

Алгоритм синтеза трехмерных фрактальных динамических структур

А. А. Теплов, К. А. Майков

Московский государственный технический университет имени Н. Э. Баумана
105005, Москва, ул. 2-я Бауманская, 5, стр. 1

e-mail: teploff.aa@gmail.com; maikov@bmstu.ru

Аннотация. Приведены результаты сравнительного анализа практически востребованных алгоритмов формирования двумерных фракталов, обладающих высоким быстродействием и низкой ресурсоемкостью. Обосновывается практическая целесообразность предложенного алгоритма синтеза трехмерных фрактальных структур, представляющего из себя функциональное единство алгоритмов формирования двоичного дерева и кривой «дракона». Результаты проведенных численных экспериментальных исследований подтвердили работоспособность предлагаемого алгоритма.

Ключевые слова: трехмерные фракталы, виртуальная реальность, формирование трехмерных фрактальных структур.

1. Введение

В настоящее время в технологии совмещенной реальности [1] существует проблема формирования объектов реального мира, отличающихся нерегулярностью структуры и ресурсоемкостью вычислений. Примерами таких объектов являются структуры земной и водной поверхности, растительный покров, сердечно-сосудистая система человека и т. д. Применение методов и объектов классической геометрии для формирования упомянутых объектов является трудоемким и порой нереализуемым [2]. Предложенная в [3] фрактальная теория принципиально способна справиться с поставленной проблемой, однако методы ее реализации представляют собой трудоемкий процесс и доведены до стадии практической применимости лишь для синтеза простейших двумерных структур. Поэтому на практике для решения задач совмещенной реальности возникает необходимость определения прототипа метода формирования динамических, трехмерных фрактальных структур, обладающего практически приемлемыми быстродействием и ресурсоемкостью.

2. Алгоритмы формирования геометрических фракталов

Среди известных алгоритмов формирования фрактальных структур можно выделить алгоритмы построения геометрических фракталов. На практике данные алго-

ритмы являются наиболее предпочтительными и функционально востребованными, так как обладают практически приемлемым быстродействием и низкой ресурсоемкостью [1]. Практически востребованные алгоритмы геометрических фракталов можно классифицировать с точки зрения древовидности структуры. К классу древовидных алгоритмов могут быть отнесены следующие алгоритмы: множество Кантора [4]; Пифагорово дерево [1]; двоичное дерево [3].

К классу недревовидных алгоритмов построения геометрических фракталов могут быть отнесены такие алгоритмы, как: кривая Коха [4]; кривая «дракона» [3]; кривая Гильберта [2]; кривая Минковского [3]; «салфетка» Серпинского [2]; «ковер» Серпинского [2]; «кладбище» Серпинского [2]. Классификация алгоритмов построения геометрических фракталов, основанная на критерии древовидности структуры, представлена на рис. 1.

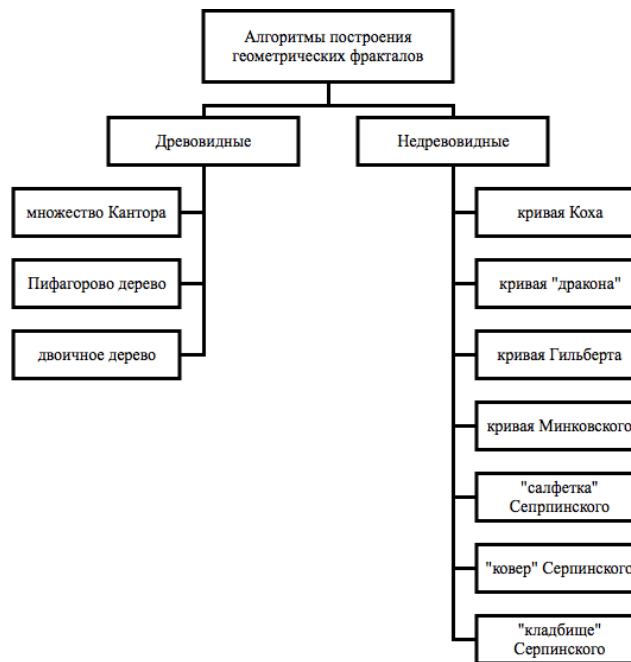


Рисунок 1. Классификация алгоритмов построения геометрических фракталов, основанная на критерии древовидности структуры

Также классификация алгоритмов может быть проведена на основе ограниченности структуры. К классу алгоритмов ограниченной структуры могут быть отнесены алгоритмы: множество Кантора; «кладбище» Серпинского; «салфетка» Серпинского; «ковер» Серпинского.

