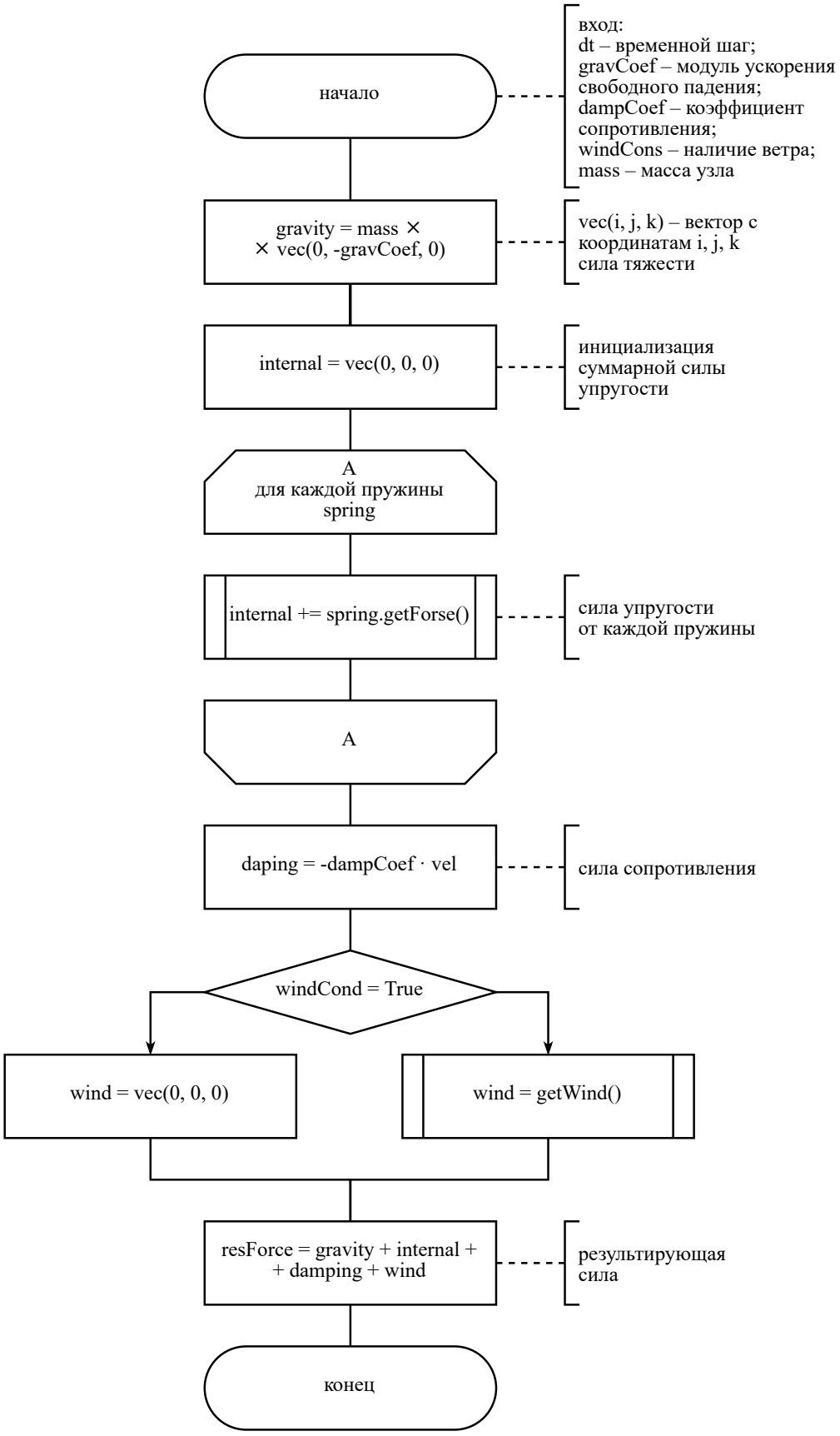

Рисунок 2.2 – Схема алгоритма вычисления результирующей силы, действующей на точечную массу

2.2.2 Алгоритм синтеза изображения

На рисунке 2.3 представлена схема алгоритма синтеза изображения с помощью выбранного API OpenGL.

