

Н.И. Витиска, Н.А. Гуляев, И.Г. Данилов

**О ПРОБЛЕМАХ И ЗАДАЧАХ ОБРАБОТКИ И ОРГАНИЗАЦИИ
ГРАФИЧЕСКИХ ДАННЫХ ПРИ ОБЪЁМНОЙ ВИЗУАЛИЗАЦИИ
НА РАСПРЕДЕЛЁННЫХ СИСТЕМАХ***

Большинство актуальных задач науки и техники неразрывно связаны с задачей визуализации – помимо привнесения наглядности и информативности в изучение, управление и проектирование природных и технических объектов и процессов, визуализация решает ряд важных задач по реализации человеко-машинного взаимодействия. Ключевым вопросом визуализации в большинстве случаев является возможность отображения всей необходимой информации за отведённое количество времени с соблюдением лимита допустимого потребления ресурсов системы. Проблемы парадигмы объёмной визуализации в подавляющем количестве случаев стоят более остро, так как в данной парадигме, как правило, производится обработка большого количества данных, что может требовать чрезмерное количество вычислительных ресурсов, либо привносит ряд проблем в управление процессом визуализации. Помимо непосредственных подходов к оптимизации параметров рендеринга и процедур рендеринга (сэмплинга и композитинга), интерес представляет исследование подходов к организации обрабатываемых данных. Сокращение любых издержек, связанных с хранением и обработкой исходных графических данных позволяет не только существенно сократить соответствующие затраты, но и реализовать ряд модификаций процедур сэмплинга и композитинга, что открывает возможность для дополнительной оптимизации процесса визуализации. В работе рассмотрены ключевые задачи организации, хранения и обработки графических данных в парадигме объёмной визуализации в контексте реализации на распределённых системах. Рассмотрен общий подход к оптимизации, применимый для случая распределённой реализации. Описана постановка задачи оптимизации, описан вариант целевой функции. Далее рассмотрена задача организации графических данных на основании пространственного расположения и свойств содержимого сцены. Определены ключевые нюансы, описан вариант решения для случая распределённой обработки при помощи комбинации известных подходов, рассмотрены теоретические и практические аспекты. После чего рассмотрен подход к низкоуровневой реализации, описан вариант представления исходных данных в виде графа, помеченного набором свойств и вариант проведения процедуры рендеринга для такой структуры.

Компьютерная визуализация; методы рендеринга; методы организации графических данных; графические данные в объёмном представлении; большие данные; распределённые графовые вычисления.

N.I. Vitiska, N.A. Gulyaev, I.G. Danilov

**PROBLEMS AND ISSUES OF PROCESSING AND ORGANIZATION OF
GRAPHICAL DATA IN VOLUME VISUALIZATION ON DISTRIBUTED
SYSTEMS**

Most of current tasks in fields of science and technology are connected to visualization, which brings explicitness and informativity in management and design of natural and technical objects and processes, it also solves a number of important tasks of human-machine interface. The key problem of visualization in most cases is to display all the necessary information in allotted amount of time and system resources. In volume visualization, large amount of data is processed, which may require an excessive amount of computing resources, or it can introduce a number of problems in management of visualization process. Currently, research and study of approaches to organization of data being processed during volume visualization is crucial. Any decrease in computational costs during storing and processing the graphic data not only allows to reduce corresponding costs, but also allows to implement a number of modifications of sampling and composi-

* Работа выполнена при финансовой поддержке РФФИ, грант № 18-07-00733.

ing procedures, which introduces a number of possibilities for additional optimization of rendering process. This paper discusses key tasks of organizing, storing and processing graphical data in volume visualization paradigm in terms of distributed implementation. A general optimization approach suitable for distributed implementation is considered. A multiparametric optimization task is described, as well as objective function. Next, a problem of organization of graphical data based on spatial location and properties of volumetric phenomenon is reviewed. Key nuances are discussed, a solution for distributed case based on a combination of well-known approaches is described, theoretical and practical aspects are considered. An approach to a low-level implementation is considered, a technique of structuring the initial data as a marked graph by a set of properties, as well as a process of carrying out the rendering procedure on such structure is described.

Computer visualization; rendering methods; graphical data organization; graphical data in volumetric representation; big data; distributed graph computing.

