**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение высшего образования**
**«Московский государственный технический университет имени Н.Э. Баумана**
**(национальный исследовательский университет)»**
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ» (ИУ)

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» (ИУ7)

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

## *НА ТЕМУ:*

## *Классификация методов создания текстур посредством генерации шумов*

| | | | |
|---|---|---|---|
| Студент | **ИУ7-53Б** | _________________ | **Авдейкина В. П.** |
| Студент | **ИУ7-55Б** | _________________ | **Амбарцумова Е. А.** |
| Студент | **ИУ7-55Б** | _________________ | **Саблина П. Р.** |
| Руководитель | | _________________ | **Кострицкий А. С.** |

*2023 г.*

# РЕФЕРАТ

Расчетно-пояснительная записка 20 с., 9 рис., 1 табл., 24 ист.

ШУМ, ГЕНЕРАЦИЯ ШУМА, ТЕКСТУРА, СОЗДАНИЕ ТЕКСТУР, АНИЗОТРОПНЫЙ ШУМ, ВЕЙВЛЕТ-ШУМ, ШУМ ПЕРЛИНА.

Объектом исследования являются методы создания текстур посредством генерации шумов.

Цель работы — классификация существующих методов создания текстур посредством генерации шумов.

В процессе иседования проанализирована предметная область создания текстур посредством генерации шумов.

В результате исследования предложены и разработаны критерии сравнения методов создания текстур посредством генерации шумов и классифицированы 3 алгоритмические реализации существующих методов.

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

За последние 15 лет генерация текстур не перестала быть актуальной задачей компьютерной графики и упоминается в [1], [2], [3], [4]. Таким образом, возникает необходимость нахождения методов, которые обеспечивают решение этой задачи. Существует ряд таких методов, суть которых заключается в использовании генерации шумов (в использовании шумов).

Цель работы — классификация существующих методов создания текстур посредством генерации шумов.

Для достижения поставленной цели требуется решить следующие задачи:

1) проанализировать предметную область создания текстур посредством генерации шумов;
2) описать основные подходы к решению задачи создания текстур посредством генерации шумов;
3) сформулировать критерии сравнения применяемых методов;
4) выполнить сравнение методов в рамках сформулированных критериев.

# 1 Анализ предметной области

Стандартной задачей компьютерной графики является визуализация изображений, способы осуществления которой делятся на растровый и векторный [5], [6]. При использовании растрового способа изображение $X$ представляется в виде набора точек (пикселей) [5], [6]:

$$X = (x_1, \dots, x_n), \tag{1.1}$$

где $x_i$ — целое число, равное значению одной из RGB-компонент цвета пикселя.

Текстура — растровое изображение, используемое в компьютерной графике для изменения внешнего вида поверхностей и других объектов без изменения их формы [7].

Псевдослучайный метод получения чисел (точек) — метод получения случайных чисел (точек) с возможностью повтора набора при повторе входных данных [8].

Шум — набор точек, полученный псевдослучайным методом [9].

Таким образом, одним из способов создания текстур является генерация шумов, результатом которой является набор точек или функций, к которым требуется применить операции для получения необходимых значений.

Согласно [10], для сравнения методов создания текстур с использованием генерации шумов выделяются следующие критерии:

1) требуемый объем памяти;
2) возможность хранения результатов в виде функций;
3) возможность создания текстуры, внешний вид которой не зависит от положения точки, на которую она накладывается, в пространстве;
4) возможность создания текстуры, внешний вид которой зависит от положения точки, на которую она накладывается, в пространстве.

Далее будут рассмотрены существующие решения задачи — методы генерации шумов, которые лежат в основе соответствующих методов создания текстур: шум Перлина, анизотропный шум, вейвлет-шум.

# 2 Обзор существующих решений

## 2.1 Шум Перлина

В [11], [12], [13] метод генерации шума Перлина описан следующим образом:

1) задать целочисленную решетку, то есть множество всех точек в пространстве, координаты которых являются целыми числами;

2) задать массив $P$, содержащий псевдослучайную перестановку, и массив $G$, содержащий псевдослучайный набор градиентов единичной длины;

3) с помощью хэш-функции $g = H(x, y, z)$, использующей массивы $P$ и $G$, связать с каждой точкой куба целочисленной решентки псевдослучайный градиент;

4) выполнить интерполяцию по трем направлениям восьми скалярных произведений $g_{i,j,k} \cdot (x - i, y - j, z - k)$, где $i, j, k$ — координаты точек куба целочисленной решетки, а $g_{i,j,k}$ — градиент в точке $(i, j, k)$.

