

УДК 004.925.84

## МЕТОД ВИЗУАЛИЗАЦИИ МУЛЬТИОБЪЁМНЫХ ДАННЫХ И ФУНКЦИОНАЛЬНО ЗАДАННЫХ ПОВЕРХНОСТЕЙ С ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

© С. И. Вяткин, Б. С. Долговесов

*Институт автоматизации и электрометрии СО РАН,  
630090, г. Новосибирск, просп. Академика Коптюга, 1  
E-mail: sivser@mail.ru*

Предложен метод визуализации мультиобъёмных данных и функционально заданных поверхностей с применением графических процессоров. В методе предусмотрена визуализация большого числа объёмов, комплекс полупрозрачных и функционально заданных объектов, сложных полупрозрачных объёмов, в том числе пересечений объёмов в конструктивном твердотельном моделировании. Рендеринг разных объёмов одновременно — более сложная проблема, чем визуализация одного объёма, потому что требуются операции пересечения и смешивания. Функционально заданные поверхности хорошо подходят для встраивания внешних объектов в объёмы.

*Ключевые слова:* функционально заданная поверхность, функции возмущения, мультиобъёмный рендеринг, рейкастинг, конструктивная твердотельная геометрия, CUDA, графический процессор (GPU).

DOI: 10.15372/AUT20210204

**Введение.** Задача визуализации заключается в том, чтобы сделать объекты, состоящие из массивов данных или нескольких связанных наборов данных, видимыми наблюдателю. Одна из важных задач объёмного рендеринга — визуализация образцов трёхмерного скалярного поля, полученных из таких источников, как рентгеновская компьютерная томография или ультразвуковые сканеры. Актуальным направлением также является визуализация структурных особенностей, в частности границ областей с одинаковыми скалярными значениями, отличными от самих скалярных значений.

Концептуально процесс визуализации наборов различных данных включает три этапа преобразований для получения изображений:

1. Классификация. Преобразует исходный набор данных в новый набор классов объектов, явно содержащий информацию, которую мы хотим видеть.
2. Моделирование. Преобразование карт параметров особенностей наборов в объекты визуальной модели.
3. Рендеринг. Визуализация объектов в модели.

В зависимости от конкретной реализации может изменяться порядок этих преобразований или осуществляться их объединение.

На этапе классификации выделяются интересные особенности, которые могут быть, например, границами регионов с идентичными значениями. Классификация может быть двоичной или недвоичной (небинарной). С одной стороны, бинарная классификация даёт результаты «всё или ничего» в отношении вероятностей существования. С другой стороны, небинарная классификация может давать любое значение в диапазоне от 0 до 1. Бинарная и небинарная классификации имеют свои преимущества и недостатки.

На этапе моделирования извлечённый набор признаков сопоставляется с набором атрибутов визуальных носителей информации в визуальной модели. Визуальная модель представляет собой моделируемую 3D-сцену, в которой размещаются несущие информацию

визуальные примитивные  $3D$ -объекты — воксели, создаваемые с помощью небинарных классификаций. Воксель может иметь различные атрибуты, включая непрозрачность и излучательность, которые могут иметь компоненты цвета.

