

<https://doi.org/10.21869/2223-1536-2023-13-4-115-130>



УДК 004.93'1; 004.932

Метод и алгоритм реализации множества Мандельброта для обработки сложноструктурируемых изображений

Р. А. Томакова¹ ✉, И. М. Ахмадулин¹, Н. Г. Нефедов¹,
Е. И. Пузырев¹, А. А. Малышев¹

¹ Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

✉ e-mail: rtomakova@mail.ru

Резюме

Цель исследований. В нашей жизни фракталы распространены повсеместно, например, в природе, в ней встречается огромное количество фигур, подобных самим себе и построенных по определенным законам. Фракталы используют в компьютерной графике, экономике, радиотехнике, физике и многих других областях. Применение фракталов в различных областях науки способно открыть множество новых возможностей. Поэтому разработка программного продукта, предназначенного для визуализации множества Мандельброта, является актуальной задачей.

Цель исследований заключается в разработке программного продукта, предназначенного для обработки сложноструктурируемых изображений на основе фрактальных методов и алгоритмов визуализации множества Мандельброта.

Методы. Для реализации программного продукта с целью визуализации множества Мандельброта был использован язык программирования Java. В качестве локального хранилища данных была использована БД SQLite. Также были применены библиотеки: JavaFX – для создания пользовательского интерфейса; SQLite JDBC 3.21.0 – для работы с БД SQLite; imgscalr – с целью масштабирования изображений; Apache Commons IO 2.6 для работы с файловой системой; ControlsFX v8.40.14 – для создания интерфейса пользователя.

Результаты. Разработанный программный продукт на основе фрактальных методов позволяет существенно выделить текстурные особенности изображений, на основании которых производится операция кластеризация выделенных объектов интереса на изображениях с целью экспортирования для дальнейшей их классификации.

Заключение. В ходе выполнения проекта реализован программный продукт для визуализации множества Мандельброта, который может быть использован для моделирования объектов, имеющих фрактальную форму, применительно к обработке сложноструктурируемых изображений, содержащих невидимые скрытые особенности объектов на изображениях.

Ключевые слова: фрактал; множество Мандельброта; последовательность; оверсемплинг.

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Метод и алгоритм реализации множества Мандельброта для обработки сложноструктурируемых изображений / Р. А. Томакова, И. М. Ахмадулин, Н. Г. Неведов, Е. И. Пузырев, А. А. Малышев // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2023. Т. 13, № 4. С. 115–130. <https://doi.org/10.21869/2223-1536-2023-13-4-115-130>.

Поступила в редакцию 09.10.2023

Подписана в печать 05.11.2023

Опубликована 22.12.2023

A Method and Algorithm for Implementing the Mandelbrot Set for Processing Complex Structured Images

Rimma A. Tomakova¹ ✉, Ildar M. Akhmadulin¹, Nikita G. Nefedov¹,
Evgeny I. Puzyrev¹, Anton A. Malyshev¹

¹ Southwest State University
50 Let Oktyabrya Str. 94, Kursk 305040, Russian Federation

✉ e-mail: rtomakova@mail.ru

Abstract

The purpose of research. In our life, fractals are ubiquitous, for example, in nature, there are a huge number of figures similar to themselves and built according to certain laws. Fractals are used in computer graphics, economics, radio engineering, physics and many other fields. The use of fractals in various fields of science can open up many new possibilities. Therefore, the development of a software product designed to visualize the Mandelbrot set is an urgent task.

The purpose of the research is to develop a software product designed for processing complex structured images based on fractal methods and algorithms for visualizing the Mandelbrot set.

Methods. The Java programming language was used to implement the software product in order to visualize the Mandelbrot set. The SQLite database was used as a local data store. Libraries were also used: JavaFX – to create a user interface; SQLite JDBC 3.21.0 – to work with the SQLite database; imgscalr – to scale images; Apache Commons IO 2.6 to work with the file system; ControlsFX v8.40.14 – to create a user interface.

Results. The developed software product based on fractal methods makes it possible to significantly highlight the textural features of images, on the basis of which the clustering operation of selected objects of interest in images is performed in order to export them for further classification.

Conclusion. During the implementation of the project, a software product for visualization of the Mandelbrot set was implemented, which can be used to model objects having a fractal shape, in relation to the processing of complexly structured images containing invisible hidden features of objects in the images.

Keywords: fractal; Mandelbrot set; sequence; oversampling.

Conflict of interest: The Authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Tomakova R. A., Ahmadulin I. M., Nefedov N. G., Puzyrev E. I., Malyshev A. A. A Method and Algorithm for Implementing the Mandelbrot Set for Processing Complex Structured Images. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie* = *Proceedings of the Southwest State University. Series: Control, Computer Engineering, Information Science. Medical Instruments Engineering*. 2023; 13(4): 115–130. (In Russ.) <https://doi.org/10.21869/2223-1536-2023-13-4-115-130>.

Received 09.10.2023

Accepted 05.11.2023

Published 22.12.2023

Введение

Появление и развитие компьютерной техники за последние десятилетия кардинально изменило практически все сферы человеческой жизни. Компьютерные системы повсеместно используются в производстве, образовании, медицине, искусстве и т. д.

Появились также виды деятельности, которые не могли бы существовать без возможностей использования вычислительной техники. Одним из них является создание фрактальных изображений (фракталов), предназначенных для использования в различных областях знаний [1; 2; 3; 4]. В настоящее время широко внедряются фрактальные методы применительно для идентификации и распознавания объектов на изображениях по яркостным характеристикам в тех случаях, когда классические методы не дают значимых результатов. В работах [5; 6; 7] показано использование фракталов для решения задач обработки текстурных изображений с целью выделения границ анализируемых объектов на сложноструктурируемых изображениях, полученных при аэрокосмической съемке, медицинской рентгенографии, а также при анализе порошков и пористых материалов. В основу обработки изображений с использованием фрактальных методов положен принцип подобия цветных текстур, основанный на статистических характеристиках анализируемых пространственных объектов с учетом информативных признаков яркости и цвета. Следует заметить, что оценка фрактальных текстур при сегментации изображений [8; 9; 10] осуществляется с учетом их размеров и

масштаба по изменениям анализируемых характеристик в исследуемом итерационном процессе.

