

## АЛГОРИТМ СВЕРТКИ 2D ОБЪЕКТОВ В ИНВАРИАНТНЫЙ И ВЕРОЯТНОСТНЫЙ «ГЕНЕТИЧЕСКИЙ КОД»

Р. Ф. Хаматнуров<sup>1,a</sup>, Т. В. Гавриленко<sup>1,2,b</sup>

<sup>1</sup> Сургутский филиал федерального государственного автономного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Национального исследовательского центра «Курчатовский институт», г. Сургут, Российская Федерация

<sup>2</sup> Сургутский государственный университет, г. Сургут, Российская Федерация

<sup>a</sup> ORCID: <https://orcid.org/0009-0004-4239-8720>, ✉ [ruslan.khamatnuroff@yandex.ru](mailto:ruslan.khamatnuroff@yandex.ru)

<sup>b</sup> ORCID: <https://orcid.org/0000-0002-3243-2751>, [taras.gavrilenko@gmail.com](mailto:taras.gavrilenko@gmail.com)

*Аннотация:* в статье представлен авторский алгоритм свертки двумерных объектов (изображений) в компактный, линейный и вероятностный «генетический код», основанный на биологических механизмах хранения и передачи наследственной информации. Представленный подход имитирует процессы клеточного деления, дифференциации и мутаций, рассматривая пиксель двумерного изображения как условную клетку, а его цвет — как тип клетки. Алгоритм реализует вероятностный обход изображения с использованием случайного выбора направления, что обеспечивает инвариантность к порядку обработки и моделирует стохастичность биологических систем. Каждый элемент кодируется в виде правила, содержащего идентификатор цвета, ссылки на соседние элементы и специальные маркеры для обозначения тупиковых состояний. Представлен метод оптимизации за счет отсечения нулевых значений, перехода к относительным ссылкам и фиксации длины строк для эффективного доступа. В алгоритм введены два типа мутаций: параметрические (изменение цвета) и привнесенные (нарушение топологии), позволяющие моделировать эволюцию структуры. Результатом работы алгоритма является текстовый файл, содержащий правила деления, палитру цветов и метаинформацию, достаточные для точного воспроизведения исходного объекта. Эксперименты на изображениях различной сложности подтвердили корректность свертки и возможность моделирования мутаций. Разработанный метод открывает перспективы для применения в биоинформатике, робототехнике и адаптивных системах, включая создание самоорганизующихся и самовоспроизводящихся структур.

*Ключевые слова:* алгоритм свертки, изображения, генетический код, мутации.

*Благодарности:* работа выполнена в рамках государственного задания НИЦ «Курчатовский институт» — НИИСИ по теме № FNEF-2024-0001 «Создание и реализация доверенных систем искусственного интеллекта, основанных на новых математических и алгоритмических методах, моделях быстрых вычислений, реализуемых на отечественных вычислительных системах» (1023032100070-3-1.2.1).

*Для цитирования:* Хаматнуров Р. Ф., Гавриленко Т. В. Алгоритм свертки 2D объектов в инвариантный и вероятностный «генетический код». *Успехи кибернетики*. 2025;6(3):112–122.

*Поступила в редакцию:* 03.08.2025.

*В окончательном варианте:* 09.09.2025.

## CONVOLUTION OF 2D OBJECTS INTO AN INVARIANT AND PROBABILISTIC “GENETIC CODE”

R. F. Khamatnurov<sup>1,a</sup>, T. V. Gavrilenko<sup>1,2,b</sup>

<sup>1</sup> Surgut Branch of Scientific Research Institute for System Analysis of the National Research Centre “Kurchatov Institute”, Surgut, Russian Federation

<sup>2</sup> Surgut State University, Surgut, Russian Federation

<sup>a</sup> ORCID: <https://orcid.org/0009-0004-4239-8720>, ✉ [ruslan.khamatnuroff@yandex.ru](mailto:ruslan.khamatnuroff@yandex.ru)

<sup>b</sup> ORCID: <https://orcid.org/0000-0002-3243-2751>, [taras.gavrilenko@gmail.com](mailto:taras.gavrilenko@gmail.com)

*Abstract:* we developed an algorithm for convolving two-dimensional images into a compact, linear, and probabilistic “genetic code” inspired by biological mechanisms of storing and transmitting hereditary information. Our approach simulated processes of cell division, differentiation, and mutation by treating each pixel of a two-dimensional image as a conditional cell and its color as a cell type. The algorithm implemented a probabilistic image traversal with random direction selection, which ensured invariance to processing order and reproduced the stochasticity of biological systems. Each element was encoded as a rule containing a color identifier, links to neighboring elements, and special markers indicating dead ends.

We proposed an optimization method that removed zero values, switched to relative links, and fixed string length to improve access efficiency. We also introduced two types of mutations: parametric (color changes) and structural (topology violations), which allowed us to model structural evolution. The algorithm produced a text file containing division rules, a color palette, and meta-information sufficient to reconstruct the original image. Experiments with images of varying complexity confirmed the correctness of the convolution and demonstrated the possibility of modeling mutations. This method creates prospects for applications in bioinformatics, robotics, and adaptive systems, including the development of self-organizing and self-reproducing structures.