Третий этап преобразований — это рендеринг объектов. Рейкастинг — самый распространённый подход для рендеринга объёма благодаря своей гибкости при генерации изображения высокого качества [1, 2]. Большой объём однородных данных, содержащихся в объёмной модели, очень хорошо распараллеливается при вычислениях с помощью программно-аппаратных шейдеров графического процессора (GPU). Сегодня обычный графический процессор способен отобразить один объём данных высокого разрешения с высокой частотой кадров. Однако отображение одного однородного объёма не подходит для многих приложений. Использование комбинации различных объёмных наборов данных становится популярным в медицинских приложениях, в которых необходимы как анатомические, так и функциональные структуры. Мультиобъёмный рендеринг — это рендеринг нескольких объёмов. Таким образом, мультиобъёмный рендеринг необходим при обработке нескольких наборов данных, которые обычно имеют разную ориентацию и разрешение. В медицинской визуализации такие методы получения анатомических изображений, как компьютерная томография, магнитно-резонансная томография (МРТ), УЗИ, и методы получения функциональных изображений, как позитронная эмиссионная томография, однофотонная эмиссионная компьютерная томография, функциональная МРТ, используются для обследования пациента [3, 4]. Более того, модели медицинских инструментов и результаты сегментации необходимы для компьютерной хирургии. Современные методы имеют ограниченные возможности для интерактивного исследования мультиобъёмных данных в комбинации с полигональными моделями. Рендеринг нескольких объёмов одновременно в композиции с поверхностями осуществить труднее, чем визуализацию отдельных объёмов, так как требуется обработка перекрытий объёмов и смешивание цветов [5, 6]. Пересчёт объёмов в единой системе координат нежелателен, так как это приводит к потере качества или увеличению требований к памяти. Обязательное требование в виртуальной хирургии — сочетание объёмов с поверхностями. Функциональные модели хорошо подходят для встраивания посторонних объектов (инструментов, имплантатов и др.) в объёмные данные.

Целью представленной работы является создание метода визуализации мультиобъёмных данных и функционально заданных поверхностей для медицинских приложений.

**Способ задания поверхностей.** Технологии визуализации для медицинских приложений применяют описание поверхностей объектов с высоким разрешением, для чего используются функции отклонения (второго порядка) от базовой квадрики [7]. Функция задаётся алгебраическим неравенством второй степени с тремя неизвестными  $x, y, z$  в виде  $F(x, y, z) \geq 0$ . Рассматриваются поверхности как замкнутые подмножества евклидова пространства, определяемые описывающей функцией  $F(x, y, z) \geq 0$ , где  $F$  — непрерывная вещественная функция;  $x, y, z$  — задаваемая координатными переменными точка в  $\mathbb{E}^3$ . Поверхности строятся с помощью квадратик и представляются композицией базовой квадрики и возмущений:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z), \quad (1)$$

где  $f_i$  — формфактор;  $R_i(x, y, z)$  — возмущение:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{если } Q_i(x, y, z) \geq 0; \\ 0, & \text{если } Q_i(x, y, z) < 0, \end{cases} \quad (2)$$

здесь  $Q_i(x, y, z)$  — возмущающая квадрика.



Рис. 1. Функционально заданные имплантаты

Геометрическая модель создаёт условия для конструирования объектов и их композиций различной сложности. Для этого используется множество геометрических операций  $\Phi$ , определяемое математически следующим образом [8]:

$$\Phi_j: M^1 + M^2 + \dots + M^n \rightarrow M. \quad (3)$$

Для формирования моделей сложных объектов на базе функций возмущения используются теоретико-множественные операции. Они осуществляются с применением булевых операций объединения и пересечения. Бинарная операция ( $n = 2$ ) (3) объектов  $G_1$  и  $G_2$  означает операцию  $G_3 = \Phi_j(G_1, G_2)$  с определением

$$f_3 = \psi(f_1(\mathbf{X}), f_2(\mathbf{X})) \geq 0, \quad (4)$$

где  $\psi$  — непрерывная вещественная функция двух переменных.

Проведённые экспериментальные исследования показали, что коэффициенты сжатия полигональных объектов в зависимости от тестовых моделей различного уровня детализации варьировались от 10 до 100 и выше [9]. Таким образом, можно сократить базу данных имплантатов (рис. 1).

