

Алгоритм адаптивного разбиения облака точек с использованием иерархии ограничивающих объемов

Е. С. Глумова

Нижегородский государственный технический университет им. Р. Е. Алексеева,
Нижний Новгород, Россия

Аннотация. В исследовании представлены результаты разработки алгоритма разбиения пространства на основе структуры иерархии ограничивающих объемов BVH (Bounding Volume Hierarchy) для задачи трансляции пространственных данных. Цель исследования – разработать алгоритм разбиения облака точек, адаптирующийся к локальной плотности распределения точек. В данной работе предлагается следующий подход разбиения облаков точек на основе BVH-структуры с адаптивным критерием стоимости разбиения point-SAH с поправками на контраст плотностей и высот, статистической проверкой на наличие явной кластерной структуры данных на основе алгоритма g-means, динамическими порогами на квантованной решетке. Комбинирование этих механизмов позволяет ускорить процессы импорта облаков точек за счет более качественного распределения точек по блокам, минимизировать пустые блоки и облегчить обработку на аппаратных ресурсах, не обладающих высокой вычислительными мощностями. Представлено подробное описание алгоритма, его математической модели, а также эксперименты, демонстрирующие влияние параметров на качество разбиения. Обсуждаются ограничения текущего исследования и направления дальнейшей работы.

Ключевые слова: облако точек, трехмерное моделирование, иерархические структуры, BVH, SAH, g-means, CloudCompare.

Algorithm Adaptive Point Cloud Partitioning based on Bounding Volume Hierarchy

E. S. Glumova

Nizhny Novgorod State Technical University named after R.E. Alekseev, Nizhny Novgorod, Russia

Abstract. The research presents a novel space partitioning algorithm based on Bounding Volume Hierarchies (BVH) designed for the efficient translation of spatial data. The research objective is to develop a point cloud partitioning algorithm that adapts to local variations in point density. To this end, this paper proposes a BVH construction approach featuring several key components: an adaptive point-SAH splitting criterion with cost adjustments for density and height contrast; a statistical test based on the g-means algorithm to detect distinct cluster structures; and dynamic termination thresholds applied on a quantized grid. The combination of these mechanisms accelerates point cloud import and processing by ensuring a more balanced distribution of points across blocks, minimizing empty partitions, and reducing the computational load on low-power hardware. The paper provides a detailed description of the algorithm, its formal mathematical model, and a series of experiments demonstrating the impact of its parameters on the partitioning quality. Finally, the limitations of the current research and potential avenues for future work are discussed.

Keywords: point cloud, 3D modelling, Hierarchical Structures, BVH, SAH, g-means, CloudCompare.

Введение

Трехмерное моделирование играет все более важную роль в управлении жизненным циклом промышленных и гражданских объектов. В основе этого процесса лежат технологии наземного лазерного сканирования (НЛС), обеспечивающие получение высокоточных цифровых моделей. Одной из наиболее перспективных методологий является Scan-to-BIM, позволяющая интегрировать данные НЛС в цифровые информационные модели (ЦИМ) для решения задач на всех этапах жизненного цикла объекта – от проектирования до эксплуатации и демонтажа [1].

Результатом НЛС является облако точек – неструктурированный набор трехмерных координат. Работа с ним сопряжена с проблемой большого объема данных, который накладывает серьезные ограничения на методы их хранения, обработки и трансляции между программными средами. Для решения этой проблемы традиционно применяются иерархические методы пространственного разбиения. Принято различать структуры, управляемые пространством, и структуры, управляемые данными. Несмотря на преимущества структур, управляемых данными, стандартные алгоритмы их построения не в полной мере адаптируются к локальным статистическим особенностям распределения точек. Гипотеза данной работы заключается в том, что интеграция статистического анализа плотности непосредственно в критерии построения иерархии ограничивающих объемов BVH позволит

значительно повысить качество разбиения, формируя блоки, которые более точно соответствуют семантическим кластерам в облаке точек.

Целью данного исследования является разработка алгоритма разбиения облака точек на основе BVH, адаптирующегося к локальной плотности. Для проверки гипотезы предложен алгоритм построения BVH-дерева, включающий следующие ключевые элементы:

- адаптивный критерий стоимости разбиения point- SAH с поправками на контраст плотностей и высот;
- статистическая проверка на наличие явной кластерной структуры данных на основе алгоритма g-means;
- динамический порог для листовых узлов, зависящий от их физического объема, что предотвращает избыточное дробление в разреженных областях;
- механизм устойчивого разделения на основе квантованной решетки и строгих полуинтервалов для исключения потерь или дублирования точек.

