

Исследование методов пространственного индексирования динамических сцен на основе регулярных октодеревьев

В.А.Золотов, К.С.Петрищев, В.А.Семенов

vladislav.zolotov@ispras.ru | k_petrishchev@ispras.ru | sem@ispras.ru

Москва, Россия, Институт системного программирования РАН

Статья посвящена развитию и исследованию методов пространственного индексирования и анализа сложных динамических сцен, возникающих в приложениях компьютерной графики, робототехники, анимации, виртуальной и дополненной реальности, САПР, системах nD-моделирования и планирования проектов. Подобные сцены представляются композицией большого числа протяженных геометрических объектов, проявляющих индивидуальное динамическое поведение. Главное внимание в статье уделяется алгоритмам исполнения типовых пространственных запросов с использованием регулярных динамических октодеревьев. В частности, исследуются алгоритмы определения столкновений, выборки по заданной области, поиска ближайшего соседа. Для введенных модельных наборов данных на основе вероятностного анализа выводятся оценки сложности для построения индексов и исполнения типовых запросов в среднем. Полученные оценки существенно улучшают известные пессимистические результаты и служат обоснованием целесообразности применения регулярных октодеревьев для пространственного индексирования масштабных динамических сцен. Результаты проведенных вычислительных экспериментов подтверждают полученные теоретические результаты и иллюстрируют возможности создания эффективных приложений компьютерной графики в условиях перманентно растущей сложности визуальных моделей.

Ключевые слова: компьютерная графика, пространственное индексирование, анализа динамических сцен, регулярные динамические октодеревья.

Octree-based approach to spatial indexing of complex dynamic scenes

V.A.Zolotov, K.S.Petrishchev, V.A.Semenov

Moscow, Russia, Institute for System Programming of the RAS

The paper is dedicated to theoretical research of spatial indexing methods in conformity to three dimensional scenes arising in CAD/CAM systems, robotics, virtual and augmented reality applications. Special attention is given to the decomposition methods based on regular dynamic octrees. A particular version of the octrees is described and investigated to satisfy to the efficiency requirements for typical spatial queries in complex dynamic scenes with the extended borders objects. To perform the needed complexity analysis, the developed octree structure is analyzed against typical spatial queries like collision detection, frustum culling. To obtain more relevant estimates of the complexity on average rather than known pessimistic estimates in worst case, model scene datasets have been introduced. They assume uniform distribution of equal size objects over the scene volume and, being parametric, allow simulation of wide range of industry meaningful scenes. Some auxiliary lemmas about octree depth and distribution of the scene objects over octree levels have been proven based on probabilistic analysis. Final theorem statements provides for the complexity estimates of query evaluation for the model scenes. The obtained results show that asymptotic complexity grows with the object sizes and the scene occupancy. However, the estimates on average improve known results in worst case and can be considered as a theoretical background to introduce the investigated indexing structures to practical applications.

Keywords: computer graphics, spatial indexing, analysis of dynamic scenes, regular dynamic octree.

Введение

Стремительный рост объемов информации, а также необходимость ее анализа приводят к развитию новых подходов к управлению данными и, в частности, методов пространственного индексирования, без которых невозможен быстрый поиск и обработка в геоинформационных базах данных, системах логистического обеспечения, системах автоматизации проектирования, системах управления проектами. Как правило, популярные универсальные и специализированные СУБД предусматривают для этих целей средства пространственного индексирования и поиска. Подобные средства успеш-

но справляются с обработкой статической информации, однако часто не приспособлены для данных, подлежащих перманентным изменениям [1]. Проблемы эффективного поиска и анализа еще более усложняются, когда информация представляет собой не просто массивы точек, а структурированные наборы данных, например, множества объектов с протяженными пространственными границами [6]. Класс подобных приложений чрезвычайно широк и охватывает не только перечисленные выше прикладные области, но и многочисленные системы компьютерной графики, визуализации и анимации. Эти факторы определяют актуальность темы и огромный интерес, как

со стороны научного сообщества, так и производителей системного и прикладного программного обеспечения. В частности, активные разработки в этой области ведут компании Google, Oracle, IBM, Autodesk, Bentley, Intergraph, AVEVA, сталкивающиеся с проблемой увеличения объемов анализируемой пространственно-временной информации.