**Рендеринг функционально заданных поверхностей.** Эта задача напоминает методы визуализации объёмных данных, которые часто применяются в объёмной визуализации (в подобных подходах задана функция плотности). Основным их отличием является наличие дискретных данных. В нашем же случае есть аналитически заданная функция плотности, что позволяет более эффективно осуществлять поиск точек пересечения лучей с поверхностями. Для того чтобы программа работала эффективно, надо учитывать особенности применяемого аппаратного обеспечения. В последнее время увеличение вычислительной мощности происходит за счёт параллельных вычислений. Поэтому метод адаптирован для использования вычислительных ресурсов графического процессора. В [10] описан метод визуализации функционально заданных объектов без рекурсивного деления объектного пространства по четверичному дереву, в котором куб делится на части в плоскости  $XU$  соответственно пикселям на изображении. Такой подход был выбран из-за специфики архитектуры графических процессоров, в которых условные переходы — дорогие операции. Однако при этом приходится сканировать всё экранное пространство лучами, даже если проекция объекта на экран небольшая. Кроме условных переходов, существует проблема балансировки, и накладные расходы на её решение будут превосходить выигрыш в производительности от рекурсивного деления объектного пространства по четверичному дереву. С учётом большого количества вычислительных процессоров одновременно будет происходить проверка сразу нескольких лучей. В большей части графических процессоров, поддерживающих CUDA (программно аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia), содержится не

менее 128 скалярных вычислительных ядер. Следовательно, будет отбрасываться относительно большая часть куба. Тем не менее для небольших и средних объектов такой метод не эффективен. Кроме того, он плохо подходит для комбинированного рендеринга поверхностей и объёмов, что будет отмечено далее.

Поэтому был разработан другой метод [11]. Процесс растривания разбивается на два этапа и распределяется между центральным процессором управления (ЦПУ) и графическим процессором. Центральный процессор выполняет деление объектного пространства по четверичному дереву до клетки определённого размера. Преимущество такого подхода в том, что можно отбросить на ранней стадии большие части куба, в которых нет заданного объекта. Примитивами промежуточного описания являются фрагменты пересечения геометрических объектов с клетками. Фрагментами мы называем части поверхностей (1) и объёмов, принадлежащих данной клетке. Все оставшиеся вычисления выполняются на графическом процессоре.

**Полупрозрачные объёмы.** При сканировании сцены по  $z$ -координате (сканирование объёма в глубину) алгоритм работает до тех пор, пока полностью не будет пройден объём или не накопится определённое значение непрозрачности больше некоторого порогового значения. Отличием от алгоритма трассировки лучей является то, что в целях увеличения скорости вычислений не отслеживаются вторичные лучи. Кроме этого, при моделировании прохождения света через полупрозрачные объёмы пренебрегаем преломлением и затуханием вторичных лучей. В данной модели остаётся только отражение и затухание света на пути следования от объекта к глазу наблюдателя.

Формулу, по которой вычисляется цвет пикселя, можно выразить следующим образом [12]:

$$P_\lambda = \sum_{n=0}^N I_{\lambda n} \Omega_n \prod_{m=0}^{n-1} (1 - \Omega_m), \quad (5)$$

где  $P_\lambda$  — конечный цвет пикселя;  $\lambda$  —  $r$ ,  $g$  или  $b$  цвет (красный, зелёный или синий соответственно);  $I_{\lambda n}$  — интенсивность в  $n$ -м вокселе, вычисленная по модели освещения Фонга;  $\Omega_n$  — непрозрачность  $n$ -го вокселя;  $I_{\lambda 0}$  — отражённый свет от первой точки на луче сканирования;  $I_{\lambda n}$  — цвет фона;  $\Omega_n = 1$ . Преодоление порога можно отследить следующим образом: если на  $k$ -м шаге общая прозрачность  $(1 - \Omega_0)(1 - \Omega_1) \dots (1 - \Omega_{k-1})$  становится меньше некоторого  $\varepsilon$ , то это значит, что вклад всех вокселей, следующих за  $k$ -м вокселем, будет малым, поэтому сканирование можно прекратить.

Существует проблема с пересекающимися прозрачными объектами из-за необходимости выбора области пространства, принадлежащей обоим телам. Как, не решая задачу логического конструирования, выбрать материал области пересечения? Решение этой проблемы состоит в присвоении объектам сцены приоритетов (рис. 2).