*Keywords:* convolution algorithm, images, genetic code, mutations.

*Acknowledgements:* this study is a part of the FNEF-2024-0001 government order contracted to the Scientific Research Institute for System Analysis of the National Research Centre “Kurchatov Institute”, project No. 1023032100070-3-1.2.1 Development and Implementation of Trusted Artificial Intelligence Systems Based on new Mathematical Methods and Algorithms, Fast Computing Models for Domestic Computing Systems.

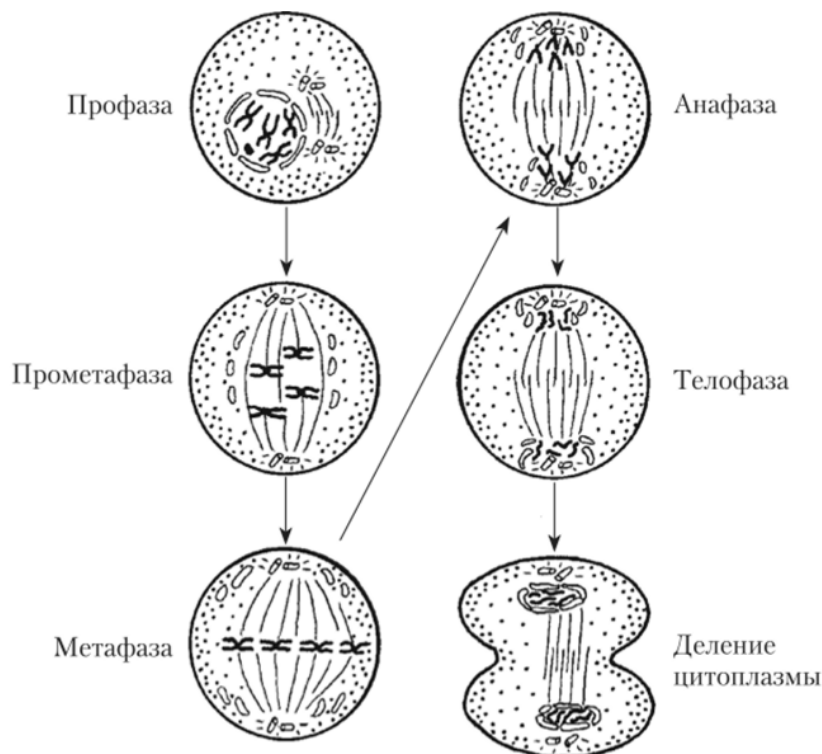
*Cite this article:* Khamaturov R. F., Gavrilenko T. V. Convolution of 2D Objects into an Invariant and Probabilistic “Genetic Code”. *Russian Journal of Cybernetics*. 2025;6(3):112–122.

*Original article submitted:* 03.08.2025.

*Revision submitted:* 09.09.2025.

Современные исследования в области вычислительных систем все чаще опираются на междисциплинарные подходы, сочетающие идеи биологии, кибернетики и информатики. Биологические системы обладают уникальными механизмами хранения, передачи и воспроизведения информации, которые обеспечивают формирование сложных топологических структур и их эволюцию. Одним из наиболее показательных примеров является молекула дезоксирибонуклеиновой кислоты (ДНК), содержащая наследственный код организма. Благодаря процессу репликации и деления клеток информация передается с высокой точностью, при этом допускаются незначительные отклонения, что обуславливает появление мутаций и биологическое разнообразие.

Особый интерес вызывает механизм, обеспечивающий из ограниченного набора элементарных правил в живых организмах воспроизведение сложных форм и функций. ДНК представляет собой систему инструкций, задающих правила построения организма. Клетка хранит полный набор этих данных и передает их при делении, что обеспечивает точное воспроизведение топологии (рис. 1).



**Рис. 1.** Деление клетки [1]

В процессе деления клетки генетическая информация полностью копируется и передается дочерним клеткам, что обеспечивает развитие организма и передачу наследственных признаков [2]. Одной из ключевых особенностей является воспроизведение заданной топологии, по мере формирования которой организм приобретает функциональные характеристики. Структура распределения клеток определяет взаимосвязи между элементами, обеспечивая форму, координацию и возможности организма. В качестве примера можно рассмотреть развитие насекомых. Так, оса формируется из одной клетки путем последовательных делений, что приводит к появлению целостного организма с характерными морфологическими и функциональными признаками, включая способность к полету и ориентации в пространстве, которую она приобретает сразу после окончания формирования конечной топологии. Стадии развития осы представлены на рис. 2.



