СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Задача построения реалистичных трехмерных сцен, учитывающих оптические свойства поверхностей и источников света, является центральной в компьютерной графике. Для решения этой задачи было предложено множество алгоритмов. Генри Гуро и Том Фонг предложили алгоритмы реалистичной раскраски поверхностей, Эд Кэтмул ввел концепцию z-буфера, а Тернер Уиттед предложил трассировку лучей.

Цель работы — реализовать алгоритм построения реалистичных трехмерных сцен объектов.

Для достижения поставленной цели необходимо решить следующие задачи:

1) Проанализировать предметную область, рассмотреть известные подходы и алгоритмы;
2) Спроектировать ПО для построения реалистичных сцен;
3) Реализовать выбранные алгоритмы для построения трехмерной сцены;
4) Исследовать характеристики разработанного ПО.

1 Аналитическая часть

1.1 Оптические эффекты, учитываемые при построении реалистичного изображения

При построении реалистичных изображений необходимо учитывать такие эффекты оптики, как:

— Отражающие свойства поверхностей: диффузное и зеркальное отражения;

— Преломление части света при взаимодействии луча с прозрачной поверхностью;

— Рассеяние света.

Поскольку в данной работе не используются прозрачные объекты, будем рассматривать только отражение и рассеяние света.

Диффузная составляющая явления отражения света от поверхностей описывается следующим уравнением:

$$I_{\text{д}} = I_{\text{и}} cos\theta \cdot k_{\text{д}}, \tag{1.1}$$

где $I_{\text{и}}$ – интенсивность источника освещения, θ – угол между нормалью к поверхности в точке отражения и вектором, направленным из точки отражения к источнику света, $k_{\text{д}}$ -– коэффициент диффузного отражения.

Зеркальная составляющая явления отражения света описывается уравнением:

$$I_{\text{з}} = I_{\text{и}} cos^n\alpha \cdot k_{\text{з}}, \tag{1.2}$$

где $I_{\text{и}}$ – интенсивность источника освещения, α – угол падения луча, $k_{\text{з}}$ – коэффициент зеркального отражения, степень n косинуса характеризует концентрированность света в направлении отражения.

Рассеяние света в компьютерной графике обычно описывают следующей формулой:

$$I' = I_{\text{р}} k_{\text{р}}, \tag{1.3}$$

где $I_{\text{р}}$ – интенсивность рассеянного света, $k_{\text{р}}$ – коэффициент диффузного отражения рассеянного света.

Также необходимо учесть затухание света с расстоянием. Интенсивность света падает обратно пропорционально квадрату расстояния до источника, однако как показывает практика, большей реалистичности можно добиться при линейном затухании [1]. В этом случае получаем следующий результат:

$$I = I_{\text{р}} k_{\text{р}} + \frac{I_{\text{и}} cos\theta \cdot k_{\text{д}} + I_{\text{и}} cos^n\alpha \cdot k_{\text{з}}}{d + K}, \tag{1.4}$$

где d – расстояние до источника освещения, K – произвольная константа.

1.2 Алгоритмы удаления невидимых линий и поверхностей

Все алгоритмы удаления невидимых линий работают либо в пространстве объектов, либо в пространстве изображений.

1.2.1 Алгоритм Робертса

Этот алгоритм позволяет определить, какие рёбра или части рёбер объектов сцены видимы, а какие заслонены гранями других объектов. Он работает в объектном пространстве и использует проекции объектов на картинную плоскость для анализа видимости [2].

1.2.2 Алгоритм Варнока

Алгоритм Варнока основывается на рекурсивном разбиении окна на подокна всякий раз, когда оно не пусто, до окна размеров в 1 пиксель, после чего определяется ближайший к наблюдателю объект, пиксель закрашивается цветом этого объекта.

1.2.3 Алгоритм, использующий z-буфер

Алгоритм основывается на хранении глубины (z-значения) каждого пикселя экрана и определении ближайшей к экрану точки при разложении многоугольников в растр. Главное преимущество алгоритма – его простота. Кроме того, этот ал-горитм решает задачу об удалении невидимых поверхностей и делает тривиальной визуализацию пересечений сложных поверхностей – сцены могут быть любой сложности. Основной недостаток алгоритма – большой объем требуемой памяти [3].

1.2.4 Алгоритм обратной трассировки лучей

Метод прямой и обратной трассировки лучей заключается в том, что от момента испускания лучей источником света до момента попадания в камеру, траектории лучей отслеживаются, и рассчитываются пересечения лучей с лежащими на траектории объектами. При этом луч может быть поглощен, диффузно и/или зеркально отражен или, в случае прозрачности некоторых объектов, преломлен [4].