В данной работе используются квадрики и объединения квадрик в качестве ограничивающих оболочек для объёмных данных (рис. 3). Механизм автоматического построения



Рис. 2. Примеры изображений объёмов с различными приоритетами

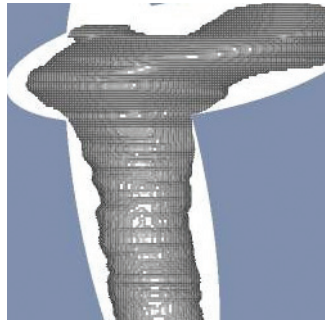


Рис. 3. Трёхмерная текстура, ограниченная оболочкой в виде объединения двух квадрик

оболочки является задачей, сходной по сложности с задачей сегментации, и выходит за рамки данной работы.

**Гибридный рендеринг объёмов и поверхностей.** Метод позволяет визуализировать несколько объёмов в комбинации с функциональными поверхностями. Применяется метод визуализации [11] в качестве базового, который адаптируется к объёмам. Как уже описывалось выше, процесс растривания разбивается на два этапа. Первый этап заканчивается, когда рассматриваемой части соответствует клетка определённого размера. Прimitives промежуточного описания являются фрагменты пересечения геометрических объектов с клетками (уравнения поверхностей). Вторым этапом вычислений — обработка списка объектов, определение видимости и цвета пикселей. Применяется рейкастинг изображения со сложными пересечениями объёмных данных с поверхностями, используется иерархический обход пространства для вычисления цвета в пикселе. Другими словами, процесс обработки пространства таков: объём обходится по четверичному дереву (в экранном пространстве — функция ЦПУ), листьями которого являются двоичные поддерева (соответствует лучам, направленным вглубь экрана, — функция графического процессора).

В процессе двоичного обхода определяется приоритет по  $z$ -координате в случае квадрик, а также для точного вычисления фактора возмущения в случае функционально заданных поверхностей. При делении довольно эффективно отбраковываются большие куски пустого пространства как на уровне четверичного деления, так и двоичного (рис. 4). После нахождения пересечения с оболочкой сканируется её внутренняя часть, заполненная 3D-текстурой.

Объёмные данные хранятся в виде трёхмерной текстуры [13]. Используется 3D-текстурная карта для объёмных наборов данных. Карта текстур — это одна большая 3D-текстура, которая назначается одному текстурному блоку. Максимальный размер текстурной карты зависит только от доступной графической памяти. Все вычисления выполняются на графическом процессоре.

Граф сцены включает операции конструктивной твердотельной геометрии, функционально заданные поверхности с объёмными текстурами и свойствами материала. Внутренние узлы дерева связаны с операциями твердотельной геометрии, листья ссылаются к ограничивающей объём оболочке, к объёмной текстуре и материалу с его свойствами. На рис. 5 показано объединение непрозрачного объекта (имплантата) с несколькими полупрозрачными объёмами. Обработка фрагментов выполняется в параллельно-пиксельном режиме для клеток размером  $8 \times 8$  пикселей и отслеживает прохождение лучей по объёмам, представленным как 3D-текстуры. Для увеличения скорости вычислений используется кэш когерентности текстур. Система рендеринга проходит сцену один раз в отличие

