

Шуткин Василий Николаевич

**Метод иерархических динамических уровней детализации для рендеринга
больших трехмерных сцен с детерминированной динамикой**

Специальность 2.3.5 — Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В.П. Иванникова Российской академии наук.

Научный руководитель: **Семенов Виталий Адольфович**, доктор физико-математических наук, профессор

Официальные оппоненты: **Конопацкий Евгений Викторович**, доктор технических наук, доцент, проректор по научной работе и цифровому развитию Федерального государственного бюджетного образовательного учреждения высшего образования «Нижегородский государственный архитектурно-строительный университет»

Фролов Владимир Александрович, кандидат физико-математических наук, старший научный сотрудник отдела компьютерной графики и вычислительной оптики Федерального государственного учреждения «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук»

Ведущая организация: Федеральное автономное учреждение «Государственный научно-исследовательский институт авиационных систем»

Защита состоится 23 апреля 2026 г. в 16:00 на заседании диссертационного совета 24.1.120.01 при Федеральном государственном бюджетном учреждении науки Институте системного программирования им. В.П. Иванникова Российской академии наук по адресу: 115035, г. Москва, ул. Садовническая, д. 41, ст. 2.

С диссертацией можно ознакомиться в библиотеке и на сайте Федерального государственного бюджетного учреждения науки Института системного программирования им. В.П. Иванникова Российской академии наук.

Автореферат разослан « ____ » _____ 2026 г.

Ученый секретарь

диссертационного совета 24.1.120.01,
кандидат физико-математических наук

Турдаков Д.Ю.

Общая характеристика работы

Актуальность исследования. Рендеринг сложных трехмерных сцен является одной из ключевых проблем компьютерной графики. Он возникает в таких предметных областях, как научная визуализация, автоматизация проектирования, инженерии и производства (CAD/CAM/CAE), цифровые двойники, визуальное моделирование зданий, сооружений и городской инфраструктуры, геоинформационные системы, компьютерные игры и анимация, виртуальная и дополненная реальность. Сложность сцен определяется множеством факторов: числом объектов с индивидуальными геометрическими моделями, типизацией и способами параметризации аналитических моделей, степенью детализации приближенных моделей (полигональных сеток, облаков точек, гауссовых сплаттов), реалистичностью применяемых материалов и текстур, композиционным представлением сцены и объектов, смешанным и разноскоростным характером динамики объектов. По мере роста производительности вычислительной техники, графических процессоров и кластеров, растут ожидания разработчиков прикладного программного обеспечения в отношении возможного увеличения сложности трехмерных сцен, которые могут быть отображены с требуемой частотой кадров и надлежащим визуальным качеством.

Для этих целей обычно применяют комбинации методов и техник оптимизации, среди которых следует выделить отсечение конусом видимости (англ. frustum culling), отсечение окклюзии (англ. occlusion culling), выделение значимых объектов, упрощение геометрии и применение уровней детализации (англ. level of detail, LOD), оптимизацию графического конвейера и балансировку вычислительной нагрузки между CPU и GPU. Иерархические уровни детализации (англ. hierarchical level of detail, HLOD) являются одним из наиболее перспективных методов оптимизации рендеринга сложных сцен благодаря масштабируемости по отношению к числу и сложности объектов (и возможности хранения представления сцены во внешней памяти). Однако данный метод не применим для динамических сцен, поскольку подготовка уровней детализации требует значительных вычислительных ресурсов, а их перманентное обновление в ходе событийного моделирования сцены неизбежно приводит к замедлению процессов рендеринга. Актуальным, в связи с этим, представляется создание методов, обладающих достоинствами иерархических уровней детализации HLOD, но применимых к динамическим сценам в практически важных классах графических приложений.

Целью работы является разработка и исследование методов рендеринга больших динамических трёхмерных сцен с поддержкой консервативного (гарантирующего заданную пространственную и временную точность) и интерактивного (стремящегося обеспечить

максимально возможную частоту генерации изображений при приемлемом уровне реализма) режимов отображения в индустриально значимых приложениях.

Основные задачи, решаемые в работе:

- Анализ существующих методов оптимизации рендеринга сложных сцен и исследование возможностей обобщения методов уровней детализации на динамический случай;
- Разработка и исследование метода уровней детализации для выбранного класса динамических полигональных сцен в основной и внешней памяти;
- Разработка и исследование алгоритмов консервативного и интерактивного рендеринга уровней детализации во внешней памяти;
- Программная реализация предложенного метода и разработанных алгоритмов для вычисления и рендеринга уровней детализации;
- Апробация разработанных метода и алгоритмов в результате проведения вычислительных экспериментов на синтетических и реальных индустриальных сценах и выработка практических рекомендаций.

Научная новизна:

- Выделен класс полигональных сцен с детерминированным дискретно-непрерывным характером динамики, допускающий конструктивное обобщение методов уровней детализации и имеющий важные индустриальные приложения;
- Предложен метод иерархических динамических уровней детализации HDLOD (сокращение от англ. Hierarchical Dynamic Level of Detail) для представления сцен с детерминированной динамикой.
- Предложен алгоритм вычисления уровней детализации HDLOD на основе классификации объектов динамической сцены, многоуровневой кластеризации и серий полигональных упрощений. Вычисленные уровни детализации позволяют оперативно формировать альтернативные упрощенные представления сцены с учетом текущего положения пространственной камеры и временного репера, разрешения устройства отображения и заданной визуальной точности;
- Предложены алгоритмы консервативного и интерактивного рендеринга уровней детализации HDLOD во внешней памяти, которые предусматривают отдельное хранение иерархии уровней и геометрических моделей, а также кэширование моделей в основной и видеопамати с асинхронной загрузкой и вытеснением. Благодаря возможности переключения между консервативным и интерактивным режимами

непосредственно во время исполнения целевого приложения обеспечивается сбалансированное по производительности и качеству отображение сложных сцен;

- Получены теоретические оценки сложности основных алгоритмов, а также проведены серии вычислительных экспериментов, которые подтверждают эффективность разработанного метода HDLOD и возможность его конструктивного применения для широкого класса индустриальных приложений, включая визуальное моделирование строительных проектов и инфраструктурных программ.

Теоретическая важность проведенного исследования определяется выполненной формализацией практически значимого класса полигональных сцен с детерминированным дискретно-непрерывным характером динамики и введенными функциями близости объектов, необходимыми для пространственно-временной кластеризации сцен.

Теоретически важными представляются проведенное обобщение популярного метода иерархических уровней детализации на класс динамических полигональных сцен, разработанные и формально описанные алгоритмы вычисления и рендеринга уровней детализации HDLOD с введенной функцией визуального качества (погрешности), а также полученные теоретические оценки вычислительной сложности разработанных алгоритмов.

Практическая значимость полученных результатов заключается в возможности применения разработанного метода для рендеринга широкого класса сцен, не помещающихся в основную память, видеопамяти и требующих значительных вычислительных ресурсов CPU и GPU для отображения в консервативном и интерактивном режимах просмотра. В частности, метод демонстрирует высокую производительность и масштабируемость для больших полигональных сцен с доминирующими дискретными событиями, характерными для моделей сложных индустриальных проектов и масштабных инфраструктурных программ.

Благодаря достигнутым характеристикам метод нашел применение в ряде прикладных проектов, имеющих целью создание приложений визуального пространственно-временного моделирования индустриальных проектов, градостроительного планирования, управления требованиями и замечаниями в архитектурно-строительной отрасли.

Методология и методы исследования. В работе применены методы компьютерной графики, вычислительной геометрии, линейной алгебры и теории алгоритмов.

Достоверность и обоснованность результатов. Все разработанные методы и алгоритмы были реализованы и прошли экспериментальную апробацию.

Основные положения, выносимые на защиту:

1. Метод иерархических динамических уровней детализации HDLOD для представления и рендеринга сцен с детерминированным характером динамики.

2. Алгоритм вычисления уровней детализации HDLOD во внешней памяти на основе классификации динамических объектов, кластеризации объектов и полигональных упрощений.
3. Алгоритмы консервативного и интерактивного рендеринга HDLOD во внешней памяти.
4. Результаты вычислительных экспериментов и приложения.

Апробация работы. Основные результаты работы докладывались на международных и всероссийских конференциях:

- 1) CGVCVIP 2019: 13th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing 2019 (Porto, Portugal, 16-18 July 2019).
- 2) 26th ISTE International Conference on Transdisciplinary Engineering (Kashiwa Campus, The University of Tokyo, Japan, July 30 - August 1, 2019).
- 3) 29-я Международная конференция по компьютерной графике и машинному зрению ГрафиКон 2019 (23–26 сентября 2019, БГТУ, г. Брянск, Россия).
- 4) 31-я Международная конференция по компьютерной графике и машинному зрению ГрафиКон 2021 (27–30 сентября 2021, НГТУ им. Р.Е. Алексеева, г. Нижний Новгород, Россия).
- 5) 35-я Международная конференция ГрафиКон 2025 (30 сентября – 2 октября 2025 г., Поволжский государственный технологический университет, г. Йошкар-Ола, Россия).

Публикации. Основные результаты по теме диссертации изложены в 11 печатных изданиях [1-11], 2 из которых изданы в журналах, рекомендуемых ВАК [2, 3], 2 работы индексируются в Web of Science [1, 2] и 5 — в Scopus [1, 2, 4, 8, 9]. Получено свидетельство о государственной регистрации программы для ЭВМ [12].

Личный вклад автора. Основные результаты диссертации, в том числе положения, выносимые на защиту, получены лично автором. Научным руководителем поставлены задачи и выполнены редакторские правки опубликованных работ. В рамках предложенного В.А. Семеновым подхода к пространственно-временному моделированию индустриальных сцен автором самостоятельно осуществлены разработка, формализация, обоснование и исследование HDLOD метода и связанных с ним алгоритмов вычисления, интерактивного и консервативного рендеринга иерархических динамических уровней детализации [1-9]. Автором лично осуществлена программная реализация основных алгоритмов, включая генератор уровней детализации HDLOD в составе сервиса управления замечаниями [10, 12].