В нашей работе [8] были системно проанализированы фундаментальные семейства методов индексации и поиска многомерных данных. В частности, рассматривались структуры поиска на интервалах, сбалансированные ветвистые деревья во внешней памяти, бинарные деревья пространственной декомпозиции, префиксные деревья, нерегулярные и регулярные многоуровневые сетки, метрические деревья, а также связанные с ними разнообразные методики хэширования, расщепления и кластеризации.

Большое внимание было уделено методам пространственной индексации на основе регулярных октодеревьев. Главным их достоинством является простота развертывания и модификации индексов, обусловленная априори известным положением секущих плоскостей. Это позволяет относительно легко обновлять индексы и обеспечивать эффективность исполнения типовых пространственных запросов при перманентных изменениях самих данных. Однако данные методы не обеспечивают сбалансированность структур индексов и могут приводить к их деградации в тех случаях, когда данные неравномерно распределены по пространству или имеют протяженные границы. Известные оценки сложности в подобных наихудших случаях приводят к пессимистическим выводам и не определяют реальных границ применимости методов декомпозиции на основе регулярных октодеревьев. По-видимому, они нуждаются в более детальном исследовании и развитии, исходя из оценок сложности в среднем, которые могут быть получены на основе вероятностного анализа основных алгоритмов при необходимой конкретизации условий прикладных задач.

Пространственная декомпозиция на основе октодеревьев

Начнем с анализа классического метода пространственной декомпозиции, основанного на регулярных октодеревьях. В трехмерном случае параллелепипед, пространственно ограничивающий весь набор данных (или в англоязычной литературе AABV — Axis Aligned Bounding Box), разбивается на восемь равных частей плоскостями, перпендикулярными каждой из координатных осей. Процесс рекурсивно применяется до тех пор, пока количество элементов данных в каждом вновь образованном октанте не окажется ниже некоторого предустановленного порога $m > 1$. С результирующим

пространственным разбиением ассоциируется соответствующее дерево. Корень дерева соответствует исходному параллелепипеду, содержащему в себе весь набор данных, а вершины — вложенным октантам, группирующим данные на разных иерархических уровнях пространственной декомпозиции. Пример декомпозиции и полученного октодеревья приведен на рис. 1.

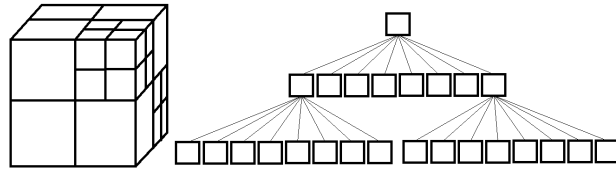


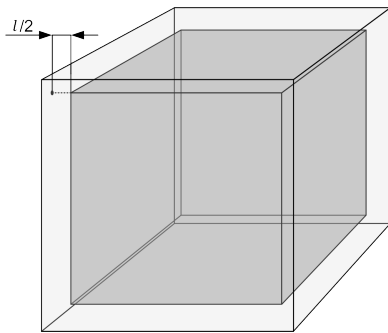
Рис. 1: Пример пространственной декомпозиции и соответствующего ей октодеревья.

Заметим, что способ ассоциирования элементов данных с вершинами октодеревья, в конечном счете, определяется способом их дальнейшей локализации и поэтому зависит от их пространственной семантики. Точечные данные могут быть непосредственно ассоциированы с листовыми октантами. Однако геометрические объекты с протяженными границами могут занимать определенный объем исходного параллелепипеда. Поэтому более рациональным представляется их ассоциирование с теми октантами, в которых они могут быть размещены полностью [5]. Для такого способа есть и иные причины, связанные с возможным пересечением секущими плоскостями даже небольших объектов и невозможностью установить простое соответствие между ними и листовыми октантами.