**Рис. 2.** Стадии формирования тела осы [3]

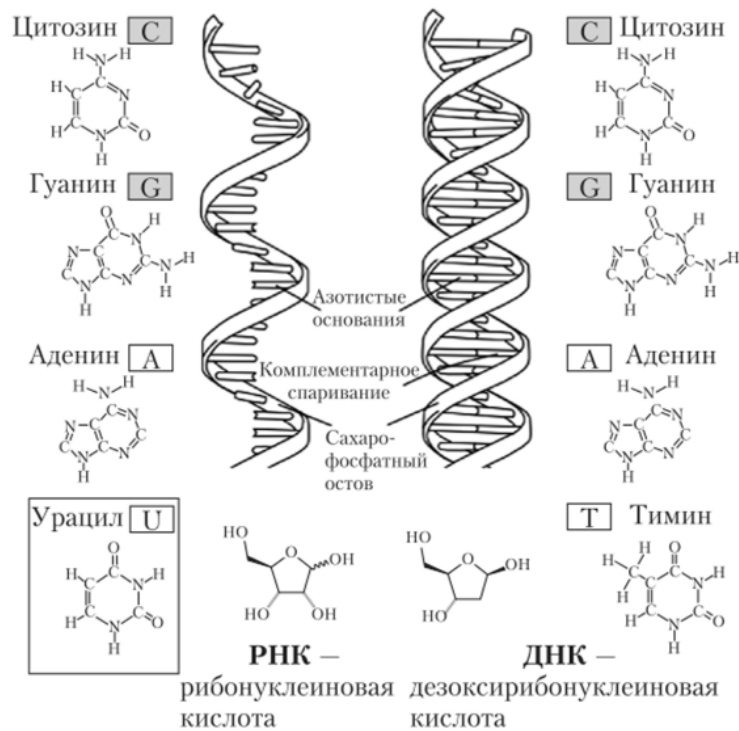
Подобный подход может быть использован в моделировании данных, где возникает необходимость эффективного кодирования и восстановления сложных структур.

В исследовании рассматривалась возможность построения универсальных алгоритмов кодирования информации, позволяющих кодировать исходные 2D и 3D объекты в компактный, линейный набор правил, аналогичных «генетическому коду». Такой подход открывает возможности для исследования следующих процессов:

- имитация биологических процессов (деление клеток, наследование, мутации);
- создание адаптивных самоорганизующихся систем;
- разработка методов эффективного кодирования и передачи информации;
- моделирование робототехнических комплексов, способных воспроизводить и при необходимости изменять структуру и топологию в зависимости от задачи.

Биологические системы обладают уникальной способностью к воспроизведению и саморегуляции, одним из ключевых процессов является деление клетки, в ходе которого наследственная информация копируется и передается дочерним клеткам. Этот процесс обеспечивает сохранение общей структуры и топологии организма и одновременно допускает возможность появления незначительных изменений.

В основе наследования лежит молекула ДНК (дезоксирибонуклеиновой кислоты), представляющая собой двухцепочечную спираль, образованную нуклеотидами. Каждая цепь формируется последовательностью азотистых оснований: аденина (А), тимина (Т), гуанина (G) и цитозина (С) (рис. 3). Благодаря принципу комплементарности (А–Т и G–С) обеспечивается точное копирование информации при репликации. Последовательность нуклеотидов задает структуру белков, которые, в свою очередь, определяют структуру, топологию и функции каждой отдельной клетки.



**Рис. 3.** Структура ДНК [4]

Мутации играют особую роль в формировании разнообразия живых систем. Их можно разделить на:

- параметрические — небольшие изменения, не нарушающие целостность структуры (например, вариации в окраске, форме или размере);
- привнесенные — затрагивающие топологию организма (например, перестройки хромосом или аномалии развития органов).

С точки зрения моделирования данные биологические механизмы можно интерпретировать как алгоритмы свертки и развертки информации. Каждая клетка хранит полный набор правил, необходимых для воспроизведения организма. Деление клетки соответствует процессу копирования правил и передачи их новым элементам системы. Мутации вносят вероятностные изменения, обеспечивая вариативность и возможность адаптации [5].

В контексте вычислительных систем в качестве аналога клетки будем рассматривать агента. Совокупность агентов образует мультиагентную систему, где каждый элемент выполняет локальные действия, но в совокупности они формируют сложную структуру. Таким образом, в разработанной модели:

- агент — условная клетка;
- правило — условный ген;
- обход объекта — процесс деления и дифференцирования клеток;
- мутация — вариация в наборе правил.

Предложенный алгоритм основан на имитации биологических принципов наследования информации и построения топологии организма. В качестве модели организма и исходных данных используется двумерное изображение в формате BMP. При этом каждый элемент изображения рассматривается как условная клетка, а цвет элемента — как тип клетки (по аналогии с дифференциацией). Первоначально был разработан алгоритм, позволяющий преобразовать исходный объект в набор правил, аналогичных «генетическому коду», обеспечивающий возможность в дальнейшем воспроизведения исходной формы и внесения мутаций.

#### Алгоритм обхода изображений

Обход изображения реализован по вероятностному принципу. На каждом шаге выбирается случайное направление движения (вверх, вниз, влево, вправо). Такой подход обеспечивает вариативность

структуры выходного кода и моделирует стохастический характер наследования в биологических системах.