Одним из наиболее известных видов фракталов является множество Мандельброта [10; 11; 12]. Фрактальное изображение, полученное на его основе, по сути является раскрашенным участком комплексной плоскости. Создание такого изображения в приемлемом качестве требует большого количества вычислений.

Множество Мандельброта [12; 13] определяется с помощью рекуррентной формулы:

$$Z_{n+1} = Z_n^k + C, \quad (1)$$

где Z и C – комплексные числа, причём Z_0 равно 0; C – координаты точки, для которой производится расчёт, в виде комплексного числа; k – показатель степени, традиционно равен 2.

При вычислении последовательности для выбранной точки на плоскости (C) по этой формуле такая последовательность может иметь разное поведение. Она может быть ограниченной внутри какой-либо области (т. е. модули членов такой последовательности не превышают некоего предельного значения), а может стремиться в бесконечность, в зависимости от значения C . Точки, для которых эта последовательность ограничена, считаются принадлежащими множеству Мандельброта [14; 15].

Авторы статьи разработали программный продукт, предназначенный для визуализации множества Мандельброта Fractalomania применительно к обработке сложноструктурируемых изображений.

Материалы и методы

В разработанном программном продукте реализованы следующие функции редактирования фрактала:

- выбор одной из трёх вариаций множества Мандельброта;
- задание требуемой области координатной плоскости;
- задание требуемой «глубины прорисовки» множества;
- редактор палитры для раскрашивания заданной области множества;
- выбор из двух методов применения созданной палитры к выбранной области, а также диапазон значений, к которому палитра будет применяться;
- выбор одного из нескольких уровней оверсемплинга – методики улучшения качества итогового изображения;
- включение или выключение плавности перехода цвета («Плавный градиент»).

Другие реализованные функции программы:

- сохранение созданных фракталов в локальной БД (SQLite);
- экспорт полученных фракталов в файл формата PNG.

Для реализации данного программного продукта был выбран язык программирования Java как одна из технологий, предоставляющих удобные возможности для создания графического пользовательского интерфейса [16; 17], а также создания кросс-платформенных приложений. Для хранения данных используется локальная БД SQLite.

Также в программе использованы следующие библиотеки [18; 19; 20] и компоненты:

- JavaFX и ControlsFX v8.40.14 для создания интерфейса пользователя;
- SQLite JDBC 3.21.0 для работы с БД SQLite (<https://github.com/xerial/sqlite-jdbc>);
- imgscalr для масштабирования изображений (<https://github.com/rkalla/imgscalr>);
- Apache Commons IO 2.6 для работы с файловой системой (<http://commons.apache.org/proper/commons-io/>).

Для формирования фрактального изображения пользователю необходимо задать нужную область координатной плоскости, а также задать для этой области палитру цветов и способ её применения.

В главном окне программы Fractalomania, которое изображено на рисунке 1, пользователь имеет возможность создать один из трёх типов множества Мандельброта. Они отличаются показателем степени k в формуле $Z_{n+1} = Z_n^k + C$.

На рисунке 2 показано окно редактирования фрактала. Пользователь может выбрать нужную область на координатной плоскости с помощью мыши. Путём перетаскивания и с помощью колёсика мыши изменяется позиция на комплексной плоскости и масштаб, т. е. размер окрестности вокруг выбранной позиции. Задать цветовую палитру можно с помощью редактора палитры. Редактор палитры позволяет создавать, изменять и удалять цвета, а также менять их положение в палитре. Также необходимо задать параметры применения палитры и другие параметры фрактала.

