

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ  
ИЗОБРАЖЕНИЙ В ТЕХНИКЕ STRING ART**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование  
информационных систем

факультета компьютерных наук и информационных технологий

Инюткина Ивана Алексеевича

Научный руководитель:

Старший преподаватель \_\_\_\_\_ Казачкова А.А.

Зав. кафедрой:

Ученая степень, звание \_\_\_\_\_ Огнева М.В.

Саратов 2026

## ВВЕДЕНИЕ

Современное развитие информационных технологий, компьютерной графики и цифрового искусства приводит к активному внедрению алгоритмических методов генерации изображений в художественную и инженерную деятельность. В последние годы особую популярность получили системы генеративного искусства, использующие математические модели и алгоритмы обработки изображений для автоматического создания визуального контента.

Одним из направлений генеративного искусства является String Art (ниткография), представляющая собой способ формирования изображения посредством натягивания нитей между гвоздями, расположенными по контуру определенной формы. Визуальное представление изображения достигается за счёт различной плотности нитей.

Актуальность работы обусловлена ростом интереса к генеративному и алгоритмическому искусству, использующему методы компьютерной графики, цифровой обработки изображений и математического моделирования, а также развитием автоматизированного производства художественных объектов с применением ЧПУ-станков, роботизированных систем и технологий цифровой фабрикации.

**Цель бакалаврской работы** – разработка веб-приложения для автоматической генерации изображений в технике String Art.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Провести обзор существующих аналогов.
2. Исследовать методы обработки изображений и алгоритмы компьютерной графики.
3. Реализовать алгоритм генерации изображения.
4. Реализовать алгоритм размещения гвоздей и построения линий между ними.
5. Спроектировать архитектуру веб-приложения для генерации String Art изображений.
6. Реализовать серверную часть приложения.

7. Реализовать клиентскую часть приложения.
8. Реализовать механизм потоковой визуализации процесса генерации изображения в режиме реального времени.
9. Выполнить оптимизацию производительности алгоритмов генерации и использования памяти.
10. Реализовать экспорт результатов генерации в форматы, пригодные для использования в ЧПУ-системах.
11. Провести тестирование разработанного программного обеспечения.

**Методологическую основу** работы составляют исследования в области компьютерной графики, цифровой обработки изображений, алгоритмов генеративного искусства и высокопроизводительных вычислений. В работе использовались методы обработки изображений, алгоритм Брезенхема, подход Residual Image Approach, Error-Based Optimization, жадные алгоритмы выбора линий (Greedy Algorithms), методы итеративного уточнения решения (Iterative Refinement), технологии многопоточной обработки данных и клиент-серверные архитектуры.

Методологические основы представлены в работах M. Fraile-Narváez, M. I. Chidean, S. Bresenham, J. D. Foley, A. van Dam, A. Watt, J. Hughes, D. Knuth и других исследователей в области компьютерной графики, алгоритмов визуализации и цифрового производства.

**Теоретическая значимость работы** заключается в систематизации существующих подходов к генерации String Art изображений, анализе современных методов оптимизации алгоритмов построения ниточных изображений, а также исследовании возможностей применения методов цифровой обработки изображений в задачах генеративного искусства.

**Практическая значимость** заключается в разработке функционирующего веб-приложения, позволяющего автоматически генерировать изображения в технике String Art, осуществлять потоковую визуализацию процесса генерации и экспортировать результаты в форматы JSON, CSV и G-code для последующего использования в ЧПУ-системах.

**Структура и объём работы.** Бакалаврская работа состоит из введения, трёх разделов, заключения, списка использованных источников и тринадцати

приложений. Общий объём работы составляет 120 страниц. Основное содержание занимает 61 страницу. Работа содержит 13 приложений с исходным кодом программных модулей и конфигурационных файлов. Список использованных источников начинается с 62 страницы.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Первый раздел** «Анализ методов и алгоритмов генерации String Art изображений»

Первый раздел посвящён исследованию существующих подходов к автоматической генерации изображений в технике String Art, анализу современных программных решений, а также изучению методов компьютерной графики и цифровой обработки изображений, используемых при создании ниточных изображений.

