

DOI 10.58351/2949-2041.2025.22.5.008

Малышев Иван Алексеевич, магистрант,
МГТУ им. Н. Э. Баумана, Москва

Кивва Кирилл Андреевич,
старший преподаватель
МГТУ им. Н. Э. Баумана, Москва

МЕТОДЫ ДИНАМИЧЕСКОЙ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ ЛАНДШАФТА

Аннотация: В статье проводится обзор существующих методов динамической процедурной генерации ландшафта. Кратко описаны актуальность темы, понятие динамической процедурной генерации ландшафта, и примеры реализации методов. Сделаны выводы о преимуществах и недостатках рассматриваемых методов, а также приведены области применения методов.

Ключевые слова: динамическая процедурная генерация ландшафта, карта высот, смещение средней точки, шум Перлина, генеративно-состязательная сеть, методы моделирования эрозии.

Введение

Генерация ландшафта является одной из задач процедурной генерации объектов в компьютерной графике. Процедурная генерация ландшафта нацелена на алгоритмическое создание ландшафта с использованием элемента случайности в виде карт высот, отражающих различные рельефы, например, горы и долины. Такой подход позволяет ускорить процесс разработки виртуальных ландшафтов. Генерация ландшафта развивалась вместе с компьютерной графикой, виртуальным моделированием и компьютерными играми [1].

Динамическая процедурная генерация ландшафта — это метод создания ландшафта в реальном времени [2]. В процессе перемещения виртуальной камеры вдоль трехмерной модели ландшафта создаются новые области. Это создает иллюзию “безграничного” ландшафта. Также может быть предусмотрена адаптивная генерация ландшафта различного качества, когда детализация ландшафта зависит от расстояния до наблюдателя.

Примерами использования динамической процедурной генерации ландшафта являются такие компьютерные игры, как *Minecraft* [3] и *No Man's Sky* [4].

Обзор существующих методов динамической процедурной генерации ландшафта Метод на основе случайного смещения средней точки

Данный метод в качестве значения высоты некоторой точки ландшафта использует среднее арифметическое высот точек, между которыми данная точка находится, с прибавлением некоторого случайного смещения. Реализацией метода для двумерного случая является алгоритм *Diamond-Square* [5].

Алгоритм начинает работу с плоской карты высот, размер которой соответствует формуле $2^n + 1$ точек, $n \in \mathbb{N}$. В четырех угловых точках карты высот устанавливаются начальные значения высот. Шаги *diamond* и *square* выполняются поочередно до тех пор, пока все значения карты высот не будут установлены. Шаг *diamond* — исходное изображение делится на четыре квадранта и устанавливается срединная точка, которой присваивается среднее арифметическое значений четырех угловых точек, прибавляя случайное значение. Шаг *square* — берутся средние точки граней тех же квадрантов, в которые устанавливается среднее значение от четырех соседних с ними по осям точек с прибавлением случайного значения. Далее каждый квадрант интерпретируется как новые изображения и с ними выполняются те же шаги. Случайное число варьируется в промежутке $[-R^i, R^i]$, где R — фактор шероховатости в промежутке от 0 до 1, а i — номер итерации (шаг *diamond* и шаг *square* — одна итерация). Соответственно, при каждой итерации модуль случайного значения, прибавляющегося к срединным точкам, уменьшается [5].



Настройка значения фактора шероховатости влияет на смену высот ландшафта по мере углубления в рекурсивный процесс. Значения фактора шероховатости, близкие к 1, ведут к созданию более хаотичного ландшафта, тогда как с приближением значения фактора шероховатости к нулю ландшафт становится все более гладким [6].

Алгоритм также позволяет индивидуально корректировать диапазоны случайных смещений в зависимости от координат или высоты, что может быть использовано для придания рельефу более размытого вида путем умножения размера диапазона смещений на среднее значение высоты при расчете новых значений. Это приводит к тому, что участки на низкой высоте становятся более гладкими, имитируя тем самым отложение эродированного материала [7].

Данный метод может порождать дефекты из-за так называемой “проблемы сгибания”, которая заключается в возникновении складок или неоднородностей наклона вдоль границ [8]. Пример проблемы представлен на рисунке 1.



Рис. 1. Пример “проблемы сгибания” алгоритма *Diamond-Square* [8]

Методы на основе градиентного шума

Градиентный шум (шумовая функция) — разновидность алгоритмов генерации шума, основанных на создании решетки случайных градиентов, которые затем интерполируются для получения значений в точках, лежащих между узлами решетки [9]. Для генерации ландшафта используются такие разновидности градиентного шума, как шум Перлина и симплексный шум.

