

*На правах рукописи*



**Беляевский Кирилл Олегович**

**МЕТОДЫ И АЛГОРИТМЫ ФОРМИРОВАНИЯ И ИСПОЛЬЗОВАНИЯ  
ОКТОДЕРЕВА ДЛЯ ОБРАБОТКИ ОБЛАКА ТОЧЕК ЛАЗЕРНОГО  
СКАНИРОВАНИЯ В ОГРАНИЧЕННОМ ОБЪЕМЕ ОПЕРАТИВНОЙ  
ПАМЯТИ**

Специальность 05.13.01 – Системный анализ, управление и обработка  
информации (технические системы)

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Санкт-Петербург – 2020

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Санкт-Петербургский политехнический университет Петра Великого».

Научный руководитель: **Мелехин Виктор Федорович**, доктор технических наук, профессор Высшей школы интеллектуальных систем и суперкомпьютерных технологий ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого».

Официальные оппоненты: **Орлова Юлия Александровна**, доктор технических наук, доцент, заведующая кафедрой «Программное обеспечение автоматизированных систем» ФГБОУ ВО «Волгоградский государственный технический университет»;

**Яворский Ростислав Эдуардович**, кандидат физико-математических наук, советник ректора ФГАОУ ВО «Национальный исследовательский Томский политехнический университет».

Ведущая организация: ФГБОУ ВО «Санкт-Петербургский государственный университет».

Защита диссертации состоится «24» ноября 2020 г. в \_\_\_ часов \_\_\_ минут на заседании диссертационного совета Д.002.199.01, созданного на базе Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» по адресу: 199178, Санкт-Петербург, 14 линия В.О., 39.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» наук <http://www.spcras.ru>.

Автореферат разослан « \_\_\_ » \_\_\_\_\_ 2020г.

Ученый секретарь  
диссертационного совета Д 002.199.01  
кандидат технических наук



А.А. Зайцева

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы диссертации.** Одним из наиболее прогрессивных методов сбора цифровой пространственной информации на текущее время является лазерное сканирование (ЛС). Главными достоинствами устройств лазерного сканирования считаются высокая точность, возможность автоматизации процесса сбора данных, высокая разрешающая способность измерений. Результатом таких измерений является трехмерное облако точек, с высокой точностью отражающее геометрию объекта исследования. Современные лазерные сканеры могут быть компактными и мобильными, их точность и скорость работы на текущий момент выросла на порядок, а в некоторых случаях на несколько порядков.

Лазерные сканирующие системы позволяют производить миллионы измерений в секунду, а размеры получаемых облаков точек могут достигать нескольких сотен гигабайт, что предъявляет высокие требования к вычислительным ресурсам при обработке таких данных. В результате имеет место очевидное противоречие между быстрым развитием технологий ЛС и возможностями обработки результатов, на разрешение которого направлена настоящая работа.

Используемые стандартные методы обработки данных лазерного сканирования не позволяют в полной мере использовать возможности ЛС. На текущий момент существует и используется множество алгоритмов обработки облаков точек, например, предназначенных для построения полигональной поверхности или цифровой модели рельефа, сегментации по различным признакам, объединения нескольких облаков точек, сбора статистической информации, фильтрации, построения проекций и т.п. Обработка результатов лазерного сканирования в настоящее время зачастую связана с конкретным разработчиком/производителем оборудования для ЛС и программного обеспечения (ПО), предназначенного для этого.

Среди открытого ПО наблюдается нехватка инструментов для обработки облаков точек, чей размер превышает доступную оперативную память (ОП). Так как объемы доступной оперативной памяти могут варьироваться, и зачастую не способны вместить требуемое количество данных, а рост объемов оперативной памяти значительно отстает от роста объемов облаков точек, вызванного развитием технологий ЛС, задача организации обработки облака точек при ограниченном объеме оперативной памяти является актуальной.

В диссертационной работе представлены результаты разработки методов и алгоритмов решения задачи обработки больших облаков точек с использованием структурирования информации при помощи октодеревя и различных подходов к выделению (аллокации) памяти, что позволяет разрешить противоречие между все возрастающим объемом данных ЛС и ограничений по оперативной памяти.

**Степень разработанности темы исследования.** Октодерево широко используется в различных областях, связанных с большими объемами трехмерных данных, начиная с момента представления его в начале 80-х годов. В середине 80-х были рассмотрены точки в качестве примитивов визуализации для моделей с замкнутыми поверхностями. На текущий момент получение сцен реального мира с использованием технологии трехмерного сканирования является стандартной техникой, область применения которой варьируется от сканирования механических

деталей до сканирования памятников исторического наследия и городов. Среди выдающихся исследователей указанных направлений присутствуют Мигер Д., Левой М., Уитед Т., Русинкевич Ш., Цвикер М., Паули М., Нолл О., Гросс М., Виттер Д. С., Самет Х., Шибauer К., Шуц М.

Однако, на текущий момент, ввиду существенного роста объемов облаков точек, в системах обработки данных не хватает оперативной памяти, а при использовании внешней памяти (ВП) существенно снижается производительность. В существующих широко используемых открытых программах и библиотеках обработки облаков точек предполагается их размещение в оперативной памяти. При разработке средств повышения производительности при размещении облака точек во внешней памяти актуально предусмотреть использование этих средств со стандартными программами обработки. Таким образом, исследования в области формирования октодерев для обработки больших облаков точек, позволяющего сохранить приемлемую производительность обработки при росте размера облака, сократить нагрузку на файловую систему, а также обеспечить возможность интеграции с другими программными решениями, являются актуальным и перспективным направлением научных исследований.

**Целью работы** является снижение затрат времени на обработку облака точек при хранении во внешней памяти за счет новых методов, алгоритмов и программного обеспечения организации хранения и доступа к этой информации, снижающих потребление оперативной памяти. Для достижения цели в диссертационной работе поставлены и решены следующие **задачи**:

1. Системный анализ процессов формирования и использования октодерев по облаку точек ЛС при использовании внешней памяти, включающий анализ методов, алгоритмов и структур данных, применяемых для обработки в оперативной и внешней памяти, их декомпозицию на составные компоненты, исследование их взаимодействия между собой, центральным процессором, внешней и оперативной памятью с целью выделения компонентов, за счет новой организации которых можно уменьшить затраты времени.

2. Постановка задачи снижения затрат времени на обмен с внешней памятью, определение критерия и показателей для оценивания эффективности реализации вычислительного процесса обработки. Формулирование гипотезы об организации вычислительного процесса обработки и структур данных, позволяющей сократить временные затраты на использование внешней памяти, и формирование подзадач на последующее исследование и уточнение предположений, основанных на данной гипотезе.

3. Разработка новых методов организации вычислительного процесса обработки облака точек, основанных на выдвинутой гипотезе. Разработка компонентов, алгоритмов и структур данных таких систем, исследование взаимодействий между ними, системным ПО, оперативной и внешней памятью.

