

МЕТОДЫ ГЕНЕРАЦИИ ТОЧЕК НА ПОВЕРХНОСТИ ПОЛУСФЕРЫ, ИСПОЛЬЗУЕМЫЕ В АЛГОРИТМАХ С ТРАССИРОВКОЙ ЛУЧЕЙ

Сакун И.Ю., Красковский П.Н.

Белорусский государственный университет информатики и радиоэлектроник
г. Минск, Республика Беларусь

Красковский П.Н. – ст. преподаватель

Аннотация. В статье рассмотрены различные методы генерации точек на поверхности полусферы, которые могут быть использованы в алгоритмах визуализации с использованием трассировки лучей. Представлен сравнительный анализ методов генерации точек на поверхности полусферы на примере вычисления окружающего затенения.

Ключевые слова. Уравнение рендеринга, метод Монте-Карло, равномерное распределение, решетка Фибоначчи.

Введение. Главной задачей визуализации трехмерной сцены является вычисление уравнения рендеринга [1] для каждой точки поверхности, видимой на экране:

$$L_o(v) = L_e(v) + \int_{\Omega} f_r(v, l) \cdot L_i(l) \cdot \cos(\theta) \cdot d\omega, \quad (1)$$

где $L_o(v)$ – яркость точки поверхности в направлении v ; $L_e(v)$ – яркость света, излученного самой поверхностью; $f_r(v, l)$ – двулучевая функция отражательной способности; $L_i(l)$ – яркость света, падающего на поверхность; θ – угол между нормалью к поверхности и направлением l .

На рисунке 1 представлена иллюстрация уравнения рендеринга.

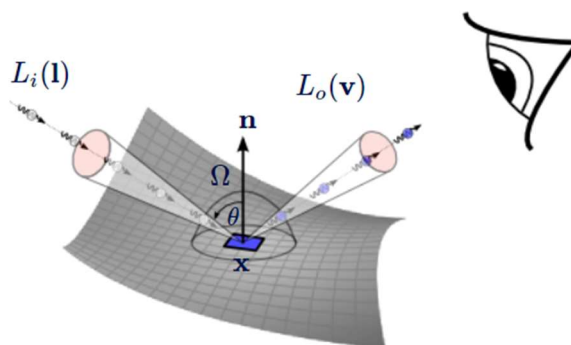


Рисунок 1 – Иллюстрация уравнения рендеринга

При переходе от телесного угла к сферическим углам уравнение рендеринга принимает следующий вид:

$$L_o(v) = L_e(v) + \int_0^{2\pi} \int_0^{\pi/2} f_r(v, l) \cdot L_i(l) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \cdot d\varphi. \quad (2)$$

Метод Монте-Карло. Поскольку на значения функции $L_i(l)$ может влиять множество различных факторов, вычисление интеграла из уравнения рендеринга необходимо выполнять численно, например, методом Монте-Карло. Решение интеграла методом Монте-Карло имеет следующий общий вид:

$$\int_a^b f(x) \cdot dx \approx \frac{1}{N} \sum_{n=1}^N \frac{f(x_n)}{p(x_n)}, \quad (3)$$

где x_n – некоторая случайная величина на отрезке $[a, b]$; $p(x)$ – плотность вероятности случайной величины.

Для ускорения сходимости интеграла необходимо выбирать значения x_n таким образом, чтобы значение $f(x_n)$ давало наибольший вклад в сумму. Для этого необходимо задать такое распределение случайной величины, при котором плотность вероятности $p(x)$ похожа на функцию $f(x)$. Такой метод выбора случайных значений функции $f(x)$ в методе Монте-Карло называется выборкой по значимости.

Также стоит отметить, что выбор случайной точки на поверхности полусферы зависит от случайного выбора углов φ и θ , поэтому распределение точек необходимо рассматривать как совместное распределение φ и θ .

Ниже рассмотрены наиболее распространенные методы распределения точек на поверхности полусферы.

