УДК 004.925.3
DOI: 10.25686/978-5-8158-2474-4-2025-286-297

Трассируемые мультиобъекты на основе 3D-гауссианов

П. Ю. Тимохин, М. В. Михайлюк
Научно-исследовательский институт системных исследований
Национального исследовательского центра «Курчатовский институт», Москва, Россия

Аннотация. Важным строительным блоком современного виртуального окружения является мультиобъект – совокупность виртуального объекта-образца и наборов отличительных признаков его экземпляров. Данная работа посвящена направлению визуализации сложных процедурных мультиобъектов с помощью аппаратно-ускоренной трассировки лучей. В частности, рассматривается задача построения трассируемых мультиобъектов, основанных на процедурных примитивах нового типа – 3D-гауссианах. В работе исследуются особенности таких примитивов, предлагаются методы и подходы к построению эффективной модели облака 3D-гауссианов, состоящей из ограничивающих параллелепипедов. Выполнена апробация предложенных решений, включающая сравнительный анализ качества и скорости визуализации отдельного экземпляра образца травы, а также обширного массива, содержащего 4 млн экземпляров. Полученные решения могут быть использованы в системах виртуального окружения, научной визуализации, при разработке визуализаторов нового поколения, основанных на 3D-гауссианах, и др.

***Ключевые слова*:** виртуальное окружение, мультиобъектная визуализация, облако точек, 3D-гауссиан, трассировка лучей, GPU

Traceable multiobjects based on 3D gaussians

P. Y. Timokhin, M. V. Mikhaylyuk
Scientific Research Institute for System Analysis of the National Research Centre
"Kurchatov Institute", Moscow, Russia

Abstract. An essential building block of modern virtual environment is the multiobject, defined as a composition of a sample virtual object and sets of distinctive features of its instances. This work is devoted to the visualization of complex procedural multiobjects by means of hardware-accelerated ray tracing. In particular, the task of constructing traceable multiobjects based on a new type of procedural primitives, 3D Gaussians, is considered. The paper examines the features of such primitives, and suggests methods and approaches for constructing an effective 3D Gaussian cloud model consisting of bounding boxes. The proposed solutions were tested, including a comparative analysis of the quality and speed of visualization of a single grass sample, as well as an extensive array containing 4 million instances. The obtained solutions can be used in virtual environment systems, scientific visualization, in the development of a new generation of visualizers based on 3D Gaussians, etc.

Keywords: virtual environment, multiobject visualization, point cloud, 3D-Gaussian, ray tracing, GPU

Введение

Современное виртуальное окружение строится на основе гибридного подхода, при котором традиционные трехмерные полигональные сцены объединяются с процедурными объектами, вычисляемыми в процессе визуализации [1, 2]. В контексте моделирования открытых пространств [3] особенно востребованы процедурные *мультиобъекты*, работающие по принципу «многое в малом» (от лат. multum in parvo), которые в реальном времени развертываются во множество экземпляров, в общем случае отличающихся друг от друга. Данная тенденция получила мощную поддержку в архитектуре аппаратно-ускоренной трассировки лучей [4, 5], реализованной в современных графических процессорах (GPUs), в рамках которой появились *трассируемые мультиобъекты*.

Базовая концепция трассируемого мультиобъекта [6] состоит из двух основных компонент (рис. 1). Первая компонента (далее ***ядро***) описывает объект-образец и включает в себя его процедурную модель[1], а также его модель в виде ограничивающих параллелепипедов (axis-aligned bounding boxes, AABBs). Вторая компонента (далее ***модификатор***) отвечает за вариативность экземпляра объекта-образца и включает в себя базовый набор его отличительных признаков (положение, ориентация, масштаб). В процессе синтеза кадра визуализации модификаторы экземпляров применяются к ядру мультиобъекта

[1] В данной работе под процедурной моделью объекта-образца понимается совокупность его параметрического описания и алгоритма пересечения с лучом.

на конвейере трассировки лучей GPU (далее ***RT-конвейере***), в результате чего формируется изображение множества экземпляров объекта-образца.

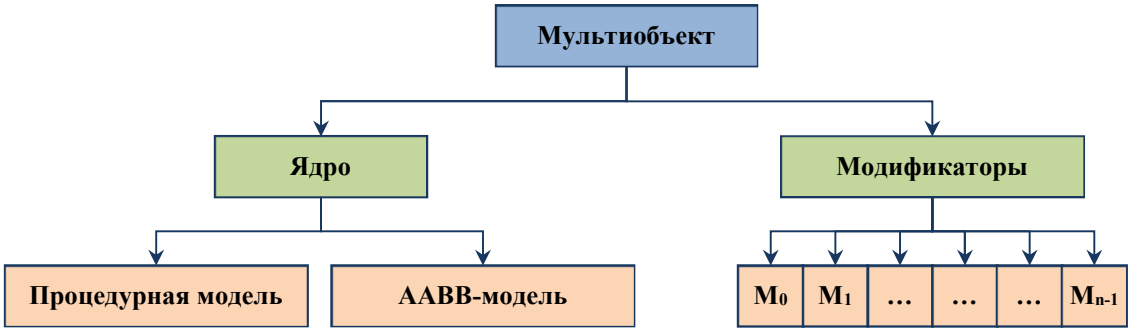

Рисунок 1. Общая структура трассируемого мультиобъекта

Одним из перспективных направлений является исследование и разработка трассируемых мультиобъектов с ядрами, основанными на *облаках точек*. Главным преимуществом таких мультиобъектов является возможность в компактной и универсальной форме описывать целые классы разнообразных объектов окружающей среды, имеющих сложную и уникальную геометрию (например, элементы растительности).

В референсной работе [7] продемонстрирован принцип построения ядра мультиобъекта на примере облака точек, сгенерированного случайным образом. В этой работе процедурная модель облака точек рассматривается как совокупность *примитивных* процедурных моделей его точек (используются сферы и кубы), а AABB-модель – как совокупность AABBs этих примитивных моделей. Такой подход относительно просто реализуется, однако характеризуется тем, что увеличение размера облака точек приводит к аналогичному росту числа AABBs. Ввиду того что RT-конвейер может одновременно обрабатывать лишь ограниченное число AABBs (обработка выполняется специальным аппаратным блоком), большой «наплыв» AABBs создает очередь и приводит к образованию «бутылочного горлышка», тормозящего работу всего RT-конвейера. Кроме того, такой большой рост числа AABBs приводит к нерациональному расходу видеопамяти, что также является серьезным ограничением.