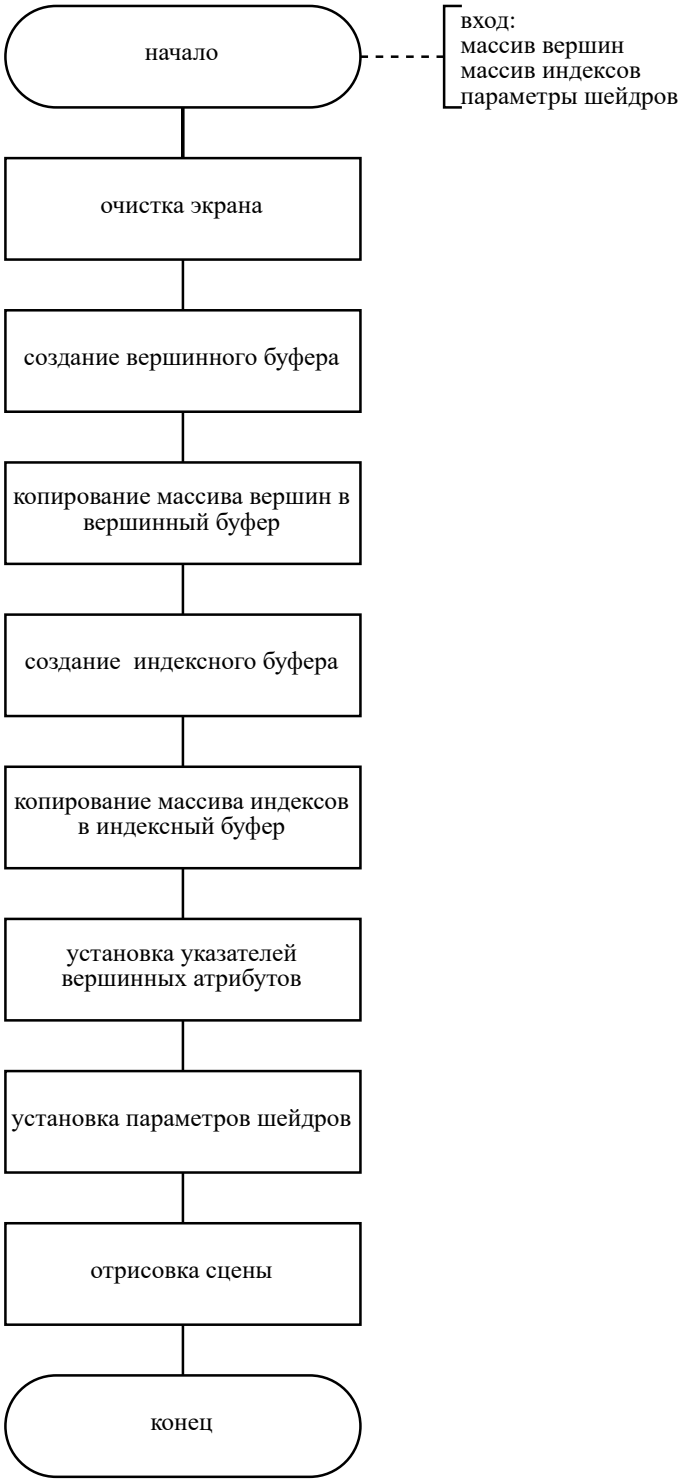

Рисунок 2.3 – Схема алгоритма синтеза изображения

2.3 Описание структуры программы

На рисунке 2.4 представлена диаграмма классов разрабатываемого программного обеспечения.