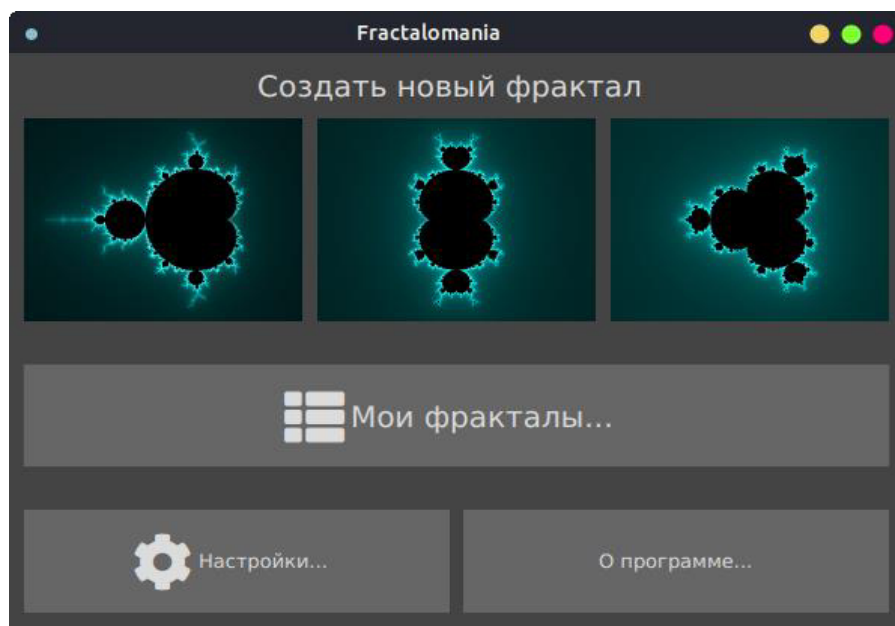


Рис. 1. Главное окно программы Fractalomania

Fig. 1. The main window of the Fractalomania program

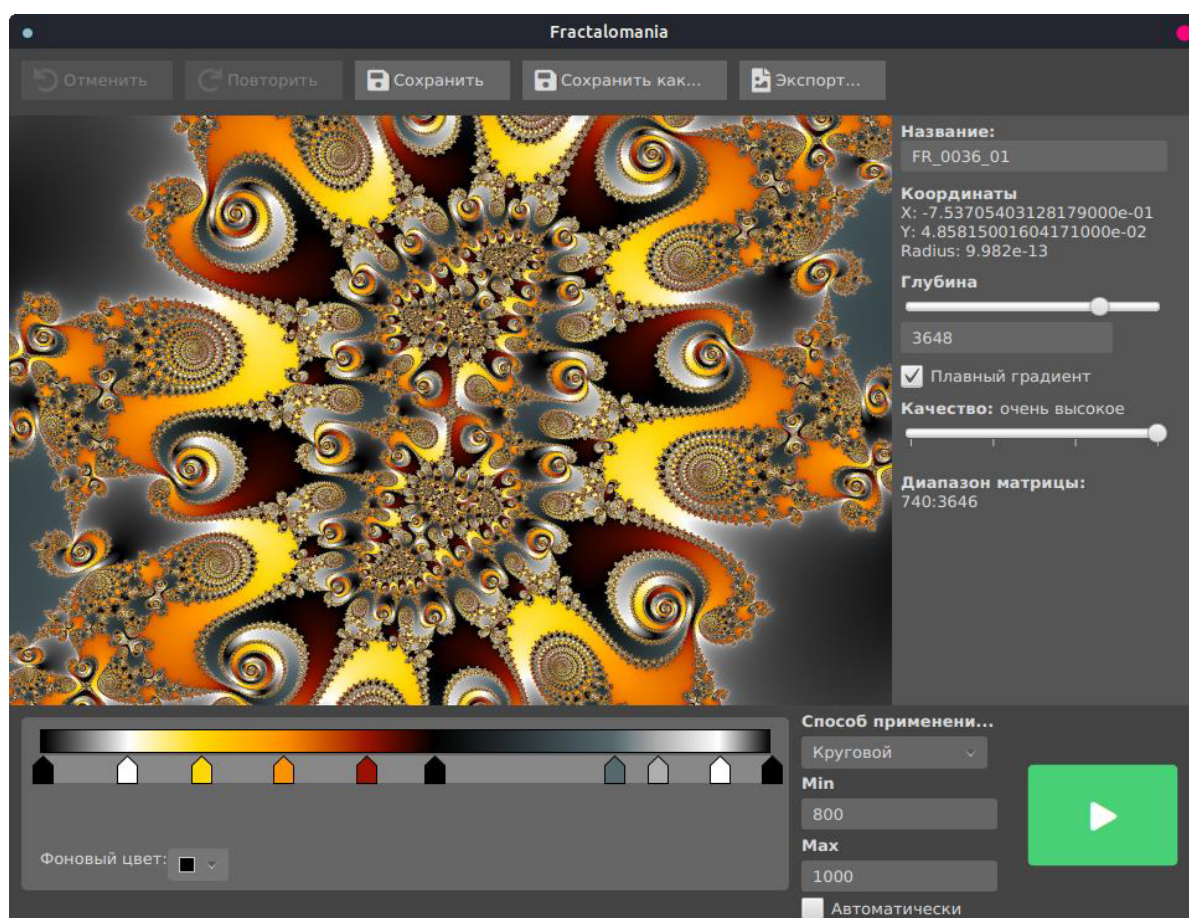


Рис. 2. Окно редактирования фрактала

Fig. 2. Fractal editing window

Принцип применения палитры будет описан далее в алгоритме генерации фрактального изображения.

Одним из важных параметров является глубина прорисовки фрактала (задаваемая параметром «Глубина» в интерфейсе пользователя). Величина этого параметра определяет максимальное

количество итераций рекурсивной формулы при генерации фрактала. Рассмотрим влияние этого параметра на полученное изображение (рис. 3).

Флажок «Плавный градиент» включает режим перехода цвета со ступенчатого на более плавный (рис. 4).

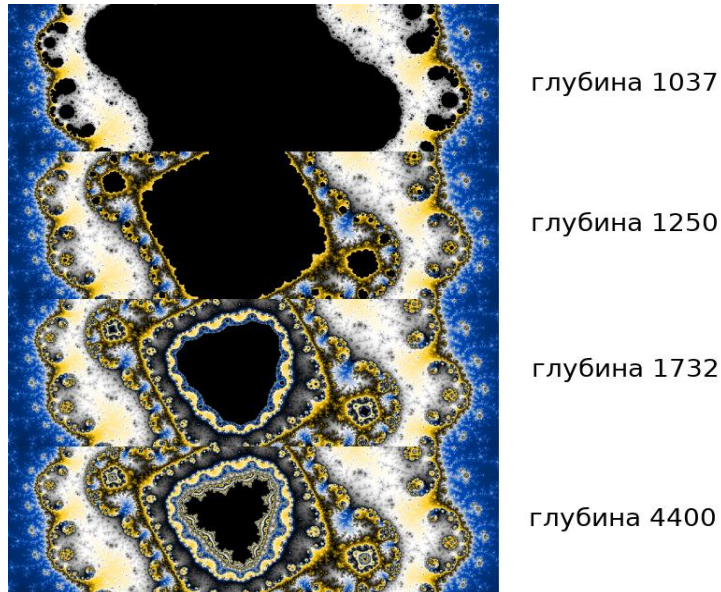


Рис. 3. Влияние значения глубины прорисовки на итоговое изображение

Fig. 3. The effect of the drawing depth value on the final image

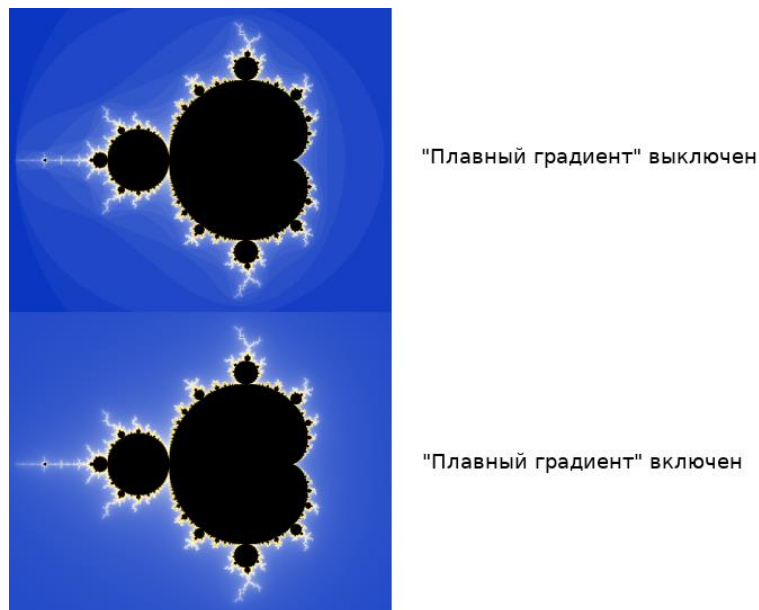


Рис. 4. Влияние режима «Плавный градиент» на итоговое изображение

Fig. 4. The effect of the "Smooth Gradient" mode on the final image

Таким образом, изменяя указанные параметры, можно добиться желаемого визуального эффекта.