1.3 Методы закраски

1.3.1 Простая закраска

Если предположить, что источник света находится на бесконечности, то лучи света, падающие на поверхность, параллельны между собой. Если к этому добавить условие, что наблюдатель находится в бесконечно удаленной точке, то эффектом ослабления света с увеличением расстояния от источника также можно пренебречь. При таких вводных плоская грань во

всех ее точках имеет одинаковую интенсивность освещения, поэтому она закрашивается одним цветом.

При закрашивании этим методом на стыке соседних граней неизбежно будут проявляться ребра, поскольку соседние грани с различными направлениями нормалей имеют разный цвет [5].

1.3.2 Закраска методом Гуро

Один из способов устранения дискретности интенсивностей закрашивания был предложен Гуро. Его метод заключается в том, что используются не нормали к плоским граням, а нормали к аппроксимируемой поверхности, построенные в вершинах многогранника. После этого вычисляются интенсивности в вершинах, а затем во всех внутренних точках многоугольника выполняется билинейная интерполяция интенсивности. К недостаткам метода Гуро следует отнести то, что он хорошо работает только с диффузной моделью отражения. Форма бликов на поверхности и их расположение не могут быть адекватно воспроизведены при интерполяции на многоугольниках. Кроме того, есть проблема построения нормалей к поверхности. В алгоритме Гуро нормаль в вершине многогранника вычисляется путем усреднения нормалей к граням, примыкающим к этой вершине. Такое построение сильно зависит от характера разбиения.

Еще один недостаток закраски изображен на рисунке 1.1. Если нормали к вершинам B, C, D вычислить усреднением нормалей к многоугольникам, то они будут одинаково ориентированы, то есть интенсивность в этих точках будет равной. При линейной интерполяции от B до D значение интенсивности получится постоянным, и поверхность на данном участке будет выглядеть плоской [5].

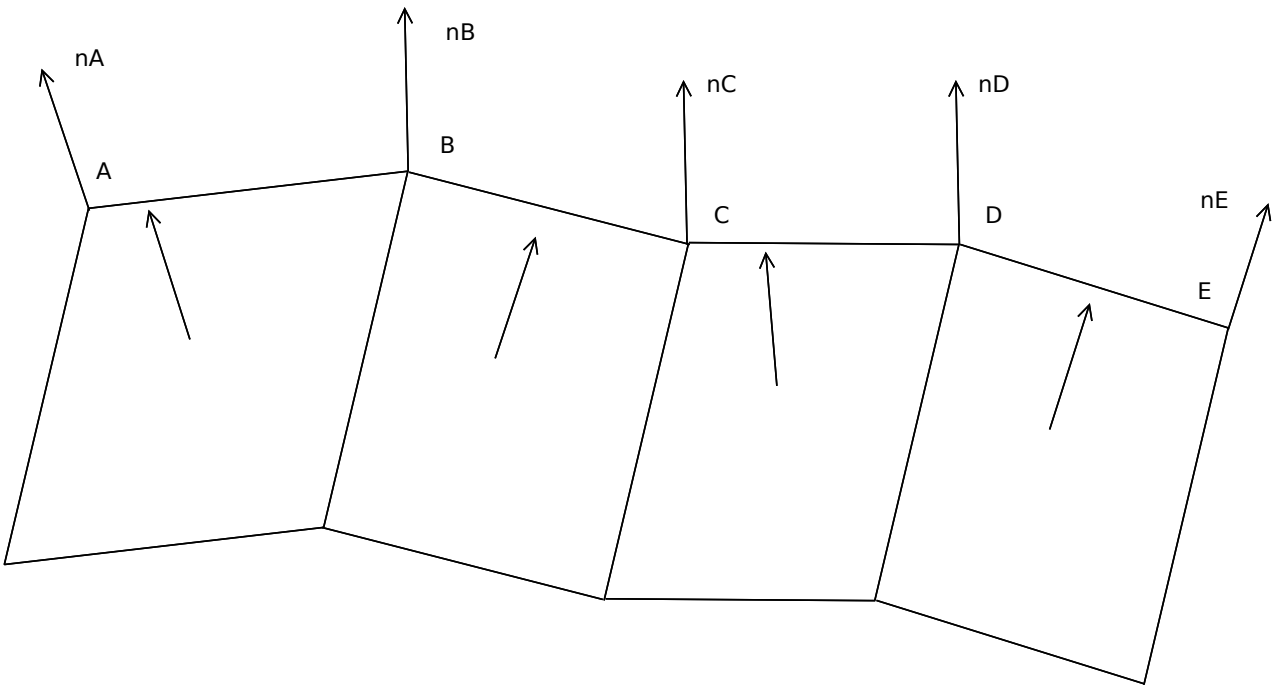

Рисунок 1.1 — Пример некорректной работы закраски Гуро

1.3.3 Закраска методом Фонга

Фонг предложил вместо интерполяции интенсивностей произвести интерполяцию вектора нормали к поверхности на сканирующей строке. Этот метод требует больших вычислительных затрат, поскольку формулы интерполяции применяются уже к трем компонентам вектора нормали, но зато дает лучшую аппроксимацию кривизны поверхности. Поэтому зеркальные свойства поверхности воспроизводятся гораздо лучше.

