

Визуализация геометрических моделей гранных тел в точечном исчислении

Е.В. Конопацкий^{1,А}, К.В. Рябинин^{2,В}, А.А. Бездитный^{3,С}

^А Нижегородский государственный архитектурно-строительный университет

^В Пермский государственный национальный исследовательский университет

^С Севастопольский филиал ФГБОУВО «Российский экономический университет имени Г.В. Плеханова»

¹ ORCID: 0000-0003-4798-7458, e.v.konopatskiy@mail.ru

² ORCID: 0000-0002-8353-7641, kostya.ryabinin@gmail.com

³ ORCID: 0000-0003-0528-9731, bezdytniy@gmail.com

Аннотация

В работе на примере гранных тел представлен подход к визуализации моделей геометрических тел в виде трёхпараметрического множества точек, принадлежащих трёхмерному пространству. Для визуализации геометрических тел с учётом обеспечения аппаратной поддержки графических процессоров используется метод Ray marching. Реализация этого метода предусматривает определение знаковой функции расстояния (signed distance function), которая сводится к задаче определения множества точек пересечения лучей проецирования с визуализируемым геометрическим телом. После чего для каждого пикселя экрана определяется его цвет в соответствии с тем, проходит луч через геометрическое тело или нет. Аналитическое описание геометрических тел и решение задачи их пересечения с проецирующими лучами решены в рамках математического аппарата точечного исчисления. В результате сделан вывод, что предложенный подход себя оправдывает, обеспечивая высокую производительность рендеринга и полное отсутствие артефактов при визуализации гранных тел.

Ключевые слова: твердотельное моделирование, геометрическое тело, точечное исчисление, визуализация, Ray marching, знаковая функция расстояния, трёхпараметрическое множество точек.

1. Введение

В работе [1] была сформулирована проблема визуализации твердотельных моделей, представленных в виде трёхпараметрического множества точек в трёхмерном пространстве. Источником проблемы послужила новая концепция определения геометрических тел в виде трёхпараметрического множества точек в трёхмерном пространстве [2, 3]. Проведенный в работе [1] анализ существующих подходов к визуализации трёхпараметрического множества точек, описанного системой параметрических уравнений, показал отсутствие программных решений, способных реализовать визуализацию трёхмерных тел, полученных в соответствии с новой концепцией. Учитывая аналитическую природу аппарата моделирования геометрических тел для обеспечения ускорения визуализации на существующих графических процессорах, было принято решение использовать метод Ray marching, который используется для рендеринга сцен в режиме реального времени. Метод Ray marching [4-6] по аналогии с методом Ray tracing [7-10] нашёл широкое применение при решении целого множества задач научной визуализации в отечественной и зарубежной практике. Разнообразие задач, в свою очередь, породило множество

вариантов реализации визуального представления сцен в режиме реального времени. Но в большинстве случаев задача сводится к определению функции пересечения визуализируемого объекта с лучом и определения знаковой функции расстояния (англ. Signed Distance Function [11-13]). Исходя из того, что геометрические тела в виде трёхпараметрического множества точек в трёхмерном пространстве параметризованы в точечном исчислении [14-15], для определения функции расстояния необходимо решить задачу пересечения геометрических тел с проецирующими лучами в точечном исчислении.

2. Определение точки пересечения луча с геометрическим телом

В точечном исчислении проецирующий луч удобно определить в виде отрезка прямой. Тогда задача сводится к определению точки пересечения отрезка прямой с геометрическим телом. Прямая однозначно определяется двумя точками. Тогда пучок проецирующих лучей можно задать, зафиксировав одну точку в пространстве, а вторую представить в виде текущей точки картинной плоскости.

Точечное уравнение отрезка прямой в точечном исчислении имеет следующий вид:

$$N = P\bar{t} + Qt,$$

где P и Q – точки, через которые проходит прямая, которые определяются своими координатами;

t – текущий параметр;

$\bar{t} = 1 - t$ – дополнение параметра до 1.

Такая параметризация прямой обеспечивает её прохождение через точку P при $t = 0$ и через точку Q при $t = 1$. Если параметр $t \in (-\infty; +\infty)$, то получим бесконечную прямую, проходящую через точки P и Q . А при $t \in [0; 1]$ получим непосредственно отрезок PQ . Эту особенность можно использовать для определения точек пересечения луча с геометрическим телом, если расположить одну точку перед геометрическим телом, а вторую за ним. Тогда точки пересечения луча с геометрическим телом будут попадать в диапазон значений параметра $t \in [0; 1]$. Для определения пучка лучей необходимо фиксированную точку расположить с одной стороны геометрического тела, а текущую, принадлежащую плоскости, – с другой. Такую плоскость можно задать, используя любые три точки, не лежащие на одной прямой. Например, формируя пучок лучей, проходящих через точку P (рис. 1), можно задать плоскость $A_1A_2A_3$ отношениями на сторонах треугольного симплекса:

$$Q = A_1(1 - \lambda - \mu) + A_2\lambda + A_3\mu,$$

где λ и μ – текущие параметры, которые определяют положение точки Q в плоскости $A_1A_2A_3$. В пределах внутренней части параллелограмма образованного на основе треугольника $A_1A_2A_3$ эти параметры изменяются от 0 до 1, но в общей сложности могут принадлежать интервалу $(-\infty; +\infty)$.

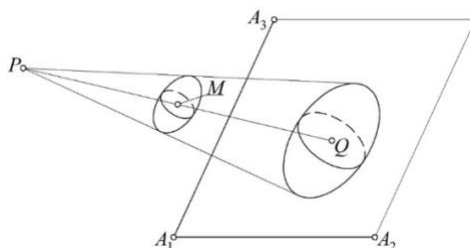


Рис. 1. Геометрическая схема определения пучка лучей в трёхмерном пространстве

На рисунке 1 текущей точкой M показано некоторое геометрическое тело. В общем виде точечное уравнение трёхмерного тела определяется следующим точечным уравнением:

$$M = Ap(u, v, w) + Bq(u, v, w) + Cr(u, v, w) + Ds(u, v, w),$$

где A, B, C и D точки трёхмерного симплекса (любые четыре точки, не лежащие в одной плоскости);

$p(u, v, w), q(u, v, w), r(u, v, w)$ и $s(u, v, w)$ – любые непрерывные и дифференцируемые функции от параметров u, v, w .