Также имеется возможность сохранить фрактал в локальном хранилище с

указанием его названия. На рисунке 5 показано окно списка сохранённых фракталов.

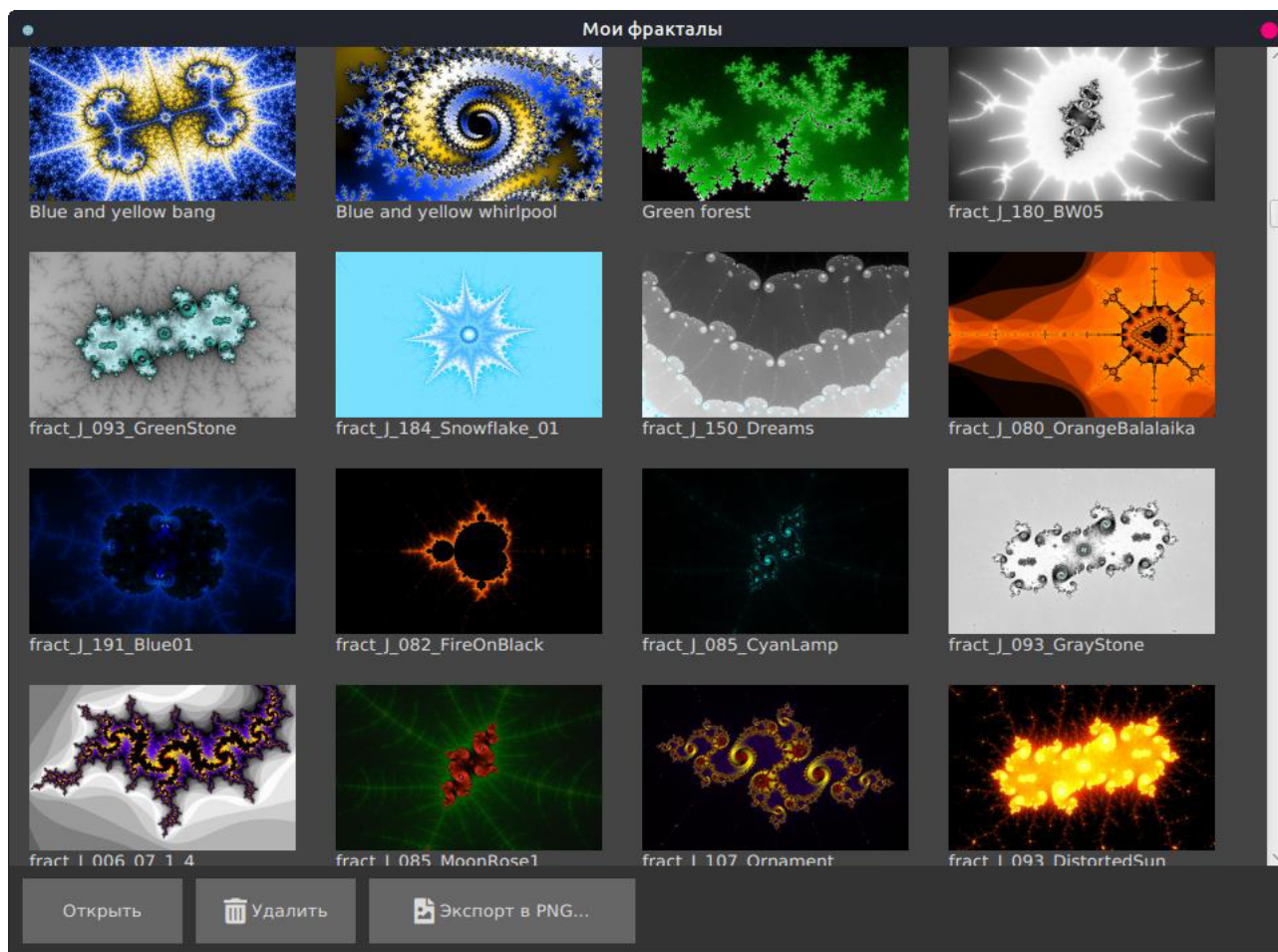


Рис. 5. Сохранённые фракталы

Fig. 5. Stored fractals

Сохранённые ранее фракталы можно открывать снова для редактирования, создавать на их основе новые, а также экспортировать в файлы формата PNG. Экспортировать изображение можно также из окна редактирования фрактала.

Алгоритм генерации фрактального изображения. Визуализация фрактала происходит в два этапа:

- *Первый этап* – расчёт матрицы. На этом этапе заданный участок координатной плоскости разбивается на одинаковые квадратные участки по количеству пикселей итогового изображения. Затем для каждого такого участка происходит расчёт целочисленного значения. Для расчёта выбирается точка, являющаяся центром квадрата. Для расчёта значения этой точки используется

уже упомянутая рекурсивная формула (1):

$$Z_{n+1} = Z_n^k + C,$$

где Z и C – комплексные числа, причём Z_0 равно 0; C – координаты точки, для которой производится расчёт, в виде комплексного числа; k равно 2, 3 или 4 (в зависимости от выбранного пользователем типа фрактала).