Нормали к поверхности в вершинах многогранника вычисляются так же, как и в методе Гуро, после чего выполняется билинейная интерполяция в сочетании с построчным сканированием. После построения вектора нормали в очередной точке вычисляется интенсивность [5].

1.4 Построение теней

Тени делятся на два вида: собственные и проекционные. Определение собственных теней не представляет сложностей: точка наблюдателя совмещается с положением источника света и решается задача определения видимых граней. Видимые грани окажутся освещенными, невидимые – в тени.

Если источник света находится на бесконечном расстоянии, то при падении света на объект сцены будет образовываться полоса тени, которую можно задать уравнением. Если же источник света находится не на бесконечном расстоянии, возникает потребность нахождения проекции грани в тени на поверхность другого тела [6].

Стоит отметить, что в случае алгоритма обратной трассировки лучей задача построения теней сводится к тому, чтобы при нахождении точки пересечения луча с объектом сцены провести еще один луч от этой точки к источнику света. Если на пути луча встречается преграда в виде поверхности какого-либо объекта сцены, точка находится в тени.

Вывод

В данном разделе были рассмотрены основные методы построения реалистичных изображений. Для удаления невидимых ребер, учета отражения света и теней лучше всего в данной работе подойдет алгоритм обратной трассировки лучей в силу своей универсальности и реалистичности результата.

2 Конструкторская часть

В данном разделе рассматриваются схемы алгоритмов визуализации сцены объектов методом трассировки лучей.