Равномерное распределение. Наиболее простым методом генерации точек на полусфере является выбор случайных углов φ и θ с равномерным распределением.

Функция совместной плотности вероятности для равномерного распределения [2]:

$$p(\varphi, \theta) = \frac{1}{\pi/2} \cdot \frac{1}{2\pi} = \frac{1}{\pi^2}. \quad (4)$$

Функции распределения φ и θ примут следующий вид:

$$F(\varphi) = \frac{\varphi}{2\pi}, \quad (5)$$

$$F(\theta) = \frac{2\theta}{\pi}. \quad (6)$$

Для генерации φ и θ с заданными ограничениями необходимо вычислить обратные функции распределения и применить метод обратного преобразования. При этом обратные функции распределения примут следующий вид:

$$\varphi = 2\pi \cdot u, \quad (7)$$

$$\theta = \frac{\pi}{2} \cdot v. \quad (8)$$

где u и v – равномерные случайные величины на отрезке .

На рисунке 2 представлена визуализация равномерного распределения.

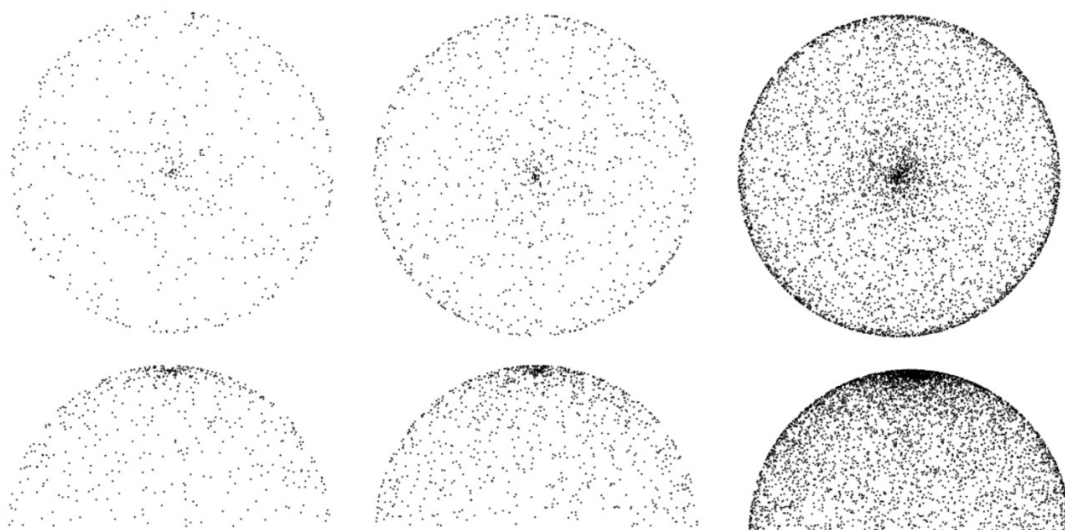


Рисунок 2 – Генерация точек на поверхности сферы с равномерным распределением (500 точек, 1000 точек, 5000 точек)

Из рисунка 2 видно, что большинство точек расположено ближе к полюсу, поэтому данное распределение подойдет для функций, которые дают наибольшие значения на полюсе полусферы.

Синусное распределение. Функция совместной плотности синусного распределения [2] имеет следующий вид:

$$p(\varphi, \theta) = \frac{1}{2\pi} \cdot \sin(\theta). \quad (9)$$

Функции распределения φ и θ :

$$F(\varphi) = \frac{\varphi}{2\pi}, \quad (10)$$

$$F(\theta) = 1 - \cos(\theta). \quad (11)$$

Обратные функции распределения:

$$\varphi = 2\pi \cdot u, \quad (12)$$

$$\theta = \arccos(1 - v). \quad (13)$$

На рисунке 3 представлена визуализация синусного распределения.