Описанная проблема исследована в нашей первой работе [8] в контексте задачи визуализации детализированных облаков точек, полученных в результате 3D-сканирования реальных объектов (десятки миллионов точек). В нашем исследовании предложено рассматривать процедурную модель облака точек как совокупность процедурных моделей *групп точек*, соответствующих листьям разреженного воксельного октодерева. Было установлено, что AABB-модель, построенная по таким группам точек, хорошо вписывается в иерархию ограничивающих объемов (BVH) [9], на основе которой работает RT-конвейер, а также содержит существенно меньше AABBs, чем размер облака точек[2]. По сравнению с референсным подходом [7] разработанное «групповое» ядро обеспечивает прирост производительности до 47 % и сокращение накладных расходов видеопамяти в 1,9-5,4 раз.

Полученные результаты легли в основу *расширенной концепции* трассируемого мультиобъекта (РК), разработанной в нашем втором исследовании [10]. В рамках данной концепции предложено оснастить «групповое» ядро *дополненными модификаторами*, которые в отличие от базовых модификаторов [11] изменяют не только положения, ориентации и масштабы экземпляров объекта-образца, но и их цвета (оттенки). В работе предложен метод задания таких модификаторов, основанный на псевдослучайной выборке из нормального распределения, который обеспечивает эффект уникальности генерируемых экземпляров объекта-образца. В ходе исследования была показана высокая эффективность РК на примере задачи визуализации в реальном времени обширных виртуальных лесных массивов, состоящих из миллионов деревьев (одно дерево содержит порядка миллиона точек).

Настоящая работа посвящена развитию РК в части поддержки работы с *двусторонними объектами-образцами*, как, например, трава или листья. Характерной чертой таких объектов является наличие

[2] На практике один AABB охватывает до 4-16 точек.

тонких изогнутых поверхностей, которые проблематично описать с помощью процедурных сфер (для получения визуально-непрерывной поверхности требуется огромное количество сфер малого радиуса). Эффективным путем решения данной проблемы является переход к представлению объекта-образца в виде облака *3D-гауссианов* [12]. По сути, 3D-гауссиан представляет собой трехосный эллипсоид, параметры которого подобраны таким образом, чтобы он аппроксимировал определенный участок поверхности объекта. Благодаря тому, что 3D-гауссианы покрывают анизотропные области, с их помощью можно моделировать объекты-образцы любой формы (в том числе, тонкие и изогнутые), затрачивая при этом существенно меньшее количество примитивов по сравнению с облаком сфер. В рамках описанного развития РК в данной работе предлагается метод объединения 3D-гауссианов и аппаратно-ускоренной трассировки лучей, основанный на модификации оригинального «группового» ядра трассируемого мультиобъекта.

Исходные данные

В нашей работе исходными данными для построения трассируемых мультиобъектов являются облака 3D-гауссианов, заданные с помощью *базовой* параметрической модели. Данная модель описывает следующие геометрические и визуальные свойства 3D-гауссиана: форму, положение, ориентацию, диффузный цвет и прозрачность.

Геометрические свойства задаются с использованием локальных систем координат 3D-гауссиана и объекта-образца (далее системы GCS и OCS, см. рис. 2*а*). В частности, форма 3D-гауссиана определяется тройкой коэффициентов $\{a, b, c\}$ сжатия/растяжения единичной сферы вдоль осей системы GCS, а положение и ориентация задаются относительно системы OCS с помощью координат P_c центра эллипсоида и единичного кватерниона q.

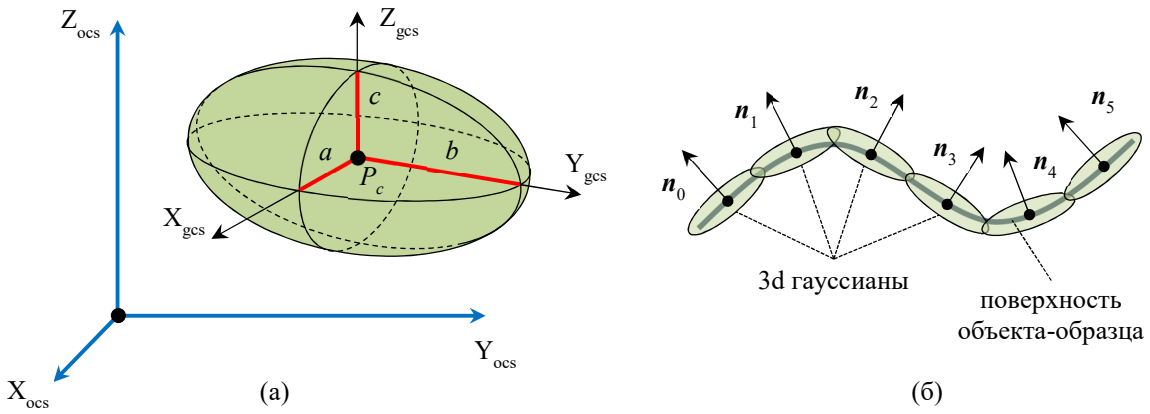

Рисунок 2. Параметры 3D-гауссиана: *а*) геометрические, *б*) для расчета освещенности

Визуальные свойства 3D-гауссиана описываются согласно теории переноса излучения [13]. В рамках этой теории цвет 3D-гауссиана задается не в явном виде, а с помощью коэффициентов сферических гармонических функций (гармоник) 0-го, 1-го и более высоких порядков. Такой подход позволяет хранить в компактном виде видозависимые визуальные свойства 3D-гауссиана (блики, отражения, анизотропные материалы), а также обеспечивает непрерывность их изменения в зависимости от угла обзора. В базовую параметрическую модель входит тройка коэффициентов $\{k_{sh0}, k_{sh1}, k_{sh2}\}$ сферической гармоники 0-го порядка, определяющая диффузный цвет 3D-гауссиана. Данные коэффициенты связаны с rgb-цветом следующим соотношением:

$$(r,g,b) = (k_{sh0}, k_{sh1}, k_{sh2})\, Y_0^0 + (0.5, 0.5, 0.5), \qquad (1)$$

где $Y_0^0 = 1/\sqrt{4\pi} \approx 0.282095$ – сферическая гармоника 0-го порядка [14]. Прозрачность 3D-гауссиана задается с помощью привычного коэффициента $\alpha \in [0,1]$ [не]прозрачности. Кроме перечисленных выше параметров, в базовую модель 3D-гауссиана также входит нормаль n к поверхности аппроксимируемого объекта-образца в центре P_c эллипсоида (см. рис. 2*б*). Данный параметр необходим для корректного расчета освещенности 3D-гауссиана.