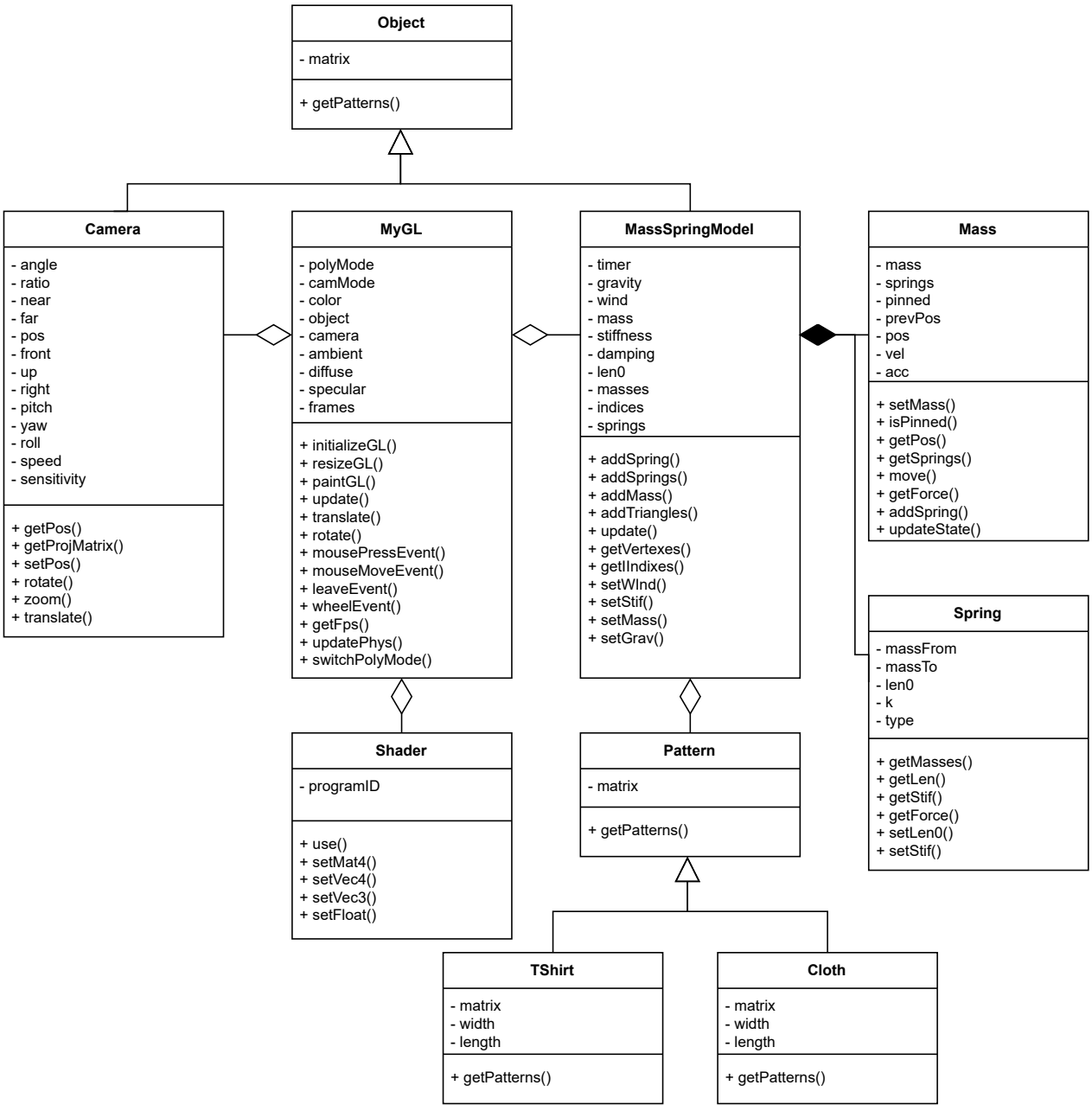

Рисунок 2.4 – Диаграмма классов

В разрабатываемом программном обеспечении реализуются следующие классы:

– MyGL – класс-виджет для отрисовки сцены;

– Shader — класс подключения шейдров;

– Object — базовый класс объектов сцены;

– Camera — класс камеры;
– MassSpringModel — класс массо-пружинной модели;
– Mass — класс точечной массы;
– Spring — класс пружины;
– Pattern — базовый класс тканных изделий;
– Cloth — класс прямоугольной ткани;
– TShirt — класс футболки.

