



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:

*«Макетирование и визуализация загородной
местности»*

Студент ИУ7-56Б
(Группа)

(Подпись, дата)

Вольняга М. Ю.
(И. О. Фамилия)

Руководитель курсовой работы

(Подпись, дата)

Мальцева Д. Ю.
(И. О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ	3
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	4
ВВЕДЕНИЕ	5
1 Аналитический раздел	6
1.1 Способы определения моделей	6
1.2 Методы представления трехмерных поверхностей	7
1.3 Формализация объектов сцены	9
1.4 Алгоритмы удаления скрытых линий и поверхностей	10
1.5 Алгоритмы построения теней	14
2 Конструкторский раздел	16
2.1 Z-буферный алгоритм	16
2.1.1 Формальная запись	16
2.2 Модифицированный Z-буферный алгоритм	19
2.2.1 Формальная запись	19
2.3 Выбор используемых типов и структур данных	20
3 Исследовательская часть	21
3.1 Вывод	21
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23
ПРИЛОЖЕНИЕ А	24

ОПРЕДЕЛЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие термины с соответствующими определениями.

Бикубический — это термин, который указывает на то, что уравнения, описывающие координаты точек поверхности, содержат две пары параметров с показателями степени, не превышающими третьей [1]

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие сокращения и обозначения.

Программное обеспечение — ПО

ВВЕДЕНИЕ

ПО для макетирования и визуализации загородной местности может применяться в ландшафтном дизайне, архитектуре и планировании территории. Визуализация при планировании местности улучшает коммуникацию и снижает риск недопонимания между заказчиком и исполнителем [2].

Цель работы — разработка программного обеспечения для макетирования и визуализации загородной местности.

Для достижения поставленной цели требуется решить следующие задачи:

- формально описать структуру объектов;
- выбрать алгоритмы трехмерной графики для визуализации сцены и объектов;
- формально записать выбранные алгоритмы;
- выбрать структуры данных для объектов сцены;
- выбрать язык программирования;
- реализовать выбранные алгоритмы;
- разработать программное обеспечение для макетирования и визуализации загородной местности.

1 Аналитический раздел

В данном разделе описаны способы описания моделей, методы представления трехмерных поверхностей, формализация объектов сцены, алгоритмы удаление скрытых линий и поверхностей и алгоритмы построения теней.

1.1 Способы определения моделей

В системах трехмерного моделирования используются каркасные, поверхностные и объемные твердотельные модели. Правильный выбор метода определения моделей на сцене определяет размер и визуализацию модели на сцене.

Каркасная модель — это простейшая модель трехмерного объекта, представляющая собой совокупность вершин, соединенных между собой ребрами [1].

Главным недостатком данной модели является отсутствие информации о поверхности объекта, что делает невозможным разграничение внутренних и внешних граней, например, как на рисунке 1.1. Каркасная модель занимает меньше памяти и эффективна для простых задач [3].

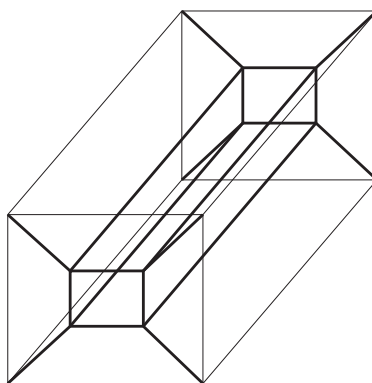


Рисунок 1.1 – Пример каркасной модели

Поверхностная модель, в отличие от каркасной, включает в себя не только вершины и ребра, но также поверхности, создавая визуальный контур объекта [1]. Каждый объект в данной модели обладает внутренней и внешней частью, как на рисунке 1.2.

В основу поверхностной модели положены два основных математических положения:

- любую поверхность можно аппроксимировать многогранником, где каждая грань представляет собой простейший плоский многоугольник [3];
- в модели допускаются не только плоские многоугольники, но и поверхности второго порядка, а также аналитически неопределяемые поверхности [3].

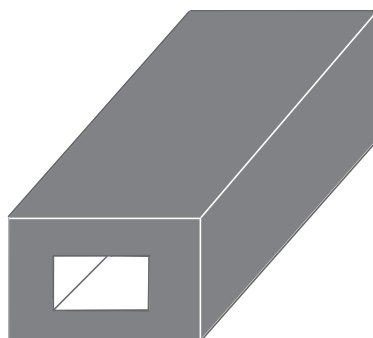


Рисунок 1.2 – Пример поверхностной модели

Недостаток поверхностной модели — отсутствует информация, о том, с какой стороны поверхности находится материал, а с какой пустота.