4. Планирование и проведение экспериментальных исследований с целью оценки эффективности предложенных методов для различных задач обработки облака точек и подтверждения правильности выдвинутой гипотезы. Определение показателей, характеризующих вычислительный процесс обработки облака точек при

использовании внешней памяти, и разработка вычислительных экспериментов для получения этих показателей и сравнения с существующими реализациями. Разработка методики выбора параметров октодерева в зависимости от целевой направленности обработки. Исследование возможности применения предложенных решений в сторонних программных библиотеках обработки облаков точек, ориентированных на работу в оперативной памяти, и разработка соответствующих практических рекомендаций.

**Объектом исследования** является структура данных облака точек ЛС при использовании внешней памяти и процесс обработки этой информации.

**Предметом исследования** являются методы и алгоритмы формирования октодерева облаков точек при использовании внешней памяти и организация управления обработкой информации, обеспечивающая уменьшение затрат времени на обмена с внешней памятью.

**Научная новизна** предлагаемой диссертации состоит в следующем:

1. Разработаны концептуальные модели организации обработки облака точек, формирования октодерева, компонентов вычислительного процесса обработки облака точек во внешней памяти, позволившие за счет декомпозиции этих процессов выделить операции, выполнение которых с использованием предложенных способов организации доступа к блокам данных дало возможность существенно ускорить обработку. Разработаны модели вычислительного процесса обработки облака точек, анализ которых позволил предложить ряд модификаций, повышающих эффективность процесса обработки.

2. В результате анализа организации данных и вычислительного процесса предложена модель иерархической структуры данных октодерева, использующаяся при обработке облака точек в оперативной или внешней памяти и позволяющая ускорить выполнение операций доступа к данным октодерева, а также отличающаяся возможностью масштабирования для работы с данными произвольной размерности и возможностью динамического расширения структуры октодерева без увеличения его глубины.

3. Разработаны два метода, позволяющие добиться снижения временных затрат на использование внешней памяти при ограничении потребления оперативной памяти в процессе обработки облака точек, основанные на выдвинутой гипотезе о изменении способов доступа, размещения и идентификации блоков данных октодерева во внешней памяти. Первый метод использует асинхронную систему кеширования, модифицированную процедуру формирования октодерева и отличается возможностью присоединения узлов октодерева к общему файлу. Как следствие, он характеризуется сокращением количества создаваемых в процессе формирования файлов и уменьшением затрачиваемого времени на обмен данными. Второй метод использует механизм отображения памяти совместно с алгоритмом динамической аллокации, а также применяет целочисленную иерархическую модель октодерева для ускорения доступа к данным. Отличается возможностью прямого доступа к данным во внешней памяти, сокращением количества создаваемых в процессе формирования файлов до одного и, как следствие, уменьшением затрачиваемого времени на обмен данными.

4. Предложен способ обработки больших облаков точек в сторонних библиотеках, ориентированных на работу в оперативной памяти, позволяющий помимо ограниченной оперативной памяти использовать внешнюю память, обеспечивая прямой доступ к данным за счет механизмов отображения памяти и динамической аллокации. Получены экспериментальные результаты, подтверждающие возможность такой обработки без существенного падения производительности в сравнении с обработкой информации только в основной памяти и доказывающие возможность работы с облаком точек, превышающим объемы оперативной памяти.

**Теоретическая значимость работы.** Разработаны концептуальные модели организации обработки облака точек, формирования октодерева и компонентов вычислительного процесса обработки облака точек во внешней памяти, позволившие провести анализ процесса обработки и выделить этапы и компоненты, модификация которых с учетом структуры данных и новых способов кодирования доступа позволила повысить производительность при использовании внешней памяти для октодерева.

**Практическая ценность работы** заключается в создании программной системы, предназначенной для ограничения потребления оперативной памяти в процессе обработки облаков точек и подтверждающей теоретические результаты работы. Предложенные подходы, методы и алгоритмы позволяюткратно снизить потребление оперативной памяти при выполнении типовых операций обработки облаков точек, сохраняя при этом приемлемую производительность.

**Методология и методы диссертационного исследования.** В качестве методической и теоретической основы в данном диссертационном исследовании используются методы системного анализа и обработки информации, теория множеств и отношений. При разработке архитектуры программного обеспечения применяется объектно-ориентированный подход.

**Положения, выносимые на защиту.** На основе проведенной работы и ее экспериментальной апробации на защиту выносятся следующие положения:

1. Концептуальные модели формирования октодерева и вычислительного процесса обработки облака точек во внешней памяти, предназначенные для выделения компонентов и этапов, модификацией которых с учетом особенностей структуры облака точек можно сократить затраты времени при обработке.

2. Гипотеза об уменьшении затрат времени на обращения к внешней памяти при обработке облака точек за счет изменения способов доступа, размещения и идентификации блоков данных октодерева во внешней памяти, позволяющих сократить количество файловых операций и создаваемых файлов, а также снизить количество задержек, обусловленных файловой системой. Сформированные на базе данной гипотезы предположения об организации вычислительного процесса позволили выделить этапы и компоненты обработки, которые стали предметом детального исследования, что позволило предложить новые методы обработки, являющиеся основными результатами.

3. Метод и алгоритм предобработки информации облака точек лазерного сканирования, заключающийся в ее структурировании путем формирования октодерева на базе асинхронной двухуровневой системы кеширования, использующий

внешнюю память при превышении обрабатываемой информацией объема оперативной памяти и позволяющий сократить количество создаваемых файлов для снижения влияния обменов с внешней памятью на производительность.

4. Метод и алгоритм предобработки информации облака точек лазерного сканирования, заключающийся в ее структурировании путем формирования октодерева на базе механизма отображения памяти, использующий внешнюю память для хранения информации с возможностью прямого доступа и интерактивной модификации, а также позволяющий ускорить выполнение операций доступа к данным октодерева за счет кодирования с использованием целочисленной арифметики и сократить количество создаваемых файлов до одного для снижения влияния обменов с внешней памятью на производительность.

**Реализация результатов работы.** Внедрение представленных алгоритмов и методов было произведено в рамках проекта ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014 - 2020 годы» по теме: «Исследование и разработка алгоритмов и программных средств по обработке, хранению и визуализации данных лазерного сканирования и фотосъемки» (Уникальный идентификатор проекта RFMEFI58417X0025) индустриальным партнером ООО «Экоскан» (Соглашение о предоставлении субсидии от 03.10.2017 г. № 14.584.21.0025). Представленные в работе методы и алгоритмы используются в качестве алгоритмического и методического обеспечения для разработки программного комплекса, предназначенного для обработки, хранения и визуализации данных лазерного сканирования.

**Обоснованность и достоверность** предложенных методов и алгоритмов подтверждается согласованностью теоретических положений и результатов, полученных при практической реализации предложенных методов и алгоритмов, апробацией основных теоретических положений диссертации в печатных трудах и докладах, а также положительными результатами внедрения основных положений диссертации.

