

And the large, multi-center sample allows us to stress-test the entire automated pipeline, from segmentation to simulation. By running the pipeline on data from different devices, pathologies, and voxel sizes, we can assess its robustness and ensure that the framework is not limited to a specific type of scan data.

Proposed in this study for reconstructing the computational model of the human lumbar spine is a sequentially executed, modular pipeline.

Data processing stage: Input the original CT image and preprocess the data, such as grayscale normalization, which can standardize the grayscale value of the CT image to ensure the consistency of data under different equipment and different scanning conditions.

1. Geometric model reconstruction: During the model reconstruction phase, an automated segmentation algorithm (such as a deep learning model) is used to separate each vertebra from the CT image. Surface mesh reconstruction is then performed, converting the segmented vertebral regions into a 3D surface mesh model (such as in STL format). Simultaneously, the generated mesh is smoothed to remove noise and irregularities, improving model quality. Finally, a 3D spinal model is obtained through segmentation.

2. Based on the generated geometric model, a biomechanical model is established, combined with the Gibbon toolbox. Using tools such as MATLAB, we developed an automated program to implement lumbar spine finite element meshing, load path and nucleus pulposus area delineation, and the setting of material properties and boundary conditions. We also designed a user interface for the automated lumbar spine model processing program, enabling users to easily load, process, and export lumbar spine model data.

This article details a human lumbar spine reconstruction process based on a public dataset. Parameterization is the core technical component of the reconstruction, while the public dataset serves as the starting point and quality assurance for the process. These components are closely interwoven with this core component, forming a complete, reliable, and auditable model reconstruction solution.

References

1. Kurachka K. S., Tsalka I. M. Vertebrae detection in X-ray images based on deep convolutional neural networks. 2017 IEEE 14th International Scientific Conference on Informatics. – IEEE, 2017, pp. 194–196.
2. Kurachka Kanstantsin, Ren Huanhai, Wang Xuemei .Comparative Analysis of Deep Learning Models for Lumbar Vertebrae Segmentation in MRI Images. 2025 Pattern Recognition And Information Processing (PRIP 2025), pp. 176–179.
3. Kanstantsin Kurachka, Xuemei Wang. Automation of primary diagnostics of diseases of the human lumbar spine using intelligent analysis of ct images // 2025 Open Semantic Technologies for Intelligent Systems (OSTIS-2025), pp. 355–360.

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ КОНТЕНТА

И. В. Морозько, И. И. Суторьма

*Гомельский государственный технический университет
имени П. О. Сухого, Республика Беларусь*

Рассмотрена процедурная генерация контента, алгоритмы процедурной генерации, преимущества их использования в разработке программного обеспечения, области их применения, существующие подходы и методы их оптимизации.

Ключевые слова: алгоритмы, процедурная генерация, разработка программного обеспечения, оптимизация.

PROCEDURAL CONTENT GENERATION

I. V. Marozka, I. I. Sutorma

Sukhoi State Technical University of Gomel, Republic of Belarus

This article discusses procedural content generation, procedural generation algorithms, the advantages of their use in software development, their areas of application, existing approaches and methods of their optimization.

Keywords: algorithms, procedural generation, software development, optimization.

Процедурная генерация контента (ПГК) – одно из наиболее активно развивающихся направлений в сфере информационных технологий. Под ПГК понимают автоматическое или полуавтоматическое создание и динамическое изменение элементов: объектов 2D/3D-графики, визуальных эффектов, звуков, музыки и других компонентов виртуальной среды. Современные технологии позволяют создавать огромные виртуальные пространства и сложные симуляции, однако ручное проектирование требует колоссальных затрат времени и ресурсов. Использование ПГК решает сразу несколько задач: ускоряет процесс разработки, снижает нагрузку на команду и обеспечивает разнообразие уникальных объектов, что особенно важно для симуляций и тренажеров [1].

Исследования в этой области сосредоточены на разработке алгоритмов, способных автоматически формировать ландшафты, объекты и иные структуры. Такие алгоритмы позволяют создавать большие и детализированные виртуальные пространства, где ручная работа была бы практически невозможна [2, с. 59].