Твердотельная модель отличается от поверхностной тем, что включает информацию о расположении материала с обеих сторон поверхности [3].

Выбор определения моделей

В данной задаче наиболее оптимальными являются поверхностные модели объектов, так как каркасные модели недостаточно полно представляют форму объекта, а твердотельные модели избыточны.

1.2 Методы представления трехмерных поверхностей

Метод полигональной сетки представляет объект в виде связанной между собой сетки плоских многоугольников [1], как показано на рисунке 1.3.

Метод параметрических бикубических кусков использует математические формулы, описывающие координаты поверхностей. Этот подход обеспечивает высокую точность при описании поверхности и требует меньше элементов для представления сложных форм, в сравнении с полигональными сетками. Однако алгоритмы, работающие с бикубическими кусками сложнее [1].

В рамках данной задачи выбран метод полигональной сетки. Этот выбор обоснован геометрической простотой объектов сцены и отсутствием необхо-

димости использования сложных математических формул. Использование полигональной сетки позволит применять более простые алгоритмы для обработки объектов [1].

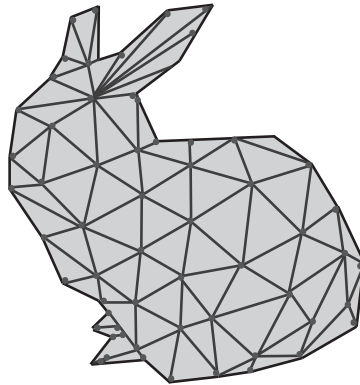


Рисунок 1.3 – Пример полигональной сетки, изображающей кролика

Способы описания полигональных сеток

Наиболее распространенные методы представления полигональных сеток, рассматриваются в [4].

- 1) Список граней — распространенный метод представления трехмерных моделей, описывает объект как множество граней и вершин, где каждая грань имеет минимум 3 вершины. *Преимущества:* простота поиска вершин грани, динамическое обновление формы без изменения связности граней. *Недостатки:* трудности при операциях разрыва и объединения граней, а также проблемы с поиском граней.
- 2) Вершинное представление — это метод представления модели через множества вершин, которые связаны между собой. В качестве *преимущества* можно выделить его простоту. К *недостаткам:* отсутствие явной информации о гранях и ребрах, а также редкое использование в современных системах визуализации.
- 3) «Крылатое» представление — метод, представляющий модель как упорядоченное множество граней вокруг ребра. *Преимущество:* решение проблемы перехода от ребра к ребру через упорядоченное множество граней. *Недостаток:* высокие требования к памяти из-за увеличивающейся сложности структуры

Выбор способа описания полигональной сетки

Для работы выбран метод представления моделей через список граней, обеспечивающий ясное описание и удобный доступ к элементам сетки. Этот подход упрощает модификацию моделей, включая добавление, удаление и изменение граней и вершин.

1.3 Формализация объектов сцены

Сцена — это прямоугольный параллелепипед, состоящий из сетки квадратов, на которой размещаются объекты. Пользователь определяет границы сцены, указывая количество квадратов по ширине и длине. Размер каждого квадрата является постоянным и устанавливается программно. Цвет сцены — светло-зеленый.

Объекты сцены

- 1) Программа включает два типа домов: «стандартный» (одноэтажное здание, 3×2 квадрата) и «премиальный» (двухэтажное сооружение с гаражом, 3×3 квадрата, включая гаражные стены, крышу и ворота). Каждый дом представляет собой единый объект, но состоит из нескольких компонентов: *крыша* (состоит из двух треугольников и двух трапеций, коричневого цвета), *стены* (формируются вертикальными плоскостями, бежевого цвета), *окна* (вертикальные плоскости, расположены в середине стены, голубого цвета), *дверь* (вертикальная плоскость, расположена посередине стены, коричневого цвета), *ворота* (вертикальная плоскость, расположена посередине двух стен, коричневого цвета).
- 2) Дороги занимают один квадрат сцены и имеют темно-серый цвет.
- 3) Тротуары аналогичны дорогам, светло-серого цвета.
- 4) Машины цельный объект из четырех колес, шести стекол и кузова. В программе имеется выбор модели машины, различного цвета. Компоненты машины: *кузов* (формируется набором плоскостей, имитирующих реальные автомобили, цвет красный или серый), *колеса* (представляют собой цилиндрические объекты, черного цвета), *окна* (плоскости, встроенные в кузов автомобиля, голубого цвета).
- 5) Деревья состоят из листвы и ствола, представляют цельный объект

сцены и занимают один квадрат сцены. Листва имеет зеленый цвет, а ствол коричневый;

6) Кусты схожи с деревьями, но без ствола. Занимают один квадрат сцены.