**Апробация результатов работы.** Результаты исследований докладывались на следующих конференциях: 19th International Conference on Computational Science and its Applications, Saint-Petersburg, 2019; XXI Международная конференция по мягким вычислениям и измерениям, Санкт-Петербург, 2018г.

Решение апробировано компанией ООО «ЭкоСкан» в программном комплексе для обработки, хранения и визуализации данных лазерного сканирования и фотосъемки. Практическим результатом использования диссертационной работы является использование предложенного подхода для снижения загрузки оперативной памяти в процессе работы с облаками точек. Это позволяет выполнять обработку облаков точек лазерных отражений, размер которых значительно превышает доступные объемы оперативной памяти.

Зарегистрированы следующие результаты интеллектуальной деятельности (РИД): «Программа для создания растровых проекций», №2018665901, 11.12.2018; «Программа для сортировки и фильтрации облака точек», №2018666456,

17.12.2018; «Программное средство построения проекций облака точек, полученных от мобильного комплекса лазерного сканирования», № 2015617817, 22.07.2015.

**Публикации.** Автором опубликовано по теме диссертации 7 печатных работ, среди них 2 работы в рецензируемых журналах из перечня ВАК и 2 работы, включенные в систему цитирования Scopus, а также зарегистрировано 7 Результатов Интеллектуальной Деятельности.

**Структура и объем работы.** Диссертационная работа объемом в 204 машинописных страницы состоит из введения, 4 разделов, списка литературы из 98 наименований, списка сокращений и условных обозначений, 78 иллюстраций, 12 таблиц, 14 листингов, 1 приложения и предметного указателя.

### ОСНОВНОЕ СОДЕРЖАНИЕ ДИССЕРТАЦИИ

**Во введении** содержится обоснование актуальности диссертационной работы, определены цели, объект и предмет исследований, а также методы исследований. Сформированы основные научные результаты, выносимые автором на защиту, а также их практическая ценность.

**В первой главе** проводится анализ процессов получения, обработки и преобразования данных лазерного сканирования. Проводится обзор существующих методов обработки облаков точек в ограниченном объеме потребляемой оперативной памяти. Рассматриваются структуры разбиения пространства и их применение для организации обработки облака точек с использованием вторичных систем хранения данных. Формируются гипотезы об уменьшении затрат времени за счет предлагаемых способов размещения октодерев и организации взаимодействия между оперативной и внешней памятью. Производится постановка задачи снижения затрат времени на обмен с внешней памятью за счет организации хранения октодерев облака точек в оперативной и внешней памяти и организации управления обработкой информации с учетом особенностей структуры октодерев.

Данные лазерного сканирования представлены облаком точек. Облако точек — это неупорядоченное множество точек в трехмерном евклидовом пространстве, полученное в результате трехмерного сканирования объекта, и представляющее поверхность этого объекта:

$$P = \{p_1, \dots, p_s\}. \quad (1)$$

Точка из облака точек представляет собой вектор (кортеж) координат ( $a_{coord}$ ), нормалей ( $a_{normal}$ ), цвета ( $a_{color}$ ) и прочей информации, используемой при построении облака точек:

$$p_i = (a_{coord}^i, a_{color}^i, a_{normal}^i, \dots), p_i \in P, \quad (2)$$

где  $a_{coord} = (x, y, z)$ , где  $x, y, z$  — трехмерные координаты точки;

$a_{color} = (r, g, b)$ , где  $r, g, b$  — красная, зеленая и синяя компоненты цвета;

$a_{normal} = (nx, ny, nz)$ , где  $nx, ny, nz$  — компоненты нормали.

На рисунке 1 приведена обобщенная концептуальная модель обработки данных ЛС.



Рисунок 1 – Концептуальная модель организации обработки облака точек

Облако точек является неорганизованной структурой данных. Для ускорения операций пространственного поиска обычно используются специальные структуры данных, описывающие объем, занимаемый облаком точек, и позволяющие идентифицировать каждую точку в облаке как принадлежащую к определенной области, что позволяет снизить временную сложность поиска до  $O(\log n)$ . Далее будем называть подобную структуру данных структурой разбиения пространства.

Рассмотрены методы, основанные на использовании структур разбиения, сжатия данных, потоковой обработки. В процессе анализа существующих реализаций обработки данных облака точек с использованием внешней памяти было выявлено, что механизмы файловой системы (в частности, способы идентификации файлов, получения к ним доступа и изменения их размеров), использующиеся при взаимодействии с внешней памятью, показывают падение производительности при работе с большим количеством файлов и могут вызывать ошибки при превышении максимального количества файлов в файловой системе или файловых дескрипторов процесса.

Таким образом, существует возможность ускорения выполнения операций взаимодействия с внешней памятью для октодерева, в котором для хранения блоков данных используются отдельные файлы (с целью обеспечения возможности наращивания их размеров в результате заполнения октодерева), при помощи более низкоуровневого управления внешней памятью в пределах одного или нескольких файлов взамен более широко направленных механизмов файловой системы.

На основе данного наблюдения была выдвинута гипотеза об уменьшении затрат времени на обращения к внешней памяти при обработке облака точек за счет изменения способов доступа, размещения и идентификации блоков данных октодерева во внешней памяти, позволяющих сократить количество файловых операций и создаваемых файлов, а также снизить количество задержек, обусловленных файловой системой.

На рисунке 2 (слева) приведена модель формирования октодерева, основанная на онтологическом принципе и показывающая связи между исходным облаком точек, элементами иерархии октодерева и непосредственно блоками данных в оперативной или внешней памяти:

$$O = \langle P, N, R^n, L, R^{nl}, R^{lp}, V, V^{RAM}, V^{HDD} \rangle, \quad (3)$$

где

**Исходные данные:**

$P = \{p_1, \dots, p_s\}$  – множество точек в облаке точек;



ция операций в ОП (б) и ВП (в). Помимо повышения эффективности общих операций обработки облака точек (например, скорости обхода октодеревя), был выделен ряд мер, связанных с взаимодействием с внешней памятью:

- Сокращение числа операций во внешней памяти за счет кеширования данных в оперативной памяти;
- Сокращение числа операций с ФС за счет введения более низкоуровневых процедур доступа к данным;
- Сокращение количества создаваемых файлов для снижения нагрузки на ФС;
- Асинхронное выполнение части операций для снижения задержек взаимодействия с внешней памятью;
- Сокращение количества обменов с внешней памятью за счет модификации алгоритма обработки данных (на примере алгоритма формирования октодеревя).

В дополнение к повышению эффективности взаимодействия с внешней памятью также были рассмотрены возможности повышения эффективности операций обхода и формирования октодеревя. Они включают в себя меры по сокращению числа операций с плавающей точкой и модификацию алгоритма формирования, позволяющую сократить объем пересылаемых данных.

