

Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра суперкомпьютеров и квантовой информатики

Арбузов Николай Романович

**Исследование эффективности использования kd-деревьев при параллельной обработке облаков точек**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Научный руководитель:**

к.ф.-м.н., доцент

И. М. Никольский

Москва, 2023

Оглавление

[Введение 3](#_Toc131684607)

[Обзор литературы 3](#_Toc131684608)

[Постановка задачи 4](#_Toc131684609)

[Список литературы 6](#_Toc131684610)

# Введение

Обработка облаков точек является важной задачей в таких областях, как компьютерное зрение, графика, геодезия и многих других. Одним из способов индексации облаков точек является использование kd-деревьев, которые позволяют эффективно выполнять операции поиска, добавления и удаления точек. Однако, большинство существующих алгоритмов обработки облаков точек с использованием kd-деревьев являются последовательными и не могут полностью использовать мощность параллельных вычислений.

Целью данного исследования является создание параллельного алгоритма индексирования облаков точек с помощью kd-дерева, и исследование его эффективности. Для достижения данной цели были поставлены следующие задачи:

* Изучение существующих работ по использованию kd-деревьев в обработке облаков точек и параллельных алгоритмов обработки облаков точек
* Разработка методики исследования, включающей описание используемых данных и методов обработки облаков точек, а также алгоритма параллельной обработки облаков точек с использованием kd-деревьев. Планируется создать алгоритм индексации данных, распределенных по узлам суперкомпьютера
* Проведение экспериментального исследования для сравнения эффективности параллельной обработки облаков точек с использованием kd-деревьев и последовательной
* Анализ результатов экспериментов и оценка эффективности использования kd-деревьев в параллельной обработке облаков точек.

Результаты данного исследования могут быть использованы для улучшения существующих алгоритмов обработки облаков точек с использованием kd-деревьев и разработки новых параллельных алгоритмов.

# Обзор литературы

С того момента, как в 1975 году Джон Бентли создал и описал принципы работы kd-дерева (Bentley J., 1975), многие ученые различными способами улучшали методы применения данной структуры, например, используя приближенные вычисления с помощью Akd-дерева (Approximate kd-дерева), в котором для решения задачи ICP (Iterative Closest Point, то есть алгоритм поиска ближайшего соседа) ищется не действительно ближайший сосед, а ближайшая точка из того блока, в котором располагается заданная точка (Greenspan M., Yurick M., 2003). Другим примером улучшения может служить поиск в kd-дереве с кэшированием (Cashed kd-tree search) суть улучшения состоит в том, что узлы помимо указателей на свои дочерние узлы так же хранят и указатель на родительский узел, корневой узел хранит нулевой указатель, таким образом время поиска уменьшается вплоть до 50%, из-за перехода на уровень выше во время возможных тестов ball-within-bounds, тестов на поиск возможных ближайших точек в соседей узлах (Nuchter A., Lingemann K., Hertzberg J., 2007).

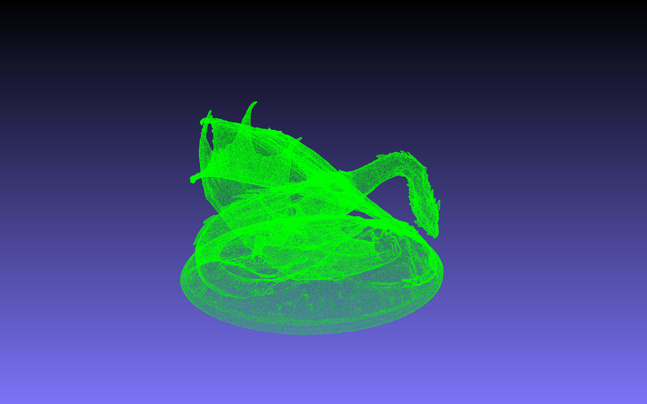
В работе с пространственными данными часто используются kd-деревья, так как имеют архитектуру, заточенную именно под них. Среди областей применения можно выделить такие, как трассировка лучей в реальном времени с использованием обычных процессоров (Shevtsov M., Soupikov A., Kapustin A., 2007) и с использованием графических ускорителей (Zhou K. et al., 2008). Помимо трассировки лучей, kd-деревья могут также использоваться и при обработке облаков точек, например, полученных с помощью датчиков LiDAR (Zhou H. et al., 2021).

Как известно, помимо kd-деревьев для индексации пространственных данных часто используются такие структуры, как октодерево, r-дерево и так далее. Октодерево – это древовидная структура данных, узлы которой окружают некоторую кубическую секцию пространства и либо являются листьями, либо имеют по 8 потомков (Meagher D., 1980). В отличии от kd-деревьев октодеревья часто бывает несбалансированным ввиду своей идеи выделать сразу по 8 дочерних узлов при необходимости. Вследствие этого, как описано в работе Адамсона и Воркапика (Adamsson M., Vorkapic A., 2016), октодеревья занимают в несколько раз больше места в оперативной памяти и затрачивают в несколько раз больше времени на выполнения поиска. Если брать r-деревья, то они представляют из себя древовидную структуру, узлы которой являются вложенными прямоугольниками (параллелепипедами), окруженными корневым узлом (Guttman A., 1984). Однако, как описано в работе Нарасимхулу (Narasimhulu Y. et al., 2021), r-дерево так же уступает kd-дереву по временным показателям, это происходит из-за того, что как алгоритм построения kd-дерева, так и алгоритмы поиска в нем проще и требует меньших вычислительных мощностей по нескольким параметрам: kd-дерево – это, в отличие от r-дерева, бинарное дерево, а значит каждый узел имеет максимум 2 дочерних узла, в то время, как в r-дереве их может быть больше.