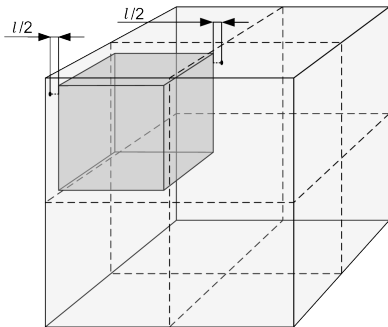
Определение 1. *Октодеревом со строгой многоуровневой локализацией объектов и верхней границей кардинальности ячеек m назовем регулярным и обозначим $Octree(m)$.*

Регулярное октодеревье может использоваться для решения различного рода вычислительных задач, связанных с поиском и анализом пространственно-временных данных. Естественно, что затраты на построение дерева должны сполна компенсироваться более быстрым решением целевых задач. Оценим затраты на построение октодеревья. Заметим, что подобные оценки могут существенно зависеть от специфики прикладных данных и анализ наихудшего случая, как правило, приводит к довольно пессимистическим результатам, не отражающим реальные показатели производительности для большинства приложений. Поэтому получим оценки сложности в среднем на основе вероятностного анализа некоторого упрощенного набора данных.

Для этого рассмотрим набор данных, связанный с визуальным анализом пространственно-трехмерных сцен и допускающий содержательную параметризацию. Пусть сцена представляется набором n трехмерных геометрических объектов, независимо случайным образом расположенных внутри единичного куба. Ограничивающим объемом каждого объекта является куб с характерным размером сторон $0 \leq l \leq 1$. В рамках подобной постановки x, y, z -координаты центров являются независимыми равномерно распределенными величинами на соответствующих отрезках $[l, 1 - l]$. Для обсуждаемых задач анализа сцен будет применяться единая техника пространственной локализации объектов на основе их ограничивающих объемов, поэтому в дальнейшем мы не делаем никаких различий между понятиями геометрического объекта и его ограничивающего параллелепипеда.



(а) область возможного расположения центров объектов внутри ограничивающего объема набора данных.



(б) область возможного расположения центров объектов, локализуемых в октантах следующего уровня.

Рис. 2: Область возможного расположения центров объектов внутри ограничивающего объема набора данных, а также внутри октантов следующего уровня.

Рисунок 2 иллюстрирует область возможного расположения центров объектов внутри объема всей

сцены, а также внутри октантов следующего уровня. Назовем величину

$$\rho(n, l) = \frac{nl^3}{(1-l)^3}$$

фактором пространственной наполненности.

Определение 2. Набор n одинаковых кубов с ребрами, ориентированными вдоль главных координатных осей и имеющими размер $0 \leq l < 1/2$, назовем модельным и обозначим как $S(n, l)$, если они независимым и случайным образом помещены в единичный куб.

Заметим, что при относительном размере объектов $l \geq 1/2$ октарная структура вырождается, поскольку все объекты локализируются в самом верхнем октанте при глубине дерева $h = 1$. Поэтому при дальнейшем рассмотрении мы ограничимся содержательным случаем $l < 1/2$. Исследуем свойства регулярных октодеревьев, развернутых для модельного набора данных.

Утверждение 1. Пусть регулярное октарное дерево глубины $h \leq \lceil \log_2 \frac{1}{l} \rceil$ построено для модельного набора $S(n, l)$. Тогда математическое ожидание числа объектов в октантах i -го уровня и ниже ($1 \leq i \leq h$) есть

$$n_i = n \frac{1 - 2^{i-1}l^3}{(1-l)^3}$$

Утверждение 2. Пусть регулярное октарное дерево глубины $h \leq \lceil \log_2 \frac{1}{l} \rceil$ построено для модельного набора $S(n, l)$, тогда математическое ожидание числа объектов в октанте i -го уровня есть

$$N_i = \begin{cases} n \frac{(1-2^{i-1}l)^3 - (1-2^i l)^3}{8^{i-1}(1-l)^3}, & \text{если } 1 \leq i < h; \\ n \frac{(1-2^{i-1}l)^3}{8^{i-1}(1-l)^3}, & \text{если } i = h. \end{cases}$$

Из приведенных утверждений следует следующая теорема об ожидаемой глубине регулярного октарного дерева.