В отличие от расширенных моделей [12, 15], включающих в себя коэффициенты гармоник более высокого порядка, базовая модель 3D-гауссиана занимает существенно меньше памяти, что особенно важно в контексте мультиобъектной визуализации. Также базовая модель поддерживается современными 3D-редакторами [16, 17], что позволяет оценивать корректность получаемых результатов и проводить их сравнительный анализ.

Постановка задачи

В рамках данного исследования мы будем работать с облаками 3D-гауссианов, полученными из непрозрачных высокополигональных моделей (далее будем называть такие облака *восстановленными*). По сравнению с каноническим итеративным методом извлечения облака 3D-гауссианов из серии снимков [12], процесс восстановления облака 3D-гауссианов по полигональной модели реализуется значительно проще и быстрее, ввиду наличия априорной информации о геометрии объекта. Для подготовки восстановленных облаков мы используем продвинутый конвертер [18], который строит 3D-гауссиановую аппроксимацию в масштабе реального времени (< 1 мс) на основе шейдерной обработки меша на графическом конвейере GPU. Такой подход позволяет быстро создавать детализированные облака 3D-гауссианов (миллионы элементов) для широкого круга объектов-образцов, а также сравнивать качество получаемой аппроксимации с эталонной высокополигональной моделью.

Пусть имеется объект-образец, заданный в виде восстановленного облака 3D-гауссианов, а также задан дополненный модификатор (матрица трансформации и цвет оттенка) некоторого экземпляра этого объекта-образца. Нашей задачей будет отображение данного экземпляра на RT-конвейере с помощью рейкастинга (частный случай обратной трассировки лучей). Чтобы сформулировать эту задачу, вначале кратко рассмотрим особенности реализации рейкастинга непрозрачных процедурных объектов на RT-конвейере.

Как известно, при рейкастинге отображение объекта реализуется на основе поиска ближайшей точки пересечения луча, испущенного из позиции наблюдателя (*первичного луча*), с примитивами, из которых состоит этот объект (рис. 3). В рассматриваемой задаче отображаемый объект состоит из процедурных примитивов (3D-гауссианов), о геометрии которых RT-конвейер ничего не знает, поэтому его работа должна быть дополнена стадией шейдера расчета пересечений (далее *I-шейдер*) [6]. Данная стадия отвечает за расчет параметра t_{hit} точки пересечения луча с процедурным примитивом и вызывается RT-конвейером автоматически, когда луч пересекает AABB, в котором находится этот процедурный примитив. После обработки I-шейдером всех AABBs, лежащих на пути луча, RT-конвейер автоматически определяет параметр t_{chit} точки ближайшего пересечения (см. рис. 3) и вызывает для нее стадию шейдера ближайшего пересечения (далее *CH-шейдер*). На данной стадии выполняется расчет цвета луча, который в дальнейшем используется для закраски соответствующего пиксела.

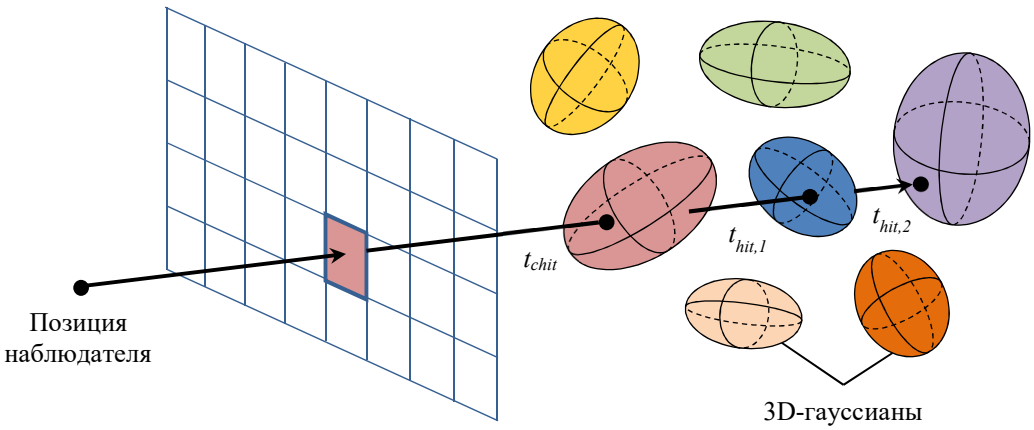

Рисунок 3. Отображение облака 3D-гауссианов с помощью рейкастинга

Исходя из описанного выше, рассматриваемую задачу можно сформулировать в виде следующих трех подзадач: а) построения AABB-модели облака 3D-гауссианов объекта-образца, б) разработки I-шейдера пересечения луча с 3D-гауссианом экземпляра объекта-образца и в) разработки CH-шейдера расчета

цвета луча. Вопросы реализации I-шейдера и CH-шейдера во многом пересекаются с нашим предыдущим исследованием [10] (отличие состоит в использовании в I-шейдере алгоритма пересечения луча с эллипсоидом [19] вместо сферы). Ввиду этого более подробно остановимся на предлагаемом методе решения первой подзадачи.

Метод построения AABB-модели облака 3D-гауссианов

Построение правильной AABB-модели процедурного объекта является важнейшим фактором, определяющим эффективность мультиобъектной визуализации и аппаратного ускорения трассировки лучей в целом. Если отображаемый объект состоит из небольшого числа процедурных примитивов, то стандартным решением будет заключение каждого такого примитива в свой AABB. Однако, если таких примитивов огромное количество (как в нашем случае) и, что особенно важно, они плотно расположены и пересекаются друг с другом (см. рис. 2*б*), то реализация стандартного подхода будет приводить к двум серьезным проблемам.

Первой проблемой является избыточное разрастание BVH-иерархии – древовидного представления виртуальной сцены, по которому RT-конвейер «понимает», где и какой экземпляр объекта-образца находится. В рамках базовой концепции трассируемого мультиобъекта [6] размер BVH-иерархии тесно связан с размером AABB-модели объекта-образца и числом его экземпляров. Соответственно, огромное количество AABBs в объекте-образце будет приводить к громоздкой BVH-иерархии и высоким накладным затратам на ее хранение и аппаратную обработку RT-конвейером (см. введение).