Вывод

В данном разделе были описаны требования к программному обеспечению, представлены схемы алгоритмов и структура программы.

3 Технологическая часть

3.1 Средства реализации

Для реализации программного обеспечения был выбран язык программирования высокого уровня Python [22], что обусловлено:

– поддержкой объектно-ориентированного программирования;

– наличие необходимых библиотек для реализации поставленной задачи;

– доступностью и распространенностью учебной литературы и документации.

В качестве среды разработки выбран текстовый редактор Vim [23] с установленными плагинами автодополнения и поиска ошибок в процессе написания. Данный редактор предоставляет возможности быстрого перемещения по тексту программы и доступа к командной строке, что повышает скорость разработки.

Для разработки интерфейса программного обеспечения использован Qt Desingner [24], предоставляющий возможность автоматического преобразования построенного интерфейса в код на необходимом языке.

Также в процессе разработки использовались следующие библиотеки:

– PyQt5 для реализации работы с интерфейсом;

– numpy и ctypes – для преобразования типов Python к типам, с которыми работает OpenGL;

– glm – математическая библиотека для работы с векторами и матрицами.

3.2 Реализация алгоритмов

На листинге 3.1 представлена реализация алгоритма обновления положений всех точных масс. На листинге 3.2 представлена реализация алгоритма вычисления нового положения точечной массы. На листинге 3.3 представлена реализация вычисления результирующей силы, действующей на точечную массу. На листинге 3.4 представлена реализация вычисления силы упругости от пружины.

Листинг 3.1 – Алгоритм обновления положений всех точечных масс

```
    def update(self, dt):
        for mass in self.masses:
            mass.updateState(dt, self.gravity, self.damping, self.wind)
```

Листинг 3.2 – Алгоритм вычисления нового положения точечной массы

```
    def updateState(self, dt, gravity, damping, wind):
        if self.isPinned():
            return

        self.curTime += dt
        curPos = self.pos

        self.acc = self.getForce(gravity, damping, wind) / self.mass
        self.pos = 2 * self.pos - self.prevPos + self.acc * dt * dt / 2

        self.vel = (self.pos - curPos) / dt
        self.prevPos = curPos
```

Листинг 3.3 – Алгоритм вычисления результирующей силы, действующей на точечную массу

```
    def getForce(self, gravCoef, dampCoef, windCoef):
        if self.pinned:
            return glm.vec3()

        gravity = self.mass * glm.vec3(0, -gravCoef, 0)

        internal = glm.vec3()
        for spring in self.springs:
            internal += spring.getForce()

        damping = -dampCoef * self.vel

        x, y, z, t = self.pos.x, self.pos.y, self.pos.z, self.curTime
        wind = glm.vec3(
                    0.1 * sin(10 * t),
                    0,
                    0.1 * abs(sin(z + 10 * t))) if windCoef else glm.vec3()

        return gravity + internal + damping + wind
```

Листинг 3.4 – Алгоритм вычисления силы упругости пружины

```
1     def getForce(self):
2         curLen = self.getVector()
3         vecLen0 = curLen / glm.length(curLen) * self.len0
4 
5         return self.k * (curLen - vecLen0)
```

Вывод

В данном разделе описаны средства реализации программного обеспечения, а также представлены листинги реализованных алгоритмов.

4 Исследовательская часть

4.1 Примеры работы программы

На рисунке 4.1 представлен результат работы программы на базовых настройках. На рисунке 4.2 и 4.3 представлено полученное изображение футболки с измененным цветом спереди и сзади соответственно. На рисунке 4.4 представлено изображение ткани, используемое для проведение эксперимента, описанного в следующем подразделе. Для изменения параметров модели и режима работы программы используются панели управления, представленные на рисунках 4.5-4.9.