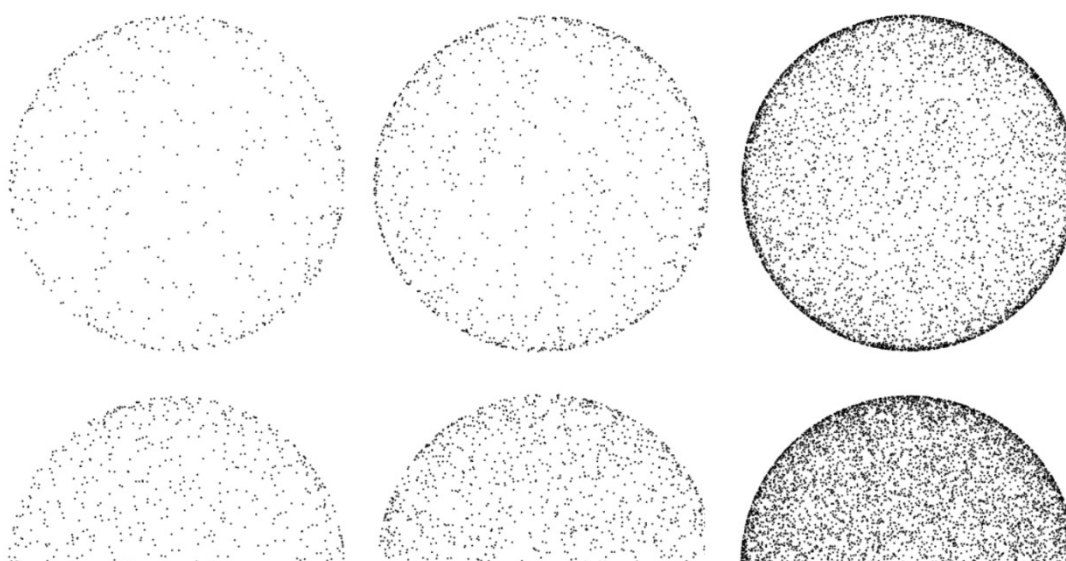


Рисунок 3 – Генерация точек на поверхности сферы с синусным распределением (500 точек, 1000 точек, 5000 точек)

Из рисунка 3 видно, что точки располагаются равномерно по всей поверхности полусферы. Такое распределение подойдет для функций, дающих одинаковое значение в любом из направлений.

Синусно-косинусное распределение. Функция совместной плотности синусно-косинусного распределения [2] имеет следующий вид:

$$p(\varphi, \theta) = \frac{1}{\pi} \cdot \sin(\theta) \cdot \cos(\theta). \quad (14)$$

Функции распределения φ и θ :

$$F(\varphi) = \frac{\varphi}{2\pi}, \quad (15)$$

$$F(\theta) = \sin^2(\theta). \quad (16)$$

Обратные функции распределения:

$$\varphi = 2\pi \cdot u, \quad (17)$$

$$\theta = \arcsin(\sqrt{v}). \quad (18)$$

На рисунке 4 представлена визуализация синусно-косинусного распределения.