Условием принадлежности текущей точки M к пространству трёхмерного симплекса $ABCD$ является:

$$p(u, v, w) + q(u, v, w) + r(u, v, w) + s(u, v, w) = 1.$$

Исходя из этого, любую из четырёх свободных функций от параметров u, v, w можно исключить.

При пересечении отрезка прямой и геометрического тела для всех общих точек будет справедливым точечное уравнение:

$$P\bar{t} + Qt = Ap(u, v, w) + Bq(u, v, w) + Cr(u, v, w) + Ds(u, v, w).$$

Для определения параметров u, v, w и t необходимо выполнить по координатный расчёт [14]. Для трёхмерного пространства имеем:

$$\begin{cases} x_P\bar{t} + x_Qt = x_Ap(u, v, w) + x_Bq(u, v, w) + x_Cr(u, v, w) + x_Ds(u, v, w) \\ y_P\bar{t} + y_Qt = y_Ap(u, v, w) + y_Bq(u, v, w) + y_Cr(u, v, w) + y_Ds(u, v, w) \\ z_P\bar{t} + z_Qt = z_Ap(u, v, w) + z_Bq(u, v, w) + z_Cr(u, v, w) + z_Ds(u, v, w) \end{cases}$$

Таким образом, получим систему из трёх уравнений с четырьмя неизвестными. Для её решения следует понимать, что предельные значения параметров геометрического тела u, v, w определяют оболочку его поверхности, точки пересечения с которой как раз и нужно найти. Поскольку геометрическое тело может занимать относительно луча совершенно разные положения в пространстве, заранее нельзя определить, какие две из поверхностей оболочки геометрического тела пересечёт луч. Поэтому необходимо решить шесть систем – частных случаев, которые являются следствием использования предельных значений параметров. В большинстве случаев параметры точечных уравнений изменяются от 0 до 1. Тогда необходимо поочередно фиксировать значения параметров ($u=0, u=1, v=0, v=1, w=0, w=1$) и получать частные решения системы параметрических уравнений. Из всех решений необходимо выбрать только те, которые обеспечивают полученные значения параметров в пределах от 0 до 1. Таких решений будет всего два, если луч пересекает геометрическое тело, и одно – если касается.

3. Визуализация твердотельных моделей на примере гранных тел

Рассмотрим пример определения точек пересечения отрезка PQ с телом тетраэдра (рис. 2).

В работах [1, 2] приведено точечное уравнение тела тетраэдра, которое определяется точками трёхмерного симплекса $ABCD$ и тремя линейными параметрами u, v, w :

$$M = Auv\bar{w} + B\bar{v}\bar{w} + C\bar{u}v\bar{w} + Dw,$$

где $\bar{u} = 1 - u, \bar{v} = 1 - v$ и $\bar{w} = 1 - w$.

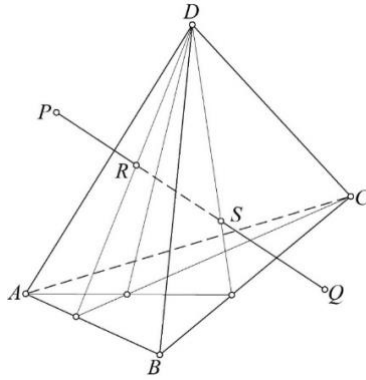


Рис. 2. Геометрическая схема пересечения отрезка PQ с телом тетраэдра

Исходя из описанного выше метода, получим:

$$P\bar{t} + Qt = Auv\bar{w} + B\bar{v}w + C\bar{u}v\bar{w} + Dw$$

⇓

$$\begin{cases} x_P\bar{t} + x_Qt = x_Auv\bar{w} + x_B\bar{v}w + x_C\bar{u}v\bar{w} + x_Dw \\ y_P\bar{t} + y_Qt = y_Auv\bar{w} + y_B\bar{v}w + y_C\bar{u}v\bar{w} + y_Dw \\ z_P\bar{t} + z_Qt = z_Auv\bar{w} + z_B\bar{v}w + z_C\bar{u}v\bar{w} + z_Dw \end{cases}$$

⇓

$$\begin{cases} x_Auv\bar{w} + x_B\bar{v}w + x_C\bar{u}v\bar{w} + x_Dw - x_P = (x_Q - x_P)t \\ y_Auv\bar{w} + y_B\bar{v}w + y_C\bar{u}v\bar{w} + y_Dw - y_P = (y_Q - y_P)t \\ z_Auv\bar{w} + z_B\bar{v}w + z_C\bar{u}v\bar{w} + z_Dw - z_P = (z_Q - z_P)t \end{cases}$$

Далее необходимо поочерёдно фиксировать предельные значения текущих параметров тела тетраэдра ($u = 0, u = 1, v = 0, v = 1, w = 0, w = 1$) и решать полученные системы уравнений.