Предложенная комбинация эвристик и механизмов направлена на создание разбиения, которое не только ускоряет процессы импорта и обработки данных за счет сбалансированного распределения нагрузки, но и формирует блоки, более соответствующие естественной структуре сканируемого объекта.

Обзор существующих подходов

Иерархические структуры являются фундаментальным инструментом для организации пространственных данных. Классические подходы, такие как k-d деревья и октодеревья, имеют известные ограничения при работе с неоднородными облаками точек. K-d дерево может становиться чрезмерно глубоким и создавать геометрически несбалансированные ячейки, тогда как октодерево страдает от генерации множества пустых узлов в разреженных областях и недостаточной детализации в плотных. Для решения этих проблем были предложены гибридные структуры, например, сочетающие октодерево на верхних уровнях с локальными R*-деревьями в листьях [2]. Однако построение таких сложных индексов для сверхбольших облаков точек сопряжено со значительными вычислительными затратами.

С ростом объемов данных ключевыми направлениями исследований стали обработка данных, превышающих объем оперативной памяти, и параллелизм. Значимой работой является метод, который позволяет строить октодерево для миллиардов точек за линейное время путем потоковой обработки данных с диска [3]. Этот подход, реализованный в системе Potree [4], стал стандартом для веб-визуализации. Для ускорения построения на параллельных архитектурах, таких как GPU, широко используются методы, основанные на кодах Мортон [5], которые сводят задачу построения иерархии к высокоэффективной сортировке. Однако качество таких деревьев, как правило, уступает структурам, построенным с использованием более сложных эвристик, поскольку коды Мортон эквивалентны равномерному разбиению пространства и не учитывают реальную форму облака точек.

Наиболее гибкой структурой, управляемой данными, является иерархия ограничивающих объемов BVH. В отличие от октодеревьев, BVH группирует геометрию, а не делит пространство, что делает структуру изначально адаптивной. Для построения качественных BVH широко применяется эвристика площади поверхности SAH [6], которая оценивает «стоимость» разделения узла на основе площадей ограничивающих объемов и количества примитивов в них. Критерий SAH нацелен на минимизацию вероятности пересечения условного запроса (например, луча при трассировке) с дочерними узлами и стал стандартом для построения высококачественных деревьев. Однако классический SAH предполагает равномерное распределение запросов по объему узла, что не вполне релевантно для задачи разбиения облака точек, где целью является семантическая и плотностная однородность блоков.

Развитие подходов к обработке «вне ядра» также повлияло на форматы хранения данных. Если ранние системы, такие как EPT (Entwine Point Tiles) [7], генерировали огромное число мелких файлов, то современные форматы, такие как COPC (Cloud-Optimized Point Cloud) [8], интегрируют иерархический индекс непосредственно в единый laz-файл. Это подтверждает тенденцию на использование встроенных иерархических структур для обеспечения быстрого потокового доступа к данным. Такие библиотеки, как PDAL [9] и LAStools [10], предоставляют конвейеры для обработки и преобразования данных в эти форматы. Тем не менее, они в первую очередь ориентированы на хранение и трансляцию,

а не на семантически осмысленное разбиение сцены. Таким образом, остается актуальной задача совершенствования самих критериев построения иерархий, чтобы они лучше соответствовали внутреннему устройству и плотностным характеристикам облаков точек.

Теория

Полный цикл работы алгоритма разбиения облака точек на основе BVH-структуры, адаптирующегося к локальной плотности, отражает рисунок 1.

Процесс состоит из трех основных этапов:

- квантизация пространства;
- рекурсивное деление пространства на основе статистической или плотностной проверок;
- постобработка и слияние узлов.