Процедурная генерация используется в самых разных областях:

- игровая индустрия: создание уникальных миров, уровней и персонажей;
- симуляторы и тренажеры: генерация реалистичных ландшафтов и сценариев для обучения пилотов, водителей или военных специалистов;
- архитектура и дизайн: автоматическая генерация планировок зданий и интерьеров;
- музыка и искусство: создание уникальных композиций и визуальных эффектов.

ПГК становится универсальным инструментом, применимым в самых разных сферах, однако применение ПГК снижает степень контроля разработчика над результатом, что порождает дополнительную задачу – изучение влияния сгенерированного контента на восприятие пользователя [3, с. 195].

Одними из самых простых и широко используемых методов ПГК являются алгоритм случайной генерации. Он основан на генераторах псевдослучайных чисел, которые задают параметры объектов. Преимущества: простота реализации, высокая скорость работы, легкая интеграция с другими системами. Недостатки: риск получения неестественных или однообразных результатов, трудности с обеспечением качества. Пример использования такого алгоритма может служить генерация ландшафтов с помощью шума Перлина, позволяющего создавать реалистичные горы, долины и пещеры.

Грамматические системы используют формальные правила для построения структур. Преимущества: предсказуемость результата, контроль над структурой, гибкость изменения правил. Недостатки: сложность настройки и значительные вычислительные затраты. Применяются для генерации архитектуры зданий и интерьеров, что позволяет создавать реалистичные пространства, соответствующие определенным стилям.

Эволюционные алгоритмы имитируют процесс естественного отбора: создается популяция решений, применяются мутации и скрещивания, выбираются лучшие ва-

рианты. Преимущества: эффективность при оптимизации сложных систем, способность находить решения в многомерных пространствах, адаптивность. Недостатки: высокая вычислительная стоимость, необходимость точной настройки параметров, непредсказуемость первых результатов.

Машинное обучение. Нейронные сети обучаются на примерах существующего контента и создают новые объекты. Преимущества: высокая реалистичность, способность к самообучению, сходство с человеческим творчеством. Недостатки: потребность в больших объемах данных, сложность настройки и высокая нагрузка на вычислительные ресурсы. Этот метод активно используется в генерации изображений, музыки и даже сюжетов.

Для практического применения ПГК критически важна оптимизация. Цель оптимизации – повысить эффективность генерации контента и снизить нагрузку на систему.

Основные методы:

– *кэширование*. Сохранение ранее сгенерированных данных для повторного использования. Ускоряет загрузку и снижает нагрузку на процессор, но требует дополнительной памяти и усложняет обновление контента;

– *параллельная обработка*. Разделение задач между потоками или устройствами. Позволяет ускорить генерацию, но требует сложной синхронизации и усложняет отладку;

– *оптимизация памяти*. Использование компактных структур данных, методов сжатия и управления ресурсами. Повышает производительность, но усложняет код и может замедлить генерацию из-за дополнительных операций;

– *адаптивное масштабирование сложности*. Динамическая подстройка уровня детализации под ресурсы устройства. Обеспечивает стабильность работы на разных платформах, но требует постоянного мониторинга производительности и может снижать качество при высокой нагрузке.

Оптимизация должна учитываться уже на этапе проектирования архитектуры системы. Грамотный выбор структур данных и алгоритмов значительно упрощает дальнейшую разработку и снижает затраты на поддержку.

Процедурная генерация контента обладает огромным потенциалом для повышения эффективности и креативности в решении множества задач. Современные алгоритмы, включая машинное обучение, позволяют создавать контент, сопоставимый с традиционным. Ключевыми задачами остаются оптимизация производительности, разработка гибких систем параметризации и методов оценки качества. Комбинация программирования, ИИ и визуализации открывает возможности для создания сложных генераторов, применимых в играх, симуляциях и других областях, где требуется масштабное виртуальное окружение.

Л и т е р а т у р а

1. Процедурная генерация / Wikipedia. – URL: https://ru.wikipedia.org/wiki/Процедурная_генерация (дата обращения: 11.10.2025).
2. Шорт, Т. Процедурная генерация в гейм-дизайне / пер. с англ. М.С. Рыжиковой ; под ред. Т. Шорт и Т. Адамса. – М. : ДМК Пресс, 2020. – 344 с.
3. Shaker, N. Procedural Content Generation in Games (Computational Synthesis and Creative Systems) / N. Shaker, J. Togelius, M. Nelson. – С. : Springer, 2016. – 253 p.