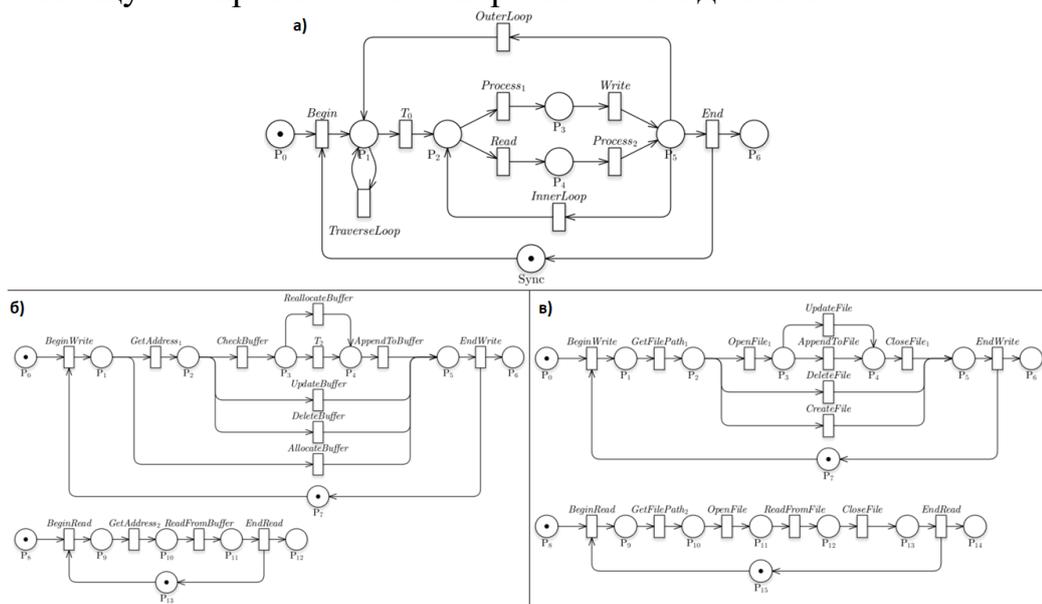


Рисунок 3 – Обобщенный процесс обработки облака точек (а) и детализация операций в ОП (б) и ВП (в)

В данной работе выдвигается предположение, что модификация структурной организации данных и вычислительного процесса обработки одним из предложенных далее методов (Таблица 1) улучшит производительность обработки в ВП. Предложено два метода, отличающихся способом взаимодействия с внешней памятью, и позволяющие произвести экспериментальное исследование мер по повышению эффективности вычислительного процесса. Первый метод позволяет исследовать процедуру обработки, не отказываясь полностью от операций файловой системы. Во втором методе выполнение функций файловой системы (в основном –

выделение блоков данных (файлов) и наращивание их размеров) возьмет на себя операционная система и механизм динамической аллокации.

Таблица 1 – Предложенные модификации структурной организации данных и вычислительного процесса обработки

Метод на основе системы кеширования. Модификация системы обработки во внешней памяти.	Метод на основе механизма отображения. Модификация системы обработки в оперативной памяти.
Сократить количество операций с плавающей точкой, что позволит ускорить операции поиска и обхода узлов.	Подменить буферы данных, выделенные в оперативной памяти, на буферы, выделенные в отображаемой памяти.
Сократить количество создаваемых файлов, что позволит сократить накладные расходы на взаимодействие с ФС.	Для управления блоками данных в отображаемой памяти использовать механизм динамической аллокации.
Модифицировать процедуру формирования октодеревя, что позволит сократить количество чтений из ВП.	Отказаться от операций с плавающей точкой в процессе обхода и поиска узлов.
Выполнять часть операций асинхронно, что позволит снизить задержки, связанные с взаимодействием с ВП.	Модифицировать иерархическую структуру октодеревя для применения целочисленных операций и упрощения доступа и модификации.

Введем  $X = \{x_i, i = 1 \dots n\}$  – конечное множество возможных вариантов реализации вычислительного процесса обработки облака точек, где  $x_i$  – номер варианта обработки. Тогда, согласно выдвинутой гипотезе, существует собственное подмножество вариантов реализации на основе системы кеширования  $X^{cache} \subset X$ , и собственное подмножество вариантов реализации на основе механизма отображения памяти  $X^{mmap} \subset X$ . Также в целях сравнения рассматриваются варианты реализации с обработкой в оперативной памяти  $x^{ram}$  и обработкой при помощи системы кеширования и механизмов ФС  $x^{std}$ .

Формализация вычислительного процесса обработки для различных вариантов обработки выполняется при помощи сетей Петри, и представлена двудольным ориентированным мультиграфом:

$$G_\chi = \langle P_\chi, T_\chi, E_\chi \rangle, \quad (4)$$

где  $\chi \in \{1, 2, 3, 4\}$  – индекс, характеризующий вариант реализации вычислительного процесса:  $x_1^{cache} \in X^{cache}$ ,  $x_2^{mmap} \in X^{mmap}$ ,  $x_3^{ram}$ ,  $x_4^{std}$ , причем  $x^{cache}$  и  $x^{mmap}$  соответствуют конкретным вариантам обработки на базе соответствующих компонентов вычислительного процесса, получение которых производится во второй главе;

$P_\chi$  – конечное непустое множество позиций;

$T_\chi$  – конечное непустое множество переходов (или событий);

$E_\chi$  – конечное множество дуг,  $E_\chi \subseteq P_\chi \times T_\chi \cup T_\chi \times P_\chi$ .

В качестве показателя эффективности предложенного варианта обработки предлагается выбрать показатель общих затрат времени обработки  $K_\chi^{time}$ , которое требуется минимизировать:

$$K_\chi^{time} \rightarrow \min \quad (5)$$

и сравнить его с временем обработки реализации в оперативной памяти  $K_3^{time}$  на одинаковом наборе данных.

Рассмотрим ограничения, которым должны удовлетворять выбранные варианты обработки. Основным ограничением является пиковая потребляемая оперативная память в процессе обработки:

$$M_{\chi}^{ram}(t) \leq M^{limit}, t \in T, \quad (6)$$

где  $M_{\chi}^{ram}(t)$  – объем потребления оперативной памяти в момент времени  $t$ ;  
 $M^{limit}$  – лимит потребления оперативной памяти;  
 $t \in T$  – множество моментов времени обработки облака точек.

В заключение главы, произведена декомпозиция общей задачи повышения эффективности реализации процесса обработки на следующие подзадачи:

1. Анализ структурной организации октодерева и механизмов взаимодействия оперативной и внешней памяти. Исследование способов сокращения размеров узлов в ОП, их идентификации, а также повышения эффективности операций обхода и поиска в октодереве;

2. Исследование подмножества вариантов реализации вычислительного процесса обработки во внешней памяти на основе системы кеширования ( $X^{cache}$ ). Выбор политики вытеснения в системе кеширования. Организация локальности данных по пространственному признаку;

3. Исследование подмножества вариантов реализации вычислительного процесса обработки во внешней памяти на основе механизма отображения памяти ( $X^{map}$ ). Исследование механизмов низкоуровневого управления памятью на внешней системе хранения.