Вместе с соавторами выполнен ряд оптимизаций алгоритмов, а именно: пространственно-временного индексирования [2], отсечений окклюзии на основе окто-деревьев [2], полигональных упрощений на основе анализа видимости [3, 11]. Совместно также подготовлены

тестовые наборы на основе индустриальных моделей в формате IFC и организованы вычислительные эксперименты [2, 5, 11].

Объём и структура работы. Диссертация состоит из введения, семи глав и заключения. Полный объём диссертации составляет 161 страницу, включая 35 рисунков и 20 таблиц. Список литературы содержит 135 наименований.

Содержание работы

Во введении обосновывается актуальность диссертационной работы, определяются цели и задачи исследования, формулируются научная новизна и практическая значимость полученных результатов, а также перечисляются положения, выносимые на защиту.

Первая глава посвящена обзору существующих подходов к рендерингу больших трёхмерных сцен. В разделе 1.1 кратко описываются подходы, связанные удалением невидимых поверхностей, такие как отсечение по области видимости и отсечение окклюзии.

В разделе 1.2 рассматриваются подходы, связанные с упрощением геометрических представлений. Дискретные уровни детализации (раздел 1.2.1) подразумевают создание нескольких альтернативных версий одной модели. Во время визуализации выбирается версия модели со степенью детализации, наиболее подходящей для текущего положения камеры, расстояния до объекта и разрешения экрана. Непрерывные уровни детализации (раздел 1.2.2), в отличие от дискретных, хранят геометрию в специальной структуре данных, используя которую можно извлечь модель с требуемой степенью детализации непосредственно во время рендеринга. Зависимые от камеры уровни детализации (раздел 1.2.3) являются расширением непрерывных уровней детализации и позволяют проводить уточнение модели только в необходимых областях, что позволяет эффективно визуализировать большие или протяжённые объекты. Наконец, иерархические уровни детализации (раздел 1.2.4) позволяют преодолеть проблему большого числа объектов в сцене путем формирования упрощённых представлений для целых групп объектов, преимущественно находящихся в непосредственной близости друг от друга. И хотя иерархические уровни детализации являются наилучшим решением для больших статических сцен, они оказываются неэффективными в динамических сценах, поскольку требуют постоянных пересчётов при изменениях в сцене.

В разделе 1.3 рассматриваются методы полигонального упрощения. Целью упрощения является сокращение количества полигонов в модели при сохранении внешнего вида оригинальной модели насколько это возможно. Упрощение полигональных сеток путём итеративного стягивания рёбер (раздел 1.3.1) является одним из наиболее распространённых

способов. При стягивании ребра происходит вырождение и удаление треугольников, опирающихся на это ребро. Преимущество этой операции заключается в том, что при соблюдении некоторых правил она может гарантировать сохранение топологии полигональной сетки. Вместо стягивания рёбер можно использовать стягивание произвольных пар вершин (раздел 1.3.2). Стягивание соседних, но несвязанных ребром вершин позволяет закрыть отверстия и туннели с улучшением качества итогового представления. Стягивание треугольника (раздел 1.3.3), удаление вершины (раздел 1.3.4) и объединение полигонов (раздел 1.3.5) являются другими способами итеративного упрощения полигональных сеток. Пространственная кластеризация вершин (раздел 1.3.6) лежит в основе другого подхода: соседние вершины объединяются в один кластер и заменяются на одну новую вершину. После подмены вершин вырожденные и дублированные рёбра и треугольники удаляются, что приводит к упрощению полигональной сетки. Как правило, для формирования кластеров используется регулярная пространственная сетка. Итоговая степень полигонального упрощения в таком случае зависит от разрешения регулярной сетки. В разделе 1.3.7 обсуждаются вспомогательные техники, такие как оболочки для контроля точности, карты нормалей для улучшения внешнего вида модели, а также упрощения на основе пакетного рендеринга. Раздел 1.3.8 посвящён формированию серий уровней детализации в ходе упрощения, а в разделе 1.3.9 обсуждаются возможные способы оценки погрешности упрощенных полигональных моделей.

Раздел 1.4 посвящён важной проблеме рендеринга больших сцен во внешней памяти. Нехватка основной или видеопамати является главным препятствием при визуализации больших сцен. Для его преодоления было разработано множество внешних алгоритмов (англ. external algorithm, out-of-core algorithm). Далее будем называть задачу рендеринга в условиях ограниченной памяти внешним рендерингом (по аналогии с англоязычным термином out-of-core rendering). В разделе 1.4.1 обсуждаются способы организации и хранения геометрического представления сцены для внешнего рендеринга, а также политики загрузки данных и освобождения памяти. Особое внимание уделяется технике спекулятивной опережающей загрузки (англ. prefetching), предназначенной для оценки востребованности данных и их заблаговременном размещении в основной или видеопамати. В разделе 1.4.2 представлен обзор литературы по теме рендеринга больших сцен с использованием уровней детализации.

Раздел 1.5 посвящён специальным методам для рендеринга зданий и городской инфраструктуры, а в разделе 1.6 обсуждаются попытки применения уровней детализации к частным классам динамических сцен.

Во второй главе приводится формальное описание метода иерархических динамических уровней детализации HDLOD и излагаются основные принципы генерации HDLOD и их рендеринга.

В разделе 2.1 определяется класс сцен с детерминированной динамикой, для которых применим метод HDLOD, и приводится его формальное описание. Пусть сцена $S(t)$ определена в трёхмерном евклидовом пространстве E^3 на моделируемом временном отрезке $t \in [0, T]$ и представлена как множество объектов $s(g_s, p_s(t), m_s(t)) \in S$. Каждый объект имеет неизменное геометрическое представление $g_s \subseteq E^3$. Статус присутствия объекта в сцене определяется соответствующей функцией $p_s(t): [0, T] \rightarrow \{0, 1\}$ таким образом, что $p_s(t) = 1$, если объект s присутствует в сцене в момент времени t , и $p_s(t) = 0$, если отсутствует. Положение объекта определяется функцией положения $m_s(t): [0, T] \rightarrow M$, где M — множество матриц преобразования размерности 4×4 .

Примеры функций присутствия представлены на рис. 1. Сцена данного класса, связанная с пространственно-временным (4D) моделированием процессов строительства стадиона в соответствии с календарно-сетевым графиком проектных работ, представлена на рис 2.

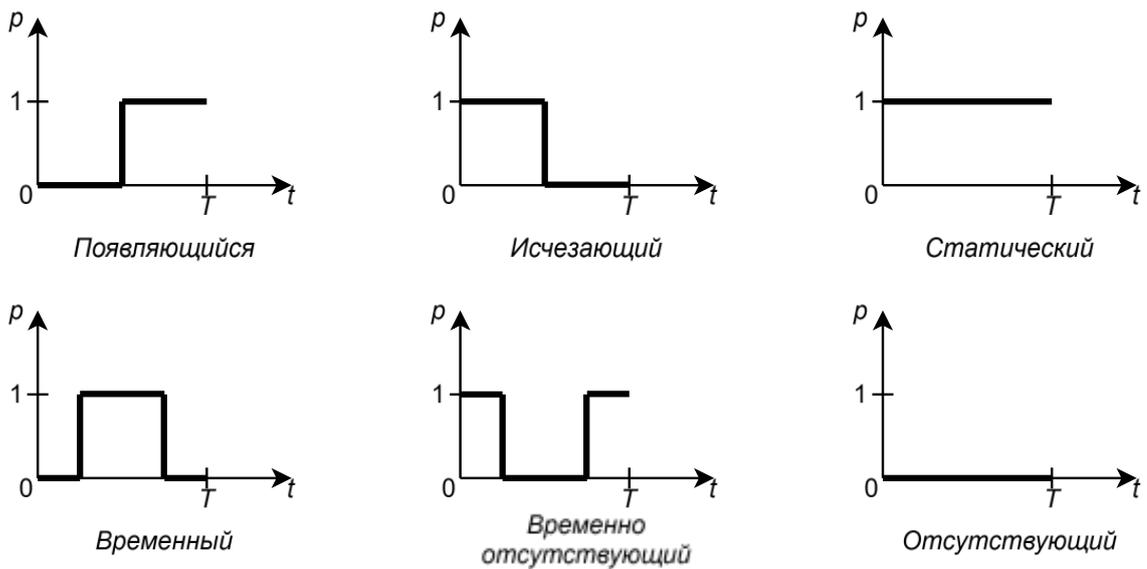


Рисунок 1 — Примеры некоторых типовых функций присутствия объекта.