2.1 Алгоритм трассировки лучей

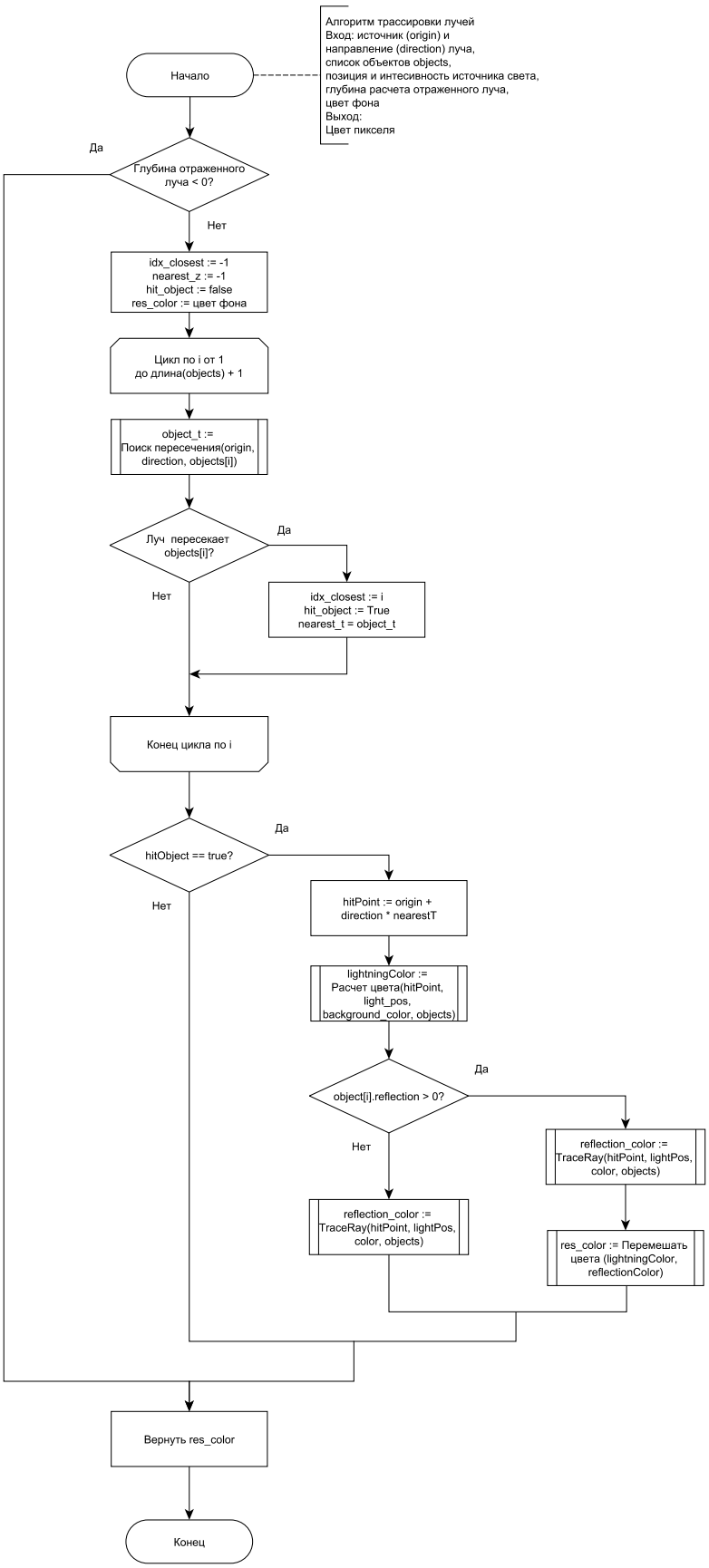

Рисунок 2.1 — Схема алгоритма трассировки лучей

2.2 Алгоритмы нахождения точки пересечения луча с объектом сцены

В данной работе объекты сцены представлены одним из следующих типов:

— сфера;

— куб;

— шахматные фигуры, поверхность которых состоит из треугольных полигонов.

2.2.1 Алгоритм нахождения точки пересечения луча со сферой

Пусть $\vec{c}$ – центр сферы, r – ее радиус. При известных $\vec{p_0}$ и $\vec{u}$ – точке начала и вектора направления луча соответственно, можно выразить точку луча следующим образом:

$$\vec{p(t)} = \vec{p_0} + \vec{u} \cdot t, \qquad t \geq 0. \tag{2.1}$$

Тогда для точки $\vec{p(t)}$ луча, лежащей на сфере, справедливо равенство

$$|\vec{p(t)} - \vec{c}| - r = 0. \tag{2.2}$$

Подставляя 2.1 в 2.2 получим

$$|\vec{p_0} + t \cdot \vec{u} - \vec{c}| - r = 0, \tag{2.3}$$

откуда получаем квадратное уравнение

$$A^2 \cdot t + B \cdot t + C = 0, \tag{2.4}$$

где $A = \vec{u}$, $B = 2\vec{u}(\vec{p_0} - \vec{c})$, $C = (\vec{p_0} - \vec{c})^2 - r^2$.

Таким образом, t выражается как

$$t = \frac{-B \pm \sqrt{B^2 - 4C}}{2}, \tag{2.5}$$

при этом выбирается меньшее t (ближайшая точка) и проверяется на положительность.

На рисунке 2.2 приведена схема алгоритма, реализующая описанные вычисления.