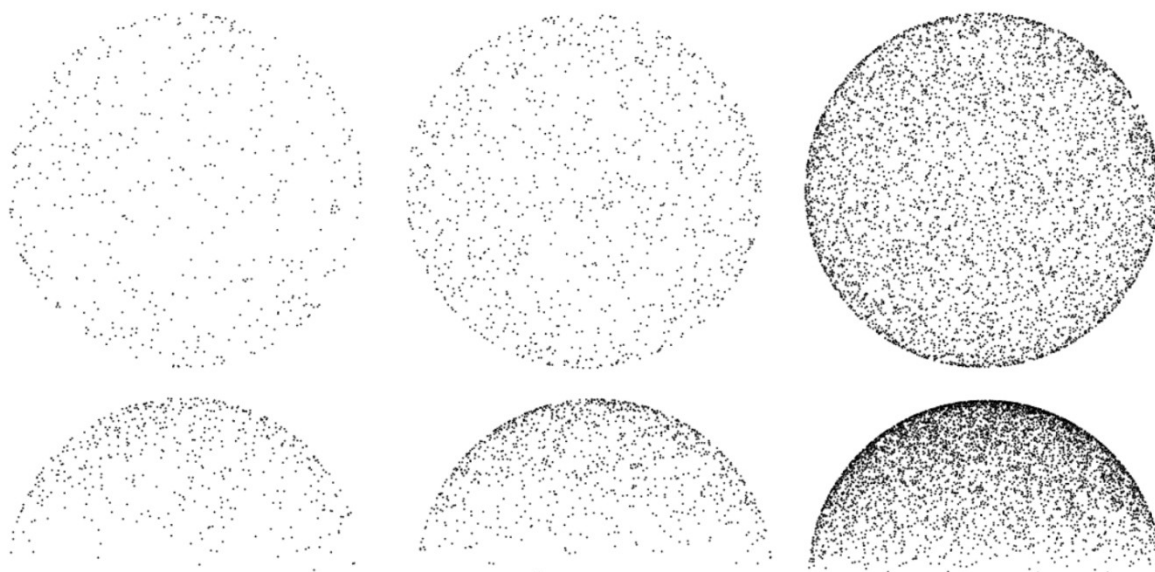


Рисунок 4 – Генерация точек на поверхности сферы с синусно-косинусным распределением (500 точек, 1000 точек, 5000 точек)

Из рисунка 4 видно, что плотность точек уменьшается по мере приближения к экватору, поэтому данное распределение подойдет для функций, дающих наибольшие значения между полюсом и экватором.

Решетка Фибоначчи. На рисунках 2-4 можно обратить внимание на то, что при небольшом количестве точек на поверхности полусферы имеются области с низкой плотностью точек, что негативно сказывается на сходимости интеграла. В контексте визуализации трехмерных сцен с применением трассировки лучей высокое расхождение точек на интегрируемой полусфере негативно сказывается на качестве получаемого изображения. Повышение количества генерируемых точек улучшает сходимость интегрирования, однако приводит к высоким потерям производительности.

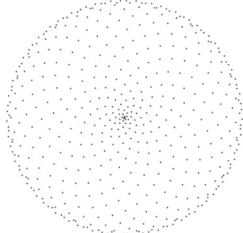
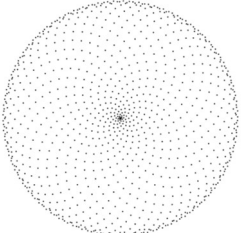
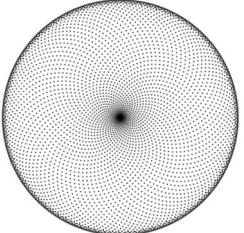
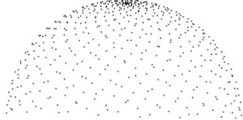
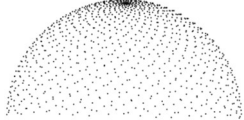
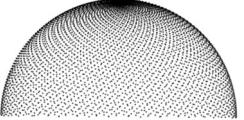
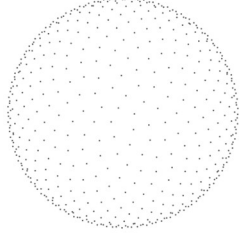
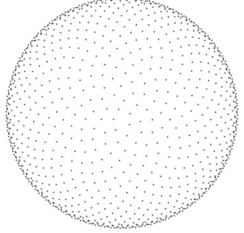
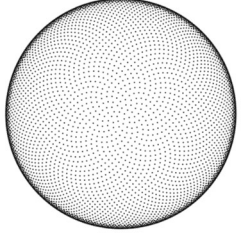
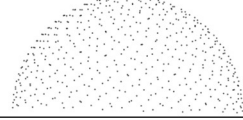
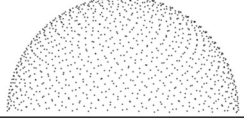
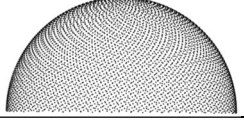
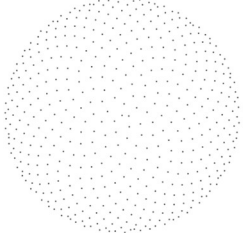
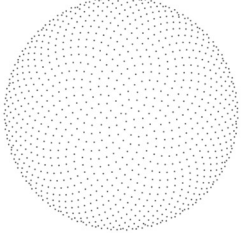
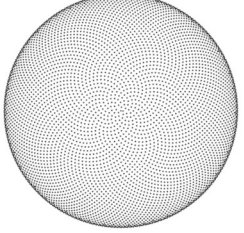
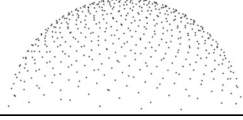
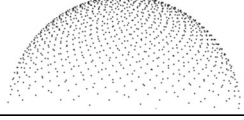
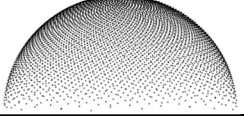
Для улучшения сходимости интеграла при небольшом количестве точек необходимо заменить генератор псевдослучайной последовательности на генератор последовательности с низким расхождением, также называемой квазислучайной последовательностью. Одной из таких последовательностей является решетка Фибоначчи [3]. Для двумерного случая решетка Фибоначчи задается следующим образом:

$$(x_i, y_i) = \left(\frac{i+1}{\phi} \bmod 1, \frac{i+0,5}{n} \right), \quad (19)$$

где n – количество точек; $\phi = \frac{1+\sqrt{5}}{2}$; $i = 0, 1, 2, \dots, n$.

В таблице 1 представлена визуализация решетки Фибоначчи с различными распределениями. Из таблицы видно, что расстояние между соседними точками постоянно по всей площади полусферы. Решетка Фибоначчи обеспечивает более равномерное заполнение полусферы небольшим количеством точек, что положительно сказывается на производительности и сходимости интеграла.

Таблица 1 – Решетка Фибоначчи на поверхности полусферы с различными распределениями

Распределение	Количество точек		
	500	1000	5000
Равномерное			
			
Синусное			
			
Синусно-косинусное			
			

Сравнительный анализ. На примере вычисления коэффициента окружающего затенения (ambient occlusion) был проведен сравнительный анализ описанных выше распределений. Коэффициент вычисляется путем нахождения соотношения между падающим на поверхность светом с учетом препятствий поблизости и падающим на поверхность светом при полном отсутствии препятствий. Преобразовав уравнение рендеринга (2) для случая окружающего затенения, было получено следующее уравнение:

$$AmbientOcclusionFactor = \frac{1}{\pi} \int_0^{2\pi} \int_0^{\pi/2} L_i(l) \cdot \cos(\theta) \cdot \sin(\theta) \cdot d\theta \cdot d\varphi, \quad (20)$$

где $L_i(l)$ принимает значение 1, если луч пересекается с препятствием на некотором расстоянии d от поверхности, и 0, если препятствия нет.