Объекты сцены определены заранее и не могут быть изменены. ПО позволяет перемещать или удалять объекты на сцене. Есть ограничение на размещение моделей на сцене, а именно, машина может размещаться только на дороге. Дома, дороги, тротуары, деревья и кусты могут размещаться только на свободных квадратах сцены. Размер сцены ограничен.

Источник света

Для моделирования освещения в компьютерной графике обычно используются три основных типа источников света: точечные, направленные и общий свет (Ambient Light) [5]. В данной работе выбор сделан в пользу точечного источника света.

Точечный источник света излучает свет равномерно во все стороны из определенной точки в трехмерном пространстве, что позволяет эффективно управлять освещением и тенями на сцене, учитывая положение объектов относительно источника света. Положение источника света устанавливается относительно текущей точки наблюдения с помощью последовательных поворотов по осям X и Y [5].

1.4 Алгоритмы удаления скрытых линий и поверхностей

Алгоритмы удаления невидимых частей сцены классифицируются на основании следующих критериев:

- 1) выбор удаляемых частей: линий, ребер, поверхностей, объемов;
- 2) порядок обработки элементов сцены: удаление может быть определено процессом визуализации или выполняться в произвольном порядке;
- 3) зависимость от системы координат: существуют алгоритмы, работающие в пространстве объектов, где каждая из N граней объекта сравнивается с остальными $N - 1$ гранями (объем вычислений растет как N^2), а также алгоритмы, работающие в пространстве изображения, когда для

каждого пикселя изображения определяется, какая из N граней объекта видна (при разрешении экрана $M \times M$ объем вычислений растет как $M^2 \times N$) [6].

Алгоритм Робертса применяется к выпуклым объектам в трехмерном пространстве, представленным как многогранники, образованные пересечением плоскостей [6].

- 1) Удаление нелицевых плоскостей перед определением видимости.
- 2) Сравнение каждого ребра с каждым объектом (вызывает значительный объем вычислений, растущий квадратично по числу объектов в сцене).
- 3) Вычисляются новые ребра от пересечения объектов.

Z-буферный алгоритм удаления поверхностей

Алгоритм представляет собой расширение обычного буфера кадра, который хранит цвета каждого пикселя. В Z-буферном алгоритме, дополнительно к каждому пикселю присваивается значение глубины (Z-координата). При добавлении нового пикселя в буфер, сравнивается его Z-координата с той, что уже находится в буфере. Если новый пиксель ближе к наблюдателю (т.е. его Z-координата больше), его атрибуты и Z-координата записываются в буфер [6]. Пример работы алгоритма на рисунке 1.4.

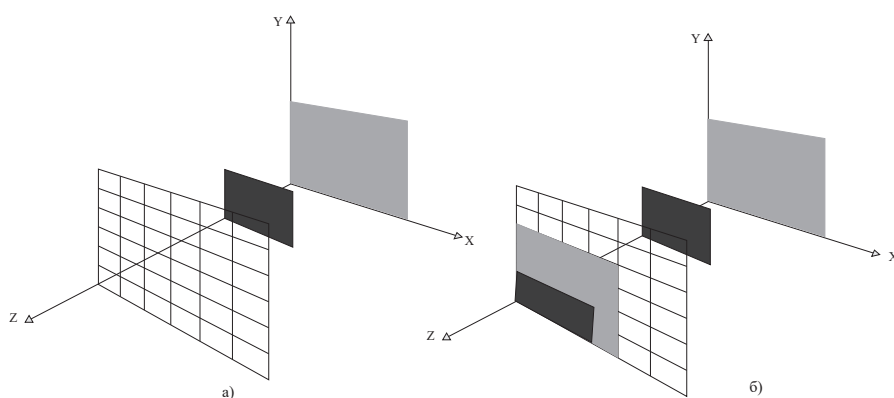


Рисунок 1.4 – Работа алгоритма с Z-буфером

Преимущества Z-буферного алгоритма: поддерживает сцены произвольной сложности, отсутствует необходимость в сортировке, что требуется в других алгоритмах, вычислительная сложность линейно зависит от числа анализируемых поверхностей [6].