В начале раздела рассмотрены существующие программные средства для генерации String Art изображений. Выполнен анализ популярных веб-сервисов Picgen.ru, Michael-crum.com и Stringart3d.com. Для каждого решения были изучены возможности настройки параметров генерации, поддерживаемые форматы экспорта, способы размещения гвоздей и качество итогового результата. Анализ показал, что большинство существующих решений обладают рядом ограничений: недостаточной гибкостью настройки параметров, отсутствием потоковой визуализации процесса генерации, ограниченными возможностями оптимизации производительности и отсутствием полноценной интеграции с ЧПУ-системами.

Далее рассмотрены теоретические основы технологии String Art. Показано, что формирование изображения достигается за счёт различной плотности пересечения нитей между опорными точками. В отличие от традиционной растровой графики, визуальное восприятие изображения строится на распределении линий и их наложении друг на друга. Более тёмные области изображения содержат большее количество соединений между гвоздями, тогда как светлые области формируются меньшим числом линий.

Особое внимание уделено методам цифровой обработки изображений, которые используются на этапе подготовки исходных данных. Рассмотрены операции масштабирования изображений, преобразования в оттенки серого, нормализации яркости и контрастности, а также инверсии изображения. Выполнение данных операций позволяет подготовить изображение для дальнейшего анализа алгоритмом генерации и повысить качество итогового

результата.

В разделе подробно исследованы основные алгоритмические подходы к генерации String Art изображений. Рассмотрен метод Residual Image Approach, основанный на последовательном уменьшении остаточной ошибки между исходным изображением и формируемым результатом. Изучен подход Error-Based Optimization, предполагающий выбор очередной линии на основе минимизации ошибки аппроксимации. Также проанализированы жадные алгоритмы выбора линий и методы итеративного уточнения решения.

Проведено сравнение рассмотренных методов по критериям качества генерации, вычислительной сложности и требованиям к вычислительным ресурсам. По результатам анализа был сделан вывод о целесообразности применения жадного алгоритма выбора линий в сочетании с механизмом остаточной ошибки, поскольку данный подход обеспечивает высокую скорость работы при сохранении приемлемого качества итогового изображения.

Кроме того, рассмотрены современные методы оптимизации производительности, включая использование алгоритма Брезенхема для построения линий, предварительное вычисление данных, применение многопоточной обработки и оптимизацию использования оперативной памяти.

По итогам первого раздела были определены основные требования к разрабатываемой системе и выбран алгоритмический подход, положенный в основу программной реализации.

## **Второй раздел «Проектирование системы генерации String Art»**

Во втором разделе выполнено проектирование программной системы автоматической генерации изображений в технике String Art.

На первом этапе проектирования была сформулирована постановка задачи и определены основные функциональные требования к программному обеспечению. Разрабатываемая система должна обеспечивать загрузку исходного изображения, настройку параметров генерации, визуализацию процесса построения изображения в режиме реального времени, экспорт результатов в различные форматы и возможность дальнейшего

использования полученных данных для управления ЧПУ-оборудованием.

Для реализации указанных требований была выбрана клиент-серверная архитектура. Данный подход позволяет разделить вычислительную часть приложения и пользовательский интерфейс. Серверная часть выполняет ресурсоёмкие вычисления, связанные с обработкой изображений и генерацией линий, а клиентская часть отвечает за взаимодействие пользователя с системой и отображение промежуточных и итоговых результатов.

В рамках проектирования архитектуры были определены основные программные модули системы. Серверная часть включает модуль обработки изображений, модуль генерации гвоздей, модуль генерации линий, модуль экспорта результатов и модуль взаимодействия через HTTP API. Клиентская часть содержит компоненты пользовательского интерфейса, средства визуализации процесса генерации и инструменты управления параметрами алгоритма.