Введение. В настоящее время в различных областях науки и техники широко применяются распределённые системы реального времени [1], которые чаще всего представляют собой информационно-управляющие системы, содержащие аппаратно-программные комплексы, решающие различные задачи в предметных областях. Многие актуальные задачи требуют решения дополнительных задач, одной из главных сопутствующих задач во многих случаях является задача визуализации. Решение задачи визуализации может быть нацелено на реализацию обратной связи с оператором системы, либо визуализация может быть неотъемлемой частью решаемой задачи, что актуально для многих систем моделирования, проектирования, систем дополненной реальности. Тем не менее, процесс решения задачи визуализации зачастую является крайне ресурсоёмким, что может приводить к снижению производительности не только непосредственной части, реализующей визуализацию, но и всей системы в целом. В зависимости от парадигмы визуализации, затраты вычислительных ресурсов и точность получаемого результата могут быть различными. Для парадигмы поверхностно-граничной визуализации в настоящее время существуют аппаратные решения, выполняющие большинство задач в рамках процесса рендеринга, снижая таким образом нагрузку на центральный процессор. Однако существует большое количество актуальных задач, связанных с отображением динамических сцен, визуализацией физически корректных симуляций взаимодействия света и субстанции, отображением объектов сложной неоднородной структуры, в которых парадигма поверхностно-граничной визуализации, несмотря на более высокую производительность, не позволяет получать результат требуемого уровня точности.

Решением в данном случае является применение прямой объёмной визуализации, которая позволяет получать высокоточные результаты для описанного круга задач. Однако существенными проблемами при применении прямой объёмной визуализации являются не только известные проблемы высокой ресурсоёмкости и отсутствия специализированных аппаратных решений, но и относительная сложность регулирования соотношения точности и производительности [2]. Последнее обстоятельство является наиболее значительным препятствием, поскольку во многих случаях результат приемлемой точности может быть получен с гораздо более низким уровнем затрат системных ресурсов. Это означает, что выполнение определённых процедур рендеринга вносит крайне малый вклад в итоговый результат, однако потребляет соответствующее количество системных ресурсов. Подход к оптимизации всего процесса визуализации в первую очередь должен быть ориентирован на сокращение издержек именно такого типа. Таким образом, первым этапом в ходе проектирования и разработки системы, выполняющей визуализацию, является задача определения подхода к оптимизации.

Подход к оптимизации. Задача оптимизации может быть определена как задача многопараметрической оптимизации со следующими постановками:

1. Максимизация точности за счёт последовательного связанного повышения значений всех параметров рендеринга при обеспечении необходимой производительности не ниже заданного значения, например, для сохранения частоты кадров не ниже 15 кадров в секунду;

2. Максимизация конкретного параметра процедуры рендеринга, определяющего наибольший вклад (в зависимости от конкретной ситуации) в точность результата, например, максимизация количества вторичных лучей при использовании модели поглощения-излучения.

Реализация многопараметрической оптимизации очевидно должна производиться в автоматическом режиме, что предполагает перенос всех действий выбора и принятия решений с человека на соответствующую составляющую системы. Вводимые параметры могут свести цель принятия оптимального решения к построению компромиссной целевой функции, в которую для оптимизируемых параметров вводятся коэффициенты веса k , определяемые путём экспертных оценок. Предполагается, что в реальных ситуациях принятие оптимального решения связывается с выбором наилучшего варианта из множества попарно допустимых и вновь предлагаемых. При этом здесь трудность выбора связана с определением совокупности параметров рендеринга, для которых целевая функция из всех рассчитанных окажется максимальной. Тем не менее, возможным остаётся тот факт, что из допустимых вариантов окончательный вариант окажется лучшим из выбранных в локальном смысле, однако не в глобальном.

Например, целевая функция может иметь следующий вариант представления с учётом попарной оптимизации для параметров точности результата и производительности:

$$Vid_i = k_{1i} \frac{\phi_C}{\phi_N} - k_{2i} \frac{\psi_C}{\psi_M}, \quad (1)$$

где ϕ_C – текущая точность; ϕ_N – достижимая (нормированная) точность; ψ_C – текущая производительность; ψ_M – допустимая производительность.

Текущие и достижимые показатели точности могут быть определены на основании одного из известных методов анализа изображений [3–5]. В свою очередь, текущая производительность может быть оценена расчётным [6] или экспериментальным путём [7] для соответствующей ветви распределённой системы реального времени и представлена, например, как количество кадров в секунду. В выбранной компромиссной целевой функции (1) обеспечиваются параметры различной размерности, включая нормирование по некоторому заданному значению, а учёт относительной важности одного из двух оптимизируемых параметров задаётся с помощью коэффициентов веса k_{1i} и k_{2i} .