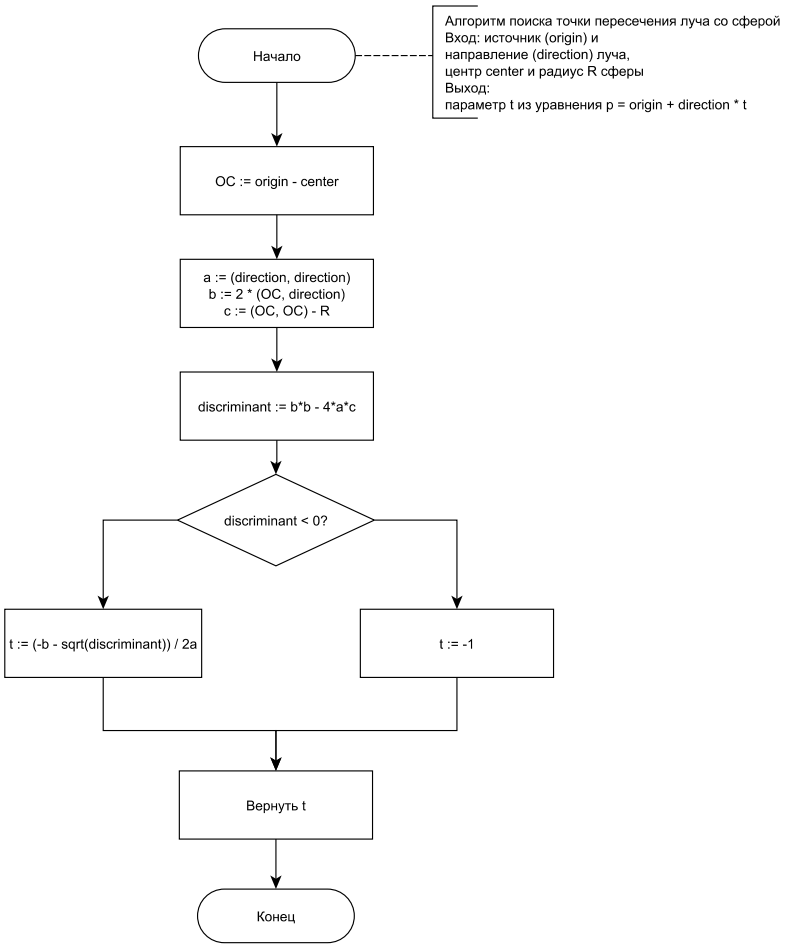

Рисунок 2.2 — Схема алгоритма нахождения точки пересечения луча и сферы

2.2.2 Алгоритм нахождения точки пересечения луча и куба

Алгоритм нахождения точки пересечения луча с кубом использует представление луча в форме, описанной формулой 2.1, а также идею последовательного обрезания луча каждой из граней куба [7]. Алгоритм представлен на схемах 2.3-2.4.