В [13] уточняется, что в оригинальной версии алгоритма генерации шума Перлина для интерполяции по трем направлениям была выбрана функция сглаживания

$$s(t) = 3t^2 - 2t^3, \tag{2.1}$$

где $t$ — это значение координаты точки. Вторая производная функции (2.1) $6 - 12t$ не равна нулю при $t = 0$ и $t = 1$. Это ненулевое значение создает на выровненных по координатам гранях соседних кубических ячеек разрывы второго рода [13], влияние которых на внешний вид полученной текстуры отображено на рисунке 1.

Для решения этой проблемы функция (2.1) заменяется на

$$6t^5 - 15t^4 + 10t^3, \tag{2.2}$$

так как ее первая и вторая производные равны нулю при $t = 0$ и $t = 1$ [13]. На рисунке 2 из [13] приведен результат замены функции сглаживания.

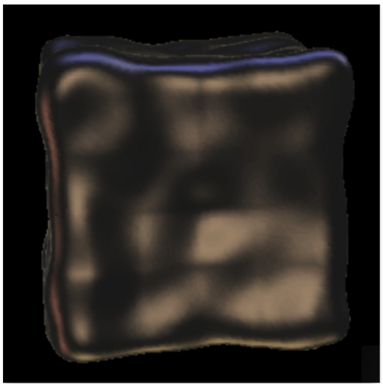

Рисунок 1 — Шум Перлина, полученный с использованием функции сглаживания (2.1), рисунок Кена Перлина [13]

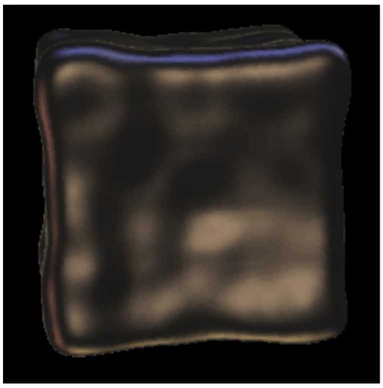

Рисунок 2 — Шум Перлина, полученный с использованием функции сглаживания (2.2), рисунок Кена Перлина [13]

Следующей проблемой, описанной в [13], является то, что в массиве $G$ содержатся градиенты единичной длины, но в кубической решетке расстояния от центра куба до его ребер укорочены вдоль осей и удлинены вдоль диагоналей между противоположными углами куба. Это приводит к тому, что близлежащие градиенты сливаются друг с другом, из-за чего шумовая функция принимает аномально высокие значения в этих областях [13], что отображено на рисунке 3 из [13].

Для решения данной проблемы массив $G$ заменяют набором из 12 векторов, определяющих направления от центра куба до его ребер. Чтобы избежать затрат на деление на 12, Перлином предложено увеличить набор до 16 градиентов, добавляя в него дополнительные $(1, 1, 0)$, $(-1, 1, 0)$, $(0, -1, 1)$ и $(0, -1, -1)$ [13]. На рисунке 4 из [13] приведен результат замены массива градиентов $G$.

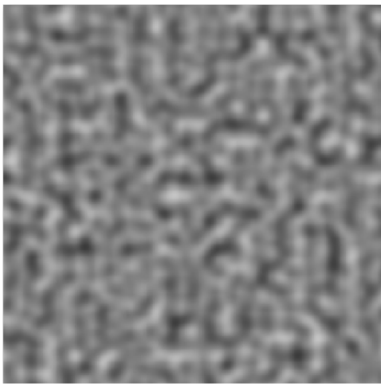

Рисунок 3 — Шум Перлина, полученный с использованием старого массива градиентов $G$, рисунок Кена Перлина [13]

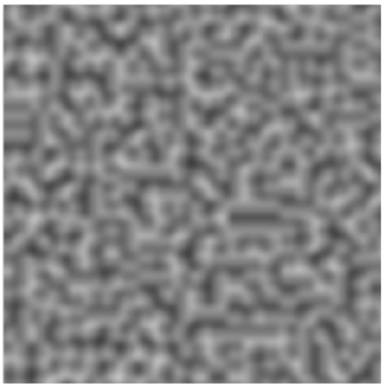

Рисунок 4 — Шум Перлина, полученный с использованием нового массива градиентов $G$, рисунок Кена Перлина [13]

## 2.2 Анизотропный шум

Генерация анизотропного шума [14] на поверхности разбивается на несколько этапов:

1) задание метрики,

2) генерация эллипсов,

3) применение фильтра Гаусса.