**Во второй главе** производится анализ и уточнение различных вариантов реализации вычислительного процесса обработки и механизмов взаимодействия оперативной и внешней памяти, предложенных в первой главе.

Проведена оценка сложности доступа к данным облака точек в исходном виде и после построения октодерева для операций вставки, удаления, получения и поиска элемента (точки). Проведен анализ организации октодерева облака точек в памяти ВС. Структура разбиения пространства представлена набором узлов  $N = \{n_1, \dots, n_m\}$  и листьев  $L = \{l_1, \dots, l_k\}$ , каждый из которых имеет уникальный идентификатор  $R^{id} = \{r^{id} | r^{id} = id(x), id(x): x \rightarrow r^{id}, x \in N \cup L\}$ , а также множество дуг, представляющих отношения между узлами и листьями  $R^{nl} = \{R_1^{nl}, \dots, R_k^{nl}\}$ .

Рассмотрено представление октодерева в памяти компьютера. Для подсчета занимаемой ими оперативной и внешней памяти, введем:

$V$  – множество блоков данных, принадлежащих листьям

$$V = \{v_1, \dots, v_k\}, \quad (7)$$

где  $v_i$  – блок данных, принадлежащий листу октодерева  $l_i$ ;

$R^{LeafId}$  – множество идентификаторов листьев, по которым также ведется обращение к блокам данным листьев

$$R^{LeafId} \subset R^{id} | \forall a \in R^{LeafId}: id^{-1}(a) \in L, \quad (8)$$

где  $id^{-1}(a)$  – функция получения узла или листа по идентификатору;

$U$  – упорядоченная последовательность запросов к блокам данных в процессе обработки

$$U = (u_1, \dots, u_n), u \in R^{LeafId}, \quad (9)$$

где  $u_i$  – это обращение к блоку данных листа октодеревы по его идентификатору в момент времени  $t_i$  при условии что время дискретно, и в один момент времени может быть только одно обращение.

Введем  $V^{RAM}(t)$  – множество блоков данных, находящихся в оперативной памяти в момент времени  $t$ , и  $V^{HDD}(t)$  – множество блоков данных, находящихся во внешней памяти в момент времени  $t$ . Также можно выразить множество создаваемых файлов во внешней памяти как  $F(t) = \{F_i\}, 0 \leq i \leq k$ , где  $k$  – общее количество блоков данных во внешней памяти в момент времени  $t$ .

Для подсчета потребляемой памяти введем функцию получения размера блока данных в байтах  $M(v): v \rightarrow b, v \in V, b \in \mathbb{R}$ . Тогда объем потребляемой оперативной памяти можно выразить как  $M^{RAM}(t) = \sum_{v \in V^{RAM}(t)} M(v)$ , а объем потребляемой внешней памяти как  $M^{HDD}(t) = \sum_{v \in V^{HDD}(t)} M(v)$ .

Преобразование расположения данных при операциях обработки  $u_i$  соответствует определению:

$$f(t_i, U): V^{RAM}(t_i), V^{HDD}(t_i), F(t_i) \xrightarrow{U} V^{RAM}(t_{i+1}), V^{HDD}(t_{i+1}), F(t_{i+1}). \quad (10)$$

Проведен анализ иерархической модели октодеревы и способов идентификации узлов. Рассмотрены общая структура узла октодеревы и способы ее компактного представления. Рассмотрен способ представления иерархической структуры октодеревы на арифметике с плавающей точкой. Предложена иерархическая модель октодеревы с использованием целочисленной арифметики, преимуществом которой является ускорение выполнения операций обхода октодеревы, а также возможность применения гибридного подхода в описании иерархии октодеревы.

Произведен анализ и уточнение компонент и деталей реализации для рассматриваемых семейств альтернатив организации вычислительного процесса обработки. Дополнительно, в целях сравнения, рассматривается организация вычислительного процесса обработки в оперативной памяти.

**Формирование октодеревы в оперативной памяти.** Объем используемой оперативной памяти в таком октодереве не снижается за счет использования внешней памяти и может превысить лимит потребления. Не удовлетворенное требование к ограничению объема потребления оперативной памяти делает данный метод непригодным для решения поставленной задачи.

$$\begin{aligned} M^{RAM}(P, x_3^{ram}, t_i) &> M^{limit}, \\ K_{time}(P, x_3^{ram}) &= 0, \\ K_{files}(P, x_3^{ram}, t_i) &= 0, \end{aligned} \quad (11)$$

где  $x_3^{ram}$  – вариант реализации с обработкой в оперативной памяти;

$P$  – произвольное облако точек, чей размер превышает доступные объемы оперативной памяти;

$K_{files}$  – количество созданных файлов в момент времени  $t_i$ .

**Использование системы кеширования данных.** Данный метод выполняет управление размещением блоков данных при помощи двухуровневой системы кеширования, в которой первым уровнем является размещение блока в оперативной памяти, а вторым уровнем – размещение блока на жестком диске. При вытеснении блоков используется политика Least Recently Used (LRU). Потребляемая память лимитируется заданным значением, однако в процессе создается большое количество файлов, в худшем случае соразмерное количеству узлов в октодереве. Затраты времени на выполнение сервисных операций растут с количеством создаваемых файлов.

$$\begin{aligned} M^{RAM}(P, x_1^{cache}, t_i) &\leq M^{limit}, \\ K_{time}(P, x_1^{cache}) &= T^{read} + T^{write} + T^{service}, \\ K_{files}(P, x_1^{cache}, t_i) &= \begin{cases} k, & 0 \leq i < n, \\ 1, & i = n \end{cases}, \end{aligned} \quad (12)$$

где  $P$  – произвольное облако точек, чей размер превышает доступные объемы оперативной памяти;

$x_1^{cache}$  – метод обработки, принадлежащий вариантам обработки на основе системы кеширования  $X^{cache}$ ;

$T^{service} = T^{create} + T^{open} + T^{append}$ ,  $T^{create}$  – затраты времени на создание файла,  $T^{open}$  – затраты времени на операции получения файлового дескриптора,  $T^{append}$  – затраты времени на наращивание размеров файла;

$k$  – количество узлов октодеревя;

$n$  – количество операций с октодеревом.

**Использование механизма отображения памяти.** Данный механизм позволяет выполнять отображение содержимого файла на диапазон виртуальных адресов процесса, что дает возможность напрямую работать с сохраненными блоками точек как с данными в оперативной памяти, а также сократить количество создаваемых файлов до одного.

$$\begin{aligned} M^{RAM}(P, x_2^{map}, t_i) &\leq M^{limit}, \\ K_{time} &= T^{cache} + T^{write} + T^{service}, \\ K_{files}(P, x_2^{map}, t_i) &= 1, \end{aligned} \quad (13)$$

где  $x_2^{map}$  – метод обработки, принадлежащий вариантам обработки на основе системы отображения памяти  $X^{map}$ ;

$T^{service} = T^{allocate} + T^{free}$ ,  $T^{allocate}$  – затраты времени на аллокацию блока данных на отображаемой памяти,  $T^{free}$  – затраты времени на освобождение блока данных в отображаемой памяти.