К классу алгоритмов неограниченной структуры могут быть отнесены следующие алгоритмы: Пифагорово дерево; двоичное дерево; кривая Коха; кривая «дракона»; кривая Гильберта; кривая Минковского.

Классификация алгоритмов построения геометрических фракталов, основанная на критерии ограниченности структуры, представлена на рис. 2.

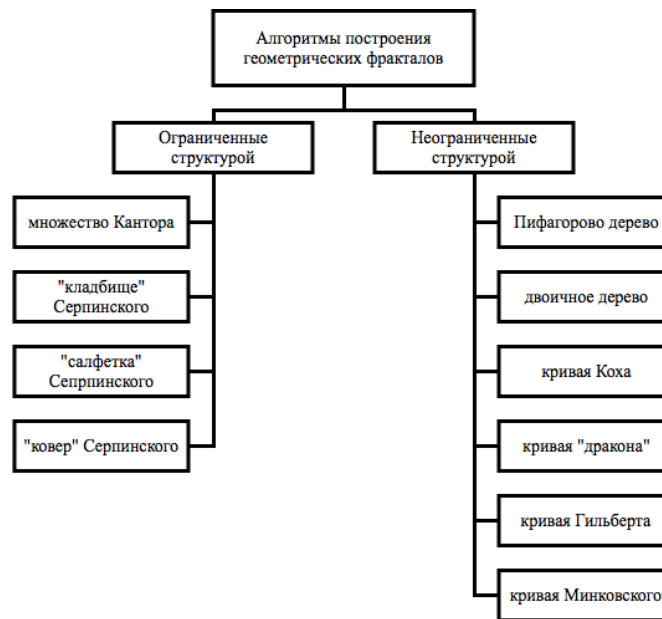


Рисунок 2. Классификация алгоритмов построения геометрических фракталов, основанная на критерии ограниченности структуры

Проведем анализ функциональных и структурных особенностей основных алгоритмов формирования геометрических фракталов.

3. Методика вычисления сложности алгоритмов построения двумерных геометрических фракталов и фрактальной размерности структурных характеристик

Все перечисленные в п. 2 алгоритмы построения геометрических фракталов имеют рекурсивный характер, что вытекает из главного фрактального свойства — самоподобия [5]. Метод подстановки (итераций) [6] является широко используемым методом определения алгоритмической сложности рекурсивных алгоритмов и заключается в последовательной замене рекуррентной части в выражении для получения новых формализаций. Замена производится до тех пор, пока не будет достигнуто формализованное представление общего принципа в виде нереккуррентной форму-

лы, после чего необходимо произвести доказательство полученной формулы методом математической индукции [7].

Одной из важных характеристик фрактальной структуры является ее фрактальная размерность [2], показывающая, насколько плотно и равномерно элементы данного множества заполняют евклидово пространство ($D < E$). Фрактальная размерность определяется следующей формулой:

$$D = -\lim_{\varepsilon \rightarrow \infty} \frac{\ln N_\varepsilon}{\ln \varepsilon}, \quad (1)$$

где N_ε — минимальное количество n -мерных окружностей радиуса ε , необходимых для покрытия множества.

Применим метод подстановки для анализа вышеупомянутых алгоритмов, а также вычислим фрактальную размерность для каждой из структур.

3.1. Анализ алгоритмов формирования геометрических фракталов

Проведя сравнительный анализ алгоритмов формирования геометрических фракталов, можно составить результирующую таблицу (см. табл. 1).

Таблица 1. Результаты сравнительного анализа алгоритмов формирования геометрических фракталов

Наименование	Алгоритмическая сложность, $O(N)$	Фрактальная размерность, D	Ограниченность
Пифагорово дерево	$O(2^n)$	2	Нет
двоичное дерево	$O(2^n)$	2	Нет
«кладбище» Серпинского	$O(4^n)$	1.262	Да
«ковер» Серпинского	$O(8^n)$	1.893	Да
кривая Гильберта	$O(4^n)$	2	Нет
кривая Коха	$O(4^n)$	1.262	Нет
кривая Минковского	$O(8^n)$	1.5	Нет
кривая «дракона»	$O(2^n)$	1.262	Нет
множество Кантора	$O(2^n)$	0.63	Да
«салфетка» Серпинского	$O(3^n)$	1.585	Да

Из табл. 1. видно, что такие фрактальные алгоритмы, как кривая «дракона», Пифагорово дерево и двоичное дерево, имеют приемлемую алгоритмическую сложность в виде: $O(2^n)$ среди своих представителей и удовлетворяют критериям высокого быстродействия, низкой ресурсоемкости, не обладают органичностью структуры, имеют высокую фрактальную размерность в виде: $D = 2$, причем, такие

алгоритмы, как: Пифагорово дерево и двоичное дерево имеют древовидную структуру.