Особое внимание уделено проектированию взаимодействия между frontend и backend частями приложения. Для передачи данных используется REST API, позволяющий запускать генерацию изображений, выполнять экспорт результатов и получать информацию о состоянии системы. Разработанный интерфейс взаимодействия обеспечивает независимость клиентской и серверной частей и допускает дальнейшее расширение функциональности без изменения существующих компонентов.

Для организации визуализации процесса генерации была выбрана технология Server-Sent Events. Использование SSE позволяет поддерживать постоянное соединение между сервером и клиентом и передавать промежуточные результаты генерации по мере их появления. Благодаря этому пользователь может наблюдать процесс построения изображения в режиме реального времени без необходимости ожидания завершения всех вычислений.

Также в разделе выполнено проектирование структуры данных, используемых при генерации изображений. Определены форматы хранения информации о гвоздях, линиях, параметрах генерации и промежуточном

состоянии алгоритма.

Отдельно рассмотрено проектирование подсистемы экспорта результатов. Предусмотрена возможность сохранения данных в форматах JSON, CSV и G-code. Использование формата G-code обеспечивает интеграцию разработанного программного обеспечения с оборудованием числового программного управления и позволяет использовать результаты генерации при изготовлении физических объектов.

В результате проектирования была сформирована масштабируемая архитектура программной системы, обеспечивающая высокую производительность, удобство сопровождения и возможность дальнейшего развития приложения.

### **Третий раздел «Программная реализация»**

Третий раздел посвящён практической реализации разработанной системы генерации изображений в технике String Art.

Серверная часть приложения реализована на языке программирования Go. Выбор данного языка обусловлен его высокой производительностью, эффективной работой с памятью и встроенной поддержкой конкурентных вычислений. Клиентская часть реализована с использованием фреймворка Next.js и технологии Canvas API, обеспечивающих интерактивное взаимодействие пользователя с системой.

На этапе предобработки выполняется подготовка изображения к генерации. Исходное изображение масштабируется до заданных размеров, преобразуется в оттенки серого и сохраняется в виде массива яркостей. Дополнительно реализованы механизмы регулировки яркости, контрастности и гамма-коррекции. Для корректного формирования ниточной композиции предусмотрена возможность инверсии яркости изображения.

Реализован модуль генерации гвоздей, отвечающий за формирование опорных точек, между которыми строятся линии. Пользователь может задавать количество гвоздей и их расположение. В разработанной системе поддерживаются различные схемы размещения гвоздей, что позволяет использовать рамки различной формы.

Основой программной реализации является модуль генерации линий.

Алгоритм использует жадный подход и анализирует множество возможных соединений между текущим гвоздём и всеми остальными кандидатами. Для оценки качества каждой линии вычисляется её влияние на остаточную ошибку изображения. Линия, обеспечивающая максимальное улучшение результата, выбирается для добавления в композицию.

Для построения и анализа линий используется алгоритм Брезенхема. Его применение позволяет эффективно обходить пиксели линии без использования операций с плавающей точкой и значительно уменьшает вычислительные затраты.

Одним из наиболее важных этапов реализации стала оптимизация производительности системы. Поскольку количество потенциальных соединений между гвоздями может достигать десятков тысяч, последовательный перебор кандидатов существенно замедляет генерацию изображения. Для решения данной проблемы была реализована параллельная обработка кандидатов с использованием горутин языка Go. Количество воркеров определяется автоматически на основе числа доступных логических ядер процессора.

Дополнительное ускорение достигнуто за счёт использования inline-реализации алгоритма Брезенхема, применения таблиц предвычисления и оптимизации работы с памятью. Вместо хранения промежуточных растровых изображений система обновляет только те пиксели, через которые проходит очередная линия. Такой подход позволил значительно сократить объём используемой памяти и уменьшить количество операций выделения ресурсов.

Для передачи промежуточных результатов генерации реализован механизм потоковой передачи данных посредством Server-Sent Events. Сервер формирует поток данных, содержащий информацию о новых линиях, а клиентская часть отображает изменения на Canvas в режиме реального времени. Использование пакетной передачи данных позволило снизить сетевые накладные расходы примерно на 90 процентов.