Первоначально имеет смысл оценить в реализуемом подходе значения Vid_i для трёх крайних ситуаций, при которых, во-первых, имеет приоритет лишь точность результирующего изображения (поэтому $k_{1i} = 1$ и $k_{2i} = 0$); во-вторых, точность и производительность одинаково важны (таким образом, $k_{1i} = 0,5$ и $k_{2i} = 0,5$) и в третьих, важна лишь скорость и следовательно $k_{1i} = 0$, а $k_{2i} = 1$. Среди рассчитанных Vid_i следует зафиксировать лучший вариант и для него определить варианты изменения одного из коэффициентов. Если же будет возможным с помощью нормирующих параметров и изменения весовых коэффициентов получить приемлемый вариант, то имеет смысл проанализировать новую пару оптимизируемых параметров. В этом случае может быть вновь применена попарная оптимизация, но уже с новыми оцениваемыми параметрами.

Тем не менее, в большинстве случаев получение оптимального набора параметров рендеринга составляет лишь часть возможных действий, направленных на оптимизацию процесса визуализации. Другая часть возможных действий связана с

исходными графическими данными, то есть может быть реализована за счёт подбора адекватного подхода к организации хранения и обработки графических данных. Более того, многие параметры рендеринга имеет смысл регулировать лишь при учёте свойств содержимого сцены, определение и вычисление которых зачастую невозможно без наличия определённой структуры организации. Таким образом, следующей важной задачей является подбор адекватного способа организации содержимого сцены на основании расположения и ряда других свойств объёмного феномена.

Подход к организации содержимого сцены. Исходя из того факта, что наиболее распространённым видом представления графических данных является воксельный вид или равномерная воксельная решётка (Regular Voxel Grid), то в качестве базы для подхода к организации содержимого сцены на основании свойств и расположения объёмного феномена могут быть рассмотрены два варианта. Первый вариант заключается в организации содержимого сцены при помощи иерархической структуры, обеспечивающей регулярное пространственное разбиение – октарного дерева (Octree) [8, 9]. Вторым вариантом является использование линейной структуры, реализующей нерегулярное пространственное разбиение [10]. В этом случае организующая структура будет включать ограничивающие примитивы (Bounding Volume), расположенные в пространстве сцены, содержащие во внутреннем подпространстве часть объёмного феномена. Выбор варианта организации содержимого сцены, как правило, основывается на особенностях конкретной задачи, поскольку универсального способа в настоящее время не известно.

В определённых случаях применение одного из «базовых» подходов к организации содержимого сцены может быть исчерпывающим решением в рамках конкретной задачи. Однако необходимость обновления содержимого сцены (как полностью, так и некоторых частей), необходимость независимой обработки частей сцены, а также необходимость извлечения дополнительных данных, используемых, например, при упрощении вычислительных процедур, может приводить к неизбежности усложнения способа организации содержимого. Таким образом, имеет место необходимость модификации, заключающейся, например, во включении в исходную структуру дополнительных конструктивных атрибутов и сведений, необходимых для выполнения описанных выше действий. Так, в подходе, предложенном в [11], исходная структура, определяющая одноуровневые ортогонально ориентированные параллелепипеды (рис. 1), усложнена за счёт включения дополнительных сведений относительно свойств каждого «хаба» (hub), т.е. части объёмного феномена и соответствующего ограничивающего примитива.

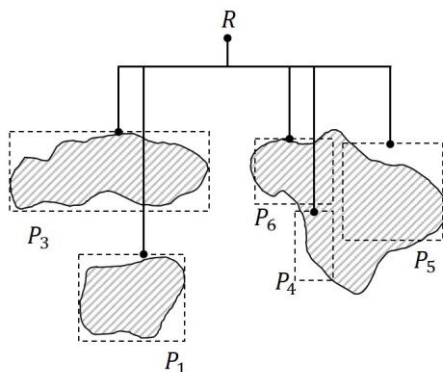


Рис. 1. Организация графических данных на основании пространственного расположения содержимого сцены

Таким образом, каждый ограничивающий примитив, аппроксимирующий определённую часть объёмного феномена, является частью хаба сцены, т.е. составляющей, отвечающей за пространственное разделение. С другой стороны, хаб сцены может содержать аналогичные составляющие, отвечающие и за кластеризацию по какому-то иному признаку. Таким образом, пространственное расположение не является единственным критерием для выполнения организации содержимого сцены при помощи такого «усложнённого» варианта организующей структуры.

Процесс формирования такой структуры отличается от «стандартного» варианта, поскольку помимо определения непосредственного аппроксимирующего примитива, производятся дополнительные действия, такие как, например, вычисление усреднённого визуального значения, такого как цвет или непрозрачность (рис. 2), или определение дисперсных характеристик, используемых при оптимизации, как показано в [12].