3.2. Выбор прототипа

Практические особенности задачи построения динамических трехмерных древовидных объектов определяют такие требования к алгоритмам формирования фракталов, как высокое быстродействие и низкая ресурсоемкость, а также практическую возможность отображения многообразия фрактальной структуры, свойствами которой являются: нерегулярность, квазипериодичность и высокая масштабная инвариантность. Как следует из приведенных выше результатов анализа, рассмотренные алгоритмы решают лишь задачу формирования двумерных фракталов и не удовлетворяют в полной мере предложенным требованиям.

Поэтому возникает практическая необходимость построения алгоритма решения поставленной проблемы, удовлетворяющего следующими функциональными критериями, такими как:

- приемлемая алгоритмическая сложность;
- древовидная структура;
- высокая фрактальная размерность;
- высокая детализация структуры.

В связи с вышеизложенным представляется практически необходимым построить прототип метода, который представляет собой функциональное единство описанных выше алгоритмов, таких как кривая «дракона» и двоичное дерево, что позволяет сформировать трехмерную древовидную квазипериодичную структуру. Алгоритм двоичного дерева придаст структуре древовидный характер, а кривая «дракона» обеспечит приемлемую масштабную инвариантность каждой из компонент вычисленного дерева.

3.3. Алгоритм синтеза трехмерных фрактальных динамических структур

Схема алгоритма синтеза трехмерных фрактальных динамических структур представлена на рис. 3. Этапы формирования трехмерных структур алгоритмами двоичного дерева и кривой «дракона» являются наиболее ресурсоемкими и представляют особый интерес в представленной схеме. Так как речь идет о формировании динамических трехмерных структур, необходимо создать асинхронный алгоритм, при помощи которого возможно организовать построение структур «на лету». Данный подход позволит параллельно выполнять мелкие задачи, избегая синхронного блокирующего выполнения трудоемкого процесса.

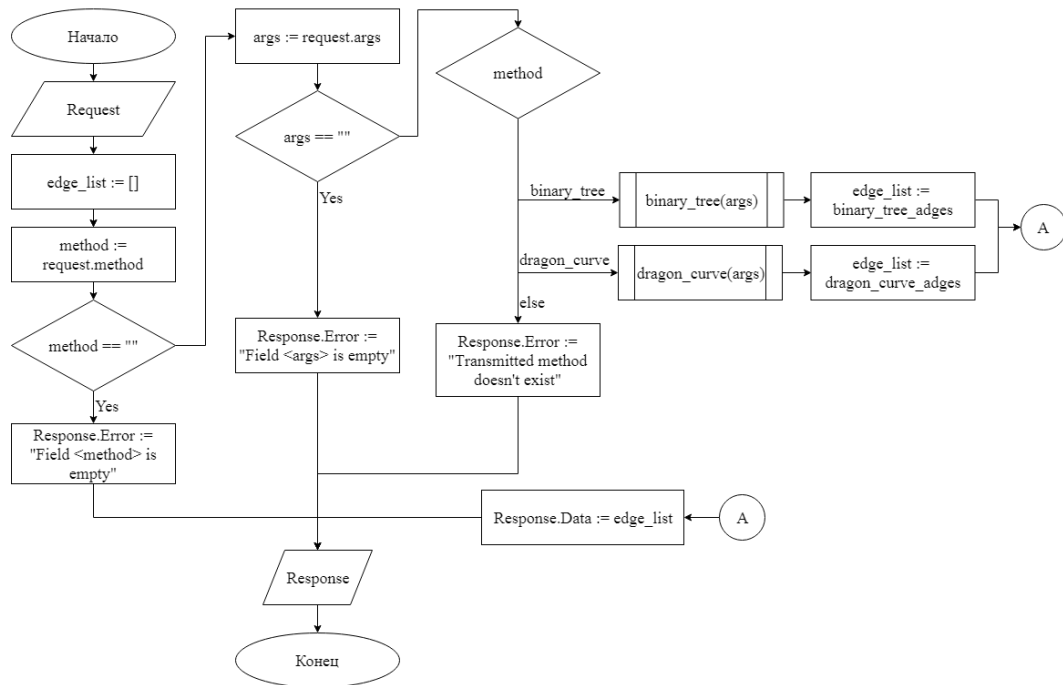


Рисунок 3. Схема алгоритма формирования трехмерных фрактальных динамических структур