Для проведения вычислительных экспериментов принимаем следующие тестовые координаты исходных точек:

$$A(0;0;0), B(3;5;0), C(8;3;0), D(4;2;7), P(7;5;4), Q(0;1;2).$$

Используя координаты исходных точек, получим 6 систем параметрических уравнений. Системы при $v=0$ и $w=1$ не имеют решения. При $v=1$ и $w=0$ – полученные решения выходят за диапазон значений параметров $[0;1]$. При $u=0$

значения параметров удовлетворяют требуемым условиям $t = \frac{9}{22}, v = \frac{1}{4}, w = \frac{5}{11}$, а при

$u=1$ – $t = \frac{12}{19}, v = \frac{4}{9}, w = \frac{52}{133}$. Таким образом получается, что на интервале значений

параметра $t \in \left[\frac{9}{22}; \frac{12}{19} \right]$ часть отрезка PQ находится внутри тетраэдра, а на интервалах

$t \in \left(-\infty; \frac{9}{22} \right)$ и $t \in \left(\frac{12}{19}; +\infty \right)$ – за пределами тетраэдра $ABCD$.

Предложенный подход может быть использован для визуализации не только гранных, но и других геометрических тел. Вместе с тем, применительно к гранным телам можно реализовать и другой подход, основанный на пересечении проецирующего луча с плоскостью.

В общем случае, отрезок PQ (рис. 2) может иметь две точки пересечения с пирамидой $ABCD$, одну точку пересечения (когда отрезок пересекает ребра и вершины пирамиды) и вообще не иметь точек пересечения.

В качестве частного случая определим точку S пересечения отрезка PQ с гранью BCD . Для этого зададим точечное уравнение прямой с помощью параметра λ : $S = P\bar{\lambda} + Q\lambda$, где $\bar{\lambda} = 1 - \lambda$. Точки P и Q определяются в симплексе $ABCD$ с помощью соответствующих параметров:

$$P = Ap_P + Bq_P + Cr_P + Ds_P \Rightarrow \begin{cases} x_P = x_A p_P + x_B q_P + x_C r_P + x_D s_P \\ y_P = y_A p_P + y_B q_P + y_C r_P + y_D s_P, \\ z_P = z_A p_P + z_B q_P + z_C r_P + z_D s_P \end{cases}$$

$$Q = Ap_Q + Bq_Q + Cr_Q + Ds_Q \Rightarrow \begin{cases} x_Q = x_A p_Q + x_B q_Q + x_C r_Q + x_D s_Q \\ y_Q = y_A p_Q + y_B q_Q + y_C r_Q + y_D s_Q, \\ z_Q = z_A p_Q + z_B q_Q + z_C r_Q + z_D s_Q \end{cases}$$

где $p_P = \frac{V_{PBCD}}{V_{ABCD}}$, $q_P = \frac{V_{APCD}}{V_{ABCD}}$, $r_P = \frac{V_{ABPD}}{V_{ABCD}}$, $s_P = \frac{V_{ABCP}}{V_{ABCD}} = 1 - p_P - q_P - r_P$,

$$p_Q = \frac{V_{QBCD}}{V_{ABCD}}, q_Q = \frac{V_{AQCD}}{V_{ABCD}}, r_Q = \frac{V_{ABQD}}{V_{ABCD}}, s_Q = \frac{V_{ABCQ}}{V_{ABCD}} = 1 - p_Q - q_Q - r_Q.$$

Объёмы соответствующих тетраэдров V_i можно легко определить через координаты вершин. При этом умножение на $1/6$ можно упразднить, т.к. оно всё равно сократится. В качестве примера приведём определитель для вычисления объёма тетраэдра $ABCD$. Все остальные объёмы ориентированных тетраэдров в симплексе $ABCD$ находятся аналогично.

$$V_{ABCD} = \begin{vmatrix} x_A - x_D & y_A - y_D & z_A - z_D \\ x_B - x_D & y_B - y_D & z_B - z_D \\ x_C - x_D & y_C - y_D & z_C - z_D \end{vmatrix}.$$

Далее определяем такое значение параметра λ , при котором объём тетраэдра $SBCD$ равен нулю. В соответствии с S -теоремой точечного исчисления [14], получим:

$$\begin{vmatrix} (p_Q - p_P)\lambda + p_P & (q_Q - q_P)\lambda + q_P & (r_Q - r_P)\lambda + r_P & (s_Q - s_P)\lambda + s_P \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 0.$$

⇓

$$\lambda = \frac{p_P}{p_P - p_Q}; \quad \bar{\lambda} = \frac{p_Q}{p_Q - p_P}.$$

$$S = P \frac{p_Q}{p_Q - p_P} + Q \frac{p_P}{p_P - p_Q} \Rightarrow \begin{cases} x_S = x_P \frac{p_Q}{p_Q - p_P} + x_Q \frac{p_P}{p_P - p_Q} \\ y_S = y_P \frac{p_Q}{p_Q - p_P} + y_Q \frac{p_P}{p_P - p_Q}. \\ z_S = z_P \frac{p_Q}{p_Q - p_P} + z_Q \frac{p_P}{p_P - p_Q} \end{cases}$$

Полученное уравнение точки S можно также определить в исходном симплексе $ABCD$ путём замены соответствующими уравнениями точек P и Q , но для визуализации тетраэдра в этом нет необходимости.

Аналогичным образом определяем точку R :

$$R = P \frac{r_Q}{r_Q - r_P} + Q \frac{r_P}{r_P - r_Q} \Rightarrow \begin{cases} x_R = x_P \frac{r_Q}{r_Q - r_P} + x_Q \frac{r_P}{r_P - r_Q} \\ y_R = y_P \frac{r_Q}{r_Q - r_P} + y_Q \frac{r_P}{r_P - r_Q} \\ z_R = z_P \frac{r_Q}{r_Q - r_P} + z_Q \frac{r_P}{r_P - r_Q} \end{cases}$$

В результате получаем следующие условия:

$$\lambda \in \left[\frac{r_P}{r_P - r_Q}; \frac{p_P}{p_P - p_Q} \right] - \text{отрезок луча лежит внутри тела тетраэдра.}$$

$$\lambda \in \left(-\infty; \frac{r_P}{r_P - r_Q} \right) \cap \left(\frac{p_P}{p_P - p_Q}; +\infty \right) - \text{отрезок луча не принадлежит телу тетраэдра.}$$

Если исходить из взаимного положения центра проецирования и картинной плоскости, как показано на рисунке 1, то $\lambda \in [0;1]$. Вместе с тем возможны частные случаи, когда проецирующий луч принадлежит одной из граней тетраэдра, совпадает с его ребром или проходит через вершину. На описанный выше вычислительный алгоритм это никак не влияет.