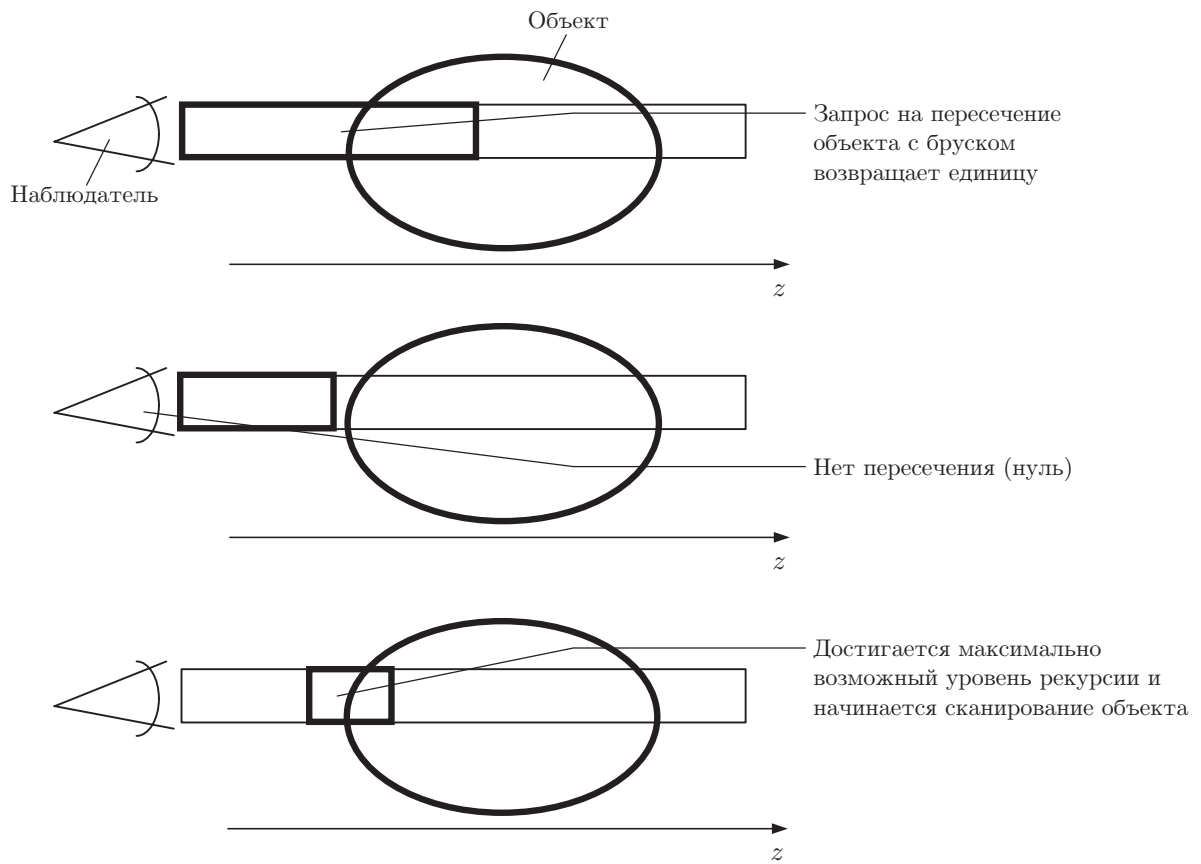


Рис. 4. Двоичное деление луча

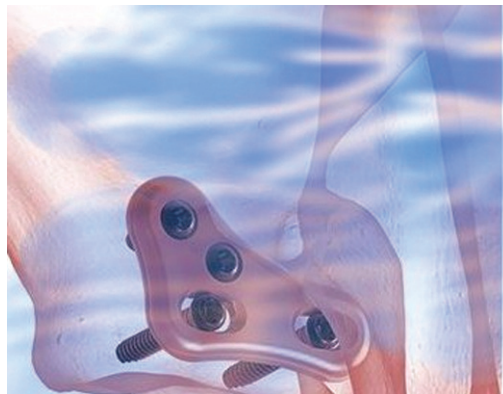


Рис. 5. Визуализация комбинированной анатомической 3D-модели (ткани и костная структура — объёмные данные (3D-текстура), имплантат — функционально заданные поверхности)

от многопроходных методов [14, 15]. Во время бинарного деления луча определяется ближайший к наблюдателю объект и ближайшая точка пересечения с поверхностью, таким образом, обеспечивается сортировка объектов, ближайших к наблюдателю. В результате происходит накопление цвета в пикселе от ближних объектов к дальним, т. е. в прямом приоритетном порядке, в случае если объекты полупрозрачные. Непрозрачные поверхности маскируют более дальние от наблюдателя объекты. Окончательное отслеживание лучей и затенение требуют смешивания нескольких пересекающихся объёмных объектов и эффективного накопления отдельных объёмных вкладов вдоль лучей.