В процессе обхода выполняются следующие шаги:

1. Определение текущего элемента (пикселя).
2. Считывание его характеристик (цвет, положение, соседи).
3. Формирование правила и его запись в промежуточную структуру.
4. Перемешивание массива направлений и переход к следующему элементу.
5. Реализация мутаций.

В самом начале имеется массив с возможными ходами: вверх, вниз, влево и вправо. За один проход цикла может обработаться до 4 пикселей. Перед каждой итерацией цикла данный массив перемешивается, и таким образом порядок обхода меняется. Более подробно алгоритм представлен на блок-схеме ниже (рис. 4). На вход методу подается матрица, заполненная нулями, размером, равным размеру входного изображения. Из стартовой точки начинается формирование пути. На выходе мы получим массив, который будет содержать сформированный путь. В самом методе присутствует массив, содержащий направления обхода, и перед каждой итерацией он перемешивается, чтобы обходить изображения каждый раз по-разному.

После сформированного пути начинается обход для считывания цвета пикселя и формирования правила. На данном этапе проявляются мутации.

На вход методу формирования правил подается пустая матрица размером, равным размерам изображения, точка, с которой начинается обход изображения, путь до обрабатываемого изображения, матрица, содержащая обход изображения, путь до выгрузки набора правил, переменная, которая участвует в параметрических мутациях, и переменная, участвующая в привнесенных мутациях. По умолчанию последние две переменные отключены. В этом методе сначала формируется дерево обхода, далее по этому дереву идет считывание цвета пикселя и запись его в массив цветов. Если переменная, которая отвечает за параметрическую мутацию, не равна 0, то идет поиск наиболее близкого по значению цвета; если он находится, то найденный только что цвет будет заменен на существующий.

### Формат формируемых правил

Каждый элемент изображения (например, пиксель) рассматривается как элементарная клетка, для которой формируется правило. Правило включает:

- идентификатор цвета (код, полученный из значений RGB или их преобразованной формы);
- ссылки на соседние элементы (сверху, снизу, слева, справа);
- указатели на тупиковые индексы, обозначающие отсутствие потомков;
- возможность внесения мутаций (изменение цвета или нарушение топологии).

Формат правил: `color_id up_id right_id down_id left_id`.

Первоначальная версия хранила цвета в формате ARGB — alpha, red, green and blue. Т. е. красный цвет обозначался как «255 255 0 0». Но далее было принято решение данные значения переводить с помощью математических функций в кодировку, состоящую из 4 символов. Теперь красный цвет обозначается как «7Rwk».

Сам формат правил также претерпел изменения, и финальный вариант формируется так:

1. Отсечение неиспользуемых нулей:

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 3 & \rightarrow 1 & 0 & 2 & 0 & 3 \\ 1 & 0 & 10 & 0 & 0 & 1 & 0 & 10 \end{array}$$

2. Замена абсолютных ссылок на относительные:

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 3 & \rightarrow 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 0 & 1 \end{array}$$

3. Тупиковые индексы. Теперь, если правило ссылается на другое правило, которое содержит лишь информацию о цвете пикселя, то эта ссылка указывается после специального символа «#». Например, «1 #3 #2 #4 1»: число «3» после знака «#» говорит о том, что пиксель над этим правилом будет содержать информацию лишь о своем цвете, и не будет передавать информацию соседним пикселям.