Далее перечисленные этапы рассматриваются более подробно.

### 2.2.1 Задание метрики

Тензорное поле — это отображение, которое каждой точке рассматриваемого пространства ставит в соответствие тензор [15].

Число $\lambda$ называется собственным значением тензора $A$, если найдется вектор $x$ такой, что $A \cdot x = \lambda \cdot x$. Вектор $x$ называется собственным вектором тензора $A$, соответствующим данному собственному значению [16].

Для первого этапа генерации образца анизотопного шума для визуализации тензорного поля $T$, определенного в пространстве $D \in R^2$, необходима метрика $g$, которая задает свойства образца анизотопного шума [14]. В тензорном поле $T$ $\lambda_1$ и $\lambda_2$ — собственные значения, $v_1$ и $v_2$ — соответствующие собственные векторы. Метрика для генерации образца определяется как

$$g = \frac{1}{\sqrt{\lambda_1}} \cdot v_1 \cdot v_1^T + \frac{1}{\sqrt{\lambda_2}} \cdot v_2 \cdot v_2^T \tag{2.3}$$

Полученные образцы представляют собой эллипсы, выровненные по $v_1$ и $v_2$ и масштабируемые по $\lambda_1$ и $\lambda_2$ [14].

### 2.2.2 Генерация эллипсов

Эллипсы определяются полуосями и центром.

Согласно [14] полуоси совпадают с собственными векторами, а квадраты главных радиусов масштабируются по обратным собственным значениям:

$$a^2(x_0, y_0) = \frac{1}{\lambda_1(x_0, y_0)} \tag{2.4}$$

$$b^2(x_0, y_0) = \frac{1}{\lambda_2(x_0, y_0)} \tag{2.5}$$

В произвольно заданном наборе точек генерируется эллипс — произвольная точка выбирается как центр текущего эллипса [14]. Произвольным образом выбирается центр первого генерируемого эллипса. Затем точки, находящиеся внутри границ сгенерированного эллипса, помечаются как неподходящие, так как сгенерированные на них эллипсы будут пересекаться.

Далее начинается перебор еще не обработанных точек — тех, на которых еще не сгенерирован эллипс или не отмечены, как неподходящие.

На текущей итерации на выбранной точке строится эллипс.

На рисунке 5 показан процесс отбора эллипсов для образца.

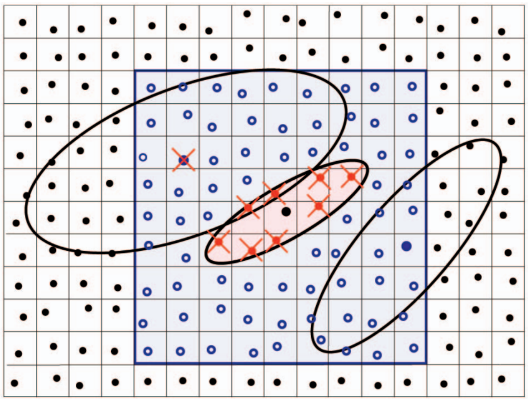

Рисунок 5 — Выбор подходящего для выборки эллипса

Если эллипс пересекается с уже построенными эллипсами, точка его центра помечается как неподходящая. Если не пересекается, точка центра помечается как центр эллипса, а точки внутри границы текущего эллипса помечаются как неподходящие.

### 2.2.3 Применение фильтра Гаусса

Фильтр Гаусса — способ снижения детализации изображения и шума в изображении [17]. В фильтре Гаусса используется функция Гаусса, которая для каждого пикселя вычисляет преобразование. Пиксель задан набором $(x, y)$, $\sigma$ — среднеквадратичное отклонение нормального распределения.