**Результаты работы.** Для оценки производительности сравним предлагаемый метод с известными подходами мультиобъёмной визуализации [14, 15] с использованием графического процессора GTX 470. В табл. 1 сравниваются частоты кадров (количество кадров в секунду), достигнутые для медицинских сцен с несколькими объёмами, которые используют несколько режимов и сканы для различных частей тела. Тесты 1 и 2 показывают разрешение объёмов и время анатомического сканирования (рис. 6). Тест 3 представляет собой сочетание анатомического сканирования объёмов с высоким разрешением и ISO-поверхностей для методов [14, 15]. В предлагаемом методе используется комбинация объёма и функциональной поверхности. Тест 4 — это объёмы в сочетании с полигональной моделью размером  $3,87 \cdot 10^3$  в качестве аналога для функциональной поверхности.

По сравнению с известными подходами основные преимущества предлагаемого метода включают в себя:

1. **Общность:** с помощью данного метода можно обрабатывать не только функционально заданные поверхности, но и полупрозрачные объёмы.

Таблица 1

Номер теста	Объёмы	Полигоны	[14]	[15]	Предлагаемый метод
1	$1 \times 512^3$	—	14–46	9–17	30
2	$2 \times 256^3$	—	24–38	6–12	37
3	$4 \times 256^3$	—	5–9	1–3	25
4	$10 \times 64^3$	$3,87 \cdot 10^3$	—	—	16

**Примечание:** в предлагаемом методе рассматриваются функционально заданные объекты, в тесте 4 их пять.

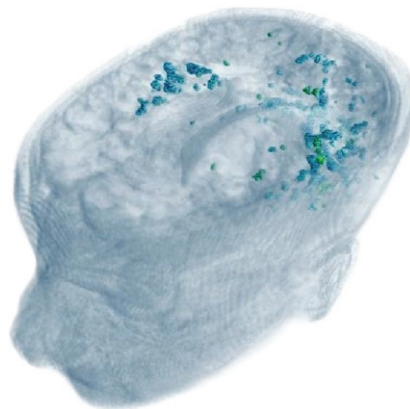


Рис. 6. Мультиобъёмные данные: череп, мозг, опухоль (верхняя часть мозга) и ткани головы

2. Сложные модели: в методе предусмотрена визуализация большого числа объёмов, комплекс полупрозрачных и функционально заданных объектов, сложных полупрозрачных объёмов, в том числе пересечений объёмов.

3. Эффективность: использование функционально заданных поверхностей позволяет сократить базу данных имплантатов и уменьшить время рендеринга.

4. Простота: метод относительно прост в реализации и не имеет проблем с надёжностью или вырождением поверхностей.

5. Хорошее приближение: предварительное применение метода для разных тестовых образцов показывает хорошее приближение с ограниченной ошибкой на границах пересечений объёмов и функционально заданных поверхностей.

**Заключение.** В данной работе создан новый подход к визуализации нескольких полупрозрачных объёмов и функционально заданных поверхностей. Метод, включающий стадию растеризации, реализован на CUDA, что позволяет полностью контролировать иерархию памяти, в частности доступ к высокой пропускной способности и низкому времени ожидания распределённой памяти. Обеспечивается интерактивная частота кадров при одновременном отображении более ста произвольно перекрывающихся объёмов. Вычисление занимает несколько десятков миллисекунд на процессорах Intel Core2 CPU E8400 3.0 GHz и GPU GTX 470.

Метод актуален для широкого спектра применения объёмного рендеринга сложных сцен, включая пересекающиеся объёмы и поверхности для медицинских приложений. Как показано в данной работе, метод является простым в реализации и позволяет использовать его для решения многих задач.