По этой формуле производится расчёт последовательности, который останавливается в одном из двух случаев:

1) модуль очередного рассчитанного значения $Z_n + 1$ больше некоторого предела (обычно равного 2). Тогда итоговым значением точки является номер итерации, на котором это произошло, т. е. n ;

2) достигнуто максимальное количество итераций (равное заданному пользователем параметру «Глубина»). Тогда считается, что данная точка лежит внутри множества Мандельброта, и у неё нет никакого значения.

Таким образом, после первого этапа мы имеем матрицу чисел, размер которой равен размеру итогового изображения в пикселях (т. е. имеем одно число на каждый пиксель изображения). Причём все числа находятся в диапазоне от 0 до значения параметра «Глубина», задаваемого в интерфейсе программы (рис. 6). Кроме элементов, соответствующих точкам, находящимся внутри множества, элементы матрицы, соответствующие точкам внутри множества, считаются пустыми.

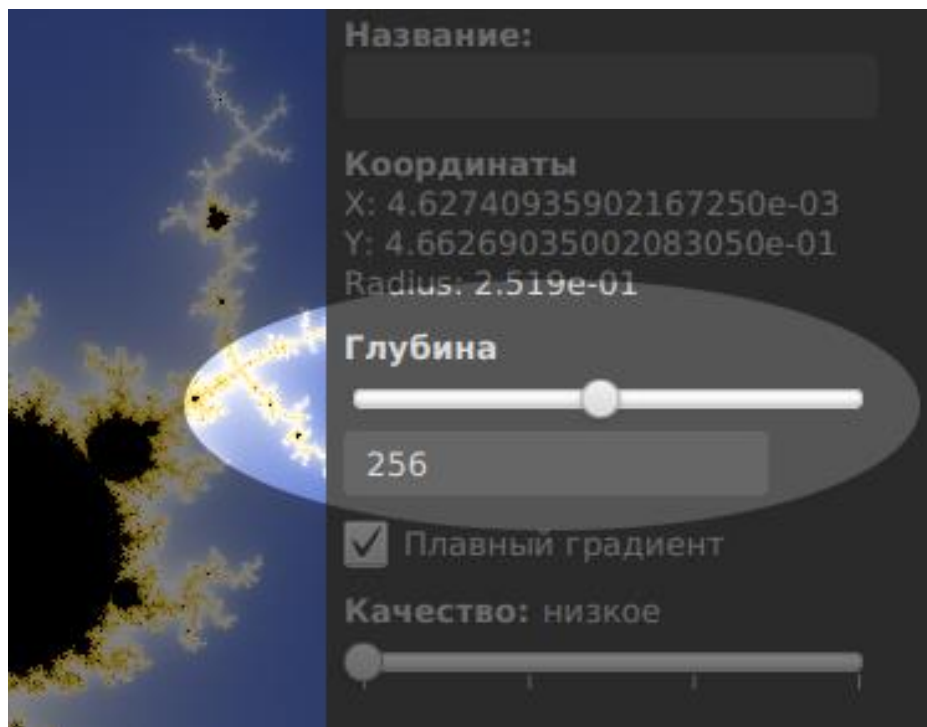


Рис. 6. Параметр «Глубина» задаёт максимальное количество итераций при расчёте матрицы

Fig. 6. The "Depth" parameter sets the maximum number of iterations when calculating the matrix

• *Второй этап* – раскрашивание. На этом этапе каждому целочисленному значению матрицы присваивается свой цвет из заданной палитры. Для присвоения конкретного значения цвета каждому значению из полученной на первом

этапе матрицы палитра как бы «накладывается» на диапазон чисел. Диапазон наложения палитры, который задаётся параметрами *min* и *max*, изображен ниже (рис. 7).

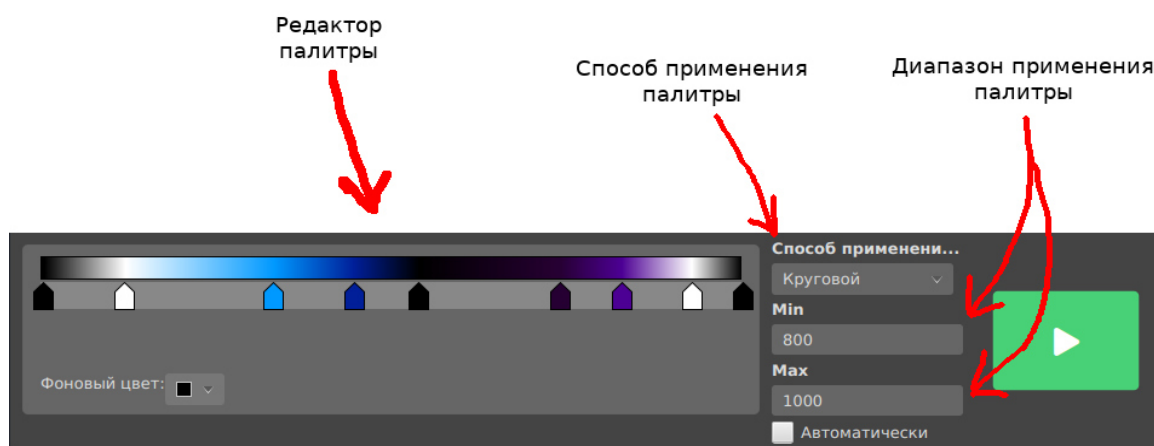


Рис. 7. Редактор палитры и задание параметров её применения

Fig. 7. Palette editor and setting parameters for its application

При этом начало палитры (левый край в редакторе палитры) соответствует минимальному значению диапазона, а конец палитры – максимальному

значению, как на рисунке 8. Промежуточные значения соответственно интерполируются.