Теорема 1. Ожидаемая глубина регулярного октарного дерева $Octree(m)$, построенного для модельного набора данных $S(n, l)$ с фактором пространственной наполненности $\rho(n, l)$, определяется следующим выражением:

$$h = \begin{cases} \lceil \log_2 \frac{2}{l+(1-l)\sqrt{\frac{m}{n}}} \rceil, & \text{если } m \geq \rho(n, l); \\ \lceil \log_2 \frac{1}{l} \rceil, & \text{если } m < \rho(n, l). \end{cases}$$

Анализ вычислительной сложности

Проанализируем вычислительные затраты на построение регулярного октарного дерева, его обновление, а также на выполнение запросов, связанных

с выборкой объектов по заданной пространственной области и определением столкновений. Воспользуемся описанным модельным набором данных и получим оценки сложности на всем диапазоне изменения параметров задачи.

Анализ сложности построения октодеревя.
Имеет место

Теорема 2. Трудоемкость построения регулярного октарного дерева $Octree(m)$, для модельного набора данных $S(n, l)$, выраженная в операциях определения принадлежности объекта одному из восьми дочерних октантов, равна $Q_{deploy} = q(n, m, l)n$, причем коэффициент $q(n, m, l)$ удовлетворяет следующим соотношениям:

$$q(n, m, l) = \lceil \frac{1}{3} \log_2 n - \frac{1}{3} \log_2 m \rceil \text{ при } l = 0;$$

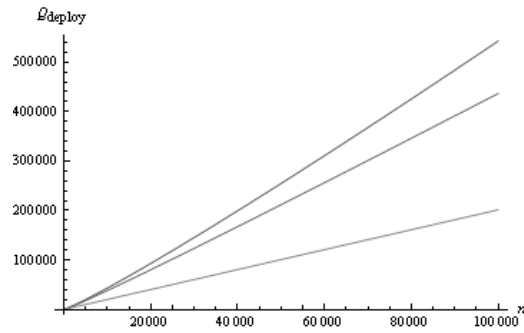
$$q(n, m, l) \leq \log_2 \frac{1}{l} \text{ при } l > 0.$$

Доказательство теоремы основывается на приведенных выше леммах и предположении, что дерево строится путем последовательного добавления объектов в сцену. Теорема имеет важное следствие, связанное с разным асимптотическим ростом сложности построения дерева для точечных и протяженных данных. Для первых ($l = 0$) сложность оценивается как $O(n \log_2 n)$, а для вторых ($l > 0$) — как $O(n)$. На рисунке 3 приведены графики сложности построения октодеревя от количества объектов в модельном наборе n при разных значениях порога наполненности октантов m , построенные для точечных (верхние кривые) и протяженных объектов (нижние кривые).

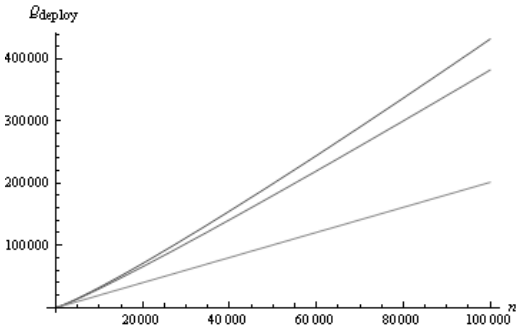
Примечательно, что в исследуемом диапазоне размерности задачи (до ста тысяч объектов) все кривые выглядят схожим образом, несмотря на наличие логарифмического фактора в асимптотической оценке сложности для точечных объектов. Согласно приведенным графикам сложность построения дерева для протяженных объектов несколько ниже, чем для точечных данных. Это объясняется тем обстоятельством, что при одинаковом общем количестве, часть объектов локализуется на более высоких уровнях октарного дерева и не возникает необходимости в их дальнейшем анализе.

Анализ сложности определения столкновений.

Перейдем к оценкам затрат на поиск столкновений объектов. Как правило, точное пересечение двух геометрических объектов сложной формы требует большого объема вычислений [2, 3]. Поэтому эффективной стратегией является предварительная локализация потенциальных столкновений с помощью простых тестов, основанных на сепарации



(a) $m = 10, l = 0, 0.01, 0.1$.



(b) $m = 100, l = 0, 0.01, 0.1$.

Рис. 3: Зависимость стоимости построения октарного дерева от количества объектов при различных габаритах объектов и параметра заполнения листовых октантов.