Функция Гаусса согласно [18]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.6}$$

Для получения готовой текстуры необходимо применить к образцу фильтр Гаусса. На рисунке 6 представлены примеры образца до применения фильтра Гаусса и после.

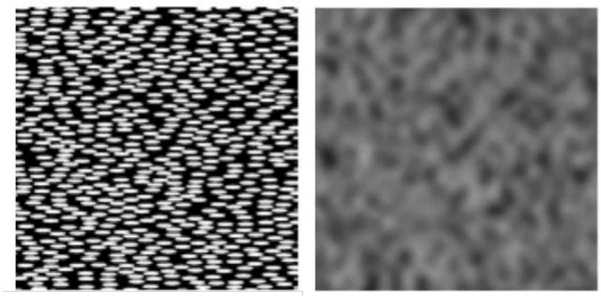

Рисунок 6 — Образец без применения фильтра Гаусса (слева), с применением фильтра Гаусса (справа)

## 2.3 Вейвлет-шум

### 2.3.1 Математические основы

Суть вейвлет-шума заключается в использовании вейвлетов.

Вейвлет — функция независимой переменной, имеющая вид короткой волны (всплеска) [19—21]. Пример вейвлета представлен на рисунке 7.

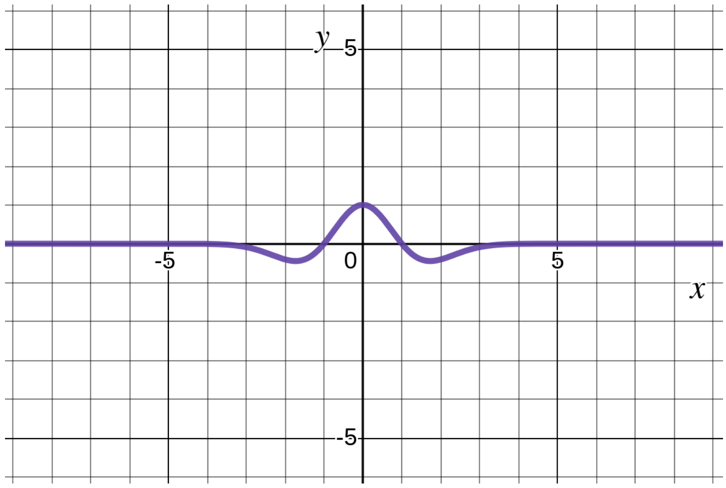


Рисунок 7 — Вейвлет, центр которого $x = 0$

Центр вейвлета — значение независимой переменной, через которое проходит вертикальная ось симметрии вейвлета.

Пространство вейвлетов $W$ — набор функций, которые могут быть представлены линейной комбинацией сдвигов и масштабирования одного вейвлета [22; 23].

Пример пространства вейвлетов представлен на рисунке 8.

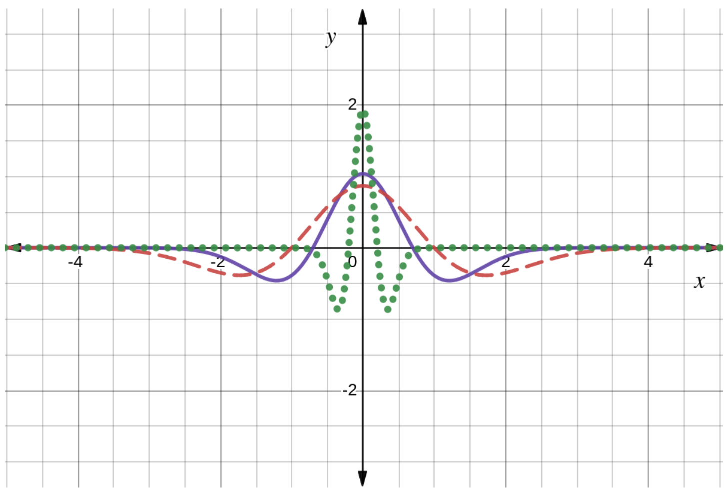


Рисунок 8 — Пространство вейвлетов

Масштабирующая функция $\xi$ — функция, для которой справедливо следующее равенство:

$$\xi(x) = \sum_k p_k \xi(2x - k), \tag{2.7}$$

при условии, что существуют такие числовые коэффициенты $p_k$ [22; 24].