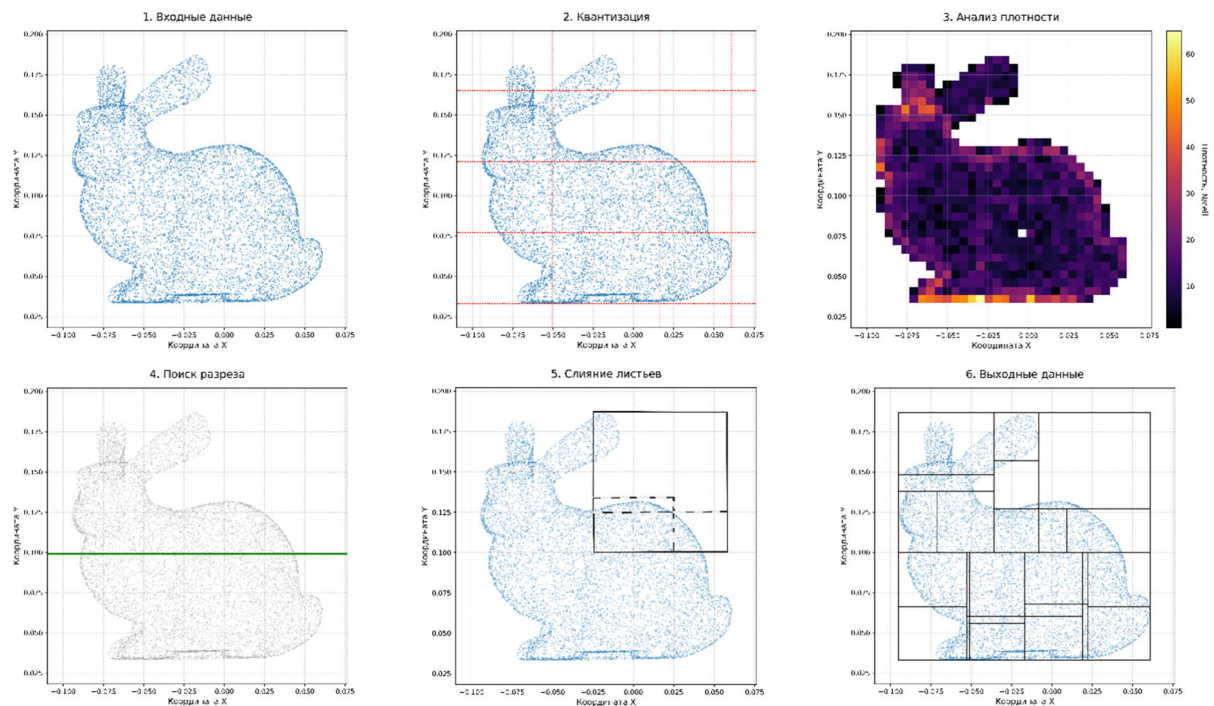


Рисунок 1. Полный цикл работы алгоритма разбиения облака точек на основе BVH-структуры (2D-проекция)

Исходные координаты точек в метрической системе преобразуются в дискретную целочисленную сетку (рис. 1, 1, 2), затем происходит анализ их плотности (рис. 1, 3). Выполняется рекурсивное деление пространства «сверху-вниз». Для каждого узла на основе геометрических и статистических проверок ищется оптимальная плоскость разреза (рис. 1, 4). После завершения основного этапа построения выполняется проход по дереву «снизу-вверх», в ходе которого избыточно мелкие соседние листья могут быть объединены (рис. 1, 5) для получения итогового сбалансированного набора блоков (рис. 1, 6).

Формальная постановка задачи выглядит следующим образом. Дано конечное множество точек $X = \{x_1, x_2, \dots, x_n\}$, являющееся подмножеством пространства R^3 . Требуется найти иерархическую структуру T , которая определяет партицию множества X на m непересекающихся подмножеств L_1, L_2, \dots, L_m . Эти подмножества L_i соответствуют листовым узлам дерева T и удовлетворяют следующим свойствам разбиения:

- объединение всех подмножеств равно исходному множеству: $\bigcup_{i=1}^m L_i = X$;
- подмножества попарно не пересекаются: $L_i \cap L_j = \emptyset$ при $i \neq j$;
- каждое подмножество непустое: $L_i \neq \emptyset$.

Искомое разбиение $P^* = \{L_1, \dots, L_m\}$ и соответствующая ему иерархия T^* должны быть оптимальными с точки зрения комплексной целевой функции $C(T)$, которая минимизирует геометрические и максимизирует статистические свойства однородности элементов разбиения. Задача

нахождения оптимальной иерархии T^* решается путем локальной оптимизации на каждом шаге рекурсивного разбиения с использованием жадного подхода, логика которого представлена на рисунке 2.

Для обеспечения вычислительной устойчивости все координаты исходного облака точек преобразуются в целочисленную сетку. Процесс построения дерева T эквивалентен рекурсивному разбиению множеств. Корневому узлу дерева соответствует исходное множество X . Каждый внутренний узел соответствует некоторому подмножеству $S \subseteq X$.

Классическая эвристика SAH для трассировки лучей оценивает стоимость разделения узла. Данный критерий модифицируется для задачи разбиения облака точек. Во-первых, опускается константа стоимости обхода узла (C_T), так как целью является не минимизация глубины дерева, а качество итоговых листовых узлов.

Во-вторых, вводятся дополнительные члены за разделение по границам с высоким контрастом плотности и высот.