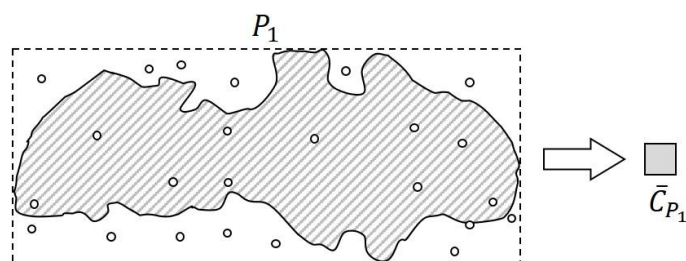


Рис. 2. Выделение свойства части феномена для соответствующего «хаба»

Также в данном случае эксплуатируется аппликативность (поэтапное формирование сцены), что позволяет производить обновление всего содержимого сцены за счёт обновления всех хабов, что в свою очередь, позволяет производить независимую обработку каждого хаба.

Наконец, непосредственные процедуры рендеринга – процедуры сэмпинга (снятия образцов) и композитинга (определения результирующего значения) в зависимости от модификации способа организации, очевидно, претерпят определённые изменения. Так, например, процедура сэмпинга будет переориентирована на поиск пересечений рассматриваемого луча и ограничивающего примитива, то есть на перебор множества ограничивающих примитивов, сформированных при помощи соответствующего способа организации, вместо множества точек. В свою очередь, процедура композитинга может, например, выполнять формирование результирующего значения на основании интервалов, образованных рассматриваемым лучом и пересечённым примитивом вместо интервалов, определяемых соседними точками сэмплов.

Так, выполнение последовательного регулярного сэмпинга при формировании аддитивных (накапливаемых) визуальных значений (например, величины накопленной непрозрачности) вдоль всего сегмента луча может быть заменено выполнением последовательного регулярного сэмпинга лишь на небольших сегментах, определяемых внутренним подпространством организующей структуры. Логичным продолжением будет являться реализация способа оптимизации, основанного на использовании вычисленных для каждого хаба дисперсных характеристиках, переопределяя каждый параметр p через полученные свойства, как, например, предложено в [13]:

$$p' = p(k_1\chi_1 + k_2\chi_2 + \dots + k_t\chi_t), \quad (2)$$

где p – исходный параметр рендеринга;

p' – переопределённый параметр рендеринга;

$k_1 \dots k_n$ – коэффициенты нормирования;

$\chi_1 \dots \chi_n$ – характеристики содержимого соответствующего хаба.

Таким образом, регулирование количества сэмплов для каждого полученного сегмента будет производиться на основании выделенных свойств соответствующей хабу части объёмного феномена. Что позволит, например, для случая вычисления величины накопленной непрозрачности сократить количество обрабатываемых точек сэмплов за счёт исключения некоторых из них из обработки, как представлено ниже (табл. 1).

Таблица 1

Некоторые результаты эмпирических исследований

| Исходное (максимальное) количество сэмплов s (шт.) | Полученное количество сэмплов (структура и оптимизация) \bar{s}' (ср., шт.) | Полученное количество сэмплов (только структура) \bar{s}' (ср., шт.) | Оптимальное количество сэмплов S_{opt} (шт.) | Полученная точность (структура и оптимизация) q |
|--|---|--|--|---|
| 256 | 131 | 198 | 139 | 0.953 |
| 384 | 289 | 325 | 271 | 0.922 |
| 256 | 233 | 242 | 229 | 0.977 |
| 512 | 391 | 399 | 362 | 0.915 |
| 128 | 104 | 116 | 92 | 0.969 |
| 256 | 154 | 201 | 174 | 0.936 |

Применение структуры организации и подхода к оптимизации количества сэмплов позволяет для простых аддитивных визуальных значений (например, для величины накопленной непрозрачности) сократить количество обрабатываемых точек сэмплов на 30–40 % с сохранением уровня точности результата не менее 90%. Таким образом, рендеринг может быть осуществлён за счёт обработки меньшего количества сэмплов, в среднем близкого к оптимальному (по количеству значащих сэмплов), что позволяет получать результирующие изображения большей степени точности при том же количестве сэмплов. Ниже (рис. 3) приведен результат визуализации при помощи модели поглощения-излучения без использования организующей структуры и с использованием организующей структуры типа ортогонально-ориентированный параллелепипед с одинаковым количеством точек сэмплов $s = 128$ в обоих случаях.