Рисунок 4.1 – Пример работы программы на настройках по умолчанию

Рисунок 4.2 – Полученное изображение футболки: вид спереди

Рисунок 4.3 – Полученное изображение футболки: вид сзади

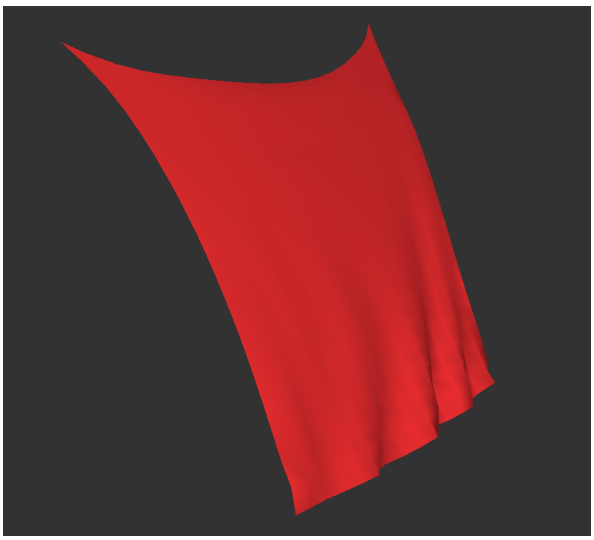

Рисунок 4.4 – Изображение используемое при проведении эксперимента

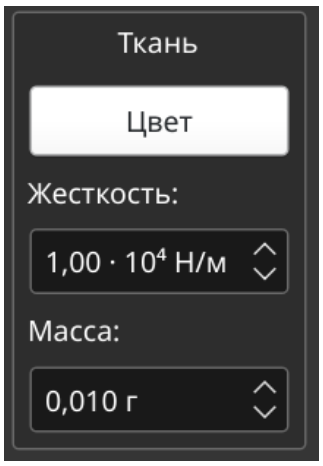

Рисунок 4.5 – Панель управления параметрами ткани

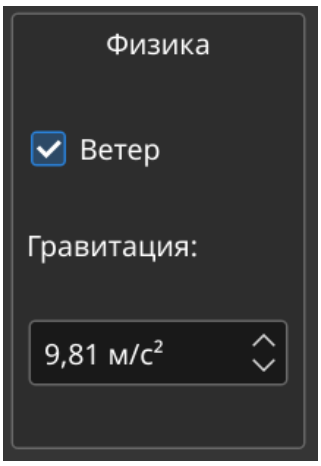

Рисунок 4.6 – Панель управления физикой

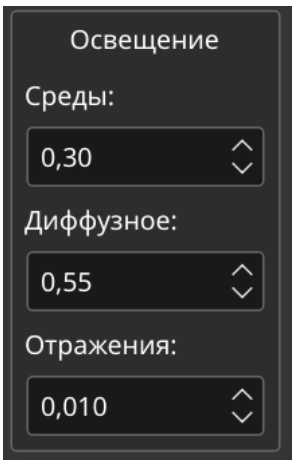

Рисунок 4.7 – Панель управления освещением

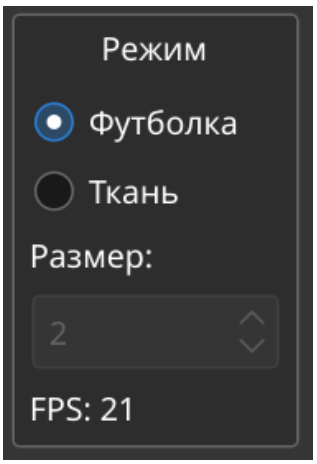

Рисунок 4.8 – Панель управления режимом работы

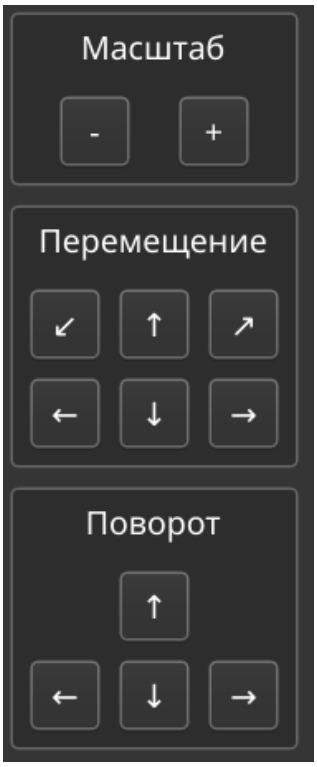

Рисунок 4.9 – Панель управления камерой

4.2 Постановка эксперимента

4.2.1 Описание эксперимента

Цель эксперимента — оценка производительности разработанного алгоритма визуализации ткани при разных количествах точечных масс.