Второй проблемой является падение производительности RT-конвейера вследствие повторной обработки процедурных примитивов в местах пересечений их AABBs. Чем больше таких пересечений, тем выше пустая трата вычислительного ресурса RT-конвейера и тем ниже скорость визуализации. Согласно практическим рекомендациям по эффективному использованию аппаратного ускорения трассировки лучей [20], таких ситуаций следует избегать, например, путем объединения пересекающихся AABBs в более крупные.

С учетом вышеизложенного в данной работе предлагается двухэтапный метод построения AABB-модели облака 3D-гауссианов. На *первом этапе* выполняется построение опорной AABB-модели, при которой для каждого 3D-гауссиана облака создается свой AABB (фактически реализуется стандартный подход). На *втором этапе* выполняется оптимизация опорной AABB-модели, при которой реализуется укрупнение пересекающихся AABBs 3D-гауссианов. Рассмотрим эти этапы более подробно.

1. Построение опорной AABB-модели

Задача построения опорной AABB-модели, по сути, сводится к построению AABB трехосного эллипсоида, положение и ориентация которого заданы координатами P_c центра и единичным кватернионом q, а полуоси – тройкой значений $\{a, b, c\}$ (см. раздел «Исходные данные»). Ввиду того что стороны AABB параллельны осям мировой системы координат (далее *системы WCS*), для его построения достаточно найти координаты P_{min} и P_{max} его диагональных вершин (рис. 4).

Для простоты примем, что система OCS облака 3D-гауссианов совпадает с мировой системой WCS. Тогда координаты P_{min} и P_{max} можно найти через касательные плоскости к эллипсоиду, параллельные плоскостям $Y_{ocs}Z_{ocs}$, $X_{ocs}Z_{ocs}$ и $X_{ocs}Y_{ocs}$:

$$x_{\min,\max} = x_c \mp L_x,\ y_{\min,\max} = y_c \mp L_y,\ z_{\min,\max} = z_c \mp L_z, \tag{2}$$

где L_x, L_y и L_z – это расстояния от центра P_c эллипсоида до касательных плоскостей $\{x_{max}, X_{ocs}\}$, $\{y_{max}, Y_{ocs}\}$ и $\{z_{max}, Z_{ocs}\}$. Из рисунка 4 легко видеть, что если эллипсоид ориентирован вдоль осей системы OCS, то длины L_x, L_y, L_z будут равны полуосям a, b, c эллипсоида. Однако в случае произвольно ориентированного эллипсоида вычисление длин L_x, L_y, L_z становится нетривиальной задачей. В работе [21] описывается аналитический подход к решению этой задачи в общем виде, основанный на представлении ориентированного трехосного эллипсоида в виде трансформированной единичной сферы, из которого следует

$$L_x = \sqrt{M_{0.0}^2 + M_{0.1}^2 + M_{0.2}^2},\ L_y = \sqrt{M_{1.0}^2 + M_{1.1}^2 + M_{1.2}^2},\ L_z = \sqrt{M_{2.0}^2 + M_{2.1}^2 + M_{2.2}^2}, \tag{3}$$

где M – матрица произвольного аффинного преобразования единичной сферы.

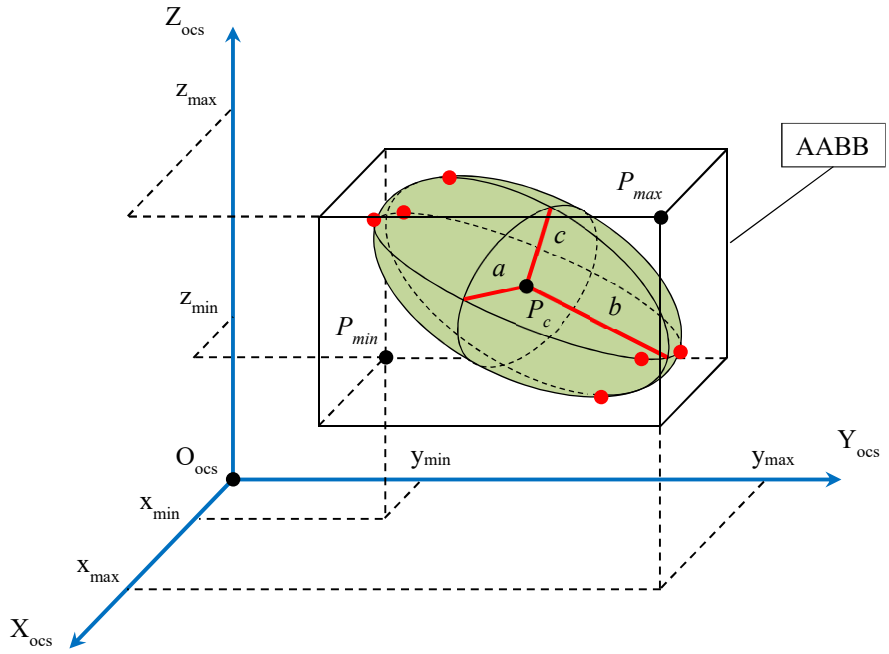

Рисунок 4. Ограничивающий параллелепипед 3D-гауссиана

Разовьем подход [21] применительно к 3D-гауссианам. Для этого вначале определим матрицу M 3D-гауссиана. Согласно теории аффинных преобразований [22] запишем для нее следующее выражение:

$$M = TR_qS, \tag{4}$$

где T – матрица переноса 3D-гауссиана (пространства) на вектор $\boldsymbol{O_{ocs}P_c}$, R_q – матрица поворота пространства, соответствующая кватерниону q, а S – матрица растяжения/сжатия пространства в a, b и c раз вдоль осей X_{gcs}, Y_{gcs}, Z_{gcs} (см. раздел «Исходные данные»). Выразим матрицу R_q через компоненты кватерниона q согласно [22] и раскроем поэлементно произведение матриц (4):

$$M = \begin{pmatrix} a(1-2(y^2+z^2)) & 2b(xy-wz) & 2c(xz+wy) & x_c \\ 2a(xy+wz) & b(1-2(x^2+z^2)) & 2c(yz-wx) & y_c \\ 2a(xz-wy) & 2b(yz+wx) & c(1-2(x^2+y^2)) & z_c \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

где x, y, z, w – компоненты кватерниона q.