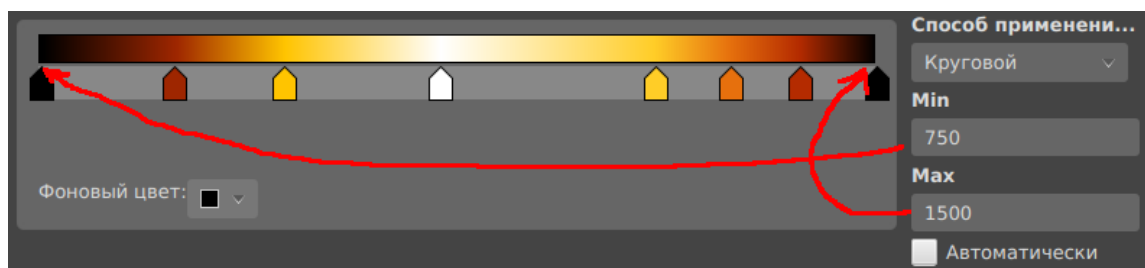


Рис. 8. Палитра накладывается на заданный диапазон значений

Fig. 8. The palette is superimposed on the specified range of values

При использовании данного метода наложения необходимо решить, как быть со значениями, выходящими за пределы диапазона наложения.

Для этого предусмотрено два способа применения палитры – простой и круговой. При простом способе для таких значений матрицы два крайних

цвета, соответствующие минимальному и максимальному значению диапазона, как бы продолжают в бесконечность влево и вправо от крайних значений. При круговом способе палитра просто бесконечно повторяется влево и вправо. Схема представлена ниже (рис. 9).

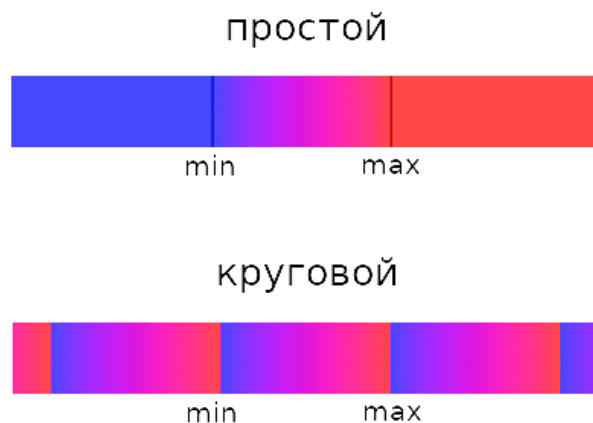


Рис. 9. Два способа применения палитры

Fig. 9. Two ways to apply the palette

Точки, принадлежащие множеству Мандельброта (т. е. те, для которых значения в матрице пустые), окрашиваются в фоновый цвет (задаваемый также в редакторе палитры).

Результаты и их обсуждение

При генерации фрактального изображения описанным выше способом итоговое изображение будет иметь заметные дефекты, выглядящие как

«шум», особенно заметные на более сложных участках фрактала. Пример представлен ниже (рис. 10).

Это связано с тем, что фрактал имеет сложную структуру, при расчёте значения элемента матрицы алгоритм может попасть в точку, значение которой (а соответственно и цвет) будет заметно отличаться от среднего значения окружающих точек.

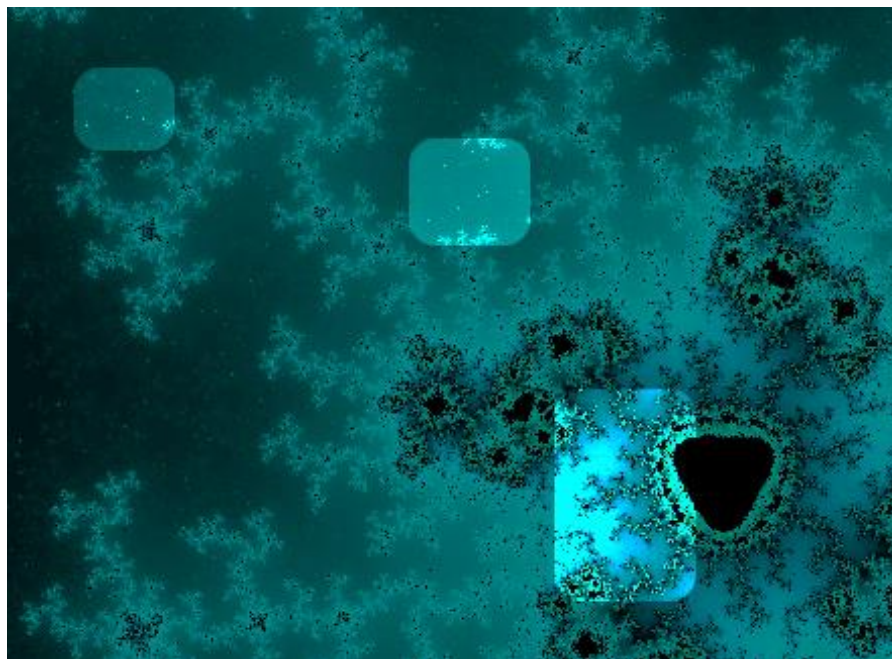


Рис. 10. Пример шумов на итоговом изображении

Fig. 10. Example of noise in the final image

Так происходит из-за того, что точки комплексной плоскости, лежащие близко к границе множества Мандельброта (но не входящие в него), являются точками бифуркации, т. е. при сколь-

угодно малом смещении характер их поведения может резко меняться. Схематично принцип появления шумов представлен ниже (рис. 11).



Рис. 11. Принцип появления шумов на сгенерированном изображении

Fig. 11. The principle of the appearance of noise in the generated image

На рисунке 11 направление взгляда находится в комплексной плоскости XY , а рассчитанные значения точек отложены на оси Z , т. е. мы видим как бы «разрез» поверхности, представляющей множество значений.

Для уменьшения данного эффекта применяется метод оверсемплинга, заключающийся в выборе меньшего интервала в кратное число раз между рассчитываемыми точками и дальнейшим сжатием размера изображения. В этом случае происходит генерация изображения с разрешением, кратно превышающим требуемое. На этом изображении также будут шумы. Затем следует

сжатие полученного изображения до требуемого разрешения. В итоге после уменьшения разрешения изображения цвета соседних точек усредняются, и изображение получается более ровным (рис. 12).

Уровень оверсемплинга задаётся параметром «Качество» в окне редактирования фрактала, элемент интерфейса представлен ниже (рис. 13), где «низкое» качество соответствует отсутствию оверсемплинга, «очень высокое» – множителю 4 (т. е. генерируется промежуточное изображение с разрешением в 4 раза большим требуемого, а затем происходит сжатие до требуемого размера).