Для получения изображения разработанный алгоритм на каждом шаге вычисляет новое положение каждой частицы или точечной массы, поэтому

с увеличением числа частиц проводится больше вычислений. С целью определения оптимального числа точечных масс был проведен эксперимент по поиску зависимости количества кадров в секунду (Frames Per Second, FPS) от количества частиц.

4.2.2 Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование:

– Операционная система: Manjaro [25] Linux x86_64.
– Память: 8 GiB.
– Процессор: Intel® Core™ i5-8265U [26].

Тестирование проводилось на ноутбуке, включенном в сеть электропитания. Во время тестирования ноутбук был нагружен только встроенными приложениями окружения, окружением, а также непосредственно системой тестирования.

4.2.3 Результаты эксперимента

Результаты эксперимента приведены в таблице 4.1. На рисунке 4.10 представлен график зависимости FPS от количества точечных масс (т. м.).

Таблица 4.1 – Результаты эксперимента

Размер, число т. м.	**Производительность, кадр/сек**
100	60
300	50
500	31
700	22
900	16
1100	13
1300	10
1600	8
3600	5
6400	3

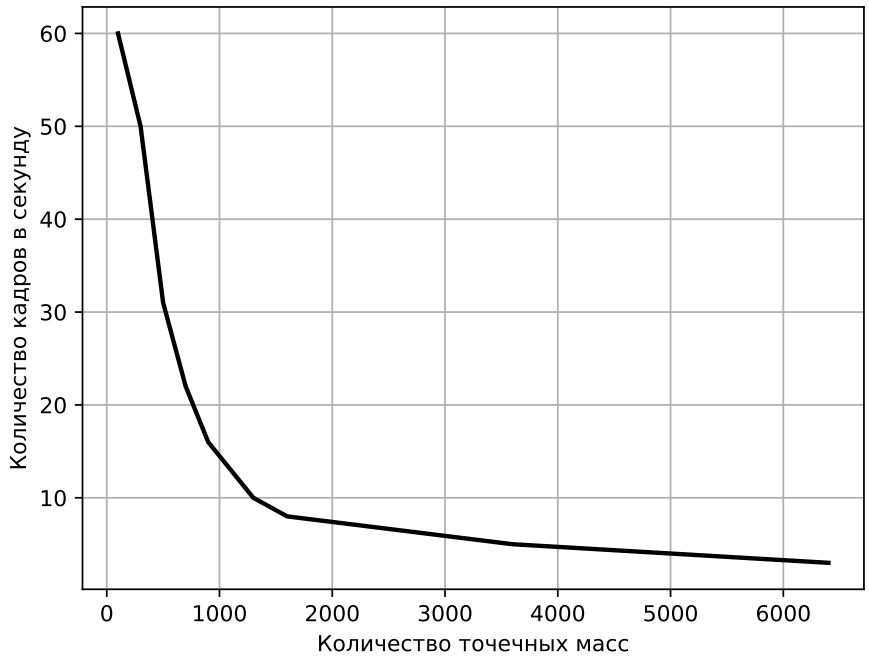

Рисунок 4.10 – График зависимости FPS от числа точечных масс

Вывод

По результатам эксперимента можно сделать вывод о том, что количество кадров в секунду экспоненциально уменьшается при увеличении количества точечных масс. Для человеческого глаза оптимальной является частота 24 кадра в секунду, что достигается при количестве точечных масс не большем 700. При этом в результате визуальной оценки можно сделать вывод о том, что при частоте 10-20 кадров в секунду отдельные изображения все еще неразличимы, наблюдается только замедление анимации, что говорит о возможности получения реалистичной анимации ткани при количестве точечных масс до 1300, которого достаточно для построения изображения футболки, состоящей из 1008 точечных масс.