В результате модифицированная стоимость point-SAH (C'_{split}) для разреза по оси j и порогу t определяется следующим образом (1):

$$C'_{split}(S, j, t) = (SA(AABB(S_L)) \cdot |S_L| + SA(AABB(S_R)) \cdot |S_R|) - \lambda_D \cdot \Delta_{dens} - \lambda_H \cdot \Delta_{height} \quad (1)$$

где $S_L = \{x \in S \mid x_j \leq t\}$ и $S_R = \{x \in S \mid x_j > t\}$ – дочерние множества точек;

$|S_L|, |S_R|$ – количество точек (мощность множеств) в левом и правом подузлах;

$SA(AABB(S_L)), SA(AABB(S_R))$ – площади поверхностей $AABB$ дочерних узлов;

λ_D и λ_H – это неотрицательные весовые коэффициенты, управляющие влиянием членов за плотность и высоту соответственно;

Δ_{dens} – нормированный контраст плотности между дочерними множествами S_L и S_R ;

Δ_{height} – нормированный контраст высот между дочерними множествами S_L и S_R .

Контраст плотности (Δ_{dens}) определяется как нормированная разность оценок плотности в дочерних узлах (2).

$$\Delta_{dens} = \frac{|\rho(S_L) - \rho(S_R)|}{\rho_*} \quad (2)$$

Здесь $\rho(S_i) = \frac{|S_i|}{\mu(AABB(S_i))}$ – оценка плотности точек в узле S_i , где $\mu(\cdot)$ – это пространственная мера (объем $AABB$ для 3D-анализа или площадь его проекции на плоскость XY для 2.5D-анализа);

ρ_* – это целевая или средняя плотность в родительском узле, используемая для нормировки.

Контраст по высоте (Δ_{height}) определяется как разность средних z -координат (высот) дочерних узлов, нормированная на общий диапазон высот в родительском узле (3).

$$\Delta_{height} = \frac{|Z_{avg}(S_L) - Z_{avg}(S_R)|}{Z_{range}(S)} \quad (3)$$

где $Z_{avg}(S_i) = \left(\frac{1}{|S_i|}\right) \cdot \sum_{x \in S_i} x^3$ – функция, вычисляющая среднее значение координаты Z (ось $j = 3$) для множества S_i ;

$Z_{range}(S) = \max_{x \in S}(x^3) - \min_{x \in S}(x^3)$ – полный диапазон высот в родительском множестве S , используемый для нормировки.

Минимизация модифицированной целевой функции C_{split} поощряет такие разрезы, которые не только создают геометрически компактные дочерние узлы, но и проходят по границам областей с различной плотностью или высотой. Таким образом, алгоритм стремится находить не просто геометрически, а семантически и статистически более осмысленные границы для разделения данных.

Для предотвращения статистически необоснованных («псевдо») делений вводится дополнительное ограничение. Разделение множества S по оси j рассматривается как допустимый кандидат только в том случае, если распределение точек вдоль этой оси является неоднородным.

Пусть $\Pi_j(S) = \{x_j \mid x \in S\}$ – множество проекций точек S на ось j .

Пусть \mathcal{H}_0 – нулевая гипотеза о том, что выборка $\Pi_j(S)$ порождена нормальным распределением.

Процесс принятия решения о разделении узла имеет иерархическую структуру. Приоритет отдается статистически значимому разрезу, предложенному проверкой g -means. В его отсутствие

рассматривается лучший кандидат с точки зрения минимизации C_{split} . Разрез по SAN принимается только в том случае, если он обеспечивает достаточное снижение стоимости. Если ни один из критериев не дает убедительного кандидата, узел становится листом.