4. Программная реализация и результаты проведения вычислительных экспериментов

Для проведения вычислительных экспериментов по визуализации гранных тел была написана тестовая программа на языке GLSL, содержащая 167 строк кода (исходный код доступен по лицензии MIT на GitHub: <https://github.com/icosaeder/tpps-raycast>). Визуализация выполнена в браузере с использованием бесплатного ресурса ShaderToy (<https://www.shadertoy.com>), предоставляющего удобный инструментарий для запуска программ, предназначенных для исполнения на видеокарте.

Цель тестовой программы состояла в проверке эффективности рендеринга гранных тел, заданных в точечном исчислении. Блок-схема разработанного алгоритма представлена на рисунке 3.

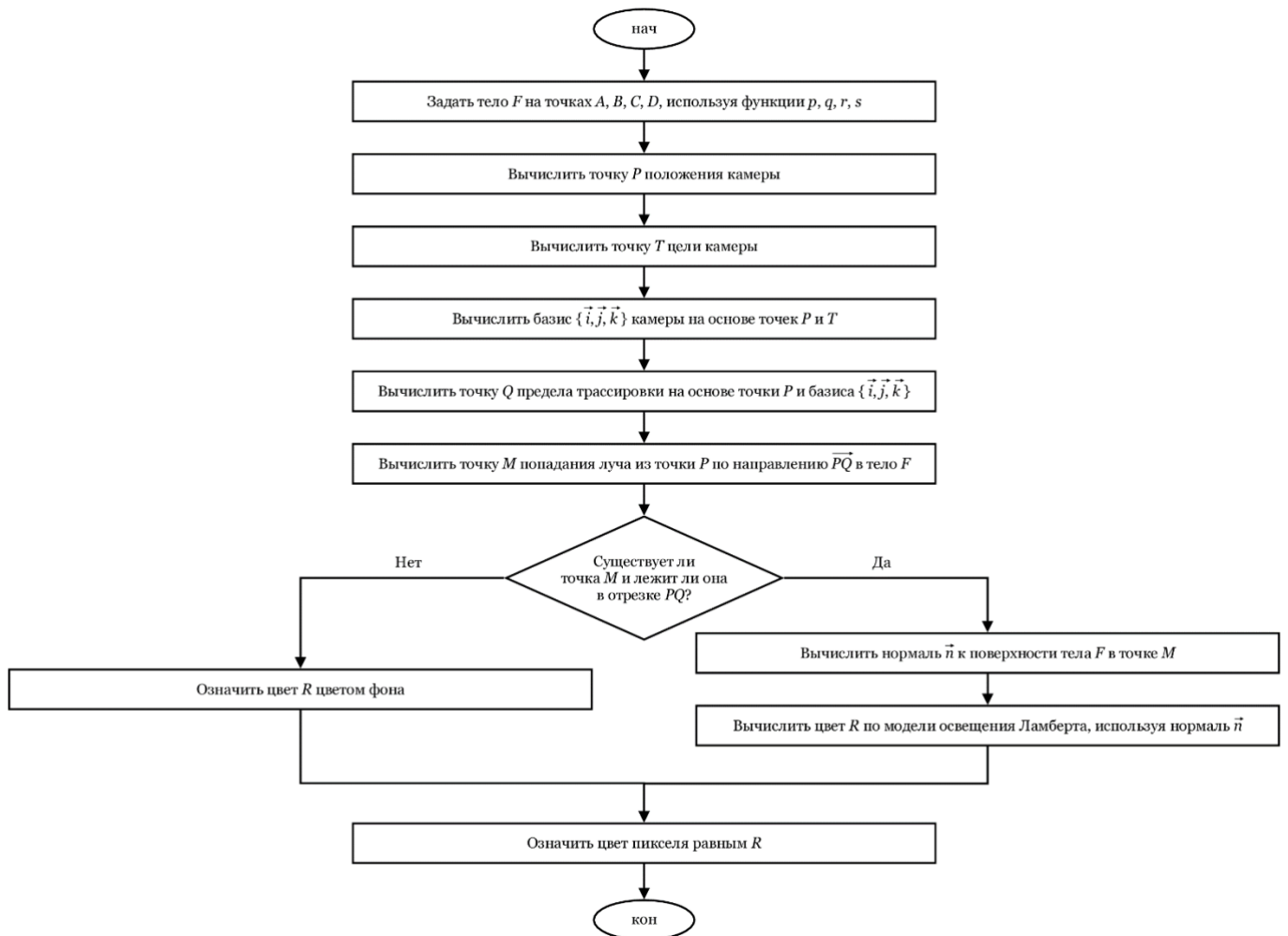


Рис. 3. Блок-схема предложенного алгоритма визуализации

На вход этого алгоритма подаётся описание гранного тела F в точечном исчислении посредством задания точек A, B, C, D и функций p, q, r, s , как описано в разделе 2. На выходе получается цвет отдельного пикселя изображения. Выполнившись для каждого пикселя изображения, этот алгоритм производит визуализацию заданного гранного тела F в ракурсе, определяемом точками P, T и Q .

Следует отметить, что хотя представленный алгоритм записан в последовательном виде, он предполагает неявный параллелизм на видеокarte в соответствии с парадигмой SIMD (англ. Single Instruction Multiple Data – одна инструкция, много данных). Параллелизм достигается путём одновременного выполнения этого алгоритма для некоторого подмножества пикселей изображения. Размер подмножества динамически определяется видеокartой на основе её аппаратных возможностей.