ЗАКЛЮЧЕНИЕ

В ходе курсовой работы было разработано программное обеспечение, предоставляющее возможность реалистичной визуализации плечевой одежды на примере футболки. Разработанное программное обеспечение также позволяет изменять параметры окружающей среды и модели ткани футболки в интерактивном режиме.

В процессе выполнения курсовой работы:

– формально описана модель ткани, как части одежды;

– описаны методы визуализации ткани и возможности соединения ее частей для получения одежды;

– разработан и реализован алгоритм визуализации футболки;

– произведена оценка производительности реализованного алгоритма.

Таким образом, все поставленные задачи были выполнены, а цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Simnett T. J. R. Real-time simulation and visualisation of cloth using edge-based adaptive meshes // University of East Anglia. — 2012.
2. Zurdo J. S., Brito J. P., Otaduy M. Animating Wrinkles by Example on Non-Skinned Cloth // IEEE Transactions on Visualization and Computer Graphics. — 2013. — Vol. 19. — P. 149–158.
3. Stuyck T. Cloth Simulation for Computer Graphics // Synthesis Lectures on Visual Computing. — 2018. — Vol. 10. — P. 1–121.
4. Keckeisen M. Physical cloth simulation and applications for the visualization, virtual try-on and interactive design of garments // University of Tübingen. — 2005.
5. Mohd Shapri N. S., Bade A., Daman D. Dynamic Cloth Simulation with Fast Self-Collision Detection // Dubai, International Conference on Advanced Computer Theory and Engineering. — 2011. — Vol. 1. — P. 127–134.
6. Kieran E., Harrison G., Openshaw L. Cloth Simulation // Bournemouth University. — 2006.
7. Ng H. N., Grimsdale R. L. Computer graphics techniques for modeling cloth // IEEE Computer Graphics and Applications. — 1996. — Vol. 16. — P. 28–41.
8. Weft — Custom Woven Textiles. — URL: `https://www.weft.design/tutorials`.
9. Yalçın M. A., Yildiz C. Techniques for Animating Cloth. — 2009.
10. Jerry W. The synthesis of cloth objects // Computer Graphics. — 1986.
11. Hinds B., McCartney J. Interactive garment design // The Visual Computer. — 2005. — Т. 6. — С. 53–61.
12. Ng H. N., Grimsdale R. L. GEOFF - A Geometrical Editor for Fold Formation // ICSC. — 1995.

13. Elastically deformable models / D. Terzopoulos, J. C. Platt, A. Barr [et al.] // SIGGRAPH '87. — 1987.

14. Li Y. Q., Dai X. Q. Biomechanical engineering of textiles and clothing. — 2006.

15. Breen D. E., House D. H., Wozny M. J. A Particle-Based Model for Simulating the Draping Behavior of Woven Cloth // Textile Research Journal. — 1993. — Vol. 64. — P. 663 – 685.

16. Provot X. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. — 1995.

17. Ненаженко Д. В., Радченко Г. И. Удаленная визуализация больших объемов данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. — 2015. — Т. 1. — С. 21–32.

18. Жильцова С. С. Анализ средств создания человеко-машинного интерфейса, ориентированного на пользователя // Вестник ПензГУ. — 2019. — Т. 2. — С. 52–58.

19. CLO — 3D Fashion Design Software. — URL: https://www.clo3d.com/.

20. Browzwear — Fashion Design Software. — URL: https://browzwear.com/.

21. PhysiX Clothing. — URL: https://developer.nvidia.com/clothing.

22. Welcome to Python. — URL: https://www.python.org.

23. welcome home : vim online. — URL: https://www.vim.org/.

24. User Interface Desing and Development. — URL: https://www.qt.io/design.

25. Manjaro — enjoy the simplicity. — URL: https://manjaro.org/.

26. Процессор Intel® Core™ i5-8265U. — URL: https://ark.intel.com/content/www/ru/ru/ark/products/149088/intel-core-i5-8265u-processor-6m-cache-up-to-3-90-ghz.html.