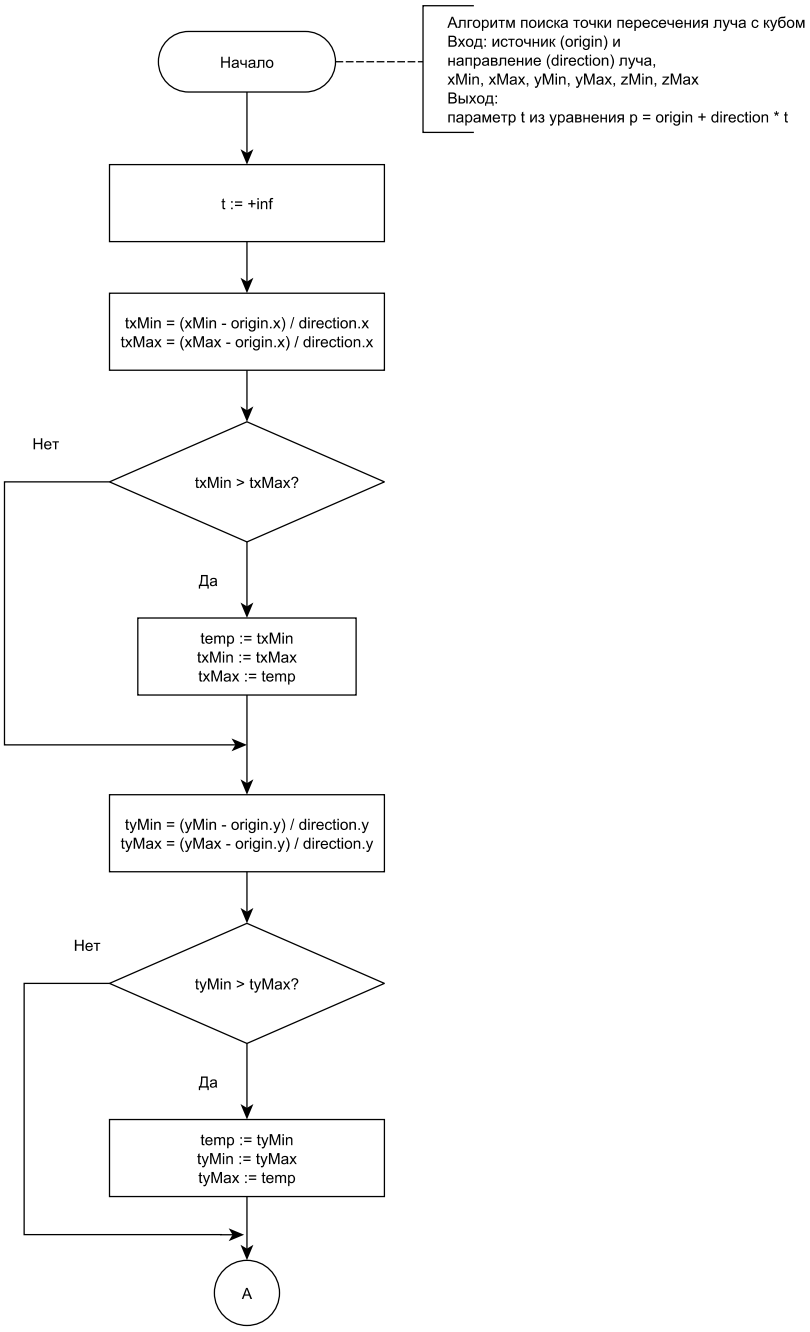

Рисунок 2.3 — Схема алгоритма нахождения точки пересечения луча и сферы

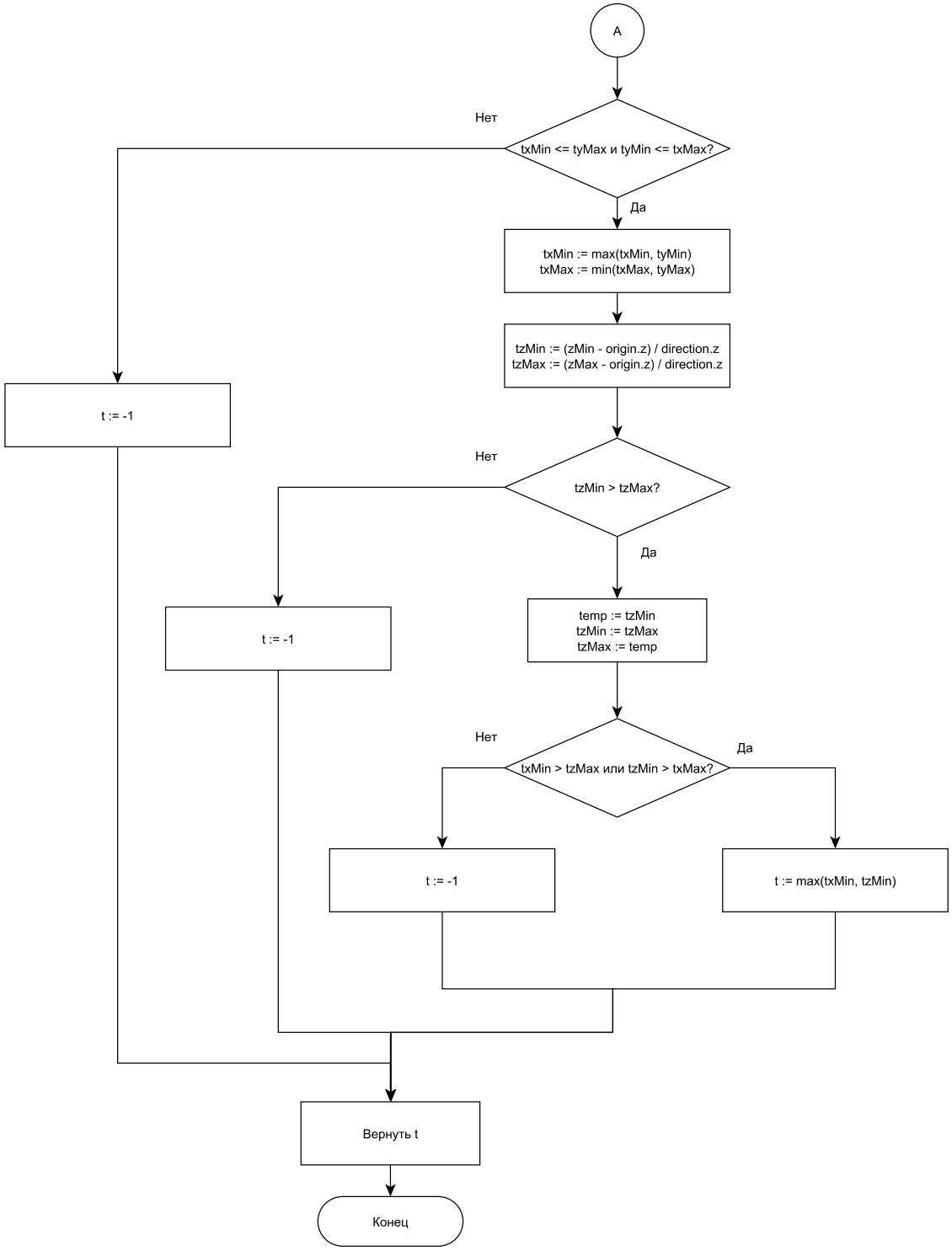

Рисунок 2.4 — Схема алгоритма нахождения точки пересечения луча и сферы

2.2.3 Алгоритм нахождения точки пересечения луча и шахматной фигуры

Поскольку модели шахматных фигур состоят из треугольных полигонов, задача нахождения точки пересечения луча и фигуры включает в себя подзадачу поиска точки пересечения с треугольником. Для решения задачи был выбран алгоритм Моллера — Трумбора, позволяющей

решить задачу без предварительного вычисления уравнения плоскости, содержащей треугольник [8].