пространства и пересечении ограничивающих объемов [4, 7]. Точная процедура определения столкновений применяется лишь в случае положительного вердикта, что происходит в приложениях относительно редко и, как результат, существенно уменьшает общее время поиска столкновений. Эффект особенно ощутим для сцен, объекты которых геометрически представлены сложными аналитическими кривыми, поверхностями или полиэдрами с большим числом граней.

Предположим, что октарное дерево развернуто для модельного набора $S(n, l)$ и предстоит выявить все пары объектов, допускающие столкновения. В наивном алгоритме пришлось бы попарно пересечь все ограничивающие объемы объектов, что привело бы к выполнению $n(n-1)/2$ операций поиска пересечения между объектами. Однако, ввиду предварительного отсеивания большого количества априори непересекающихся объектов, поиск коллизий при помощи октарного дерева может быть выполнен гораздо эффективнее. Имеет место

Теорема 3. Пусть регулярное октарное дерево $Octree(m)$ глубиной h развернуто для модельного набора данных $S(n, l)$. Тогда вычислительная трудоемкость поиска столкновений, выраженная в операциях пересечения между ограничивающими

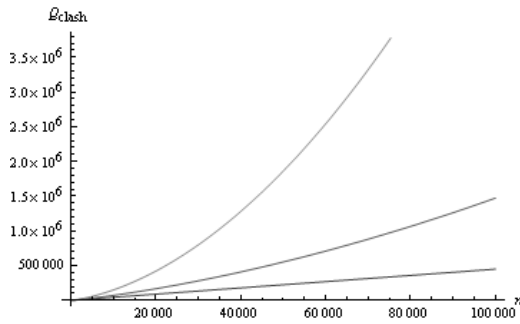
объемами, может оцениваться следующим образом:

$$Q_{clash} \leq \frac{mn}{2} + n^{\frac{4}{3}}m^{\frac{2}{3}}\left(\frac{45l}{2} + 186l^2\right) + 57l^2n^2 + o(l^2), l \rightarrow 0$$

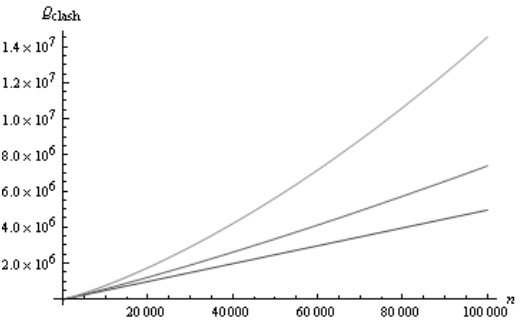
$$Q_{clash} \leq n^2, \forall l \in [0; 1]$$

где m и n ограничены.

Примечательно, что при небольших габаритах объектов, вычислительная трудоемкость поиска столкновений имеет субквадратичный рост, поскольку преобладающими оказываются первые члены суммы.



(a) при пороге заполнения октантов $m = 10$.



(b) при пороге заполнения октантов $m = 100$.

Рис. 4: График стоимости локализации столкновений в зависимости от количества объектов при различных значениях габаритов ($l = 0, l = 0.001, l = 0.003$).

На приведенных графиках (рис. 4) видно, что сложность локализации столкновений возрастает с увеличением числа объектов и их размеров. Нижние прямые на графиках соответствуют точечным данным, а средние и верхние кривые — протяженным объектам. Хотя анализ совпадений точечных данных имеет линейную сложность, а поиск столкновений протяженных объектов — квадратичную, для разреженных сцен различия оказываются не столь существенными даже для значительного числа объектов, составляющего на приведенных графиках 100000. Это служит обоснованием применимости метода в промышленных приложениях,

оперирующих большими данными и подверженных деградации производительности даже при использовании алгоритмов невысокой полиномиальной сложности.

Анализ сложности выборки объектов по заданной области.