Подставим элементы матрицы M в выражение (3), приведем подобные члены и упростим полученные выражения L_x, L_y и L_z, используя свойство единичного кватерниона $x^2 + y^2 + z^2 + w^2 = 1$:

$$\begin{aligned} L_x &= \sqrt{a^2 - 4\left(a^2-b^2\right)\left(x^2y^2+z^2w^2\right) - 4\left(a^2-c^2\right)\left(x^2z^2+y^2w^2\right) - 8\left(b^2-c^2\right)xyzw}, \\ L_y &= \sqrt{b^2 + 4\left(a^2-b^2\right)\left(x^2y^2+z^2w^2\right) - 4\left(b^2-c^2\right)\left(y^2z^2+x^2w^2\right) + 8\left(a^2-c^2\right)xyzw}, \\ L_z &= \sqrt{c^2 + 4\left(b^2-c^2\right)\left(y^2z^2+x^2w^2\right) + 4\left(a^2-c^2\right)\left(x^2z^2+y^2w^2\right) - 8\left(a^2-b^2\right)xyzw}. \end{aligned} \tag{6}$$

Нетрудно заметить, что полученное выражение (6) состоит из повторяющихся членов, которые можно предварительно вычислить. Введем следующие предвычисляемые компоненты: $k_1 = a^2 - b^2$, $k_2 = b^2 - c^2$, $k_3 = a^2 - c^2$, $A = 4k_1(x^2y^2 + z^2w^2)$, $B = 4k_2(y^2z^2 + x^2w^2)$, $C = 4k_3(x^2z^2 + y^2w^2)$, $D = 8xyzw$. Выполнив соответствующие замены в выражении (6) и подставив его в выражение (2), получим искомое выражение координат вершин AABB через параметры 3D-гауссиана:

$$\begin{aligned} x_{\min,\max} &= x_c \mp \sqrt{a^2 - A - C - k_2 D}, \\ y_{\min,\max} &= y_c \mp \sqrt{b^2 + A - B + k_3 D}, \\ z_{\min,\max} &= z_c \mp \sqrt{c^2 + B + C - k_1 D}. \end{aligned} \tag{7}$$

Отметим, что в случае 3D-гауссианов величины a, b и c могут принимать крайне малые значения, ввиду чего в подкоренных выражениях могут получаться значения в окрестности 0, что приводит к вырождению AABB. Во избежание таких ситуаций при вычислении выражения (7) мы используем стабилизацию вида $Q = \max(Q, \delta)$, где Q – подкоренное выражение, а $\delta = 10^{-2}$ – квадрат минимального размера AABB.

2. Оптимизация опорной AABB-модели

На этапе оптимизации AABB-модели выполняется слияние AABBs 3D-гауссианов, лежащих близко друг к другу, в более крупные AABBs. Ключевым аспектом данной задачи является разбиение исходного неупорядоченного облака на группы близлежащих 3D-гауссианов. Для этого в данной работе предлагается модифицированная версия *метода октантных групп точек* [8], созданного нами ранее для решения аналогичной задачи применительно к облакам точек (процедурных сфер).

Суть базового метода состоит в том, что облако точек разбивается на 8 октантов (кубических объемов), по которым распределяются все точки облака (по аналогии с раскладыванием шариков по корзинам). Если число точек в октанте превышает некоторый порог K_{max}, то этот октант разбивается на 8 подоктантов, между которыми снова распределяются его точки, и так повторяется до тех пор, пока не останутся октанты, содержащие от 1 до K_{max} точек. На финальной фазе для каждой такой октантной группы точек строится AABB (вычисляются диагональные вершины).

В оригинальной версии метода распределение точек по октантам осуществляется без учета размеров их процедурных примитивов (сфер), т.е. если точка попадает в октант, то считается, что попадает и ее сфера. Это допущение основано на том, что размер процедурных сфер у всех точек облака одинаков и изотропен (не зависит от направления измерения), вследствие чего *глубина* взаимного пересечения AABBs не будет превышать длину диагонали куба, описанного около процедурной сферы (рис. 5*а*). В отличие от облака сфер у 3D-гауссианов размеры, в общем случае, неодинаковы и анизотропны, ввиду чего применение оригинального метода группировки может приводить к более глубоким взаимным пересечениям AABBs (рис. 5*б*), что нежелательно.

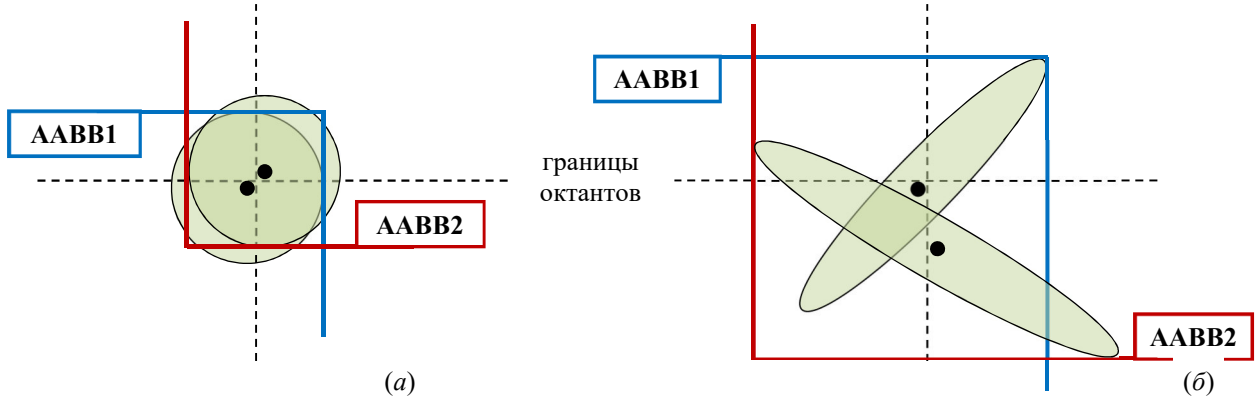

Рисунок 5. Взаимные пересечения AABBs двух групп: *а*) процедурных сфер; *б*) 3D-гауссианов

Чтобы уменьшить влияние анизотропности 3D-гауссианов на глубину взаимного пересечения AABBs, в разработанной модификации метода [8] мы исключаем из октантных групп те 3D-гауссианы, которые выходят за границы октанта. Для этого при распределении 3D-гауссианов по октантам мы проверяем не их центры (как в оригинальном методе), а диагональные вершины их индивидуальных AABB, вычисленные по формулам (7) на этапе построения опорной AABB-модели. Те 3D-гауссианы, у которых обе диагональные вершины AABB попали в октант, остаются в группе и участвуют в дальнейшем перераспределении по подоктантам. Если же хотя бы одна диагональная вершина AABB не попадает в октант, то для такого 3D-гауссиана процесс группировки считается завершенным и формируется единичная группа (состоящая из одного элемента). После формирования всех групп 3D-гауссианов (включая единичные) выполняется финальная *фаза слияния*, при которой индивидуальные AABBs 3D-гауссианов, принадлежащих одной группе, сливаются в общий AABB группы (AABB1 и AABB2 на рис. 6). Отметим, что в случае единичной группы результатом фазы слияния будет индивидуальный AABB (AABB3 на рис. 6). Полученная совокупность групповых и индивидуальных

AABBs образует в результате AABB-модель облака 3D-гауссианов, сбалансированную по количеству / глубине взаимного пересечения AABBs.