Предложенный алгоритм декомпозируется на два асинхронных этапа:

- формирование трехмерной структуры при помощи алгоритма построения двоичного дерева;
- формирование трехмерной структуры с заданной степенью инвариантности при помощи алгоритма построения кривой «дракона».

Главной отличительной особенностью этапа формирования трехмерной структуры алгоритмом построения двоичного дерева является представление результирующей структуры в виде определенной степени иерархии. Данному алгоритму присущ рекурсивный характер, что вытекает из главного фрактального свойства — самоподобия [5]. Результатом работы данного этапа является список трехмерных ребер или сообщение об ошибке в случае некорректно-переданных аргументов. Под аргументами понимаются некоторые геометрические признаки фрагментов трехмерной структуры на заданной величине иерархии. Схема этапа формирования трехмерной структуры алгоритмом построения двоичного дерева представлена на рис. 4.

Как и предыдущий этап, этап формирования трехмерной структуры с заданной степенью инвариантности при помощи алгоритма построения кривой «дракона» по своему характеру рекурсивен. Однако главной отличительной чертой данного этапа

является представление структуры с заданной масштабной инвариантностью. Некоторые фрактальные структуры целесообразно описать с помощью L-системы [8], главным свойством которой является задание самоподобной структуры в виде строки правил. На основе полученной строки необходимо сформировать трехмерные ребра. Соответственно, чем выше степень детализации, тем больше результирующая строка L-системы [8]. Схема этапа формирования трехмерной структуры с заданной степенью инвариантности при помощи алгоритма построения кривой «дракона» представлена на рис. 5.

Предполагается, что реализация алгоритма будет иметь клиент-серверную архитектуру, при этом серверу будет присущ неблокирующий [9] характер, для того чтобы удовлетворить требованиям режима реального времени. Примером реализаций таких серверов могут быть Tornado [9], Gorilla [10], Node.js [11] и т. д.

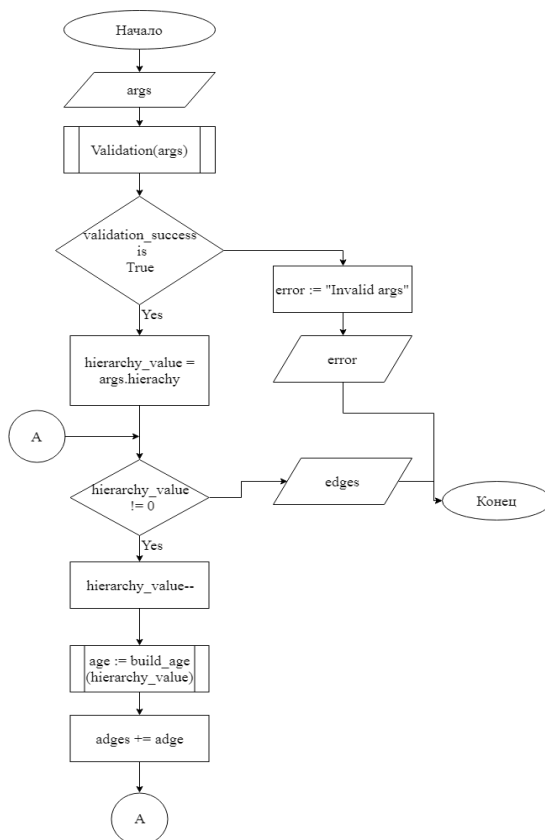


Рисунок 4. Этап формирования трехмерной структуры алгоритмом построения двоичного дерева