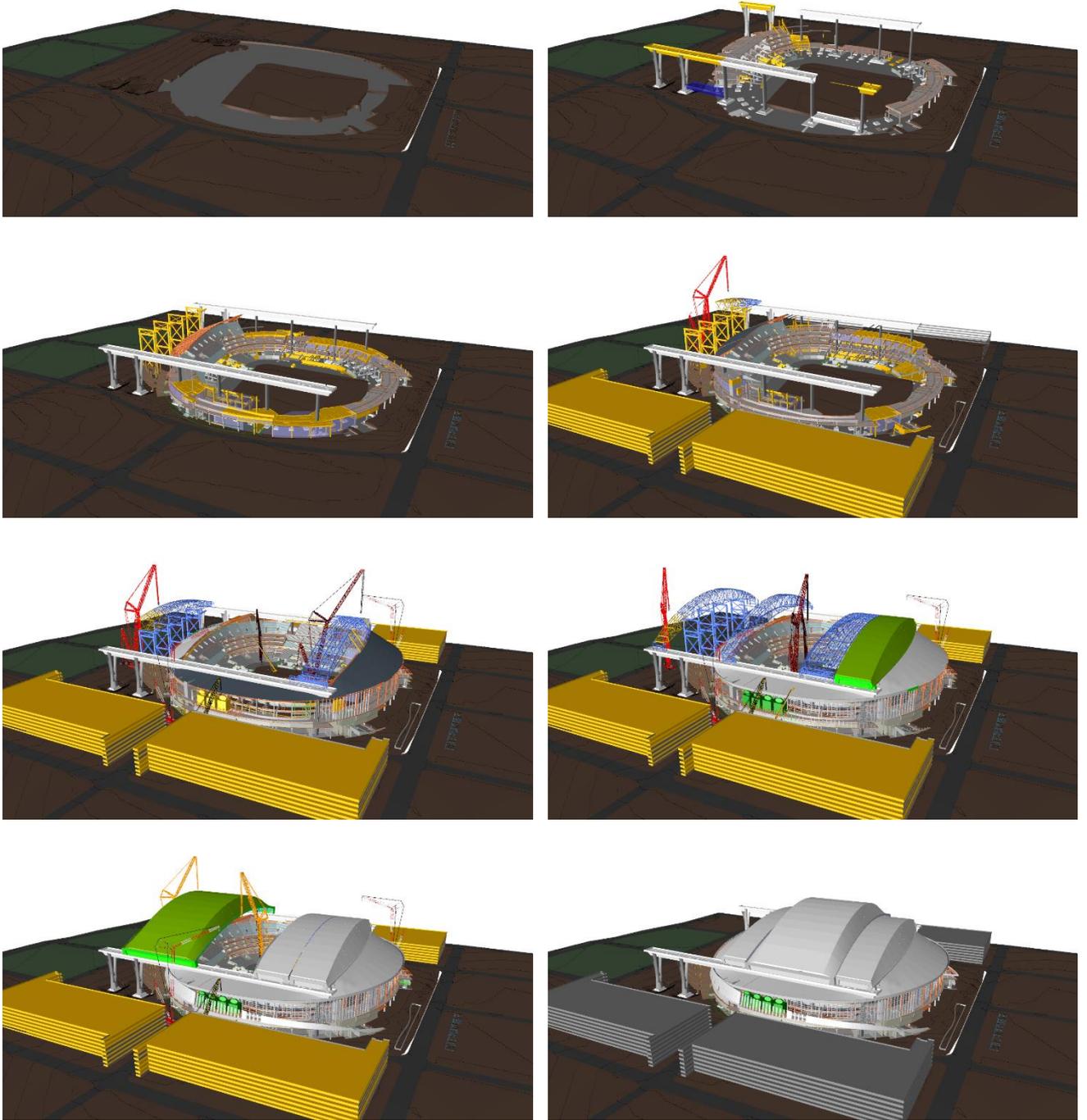


Рисунок 2 — Пример сцены с детерминированной динамикой, моделирующей процессы строительства стадиона в соответствии с календарно-сетевым графиком проектных работ.

По динамическому поведению все объекты сцены могут быть разделены на три класса. В класс *статических* объектов попадают объекты, присутствие и положение которых не изменяется на протяжении всего моделируемого периода, то есть $\forall t \in [0, T] p_s(t) = 1, m_s(t) = I$, где I — единичная матрица. Класс *псевдо-динамических* объектов состоит из объектов, которые появляются и исчезают, однако их положение остаётся неизменным: $\forall t \in [0, T] m_s(t) =$

I , но $\exists t_1, t_2 \in [0, T]$, такие что $p_s(t_1) \neq p_s(t_2)$. Наконец, класс *динамических* объектов состоит из объектов, положение которых меняется с течением времени: $\exists t_1, t_2 \in [0, T]$, такие что $m_s(t_1) \neq m_s(t_2)$. Они могут появляться/исчезать или постоянно присутствовать.

В разделе 2.2 приведено формальное описание иерархических динамических уровней детализации HDLOD как дерева $C = \{c(g_c, p_c(t), m_c(t), b_c(t), \delta_c(t))\} \cup \{<\}$, представленное множеством кластеров c и отношений агломерации $<$ между ними. Отношение агломерации $c' < c$ соответствует связи ребёнок-родитель в дереве, а также отражает тот факт, что родительский кластер $c \in C$ был сформирован из упрощённых геометрического и поведенческого представлений дочернего кластера $c' \in C$. Представление кластера является упрощённым агрегированным представлением его детей. Таким образом, кластеры у корня дерева имеют представления с наименьшим уровнем детализации, а листовые кластеры — с наибольшим. Пример дерева кластеров для представления уровней детализации HDLOD представлен на рисунке 3.

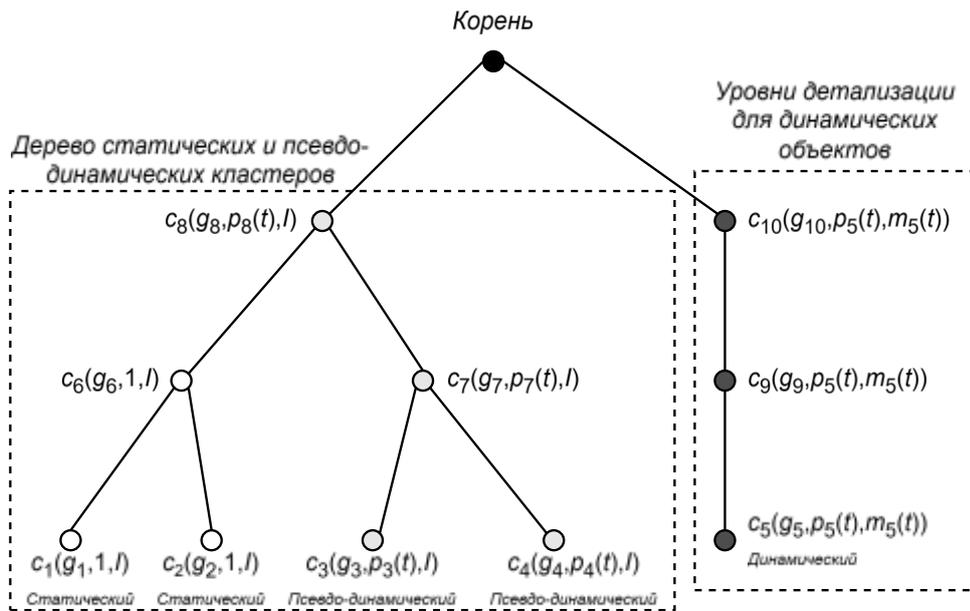


Рисунок 3 — Пример дерева кластеров для представления уровней детализации HDLOD.

Каждый кластер c имеет геометрическое представление g_c , а также функции, описывающие его динамическое поведение, определённые на всём интервале моделирования $[0, T]$. Это функция присутствия $p_c(t): [0, T] \rightarrow \{0, 1\}$, функция положения $m_c(t): [0, T] \rightarrow M$, функция ограничивающего объёма $b_c: [0, T] \rightarrow B$, где B — множество объёмов в E^3 , и функция геометрической погрешности $\delta_c: [0, T] \rightarrow [0, \Delta_c]$, где Δ_c — максимальное локальное отклонение геометрии кластера от геометрии соответствующих ему оригинальных объектов на всём временном отрезке $[0, T]$.

Задачи полигонального упрощения являются хорошо изученными. В то же время, возможностям упрощения динамического поведения объектов в сцене не уделялось должного внимания. Поскольку объекты могут проявлять разное поведение (например, в некоторый момент времени часть объектов может присутствовать в сцене, а другая – отсутствовать), статус присутствия агрегирующего их кластера оказывается неопределенным и, независимо от решения отображать его или нет, неизбежны ошибки. Для описания и контроля такого рода ошибок в работе вводится функция геометрической погрешности кластера $\delta_c(t)$, которая определяет пространственную точность упрощённого представления кластера c по сравнению с исходной геометрией объектов, содержащихся в нем и присутствующих в сцене на момент времени $t \in [0, T]$.

В разделе 2.3 описывается общая схема рендеринга уровней детализации HDLOD и основные преимущества, которые даёт такое представление. Для заданного положения камеры и заданного момента модельного времени $t_m \in [0, T]$ рендеринг сцены осуществляется путем обхода дерева кластеров в глубину, начиная с корневого узла и распространяясь к листовым узлам. При посещении каждого кластера принимается решение о том отображать ли его, пропустить, или посетить более детальные дочерние кластеры.

В первую очередь определяется видимость кластера, для чего выполняются проверки отсечения его ограничивающего объёма $b_c(t_m)$ по области видимости и, опционально, отсечения окклюзии. Если кластер признан невидимым, он и его поддереву исключаются из дальнейшего анализа. В противном случае оценивается качество кластера. Для этого по значению функции погрешности $\delta_c(t)$ определяется экранная погрешность SSE (Screen-Space Error) в пикселях. Для стандартной перспективной проекции она может быть вычислена по формуле:

$$SSE = \frac{\delta_c(t) \cdot R}{2 \cdot d \cdot \tan \frac{\theta}{2}}$$

где R — разрешение экрана в пикселях по высоте, d — расстояние от камеры до кластера, а θ — вертикальный угол раствора пирамиды видимости.

Если экранная погрешность не превышает заданного порогового значения, качество кластера считается приемлемым, и он подлежит дальнейшему анализу. В противном случае осуществляется обход его дочерних кластеров.

Если кластер удовлетворяет критерию качества, осуществляется проверка его статуса присутствия. Если $p_c(t_m) = 1$, то считается, что кластер присутствует в сцене в текущий момент времени и необходимо отобразить его геометрию g_c в положении $m_c(t_m)$. В противном случае кластер считается отсутствующим в сцене и не отображается.

Обход дерева гарантированно завершается в листовых кластерах. Поскольку геометрия листовых кластеров и их поведение заданы точно и их качество всегда приемлемо ($\delta_c(t) = 0$), схема предусматривает отображение всех объектов.

В разделе 2.4 описывается многовариантный метод HDLOD, предназначенный для одновременного отображения альтернативных сценариев без дополнительных вычислительных затрат на пересчет уровней детализации, характерных для основного метода. Это достигается благодаря разделению геометрического и поведенческого представлений и их независимой обработке.