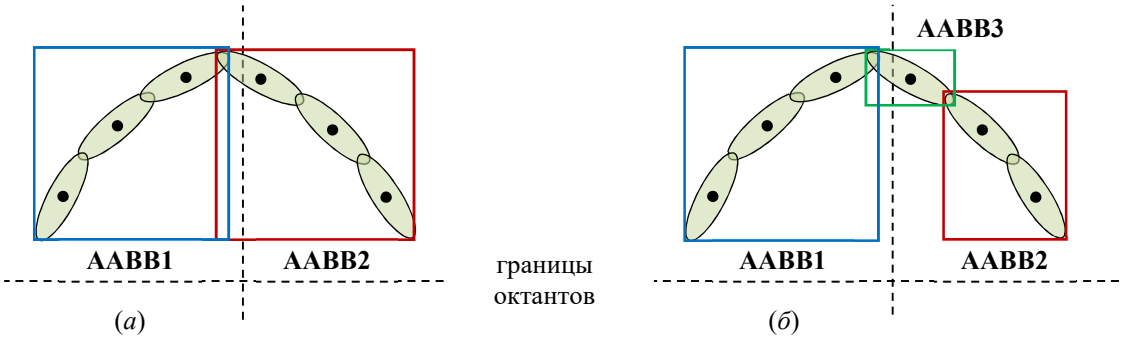

Рисунок 6. Слияние AABBs 3D-гауссианов: *а*) базовый метод; *б*) модифицированный

Результаты экспериментов

Экспериментальная часть данного исследования проводилась на базе созданного нами ранее программного комплекса MultiView мультиобъектной визуализации виртуальной растительности [10]. В состав комплекса входит CPU-модуль, отвечающий за построение ядра мультиобъекта и расчет дополненных модификаторов (см. раздел «Исходные данные»), а также GPU-модуль, реализующий отображение в реальном времени мультиобъекта во множество экземпляров. Программный комплекс написан на языках C++ и GLSL 4.6 с применением библиотеки STL, графических API OpenGL 4.6 и Vulkan 1.3.290 (для программирования аппаратно-ускоренной трассировки лучей).

На основе предложенных методов и подходов в комплексе MultiView была реализована поддержка трассируемых мультиобъектов нового типа – «облако 3D-гауссианов». Данная возможность была экспериментально исследована в ходе апробации комплекса MultiView на персональном компьютере (Intel Core i7-6800K, 16 GB RAM, NVidia RTX 2080, Full HD разрешение). Апробация предложенного решения выполнялась в три этапа.

На ***первом этапе*** был проведен сравнительный анализ качества визуализации экземпляра мультиобъекта. Для этого была предварительно разработана методика подготовки облака 3D-гауссианов из высокополигональной («high-poly») модели, выступающей в роли эталона. Данная методика включает в себя унификацию локальной системы координат «high-poly» модели с помощью открытого комплекса CloudCompare [23], конвертацию в бинарный GLB-формат с помощью свободного пакета Blender [24] и преобразование в облако 3D-гауссианов с помощью открытого комплекса Mesh2Splat [18]. В ходе сравнительного анализа была выполнена визуализация (рис. 7): а) эталонной «high-poly» модели с помощью комплекса CloudCompare; б) производного облака 3D-гауссианов с помощью интерактивного редактора SuperSplat [17]; в) производного облака 3D-гауссианов с помощью нашего решения; г) производного облака точек (сфер) с помощью решения [10].

На ***втором этапе*** с помощью комплекса MultiView было исследовано изменение среднего времени синтеза кадра (на примере образца травы из рис. 7) в зависимости от порогового значения K_{max} на высотах h наблюдения 0.1, 0.4, 0.8, 1.6 (рис. 8). Высота h задана в условных единицах (0.1 соответствует высоте образца травы), среднее время измеряется в миллисекундах, а K_{max} обозначает максимальное число 3D-гауссианов в одном AABB.

На ***третьем этапе*** с помощью комплекса MultiView была выполнена мультиобъектная визуализация травяного покрова, состоящего из 2000x2000 экземпляров образца травы. Визуализация выполнялась при K_{max} = 4, базовом цвете оттенка экземпляров (0.5, 0.7. 0.05) и дисперсии r, g, b-компонент базового цвета (0.0, 0.02, 0.01). На рисунке 9 показаны примеры кадров полученной визуализации. В рамках третьего этапа также было измерено среднее времени $t_{ср}$ синтеза изображений (в мс) при изменении высоты наблюдения от 0.1 до 12.8 (таблица «Среднее время синтеза кадра (массив экземпляров)»).