Согласно [24], если $\phi(x)$ — функция из $W$, центр которой $x = 0$, то любая функция $F(x)$ может быть представлена следующим образом:

$$F(x) = \sum_i f_i \phi(x - i), \tag{2.8}$$

где $\phi(x - i)$ — функция из $W$ с центром $x = i$, $f_i$ — некоторые числовые коэффициенты.

Пространство $S^0$ — пространство, состоящее из функций вида 2.8.

Функции $G(x)$ вида

$$G(x) = \sum_i g_i \phi(2x - i) \tag{2.9}$$

аналогично $S^0$ составляют пространство $S^1$. [24]

Если рассматриваемая в 2.9, 2.8 функция $\phi$ является масштабирующей, то $S^1$ расширяет $S^0$, то есть включает все функции $S^0$ [22; 24].

Вейвлет-анализ — процесс определения того, расширяет ли пространство $S^1$ пространство $S^0$ [24].

С учетом того, что функция $F(x)$ из 2.8 является масштабирующей, она представляется с помощью коэффициентов $f_i^{\uparrow}$ в пространстве $S^1$:

$$F(x) = \sum_i f_i^{\uparrow} \phi(2x - i) \tag{2.10}$$

Коэффициенты $f_i^{\uparrow}$ выражаются с использованием формулы 2.7:

$$f_i^{\uparrow} = \sum_k p_{i-2k} f_k \tag{2.11}$$

Согласно [24], коэффициенты $g_i^{\downarrow}$, необходимые для представления функции $G(x)$ из 2.9 в пространстве $S^0$, вычисляются с помощью вейвлет-анализа и являются равными

$$g_i^{\downarrow} = \sum_k a_{k-2i} g_k \tag{2.12}$$

Функция $G(x)$ в пространстве $S^0$ представляется как

$$G(x) = G^{\downarrow}(x) + D(x) \tag{2.13}$$

где $D(x)$ — функция из $S^1$, которая не может быть выражена в $S^0$ через какие-либо коэффициенты, а $G^{\downarrow}$ выражается через $g_i^{\downarrow}$.

### 2.3.2 Описание

Входные данные вейвлет-шума: функция $B(x)$ из $W$, изображение $R$.

Выходные данные вейвлет-шума: изображение $N$.

Изображение $X$ представляется упорядоченным набором числовых коэффициентов $\ldots, x_i, \ldots$:

$$X = (\ldots, x_i, \ldots) \tag{2.14}$$

Авторы [24] представляют алгоритм вейвлет-шума тремя семантическими шагами:

1) получение $R(x)$;
2) получение $R^{\downarrow}(x)$;
3) получение $R^{\downarrow\uparrow}(x)$;
4) получение $N(x)$.

Пример визуализации получения результата алгоритма представлен на рисунке 9.

Рассмотрим представленные шаги более подробно, основываясь на информации из [24].

Согласно формуле 2.14 изображение $R$ представляется следующим образом:

$$R = (\ldots, r_i, \ldots) \tag{2.15}$$

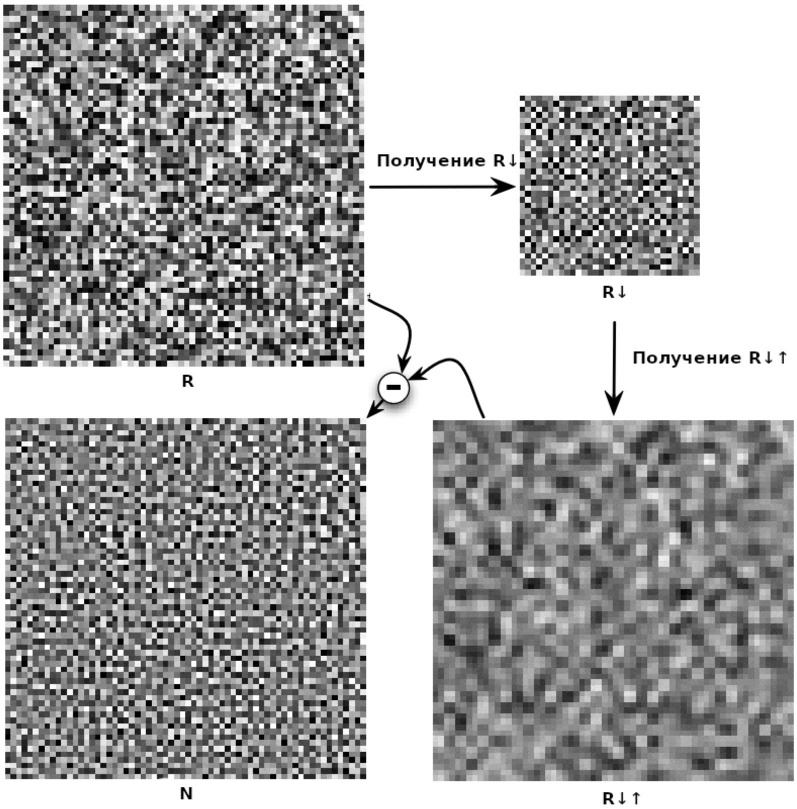


Рисунок 9 — Визуализация получения результата алгоритма вейвлет-шума согласно [24]

Аналогично формуле 2.9 выражается $R(x)$:

$$R = \sum_i r_i B(2x - i) \tag{2.16}$$

Согласно формуле 2.13:

$$R(x) = R^{\downarrow}(x) + N(x) \tag{2.17}$$

$N(x)$ выражается из формулы 2.17 как

$$N(x) = R(x) - R^{\downarrow}(x) \tag{2.18}$$

С помощью формулы 2.8 выражается

$$R^{\downarrow}(x) = \sum_i r_i^{\downarrow} B(x - i) \tag{2.19}$$

где $r_i^{\downarrow}$ представляются с помощью формулы 2.12:

$$r_i^{\downarrow} = \sum_k a_{k-2i} r_k \tag{2.20}$$

С учетом формулы 2.10 и 2.19 получается выражение

$$R^{\downarrow\uparrow}(x) = \sum_i r_i^{\downarrow\uparrow} B(2x - i), \tag{2.21}$$

где коэффициенты $r_i^{\downarrow\uparrow}$ выражаются с учетом формулы 2.11:

$$r_i^{\downarrow\uparrow} = \sum_k p_{i-2k} r_k^{\downarrow} \tag{2.22}$$

С помощью формул 2.16 и 2.19 выражение 2.18 преобразовывается как

$$N(x) = \sum_i r_i B(2x - i) - \sum_i r_i^{\downarrow} B(x - i) \tag{2.23}$$

С использованием формулы 2.21 выражение 2.23 записывается как

$$N(x) = \sum_i r_i B(2x - i) - \sum_i r_i^{\downarrow\uparrow} B(2x - i) = \sum_i n_i' B(2x - i) \tag{2.24}$$

где $n_i' = r_i - r_i^{\downarrow\uparrow}$.

Для получения изображения $N = (\dots, n_i, \dots)$ значение функции $N(x)$ вычисляется при задаваемых значениях $x$.

# 3 Сравнение существующих методов создания текстур посредством генерации шумов

На основе [10] осуществлено сравнение методов создания текстур, использующих следующие методы генерации шумов: шум Перлина, анизотропный шум, вейвлет-шум.

В таблице 1 приведены результаты сравнения, критерии которого расположены по горизонтали и включают в себя следующие:

1) требуемый объем памяти (в зависимости от периода шума $N$ и количества измерений пространства $d$);

2) возможность хранения результатов в виде функций;

3) возможность создания текстуры, внешний вид которой не зависит от положения точки, на которую она накладывается, в пространстве;

4) возможность создания текстуры, внешний вид которой зависит от положения точки, на которую она накладывается, в пространстве.

Таблица 1 — Сравнение существующих методов создания текстур посредством генерации шумов

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Шум Перлина | $O(N)$ | Да | Да | Нет |
| Анизотропный шум | $O(N^d)$ | Нет | Нет | Да |
| Вейвлет-шум | $O(N^d)$ | Да | Да | Да |

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы цель была достигнута — классифицированы методы генерации текстур с использованием генерации шумов.