Метод имеет важные приложения. В частности, метод применим для визуального моделирования индустриальных проектов и альтернативных сценариев их реализации.

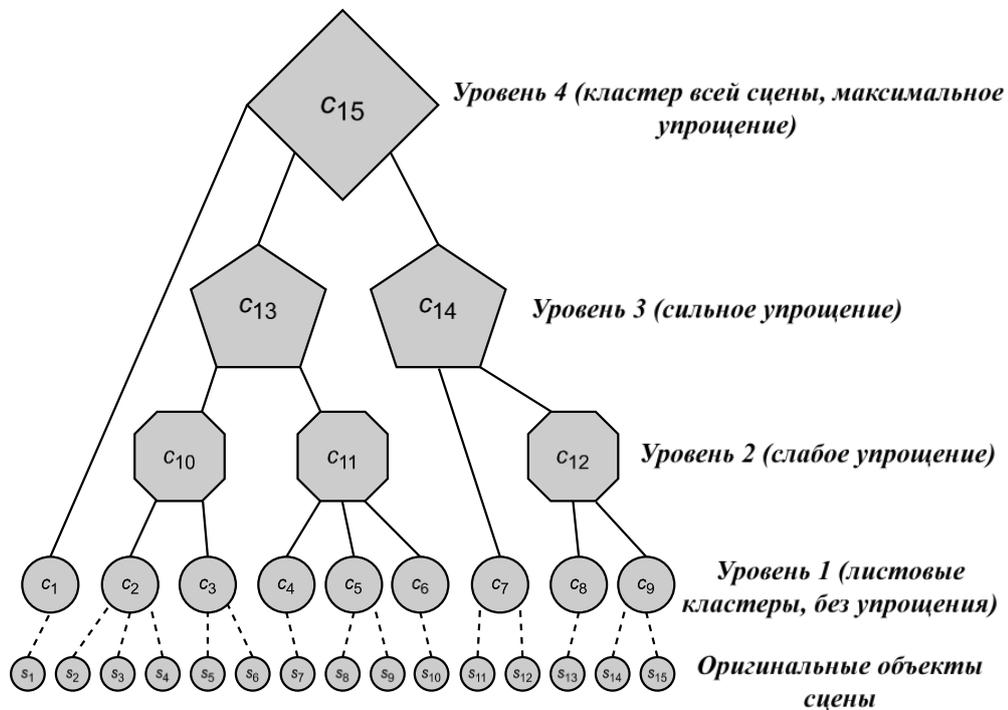


Рисунок 4 — Пример дерева кластеров для HDLOD представления. Объекты сцены $s_1 - s_{15}$ с оригинальной геометрией и динамикой включаются в состав листовых кластеров $c_1 - c_9$ первого уровня. Последние формируют кластеры $c_{10} - c_{12}$ второго уровня, которые в свою очередь — кластеры c_{13}, c_{14} третьего уровня. Вся сцена представлена корневым кластером c_{15} с наибольшим упрощением.

Третья глава посвящена описанию предложенного алгоритма вычисления уровней детализации HDLOD, доказательству его корректности и получению теоретических оценок сложности.

На первом этапе вычисления уровней детализации объекты сцены разделяются на статические, псевдо-динамические и динамические. Для каждого класса объектов применяется свой способ представления и формирования кластеров. Статические и псевдо-динамические объекты включаются в состав листовых кластеров общего дерева, которые в дальнейшем рекурсивно объединяются и формируют кластеры следующих уровней. По мере перехода к более высоким уровням геометрическое и поведенческое представление кластеров становятся все более простыми, так что корневой кластер агрегирует самое упрощённое представление всей сцены (рисунок 4).

Динамические объекты могут иметь сложные траектории движения и переменные скорости и ускорения, что препятствует их эффективной кластеризации. Поэтому для каждого динамического объекта создаётся индивидуальное поддерево кластеров с соответствующей детализацией геометрии. Кинематические характеристики объектов хранятся отдельно и не учитываются при формировании кластеров.

В разделе 3.1 описывается алгоритм построения дерева кластеров. Алгоритм начинает с оригинальных объектов сцены и последовательно объединяет их, формируя на каждом шаге следующий уровень дерева (рисунок 4).

Пусть W — размер всей сцены (например, диагональ ограничивающего параллелепипеда AABB (axis-aligned bounding box) сцены), а L — целевое число уровней дерева. Процесс построения дерева кластеров организован таким образом, чтобы на каждом уровне $l = 1, \dots, L$ сформировать кластеры размером, не превышающим некоторого заданного порогового значения $w(l)$. Для установленных значений должны выполняться следующие соотношения: $w(1) < w(2) < \dots < w(L)$, чтобы обеспечить монотонное увеличение размеров кластеров с переходом на следующие, более высокие уровни, и $w(L) \geq W$, чтобы процесс завершился на уровне L формированием единого корневого кластера для всей сцены. Ограничения по размеру $w(l)$ обеспечивают пространственную близость объединяемых объектов и кластеров. Для оценки меры близости в работе определяется функция пространственной протяжённости объектов множества O с определенными для них ограничивающими объемами $b(o)$:

$$d_S(O) = \left\| \max_{o \in O} p_{max}(b(o)) - \min_{o \in O} p_{min}(b(o)) \right\|$$

где $p_{min}(b)$ и $p_{max}(b)$ — минимальная и максимальная точки ограничивающего объёма.

Чтобы учесть не только пространственный, но и временной фактор, вводится функция временного расхождения объектов множества O , для которых на моделируемом временном отрезке T определены соответствующие функции присутствия:

$$d_T(O) = \int_0^T \left(\max_{o \in O} p_o(t) - \min_{o \in O} p_o(t) \right) dt$$

Временное расхождение объектов контролируется алгоритмом кластеризации аналогично их пространственной удаленности. Пусть Γ — временное расхождение всей сцены. Процесс построения дерева кластеров организован таким образом, чтобы на каждом уровне $l = 1, \dots, L$ сформировать кластеры с временным расхождением, не превышающим определенного порогового значения $\gamma(l)$. Для значений должны выполняться следующие соотношения: $\gamma(1) = 0$ для того, чтобы листовые кластеры включали объекты с точным геометрическим представлением на всем временном отрезке моделирования, $\gamma(1) < \gamma(2) < \dots < \gamma(L)$, чтобы обеспечить монотонное увеличение временного расхождения кластеров и их постепенное упрощение при переходе на следующие уровни, и $\gamma(L) \geq \Gamma$, чтобы процесс завершился на уровне L формированием единого корневого кластера для всей сцены.

Следующая теорема обосновывает корректность алгоритма при соответствующем выборе пороговых значений.

Теорема. Корректность алгоритма построения дерева кластеров. При выборе пороговых значений $w(l) = W/2^{L-l}$ и $\gamma(l) = \frac{l-1}{L-1}\Gamma$, $l = 1, \dots, L$ алгоритм построения дерева кластеров корректно завершается за L шагов.

В разделе 3.2 описывается алгоритм пространственно-временной кластеризации объектов. Кластеризация применяется на каждом уровне $l = 1, \dots, L$ формируемого дерева для объединения соседних объектов, удовлетворяющих пороговым условиям пространственной протяженности $w(l)$ и временного расхождения $\gamma(l)$. Для простоты изложения оригинальные объекты сцены и дочерние кластеры, на основе которых формируются кластеры следующего уровня, далее называются объектами.

Теорема. Сложность алгоритма кластеризации. Пусть n — число объектов на текущем уровне построения дерева кластеров $0 \leq l \leq L - 1$ и k — среднее число соседей, удовлетворяющих пороговому значению $w(l)$. Тогда пространственно-временная кластеризация объектов на уровне l может быть выполнена за $O(n(\log n + k))$ операций.

Теорема устанавливает приемлемую для приложений асимптотическую оценку сложности кластеризации, причем с переходом на каждый следующий уровень формируемого дерева кластеров число объектов n уменьшается, а число соседей k остается ограниченной величиной.

В разделе 3.3 описывается алгоритм вычисления геометрических моделей кластеров во внешней памяти. Алгоритм применяется после того, как дерево кластеров сформировано, и предназначен для наполнения их отображаемыми геометрическими моделями. Поскольку иерархические уровни детализации, в целом, и способ их реализации в виде дерева кластеров, в частности, используют избыточные геометрические данные, алгоритм предусматривает

хранение моделей во внешней памяти и подгрузку в оперативную память по мере необходимости. Вычисление геометрических моделей происходит по уровням дерева строго снизу-вверх из-за агрегативных особенностей метода HDLOD. При этом обеспечивается возможность крупноблочного распараллеливания процесса вычисления моделей на каждом уровне.

В разделе 3.4 описывается алгоритм полигонального упрощения, применяемый при вычислении геометрических моделей. Данная задача критически важна для всего процесса генерации HDLOD представления, поскольку является вычислительно затратной и предполагает эффективный контроль погрешности упрощаемых моделей. Для оценки погрешности в работе применяется метрика Хаусдорфа. Пусть (M, d) — метрическое пространство, A и B — непустые компактные подмножества пространства M . Метрика Хаусдорфа определяется как:

$$H(A, B) = \max(h(A, B), h(B, A)), \text{ где } h(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b)$$

Алгоритм реализует вариант метода стягивания вершин, в котором последовательно обходятся вершины полигональной модели. Для каждой вершины в заданном радиусе ε ищутся соседние вершины, которые затем объединяются с ней. Важным свойством алгоритма является сохранение вершинных атрибутов: при поиске соседей выбираются только вершины с таким же цветом, идентификатором текстуры и идентификатором объекта. Для быстрого поиска соседей алгоритм применяет пространственное индексирование, используя регулярную, равномерную по каждой оси сетку. Размер сетки выбирается, исходя из-за объема доступной памяти, локализации поиска соседей и заполненности ячеек.

Теорема. Корректность алгоритма полигонального упрощения. Алгоритм гарантирует, что геометрическая погрешность полигонального упрощения по метрике Хаусдорфа не превысит заданного параметра ε .