Шум Перлина

Шум Перлина — n -мерный градиентный шум, состоящий из набора псевдослучайных единичных векторов (направлений градиента), расположенных в определенных точках пространства и интерполированных функцией сглаживания между этими точками. Для генерации ландшафта используется двумерный вариант: в этом случае направления градиента располагаются в узлах регулярной квадратной сетки [10].

Задание значения высоты некоторой точки происходит в 2 этапа.

а) Вычисление скалярных произведений между направлением градиента и вектором, направленным из того же угла сетки, что и направление градиента, к той точке, для которой вычисляется высота.

б) Взвешенная интерполяция четырех значений с учетом близости текущей позиции к каждому углу ячейки сетки с помощью функции смешивания.

Скалярные произведения дадут значения высоты каждого из узлов сетки. Далее выполняются три операции смешивания: два смешивания значений высот по оси x и смешивание значений по оси y [6].

К функции смешивания предъявляется одно требование: она должна иметь вторую производную на конечных точках, равную нулю, чтобы шумовая функция была непрерывна на узлах сетки [10]. Такому требованию соответствует следующий полином:

$$w = 6t^5 - 15t^4 + 10t^3, \quad (1)$$

где t — параметр, $w(t)$ — функция смешивания. Кривая пятой степени как раз имеет нулевую вторую производную при $t = 0$ и $t = 1$, то есть шумовая функция – непрерывная везде [10].

В результате вычисляется конечное значение высоты в диапазоне от 0 до 1 в выбранной точке пространства, которое затем масштабируется с учетом желаемого диапазона высот [6]. Для контроля генерации шума используется набор параметров:

- масштаб (scale) – число пикселей на единицу длины сетки;
- октавы (octaves) – количество уровней детализации шума;
- лакунарность (lacunarity) – множитель, который определяет изменение частоты с ростом уровня детализации;
- стойкость (persistence) – множитель частотной амплитуды, который определяет изменение амплитуды с ростом октавы.

Таким образом, значение высоты точки с координатами (x, y) вычисляется следующим образом:

$$h(x, y) = \sum_{i=1}^{N_o} a_p^i \cdot \text{noise} \left(\frac{x \cdot a_l^i}{S}, \frac{y \cdot a_l^i}{S}, w(t) \right), \quad (2)$$

где $h(x, y)$ — высота в точке (x, y) , S — масштаб, N_o — количество октав, a_l — лакунарность, a_p — стойкость, $\text{noise}(x, y, w(t))$ — шумовая функция.

Для достижения качественных ландшафтов нужно комбинировать шумы разных масштабов и с разными количествами октав, складывая или перемножая их, увеличивая тем самым уровень детализации генерируемого ландшафта [6].

Симплексный шум

Симплексный шум является модификацией предыдущего метода, привносящей следующие улучшения [9]:

- меньшая вычислительная сложность и меньшее количество умножений;
- шум масштабируется в высшие размерности с меньшей потерей производительности;
- отсутствуют визуальные артефакты;
- шум имеет хорошо определённый непрерывный градиент, который легко вычислить;
- алгоритм достаточно прост для аппаратной реализации.

Данный алгоритм также использует регулярную сетку, только состоящую из симплексов, n -мерного обобщения треугольника. Такой подход позволяет уменьшить количество смешиваний для вычисления высоты выбранной точки [9].

В двумерном случае регулярная (симплексная) сетка состоит из равносторонних треугольников. Интерполяция производится так же, как и в шуме Перлина, то есть между значениями в углах участка пространства. Однако в случае с симплексной решеткой нужно интерполировать только три угла. Симплексную решетку можно получить из квадратной решетки следующим образом: разделить клетки обычной квадратной решетки на два равнобедренных треугольника, а затем наклонить решетку так, чтобы треугольники стали равносторонними [9].

Одной из основных проблем классического шума Перлина является то, что он предполагает последовательное интерполирование вдоль каждой оси размерности градиента. Симплексный шум вместо этого использует прямое суммирование значений вклада градиентов по каждой опорной точке текущего шаблона, где вклад представляет умножение экстраполяции градиента опорных точек шаблона на коэффициент радиально-симметричного ослабления функции. Значение радиального затухания тщательно подобрано так, чтобы оно достигало нуля к моменту пересечения границы со следующим шаблоном симплексной сетки. Это означает, что точки внутри симплекса шаблона будут зависеть только от вклада градиентов опорных точек [10].