В основе программной реализации лежит численное решение систем уравнений итерационным методом Ньютона [16]. На каждой i -й итерации считается новое приближение решения $\sigma^{(i)} = \{x^{(i)}, y^{(i)}, z^{(i)}\}$ системы F по формуле:

$$\sigma^{(i)} = \sigma^{(i-1)} - J(\sigma^{(i-1)})^{-1} F(\sigma^{(i-1)}),$$

где J – Якобиан системы F , который считается численно по формуле

$$J(\sigma) = \begin{pmatrix} \left(\frac{\partial F}{\partial x}(\sigma)\right)_x & \left(\frac{\partial F}{\partial y}(\sigma)\right)_x & \left(\frac{\partial F}{\partial z}(\sigma)\right)_x \\ \left(\frac{\partial F}{\partial x}(\sigma)\right)_y & \left(\frac{\partial F}{\partial y}(\sigma)\right)_y & \left(\frac{\partial F}{\partial z}(\sigma)\right)_y \\ \left(\frac{\partial F}{\partial x}(\sigma)\right)_z & \left(\frac{\partial F}{\partial y}(\sigma)\right)_z & \left(\frac{\partial F}{\partial z}(\sigma)\right)_z \end{pmatrix}.$$

В роли неизвестных x и y попеременно выступают параметры u , v , w , а в роли неизвестной z всегда выступает параметр t . Подстановки осуществляются по следующей схеме для всех 6 систем, описанных в разделе 2:

1. $u = 0, x = v, y = w, z = t$.
2. $u = 1, x = v, y = w, z = t$.
3. $v = 0, x = u, y = w, z = t$.
4. $v = 1, x = u, y = w, z = t$.
5. $w = 0, x = u, y = v, z = t$.
6. $w = 1, x = u, y = v, z = t$.

Частные производные в составе Якобиана вычисляются в соответствии с разностной схемой, например:

$$\frac{\partial F}{\partial x}(\sigma) = \frac{F(\{x + \Delta, y, z\}) - F(\{x - \Delta, y, z\})}{2\Delta}.$$

Экспериментально было установлено, что приемлемая точность решения достигается при $\Delta = 10^{-3}$.

Первое приближение $\sigma^{(0)}$ берётся в нашем случае тождественно равным нулю. Однако для некоторых F такое $\sigma^{(0)}$ может привести к потере решения, так как Якобиан на первой итерации может оказаться вырожденным, что приведёт к невозможности вычисления $\sigma^{(1)}$ ввиду отсутствия у вырожденного Якобиана обратной матрицы. Для защиты от такой ситуации в коде реализована проверка на вырожденность Якобиана на первой итерации, и, если он вырожден, вместо него подставляется единичная матрица, что корректирует первое приближение.

Метод Ньютона имеет два условия остановки. Первое – это достижение заданной точности решения, что определяется по формуле:

$$\sqrt{(x^{(i)} - x^{(i-1)})^2 + (y^{(i)} - y^{(i-1)})^2 + (z^{(i)} - z^{(i-1)})^2} < \Delta.$$

Второе условие остановки состоит в превышении порогового числа шагов, что свидетельствует об отсутствии у системы решения. В нашем случае экспериментально подобрано пороговое значение в 100 шагов. Эта константа обусловлена тем, что, согласно нашим наблюдениям, для гранных тел соответствующие системы решаются в среднем за 7–10 шагов (в случае, если имеют решение). Таким образом, порог в 100 шагов с запасом обеспечивает надёжную защиту от потери решения, и, при этом, не сильно перегружает графический процессор лишними итерациями.

Для того, чтобы у пользователя была возможность рассмотреть визуализируемое тело со всех сторон, реализована модель интерактивной орбитальной камеры. Позиция камеры, выражаемая точкой P (которая соответствует началу отрезка, упомянутого в разделе 2), вычисляется по следующей формуле:

$$P = \{\sin \theta \cdot \sin \varphi, \cos \varphi, \cos \theta \cdot \sin \varphi\} \cdot r,$$

где θ, φ – полярные углы, вычисленные на основе координат курсора мыши в области визуализации, r – радиус орбиты камеры, в нашем случае вычисляемый как удвоенное максимальное значение координат точек A, B, C, D , задающих трёхмерный симплекс для построения визуализируемого тела.

Центром сцены (точкой, на которую «смотрит» камера) выбирается точка T – центр масс тела. Например, для тетраэдра:

$$T = \frac{A+B+C+D}{4} \Rightarrow \begin{cases} x_T = \frac{x_A + x_B + x_C + x_D}{4} \\ y_T = \frac{y_A + y_B + y_C + y_D}{4} \\ z_T = \frac{z_A + z_B + z_C + z_D}{4} \end{cases}$$

Орты системы координат камеры определяются следующим образом:

$$\begin{aligned} \vec{k} &= \frac{T - P}{|T - P|}, \\ \vec{i} &= \frac{\{0,1,0\} \times \vec{k}}{|\{0,1,0\} \times \vec{k}|}, \\ \vec{j} &= \frac{\vec{i} \times \vec{k}}{|\vec{i} \times \vec{k}|}. \end{aligned}$$

Точка Q , соответствующая концу отрезка, упомянутого в разделе 2, определяется по формуле:

$$Q = P + \frac{\vec{i} \cdot p_x + \vec{j} \cdot p_y + \vec{k}}{|\vec{i} \cdot p_x + \vec{j} \cdot p_y + \vec{k}|} d_{max},$$

где p_x, p_y – нормализованные координаты экранного пикселя, через который осуществляется трассировка (и цвет которого необходимо определить), d_{max} – максимальная удалённость трассировки (в нашем случае экспериментально подобрана константа 100).