Теорема. Сложность алгоритма полигонального упрощения. Пусть n — число вершин полигональной модели, ε — заданная геометрическая погрешность упрощения, M — ограничение по памяти на предельное число ячеек сетки, K — целевая заполненность ячеек, V — объем сетки. Тогда вычислительная сложность алгоритма составляет $O(n + nK)$ операций, если $n \leq K \min(M, V/\varepsilon^3)$ и $O\left(n + n^2 / \min(M, V/\varepsilon^3)\right)$ операций в противном случае.

В разделе 3.5 приводятся формулы для вычисления характеристик кластеров. Функция присутствия $p_c(t)$ кластера c вычисляется путём усреднения функций дочерних кластеров $c' < c$:

$$p_c(t) = \frac{\sum_{c' < c} w_{c'} p_{c'}(t)}{\sum_{c' < c} w_{c'}}$$

где w — линейный размер кластера. Функция геометрической погрешности $\delta_c(t)$ вычисляется согласно следующей формуле:

$$\delta_c(t) = \begin{cases} \varepsilon_c(2p_c(t) - 1) + (1 - p_c(t)) \cdot 2w_c & \text{при } \frac{1}{2} \leq p_c(t) \leq 1 \\ p_c(t) \cdot 2w_c & \text{при } 0 \leq p_c(t) < \frac{1}{2} \end{cases}$$

где ε_c — геометрическая погрешность модели кластера после упрощения. После вычисления функций погрешности $\delta_c(t)$ для всех кластеров дерева, функции присутствия $p_c(t)$ округляются: $p_c(t): [0, T] \rightarrow \{0, 1\}$.

Раздел 3.6 посвящён важной оптимизации геометрических моделей — удалению внутренних граней, невидимых при обычном размещении камеры снаружи отображаемого кластера. Поскольку в оригинальных сценах индустриальных проектов и моделях кластеров, полученных агломеративным объединением элементов, значительная часть граней оказывается невидимой, эффект подобной оптимизации оказывается значительным. Удаление внутренних граней ускоряет вычисление и рендеринг моделей, сохраняя их внешний вид и снижая требования к оперативной и видеопамяти для их размещения. Отмечаются техники трассировки лучей и пакетного рендеринга, применимые к обсуждаемым задачам упрощения геометрических моделей.

В разделе 3.7 обсуждается подзадача, возникающая при вычислении иерархических уровней детализации для текстурированных моделей: построение текстурных атласов. Приводится общий алгоритм построения текстурного атласа и обсуждаются его ключевые особенности.

В разделе 3.8 предлагается и исследуется другая возможная оптимизация, связанная с инкрементальным обновлением геометрических моделей кластеров при локальных изменениях оригинальной сцены. Подобные изменения характерны для приложений коллективной работы над индустриальными проектами и требуют оперативного обновления уровней детализации HDLOD для визуализации актуальной версии проектных данных. В разделе 3.8.1 обсуждаются общие принципы инкрементального обновления уровней детализации. Их реализация на основе механизма отслеживания изменений и распространения обновлений с использованием временных меток описывается в разделе 3.8.2. Раздел 3.8.3 посвящён оценке характера и степени изменений в сцене. В случае глобальных изменений дерево кластеров строится заново и для них вычисляются все необходимые геометрические модели. В случае локальных изменений удаётся обновить только те кластеры, которые оказались затронуты данными изменениями. Инкрементальное обновление дерева кластера и анализ характера изменений совмещаются в описанных процедурах реакции на соответствующие типы изменений: добавление объектов, удаление объектов, изменение геометрии объектов, изменение поведения объектов.

Четвёртая глава посвящена подробному описанию алгоритмов консервативного и интерактивного рендеринга уровней детализации HDLOD во внешней памяти. В разделе 4.1 рассматривается эффективный способ организации и хранения дерева кластеров, который предусматривает разделение на, так называемый, скелет (структура дерева, которая загружается и постоянно находится в оперативной памяти) и их содержимое (геометрические модели, которые из-за объема данных находятся во внешней памяти и загружаются в видеопамять по мере необходимости).

В разделе 4.2 описывается алгоритм консервативного рендеринга уровней детализации HDLOD во внешней памяти, который преследует цель гарантировать заданную точность отображения сцены. Алгоритм воспроизводит общую схему рендеринга, описанную во второй главе, однако предусматривает кэширование содержимого кластеров при загрузке в видеопамять. Для избежания простоев алгоритм использует двухпоточную обработку: обход дерева кластеров и их анализ осуществляются в главном потоке, а загрузка их содержимого осуществляется асинхронно во втором потоке. Рендеринг осуществляется на графическом процессоре. Для кэша устанавливается ограничение по объему используемой видеопамяти и применяется политика вытеснения LRU (least recently used, вытеснение наиболее давно использовавшихся). Это обеспечивает плавность отображения при типовых пользовательских операциях навигации по сцене.

В разделе 4.3 описывается алгоритм интерактивного рендеринга уровней детализации HDLOD во внешней памяти, предназначенный для высокопроизводительной генерации изображений с приемлемым уровнем качества. Основной принцип алгоритма заключается в том, чтобы не ждать загрузки необходимых кластеров, а воспользоваться альтернативными представлениями, которые уже имеются в кэше. Политика блокирования используемых кластеров обеспечивает, что модели корневого и преимущественно верхних кластеров остаются в кэше и могут использоваться для оперативного рендеринга. Модели нижних кластеров, необходимые для качественного отображения, асинхронно догружаются в кэш в ходе навигации по сцене.

Теорема. Сложность алгоритма выбора кластеров. Пусть n — общее число кластеров в представлении иерархии уровней детализации HDLOD, k — число кластеров, размещённых в кэше, а m — число кластеров, которое требуется загрузить для отображения текущего кадра. Тогда выбор кластеров для интерактивного и консервативного рендеринга в наихудшем случае выполняется за $O(n \log k + mk \log k)$ операций.

В предположении, что обход и анализ дерева кластеров, загрузка и рендеринг моделей кластеров реализуются в параллельных потоках, и время выполнения процессов

пропорционально количеству обрабатываемых кластеров, имеет место следующая оценка затрат на интерактивный и консервативный рендеринг, которая приводится в разделе 4.4.

Оценка времени рендеринга кадра. Пусть n — число кластеров в представлении иерархии уровней детализации HDLOD, которые подлежат обходу и анализу для определения кластеров, отображаемых на текущем кадре. Причем k кластеров, необходимых для отображения, находятся в кэше и m кластеров требуется загрузить. Тогда время рендеринга кадра выражается следующим образом:

$$t_{frame}(n, k, m) = \max(t_{trav}, t_{load}, t_{GPU})$$

$$t_{trav} = nt_{visit} + mt_{cache}$$

$$t_{load} = mt_{read}$$

$$t_{GPU} = \begin{cases} (k + m)t_{rend} + mt_{vload} & \text{if conservative mode} \\ kt_{rend} + mt_{vload} & \text{if interactive mode} \end{cases}$$

где t_{visit} — среднее время анализа кластера при обходе дерева (включая проверки видимости, погрешности, присутствия и наличия в кэше), t_{cache} — среднее время загрузки одного кластера в кэш (включая время записи заголовка кластера и затраты на вытеснение), t_{read} — среднее время чтения модели одного кластера из внешней памяти в основную, t_{vload} — среднее время пересылки модели кластера из основной памяти в видеопамять и t_{rend} — среднее время рендеринга модели одного кластера на графическом процессоре.

Значение n зависит от положения камеры и точности, предъявляемой к отображению сцены. В большинстве случаев оно соотносится с высотой дерева кластеров. Значения k и m существенно зависят от размера кэша, а также от способа навигации по сцене и характера динамики. При интерактивном рендеринге значение m можно ограничить максимальным числом догружаемых кластеров на кадр, что позволяет обеспечить сокращение времени рендеринга кадра по сравнению с консервативным режимом.

В пятой главе описывается программная реализация разработанного метода HDLOD и связанных с ним алгоритмов вычисления и рендеринга уровней детализации. Алгоритмы реализованы в виде библиотеки программ на языке C++ с использованием графического интерфейса API OpenGL для рендеринга на GPU. Библиотека состоит из 8 модулей и насчитывает 19 621 строк кода. Целевая платформа: Microsoft Windows.

Для сохранения уровней детализации на диске используется открытый файловый формат Cesium 3D Tiles. Использование данного формата позволяет обеспечить совместимость со сторонними средствами просмотра, такими как популярные веб-движки CesiumJS и Three.js. Однако формат не предоставляет структуры данных для представления динамических уровней детализации. Чтобы преодолеть это ограничение, в работе предлагается использовать структуры метаданных, предусматриваемые стандартом 3D Tiles.

В разделе 5.1 описывается предлагаемое расширение формата 3D Tiles для сохранения иерархических динамических уровней детализации HDLOD. Раздел 5.1.1 посвящен вспомогательному формату B3DM, применяемому для представления геометрических моделей. В разделах 5.1.2 и 5.1.3 описываются способы представления уровней детализации для оригинального и многовариантного HDLOD метода в стандартном формате 3D Tiles с использованием структур метаданных.

В разделе 5.2 обсуждается программная реализация средств вычисления уровней детализации HDLOD. Описываются основные типы данных и функции. Средства рендеринга уровней детализации HDLOD рассматриваются в разделе 5.3, а в разделе 5.4 описываются средства рендеринга полигональных моделей с использованием графического интерфейса OpenGL. В разделе 5.5 описываются процедуры определения и удаления внутренних граней, используемые для оптимизации полигональных моделей. Раздел 5.6 посвящён разработанным средствам импорта моделей из некоторых популярных 3D-форматов, а также из форматов IFC и CityGML, используемых в приложениях информационного моделирования зданий, сооружений и городской инфраструктуры. В разделе 5.7 кратко описывается консольное приложение, предоставляющее пользовательский интерфейс для вычисления и рендеринга уровней детализации HDLOD для сцен, заданных в одном из поддерживаемых форматов.