В ходе тестирования установлено, что реализованные методы оптимизации обеспечивают скорость генерации порядка 800–1300 линий в

секунду при потреблении около 50 МБ оперативной памяти. Полученные показатели превосходят аналогичные JavaScript-реализации в среднем в 2,5–4 раза.

Завершающим этапом реализации стала разработка подсистемы экспорта. Полученные результаты могут сохраняться в форматах JSON, CSV и G-code. Формат G-code позволяет использовать результаты работы приложения для управления ЧПУ-оборудованием и автоматизированного создания физических объектов в технике String Art.

Таким образом, в третьем разделе была выполнена полная программная реализация разработанной системы, подтверждена работоспособность выбранных алгоритмов и доказана эффективность применённых методов оптимизации.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы разработано веб-приложение для автоматической генерации изображений в технике String Art с возможностью экспорта результата для ЧПУ-оборудования.

Поставленные задачи решены в полном объёме. Проведён анализ существующих аналогов и выявлены их недостатки. Спроектирована клиент-серверная архитектура: серверная часть реализована на Go, клиентская — на Next.js с использованием Canvas API.

Реализован алгоритм генерации на основе жадного подхода с остаточной ошибкой, использующий inline-реализацию алгоритма Брезенхема. Передача данных между backend и frontend осуществляется через Server-Sent Events с батчевой отправкой, что снижает сетевой overhead на 90%.

Применённые оптимизации — inline Bresenham, таблицы предвычисления и параллелизация через горютины — обеспечили скорость генерации 800–1300 линий в секунду, что в 2.5–4 раза превышает производительность JavaScript-реализации при потреблении памяти около 50 МБ против 400 МБ.

Результаты экспортируются в форматы G-code, JSON и CSV. Приложение развёрнуто на облачном сервере и доступно по сети. Архитектура системы допускает горизонтальное масштабирование и расширение функциональности без изменения клиентской части.

## **Основные источники информации:**

1. Fraile-Narváez M., Chidean M.I. Between machines and art: The impact of CNC technology on artistic creation // Array. — 2025. — DOI: 10.1016/j.array.2024.100369. [Электронный ресурс]. — URL: <https://www.sciencedirect.com/science/article/pii/S2590005625000931> (Дата обращения: 12.10.2025)
2. Lau Z.H., Teh C.S., Gobi N., Zain N.H.M. Automated String Art Creation: Integrated Advanced Computational Techniques and Precision Art Designing // IEEE Access. — 2025. — DOI: 10.1109/ACCESS.2025.3530970. [Электронный ресурс]. URL: <https://ieeexplore.ieee.org/document/10844087/> (Дата обращения: 20.10.2025)
3. A computational approach for spider web-inspired fabrication of string art / Korea Advanced Institute of Science and Technology [Электронный ресурс]. — URL: <https://pure.kaist.ac.kr/en/publications/a-computational-approach-for-spider-web-inspired-fabrication-of-s> (Дата обращения 22.10.2025)
4. Picgen.ru — онлайн-генератор String Art [Электронный ресурс]. — URL: <https://picgen.ru> (Дата обращения: 25.12.2025)
5. Michael-crum.com — String Art Generator [Электронный ресурс]. — URL: <https://michael-crum.com/string-art> (Дата обращения: 10.01.2026)
6. Stringart3d.com — String Art 3D Generator [Электронный ресурс]. — URL: <https://stringart3d.com> (Дата обращения: 15.01.2026)
7. Gonzalez R.C., Woods R.E. Digital Image Processing. — 4th ed. — Pearson, 2018. — 1024 p.
8. ITU-R Recommendation BT.709-6. Parameter values for the HDTV standards for production and international programme exchange. — ITU, 2015.
9. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms. — 3rd ed. — MIT Press, 2009. — 1292 p.
10. Fog A. Optimizing software in C++. — Technical University of Denmark, 2023. [Электронный ресурс]. URL: [https://www.agner.org/optimize/optimizing\\_cpp.pdf](https://www.agner.org/optimize/optimizing_cpp.pdf) (Дата обращения: 27.02.2026)