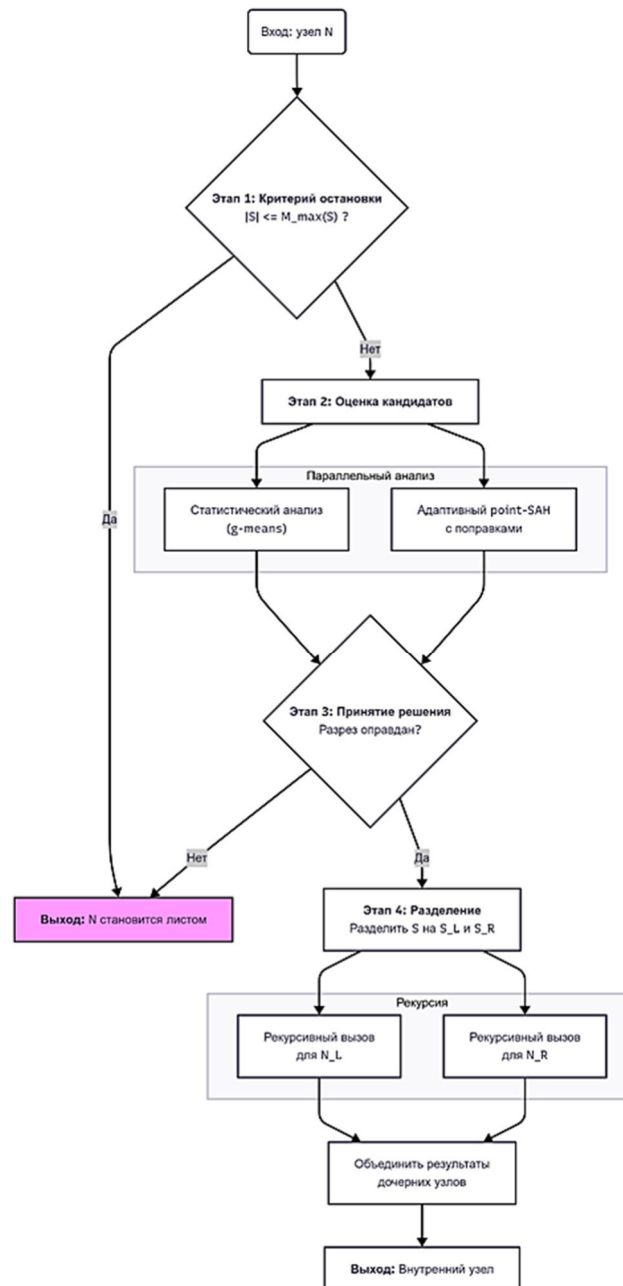


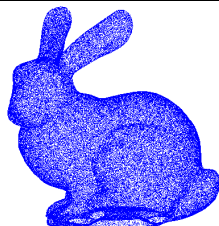
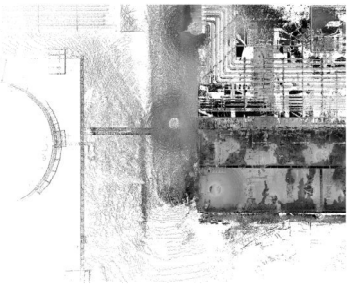
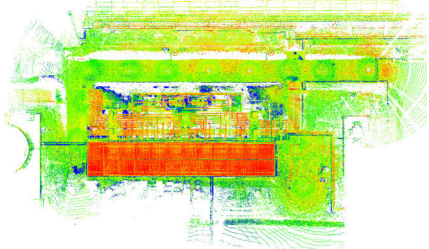
Рисунок 2. Блок-схема процесса деления для одного узла N

Жадный характер рекурсивного построения «сверху-вниз» может приводить к избыточной сегментации. Для исправления локально неоптимальных разрезов применяется этап постобработки. Алгоритм выполняет обход построенного дерева «снизу-вверх». На каждом шаге он проверяет пары соседних листовых узлов. Если суммарное количество точек в двух таких узлах не превышает динамического порога $M_{max}(S)$ для их общего родительского узла S , они сливаются, а родительский узел становится листовым. Этот этап позволяет улучшить баланс итогового разбиения, укрупняя излишне детализированные блоки.

Результаты экспериментов

Для оценки работоспособности алгоритма была проведена серия экспериментов на трех наборах данных, различных по масштабу, плотности и геометрической сложности (табл. 1).

Таблица 1. Характеристика исходных данных

№	Визуализация	Название	Число точек	Особенности
1		Stanford_Bunny	999 966 точек	Характеризуется однородной плотностью и сложной, гладкой геометрией
2		6_data	15 754 340 точек	Характеризуются наличием как крупных плоских поверхностей, так и мелких конструкций (трубопроводы), а также присутствием шума и выбросов
3		Alldata	48 389 508 точек	Данные отличаются сильно выраженной неоднородностью плотности и сложной пространственной организацией

Все эксперименты проводились на персональной вычислительной системе с CPU Intel Core i7-14700K, 64 ГБ ОЗУ, NVIDIA GeForce RTX 4060 Ti).

Поскольку главной целью работы алгоритма является подготовка блоков для дальнейшего использования в средах геометрического моделирования, то ключевым параметром можно считать максимальное число точек в блоке – размера узла. Число точек, как и занимаемый ими битовый объем, ограничивает возможность работы с облаком.

Для оценки влияния параметра размера узла на характеристики построенного BVH-дерева исследовались:

- скорость построения структуры;
- число и распределение листьев;
- устойчивость работы критерия SAH в условиях различных типов облаков точек.

В таблице 2 представлены усредненные показатели производительности алгоритма для трех тестовых наборов данных.