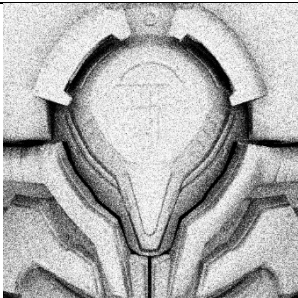
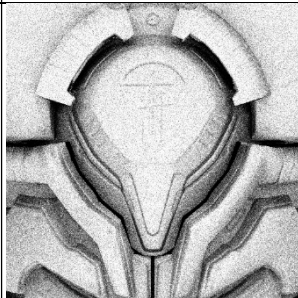
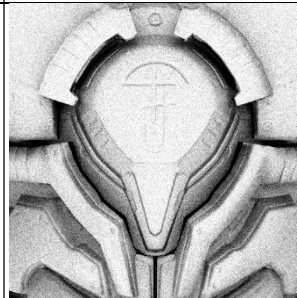
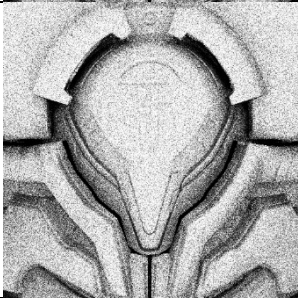
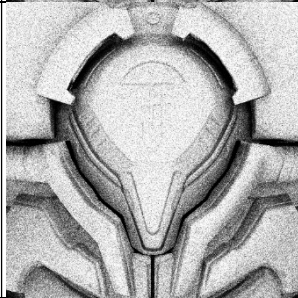
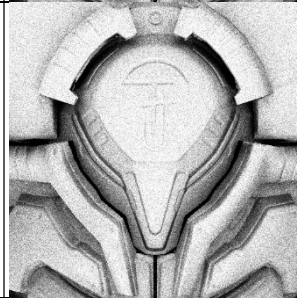
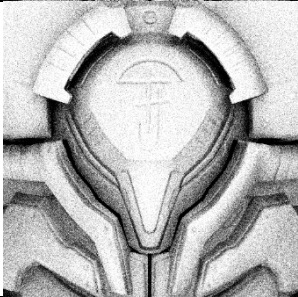
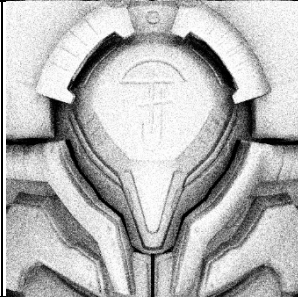
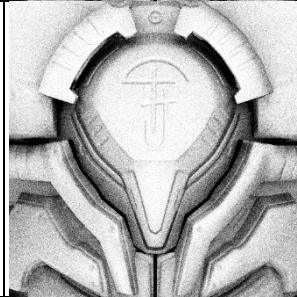
Из уравнения (20) можно сделать вывод, что наилучшим распределением для вычисления окружающего затенения является синусно-косинусное распределение, плотность вероятности (14) имеет такую же форму, как и полученное уравнение рендеринга (20).

В таблице 2 представлен сравнительный анализ различных распределений с использованием псевдослучайной последовательности для генерации углов φ и θ .

Из таблицы 2 видно, что синусно-косинусное распределение действительно показало наилучший результат по сравнению с другими распределениями. Результат, полученный с помощью синусно-косинусного распределения, имеет меньше шумов на светлых участках при любом количестве лучей.

Также стоит отметить, что синусно-косинусное распределение имеет более значительное улучшение результата при увеличении количества лучей по сравнению с другими распределениями.

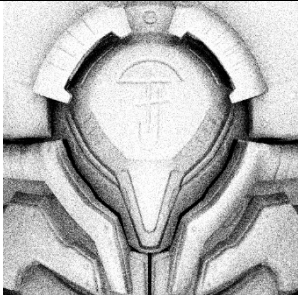
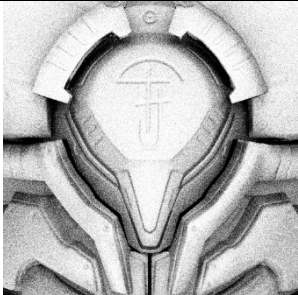
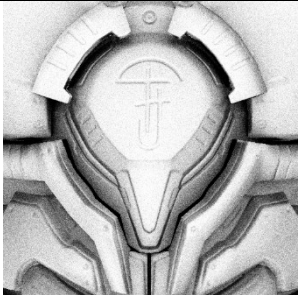
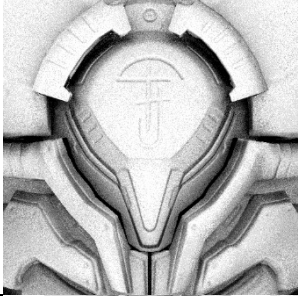
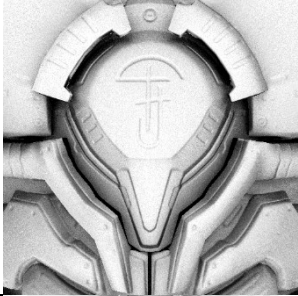
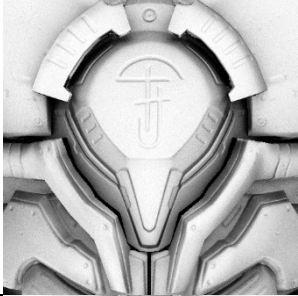
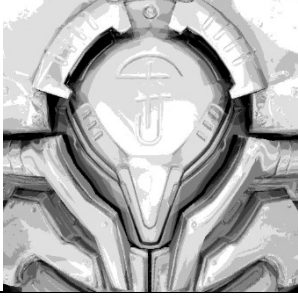
Таблица 2 – Сравнительный анализ распределений, псевдослучайная последовательность