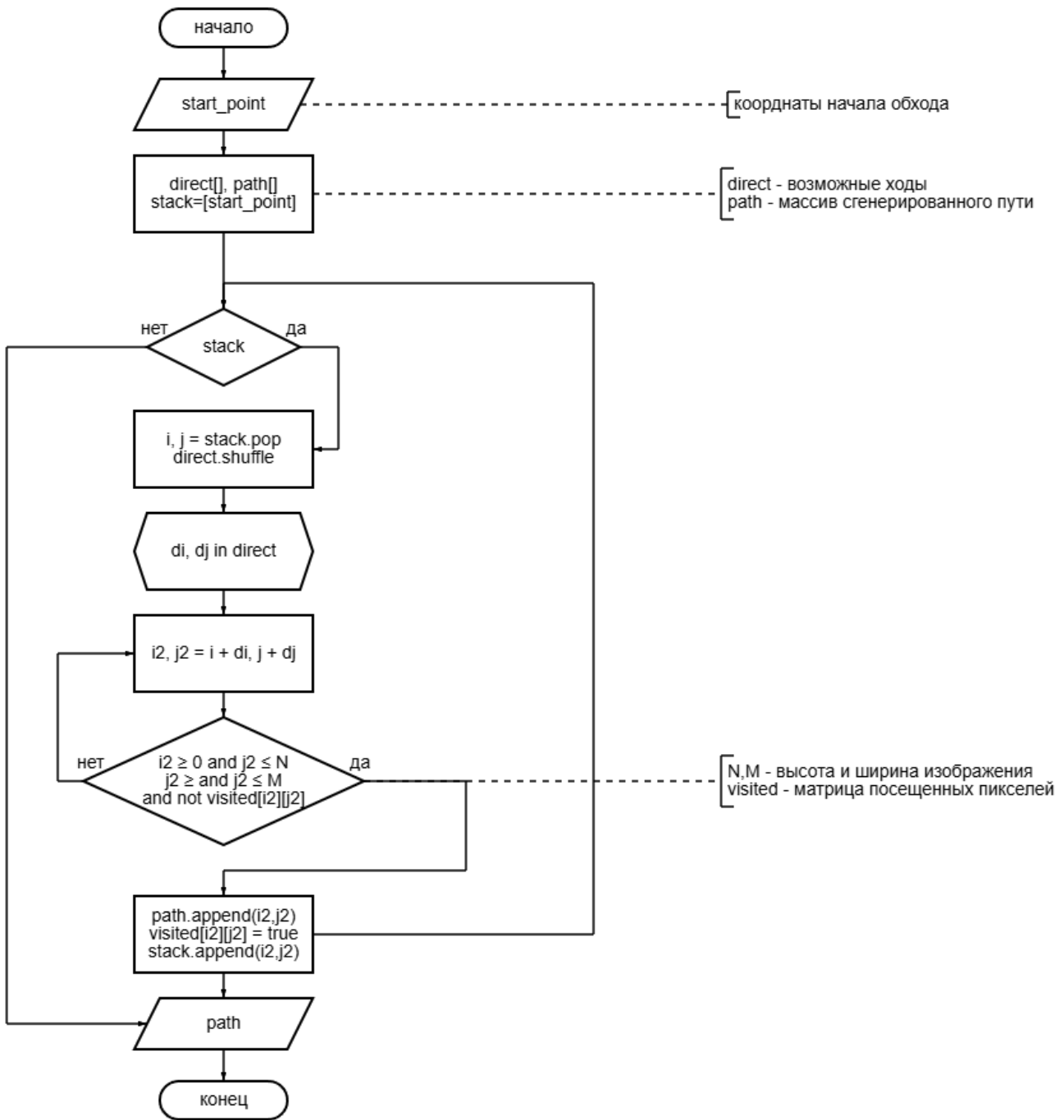


Рис. 4. Блок-схема алгоритма обхода изображения

4. Для ускорения обработки правила фиксируются в строках одинаковой длины, что позволяет обращаться к элементам по смещению, подобно работе с массивами в языках программирования.

В алгоритме предусмотрены два типа мутаций:

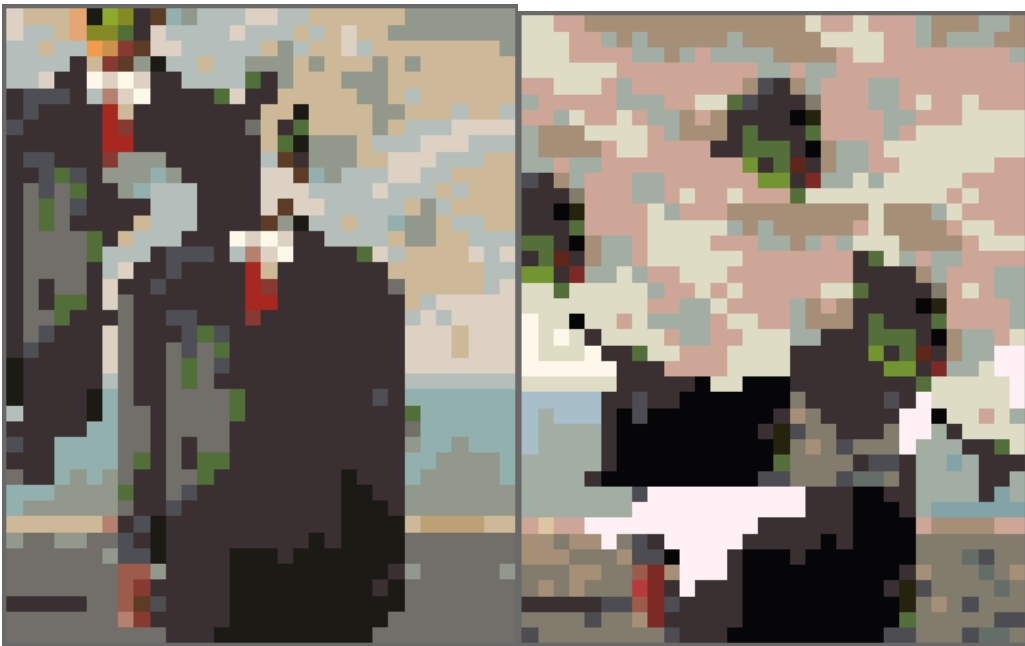
- параметрические мутации — при совпадении или близости цветовых значений используется округление к уже существующему цвету, что имитирует незначительные изменения (например, оттенок кожи или глаз) (рис. 5);
- привнесенные мутации — часть информации о соседях может быть намеренно потеряна, что приводит к деформации топологии объекта (например, отсутствие фрагмента изображения или удвоение элемента) (рис. 6).