Таблица 2. Усредненные показатели работы алгоритма по наборам данных

№	Набор данных	Число точек	Время построения дерева, с	Скорость построения дерева, млн точек/ с	Число листьев	Число узлов
1	Stanford_Bunny	999 966 точек	5.011945	0.405027	24.375	47.75
2	6_data	15 754 340 точек	14.2330961	1.407944	10.45	19.9
3	Alldata	48 389 508 точек	33.14676842	1.640115	13.89474	26.78947

Необходимо отметить, что для набора Stanford_Bunny наблюдается низкая скорость построения дерева. Этот набор данных, несмотря на свой малый объем, был включен в экспериментальную часть для демонстрации работы на равноплотностных облаках. Однако при таком количестве точек в облаке существенную долю времени занимают накладные расходы на инициализацию структур, выделение

памяти и управление потоками. В результате измеряемая производительность оказывалась заниженной по сравнению с крупными наборами данных.

Для оценки качества построения деревьев была проведена серия экспериментов на трех наборах данных, в которой изменялось только значение порога максимального числа точек в листе при фиксированных прочих настройках алгоритма. На рисунке 3 представлено распределение размеров итоговых листовых узлов при различных значениях порога. График демонстрирует возможность управлять формированием блоков путем изменения числа точек в узле. Использование модифицированного критерия SAN обеспечивает контролируемое распределение размеров листьев при изменении максимального числа точек в листе. При увеличении заданного порога медианный размер получаемых листьев закономерно возрастает. Форма кривых распределения указывает на степень однородности итоговых блоков. При более низких значениях порога кривая имеет более крутой наклон – размеры листьев плотно сконцентрированы в узком диапазоне. С увеличением порога кривая становится более полой, что указывает на увеличение разброса в размерах итоговых блоков.

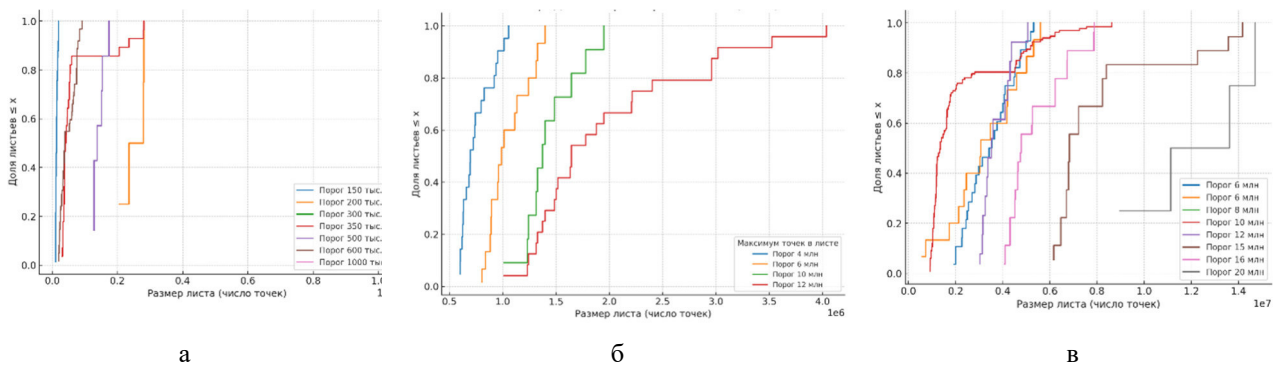


Рисунок 3. Распределение размеров листьев при различных порогах максимального числа точек в листе:
а – Stanford_Bunny; б – 6_data; в – Alldata

Также была исследована зависимость скорости построения дерева от размера листа (рис. 4). График иллюстрирует, что при достижении определенного порога дополнительные разбиения перестают быть оправданными.

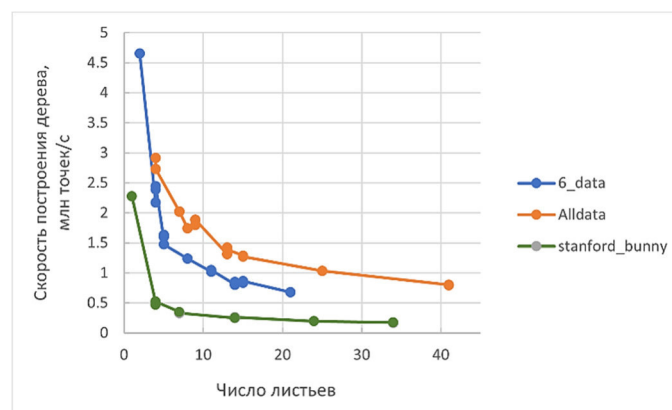


Рисунок 4. Зависимость скорости построения дерева от размера листа

Пример полученной BVH-структуры приведен на рисунке 5. Видно, что дерево является несбалансированным, разные ветви уходят на разную глубину, что порождает листья неоднородного размера. Рисунок показывает, что дерево не делится равномерно, например, значительная часть модели формирует крупный блок с пустотами, а голова оказывается в маленьком блоке (рис. 5, а). Возможна и обратная ситуация (рис. 5, б), когда дерево получается слишком детализированным, появляются мелкие блоки, хотя крупные блоки за счет малого количества точек внутри остаются большими.