Перейдем к оценкам затрат на выборку объектов, лежащих в заданной пространственной области. Для определенности будем считать, что область выборки представляет собой куб с относительным размером ребер L , приведенным к габаритам ограничивающего параллелепипеда всего набора данных. Определим характерный уровень локализации области выборки как $H = \lceil \log_2 \frac{1}{L} \rceil$. Тогда верно следующая оценка трудоемкости выборки объектов, выраженной в операциях пересечения ограничивающих объемов:

$$Q_{clip} = \begin{cases} n \frac{64 - 224 \cdot 2^h l + 336 \cdot 4^h l^2 + 8^h (11l - 81l^2 + (113 - 56h)l^3)}{8^h (1-l)^3} \\ + 8(h-1), \text{ если } h \leq H; \\ n \frac{64 - 224 \cdot 2^H l + 336 \cdot 4^H l^2 + 8^H (11l - 81l^2 + (113 - 56H)l^3)}{8^H (1-l)^3} \\ + \frac{8^{h-H} + 56H - 120}{7}, \text{ если } h > H. \end{cases}$$

На рисунке 5 представлены семейства графиков, иллюстрирующие зависимость стоимости выборки объектов от габаритов объектов и размера области выборки. Как видно, затраты растут с увеличением размера области выборки и уже при $L > 1/4$ использование октарного дерева теряет смысл. При меньших размерах применение октарного дерева оправданно, однако рост габаритов объектов может нивелировать суммарный эффект.

Анализ сложности поиска ближайших соседей.

Перейдем к оценке затрат на поиск ближайших соседей. Будем считать, что октарная структура уже развернута и для заданного объекта сцены требуется найти ближайшего соседа. Несложно показать, что в регулярном октарном дереве $Octree(m)$ глубины h , развернутом для модельного набора данных $S(n, l)$, ближайший к исходному объект не может находиться на расстоянии, превышающем размер диагонали листового октанта вдвое, т.е.

$$d_{min} < 2^{2-h} \sqrt{3}$$

Поэтому область поиска соседа всегда может быть ограничена кубом, ориентированным по главным координатным осям, с центром в исходном объекте и стороной $2^{2-h} \sqrt{3}$. Таким образом, в ходе работы алгоритма будут анализироваться лишь те октанты, которые имеют непустое пересечение с данной областью. Тогда в деревьях с глубиной, не превышающей четырех, анализу подлежат все октанты и все объекты сцены. В более глубоких деревьях на

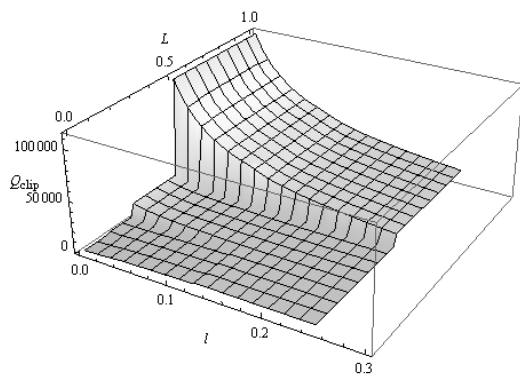
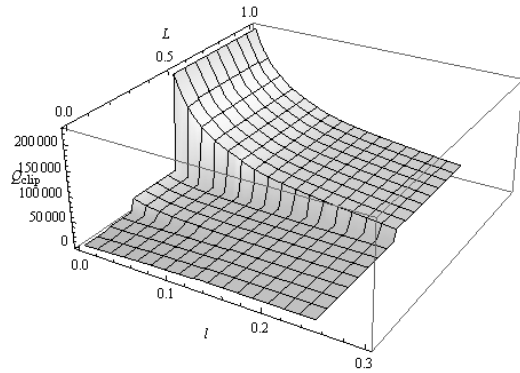
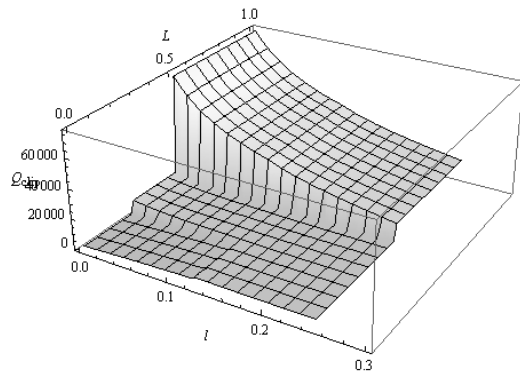
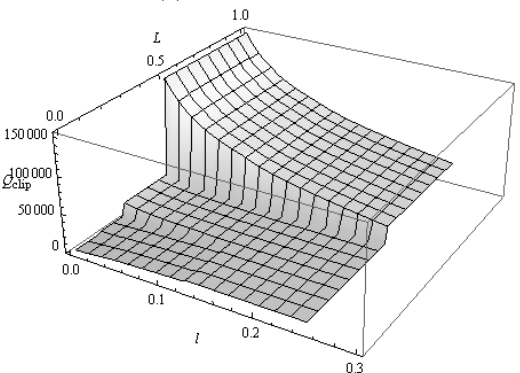
(a) $n = 5000, m = 5$.(b) $n = 10000, m = 5$ (c) $n = 5000, m = 10$ (d) $n = 10000, m = 10$

Рис. 5: Графики стоимости выборки объектов от габаритов и размера области.