Далее (рис. 4) представлен результат визуализации при помощи модели поглощения-излучения достаточно сложной сцены (уровень заполненности выше 70 %) сначала с без использования организующей структуры и оптимизации, затем с использованием организующей структуры и оптимизации, основанной на свойствах содержимого сцены. В первом случае полученная точность равна 0.64, во втором – 0.91, притом исходное количество сэмплов одинаково $s = 512$ при габаритах сцены 512^3 .

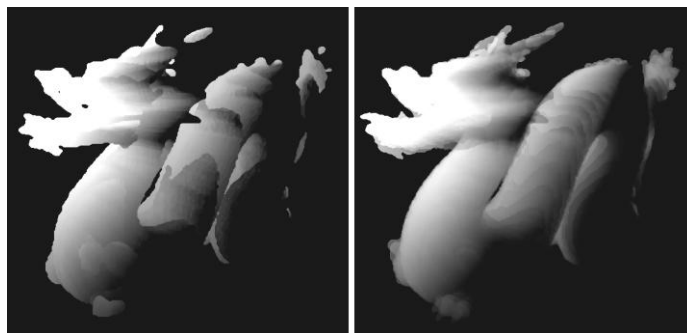


Рис. 3. Пример повышения точности результата визуализации благодаря использованию организующей структуры

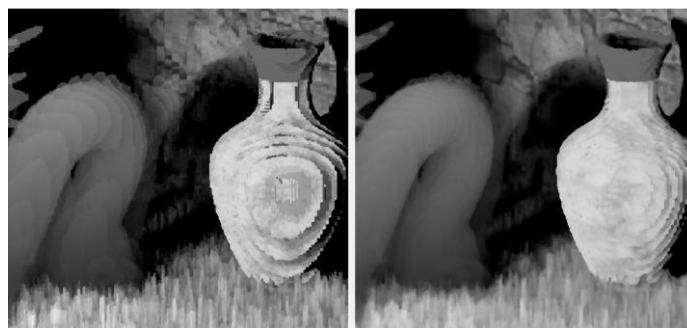


Рис. 4. Пример повышения точности результата визуализации благодаря использованию организующей структуры и оптимизации

Несмотря на то, что такой подход позволяет существенно сократить, во-первых, количество сэмплов, не считывающих значащие элементы объёма, во-вторых, количество сэмплов, являющихся излишними при конкретных свойствах содержимого сцены, достижение оптимального (близкого к оптимальному) количества сэмплов не гарантирует возможности выполнения визуализации, например, в реальном времени при ограничении на частоту кадров. Поэтому дальнейшее повышение производительности визуализации может быть возможным благодаря реализации потенциала параллелизма, заложенном в данном подходе.

Подход к реализации в распределённых системах. В контексте распределённых систем, исходя из особенностей актуальных задач, аспекты реализации имеет смысл рассматривать с точки зрения обработки больших объемов данных. Это означает, что в большинстве случаев будет превалировать работа с «крупными» единицами, что, в свою очередь, означает, что в качестве элемента обработки вполне могут быть рассмотрены гига- и мультигига-воксельные структуры [14]. В таком случае основные пути решения подобных задач – эффективная декомпозиция на подзадачи и их независимая, возможно, распределенная обработка. Сама структура данных позволяет достаточно просто описать задачу прямой объемной визуализации в терминах обработки многомерных структур или связанных элементов пространства, представленных в виде графа. Это, в свою очередь, позволяет для хранения исходных графических данных и для ряда задач обработки (например, кластеризации) использовать существующие инструменты распределенной обработки больших графов [15] как «нижний уровень» по отношению к уже имеющимся способам организации.

Определяя круг подходов, необходимо отметить, что в настоящее время в области обработки больших графов доминирует схема представления данных в виде графа, помеченного набором свойств (Labeled Property Graph или LPG): узлы и связи в таком графе могут быть помечены любым количеством свойств, выраженных в виде объектов любого типа [16]. Такая модель имеет ряд преимуществ перед давно известной и широко используемой RDF [17]. Для того, чтобы спроектировать адекватный способ (схему) представления графических данных в виде LPG необходимо, во-первых, определить, какая модель графовых вычислений будет применяться далее для обработки и, во-вторых, на какие операции при решении задачи прямой объёмной визуализации может повлиять разработанная схема. Несмотря на то, что существует несколько разных вариантов организации графовых вычислений [18], в настоящее время на практике (в силу простоты реализации) применяется в основном вершинно-ориентированный подход [19], предполагающий для каждого узла графа на каждой итерации алгоритма выполнение заданных вычислений, после чего может производиться обмен сообщениями между связанными узлами и синхронизация состояния системы. Реализуется такой подход, как правило, с использованием модели программирования BSP (Bulk Synchronous Parallel) [20].