Результатом работы алгоритма для изображения, представленного на рисунке 7, является текстовый файл (рис. 8) фиксированного формата, который содержит:

- блок правил деления;



**Рис. 5.** Пример работы алгоритма с параметрическими мутациями



**Рис. 6.** Пример работы алгоритма с привнесенными мутациями

- блок правил цветов;
- служебную строку с метаинформацией (размеры изображения, координаты начальной точки, длина правила, количество записей).

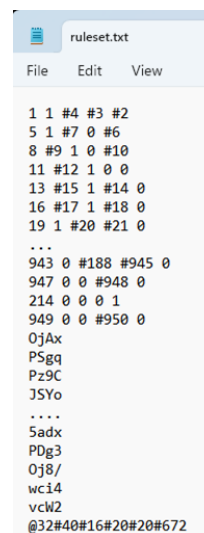
Первоначально в выходной файл помещаются правила деления (строки до первого троеточия). Они записываются подряд. При этом перед записью первого правила в системе находится самое длинное правило. И его длина становится фиксированной длиной для всех правил. Поэтому, если длина следующего правила меньше, чем фиксированная после него, запишется столько пустых символов, сколько необходимо для достижения фиксированной длины.

После правил деления записываются значения цветов (строки между первым и вторым троеточием).

В последней строке записывается строка вида «@1#2#3#4#5#6». Где знак «@» обозначает конец правил цвета, числа «1» и «2» обозначают ширину и длину обработанного изображения соответ-



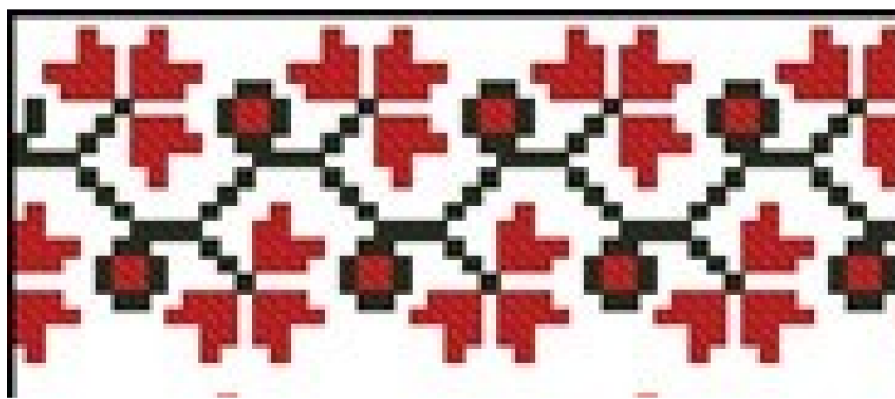
**Рис. 7.** Пример изображения, по которому сформирован выходной файл, представленный на рисунке 8



**Рис. 8.** Пример сформированного файла с правилами

ственно. Числа «3» и «4» обозначают координаты X и Y соответственно, где должно воспроизвестись первое правило. «5» — фиксированная длина правила, чтобы система могла сразу сослаться на другие правила посредством расчетов. «6» — обозначает количество правил деления, чтобы отделить правила деления от правил цветов.

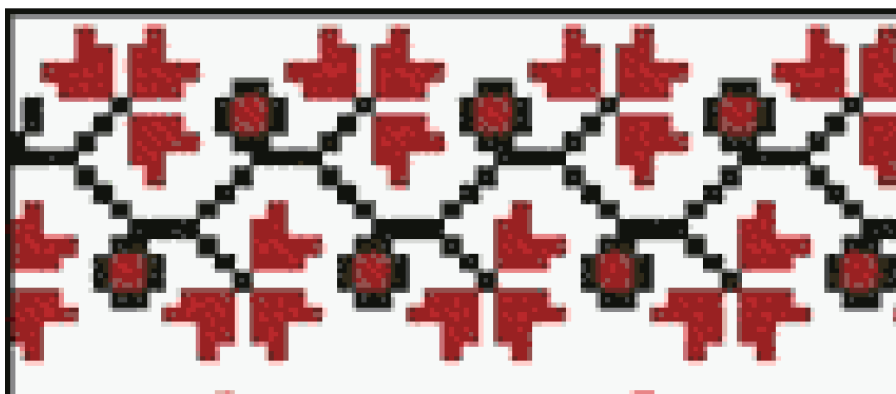
Алгоритм был протестирован на ряде изображений. При обработке картин и орнаментов наблюдались различные проявления мутаций. Параметрические мутации приводили к появлению дополнительных оттенков, а привнесенные — к изменению топологии изображения. На примере портрета удалось продемонстрировать потерю фрагментов лица при привнесенных мутациях, что иллюстрирует аналогию с биологическими сбоями в процессе деления клеток. Примеры представлены на рисунках 9–13.



**Рис. 9.** Исходное изображение орнамента

В качестве исходных данных использовались изображения небольшого разрешения (20×20, 32×40 и 184×80 пикселей), что позволило проанализировать эффективность работы алгоритма и характер возникающих мутаций.