Недостаток Z-буферного алгоритма: требует значительного объема памяти для буферов [6]

Алгоритм Варнока

Алгоритм Варнока использует принцип когерентности, по которому большие области изображения однородны. Алгоритм работает в пространстве изображения. Основная идея — разбивать окно на подокна до тех пор, пока их содержимое не станет достаточно простым для визуализации, или размер подокна не достигнет заданного предела разрешения [6; 7]

Конкретная реализация зависит от метода разбиения окна и критерия, определяющего простоту содержимого окна. В оригинальной версии алгоритма окно делится на четыре равных по размеру подокна. Вариант Вейлера и Азертона предполагает разбиение окна по ребрам изображаемых многоугольников [7].

Эффективность данного алгоритма зависит от сложности сцены. Использование полигональной сетки может замедлить его выполнение.

Алгоритм прямой и обратной трассировки лучей используются для отслеживания траектории лучей от источника света до камеры, учитывая взаимодействие с объектами на пути.

Прямая трассировка лучей строит траектории лучей от всех источников света до всех точек сцены, включая те, которые не попадают в камеру. Из-за этого, метод считается неэффективным [8].

Обратная трассировка лучей — алгоритм при котором отслеживание лучей осуществляется не от источников света, а в обратном направлении от точки наблюдения. Учтены только лучи, влияющие на формирование изображения (см. рисунок 1.5). Недостатки алгоритма включают неучет вторичного освещения, высокую вычислительную стоимость и дискретность первичных лучей. Тем не менее, алгоритм поддерживает расчет теней, многократных отражений и преломлений [8].

Выбор алгоритма удаления скрытых линий и поверхностей

Для лучшей визуальной эффективности алгоритмов была составлена таблица 1.1, в строке «Вычислительная сложность» N и C означают, количество граней и количество пикселей, соответственно.

На основании рассмотренных алгоритмов, Z-буфер является оптимальным выбором для удаления невидимых линий и поверхностей. Он обеспечит

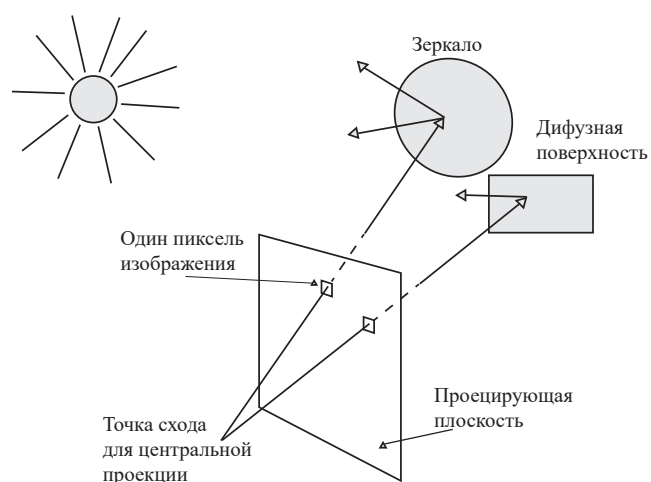


Рисунок 1.5 – Схема обратной трассировки лучей

Таблица 1.1 – Сравнение алгоритмов удаления невидимых линий и поверхностей.

	z-буфер	Обратная трассировка лучей	Варнок	Робертс
Вычисл. сложность	$O(CN)$	$O(CN)$	$O(CN)$	$O(N^2)$
Рабочее пространство	Изображение	Изображение	Изображение	Объектное
Трудность реализации	Низкая	Средняя	Средняя	Высокая
Производит. при сложной сцене	Высокая	Низкая	Средняя	Низкая
Распростран. в современном ПО	Высокая	Высокая	Низкая	Низкая
Использов. рекурсивных вызовов	Нет	Да	Да	Нет

быструю обработку сцен любой сложности и адаптируется под изменения в освещении и закраске. При небольшом размере изображения Z-буфер эффективно обрабатывает большое количество объектов без проблем с памятью.