Разрез для среднеразмерного набора 6_data приведен на рисунке 6. Как видим, итоговые ограничивающие объемы формируются с учетом всех точек, включая отдаленные артефакты, что может приводить к созданию избыточно пустых пространств внутри блоков.

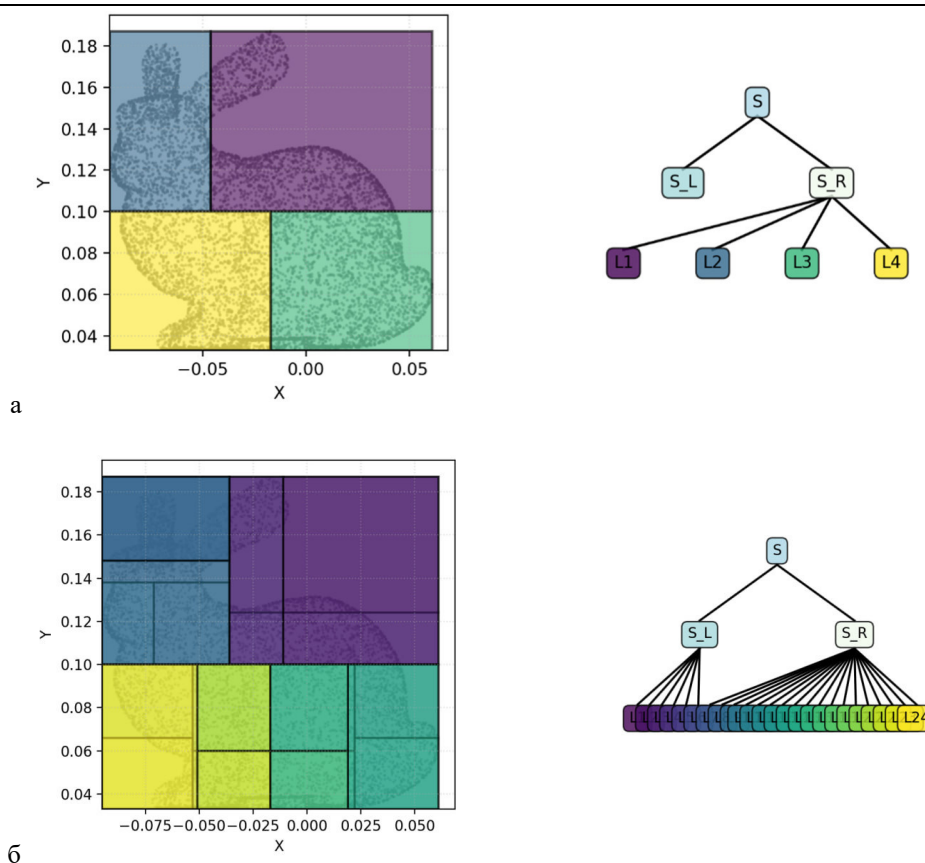


Рисунок 5. Пример несбалансированной BVH-структуры при малом числе блоков для Stanford_Bunny:

а – при малом числе блоко; б – при большом

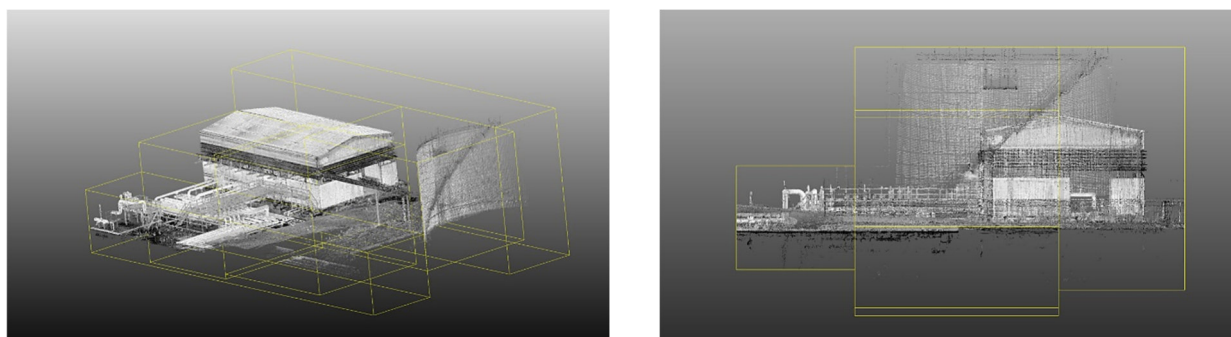


Рисунок 6. Пример разбиения на блоки для набора данных 6_data

Обсуждение результатов

Экспериментальное исследование показало, что предложенный алгоритм построения BVH-структуры решает задачу разбиения облака точек. Базовый критерий SAH обеспечивает геометрически «выгодные» разрезы, однако без дополнительных поправок он склонен формировать несбалансированные деревья. Модифицированный критерий с учетом плотностных и высотных поправок, а также с использованием g-means-теста, напротив, приводит к более равномерному распределению размеров листьев, что подтверждается результатами на рисунке 3.

Введение коэффициентов для плотности и высоты делает алгоритм чувствительным к пространственным неоднородностям. Это особенно важно для промышленных объектов и крупномасштабных сканов, где присутствуют как плотные фрагменты (конструкции, оборудование), так и разреженные (открытые площадки, шум, выбросы). Поправки предотвращают чрезмерное укрупнение или, наоборот, чрезмерное измельчение листьев, но требуют тонкой настройки.

Ограничение на число узлов может определять размер блока для процессов экспорта или импорта в среды геометрического моделирования. При слишком малом значении формируется большое число

мелких блоков и наблюдается резкое замедление построения и рост числа операций ввода/вывода. И наоборот, задав слишком большой порог, можно получить неудобные для обработки блоки.

Заключение

В данной работе представлен алгоритм разбиения облака точек на основе иерархии ограничивающих объемов структуры с критерием стоимости разбиения point-SAH, поправками на контраст плотностей и высот, статистической проверкой на наличие явной кластерной структуры данных на основе алгоритма g-means, динамическими порогами на квантованной решетке. Алгоритм демонстрирует стабильность и предсказуемость результатов даже при вариации параметров, а также предоставляет интуитивно понятные средства управления разбиением. Визуальный и статистический анализ подтвердил, что полученные блоки семантически и геометрически соответствуют объектам. По сравнению с октодеревом или равномерной сеткой BVH создает меньше пустых регионов и умеет останавливать деление в разреженных зонах, что приводит к меньшему числу файлов без потери детализации в плотных областях.

В дальнейшем необходимо исследовать влияние плотностных и высотных надбавок на качество и время разбиения, определить оптимальные значения в зависимости от свойств облака точек. Перспективным направлением является разработка методов динамического выбора коэффициентов на основе локальных статистик (средняя плотность, вариативность высот, наличие выбросов). Такой подход позволит повысить универсальность алгоритма без ручной настройки.

Список литературы

1. Глумова Е. С., Дудаков Н. Ю., Филинских А. Д. Обработка результатов наземного лазерного сканирования для создания цифровых моделей // Научно-технический вестник Поволжья. 2024. № 12. С. 119–124. EDN: JMRRPA
2. A hybrid spatial indexing structure of massive point cloud based on octree and 3D R*-tree / W. Wang et al. // Applied Sciences. 2021. Vol. 11, iss. 20. P. 9581.
3. Schütz M., Ohrhallinger S., Wimmer M. Fast out-of-core octree generation for massive point clouds // Computer Graphics Forum. 2020. Vol. 39, iss. 7. Pp. 155–167.
4. PotreeConverter. Release 2.0 // GitHub. URL: <https://github.com/potree/PotreeConverter/releases/tag/2.0> (дата обращения: 10.08.2025).
5. Pantaleoni J., Luebke D. HLBVH: Hierarchical LBVH construction for real-time ray tracing of dynamic geometry // Proceedings of the Conference on High Performance Graphics. 2010. С. 87–95.
6. Bounding Volume Hierarchies // Physically Based Rendering: From Theory to Implementation. URL: https://pbr-book.org/3ed-2018/Primitives_and_Intersection_Acceleration/Bounding_Volume_Hierarchies (дата обращения: 10.08.2025).
7. Entwine Point Tiles Specification. Hobu Inc., 2021. URL: <https://entwine.io> (дата обращения: 09.08.2025).
8. Cloud Optimized Point Cloud Specification 1.0. Hobu Inc., 2021. URL: <https://copc.io> (дата обращения: 09.08.2025).
9. Point Data Abstraction Library: Documentation. PDAL Project, 2023. URL: <https://pdal.io> (дата обращения: 11.08.2025).
10. LAStools: Efficient Lidar Processing Software. Rapidlasso GmbH, 2022. URL: <https://rapidlasso.de> (дата обращения: 11.08.2025).