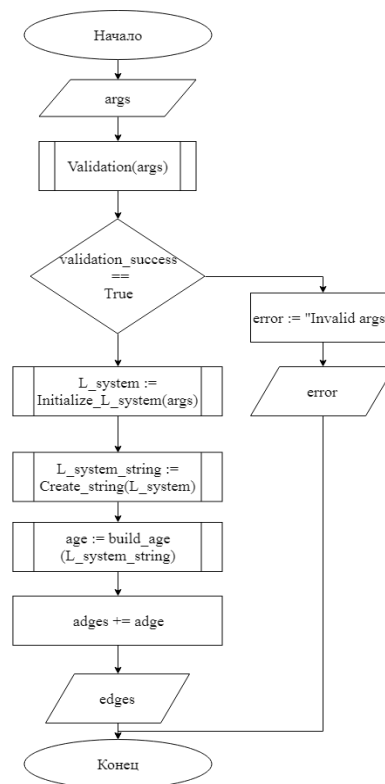


Рисунок 5. Этап формирования трехмерной структуры с заданной степенью инвариантности при помощи алгоритма построения кривой «дракона»

3.4. Численные экспериментальные исследования

Исследования проводились на аппаратной платформе со следующими характеристиками: четырехъядерный процессор Intel i5 3.2Ghz; оперативная память 16 Gb; операционная система Ubuntu 18.04 [12].

Программная реализация алгоритма синтеза трехмерных фрактальных динамических структур выполнена на языке программирования Python третьей версии [13] при помощи развитых интегрируемых библиотек, таких как Matplotlib [14], NumPy [15], SciPy [16], Turtle [18], предназначенных для осуществления научных и инженерных расчетов с последующей визуализацией результатов.

Численные экспериментальные исследования алгоритма синтеза трехмерных фрактальных динамических структур. Входные данные для экспериментальных исследований представлены последовательностью значений от 1 до 30 единиц величин иерархии структуры и от 1 до 20 единиц значений масштабной инвариантности, что определяется особенностью отображения трехмерной структуры с практически приемлемой величиной масштабной инвариантности [3]. Экспериментальные исследования представляли собой серию из ста реализаций вычислительных экспериментов для каждого из вышеуказанного значения диапазона.

Исследование зависимости времени формирования трехмерной древовидной структуры фрактальным алгоритмом «двоичное дерево» от значения величины иерархии. Обработанные результаты проведенных экспериментальных исследований зависимости времени формирования трехмерной древовидной структуры фрактальным алгоритмом «двоичное дерево» от значения величины иерархии представлены на рис. 6.

Поскольку задача формирования трехмерных древовидных структур в режиме реального времени характеризуется временными значениями, не превышающими 0.03 секунды [3], то особый интерес представляют результаты, удовлетворяющие данному временному значению. На рис. 7 представлена зависимость времени формирования трехмерной древовидной структуры фрактальным алгоритмом «двоичное дерево» от значения величины иерархии в диапазоне времени до 0.03 секунды.

Из представленной на рис. 7 зависимости видно, что при величине иерархии, не превышающей значение семи единиц, алгоритм удовлетворяет указанным выше требованиям.

Исследование зависимости времени формирования трехмерной структуры фрактальным алгоритмом «кривая дракона» от значения масштабной инвариантности. Обработанные результаты проведенных экспериментальных исследований зависимости времени формирования трехмерной структуры фрактальным алгоритмом «кривая дракона» от значения масштабной инвариантности представлены на рис. 8.

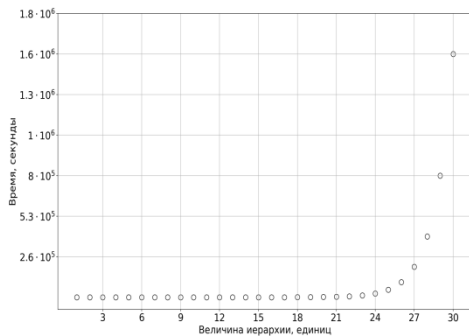


Рисунок 6. Зависимость времени формирования трехмерной древовидной структуры фрактальным алгоритмом «двоичное дерево» от значения величины иерархии

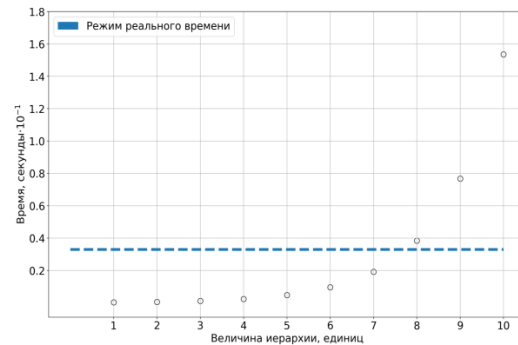


Рисунок 7. Зависимость времени формирования трехмерной древовидной структуры фрактальным алгоритмом «двоичное дерево» от значения величины иерархии в диапазоне времени до 0.03 секунды

Задача вычисления трехмерных структур с заданной масштабной инвариантностью в режиме реального времени также характеризуется временными значениями, не превышающими 0.03 секунды [3]. Следовательно, особый практический интерес возникает к результатам, время вычисления которых удовлетворяет данному временному интервалу. Из представленной на рис. 9 зависимости видно, что при масштабной инвариантности не более шести единиц алгоритм удовлетворяет вышеуказанному условию.