1.5 Алгоритмы построения теней

Модификация Z-буфера

В алгоритмах отображения сцены можно адаптировать Z-буфер для отслеживания теней. Изначально, этот буфер хранит данные о глубине каждого пикселя, но для точного отображения теней нужен дополнительный этап.

Первым шагом камера устанавливается на место источника света, позволяя занести данные о глубине каждой точки в теневой буфер. Это гарантирует, что все отмеченные точки расцениваются как освещенные [11].

Далее формируется изображение с точки зрения наблюдателя. На этом этапе каждый пиксель анализируется с учетом его расстояния до источника света и сопоставляется с данными из теневой карты. Если расстояние до пикселя больше значения из теневой карты, пиксель считается находящимся в тени и отображается с акцентом на рассеянном свете, это можно увидеть на примере рисунка 1.6, точка *B* имеет большее расстояние до источника света, чем восстановленный из карты теней, следовательно точка *B* будет в тени, а точка *A* находится на том же расстоянии от источника света, что и восстановлено из карты теней и не будет затемнена. В итоге, модификация Z-буфера позволяет эффективно учесть тени при рендеринге сцены, одновременно выполняя удаление невидимых поверхностей [11].

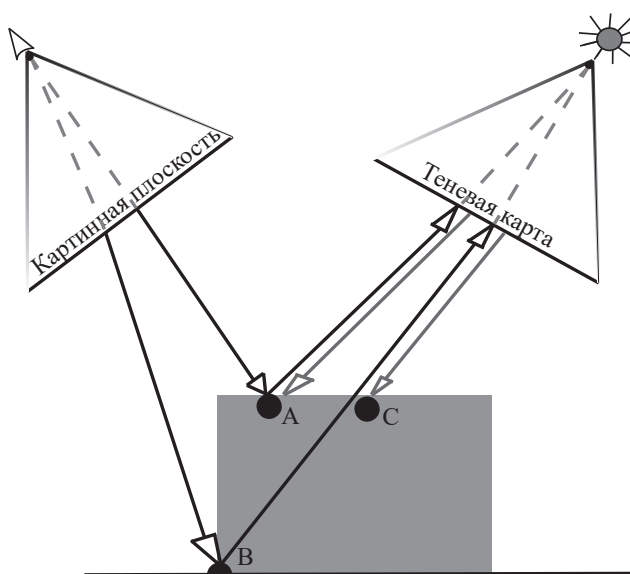


Рисунок 1.6 – Пример определения тени, алгоритмом модификации Z-буфера

Модификация Вейлера-Азертонна

На первом шаге с помощью алгоритма удаления невидимых поверхно-

стей выбираются освещенные грани, т. е. грани, которые видны из положения источника света. Для увеличения эффективности, в памяти хранятся именно эти грани, что позволяет избежать удвоения обрабатываемых граней для выпуклого многоугольника. Освещенные многоугольники помечаются и преобразуются к исходной ориентации, где они приписываются к своим прототипам в качестве многоугольников детализации поверхности. Чтобы избежать появления ложных теней, сцену рассматривают в пределах видимого или отсекающего объема, определенного положением источника света [11].

На втором шаге объединенные данные о многоугольниках обрабатываются из положения наблюдателя. Если какая-то часть не освещена, применяется соответствующее правило затенения. Если источников света несколько, то используется несколько наборов освещенных граней [11].

Выбор алгоритма построения теней

Выбор модифицированного Z-буфера для построения теней в данном контексте является оптимальным. Это обусловлено его эффективностью и совместимостью с уже используемым Z-буфером для удаления невидимых поверхностей. Использование одной и той же технологии упрощает интеграцию.

Вывод

В представленном разделе были выбраны: поверхностные модели для описания трехмерных объектов, полигональные сети с использованием списка граней для представления трехмерных поверхностей, а также удаление невидимых ребер алгоритмом Z-буфера и построение теней с использованием модификации Z-буфера. Были формализованы объекты сцены.

2 Конструкторский раздел

В данном разделе представлены требования к программному обеспечению, а также схемы алгоритмов, выбранных для решения поставленной задачи.

Программа должна предоставлять следующий функционал:

- создание сцены с определенным размером;
- размещение, удаление и перемещение отдельных объектов сцены;
- добавление источника света;
- поворот, перемещение и масштабирование сцены с объектами.

Программа должна соответствовать следующим критериям:

- Для обеспечения плавной и корректной интерактивной работы необходимо, чтобы время, в течение которого программа реагирует на действия пользователя, составляло менее одной секунды.
- обеспечение корректного реагирования на все действия пользователя.