Шестая глава посвящена апробации разработанного HDLOD метода и реализованной программной библиотеки в результате проведения серий вычислительных экспериментов на синтетических и реальных сценах, модели которых представлены на рис. 5.

В разделе 6.1 приводятся результаты, связанные с оценкой производительности рендеринга сцен в зависимости от положения камеры. Результаты демонстрируют значительное увеличение производительности рендеринга статических и динамических сцен с использованием HDLOD метода, причем по мере удаления камеры от сцены достигаемый эффект растёт с супер-линейной скоростью (рис. 6). В разделе 6.2 изучается влияние размера кэша для геометрических представлений кластеров на производительность рендеринга. Описанные эксперименты проводились в режиме консервативного рендеринга со специально подготовленными синтетическими сценами и траекториями облёта, чтобы воспроизвести критические ситуации с загрузкой и вытеснением кластеров. Эксперименты проводились для статического сценария (фиксированное модельное время) и динамического сценария (модельное время менялось в процессе облёта). Время рендеринга каждого кадра замерялось в ходе облёта сцены и отображалось в виде соответствующего графика для последующего анализа. Графики позволяют сравнить производительность рендеринга при ограничениях на объем кэша с рендерингом в основной памяти, а также сделать оптимистические оценки консервативного внешнего рендеринга (рис. 7, 8).

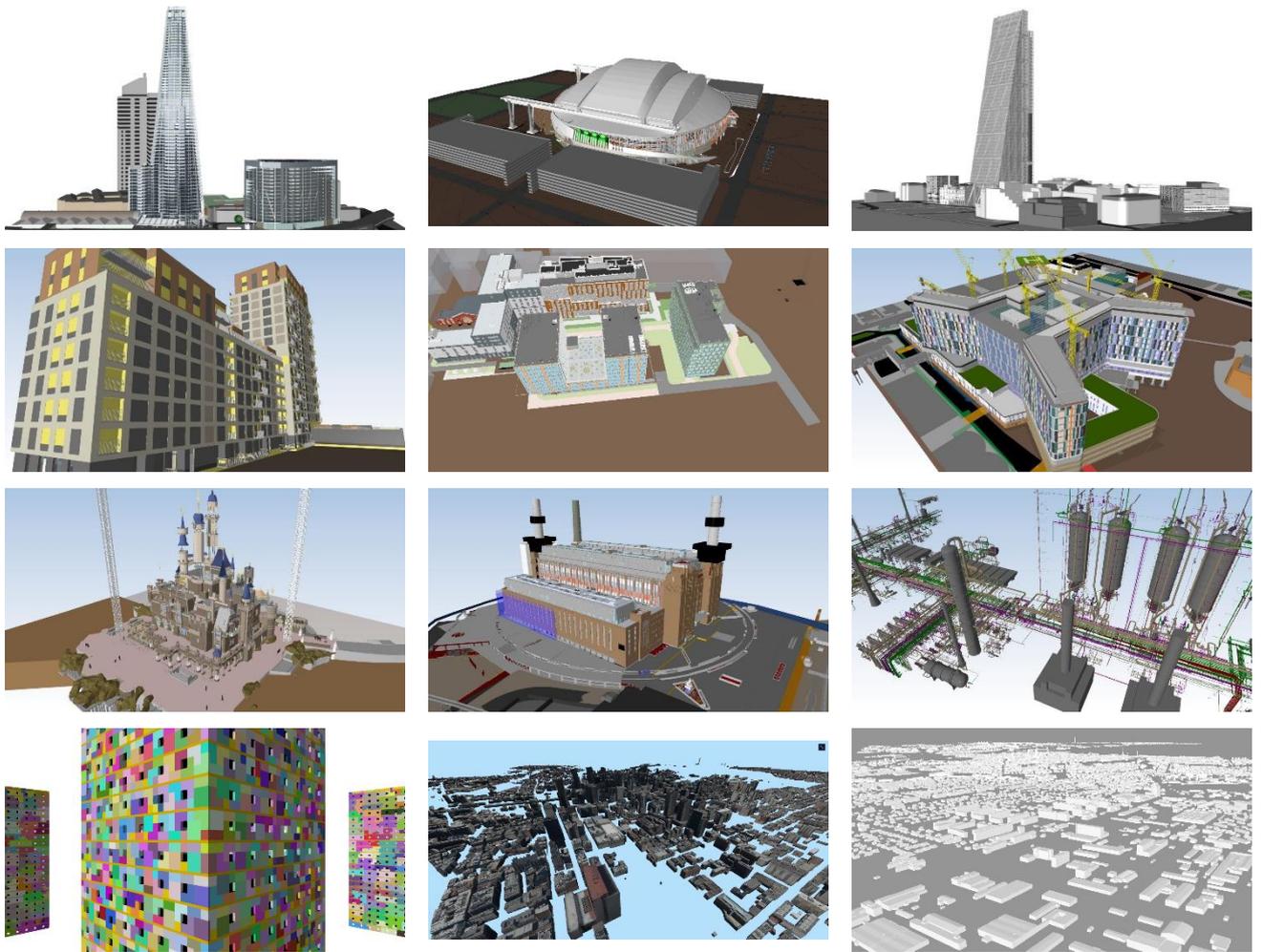


Рисунок 5 — Модели зданий, сооружений и городской инфраструктуры, используемых в экспериментах (слева-направо, сверху-вниз): небоскрёб, стадион, бизнес-центр, жилой дом, отель, клиника, развлекательный комплекс, атомная электростанция, нефтеперерабатывающий завод, синтезированная модель городской застройки, урбанистические модели Монреаля и Таллина.

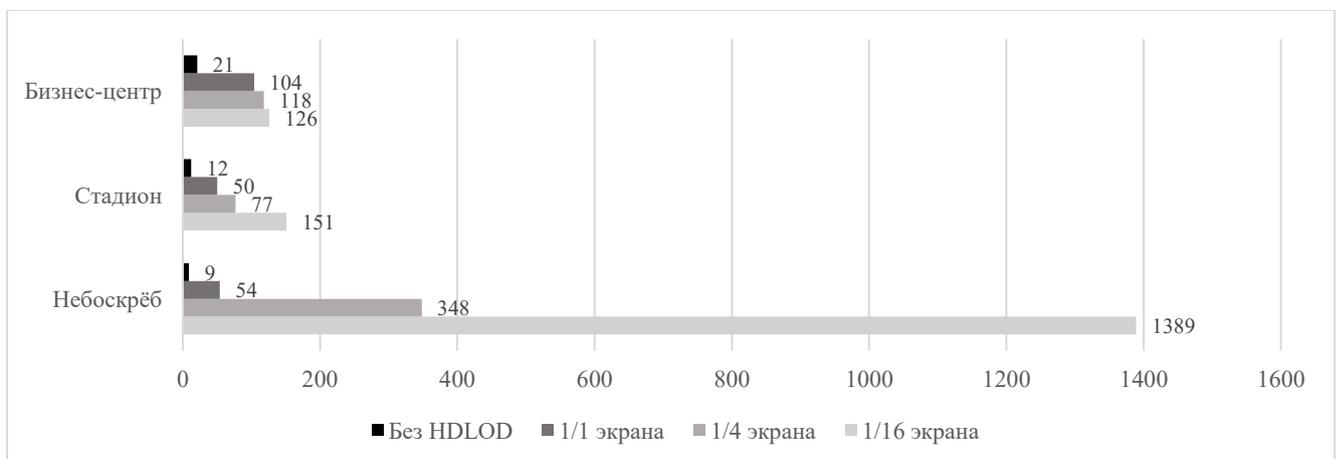


Рисунок 6 — Частота кадров (больше — лучше) при рендеринге моделей с разного расстояния, с использованием и без использования HDLOD метода.