# Постановка задачи

В качестве выборки для проведения тестов и измерений будут использованы облака точек, приведенные на Рисунке 1, а также на кубах с 1 000, 10 000 и 100 000 точек, заполненные случайным образом. При выполнении данной работы будет измеряться время построения kd-дерева и время выполнения функции поиска точек на этом дереве в заданном диапазоне для сравнения последовательной и параллельной программы и измерения эффективности параллельной версии.

Рисунок . Слева: облако точек Fantasy Dragon (https://www.artec3d.com/3d-models/fantasy-dragon). Справа: облако точек Bearded guy (https://www.artec3d.com/3d-models/bearded-guy-hd)



Параллельность алгоритма будет основана на использовании парадигмы MPI (Message Passing Interface), она предоставляет набор функций для передачи сообщений между процессами, а также для синхронизации выполнения процессов.

Для проведения опытов будут использованы локальный компьютер в части тестовых запусков и визуализации облаков, а также параллельная вычислительная система Polus для основных измерений. Локальный компьютер оснащен 4‑ядерным процессором Intel Core i5 с частотой 2 ГГц и ОЗУ объемом 16 Гб с частотой 3733 МГц. Система Polus состоит из 5 вычислительных узлов, на каждом узле установлены по 2 10-ядерных процессора IBM POWER8 и каждое ядро имеет 8 потоков, также Polus имеет общую оперативную память 256 Гбайт (в узле 5 оперативная память 1024 Гбайт) с ЕСС контролем.

# Параллельное kd-дерево

В качестве модели для изучения была создана модель распределенного kd-дерева, в которой процессоры выступают в роли узлов (Рисунок 2).

0

1

2

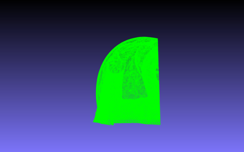
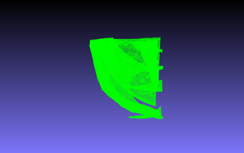
………

i

2i+1

2i+2

Рисунок . Схема распределения процессов по распределенному дереву.



Proc. 2



Proc. 1

Proc. 0

Proc. 3

Proc. 4

Proc. 5



Рисунок . Организация распределенного дерева и передачи и частей облаков точек в нем.

# Список литературы

**Adamsson M., Vorkapic A.** A comparison study of Kd­tree, Vp­tree and Octree for storing neuronal morphology data with respect to performance [Отчет] : Degree Project in Computer Science / Computer Science and Communication ; KTH Royal Institute of Technology. - Stockholm : [б.н.], 2016.

**Bentley J.** Multidimensional Binary Search Trees Used for Associative Searching [Дневник] // Communications of the ACM. - New YorkNYUnited States : Association for Computing Machinery, 1 сентябрь 1975 г.. - 9 : Т. 18. - стр. 509–517.

**Greenspan M., Yurick M.** Approximate kd tree search for efficient ICP [Конференция] // Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM 2003). - Kingston, Ontario, Canada : IEEE, 2003. - стр. 442 - 448.

**Guttman A.** R-trees: a dynamic index structure for spatial searching [Конференция] // Proceedings of the 1984 ACM SIGMOD international conference on Management of data. - Berkeley : University of California , 1984. - стр. 47–57.

**Meagher D.** Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer [Книга]. - Troy : Rensselaer Polytechnic Institute, Image Processing Laboratory, 1980. - Т. 1.

**Narasimhulu Y. et al.** CKD-Tree: An Improved KD-Tree Construction Algorithm [Конференция] // Proceedings of the ISIC 2021: International Semantic Intelligence Conference. - New Delhi, India : SCIS, University of Hyderabad, 2021. - стр. 211-218.

**Nuchter A., Lingemann K., Hertzberg J.** Cached kd tree search for ICP algorithms [Конференция] // Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007). - Osnabruck, Germany : IEEE, 2007. - стр. 419-426.

**Shevtsov M., Soupikov A., Kapustin A.** Highly parallel fast KD‐tree construction for interactive ray tracing of dynamic scenes [Дневник] // Computer Graphics Forum. - Oxford, UK : Blackwell Publishing Ltd, 2007 г.. - 3 : Т. 26. - стр. 395-404.

**Zhou H. et al.** Research on volume prediction of single tree canopy based on three-dimensional (3D) LiDAR and clustering segmentation [Дневник] // International Journal of Remote Sensing. - 2021 г.. - 2 : Т. 42. - стр. 738-755.

**Zhou K. et al.** Real-time kd-tree construction on graphics hardware [Дневник]. - [б.м.] : ACM Transactions on Graphics (TOG), 2008 г.. - 5 : Т. 27.