Для увеличения визуального качества отображаемого гранного тела и обеспечения восприятия его объёмности пользователем, реализована модель освещения по Ламберту. Нормаль \vec{n} при этом вычисляется как градиент функции расстояния с использованием разностной схемы:

$$\begin{aligned} \vec{d} &= \{\delta(P + \vec{i} \cdot \Delta, Q) - \delta(P, Q), \delta(P + \vec{j} \cdot \Delta, Q) - \delta(P, Q), \delta(P + \vec{k} \cdot \Delta, Q) - \delta(P, Q)\}, \\ \vec{n} &= \frac{\vec{d}}{|\vec{d}|}, \end{aligned}$$

где δ – функция определения расстояния до тела на отрезке PQ .

Все описанные выше расчёты выполняются для каждого пикселя экрана в режиме SIMD-параллелизма. Результаты визуализации гранных тел представлены на рисунках 4 и 5.

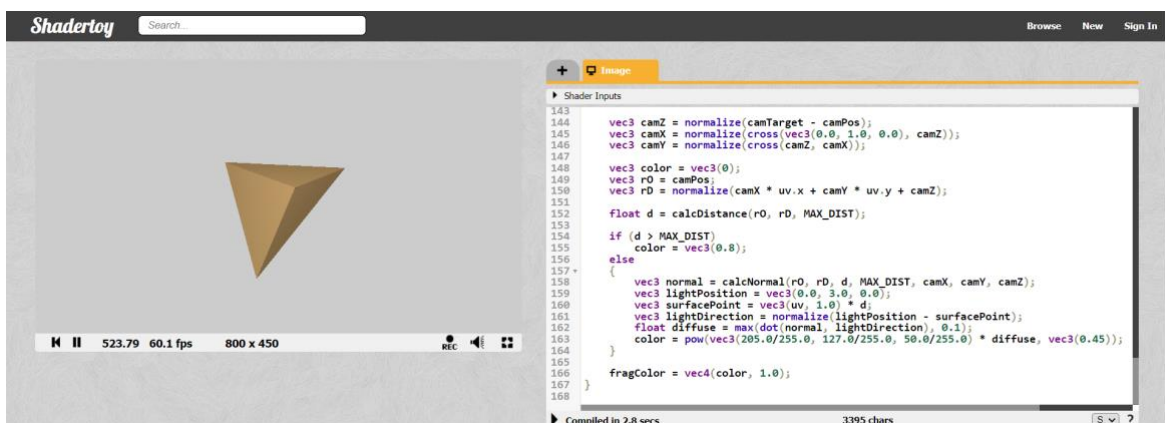


Рис. 4. Визуализация тела тетраэдра в среде сервиса ShaderToy

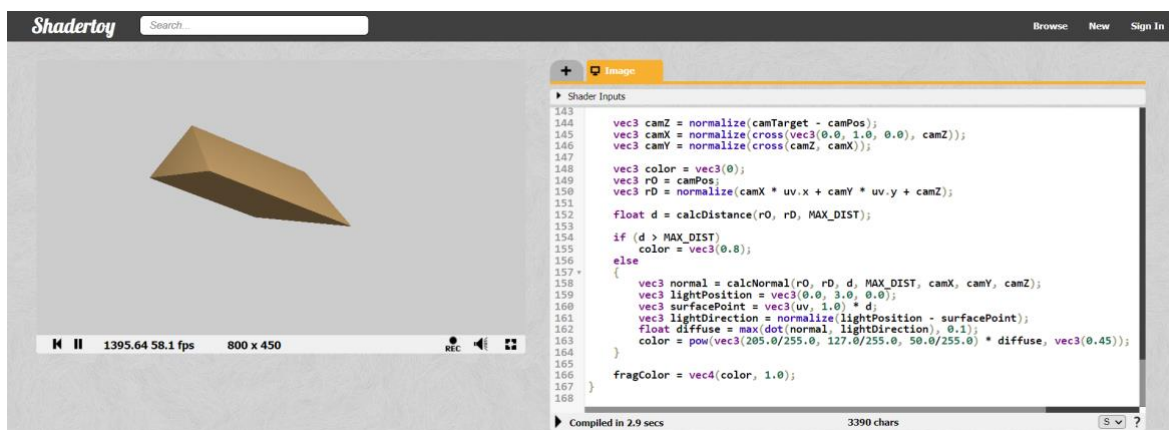


Рис. 5. Визуализация тела призмы в среде сервиса ShaderToy

5. Заключение

Как видно из результатов вычислительных экспериментов (рис. 4–5), предложенный подход к визуализации твердотельных геометрических моделей в достаточной степени себя оправдывает. Производительность визуализации обеспечивает частоту обновления изображения около 60 кадров в секунду, что соответствует скорости обновления современных неспециализированных мониторов. При этом сохраняется высокая точность изображения без каких-либо видимых артефактов. Вместе с тем, при визуализации криволинейных тел появление артефактов всё-таки возможно. Это связано с тем, что метод Ньютона не может обеспечить достаточную точность численного решения системы уравнений. Исходя из этого перспективой дальнейших исследований является усовершенствование метода определения знаковой функции расстояния (англ. Signed Distance Function) для визуализации криволинейных геометрических тел с последующей интеграцией разработанных программных решений в систему научной визуализации SciVi [17-19].

Предложенный подход к визуализации твердотельных геометрических моделей в первую очередь ориентирован на применение в системах автоматизированного проектирования, твердотельного и информационного (англ. BIM – Building Information Modeling) моделирования. Однако, учитывая широкое применение трёхмерных моделей в различных отраслях народного хозяйства, после усовершенствования предложенный подход может найти применение не только в машиностроении, строительстве и архитектуре, но и в науке, медицине, дизайне, образовании, а также в рекламной, кинематографической и видеоигровой индустрии. Перспективным видится использование предложенного подхода к определению твердотельных моделей в точечном исчислении для разработки новой технологии генерации полноценных объёмных изображений в трёхмерном пространстве по аналогии с голографическими.