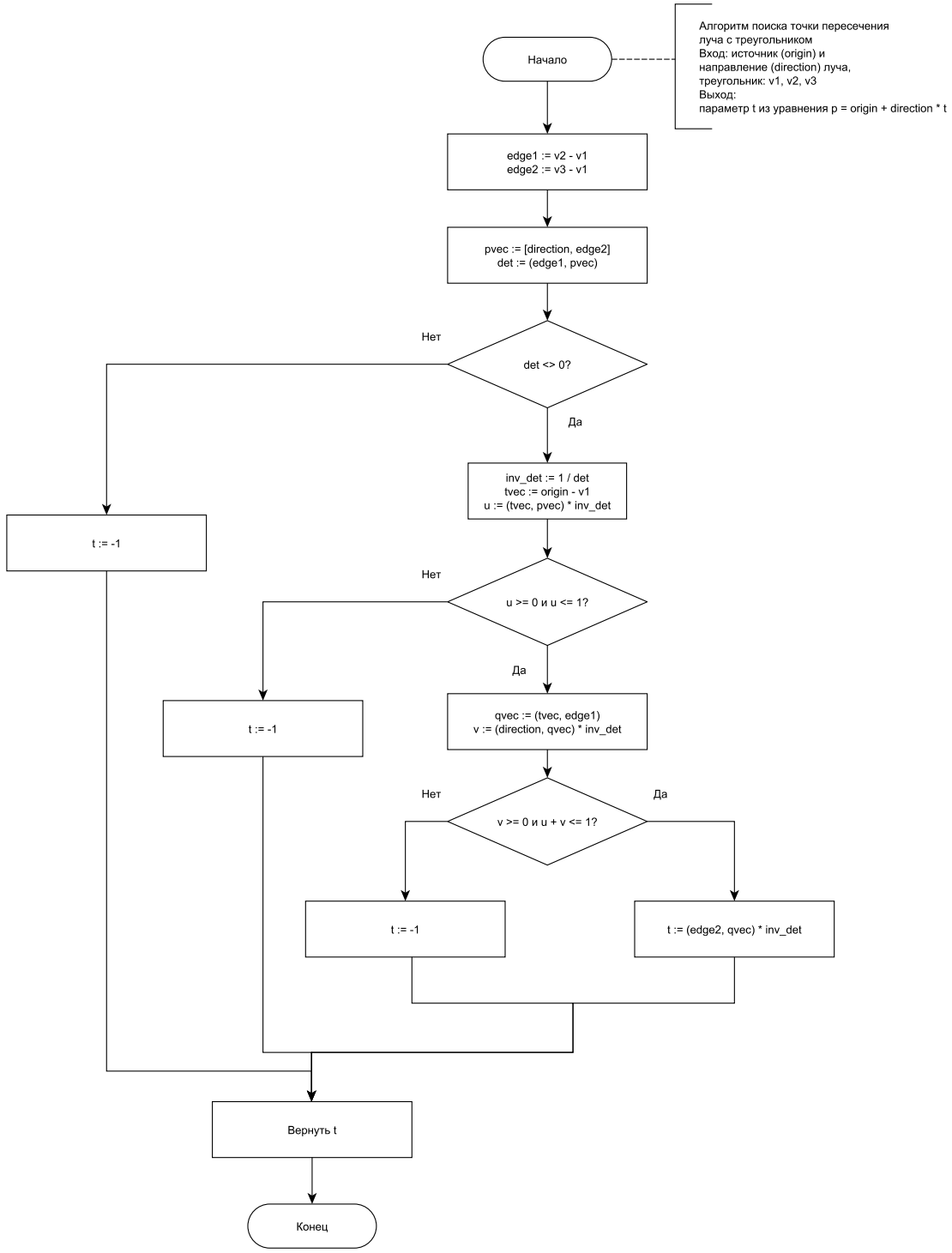

Рисунок 2.5 — Схема алгоритма нахождения точки пересечения луча и треугольника

Алгоритм нахождения точки пересечения луча и шахматной фигуры использует предварительный расчет описанной сферы при создании шахматной фигуры, и далее использует сферу, проверяя пересечение луча с описанной сферой прежде чем итерироваться по всем треугольникам фигуры: если луч не пересекает описанную сферу, то он не может пересечь фигуру. При описанной проверке используется алгоритм нахождения точки пересечения луча со сферой 2.2.