Проведенный анализ проблемы структурирования большого облака точек и способов ее решения позволил произвести уточнение рассмотренных в первой главе семейств альтернатив, определив набор используемых компонентов, способы организации и идентификации узлов октодеревя, а также механизмы взаимодействия с внешней памятью.

**В третьей главе** рассматривается реализация вычислительного процесса обработки на основе системы кеширования и механизма отображения памяти. Предложен алгоритм построения октодеревя с использованием системы кеширования и

возможностью объединения заполненных узлов в общий файл. Для обеспечения возможности использования механизма отображения памяти в октодереве предложена реализация алгоритма динамической аллокации на отображаемой памяти. Для обеспечения возможности обработки больших облаков точек сторонними библиотеками предложен способ обработки больших облаков точек путем внедрения системы аллокации отображаемой памяти в сторонние библиотеки для linux систем.

Был осуществлен переход от формального описания вычислительного процесса, представленного сетью Петри, к его программному представлению, схема которого приведена на рисунке 4 для октодеревя на базе системы кеширования (слева) и октодеревя на базе механизма отображения памяти (справа).

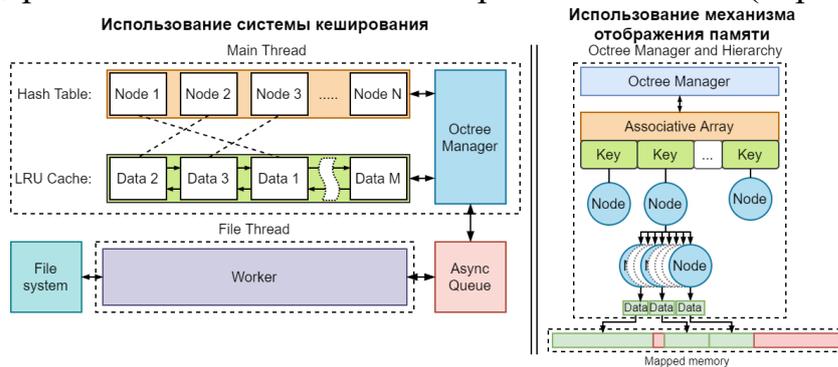


Рисунок 4 - Общая схема программной реализации октодеревя на базе системы кеширования (слева) и на базе механизма отображения памяти (справа)

В четвертой главе проведено экспериментальное исследование предложенных методов формирования октодеревя, а также основанных на них алгоритмов обработки облаков точек. Предложенные алгоритмы сравниваются с существующими решениями, рассматриваются их недостатки и ограничения. Рассмотрено применение октодеревя для решения задачи построения растровых проекций.

Произведен анализ эффективности предложенных методов при помощи измерения потребляемой памяти, времени построения и скорости поиска точек для облаков точек, приведенных в таблице 2. Выбор облаков точек производится так, чтобы обеспечить дифференциацию по размеру, плотности, топологии и методу съемки в целях демонстрации работы алгоритмов на широком спектре входных наборов данных. Представленные облака точек были получены в результате решения реальных промышленных и исследовательских задач.

Таблица 2 - Облака точек для тестирования

№	Название	Кол-во точек, млн	Размер, ГБ
1	five.las	1.2	0.03
2	polytech.las	51.6	1.8
3	smolny.las	126	4.3
4	molodezhnoe.las	1560	53.4

Произведено измерение скорости построения октодеревя в зависимости от параметров максимального количества точек в узле (Рисунок 5), а также скорости поиска в октодеревя в зависимости от параметров максимального количества точек в узле (Рисунок 6).

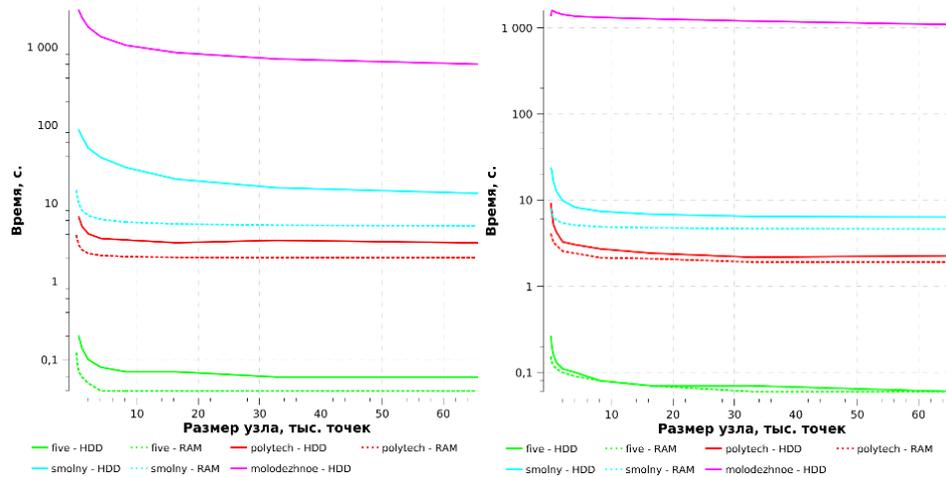


Рисунок 5 - Скорость построения октодеревя с кешированием (слева) и октодеревя с отображением (справа) в зависимости от размера узла

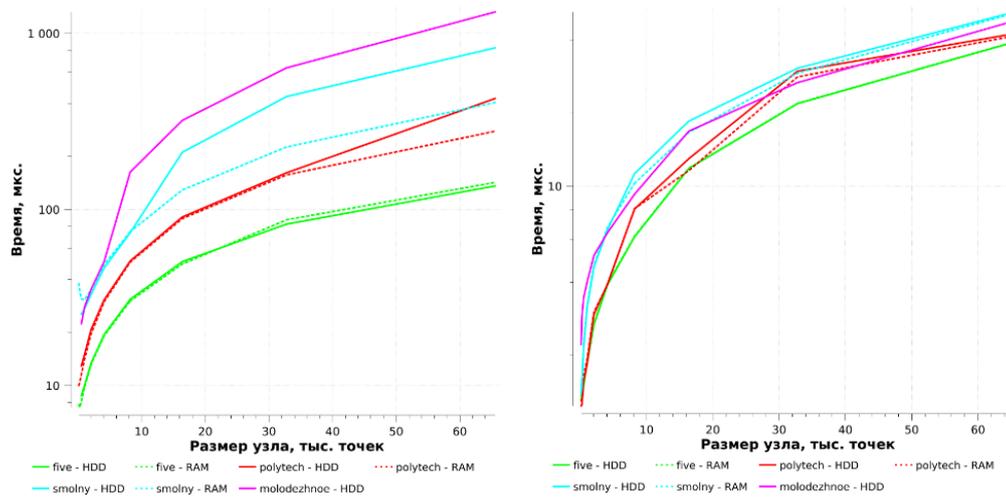


Рисунок 6 - Скорость поиска в октодереве с кешированием (слева) и октодереве с отображением (справа) в зависимости от размера узла

Установлено, что при использовании октодеревьев занимаемая облаком точек память оставалась в пределах заданных ограничений. Увеличение объемов кеша положительно влияет на скорость выполнения операций построения и поиска, увеличение максимального количества точек в узле положительно влияет на скорость построения октодеревя и скорость поиска узлов, но отрицательно влияет на скорость поиска индивидуальных точек.