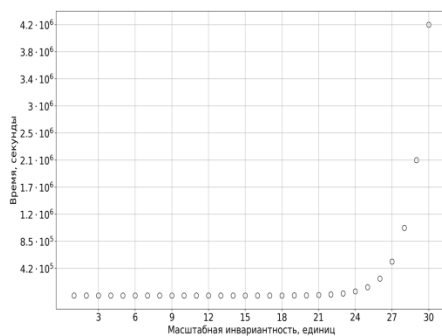


Рисунок 8. Зависимость времени формирования трехмерной структуры фрактальным алгоритмом «кривая дракона» от значения масштабной инвариантности

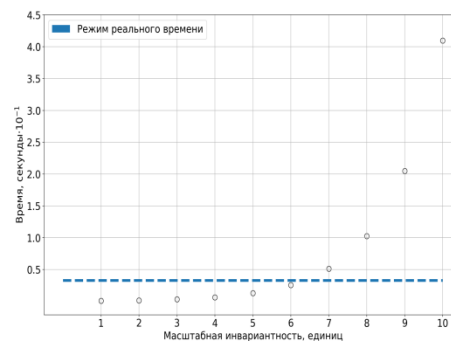


Рисунок 9. Зависимость времени формирования трехмерной структуры фрактальным алгоритмом «кривая дракона» от значения масштабной инвариантности в диапазоне времени до 0.03 секунды

При увеличении значений величины иерархии и величины масштабной инвариантности более чем на семи и шести единиц, соответственно, возникает необхо-

димось дальнейшего развития предложенного алгоритма. Одним из путей развития является применение технологий параллельных вычислений. Реализация такого технологического развития алгоритма определяет необходимость применения иных платформ программной разработки, использующих С-подобные [18] языки программирования.

4. Выводы

Проведенные численные экспериментальные исследования алгоритма синтеза трехмерных фрактальных динамических структур подтвердили работоспособность разработанных алгоритмических структур и их практически приемлемую ресурсоемкость. Анализ результатов численных экспериментов показал, что для величины иерархии и величины масштабной инвариантности, не превышающих семи и шести единиц, соответственно, алгоритм удовлетворяет требованиям реального времени.

В результате проведенного сравнительного анализа практически востребованных алгоритмов формирования двумерных фракталов показано, что рассмотренные алгоритмы не удовлетворяют практическим требованиям и решают лишь задачу формирования двумерных структур, что не является приемлемым в технологии совмещенной реальности. Этот факт определяет практическую необходимость построения прототипа метода, который бы решал поставленную проблему. Предложен прототип метода, который сочетает в себе единство алгоритмов двоичного дерева и «кривой «ракона». Предложен асинхронный алгоритм синтеза трехмерных фрактальных динамических структур, отличительной особенностью которого является формирование динамических, квазипериодических структур в режиме реального времени и который обладает приемлемой алгоритмической сложностью и ресурсоемкостью. Результаты проведенных численных экспериментальных исследований подтвердили работоспособность предлагаемого алгоритма.

Литература

- [1] Балханов В. К. Основы фрактальной геометрии и фрактального исчисления. — Улан-Удэ : Изд-во Бурятского государственного университета, 2013.
- [2] Федер Е. Фракталы : пер. с англ. — М. : Мир, 1991.
- [3] Мандельброт Б. Фрактальная геометрия природы. — М. : Институт компьютерных исследований, 2002.
- [4] Морозов А. Д. Введение в теорию фракталов. — М.-Ижевск : Институт компьютерных исследований, 2002.
- [5] Шредер М. Фракталы, хаос, степенные законы. Миниатюры из бесконечного рая. — Ижевск : НИЦ «Регулярная и хаотическая динамика», 2001.