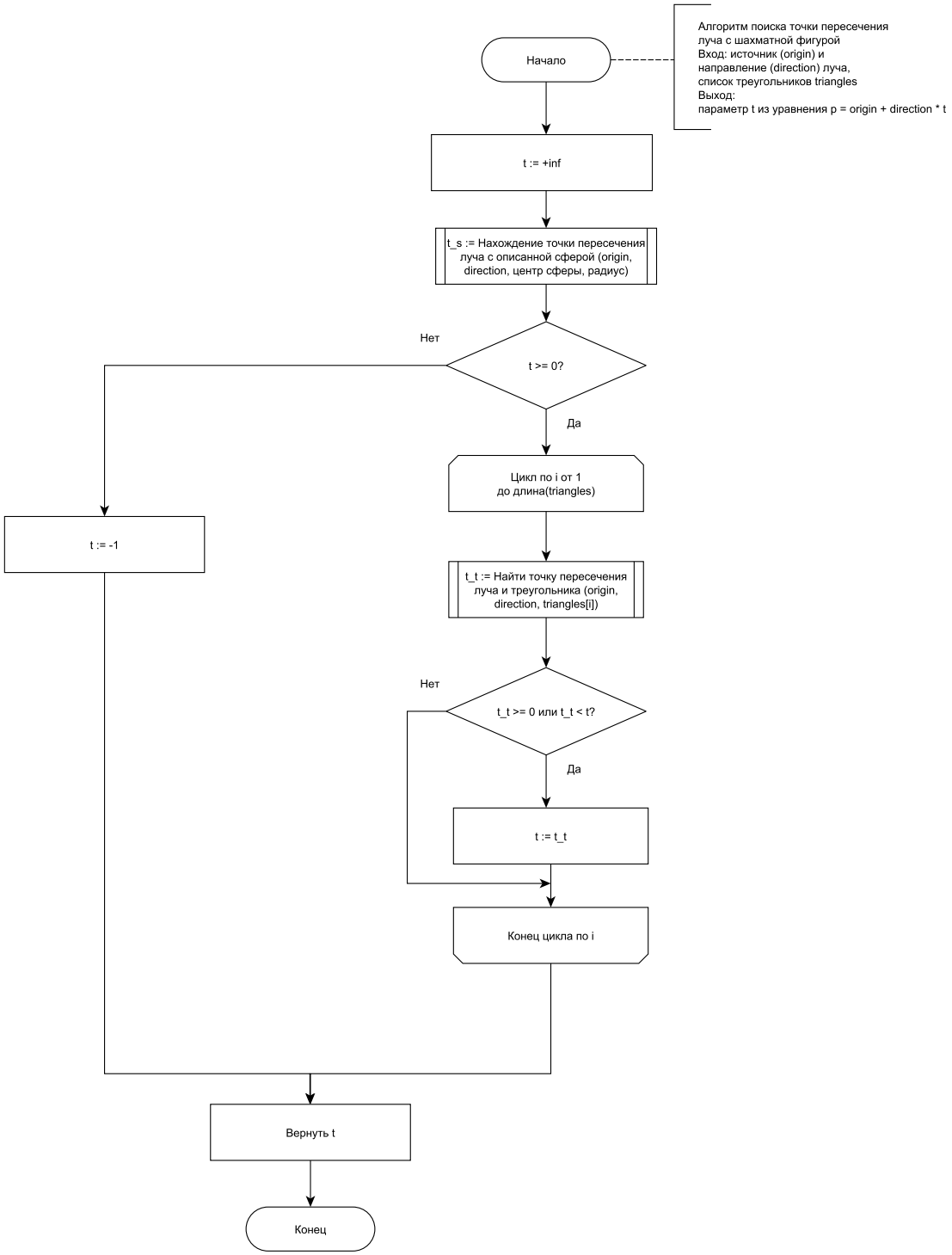

Рисунок 2.6 — Схема алгоритма нахождения точки пересечения луча и шахматной фигуры

2.3 Вывод

3 Технологическая часть

3.1 Вывод

4 Исследовательская часть

4.1 Вывод

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Простая модель освещения [Электронный ресурс]. URL: https://ychebnikkompgrafblog.wordpress.com/5-2-простая-модель-освещения/ (дата обращения 08.10.24);

2. Алгоритм Робертса удаления невидимых граней [Электронный ресурс]. URL: https://cgraph.ru/node/495 (дата обращения 06.10.24);

3. Алгоритм удаления невидимых граней, использующий z-буфер [Электронный ресурс]. URL: https://ychebnikkompgrafblog.wordpress.com/4-6-алгоритм-использующий-z-буфер/ (дата обращения 06.10.24);

4. Метод обратной трассировки лучей [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/metod-pryamoy-i-obratnoy-trassirovki/viewer (дата обращения 06.10.24);

5. Закрашивание. Рендеринг полигональных моделей [Электронный ресурс]. URL: https://intuit.ru/studies/professional_skill_improvements/1283/courses/70/lecture/21/08?page=2 (дата обращения 06.10.24);

6. Построение теней [Электронный ресурс]. URL: https://stratum.ac.ru/education/textbooks/kgrafic/addit (дата обращения 06.10.24);

7. An Introduction to Ray Tracing [Электронный ресурс]

8. Fast Minimum Storage Ray/Triangle Intersection [Электронный ресурс]. URL: https://web.archive.org/web/20170517125238/http://www.cs.virginia.edu/ gfx/Courses/2003/ImageSynth