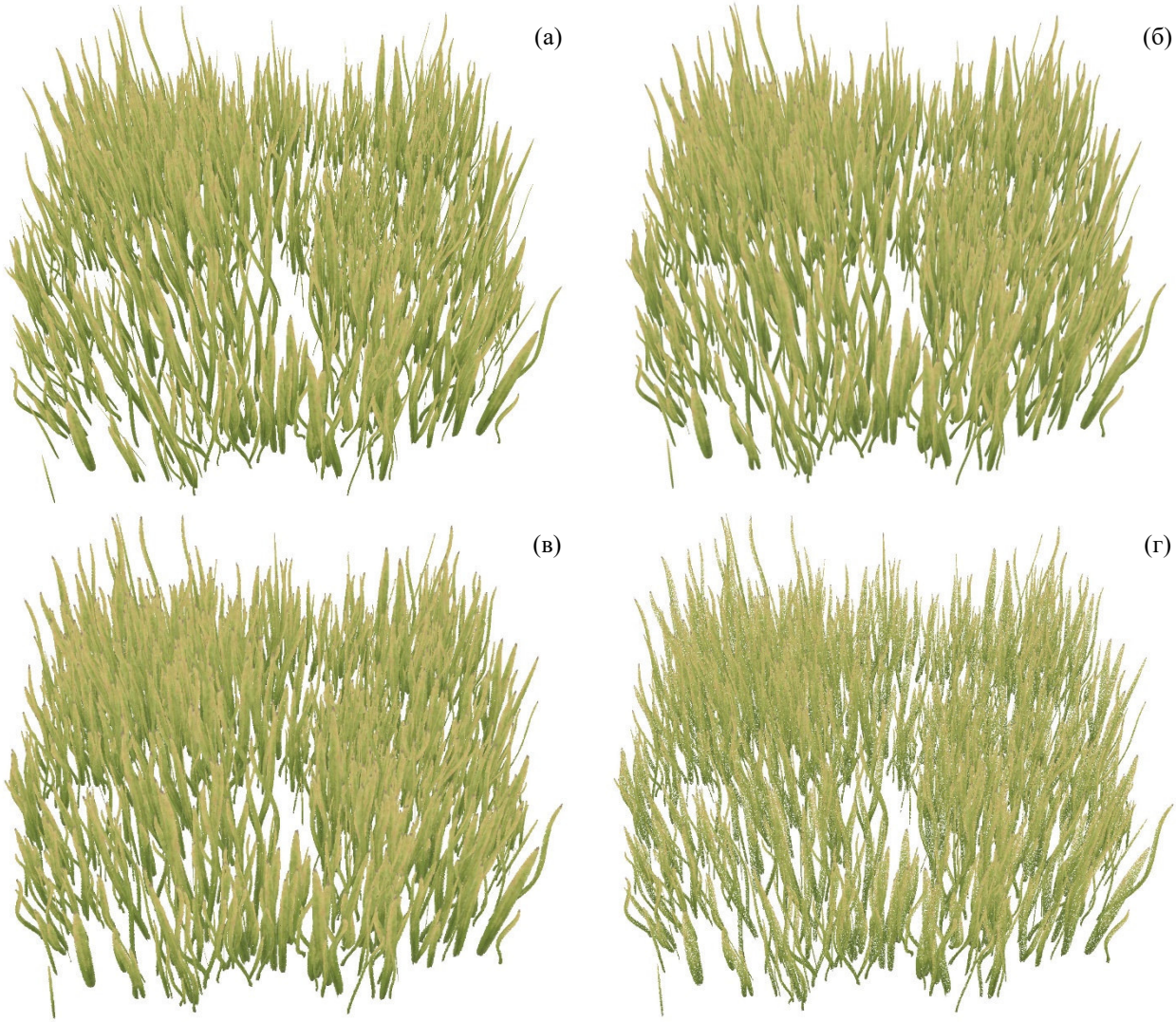

Рисунок 7. Сравнение качества визуализации виртуального образца травы: *а*) эталонная «high-poly» модель, 1085280 полигонов (CloudCompare); *б*) производное облако 3D-гауссианов, 3436498 точек (SuperSplat); *в*) производное облако 3D-гауссианов (наше решение); *г*) производное облако сфер, 3436498 точек (решение [10])

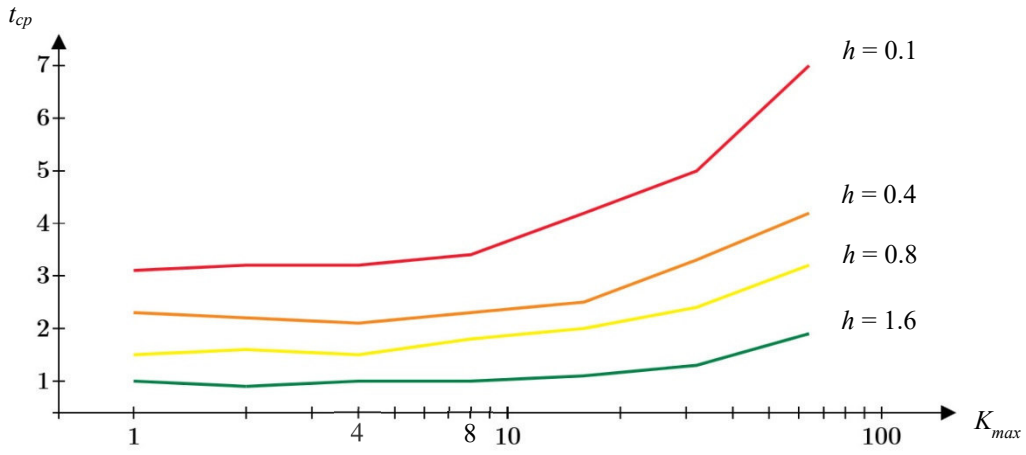

Рисунок 8. Влияние параметра Kmax на среднее время tcp синтеза кадра на различных высотах h наблюдателя (оценка выполнена для экземпляра образца травы на рис. 7)

Рисунок 9. Примеры кадров мультиобъектной визуализации травяного покрова, состоящего из 4 млн экземпляров, с помощью разработанного решения, основанного на трассируемых облаках 3D-гауссианов

Среднее время синтеза кадра (массив экземпляров), мс

h	0.1	0.2	0.4	0.8	1.6	3.2	6.4	12.8
t_{cp}	4.6	6.5	8.6	11.0	11.9	12.5	13.1	14.3

Обсуждение, выводы и заключение

Из рисунка 7 видно, что редактор SuperSplat и наше решение по качеству визуализации сопоставимы с эталонной моделью, а представление в виде облака сфер, напротив, уступает всем трем решениям ввиду отсутствия информации между сферами. Из этого также следует обратный вывод, что переход на 3D-гауссианы позволяет сократить число элементов в облаке без ущерба качеству визуализации.

Из рисунка 8 видно, что для тестируемого виртуального образца травы баланс между скоростью визуализации и расходом видеопамяти (числом AABBs) обеспечивается при значении K_{max} в диапазоне 4-8. Также в ходе экспериментальной оценки мы попытались на примере этого же виртуального образца сравнить производительность нашего визуализатора (аппаратно-ускоренная трассировка лучей) и визуализатора, входящего в комплекс Mesh2Splat (шейдерная растеризация). Однако в ходе этого испытания было обнаружено, что как только в кадр попадают все 3D-гауссианы, производительность второго визуализатора резко падает до значений 84-93 мс на кадр, независимо от высоты наблюдения. Из этого можно сделать вывод, что для задачи отображения облаков 3D-гауссианов в реальном времени целесообразно использовать аппаратно-ускоренную трассировку лучей, которая более избирательно тратит вычислительные ресурсы GPU и на порядок выигрывает в производительности по сравнению с графическим конвейером.