## СПИСОК ЛИТЕРАТУРЫ

1. **Kruger J., Westermann R.** Acceleration techniques for GPU-based volume rendering // Proc. of the 14th IEEE Conference on Visualization. Seattle, USA, 19–24 Oct., 2003. P. 38–43. DOI: 10.1109/VISUAL.2003.1250384.
2. **Stegmaier S., Strengert M., Klein T., Ertl T.** A simple and flexible volume rendering framework for graphics-hardware-based ray casting // Proc. of the 4th Eurographics / IEEE VGTC Conference on Volume Graphics. New York, USA, 20–21 June, 2005. P. 187–195. DOI:10.1109/VG.2005.194114.
3. **Avants B. B., Tustison N. J., Song G. et al.** A reproducible evaluation of Ants similarity metric performance in brain image registration // Neuroimage. 2011. **54**, N 3. P. 2033–2044. DOI: 10.1016/j.neuroimage.2010.09.025.
4. **Killiany R. J., Gomez-Isla T., Moss M. et al.** Use of structural magnetic resonance imaging to predict who will get Alzheimer’s disease // Ann. Neurology. 2000. **47**, N 4. P. 430–439.
5. **Leu A., Chen M.** Modelling and rendering graphics scenes composed of multiple volumetric datasets // Comput. Graph. Forum. 1999. **18**, Iss. 2. P. 159–171. DOI: 10.1111/1467-8659.00366.
6. **Grimm S., Bruckner S., Kanitsar A., Groller M. E.** Flexible direct multi-volume rendering in interactive scenes // Proc. of the Vision, Modeling, and Visualization Conference (VMV 2004). Stanford, USA, 16–18 Nov., 2004. P. 386–379.
7. **Вяткин С. И.** Моделирование сложных поверхностей с применением функций возмущения // Автометрия. 2007. **43**, № 3. С. 40–47.
8. **Вяткин С. И.** Преобразования функционально заданных форм // Программные системы и вычислительные методы. 2014. № 4. С. 484–499.
9. **Вяткин С. И., Долговесов Б. С.** Метод сжатия геометрических данных с применением функций возмущения // Автометрия. 2018. **54**, № 4. С. 18–25. DOI: 10.15372/AUT20180403.



10. **Вяткин С. И.** Метод бинарного поиска элементов изображения функционально заданных объектов с применением графических акселераторов // *Автометрия*. 2014. **50**, № 6. С. 89–96.
11. **Вяткин С. И.** Метод рекурсивного поиска элементов изображения функционально заданных поверхностей // *Автометрия*. 2017. **53**, № 3. С. 53–57. DOI: 10.15372/AUT20170307.
12. **Knittel G.** VERVE: voxel engine for real-time visualization and examination // *Comput. Graph. Forum*. 1993. N 3. P. 37–48. DOI: 10.1111/1467-8659.1230037.
13. **Vyatkin S. I., Dolgovesov B. S.** A 3D texture-based recursive multi-level ray casting algorithm // *Proc. of the 2nd IASTED Intern. Multi-Conference on Automation, Control, and Information Technology “Software Engineering” (ACIT 2005)*. Novosibirsk, Russia, 20–24 June, 2005. P. 92–97.
14. **Roessler F., Botchen R. P., Ertl T.** Dynamic shader generation for GPU-based multi-volume ray casting // *IEEE Comput. Graph. Applications*. 2008. **28**, N 5. P. 66–77. DOI: 10.1109/MCG.2008.96.
15. **Brecheisen R., Platel B., Vilanova A., Romenij B. T. H.** Flexible GPU-based multi-volume ray-casting // *Proc. of the Vision, Modelling and Visualization Conference*. Konstanz, Germany, 8–10 Oct., 2008. P. 1–6.

*Поступила в редакцию 16.12.2020*

*После доработки 19.02.2021*

*Принята к публикации 04.03.2021*

---