Рис. 12. Влияние оверсемплинга на итоговое изображение

Fig. 12. The effect of oversampling on the final image

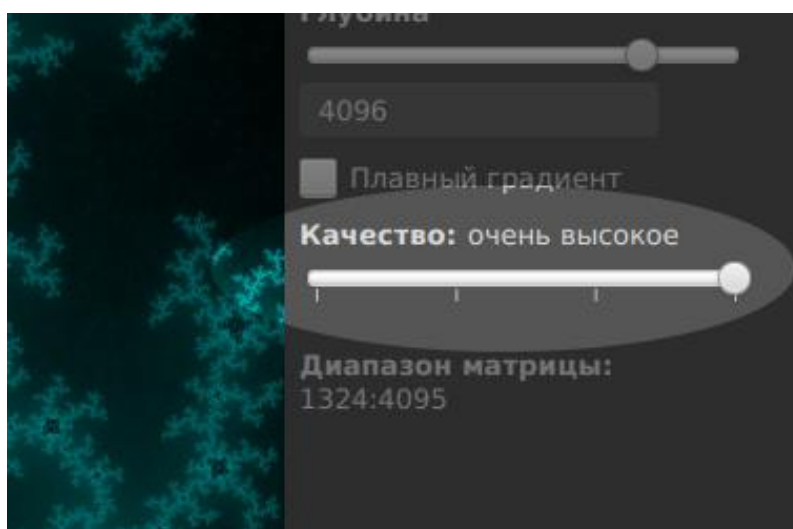


Рис. 13. Выбор качества итогового изображения

Fig. 13. Selecting the quality of the final image

Уровень оверсемплинга значительно влияет на скорость генерации фрактала, так как размер рассчитываемой матрицы увеличивается пропорционально квадрату множителя.

Выводы

Рассмотренная в данной статье программа Fractalomania реализует базовые функции визуализации множества

Мандельброта, основанные на самоподобии и масштабируемости фракталов, что позволяет экспортировать полученные фрактальные изображения для моделирования процессов в виртуальной дополненной реальности.

В перспективе разработанный программный продукт может быть доработан и оснащен новым функционалом, способным предоставить пользователю

возможность создавать объекты, имеющие фрактальную форму. Такая доработка позволит использовать программу для создания моделей, предназначенных

для анализа больших данных, а также для применить для обработки сложно-структурируемых медицинских изображений.

Список литературы

1. Новейшие методы обработки изображений / А. А. Потапов [и др.]; под ред. А. А. Потапова. М.: Физматлит, 2008. 496 с.
2. Фисенко В. Т., Фисенко Т. Ю. Исследование методов распознавания образов для систем компьютерного зрения роботов // Известия вузов. Приборостроение. 2013. Т. 56, № 5. С. 63–70.
3. Русанова И. А. Возможности фрактальной обработки изображений и сигналов в медицине. М.: LAP Lambert Academic Publishing, 2014. 140 с.
4. Бак П. Как работает природа. Теория самоорганизованной критичности. М.: URSS: Либроком, 2022. 288 с.
5. Деменок С. Л. Фрактал. Между мифом и ремеслом. М.: Ринвол, Академия исследования культуры, 2011. 296 с.
6. Методы и алгоритмы контурного анализа для задач классификации сложноструктурируемых изображений / М. В. Дюдин, А. Д. Поваляев, Е. С. Подвальный, Р. А. Томакова // Вестник Воронежского государственного технического университета. 2014. Т. 10, № 3-1. С. 54–59.
7. Метод классификации рентгенограмм на основе использования глобальной информации об их структуре / Р. А. Томакова, М. В. Томаков, И. В. Дураков, В. В. Жилин // Биомедицинская радиоэлектроника. 2016. № 9. С. 45–51.
8. Томакова Р. А., Дюдин М. В., Томаков М. В. Нейросетевые модели принятия решений для диагностики заболеваний легких на основе флюорограмм грудной клетки // Биомедицинская радиоэлектроника. 2014. № 9. С. 12–15.
9. Брежнева А. Н., Томакова Р. А., Филист С. А. Спектральный анализ сегментов изображения глазного дна для количественной оценки сосудистой патологии // Биомедицинская радиоэлектроника. 2009. № 6. С. 15–18.
10. Мандельброт Б. Б. Фракталы и хаос. Множество Мандельброта и другие чудеса. М.: Регулярная и хаотическая динамика, 2009. 392 с.
11. Деменок С. Л. Просто фрактал. М.: Страта, 2018. 308 с.
12. Секованов В. С. Элементы теории фрактальных множеств. М.: Едиториал УРСС, 2018. 248 с.
13. Гуц А. К. Комплексный анализ и кибернетика. М.: ЛКИ, 2010. 144 с.
14. Мандельброт Б. Б. Фрактальная геометрия природы. М.: Институт компьютерных исследований, 2010. 676 с.

15. Божокин С. В., Паршин Д. А. Фракталы и мультифракталы. Ижевск: Регулярная и хаотическая динамика, 2001. 128 с.
16. Прохоренок Н. А. Основы Java. СПб.: БХВ-Петербург, 2017. 704 с.
17. Прохоренок Н. А. JavaFX. СПб.: БХВ-Петербург, 2020. 768 с.
18. Васильев А. Н. Java. Объектно-ориентированное программирование. СПб.: Питер, 2012. 398 с.
19. Роберт Лафоре. Структуры данных и алгоритмы в Java. СПб.: Питер, 2016. 704 с.
20. Нимейер П., Леук Д. Программирование на Java. М.: Эксмо, 2014. 1216 с.