Другим основным этапом в работе метода является сопоставление искомой точки, в которой необходимо вычислить значение шумовой функции, с конкретным симплекс-шаблоном. Для этого нужно перевести координаты исходной точки в пространство клетки, составленной из



двух симплексов, к которой она принадлежит, а далее, сравнив абсолютные значения x и y , можно определить принадлежность точки к верхнему или нижнему симплексу, после чего останется обойти три корректных угловых точки [9].

Для контроля генерации шума используются те же параметры, что и для шума Перлина. Также данный алгоритм позволяет комбинировать шумы разных масштабов и количества октав. Все это является следствием того, что симплексный шум является модификацией шума Перлина.

Симплексный шум сложнее в реализации и реже используется по сравнению с шумом Перлина: симплексный шум применяют в тех случаях, где требуется высокая производительность [10].

Метод с использованием генеративно-сопоставительной искусственной нейронной сети Генеративно-сопоставительная сеть

Глубокое обучение или глубокие искусственные нейронные сети позволяют обрабатывать данные через несколько определенных уровней различных вычислительных функций, которые учатся создавать представление входных данных, используя обратное распространение ошибки для обновления весов в узлах, чтобы уменьшить разницу между истинными и прогнозируемыми значениями [11].

Генеративно-сопоставительная искусственная нейронная сеть представляют собой общую структуру для обучения генеративных моделей, то есть моделей, которые могут создавать выборки, а не только различать их. Генеративно-сопоставительная искусственная нейронная сеть состоит из генератора G и дискриминатора D , оба смоделированы как искусственные нейронные сети. Генератор оптимизирован для воспроизведения истинного распределения данных p_{data} , которое может быть привязано к интересующему распределению, путем генерации изображений (или любой формы данных), которые дискриминатору трудно отличить от реальных изображений, а именно — фактического распределения данных p_{data} . Одновременно перед дискриминатором ставится задача отличать реальные изображения от синтетических данных, сгенерированных G [12].

Их тренировочная процедура представляет собой минимаксную игру для двух игроков со следующей целевой функцией $L(D, G)$:

$$\min_G \max_D L(D, G) = M_{x \sim p_{data}(x)} [\ln D(x)] + M_{z \sim p_z(x)} [\ln (1 - D(G(z)))], \quad (3)$$

где:

— G — генератор;

— D — дискриминатор;

— z — вектор шума (набор случайных значений), отобранный из

предыдущего распределения шума по выбору p_z , обычно это равномерное или нормальное распределение;

— $M_{z \sim p_z(x)}$ — математическое ожидание распределения шума по выбору p_z ;

— x — реальное изображение из распределения данных p_{data} ;

— $M_{x \sim p_{data}(x)}$ — математическое ожидание распределения данных p_{data} ;

— $D(x)$ — функция распределения вероятности того, что x является реальным, а не сгенерированным изображением; — $G(z)$ — сгенерированные данные.

Существует также условная генеративно-сопоставительная искусственная нейронная сеть, в которой помимо изображений еще используется информация о классах изображений [13]. Подавая на вход генератора метку нужного класса, можно контролировать получаемые изображения. В случае процедурной генерации ландшафта, таким образом, можно контролировать генерацию нужного типа ландшафта [12].

Для условной генеративно-сопоставительной искусственной нейронной сети целевая функция будет иметь вид:

$$\min_G \max_D L(D, G) = M_{x \sim p_{data}(x)} [\ln D(x|c)] + M_{z \sim p_z(x)} [\ln (1 - D((G(z|c)|c)))], \quad (4)$$

где c — информация о классах изображений.



Обучение модели

Результатом модели является генерация изображений, имитирующих реальные спутниковые данные о рельефе планеты. Поэтому для обучения используются цифровые модели рельефа, схожие по формату с картами высот [12].

На вход генератора подается вектор шума. Единственный способ для него обучаться — это взаимодействие с дискриминатором. Дискриминатор же получает на вход либо созданные генератором данные, либо объект реальной обучающей выборки. Ошибка обучения дискриминатора рассчитывается на знании того, откуда пришли данные. В процессе обучения генератор обучается распределению исходной выборки и начинает создавать данные, все более близкие к реальным, в то время как дискриминатор становится все более точным в отличении подделки от оригинала. Генератор и дискриминатор обучаются одновременно, обновляя распределение дискриминатора, которое показывает вероятность принадлежности распределения генератора к распределению реальных данных. Обучение происходит до тех пор, пока дискриминатор может отличить реальные данные от сгенерированных [13].