В ходе выполнения работы были решены поставленные задачи:

1) проанализирована предметная область генерации текстур с использованием шумов;
2) сформулированы критерии классификации методов генерации текстур с использованием шумов;
3) описаны существующие методы генерации текстур с использованием шумов;
4) выполнено сравнение выбранных методов в рамках сформулированных критериев классификации.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. *Pai H.-Y.* Texture designs and workflows for physically based rendering using procedural texture generation // 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE). — 2019. — С. 195—198.
2. Weakly-Supervised Photo-realistic Texture Generation for 3D Face Reconstruction / X. Yin [и др.] // 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG). — 2023. — С. 1—8.
3. Digital preservation of Brazilian indigenous artworks: Generating high quality textures for 3D models / B. Trinchão Andrade [и др.] // Journal of Cultural Heritage. — 2012. — Т. 13, № 1. — С. 28—39.
4. A Papier-Mâché Approach to Learning 3D Surface Generation / T. Groueix [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2018.
5. *Порев В.* Компьютерная графика. — СПб : БХВ-Петербург, 2002. — С. 432.
6. *Боресков А. В.*, *Шикин Е. В.* Компьютерная графика. — 2017.
7. *Сморкалов А. Ю.* Математическая и программная модели генерации текстур на графических потоковых процессорах // Программные системы и вычислительные методы. — 2013. — № 1. — С. 116—128.
8. *Мухамеджанов Д. Д.*, *Левина А. Б.* Генератор псевдослучайных чисел на основе клеточных автоматов // Научно-технический вестник информационных технологий, механики и оптики. — 2018. — Т. 18, № 5. — С. 894—900.
9. *Петрова Д. О.*, *Полина А. О.* Использование шума Перлина при создании спецэффектов в компьютерной графике в кинопроизводстве // Вопросы технических и физико-математических наук в свете современных исследований. — 2022. — С. 5—8.
10. A survey of procedural noise functions / A. Lagae [и др.] // Computer Graphics Forum. Т. 29. — Wiley Online Library. 2010. — С. 2579—2600.
11. Texturing and Modeling: A Procedural Approach (third edition) / S. E. David [и др.] //. — Morgan Kaufmann Publishers, 2003. — Гл. 6.

12. *Perlin K.* An Image Synthesizer // SIGGRAPH 85. — 1985. — № 19. — С. 287—296.

13. *Perlin K.* Improving noise // SIGGRAPH 2002. — 2002. — С. 681—682.

14. Anisotropic noise samples / L. Feng [и др.] // IEEE Transactions on Visualization and Computer Graphics. — 2008. — Т. 14, № 2. — С. 342—354.

15. *Виноградов И.* Математическая энциклопедия. — Москва, 1965.

16. Собственные векторы и значения матриц [Электронный ресурс]. — Режим доступа: `https://lms2.sseu.ru/courses/eresmat/course1/razd8z1/par8_18z1.htm` (дата обращения: 06.12.2023).

17. *Мыцких-Коробанов А. Ю.* Алгоритм размытия по гауссу // Математика и ее приложения в современной науке и практике. — 2018. — С. 40—45.

18. A class of fast Gaussian binomial filters for speech and image processing / R. A. Haddad, A. N. Akansu [и др.] // IEEE Transactions on Signal Processing. — 1991. — Т. 39, № 3. — С. 723—727.

19. *Крыжевич Л.*, *Ковалев В.* Задача очистки изображения от шума и вейвлет-подходы к ее решению // Актуальные исследования в области математики, информатики, физики и методики их изучения в современном образовательном пространстве. — 2016. — Т. 1, № 1. — С. 39—44.

20. *Смоленцев Н.* Основы теории вейвлетов. Вейвлеты в MATLAB. — Москва : ДМК Пресс, 2014. — С. 628.

21. *Малла С.* Вэйвлеты в обработке сигналов. — Москва : Мир, 2005. — С. 671.

22. *Новиков Л.* Модифицированные вейвлеты в обработке данных аналитических приборов. I. основы теории. — 2006.

23. *Meyer Y.* Wavelets: Algorithms and applications. — Philadelphia : S.I.A.M., 1993. — С. 129.

24. *Cook R.*, *De Rose T.* Wavelet Noise [Электронный ресурс]. — Режим доступа: `https://graphics.pixar.com/library/WaveletNoise/paper.pdf` (дата обращения: 11.11.2023).