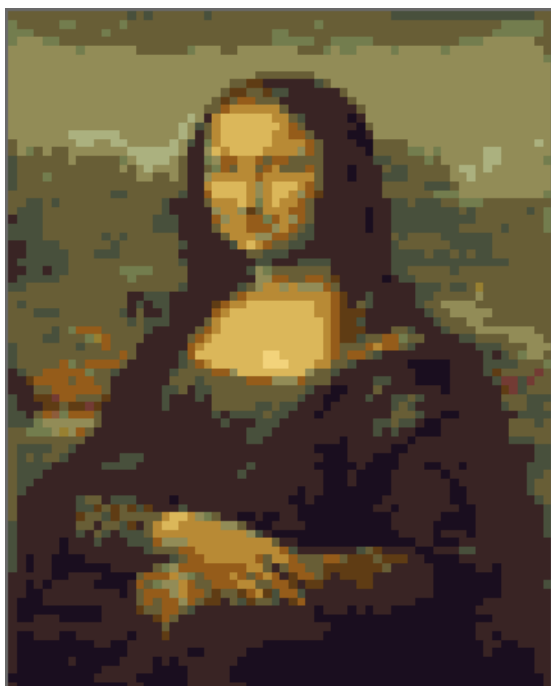
На простых тестовых изображениях система корректно формировала набор правил, число которых находилось в пределах сотен строк. Например, для изображения размером 20×20 пикселей было получено 212 правил деления и 2 правила цвета. При воспроизведении выходного файла объект полностью совпадал с исходным, что подтверждает правильность работы алгоритма при инвариантной свертке (рис. 14–15).



**Рис. 10.** Изображение орнамента с параметрическими мутациями



**Рис. 11.** Изображение орнамента с привнесенными мутациями



**Рис. 12.** Исходное изображение портрета



**Рис. 13.** Изображение портрета с привнесенными мутациями

### **Заключение**

Предложенный подход позволяет не только сохранять и воспроизводить исходные объекты, но и моделировать их эволюцию через мутации. Это делает систему перспективной для использования в