Список литературы

1. Konopatskiy E.V., Bezditnyi A.A. The Problem of Visualizing Solid Models as a Three-Parameter Point Set. *Scientific Visualization*, 2022. Vol. 14. No. 2. pp. 49-61. DOI: 10.26583/sv.14.2.05.
2. Konopatskiy E.V., Bezditnyi A.A., Lagunova M.V., Naidysh A.V. Principles of solid modelling in point calculus. *IoP conference series: Journal of Physics: Conf. Series* 1901 (2021) 012063. DOI: 10.1088/1742-6596/1901/1/012063.
3. Konopatskiy E.V., Bezditnyi A.A. Solid modeling of geometric objects in point calculus. *Proceedings of the 31st International Conference on Computer Graphics and Vision (GraphiCon 2021) Nizhny Novgorod, Russia, September 27-30, 2021*. Vol. 3027. pp. 666-672. DOI: 10.20948/graphicon-2021-3027-666-672.

4. Kettunen M., D'Eon E., Pantaleoni J., Novák J. An unbiased ray-marching transmittance estimator. *ACM Transactions on Graphics*. 2021. 40(4). DOI: 10.1145/3450626.3459937.
5. Coulon R., Matsumoto E. A., Segerman H., Trettel S. J. Ray-marching thurston geometries. *Experimental Mathematics*. 2022. DOI: 10.1080/10586458.2022.2030262.
6. Xu T., Zeng W., Wu E. Viewport-resolution independent anti-aliased ray marching on interior faces in cube-map space. Paper presented at the Proceedings - SIGGRAPH Asia 2021 Technical Communications. SA 2021. DOI: 10.1145/3478512.3488598.
7. Debelov V.A., Kozlov D.V. Visualization of Light Polarization to Debug Ray Tracing Algorithms. *Scientific Visualization*. 2013. Vol. 5. No 4. pp. 71-87.
8. Жданов Д.Д., Ершов С.В., Дерябин Н.Б. Объектно-ориентированная модель фотореалистичной визуализации сложных сцен. *Научная визуализация*. 2013. Т. 5. № 4. С. 88-117.
9. Sokolov V.G., Voloboy A.G., Potemin I.S., Galaktionov V.A. Overview of BSDF Reconstruction Methods for Rough Surfaces. *Scientific Visualization*. 2022. Vol. 14. No 3. pp. 132-151. DOI: 10.26583/sv.14.3.10.
10. Wald I., Zellmann S., Usher W., Morriscal N., Lang U., Pascucci V. Ray tracing structured AMR data using exabricks. *IEEE Transactions on Visualization and Computer Graphics*. 2021. 27(2). pp. 625-634. DOI: 10.1109/TVCG.2020.3030470.
11. Zhang K., Luan F., Wang Q., Bala K., Snavely N. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. Paper presented at the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2021. 5449-5458. DOI: 10.1109/CVPR46437.2021.00541.
12. Shen T., Gao J., Yin K., Liu M., Fidler S. Deep marching tetrahedra: A hybrid representation for high-resolution 3D shape synthesis. Paper presented at the Advances in Neural Information Processing Systems. 2021. No. 8. pp. 6087-6101.
13. Soukov S.A. Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid bodies movement. *Scientific Visualization*. 2021. 12(5). DOI: 10.26583/SV.12.5.08.
14. Конопацкий Е.В., Бездитный А.А. Точечные инструменты геометрического моделирования, инвариантные относительно параллельного проецирования. *Геометрия и графика*. 2021. Т. 9. № 4. С. 11-21. DOI: 10.12737/2308-4898-2022-9-4-11-21.
15. Конопацкий Е.В., Бездитный А.А., Кокарева Я.А., Кучеренко В.В. Особенности визуализации геометрических объектов в БН-исчислении. *Научная визуализация*, 2020. Т.12. №2. С.98-109. DOI: 10.26583/sv.12.2.08.
16. Remani C. *Numerical Methods for Solving Systems of Nonlinear Equations*. Lakehead University. 2013. 38 p.
17. Рябинин К.В., Баранов Д.А., Белоусов К.И. Интеграция информационной системы Семограф и визуализатора SCIVI для решения задач экспертного анализа языкового контента. *Научная визуализация*. 2017. Т. 9. № 4. С. 67-77. DOI: 10.26583/sv.9.4.07.
18. Ryabinin K.V., Kolesnik M.A., Akhtamzyan A.I., Sudarikova E.V. Cyber-Physical Museum Exhibits Based on Additive Technologies, Tangible Interfaces and Scientific Visualization. *Scientific Visualization*. 2019. Vol. 11. No 4. pp. 27-42. DOI: 10.26583/sv.11.4.03.
19. Ryabinin K.V., Belousov K.I. Visual Analytics of Gaze Tracks in Virtual Reality Environment. *Scientific Visualization*. 2021. Vol. 13. No 2. pp. 50-66. DOI: 10.26583/sv.13.2.04.

Visualization of Geometric Models of Faceted Solids in Point Calculus

E.V. Konopatskiy^{1,A}, K.V. Ryabinin^{2,B}, A.A. Bezdityniy^{3,C}

^A Nizhny Novgorod State University of Architecture and Civil Engineering

^B Perm State University

^C Plekhanov Russian University of Economics Sevastopol branch

¹ ORCID: 0000-0003-4798-7458, e.v.konopatskiy@mail.ru

² ORCID: 0000-0002-8353-7641, kostya.ryabinin@gmail.com

³ ORCID: 0000-0003-0528-9731, bezdytniy@gmail.com

Abstract

The paper considers the case of faceted solids and discusses visualisation of geometric solids in the form of a three-parameter set of points which belongs to a three-dimensional space. To visualize geometric solids, taking advantage of the modern GPU hardware acceleration, the Ray marching method is used. The implementation considers the definition of a signed distance function, which is reduced to the task of determining the set of intersection points of the projection rays with the rendered geometric solid. After that, for each pixel of the screen, its color is determined in accordance with whether the ray passes through the geometric solid or not. The analytical description of geometric solids and the solution of their intersection problem with projecting rays are solved within the framework of the point calculus mathematical apparatus. As a result, it was concluded that the proposed approach justifies itself, providing high rendering performance and the complete absence of visual artifacts when rendering faceted solids.