С учетом предложенного выше способа организации графических данных, в предлагаемой схеме хранимым свойством узла графа является дескриптор выделенной части объема. Что означает, что такой дескриптор содержит в себе ограничивающий примитив (например, ортогонально-ориентированный параллелепипед), являющийся частью пространственного разбиения пространства сцены и содержащий в себе часть объёмного феномена (включающую преимущественно значащие элементы объёма), а также всю остальную необходимую для проведения процедуры рендеринга информацию (например, оценки цветового значения или дисперсные характеристики).

Помимо выделения отдельных частей объёмного феномена в узлы графа, необходимо установить связи между всеми соседними узлами. При использовании вершинно-ориентированного подхода для прохождения луча через объёмный феномен в процессе визуализации на каждой BSP-итерации, что может быть реализовано с помощью передачи сообщений между вершинами, необходимо выбрать вершину-приёмник, то есть следующую по ходу луча часть объёмного феномена. Для ответа на вопрос, проходит ли заданный луч через выделенную область, здесь может быть решена тривиальная задача определения пересечения луча с ограничивающим примитивом [21]. Таким образом, в предлагаемой схеме данных, хранимым свойством связи в графе является дескриптор ограничивающего примитива (или узла сцены). В итоге, трассирование луча в таком графе объёмных данных можно рассматривать как последовательное продвижение по смежным вершинам графа «сообщения-фотона», в котором хранится информация о накопленном на определенной итерации вычислений цвете и непрозрачности.

Выводы. Повышение производительности трассировки лучей в прямой объёмной визуализации возможно за счёт сокращения количества обрабатываемых сэмплов, что может быть реализовано не только при помощи пространственного разбиения, но и при помощи регулирования количества сэмплов, основываясь на свойствах обрабатываемых графических данных. Совмещение этих подходов позволяет сократить количество излишних сэмплов в среднем до 65-75%, а в ряде случаев возможно получение близкого к оптимальному количества сэмплов (по количеству вокселей, содержащих ненулевые значения), что означает исключение всех незначащих сэмплов. Дальнейшее повышение производительности возможно за счёт исключения некоторого количества значащих сэмплов, однако в любом случае будет существовать некоторый порог, снижение количества сэмплов ниже которого будет вызывать значительное снижение точности результата, что может быть не всегда

приемлемо. Поэтому дальнейшее повышение производительности, особенно для больших объемов, возможно за счёт реализации параллельной обработки, притом используемая структура организации может быть задействована в качестве «высокоуровневой» составляющей при реализации процесса рендеринга на базе графовых вычислений. Такой подход позволяет естественным образом представить исходные данные задачи объёмной визуализации в виде графа с возможностью его последующего долговременного распределенного хранения и обработки, причем в качестве задач обработки могут выступать как непосредственно сама визуализация объема, так и любая другая вспомогательная вычислительная задача: определение, изменение и оптимизация свойств графических данных, кластеризация и т.п.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев И.А., Мельник Э.В.* Децентрализованные системы компьютерного управления. – Ростов-на-Дону: ЮИЦ РАН, 2011. – 196 с.
2. *Витиска Н.И., Гуляев Н.А.* Исследование вариантов модификации модели объёмной визуализации методом трассировки лучей // Информатизация и связь. – 2016. – № 3. – С. 28-33.
3. *Wang Z., Bovik A. C.* Mean squared error: Love it or leave it? A new look at signal fidelity measures // IEEE signal processing magazine. – 2009. – Vol. 26, No. 1. – P. 98-117.
4. *Wang Z. et al.* Image quality assessment: From error measurement to structural similarity // IEEE Trans. Image Processing. – 2004. – Vol. 13. – P. 600-612.
5. *Zhang L. et al.* FSIM: A feature similarity index for image quality assessment // IEEE transactions on Image Processing. – 2011. – Vol. 20, No. 8. – P. 2378-2386.
6. *Дубовик А.А.* Способ ускорения работы модуля описания моделей // Матер. 4-й Всероссийской научно-технической конференции «Суперкомпьютерные технологии». – 2016. – Т. 2. – С. 27-30.
7. *Wimmer M., Wonka P.* Rendering time estimation for real-time rendering // Rendering Techniques. – 2003. – P. 118-129.
8. *Knoll A.* A Survey of Octree Volume Rendering Methods // 1st IRTG Workshop. – 2006. – P. 8-19.
9. *Laine S., Karras T.* Efficient Sparse Voxel Octrees // Transactions on Visualization & Computer Graphics. – 2011. – Vol. 17, No. 8. – P. 1048-1059.
10. *Eisemann M. et al.* Fast ray/axis-aligned bounding box overlap tests using ray slopes // Journal of graphics tools. – 2007. – Vol. 12, No. 4. – P. 35-46.
11. *Витиска Н.И., Гуляев Н.А.* Метод визуализации трёхмерных сцен и объектов воксельной графики для систем имитационного моделирования // Известия ЮФУ. Технические науки. – 2015. – № 4 (165). – С. 77-87.
12. *Гуляев Н.А.* О подходе к исследованию зависимости точности объёмной визуализации от свойств графических данных // Информатика, управление и системный анализ (ИУСА-2019): Тр. V Всероссийской научной конференции молодых ученых с международным участием. – 2018. – С. 275-284.
13. *Vitiska N., Selyankin V., Gulyaev N.* An Approach to Optimization of Ray-Tracing in Volume Visualization Based on Properties of Volume Elements // Proceedings of the Third International Scientific Conference “Intelligent Information Technologies for Industry” (ITI’18). ITI’18 2018. – 2019. – Vol. 874. – P. 148-158.
14. *Гаврилов Н.И., Турпанов В.Е.* Качественная объёмная визуализация гига воксельных массивов в блочном представлении на примерах данных из медицины // Матер. конференции «ГрафиКон». – 2012. – С. 174-179.
15. *Данилов И.Г., Гуляев Н.А., Селянкин В.В.* О применимости модели графовых вычислений при прямой объёмной визуализации // Информационные технологии, системный анализ и управление (ИТСАУ-2018): Сб. трудов XVI Всероссийской научной конференции молодых ученых, аспирантов и студентов. – 2018. – Т. 3. – С. 156-159.
16. *Angles R.* The Property Graph Database Model. // AMW. – 2018.
17. *Barrasa J.* RDF Triple Stores vs. Labeled Property Graphs: What’s the Difference? // Neo4j Blog URL – <https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/> (дата обращения: 22.06.2019).
18. *Heidari S. et al.* Scalable graph processing frameworks: A taxonomy and open challenges // ACM Computing Surveys (CSUR). – 2018. – Vol. 51, No. 3. – P. 60.