Генерация ландшафта

Во время генерации ландшафта используется только генератор, поскольку дискриминатор уже выполнил свою роль в обучении генератора, и больше не нужен. Генератор создает некоторую карту высот на основе определенных параметров, которая схожа с реальным ландшафтом. Причем, создаваемые карты высот геологически точны и могут иметь признаки эрозии, так как генератор имитирует реальный ландшафт [11].

Для генерации бесконечного ландшафта можно использовать следующий подход: при генерации нового участка ландшафта использовать вектор шума, схожий с векторами уже сгенерированных соседних участков. Таким образом будет сгенерирован новый участок, схожий на границах с соседними. После чего интерполировать границы участков для избавления от возможных несостыковок между границами [12].

Методы на основе моделирования эрозии

Данные методы основаны на моделировании физических процессов, таких как гидравлическая эрозия и термоэрозия. Они не создают ландшафт с нуля, а итеративно модифицируют существующую карту высот, полученную, например, с помощью методов смещения средней точки или градиентного шума: эти методы не являются геологически точными, так как не моделируют физические процессы, происходящие с почвой в реальном мире, поэтому может потребоваться модификация карты высот для повышения качества ландшафта [14].

Гидравлическая эрозия

Гидравлическая эрозия возникает, когда движение воды по поверхности коренных пород приводит к механическому отслоению. Гидравлическая эрозия включает в себя ряд процессов механической эрозии, таких как истирание, коррозия и засоление. Существует два подхода для моделирования гидравлической эрозии, различающиеся по способу описания моделирования жидкости и по временным и пространственным масштабам — это подход Эйлера и подход Лагранжа [15].

Подход Эйлера основан на дискретной сетчатой модели, которая представляет входную карту высот, а также жидкость. Жидкость, присутствующая в ячейке, разрушает материал и откладывает материал в соседние ячейки, значение высоты которых меньше, основываясь на эрозионной силе жидкости. Эрозионная способность данного количества жидкости в ячейке зависит от ее объема и количества отложений, уже содержащихся в жидкости. Когда емкость жидкости для осаждения достигнута, материал осаждается на грунт [14].

Подход Лагранжа основан на частицах и моделирует эрозию путем моделирования потока частиц по рельефу с использованием сглаженной гидродинамики частиц. Эйлеровы методы получают решение относительно фиксированной сетки, в то время как методы Лагранжа определяют поток в терминах концентрации перемещаемых частиц [15].



Термоэрозия

Термоэрозия сочетает в себе термическое выветривание и массовое перемещение, представляющее собой движение горных пород и отложений вниз по склонам, в основном под действием силы тяжести. Это вызвано присутствием воды внутри трещин и небольшими проникновениями на границу материала. При изменении температуры различное тепловое расширение воды и материала приводит к тому, что материал ломается и падает. Термическое выветривание является одной из многих причин разрушения горных пород, крупномасштабных оползней и накопления гранулированного материала, все это приводит к кажущейся регулярности угла наклона многих склонов гор и холмов [15].

Транспортировка осуществляется под действием силы тяжести и основана на концепции, согласно которой осажденный гранулированный материал обладает внутренним трением, которое останавливает движение при достижении так называемого угла осыпи. Пусть θ обозначает угол покоя, также называемый углом осыпи, осажденного (гранулированного) материала, $\mathbf{p}$ — точка рельефа, $h = p_z$ — значение высоты точки, $s(\mathbf{p})$ — функция определения склона ландшафта в точке $\mathbf{p}$, $\rho(\mathbf{p})$ — функция определения сопротивления коренной породы в точке $\mathbf{p}$, то есть количество материала, которое должно смыть. Тогда уравнение имеет вид:

$$\frac{\partial h}{\partial t}(\mathbf{p}) = \rho(\mathbf{p})(\tan \theta - s(\mathbf{p})). \quad (5)$$

Термоэрозию можно смоделировать как процесс релаксации: на каждом временном шаге вычисляется наклон местности $s(\mathbf{p})$: если он ниже, чем $\tan \theta$, эрозии не происходит, в противном случае определенное количество материала, пропорциональное $\tan \theta - s(\mathbf{p})$, размывается и удаляется из слоя коренных пород и распределяется по соседним ячейкам. При работе со слоистой моделью количество удаленной коренной породы преобразуется в ил. Алгоритм останавливается, когда больше нет материала для перемещения [15].

Заключение

Был проведен обзор и описаны принципы работы существующих методов динамической процедурной генерации ландшафта.