верхних уровнях область поиска пересекает не более восьми октантов, а на нижних уровнях ($i \geq h - 2$) — не более $(|2^{i+2-h}\sqrt{3}| + 1)^3$. В каждом анализируемом октанте необходимо вычислить функцию расстояния для всех его объектов. Можно показать, что с учётом распределения объектов по уровням дерева при $h \geq 5$ вычислительная сложность поиска, выраженная в операциях определения расстояния между парой кубов, может быть оценена как:

$$Q_{neighbour} \leq \frac{1}{8^h(1-l)^3} (-4096n + 412^{4+h}ln + 213 \cdot 4^{1+h}l^2n - 8^h(816 + 8h(7l^3n - 8(1-l)^3)) + l(-2445 + 11n + l(2439 - 81n + l(-809 + 202n))))$$

Интерес вызывает тот факт, что в случае точечных данных ($l = 0$) стоимость поиска оценивается как $Q_{neighbour} \leq 4096m + 64 \log_8 \frac{n}{m} + 816$, что согласуется с известной асимптотической оценкой $O(\log n)$.

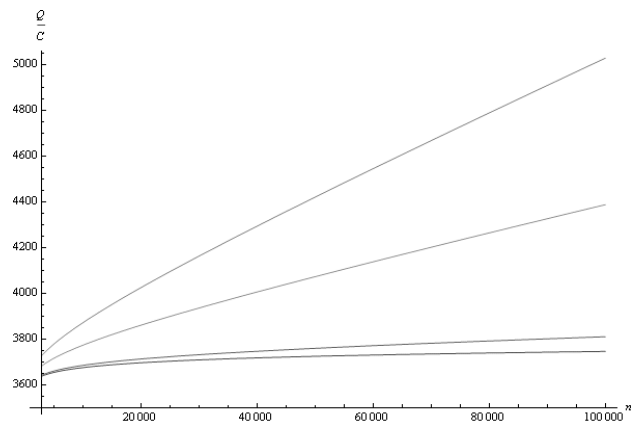


Рис. 6: Вычислительная стоимость поиска соседей в зависимости от числа и габаритов объектов.

На рисунке 6 представлено семейство кривых, отражающих зависимость стоимости поиска ближайшего соседа от числа объектов в сцене и их габаритов при $l = 0, 0.00005, 0.0005, 0.001$. Нижние кривые соответствуют меньшим значениям габаритов. Как видно, в случае точечных данных стоимость поиска имеет логарифмический рост, а с увеличением габаритов объектов проявляет линейный характер.

Результаты вычислительных экспериментов.

Для подтверждения результатов теоретического анализа были проведены серии вычислительных экспериментов с модельными наборами данных. Каждая серия экспериментов предназначалась для

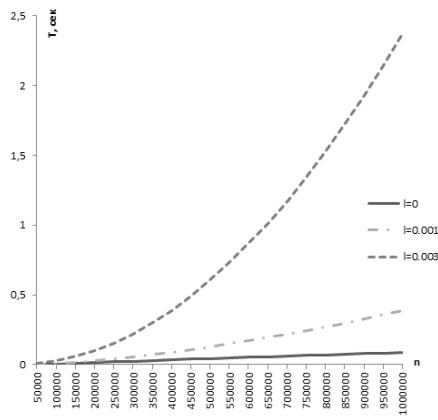
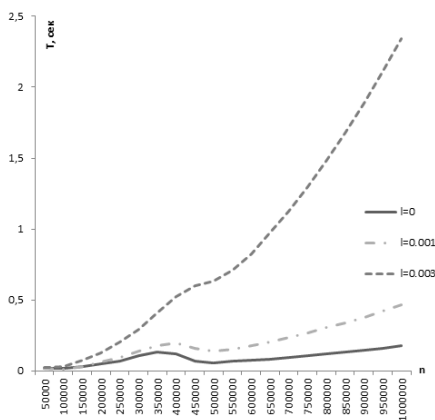
(a) $m = 10$.(b) $m = 100$.