Произведено сравнение с существующими библиотеками и программами, выполняющими работу во внешней и/или оперативной памяти (Таблица 3).

Таблица 3 - Сравнение скорости построения и потребляемой памяти для различных реализаций октодерева

Метод обработки	Параметр	Облако точек			
		<i>five.las</i> 0.03 ГБ	<i>polytech.las</i> 1.8 ГБ	<i>smolny.las</i> 4.3 ГБ	<i>molodezhnoe.las</i> 53.4 ГБ
Октодерево на базе системы кеширования	Время, сек	0.06	4.6	18.6	1075
	Память, ГБ	0.0334 (111%)	0.517 (29%)	0.586 (13%)	0.986 (1.8%)
Октодерево на базе отображения памяти	Время, сек	0.13	3.8	10	1043
	Память, ГБ	0.0334 (5%)	0.004 (<1%)	0.014 (<1%)	0.180 (<1%)
Point Cloud Library (ОП)	Время, сек	0.122	4.6	12.5	Нехватка памяти
	Память, ГБ	0.0066 (22%)	1.8 (100%)	4.3 (100%)	Нехватка памяти
Point Cloud Library (ВП)	Время, сек	1.5	32.8	152.3	Много файлов
	Память, ГБ	0.0121 (40%)	0.131 (7%)	0.420 (10%)	Много файлов
Cloud Compare	Время, сек	0.435	18.2	58.5	Нехватка памяти
	Память, ГБ	0.114 (380%)	1.6 (88%)	3.6 (83%)	Нехватка памяти

Из результатов сравнения видно, что потребление памяти (указано в процентах от размера облака точек) у предложенных в работе методов значительно меньше, чем у рассмотренных аналогов, работающих в оперативной памяти, а скорость обработки сопоставима или превосходит.

Экспериментально доказано увеличение производительности при сокращении количества создаваемых файлов в процессе обработки. Для этого было выполнено сравнение с подходом, при котором каждый узел октодерева хранится в отдельном файле. Результаты сравнения показали, что взаимодействие с внешней памятью при помощи динамической аллокации на отображаемом файле быстрее подходов, использующих множество файлов. Из рассмотренных файловых систем лучше всего себя зарекомендовала Ext4.

Экспериментально доказано сокращение потребляемой ОП при сохранении приемлемой производительности при внедрении разработанного метода на основе механизма отображения в существующие программные библиотеки. Эксперимент был поставлен путем сравнения потребления памяти и скорости выполнения на различных этапах алгоритма при работе с использованием стандартного системного аллокатора (т.е. в оперативной памяти) и с использованием аллокатора на отображаемой памяти (т.е. в файловой памяти) на облаке точек *factory.las*. Результатом было снижение скорости обработки на 6% при использовании внешней памяти. Для исследования возможности обработки облаков точек, чей размер превышает доступные объемы оперативной памяти, было использовано облако точек *molodezhnoe.las* размером 53 ГБ. На рисунке 13 приведены графики потребления файловой и оперативной памяти при обработке такого облака.

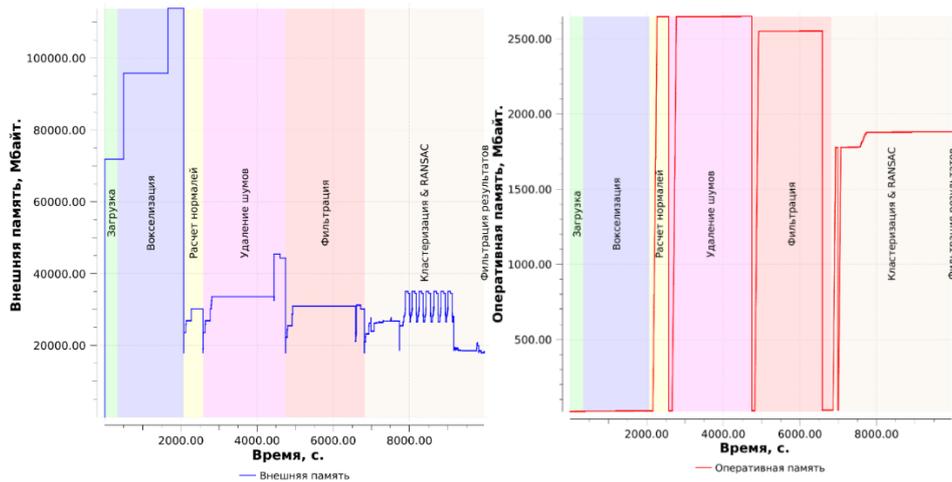


Рисунок 7 - Потребление файловой (слева) и оперативной (справа) памяти при обработке облака точек *molodezhnoe.las*

Проведенные эксперименты показали преимущества предложенных в работе методов перед рассматриваемыми альтернативами, выполняющими обработку в оперативной и внешней памяти, а именно:

- Существенное сокращение потребляемой оперативной памяти (до 99% от размера исходного облака точек, см. табл. 3).
- Низкое падение производительности в сравнении с альтернативами, выполняющими обработку в ОП (от 0% до 48% времени обработки, см. табл. 3 и разд. 4.7).
- Увеличение производительность в сравнении с альтернативами, выполняющими обработку в ВП (время обработки до 6.5% от альтернатив, см. табл. 3).

В **заключении** сформулированы основные итоги, представленные теоретическими и практическими результатами и выводами, полученными в ходе диссертационного исследования.

## ЗАКЛЮЧЕНИЕ

В работе решена задача снижения затрат времени на обмен с внешней памятью за счет организации хранения октодеревя облака точек в оперативной и внешней памяти и организации управления обработкой информации с учетом особенностей структуры октодеревя. При значительном сокращении потребляемой оперативной памяти (до 1% от размера исходного облака точек) обеспечивается незначительное увеличение времени обработки в сравнении с альтернативами, выполняющими обработку в ОП (от 0% до 48%), значительное снижение времени обработки в сравнении с альтернативами, выполняющими обработку в ВП (время обработки до 6.5% от альтернатив). Был применен системный подход для выявления компонентов системы обработки больших облаков точек, определения связей между ними, их функций и целей, механизмов управления ресурсами системы и объектно-ориентированный подход для создания необходимых абстракций и интерфейсов в процессе реализации. Совокупность разработанной модели, методов и алгоритмов, а также их практическая реализация представляют собой решение ак-

туальной научно-технической задачи обработки больших облаков точек при условии наличия ограничений на потребление оперативной памяти и позволяют снизить требования к вычислительным ресурсам системы. В ходе решения данной задачи были получены следующие результаты, составляющие итоги исследования:

1. Рассмотрены проблемы обработки и формирования октодерев, обоснована актуальность и проведен системный анализ проблемы формирования октодерев по облаку точек ЛС при использовании внешней памяти. Сформирована гипотеза об уменьшении затрат времени за счет предлагаемых способов размещения октодерев и организации взаимодействия между оперативной и внешней памятью.