19. Malewicz G. et al. Pregel: a system for large-scale graph processing // Proc. of the 2010 ACM SIGMOD International Conference on Management of data. – 2010. – P. 135-146.
20. Valiant L.G. A bridging model for parallel computation // Communications of the ACM. – 1990. – Vol. 33, No. 8. – P. 103-111.
21. Ericson C. Real-time collision detection. – CRC Press, 2004.

REFERENCES

1. Kalyaev I.A., Mel'nik E.V. Detsentralizovannye sistemy komp'yuternogo upravleniya [Decentralized computer control systems]. Rostov-on-Don: YuNTS RAN, 2011, 196 p.
2. Vitiska N.I., Gulyaev N.A. Issledovanie variantov modifikatsii modeli ob"emnoy vizualizatsii metodom trassirovki luchey [A study on modifications of visualization model for volume rendering with ray tracing], *Informatizatsiya i svyaz'* [Informatization and communication], 2016, No. 3, pp. 28-33.
3. Wang Z., Bovik A. C. Mean squared error: Love it or leave it? A new look at signal fidelity measures, *IEEE signal processing magazine*, 2009, Vol. 26, No. 1, pp. 98-117.
4. Wang Z. et al. Image quality assessment: From error measurement to structural similarity, *IEEE Trans. Image Processing*, 2004, Vol. 13, pp. 600-612.
5. Zhang L. et al. FSIM: A feature similarity index for image quality assessment, *IEEE transactions on Image Processing*, 2011, Vol. 20, No. 8, pp. 2378-2386.
6. Dubovik A.A. Sposob uskoreniya raboty modulya opisaniya modeley [The way to accelerate the work of the model description module], *Mater. 4-y Vserossiyskoy nauchno-tekhnicheskoy konferentsii «Superkomp'yuternye tekhnologii»* [Materials of the 4th All-Russian Scientific and Technical Conference "Supercomputer Technologies"], 2016, Vol. 2, pp. 27- 30.
7. Wimmer M., Wonka P. Rendering time estimation for real-time rendering, *Rendering Techniques*, 2003, pp. 118-129.
8. Knoll A. A Survey of Octree Volume Rendering Methods, *1st IRTG Workshop*, 2006, pp. 8-19.
9. Laine S., Karras T. Efficient Sparse Voxel Octrees, *Transactions on Visualization & Computer Graphics*, 2011, Vol. 17, No. 8, pp. 1048-1059.
10. Eiseemann M. et al. Fast ray/axis-aligned bounding box overlap tests using ray slopes, *Journal of graphics tools*, 2007, Vol. 12, No. 4, p 35-46.
11. Vitiska N.I., Gulyaev N.A. Metod vizualizatsii trekhmernykh stsen i ob"ektov voksel'noy grafiki dlya sistem imitatsionnogo modelirovaniya [Method of visualization of three-dimensional scenes and objects voxel graphics for simulation systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 4 (165), pp. 77-87.
12. Gulyaev N.A. O podkhode k issledovaniyu zavisimosti tochnosti ob"emnoy vizualizatsii ot svoystv graficheskikh dannykh [On the approach to the study of the dependence of the accuracy of volumetric visualization on the properties of graphic data], *Informatika, upravlenie i sistemnyy analiz (IUSA-2019): Tr. V Vserossiyskoy nauchnoy konferentsii molodykh uchenykh s mezhdunarodnym uchastiem* [Informatics, management and system analysis (IUSA-2019): Proceedings of the V all-Russian scientific conference of young scientists with international participation], 2018, pp. 275-284.