Рис. 7: Зависимость времени локализации столкновений от количества объектов в сцене при различных габаритах и пороге наполненности октантов.

получения практических оценок сложности для исследованных операций, включая построение и обновление пространственных индексов на основе регулярного октодеревя, локализация столкновений в сцене, выборка объектов по заданной пространственной области и поиск ближайших соседей. Для достоверности анализа случайным образом генерировалось множество модельных наборов с идентичными значениями параметров, а затем результаты усреднялись. Эксперименты выполнялись на персональном компьютере с типовой конфигурацией: процессор Core i7 3770 3.4GHz, 16 GB RAM (1600MHz).

Для краткости ограничимся анализом результатов, связанных с локализацией столкновений в сцене. Как видно из графиков 7, порог наполненности октантов существенно не влияет на время разре-

шения запроса, однако может приводить к немонотонному росту затрат с увеличением числа объектов. Это объясняется тем, что в октанном дереве появляется еще один уровень и время разрешения запросов претерпевает скачкообразное изменение.

Другим важным результатом является существенная зависимость затрат от числа объектов в сцене и их габаритов. Однако в случае слабо заполненных сцен можно рассчитывать на оптимистическую оценку трудоемкости пространственных запросов, характерную для точечных данных. С ростом плотности пространственного заполнения сцены затраты на локализацию столкновений неизбежно растут и в предельном случае совпадают с упомянутой пессимистической оценкой, соответствующей наивному алгоритму. Накладные расходы на обновление индексов остаются относительно низкими во всех практически содержательных случаях.

Все полученные экспериментальные данные хорошо согласуются с теоретическими результатами и качественно подтверждают выводы о перспективности использования регулярных октодеревьев для пространственного индексирования сложных динамических сцен.

Выводы

Таким образом, рассмотрено важное семейство методов декомпозиции, основанное на регулярных октантных деревьях и обеспечивающее эффективную индексацию, быстрый поиск и анализ больших пространственных данных. Для алгоритмов определения столкновений и выборки объектов по заданной области получены оценки сложности в среднем, которые улучшают известные пессимистические результаты и служат теоретическим обоснованием их применения к более широкому классу приложений.

Литература

- [1] *Cai M., Revesz P.* Parametric rectangles: an index structure for moving objects // Proceedings of the 10th COMAD International conference on management of data, 2000. – pp. 57-64.
- [2] *Gottschalk S., Lin M. C., Manocha D.* OBB Tree: a hierarchical structure for rapid interference detection // Proceedings of the SIGGRAPH'96 Conference, 1996. –pp. 171-180.
- [3] *Gottschalk S.* Collision queries using oriented bounding boxes Chapel Hill: The University of North Carolina, 2000.
- [4] *Jimenez P., Thomas F., Torras C.* 3D collision detection: a survey // Computers and Graphics. – 2001. –Vol.25, – pp.269-285.
- [5] *Kedem G.* The quad-CIF tree: a data structure for hierarchical on-line algorithms // Proceedings of the 19th Design Automation Conference, 1992. –pp. 352-357

- [6] *Semenov V.A., Kazakov K.A., Morozov S.V., Tarlapan O.A., Zolotov V.A., Degenis T.* 4D modeling of large industrial projects using spatiotemporal decomposition // *eWork and eBusiness in Architecture, Engineering and Construction*, 2010. –pp. 89-95
- [7] *Semenov V.A., Kazakov K.A., Zolotov V.A., Jones H., Jones S.* Combined strategy for efficient collision detection in 4D planning applications // *Computing in Civil and Building Engineering*, 2010. –pp. 31-39
- [8] *Золотов В.А., Семенов В.А.* Современные методы поиска и индексации многомерных данных в приложениях моделирования больших динамических сцен // *Труды Института системного программирования*. –2013. –Т.24 –С. 381-416