2.1 Z-буферный алгоритм

2.1.1 Формальная запись

- 1) инициализировать кадровый буфер и $z_{\text{буфер}}$;
- 2) кадровый буфер заполнить фоном и $z_{\text{буфер}}$ заполнить минимальным значением z ;
- 3) для каждого многоугольника сцены;

3.1 для каждого пикселя (x, y) в многоугольнике вычислить его глубину $z(x, y)$;

- i. сравнить глубину $z(x, y)$ со значением $z_{\text{буфер}}$, хранящимся в $z_{\text{буфер}}$ в этой же позиции.

Если $z(x, y) > z_{\text{буфер}}(x, y) \Rightarrow z_{\text{буфер}}(x, y) = z(x, y)$ и записать атрибут этого многоугольника (интенсивность, цвет и т. п.);

- ii. в противном случае никаких действий не производить;
- 4) вывести изображение.

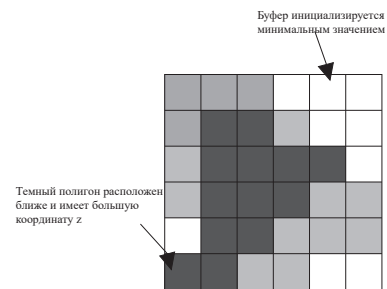


Рисунок 2.1 – Визуализация состояния Z-буфера

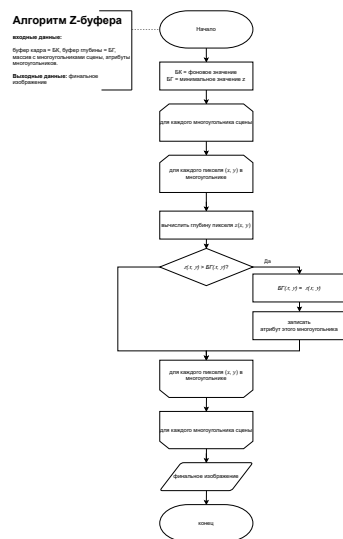


Рисунок 2.2 – Схема алгоритма Z-буфера

2.3 Выбор используемых типов и структур данных

- 1) точка задается координатами (x, y, z) ;
- 2) полигон задается тремя точками и храниться в массиве;
- 3) объект сцены, например: дом или дорога, задается полигонами в виде массива;
- 4) источник света задается углом, по осям X и Y относительно точки наблюдателя.

Вывод

В этом разделе были разобраны следующие элементы:

- 1) требования к разрабатываемому программному обеспечению;
- 2) разработана схема разрабатываемого алгоритма;
- 3) описаны структуры данных, которые будут использоваться при реализации программного обеспечения.

3 Исследовательская часть

3.1 Вывод

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Лусяк В. В.* Основы геометрического моделирования. — 2022.
2. *Lovett A. Appleton K. W.-K. B.* Using 3D visualization methods in landscape planning: An evaluation of options and practical issues // *Landscape and Urban Planning*. — 2015. — С. 85—94.
3. *Донченко В. Ю. Ч. Е. Н.* Обзор и анализ методов построения геометрических моделей сложных конструкций. — 2014.
4. *Киселев А. В. В. Г. Н.* Способы представления и размещения трехмерных моделей для прототипирования ювелирных изделий // *Материалы VI Международной научно-практической конференции (школы-семинара) молодых ученых*. — 2020. — С. 840—842.
5. *Gambetta G.* Computer Graphics from Scratch. — 2021.
6. *Головнин А. А.* Базовые алгоритмы компьютерной графики. — 2016.
7. *Провкин В. А.* Построение реалистичных изображений при помощи алгоритма Варнока и сравнение эффективности различных его реализаций. — 2017.
8. *Янова Р. Д.* Метод прямой и обратной трассировки // *Вестник науки и образования*. — 2016. — С. 29—30.
9. *Задорожный А. Г.* Модели освещения и алгоритмы затенения в компьютерной графике. — 2020.
10. *Чернявская А. Э.* Простые модели освещения 3D-объектов. Особенности цифрового моделирования света // *Современные вопросы науки и практики*. — 2021. — С. 45—47.
11. *Куров А. В.* Конспект лекций по дисциплине компьютерная графика. — 2023.

ПРИЛОЖЕНИЕ А