2. Произведена постановка задачи снижения затрат времени на обмен с внешней памятью за счет организации хранения октодерев облака точек в оперативной и внешней памяти и организации управления обработкой информации с учетом особенностей структуры октодерев. Определены критерии эффективности реализации вычислительного процесса обработки, построены концептуальные модели организации обработки облака точек, формирования октодерев, компонентов вычислительного процесса обработки облака точек во внешней памяти. Разработаны модели вычислительного процесса обработки облака точек, анализ которых позволил предложить ряд модификаций, повышающих эффективность процесса обработки.

3. На основании выдвинутых гипотез об организации вычислительного процесса и взаимодействия между оперативной и внешней памятью разработаны методы и алгоритмы решения задачи организации управления вычислительным процессом обработки данных облака точек во внешней памяти, основанные на применении асинхронной системы кеширования и механизма отображения памяти. Разработаны иерархические структуры данных октодерев, включающие реализацию на целочисленной арифметике и на базе целочисленного идентификатора.

4. Разработано программное обеспечение для выполнения обработки облаков точек ЛС с применением предложенных алгоритмов. Предложен способ обработки больших облаков точек путем внедрения системы аллокации на отображаемой памяти в сторонние библиотеки для linux систем. Разработано программное обеспечение для обработки больших облаков точек с использованием сторонней библиотеки совместно с предложенными алгоритмами.

5. Произведенное сравнение потребления оперативной памяти и скорости обработки с альтернативами, использующими оперативную и внешнюю память, показало преимущества предложенных методов. Экспериментально доказано повышение производительности при сокращении количества файлов, сокращения потребляемой памяти при сохранении производительности при внедрении в существующие библиотеки. Разработана методика выбора параметров октодерев, необходимых для разработки программного обеспечения.

**Рекомендации и перспективы дальнейшей разработки темы.** Совершенствование иерархической структуры октодерев с целью уменьшения ее объема и увеличения скорости поиска, а также обеспечения адаптации для большего количества вариаций входных данных, совершенствование механизмов хранения дан-

ных с целью ускорения процедур выборки и сохранения данных. Обеспечение модульности для получения возможности тестирования и замены отдельных компонентов октодеревя. Исследование различных политик кеширования. Использование алгоритмов сжатия данных для октодеревя на основе системы кеширования.

**Соответствие паспорту специальности.** Данное диссертационное исследование выполнено в соответствии с паспортом специальности 05.13.01 «Системный анализ, управление и обработка информации (технические системы)».

Диссертационное исследование соответствует следующим областям (номера соответствуют пунктам в паспорте специальности): п. 4 – Разработка **методов и алгоритмов** решения задач системного анализа, оптимизации, управления, принятия решений и **обработки информации**; п. 5 – Разработка **специального математического и алгоритмического обеспечения** систем анализа, оптимизации, управления, принятия решений и **обработки информации**; п. 12 – **Визуализация, трансформация и анализ информации** на основе **компьютерных методов** обработки информации.

### СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

#### ВАК:

1. Беляевский К.О. Формирование октодеревя по облаку точек при ограничении объема оперативной памяти // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2019. Т. 12. № 4. С. 97–110.

2. Беляевский К.О. Применение динамической аллокации на отображаемой памяти для обработки больших облаков точек в библиотеке PCL // Известия Самарского научного центра Российской академии наук. 2020. Т. 22, № 1. С. 56–64.

#### РИНЦ:

1. Баденко В.Л., Беляевский К.О., Волгин Д.Ю., Зотов Д.К., Федотов А.А. Гибридный подход к 3D реконструкции сложных промышленных объектов из облака точек // Мягкие вычисления и измерения: сборник трудов XXI Международной конференции, 2018. Т. 1. С. 308-311.

2. Беляевский К.О., Болсуновская М.В. Использование целочисленной арифметики для формирования октодеревя // Неделя науки СПбПУ. 2019. Т. 1. С. 122-125.

3. Беляевский К.О., Болсуновская М.В. Использование механизма отображения памяти при формировании октодеревя облака точек // Неделя науки СПбПУ. 2019. Т. 1. С. 125-128.

#### Scopus:

1. Bolsunovskaya M., Tammsaar S., Beliaevskii K., Fedotov A., Gintciak A. Experimental sample of a software module for processing of a cloud of laser scanning points for natural-technical systems development // Proceedings of the 33rd International Business Information Management Association Conference, IBIMA 2019: Education Excellence and Innovation Management through Vision 2020, 2019. Pp. 8619-8627.

2. Badenko V., Tammsaar S., Beliaevskii K., Fedotov A., Vinogradov K. Multi-threading in Laser Scanning Data Processing // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11619 LNCS, 2019. Pp. 289-305.

**Результаты интеллектуальной деятельности:**

1. Таммсаар С.В. Программа для создания растровых проекций / С.В. Таммсаар, К.О. Беляевский, Н.С. Чупин. – Свидетельство о государственной регистрации программы для ЭВМ № 2018665901 от 11.12.2018.

2. Таммсаар С.В. Программа для сортировки и фильтрации облака точек / С.В. Таммсаар, К.О. Беляевский, Н.С. Чупин. – Свидетельство о государственной регистрации программы для ЭВМ № 2018666456 от 17.12.2018.

3. Болсуновская М.В. Программа хранения облака точек лазерного сканирования для природно-технических систем / М.В. Болсуновская, А.В. Лексашов, А.М. Гинцяк, К.О. Беляевский. – Свидетельство о государственной регистрации программы для ЭВМ № 2019660673 от 09.08.2019.

4. Болсуновская М.В. Программа визуализации облака точек лазерного сканирования для природно-технических систем / М.В. Болсуновская, А.В. Лексашов, А.М. Гинцяк, К.О. Беляевский. – Свидетельство о государственной регистрации программы для ЭВМ № 2019619801 от 24.07.2019.

5. Таммсаар С.В. Программное средство построения проекций облака точек, полученных от мобильного комплекса лазерного сканирования / С.В. Таммсаар, К.О. Беляевский. – Свидетельство о государственной регистрации программы для ЭВМ № 2015617817 от 22.07.2015.

6. Беляевский К.О. Программа для гибридной обработки облака точек / К.О. Беляевский, С.В. Таммсаар. – Свидетельство о государственной регистрации программы для ЭВМ № 2019665354 от 22.11.2019.

7. Беляевский К.О. Программа для визуализации облака точек / К.О. Беляевский, С.В. Таммсаар. – Свидетельство о государственной регистрации программы для ЭВМ № 2019665556 от 25.11.2019.