References

1. Potapov A. A., eds. Novejshie metody obrabotki izobrazhenij [The latest methods of image processing]; ed. by A. A. Potapov. Moscow, Fizmatlit, 2008. 496 p.
2. Fisenko V. T., Fisenko T. Y. Issledovanie metodov raspoznavaniya obrazov dlya sistem komp'yuternogo zreniya robotov [Investigation of pattern recognition methods for robot computer vision systems]. *Ivestiya vuzov. Priborostroenie = Ivestia of Universities. Instrumentation*, 2013, vol. 56, no. 5, pp. 63–70.
3. Rusanova I. A. Vozmozhnosti fraktal'noj obrabotki izobrazhenij i signalov v medicine [Possibilities of fractal processing of images and signals in medicine]. Moscow, LAP Lambert Academic Publishing, 2014. 140 p.
4. Bak P. Kak rabotaet priroda. Teoriya samoorganizovannoj kritichnosti [How nature works. Theory of self-organized criticality]. Moscow, URSS, Librocom Publ., 2022. 288 p.
5. Demenok S. L. Fraktal. Mezhdru mifom i remeslom [Fractal. Between myth and craft]. Moscow, Rinvol, Academy of Culture Research Publ., 2011. 296 p.
6. Dyudin M. V., Povalyaev A. D., Podvalny E. S., Tomakova R. A. Metody i algoritmy konturnogo analiza dlya zadach klassifikacii slozhnostrukturiruemih izobrazhenij [Methods and algorithms of contour analysis for classification problems of complex structured images]. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta = Bulletin of the Voronezh State Technical University*, 2014, vol. 10, no. 3-1, pp. 54–59.
7. Tomakova R. A., Tomakov M. V., Durakov I. V., Zhilin V. V. Metod klassifikacii rentgenogramm na osnove ispol'zovaniya global'noj informacii ob ih strukture [Method of classification of radiographs based on the use of global information about their structure]. *Biomedicinskaya radioelektronika = Biomedical Radioelectronics*, 2016, no. 9, pp. 45–51.
8. Tomakova R. A. Dyudin M. V., Tomakov M. V. Nejrosetevye modeli prinyatiya reshenij dlya diagnostiki zabolevanij legkih na osnove flyuorogramm grudnoj kletki [Neural network models of decision-making for the diagnosis of lung diseases based on chest fluorograms]. *Biomedicinskaya radioelektronika = Biomedical Radioelectronics*, 2014, no. 9, pp. 12–15.

9. Brezhneva A. N., Tomakova R. A., Filist S. A. Spektral'nyj analiz segmentov izobrazheniya glaznogo dna dlya kolichestvennoj ocenki sosudistoj patologii [Spectral analysis of fundus image segments for quantitative assessment of vascular pathology]. *Biomedicinskaya radioelektronika = Biomedical Radioelectronics*, 2009, no. 6, pp. 15–18.
10. Mandelbrot B. B. Fraktaly i haos. Mnozhestvo Mandel'brot'a i drugie chudes'a [Fractals and chaos. The Mandelbrot set and other miracles]. Moscow, Regular and chaotic dynamics Publ., 2009. 392 p.
11. Demenok S. L. Prosto fraktal [Simply fractal]. Moscow, Strata Publ., 2018. 308 p.
12. Sekovanov V. S. Elementy teorii fraktal'nyh mnozhestv [Elements of the theory of fractal sets]. Moscow, Unitorial URSS Publ., 2018. 248 p.
13. Guts A. K. Kompleksnyj analiz i kibernetika [Complex analysis and cybernetics]. Moscow, LKI Publ., 2010. 144 p.
14. Mandelbrot B. B. Fraktal'naya geometriya prirody [Fractal geometry of nature]. Moscow, Institute of Computer Research Publ., 2010. 676 p.
15. Bozhokin S. V., Parshin D. A. Fraktaly i mul'tifraktaly [Fractals and multifractals]. Izhevsk, Regular and chaotic dynamics Publ., 2001. 128 p.
16. Prokhorenok N. A. Osnovy Java [Fundamentals of Java]. St. Petersburg, BHV-Petersburg Publ., 2017. 704 p.
17. Prokhorenok N. A. [JavaFX]. St. Petersburg, BHV-Petersburg Publ., 2020. 768 p.
18. Vasiliev A. N. Ob"ektno-orientirovannoe programmirovaniye [Java. Object-oriented programming]. St. Petersburg, Peter Publ., 2012. 398 p.
19. Laforet R. Struktury dannyh i algoritmy v Java [Data structures and algorithms in Java]. St. Petersburg, Peter Publ., 2016. 704 p.
20. Niemeyer P., Leuk D. Programmirovaniye na Java [Programming in Java]. Moscow, Eksmo Publ., 2014. 1216 p.

Информация об авторах / Information about the Authors

Томакова Римма Александровна, доктор технических наук, профессор, профессор кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: rtomakova@mail.ru, Researcher ID: O-6164-2015, ORCID: 0000-0003-152-4714

Rimma A. Tomakova, Dr. of Sci. (Engineering), Professor, Professor of the Department of Software Engineering, Southwest State University, Kursk, Russian Federation, e-mail: rtomakova@mail.ru, Researcher ID: O-6164-2015, ORCID: 0000-0003-152-4714

Ахмадуллин Ильдар Музгирович, магистрант кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: ildar.230@yandex.ru

Ildar M. Akhmadullin, Undergraduate of the Department of Software Engineering, Southwest State University, Kursk, Russian Federation, e-mail: ildar.230@yandex.ru

Нефедов Никита Геннадьевич, магистрант кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: niknefed@gmail.com, ORCID: 0009-0004-2667-2662

Nikita G. Nefedov, Undergraduate of the Department of Software Engineering, Southwest State University, Kursk, Russian Federation, e-mail: niknefed@gmail.com, ORCID: 0009-0004-2667-2662

Пузырев Евгений Игоревич, магистрант кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: puzyryov2000@mail.ru, ORCID: 0009-0008-5967-8744

Evgeny I. Puzyrev, Undergraduate of the Department of Software Engineering, Southwest State University, Kursk, Russian Federation, e-mail: puzyryov2000@mail.ru, ORCID: 0009-0008-5967-8744

Малышев Антон Александрович, студент кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: alta76@yandex.ru, ORCID: 0009-0001-4838-6710

Anton A. Malyshev, Student of the Department of Software Engineering, Southwest State University, Kursk, Russian Federation, e-mail: alta76@yandex.ru, ORCID: 0009-0001-4838-6710