Keywords: solid-state modeling, geometric solid, point calculus, visualization, Ray marching, distance sign function, three-parameter set of points.

References

1. Konopatskiy E.V., Bezdityniy A.A. The Problem of Visualizing Solid Models as a Three-Parameter Point Set. *Scientific Visualization*, 2022. Vol. 14. No. 2. pp. 49-61. DOI: [10.26583/sv.14.2.05](https://doi.org/10.26583/sv.14.2.05).
2. Konopatskiy E.V., Bezdityniy A.A., Lagunova M.V., Naidysh A.V. Principles of solid modelling in point calculus. *IoP conference series: Journal of Physics: Conf. Series* 1901 (2021) 012063. DOI: [10.1088/1742-6596/1901/1/012063](https://doi.org/10.1088/1742-6596/1901/1/012063).
3. Konopatskiy E.V., Bezdityniy A.A. Solid modeling of geometric objects in point calculus. *Proceedings of the 31st International Conference on Computer Graphics and Vision (GraphiCon 2021) Nizhny Novgorod, Russia, September 27-30, 2021*. Vol. 3027. pp. 666-672. DOI: [10.20948/graphicon-2021-3027-666-672](https://doi.org/10.20948/graphicon-2021-3027-666-672).
4. Kettunen M., D'Eon E., Pantaleoni J., Novák J. An unbiased ray-marching transmittance estimator. *ACM Transactions on Graphics*. 2021. 40(4). DOI: [10.1145/3450626.3459937](https://doi.org/10.1145/3450626.3459937).
5. Coulon R., Matsumoto E. A., Segerman H., Trettel S. J. Ray-marching thurston geometries. *Experimental Mathematics*. 2022. DOI: [10.1080/10586458.2022.2030262](https://doi.org/10.1080/10586458.2022.2030262).
6. Xu T., Zeng W., Wu E. Viewport-resolution independent anti-aliased ray marching on interior faces in cube-map space. Paper presented at the *Proceedings - SIGGRAPH Asia 2021 Technical Communications*. SA 2021. DOI: [10.1145/3478512.3488598](https://doi.org/10.1145/3478512.3488598).
7. Debelov V.A., Kozlov D.V. Visualization of Light Polarization to Debug Ray Tracing Algorithms. *Scientific Visualization*. 2013. Vol. 5. No 4. pp. 71-87.

8. Zhdanov D.D., Ershov S.V., Deryabin N.B. Object-oriented model of photorealistic visualization of complex scenes. *Scientific Visualization*. 2013. Vol. 5. No. 4. pp. 88-117.
9. Sokolov V.G., Voloboy A.G., Potemin I.S., Galaktionov V.A. Overview of BSDF Reconstruction Methods for Rough Surfaces. *Scientific Visualization*. 2022. Vol. 14. No 3. pp. 132-151. DOI: 10.26583/sv.14.3.10.
10. Wald I., Zellmann S., Usher W., Morrical N., Lang U., Pascucci V. Ray tracing structured AMR data using exabricks. *IEEE Transactions on Visualization and Computer Graphics*. 2021. 27(2). pp. 625-634. DOI: 10.1109/TVCG.2020.3030470.
11. Zhang K., Luan F., Wang Q., Bala K., Snavely N. PhySG: Inverse rendering with spherical gaussians for physics-based material editing and relighting. Paper presented at the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2021. 5449-5458. DOI: 10.1109/CVPR46437.2021.00541.
12. Shen T., Gao J., Yin K., Liu M., Fidler S. Deep marching tetrahedra: A hybrid representation for high-resolution 3D shape synthesis. Paper presented at the Advances in Neural Information Processing Systems. 2021. No. 8. pp. 6087-6101.
13. Soukov S.A. Combined signed distance calculation algorithm for numerical simulation of physical processes and visualization of solid solids movement. *Scientific Visualization*. 2021. 12(5). DOI: 10.26583/SV.12.5.08.
14. Konopatskiy E.V., Bezdityni A.A. Point tools of geometric modeling, invariant relating to parallel projection. *Geometry & Graphics*. 2021. Vol. 9. No. 4. pp. 11-21. DOI: 10.12737/2308-4898-2022-9-4-11-21.
15. Konopatskiy E.V., Bezdityni A.A., Kokareva Ya.A., Kucherenko V.V. Features visualization of geometric objects in the BN-calculus. *Scientific Visualization*. 2020. Vol. 12. No. 2. pp. 98-109. DOI: 10.26583/sv.12.2.08.
16. Remani C. *Numerical Methods for Solving Systems of Nonlinear Equations*. Lakehead University. 2013. 38 p.
17. Ryabinin K.V., Baranov D.A., Belousov K.I. Integration of Semograph information system and SCIVI visualizer for solving the tasks of lingual content expert analysis. *Scientific Visualization*. 2017. Vol. 9. No. 4. pp. 67-77. DOI: 10.26583/sv.9.4.07.
18. Ryabinin K.V., Kolesnik M.A., Akhtamzyan A.I., Sudarikova E.V. Cyber-Physical Museum Exhibits Based on Additive Technologies, Tangible Interfaces and Scientific Visualization. *Scientific Visualization*. 2019. Vol. 11. No 4. pp. 27-42. DOI: 10.26583/sv.11.4.03.
19. Ryabinin K.V., Belousov K.I. Visual Analytics of Gaze Tracks in Virtual Reality Environment. *Scientific Visualization*. 2021. Vol. 13. No 2. pp. 50-66. DOI: 10.26583/sv.13.2.04.