13. Vitiska N., Selyankin V., Gulyaev N. An Approach to Optimization of Ray-Tracing in Volume Visualization Based on Properties of Volume Elements, *Proceedings of the Third International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'18)*. IITI'18 2018, 2019, Vol. 874, pp. 148-158.
14. Gavrilov N.I., Turlapov V.E. Kachestvennaya ob"emnaya vizualizatsiya gigavoksel'nykh massivov v blochnom predstavlenii na primerakh dannykh iz meditsiny [High-quality volumetric visualization of gigavox arrays in block representation using data from medicine], *Mater. konferentsii «Grafikon»* [Proceedings of the conference "GraphiCon"], 2012, pp. 174-179.
15. Danilov I.G., Gulyaev N.A., Selyankin V.V. O primenimosti modeli grafovykh vychisleniy pri pryamoy ob"emnoy vizualizatsii [On the applicability of the graph computation model for direct volume visualization], *Informatsionnye tekhnologii, sistemnyy analiz i upravlenie (ITSAU-2018): Sb. trudov XVI Vserossiyskoy nauchnoy konferentsii molodykh uchenykh, aspirantov i studentov* [Information technologies, system analysis and management (ITSAU-2018): Proceedings of the XVI all-Russian scientific conference of young scientists, postgraduates and students], 2018, Vol. 3, pp. 156-159.
16. Angles R. The Property Graph Database Model, *AMW*, 2018.

17. *Barrasa J.* RDF Triple Stores vs. Labeled Property Graphs: What's the Difference?, *Neo4j Blog*. Available at: <https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/> (accessed 22 June 2019).
18. *Heidari S. et al.* Scalable graph processing frameworks: A taxonomy and open challenges, *ACM Computing Surveys (CSUR)*, 2018, Vol. 51, No. 3, pp. 60.
19. *Malewicz G. et al.* Pregel: a system for large-scale graph processing, *Proc. of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 135-146.
20. *Valiant L.G.* A bridging model for parallel computation, *Communications of the ACM*, 1990, Vol. 33, No. 8, pp. 103-111.
21. *Ericson C.* Real-time collision detection. CRC Press, 2004.

Статью рекомендовал к опубликованию д.т.н., профессор В.М. Курейчик.

Витиска Николай Иванович – Общество с ограниченной ответственностью «Научно-исследовательский институт многопроцессорных вычислительных и управляющих систем»; e-mail: vit614294@rambler.ru; 347905, г. Таганрог, ул. Социалистическая, 150-г.; д.т.н.; в.н.с.

Гуляев Никита Андреевич – Южный федеральный университет; e-mail: m.yo.da@yandex.ru; 347928, г. Таганрог, пер. Некрасовский, 44; кафедра МОП ЭВМ; ассистент.

Данилов Игорь Геннадьевич – e-mail: igdanilov@sfedu.ru; кафедра МОП ЭВМ; к.т.н.; доцент.

Vitiska Nikolaj Ivanovich – Scientific Research Institute of Multiprocessor Computer and Control Systems, Co Ltd.; e-mail: vit614294@rambler.ru; 150-g Socialist street, Taganrog, 347905, Russia; dr. of eng. sc.; leading researcher.

Gulyaev Nikita Andreevich – Southern Federal University; e-mail: m.yo.da@yandex.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; the department of software engineering; cand. of eng. sc.; assistant professor.

Danilov Igor Gennadievich – e-mail: igdanilov@sfedu.ru; the department of software engineering; cand. of eng. sc.; associate professor.