Распределение	Количество лучей		
	8	16	32
Равномерное			
Синусное			
Синусно-косинусное			

В таблице 3 представлен сравнительный анализ псевдослучайной последовательности и решетки Фибоначчи на примере синусно-косинусного распределения. Из таблицы видно, что использование решетки Фибоначчи вместо псевдослучайной последовательности приводит к значительному улучшению результата при любом количестве лучей. Решетка Фибоначчи позволяет достигнуть хорошего результата при меньшем количестве лучей, что положительно сказывается на производительности.

Также стоит отметить, что при использовании решетки Фибоначчи крайне важно добавлять некоторое случайное зерно к изначальным номерам точек на каждый пиксель изображения. Случайное зерно обеспечивает случайный поворот решетки вокруг нормали. Использование статического зерна приводит к ступенчатости получаемого затенения, что негативно сказывается на общем визуальном качестве итогового изображения (см. таблицу 3).

Таблица 3 – Сравнительный анализ псевдослучайной последовательности и решетки Фибоначчи

Распределение	Количество лучей		
	8	16	32
Псевдослучайная последовательность			
Решетка Фибоначчи (случайное зерно на каждый пиксель)			
Решетка Фибоначчи (статическое зерно)			

Вывод. В работе рассмотрены различные методы генерации точек на поверхности полусферы. Показана важность выбора распределения при вычислении интегралов методом Монте-Карло. Проведен анализ преимуществ использования квазислучайных последовательностей для задач, требующих равномерного заполнения поверхности. Проведен сравнительный анализ теоретических данных на примере вычисления окружающего затенения. Практические результаты полностью соответствуют теоретическим расчетам.

Список использованных источников:

1. Graphics Programming. Light and Materials [Электронный ресурс]. – Режим доступа: https://www.mathematik.uni-marburg.de/~thormae/lectures/graphics1/graphics_10_1_eng_web.html#1. – Дата доступа: 11.04.2025.
2. Graphics Programming Image-based Lighting [Электронный ресурс]. – Режим доступа: https://www.mathematik.uni-marburg.de/~thormae/lectures/graphics1/graphics_10_2_eng_web.html#1. – Дата доступа: 11.04.2025.
3. How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice [Электронный ресурс]. – Режим доступа: <https://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/>. – Дата доступа: 11.04.2025

METHODS FOR GENERATING POINTS ON THE SURFACE OF A HEMISPHERE USED IN RAY TRACING ALGORITHMS

Sakun I.Y., Kraskovsky P.N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Kraskovsky P.N. – senior lecturer

Annotation. The paper discusses various methods for generating points on the surface of a hemisphere that can be used in ray tracing visualization algorithms. A comparative analysis of methods for generating points on the surface of a hemisphere is presented using the example of calculating ambient occlusion.

Keywords. Rendering equation, Monte Carlo method, uniform distribution, Fibonacci lattice.