Метод на основе смещения средней точки прост в реализации, но может порождать дефекты, такие как “проблема сгибания”, поэтому требует постобработки для их предотвращения. Данный метод подходит для генерации скалистой местности, например, горных хребтов, но не подходит для создания гладких ландшафтов.

Методы на основе градиентного шума позволяют создавать различные типы ландшафта в зависимости от значений параметров, а также не требуют адаптации для генерации бесконечного ландшафта, так как основаны на использовании непрерывных функций. На больших масштабах могут наблюдаться повторяемость и однообразие генерируемой местности. Симплексный шум обладает рядом преимуществ по сравнению с шумом Перлина, но является более сложным в реализации и используется только в тех случаях, где требуется высокая производительность.

Метод с использованием генеративно-состязательной искусственной нейронной сети позволяет создавать уникальные и разнообразные ландшафты с глубокой детализацией и геологической точностью, причем существует возможность контролировать тип генерируемой местности при использовании условной генеративно-состязательной искусственной нейронной сети. Данный метод требует набор цифровых моделей рельефа для обучения, то есть создает ландшафт не “с нуля”, причем качество цифровых моделей влияет на качество генерируемого ландшафта, а также обладает большими требованиями к вычислительным ресурсам.

Методы на основе моделирования эрозии являются методами постобработки, которые работают с готовой картой высот. Они направлены на повышение качества исходной карты высот, которая могла быть сгенерирована простым алгоритмом генерации. Методы рассчитаны на обработку карты высот большого масштаба (ширина и высота карты высот не менее 512 пикселей): в противном случае качество ландшафта только ухудшится.



Список литературы:

1. Jain Aryamaan, Sharma Avinash, Rajan. Adaptive & Multi-Resolution Procedural Infinite Terrain Generation with Diffusion Models and Perlin Noise // Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing. 2022. С. 1–9
2. Velzel Stef. Natural Path Generation: Patch-based terrain optimization for use in games: Master's thesis. 2020.
3. Merino Timothy, Charity M, Togelius Julian. Interactive Latent Variable Evolution for the Generation of Minecraft Structures // Proceedings of the 18th International Conference on the Foundations of Digital Games. 2023. С. 1–8.
4. Официальный сайт No Man's Sky [Электронный ресурс]. Режим доступа: <https://www.nomanssky.com/> (Дата обращения 11.03.2025).
5. Пажуков Кирилл Александрович, Веселов Илья Павлович, Зяблицева Ольга Витальевна. ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ 3D ЛАНДШАФТОВ // РАЗВИТИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ: ТЕОРЕТИЧЕСКИЕ И ПРАКТИЧЕСКИЕ АСПЕКТЫ. 2022. С. 54–57.
6. Снук Грег. Создание 3D-ландшафтов в реальном времени с использованием C++ и DirectX 9. 2006.
7. A Beginners Guide to Procedural Terrain Modelling Techniques / Kati Steven Emmanuel, Christian Mathuram, Akshay Rai Priyadarshi [и др.] // 2019 2nd International Conference on Signal Processing and Communication (ICSPPC) / IEEE. 2019. С. 212–217.
8. Sainio Niko. TERRAIN GENERATION ALGORITHMS. 2023.
9. The Book of Shaders by Patricio Gonzalez Vivo and Jen Lowe [Электронный ресурс]. Режим доступа: <https://thebookofshaders.com/11/?lan=ru> (Дата обращения 11.03.2025).
10. Классический шум Перлина (Perlin noise) и Симплекс шум (Simplex noise). Wonderville Engine Blog [Электронный ресурс]. Режим доступа: <http://blog.wonderville.ru/perlin-noise-simplex-noise/> (Дата обращения 11.03.2025).
11. Spick Ryan J, Cowling Peter, Walker James Alfred. Procedural generation using spatial GANs for region-specific learning of elevation data // 2019 IEEE Conference on Games (CoG) / IEEE. 2019. С. 1–8.
12. Panagiotou Emmanouil, Charou Eleni. Procedural 3D terrain generation using Generative Adversarial Networks // arXiv preprint arXiv:2010.06411. 2020.
13. Сухань Андрей Александрович. Генеративно-состязательные нейронные сети в задачах определения трендов // Московский экономический журнал. 2019. №6.
14. Nadig Sandeep Malatesh. EXAMPLE-BASED TERRAIN AUTHORING WITH COMPLEX FEATURES. Ph.D. thesis: Purdue University Graduate School. 2022.
15. A review of digital terrain modeling / Eric Galin, Eric Gu'erin, Adrien Peytavie [и др.] // Computer Graphics Forum / Wiley Online Library. Т. 38. 2019. С. 553–577.