- [6] *Кормен Т. Х., Лейзерон Ч. И., Ривест Рю Л., Штайн К.* Алгоритмы: построение и анализ : пер. с англ. — 3-е изд. — М. : Вильямс, 2013.
- [7] *Кнут Д. Э.* Искусство программирования. Т. 3. Сортировка и поиск. — М. : Вильямс, 2008.
- [8] L-система [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/L-система>.
- [9] Tornado. [Электронный ресурс] URL: <http://www.tornadoweb.org/en/stable/#>
- [10] Gorilla. [Электронный ресурс] URL: <https://github.com/gorilla>
- [11] Node.js. [Электронный ресурс] URL: <https://nodejs.org/en/>
- [12] Ubuntu. [Электронный ресурс] URL: <https://www.ubuntu.com/>.
- [13] Python 3 версия. [Электронный ресурс] URL: <https://www.python.org/>.
- [14] Matplotlib. [Электронный ресурс] URL: <https://matplotlib.org/>.
- [15] NumPy. [Электронный ресурс] URL: <http://www.numpy.org/>.
- [16] SciPy. [Электронный ресурс] URL: <https://www.scipy.org/>
- [17] Turtle. [Электронный ресурс] URL: <https://docs.python.org/3/library/turtle.html>
- [18] *Ахо А., Сети Р., Ульман Дж.* Компиляторы. Принципы, технологии, инструменты. — М. : Вильямс, 2001.

Авторы:

Александр Александрович Теплов — аспирант, кафедра «Программное обеспечение ЭВМ и информационные технологии», Московский государственный технический университет имени Н.Э. Баумана

Константин Анатольевич Майков — доктор технических наук, профессор, профессор кафедры «Программное обеспечение ЭВМ и информационные технологии», Московский государственный технический университет имени Н.Э. Баумана

Comparative analysis of Delaunay's triangulation algorithms for the construction of three-dimensional fractal objects

A.A. Teplov, K.A. Maikov

Bauman Moscow State Technical University Baumanskaya 2-ya Str., 5, Moscow, Russia, 105005
e-mail: teploff.aa@gmail.com; maikov@bmstu.ru

Abstract. The comparative analysis results of the virtually popular algorithms of two-dimensional fractals construction with high performance and low resource consumption are described in this article.

Keywords: construction of three-dimensional fractal structures, triangulation on a given cell array, Delaunay's triangulation.

References

- [1] Balhanov V. K. (2013) *Osnovy fraktal'noj geometrii i fraktal'nogo ischisleniya*. Ulan-Udeh. [in Rus]
- [2] Feder E. (1991) *Fraktaly*. Moscow, Mir. [in Rus]
- [3] Mandel'brot B. (2002) *Fraktal'naya geometriya prirody*. Moscow. [in Rus]
- [4] Morozov A. D. (2002) *Vvedenie v teoriyu fraktalov*. Moscow, Izhevsk. [in Rus]
- [5] Shreder M. (2001) *Fraktaly, kaos, stepennye zakony. Miniatyury iz beskonechnogo raya*. [in Rus]
- [6] Kormen T. H., Lejzeron CH. I., Rivest Ryu L., Shtajn K. (2013) *Algoritmy: postroenie i analiz* (3 eds.) Moscow. [in Rus]
- [7] Knut D. E. (2008) *Iskusstvo programmirovaniya* (Vol. 3, Sortirovka i poisk) Moscow.
- [8] *L-sistema* (2018) Retrieved from <https://ru.wikipedia.org/wiki/L-система> [in Rus]
- [9] *Tornado* (2018) Retrieved from <http://www.tornadoweb.org/en/stable/#>
- [10] *Gorilla*. (2018) Retrieved from <https://github.com/gorilla>
- [11] *Node.js* (2018) Retrieved from <https://nodejs.org/en/>
- [12] *Ubuntu* (2018) Retrieved from <https://www.ubuntu.com/>.
- [13] *Python 3 Release*. Retrieved from <https://www.python.org/>.
- [14] *Mathplotlib*. Retrieved from <https://matplotlib.org/>.
- [15] *NumPy* (2018) Retrieved from <http://www.numpy.org/>.
- [16] *SciPy* (2018) Retrieved from <https://www.scipy.org/>
- [17] *Turtle* (2018) Retrieved from <https://docs.python.org/3/library/turtle.html>
- [18] Aho A., Seti R., Ulman J. (2001) *Kompilyatory. Principy, tekhnologii, instrumenty*. Moscow. [in Rus]