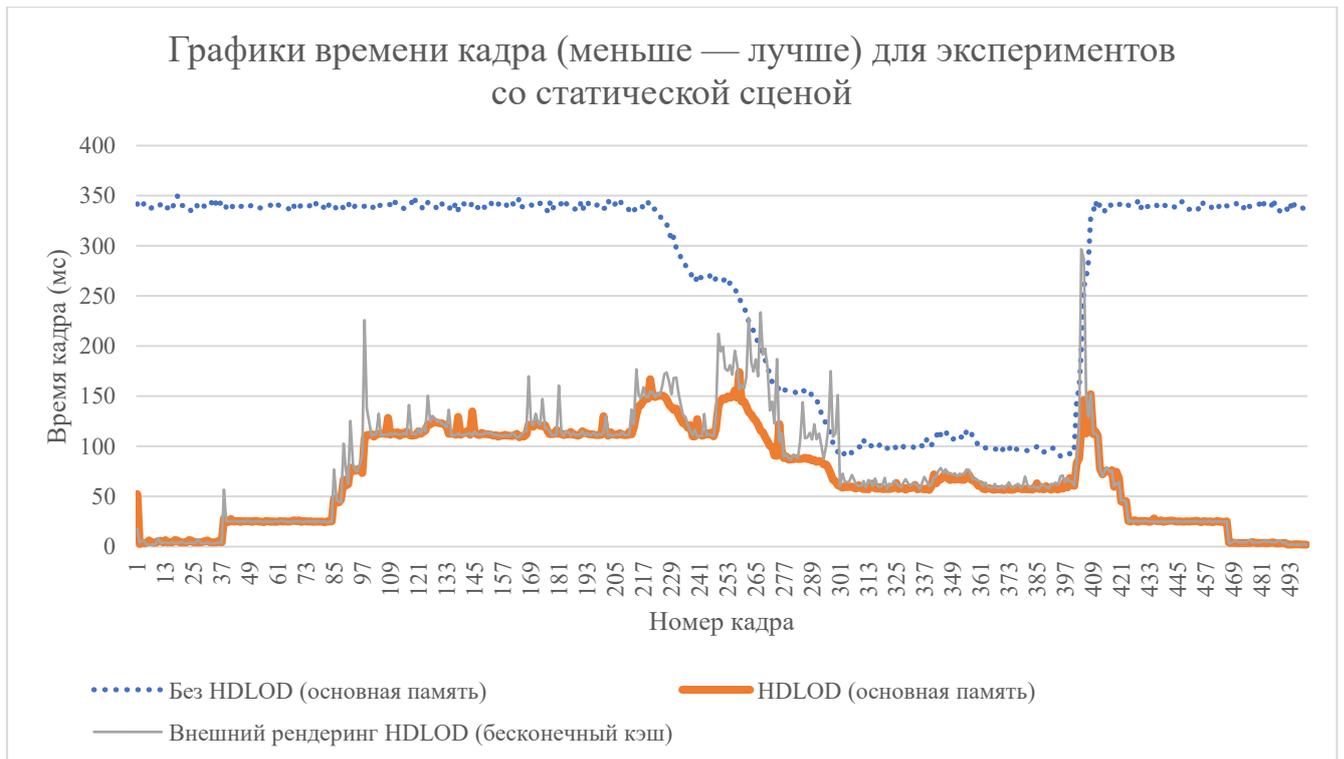


Рисунок 7 — Сравнение производительности рендеринга в основной памяти без HDLOD, с HDLOD и рендеринга HDLOD во внешней памяти с бесконечным размером кэша. Наихудший результат показывает рендеринг без HDLOD. Производительность внешнего рендеринга HDLOD с бесконечным размером кэша сравнима с производительностью HDLOD в основной памяти за исключением скачков времени кадра, обусловленных загрузкой новых кластеров.

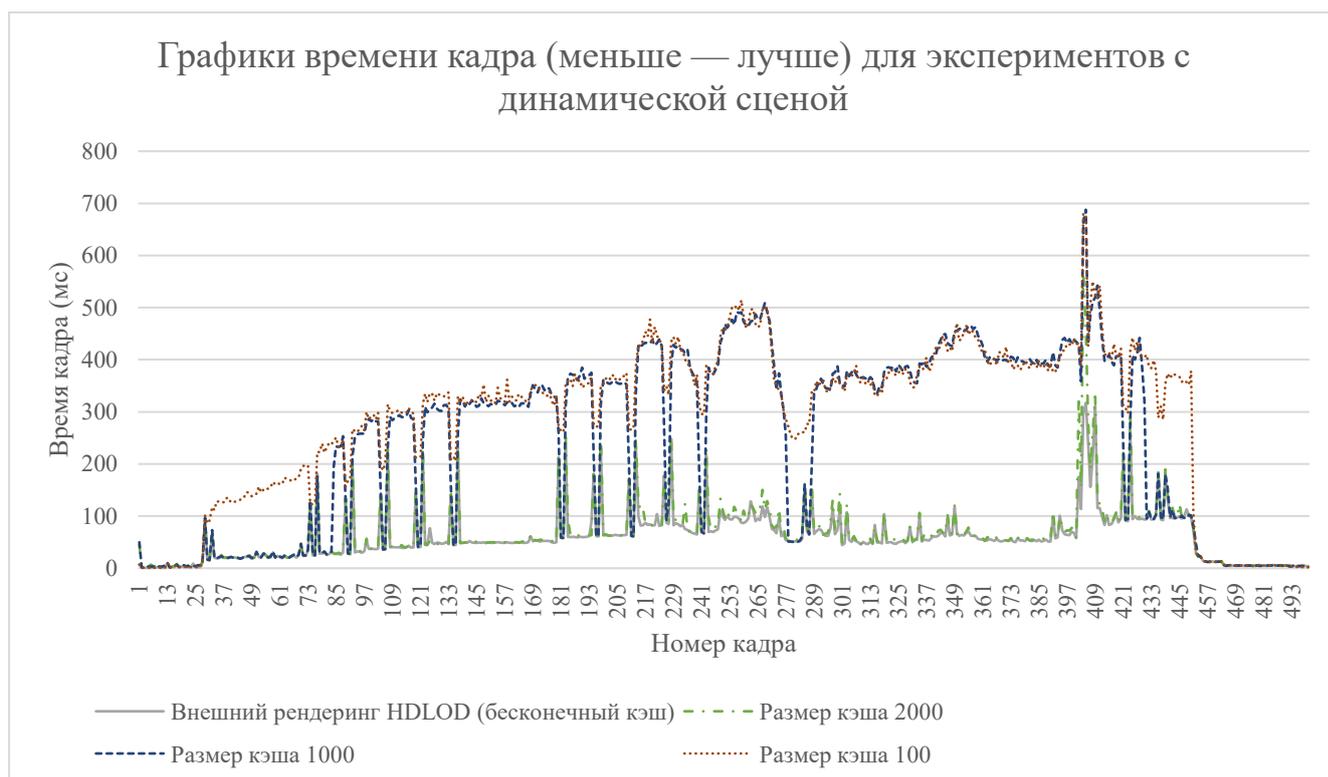
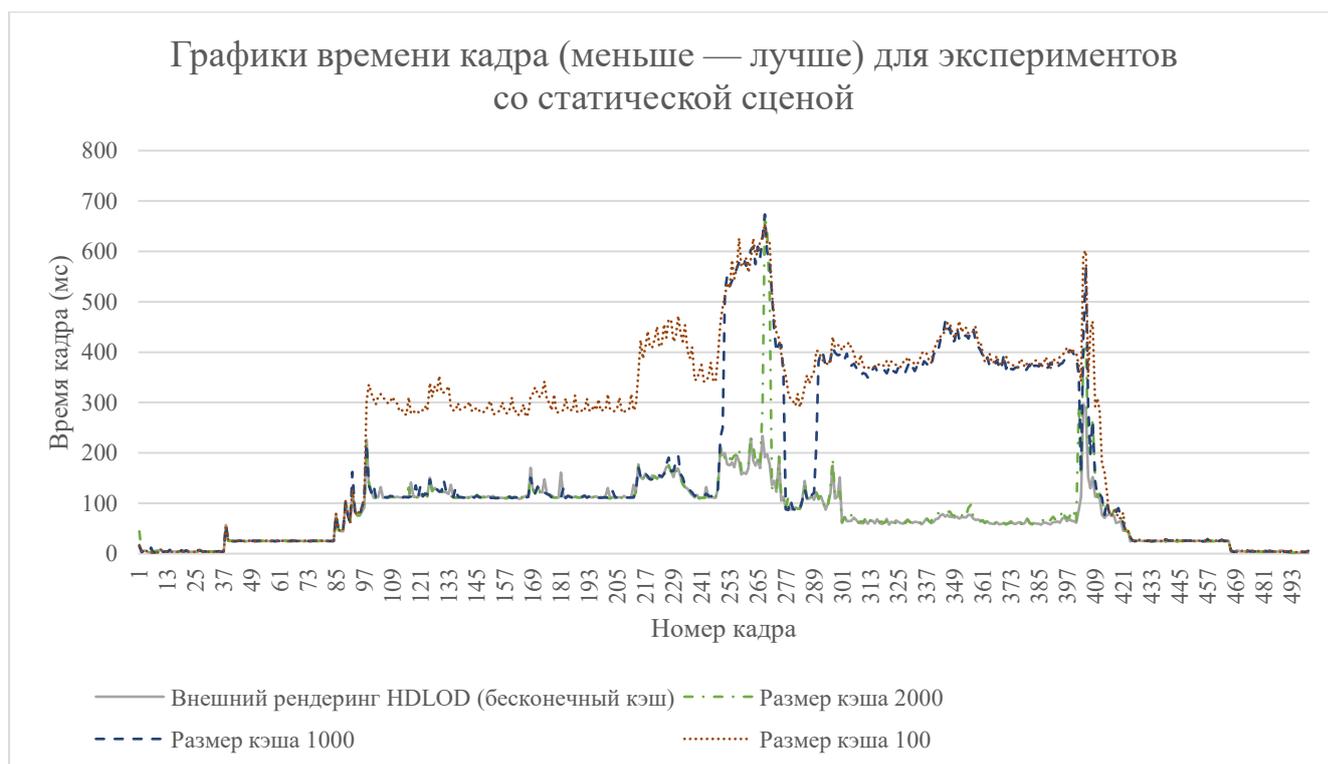


Рисунок 8 — Производительность внешнего рендеринга HDLOD при малых размерах кэша по сравнению с бесконечным размером кэша. Повышение времени кадра соответствует случаям, когда количество кластеров, выбранных для отображения, превышает размер кэша, что приводит к перманентной перезагрузке кластеров.

В разделе 6.3 описываются эксперименты, имеющие целью комплексное сравнение многовариантного HDLOD метода с оригинальным. Оригинальный метод демонстрирует наибольшую производительность во всех экспериментах, в то время как в многовариантном методе дополнительные расходы на рендеринг кластеров приводят к замедлению. Отмечаются важные достоинства многовариантного метода, связанные с отсутствием необходимости повторного вычисления уровней детализации при изменении динамических характеристик объектов (например, при изменении календарно-сетевого графика в приложениях визуального моделирования индустриальных проектов) и низким расходом памяти в случаях, когда динамическая сцена отображается в нескольких окнах с альтернативными характеристиками объектов.

В разделе 6.4 анализируются параметры метода HDLOD, влияющие на производительность рендеринга и скорость вычисления уровней детализации, а также определяющие объёмы данных для их представления. В разделе 6.4.1 обсуждаются альтернативы, связанные со способами кластеризации. Уменьшение и упрощение кластеров приводит к увеличению числа вызовов отрисовки, что негативно сказывается на производительности рендеринга. С другой стороны, слишком крупные и сложные кластеры препятствуют эффективному применению техники отсечения конусом видимости, а также приводят к нерациональному расходу видеопамати из-за включения не всегда отображаемых частей сцены. В разделе 6.4.2 исследуется влияние параметра прогрессии размеров кластеров на производительность рендеринга. В разделе 6.4.3 описываются эксперименты, на основе которых даются рекомендации по выбору параметра относительной погрешности при упрощении полигональных моделей.