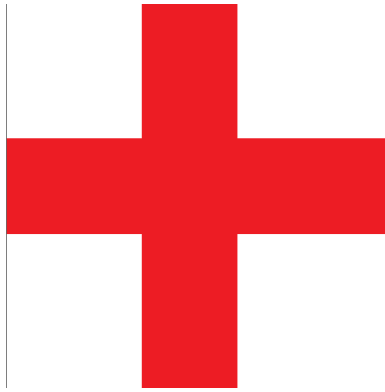


Рис. 14. Исходное изображение

```

1 #1 #1 #1 1 1 0 0 1 #1 1 #1 0 0 1 2 0 #2 1 0 2 0 0 1 #2 2 0 0 0 1
1 #1 0 #1 1 1 0 0 1 #1 1 #1 0 0 1 2 0 #2 #2 1 2 0 0 #2 1 2 2 0 0 1
1 1 0 #1 #1 1 0 1 #2 #1 1 #1 0 0 1 2 0 0 1 0 2 0 0 1 #2 2 0 0 0 2
1 1 0 0 #1 1 0 1 #2 0 1 #1 0 1 #1 2 0 0 1 0 2 0 0 1 #2 2 0 0 0 #2
1 1 #1 0 #1 1 1 #1 #2 0 1 0 0 0 1 2 0 #2 1 0 2 0 1 #2 #2 2 0 0 #2 1
1 1 #1 0 #1 1 0 1 0 0 1 0 0 #1 1 2 0 #2 1 0 2 0 #2 1 0 2 0 0 #2 1
1 1 #1 0 #1 1 #1 1 0 0 1 #1 0 1 #1 2 0 #2 1 #2 2 0 #2 1 0 2 0 0 2 1
1 1 #1 0 #1 1 #1 1 #1 0 1 0 0 0 1 1 0 1 #1 #1 2 0 1 0 2 2 #2 0 0 0
1 #1 #1 0 1 1 1 0 2 0 1 0 0 2 1 1 0 1 0 0 2 0 49 0 0 2 0 0 #2 1
1 1 0 0 #2 1 156 0 0 0 1 3 0 0 0 1 1 0 0 0 2 0 0 0 1 2 0 0 1 0 2 #2 1 0 2
1 1 0 0 #2 1 0 1 #2 0 1 0 0 0 1 2 1 #2 0 0 2 #2 0 0 1 2 0 0 1 0 2 0 6 0 0
1 #1 #1 0 1 1 0 0 1 0 1 0 0 #1 2 #2 1 0 0 2 #2 0 0 1 2 0 #2 1 0 2 1 0 0 #2
2 1 0 0 #2 1 0 1 #1 0 1 #1 0 0 1 2 1 #2 0 0 2 1 0 0 #1 2 0 1 #2 0 2 1 0 0 0
2 #2 0 0 1 1 0 #1 1 0 1 1 0 0 2 1 #2 0 0 2 1 0 0 #1 2 0 1 #2 0 2 #2 1 0 0
2 #2 0 0 1 1 0 1 #1 0 1 #1 0 0 1 2 1 #2 0 #2 2 #2 1 0 2 2 2 0 1 0 2 0 1 0 0
2 1 0 #2 #2 1 0 #1 1 0 1 1 0 0 0 2 #2 1 0 0 2 40 0 0 0 2 0 #2 2 0 2 0 0 #2 0
2 0 0 0 1 1 0 1 #1 0 1 1 0 0 0 2 #2 1 0 0 1 #1 0 0 1 2 #2 0 0 0 2 0 0 #2 0
2 0 0 0 1 1 0 1 #1 0 1 #1 0 0 2 1 2 0 2 #2 1 2 0 1 #1 0 #1 1 2 0 #2 0 1 7Rwk
2 0 0 #2 1 1 1 #2 #1 0 1 #1 1 0 0 2 7 0 0 0 1 #1 0 1 #1 2 0 0 0 #2 ////
2 0 0 #2 1 1 1 2 0 0 1 1 #2 0 0 2 0 #2 1 0 1 0 0 1 #1 1 0 1 #1 0 @20#20#10#10#13#212
2 0 0 1 0 1 0 0 0 #1 1 1 #2 0 0 2 0 1 #2 0 1 0 #1 0 1 1 #1 0 0 0
2 0 0 1 0 2 #2 1 0 0 1 1 #2 0 #1 2 0 0 1 0 1 0 0 1 2 1 0 0 0
2 0 1 #2 0 2 #2 1 #2 0 1 1 #2 0 #1 2 0 0 1 0 1 #1 0 0 1 2 0 0 0 #2
2 0 #2 1 0 2 1 #2 #2 0 1 #1 1 0 2 2 0 0 0 1 2 #2 0 0 1 2 #2 1 0 0
2 0 #2 1 0 2 #1 1 0 0 2 7 #2 0 0 2 0 0 0 #2 2 1 0 0 #2 2 1 0 0 0
2 0 #2 #2 1 2 #1 1 0 0 1 2 0 0 1 2 1 0 0 0 2 1 0 0 #2 2 1 0 0 0
2 0 0 1 0 2 #1 1 #2 0 1 0 0 4 0 2 0 0 0 1 2 #2 #2 0 1 2 1 0 0 #2
2 0 0 1 0 2 1 #2 #2 0 1 0 0 0 1 2 0 0 0 1 2 1 0 0 #2 2 #2 0 0 1
2 0 #2 1 0 1 1 #1 0 0 1 0 0 0 1 2 0 0 0 #2 2 1 0 0 #2 2 0 0 0 #2
1 0 1 #1 0 1 #1 1 0 #1 1 0 0 #1 #1 2 0 0 1 0 2 #2 1 0 #2 1 #1 0 0 1
1 0 1 #1 0 1 #1 1 0 0 1 0 0 #1 0 2 0 0 #2 1 2 #2 1 0 0 1 #1 0 0 1
1 #2 #1 1 0 1 2 0 1 0 2 0 1 0 0 2 0 0 1 0 2 2 #1 1 0 1 1 0 0 0
1 0 #1 1 0 1 0 0 65 0 2 0 1 0 0 2 0 0 #2 1 2 0 21 0 0 1 1 0 0 0
1 0 #1 #1 1 1 1 0 0 0 2 0 #2 1 0 2 0 0 #2 1 2 1 #1 0 0 2 1 #2 0 #2

```

Рис. 15. Сформированные правила по изображению на рисунке 14

биоинформатике (имитация процессов наследования и мутаций), робототехнике (самоорганизующиеся структуры), а также в адаптивных вычислительных системах.

Ограничением текущей версии является работа с простой геометрией и ограниченным набором параметров. Перспективы дальнейшего развития включают поддержку сложных 3D форм, оптимизацию параметров правил и учет функционала сворачиваемого объекта.

#### ЛИТЕРАТУРА

1. Сергеев И. Ю. *Нервная система: анатомия, физиология, нейрофармакология*. М.: Юрайт; 2020. 373 с.
2. Огурцов А. Н. *Основы биоинформатики: учебное пособие*. Харьков: НТУ «ХПИ»; 2013. 400 с.

3. Матка осы — как выглядит. *7Ogorod — интернет-энциклопедия о сельском хозяйстве*. Режим доступа: <https://7ogorod.ru/prochee/matka-osy.html>.
4. Acidum ribonucleicum. *Википедия — научная энциклопедия*. Режим доступа: [https://la.wikipedia.org/wiki/Acidum\\_ribonucleicum](https://la.wikipedia.org/wiki/Acidum_ribonucleicum).
5. Бессмертный И. А., Нугуманова А. Б., Платонов А. В. *Интеллектуальные системы: учебник*. М.: Юрайт; 2017. 250 с.
6. Советов Б. Я., Яковлев С. А. *Моделирование систем*. М.: Юрайт; 2011. 343 с.
7. Коваленко В. В. *Проектирование информационных систем*. М.: Форум; 2012. 320 с.
8. Дезоксирибонуклеиновая кислота. Общие сведения. *Propionix — пищевые технологии и биопродукты*. Режим доступа: <https://propionix.ru/dnk-prokariot-i-eukariot>.