Из рисунка 9 и данных таблицы следует, что даже при достаточно плотной расстановке экземпляров травы (не заметны места стыковки экземпляров) наше решение обеспечивает визуализацию в масштабе реального времени (не более 40 мс на кадр). Тем не менее полученные значения t_{cp} превышают на 30-40 % значения, измеренные нами в работе [10] при визуализации массивов облаков точек схожих размеров, что можно объяснить дополнительными взаимными пересечениями AABBs из-за анизотропности 3D-гауссианов.

В заключение отметим, что переход на 3D-гауссианы позволил существенно повысить общий уровень качества визуализации, по сравнению с нашим предыдущим решением [10], основанным на облаках точек, а также сделал возможным визуализацию двусторонних объектов-образцов с тонкой, изогнутой геометрией. В качестве будущей работы планируется распространить полученные результаты на построение виртуальных экосистем, состоящих из образцов растительности различных типов.

Финансирование

Публикация выполнена в рамках государственного задания НИЦ «Курчатовский институт» – НИИСИ по теме № FNEF-2024-0002 «Математическое моделирование многомасштабных динамических процессов и системы виртуального окружения».

Список литературы

1. UNIGINE: real-time 3D engine. 2005–2025. URL: https://unigine.com/.
2. Visualization of Hybrid Virtual Scenes Using Hardware-Accelerated Ray Tracing and Rasterization / Timokhin P.Yu., Mikhaylyuk M.V. // Scientific Visualization. 2024. Vol. 16, no. 3. P. 60–70. https://doi.org/10.26583/sv.16.3.06.
3. Immersive landscapes: modelling ecosystem reference conditions in virtual reality / Chandler T., Richards A.E., Jenny B., et al. // Landscape Ecology. 2022. Vol. 37. P. 1293–1309. https://doi.org/10.1007/s10980-021-01313-8.
4. Санжаров В.В., Фролов В.А., Галактионов В.А. Исследование технологии Nvidia RTX // Программирование. 2020. № 4. С. 65-72. https://doi.org/10.31857/S0132347420030061.
5. NVidia RTX Blackwell GPU Architecture // NVIDIA Corporation. 2025. URL: https://images.nvidia.com/aem-dam/Solutions/geforce/blackwell/nvidia-rtx-blackwell-gpu-architecture.pdf.
6. Rusch M., Bickford N., Subtil N. Introduction to Vulkan Ray Tracing // Ray Tracing Gems II. NVIDIA. 2021. P. 213–255. https://doi.org/10.1007/978-1-4842-7185-8_16.
7. Lefrançois M.-K. Intersection Shader // NVIDIA Vulkan Ray Tracing Tutorials. 2020–2023. URL: https://github.com/nvpro-samples/vk_raytracing_tutorial_KHR/tree/master/ray_tracing_intersection.
8. Тимохин П.Ю., Михайлюк М.В. Метод упорядочивания облаков точек для визуализации на конвейере трассировки лучей // Программирование. 2024. № 3. С. 42-53. https://doi.org/10.31857/S0132347424030054.
9. Смирнов Л.М., Фролов В.А., Волобой А.Г. Анализ производительности методов обхода двухуровневых BVH-деревьев в трассировке лучей на графических процессорах // Материалы 34-й Международной конференции по компьютерной графике и машинному зрению (Графикон 2024). 2024. С. 147–163. https://doi.org/10.25206/978-5-8149-3873-2-2024-147-163.
10. Timokhin P. Y., Mikhaylyuk M. V. Multiobject Visualization of Vast Forests in Virtual Environment Systems // Programming and Computer Software. 2025. Vol. 51, no. 3. P. 198–206. https://doi.org/10.1134/S0361768825700082.
11. Lefrançois M.-K. Ray tracing instances // NVIDIA DesignWorks. Vulkan Ray Tracing Tutorials. 2020-2024. URL: https://github.com/nvpro-samples/vk_raytracing_tutorial_KHR/tree/master/ray_tracing_instances.
12. 3D Gaussian Splatting for Real-Time Radiance Field Rendering / Kerbl B., Kopanas G., Leimkuehler T., Drettakis G. // ACM Transactions on Graphics (TOG). 2023. Vol. 42, no. 4. Article no. 139. P. 1–14. https://doi.org/10.1145/3592433.

13. Sloan P.J., Kautz J., Snyder J.M. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments // ACM Transactions on Graphics (TOG). 2002. Vol. 21, iss. 3. P. 527–536. https://doi.org/10.1145/566654.566612.

14. Ramamoorthi R., Hanrahan P. An efficient representation for irradiance environment maps // SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001. P. 497–500. https://doi.org/10.1145/383259.383317.

15. Spec-Gaussian: Anisotropic View-Dependent Appearance for 3D Gaussian Splatting / Yang Z., Gao X., Sun Y., et al. // ArXiv, abs/2402.15870. 2024. https://doi.org/10.48550/arXiv.2402.15870.

16. KIRI Engine – 3DGS Render Blender Addon. 2024-2025. URL: https://www.kiriengine.app/blender-addon/3dgs-render.

17. SuperSplat – 3D Gaussian Splat Editor. 2023-2025. URL: https://superspl.at/editor.

18. Scolari S. Mesh2Splat: Fast mesh to 3D Gaussian splat conversion. 2025. URL: https://github.com/electronicarts/mesh2splat

19. Zhou H. Rendering a Perfect Sphere in OpenGL, then an Ellipsoid. 2023. URL: https://blog.42yeah.is/rendering/opengl/2023/12/18/perfect-ellipsoid.html.

20. Sjoholm J. Best Practices for Using NVIDIA RTX Ray Tracing (Updated) // NVIDIA Technical Blog. Jul 25, 2022. URL: https://developer.nvidia.com/blog/best-practices-forusing-nvidia-rtx-ray-tracing-updated/.

21. Barnes T. Exact Bounding Boxes for Spheres/Ellipsoids. 2014. URL: https://tavian.dev/2014/ellipsoid_bounding_boxes.html.

22. Lengyel E. Mathematics for 3D Game Programming and Computer Graphics (Third Edition). Boston, MA: Course Technology PTR, 2012. 624 p.

23. CloudCompare. 3D point cloud and mesh processing software. Open Source Project, 2004-2024. URL: http://www.cloudcompare.org/.

24. Blender – The Free and Open Source 3D Creation Suite. 1998-2025. URL: https://www.blender.org/.