В разделе 6.5 приводятся результаты экспериментов, которые позволяют количественно сравнить производительность консервативного и интерактивного рендеринга на больших динамических сценах с реальными индустриальными моделями (таблица 1). Интерактивный режим обеспечивает среднюю частоту кадров в секунду, близкую к типовой частоте развёртки современных мониторов (60 Гц). Консервативный режим несколько отстаёт по этому показателю от интерактивного. Отставание становится значительным, если ориентироваться на другой важный показатель — минимальную частоту кадров в секунду. Это объясняется тем, что в ходе просмотра сцены возникают моменты, когда требуется одномоментно загрузить большое количество кластеров для обеспечения гарантированного качества консервативного рендеринга. При интерактивном рендеринге количество догружаемых кластеров на кадр ограничено, что позволяет обеспечить более равномерную частоту кадров, иногда за счет временного падения качества отображения. По результатам экспериментов делаются следующие рекомендации: интерактивный рендеринг предпочтителен для плавной навигации по сцене с оперативной

реакцией на пользовательские события. Консервативный рендеринг следует использовать для подготовки качественных видео-презентаций и серий изображений в оффлайн режиме.

Таблица 1 — Производительность рендеринга сцен в консервативном и интерактивном режимах.

Сцена	Режим рендеринга	Мин. кадр/с	Сред. кадр/с	Макс. кадр/с	Итого кадров
Клиника	Консервативный	56	59	61	17 990
	Интерактивный	57	59	60	17 997
Развлекательный комплекс	Консервативный	8	52	60	15 867
	Интерактивный	26	53	60	16 040
Электростанция	Консервативный	0	38	61	11 660
	Интерактивный	3	43	61	12 992
Модель Монреаля	Консервативный	0	32	60	10 283
	Интерактивный	38	58	62	17 657
Модель Таллина	Консервативный	40	59	60	17 953
	Интерактивный	59	59	60	17 999

В разделе 6.6 приводятся результаты сравнения реализации метода HDLOD с наиболее популярным коммерческим решением для генерации иерархических уровней детализации Cesium Ion. Поскольку Cesium Ion не поддерживает динамику, сравнительный анализ проводился для статических сцен. При этом оценивались объем сгенерированных файлов (в формате 3D Tiles) и производительность рендеринга (таблица 2).

Таблица 2 — Производительность рендеринга (в кадрах/с) иерархических уровней детализации, полученных по методу HDLOD и с помощью Cesium Ion.

	Клиника	Стадион	Замок	Завод	Отель
HDLOD	36	47	45	68	69
Cesium Ion	44	37	21	37	27

На всех моделях, кроме клиники (самая маленькая модель), метод HDLOD демонстрирует уверенное преимущество.

Раздел 6.7 посвящён результатам проведенных экспериментов с предложенными методами ускорения генерации HDLOD: распараллеливание (6.7.1), удаление внутренних граней (6.7.2) и инкрементальные обновления (6.7.3).

В седьмой главе кратко описываются программные приложения, в составе которых метод HDLOD был успешно реализован и нашел практическое применение. Основное внимание

уделяется функциональному назначению приложений и специальным требованиям к рендерингу сцен, которые удается удовлетворить с помощью разработанного метода. В разделах 7.1, 7.2 приводятся примеры приложений для визуального пространственно-временного моделирования индустриальных проектов и градостроительного планирования. В разделе 7.3 описывается разработанный веб-сервис для управления требованиями и замечаниями в архитектурно-строительной отрасли.

В заключении сформулированы основные результаты работы:

1. Проведен анализ существующих подходов к рендерингу сложных сцен, на основе которого определены возможности обобщения методов уровней детализации для класса полигональных сцен с детерминированным дискретно-непрерывным характером динамики, имеющего важные индустриальные приложения.
2. Предложен и разработан метод иерархических динамических уровней детализации HDLOD для представления и эффективного рендеринга динамических полигональных сцен выделенного класса.
3. Разработаны и исследованы алгоритмы вычисления уровней детализации HDLOD во внешней памяти на основе классификации объектов сцены, многоуровневой кластеризации и серий полигональных упрощений. Вычисленные уровни детализации позволяют оперативно формировать альтернативные упрощенные представления сцены с учетом положения пространственной камеры и временного репера, разрешения устройства отображения и заданной визуальной точности.
4. Разработаны и исследованы алгоритмы консервативного (гарантирующего заданную пространственную и временную точность) и интерактивного (стремящегося обеспечить максимально возможную частоту генерации изображений при приемлемом уровне реализма) рендеринга сцен во внешней памяти. Алгоритмы предусматривают отдельное хранение иерархии уровней и геометрических моделей, а также кэширование моделей в основной и видеопамяти с асинхронной загрузкой и вытеснением. Благодаря возможности переключения между консервативным и интерактивным режимами непосредственно во время исполнения целевого приложения обеспечивается сбалансированное по производительности и качеству отображение сложных сцен.
5. Разработана и исследована многовариантная модификация HDLOD метода для приложений, в которых происходят частые изменения динамических характеристик объектов или требуется одновременное отображение динамической сцены с альтернативными характеристиками объектов в нескольких окнах.

6. Разработанный HDLOD метод и связанные с ним алгоритмы вычисления и рендеринга уровней детализации реализованы в виде библиотеки программ на языке C++ с использованием графического интерфейса OpenGL и предложенного расширения формата 3D Tiles для хранения уровней детализации.
7. Проведена серия вычислительных экспериментов на синтетических и реальных индустриальных сценах, результаты которых подтверждают высокую эффективность разработанных метода и библиотеки и определяют область применения для широкого класса графических приложений.
8. Благодаря достигнутым характеристикам HDLOD метод нашел применение в ряде прикладных проектов, имеющих целью создание приложений визуального пространственно-временного моделирования индустриальных проектов, градостроительного планирования, управления требованиями и замечаниями в архитектурно-строительной отрасли.

Список опубликованных работ

1. Semenov V., Shutkin V., Zolotov V. Visualization of complex industrial products and processes using hierarchical dynamic LODs // *Advances in Transdisciplinary Engineering*, 2019, 10, pp. 655–664 (Q4).
2. Semenov V.A., Shutkin V.N., Zolotov V.A., Morozov S.V., Gonakhchyan V.I. Visualization of Large Scenes with Deterministic Dynamics // *Programming and Computer Software*, 2020, 46(3), pp. 223–232 (Q3).
3. Шуткин В.Н., Морозкин Н.К., Семенов В.А., Тарлапан О.А. Оптимизации генерации иерархических уровней детализации для масштабных полигональных сцен // *Труды Института системного программирования РАН*. – 2025. – Т. 37. – №. 3. – С. 311-324 (K1).
4. Semenov V., Shutkin V., Zolotov V. Faster rendering of large pseudo-dynamic scenes using hierarchical dynamic LODs // *Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conferences on Interfaces and Human Computer Interaction 2019, Game and Entertainment Technologies 2019 and Computer Graphics, Visualization, Computer Vision and Image Processing 2019*, 2019, pp. 275–282.
5. Семёнов В.А., Шуткин В.Н., Золотов В.А., Морозов С.В. Расширение метода иерархических уровней детализации для динамических сцен с детерминированным характером событий // *Труды конференции ГрафиКон-2019*, стр. 37-41, DOI: 10.30987/graphicon-2019-1-37-41.

6. Семенов В.А., Шуткин В.Н., Морозкин Н.К. Эффективный подход к 4D-визуализации масштабных строительных проектов и программ на основе иерархических динамических уровней детализации // Материалы IV Международной научно-практической конференции BIM-моделирование в задачах строительства и архитектуры (BIMAC 2021), 2021, СПб, стр. 385-393. DOI: 10.23968/BIMAC.2021.048.
7. Shutkin V., Morozkin N., Zolotov V., Semenov V. City and building information modelling using IFC standard // ECPPM 2021 – eWork and eBusiness in Architecture, Engineering and Construction: Proceedings of the 13th European Conference on Product & Process Modelling (ECPPM 2021), 2021, Taylor & Francis Group, London, ISBN 978-1-032-0438-9, pp. 406-413, DOI:10.1201/9781003191476-56.
8. V. Semenov, V. Shutkin, V. Zolotov. Conservative Out-of-Core Rendering of Large Dynamic Scenes Using HDLODs // Proceedings of the 31st International Conference on Computer Graphics and Vision (GraphiCon 2021), 2021, Nizhny Novgorod, Russia, pp. 105-115, DOI:10.20948/graphicon-2021-3027-105-115.
9. Shutkin V., Semenov V., Zolotov V., Morozkin N. Alternative HDLOD method for large scenes with multiple dynamic behaviors // Proceedings of 17th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, Porto, Portugal 16 – 18 July 2023, pp. 141-148.
10. Морозов С.В., Шуткин В.Н., Гринченко А.И. Открытые сервисы верификации и валидации цифровых информационных моделей в архитектуре и строительстве // Материалы VIII Международной научно-практической конференции Информационное моделирование в задачах строительства и архитектуры (BIMAC 2025), 2025, СПб, стр. 135-144. DOI: 10.23968/BIMAC.2025.020.
11. Шуткин В.Н., Морозкин Н.К., Семенов В.А., Тарлапан О.А. Эффективная генерация иерархических уровней детализации: параллельная обработка, инкрементальные обновления, полигональные упрощения на основе видимости //Труды конференции ГрафиКон-2025, 2025, Йошкар-Ола, Россия, стр. 261-275, DOI: 10.25686/978-5-8158-2474-4-2025-261-275.

Свидетельства о государственной регистрации программы для ЭВМ

12. Семенов В.А., Аришин С.В., Гринченко А.И., Морозкин Н.К., Тарлапан О.А., Шуткин В.Н., «Сервис управления замечаниями в строительстве», Российская Федерация, Свидетельство о государственной регистрации программы для ЭВМ № 2025668968, дата поступления заявки 18 июня 2025 г., дата государственной регистрации в Реестре программ для ЭВМ 21 июля 2025 г.