UDC 004.925.84

R.O. Savelko
O.N. Romanyuk

APPLICATION OF CELLULAR AUTOMATA FOR PROCEDURAL TEXTURE GENERATION

Vinnytsia National Technical University

Анотація

У тезах розглядається застосування клітинних автоматів для процедурної генерації текстур у комп'ютерній графіці. Описано теоретичні основи клітинних автоматів, включаючи правила переходу станів та їх класифікацію. Проаналізовано методи використання одновимірних і двовимірних клітинних автоматів, зокрема Гра «Життя» Конвея, для синтезу природних текстур – камені, деревина, вода, рослинність. Розглянуто переваги процедурного підходу: масштабованість, низькі вимоги до пам'яті та детермінований характер генерації. Наведено основне рівняння клітинного автомата та обговорено практичне впровадження методів у системах реального часу.

Ключові слова: клітинні автомати, процедурні текстури, комп'ютерна графіка, генерація текстур, правила переходу, синтез зображень, Гра «Життя».

Abstract

The paper examines the application of cellular automata (CA) for procedural texture generation in computer graphics. The theoretical foundations of cellular automata are described, including state transition rules and their classification. Methods of using one-dimensional and two-dimensional cellular automata, particularly Conway's Game of Life, for synthesising natural-looking textures such as stone, wood, water, and vegetation are analysed. The advantages of the procedural approach – scalability, low memory requirements, and deterministic generation – are discussed. The fundamental equation of a cellular automaton is presented, and practical implementation aspects in real-time rendering systems are considered.

Keywords: cellular automata, procedural textures, computer graphics, texture generation, transition rules, image synthesis, Game of Life.

Introduction

Procedural texture generation is a widely used technique in computer graphics that enables the creation of complex visual patterns through algorithmic means, without relying on large sets of pre-stored image data. Unlike bitmap textures, procedural textures are computed on the fly, offering advantages in terms of memory efficiency, resolution independence, and parametric control [1]. The relevance of developing robust procedural methods continues to grow alongside the expanding demands of real-time rendering in games, simulations, and visual effects pipelines.

Among the various algorithmic approaches to procedural generation, cellular automata (CA) occupy a special place due to their ability to produce visually complex, self-organised patterns from simple local rules. The study of CA for texture synthesis represents a promising direction that merges the fields of computational mathematics and applied computer graphics [2].

Cellular Automata: Theoretical Background

A cellular automaton is a discrete computational model consisting of a regular grid of cells, each of which exists in one of a finite number of states. The system evolves in discrete time steps according to a deterministic transition rule that maps the current state of a cell and its neighbourhood to the cell's next state [3]. For a one-dimensional CA with state set S and neighbourhood radius r, the general transition function can be written as:

$$s(i, t+1) = f(s(i-r, t), \ldots, s(i, t), \ldots, s(i+r, t))$$

where $s(i, t)$ denotes the state of cell i at time step t, r is the neighbourhood radius, and f is the local transition function. In two-dimensional grids, the neighbourhood is typically defined as the Moore neighbourhood (8 adjacent cells) or the von Neumann neighbourhood (4 orthogonal neighbours) [3].

CA are classified by the complexity of the patterns they generate. Wolfram's classification identifies four classes ranging from uniform convergence to complex, seemingly aperiodic behaviour [2]. Classes III and IV are of particular interest for texture synthesis, as they produce rich, irregular patterns reminiscent of natural materials.

Application to Procedural Texture Generation

Two-dimensional CA are particularly suitable for texture generation. Conway's Game of Life, operating on a binary grid with rules based on the count of live neighbours, produces island-like clusters that closely resemble rock surfaces, cellular biological structures, or cave maps [4]. By varying the birth and survival rule sets – collectively known as B/S notation – a practitioner can tune the generated pattern towards different target materials.

For organic and natural textures such as wood grain or water ripples, reaction-diffusion CA variants are employed. These models augment the standard CA transition with continuous-valued states and diffusion terms, yielding Turing-like patterns [1]. The resulting textures exhibit the characteristic stripe and spot motifs found in animal skin and geological formations.

Vegetation and terrain textures benefit from multi-scale CA, where separate automata operating at different spatial frequencies are composited. A coarse-scale CA defines the macro-structure (forest vs. meadow), while finer-scale automata add micro-detail (individual leaves or grass blades). This hierarchical approach aligns with the multi-resolution requirements of level-of-detail rendering systems [4].

From an implementation perspective, CA are well-suited to parallel execution on modern GPUs. Each cell update is independent of non-neighbouring cells, allowing the transition function to be mapped directly onto a fragment shader or a compute shader dispatch, with the CA grid stored as a texture buffer. Benchmarks reported in the literature indicate that GPU-based CA can sustain real-time update rates for grids up to 4096 × 4096 cells [2; 4].

Conclusions

Cellular automata provide an effective and computationally efficient framework for procedural texture generation. Their locality of interaction, deterministic behaviour, and inherent parallelism make them well-matched to GPU architectures used in contemporary rendering pipelines. The diversity of patterns achievable through rule variation – from crystalline and rocky surfaces to organic and fluid textures – positions CA as a versatile tool within the broader procedural generation toolkit. Further research directions include the integration of CA with noise-based methods and the exploration of probabilistic transition rules for stochastic texture synthesis.

REFERENCES

1. Lagae A., Lefebvre S., Cook R., DeRose T., Drettakis G., Ebert D., Lewis J.-P., Perlin K., Zwicker M. A Survey of Procedural Noise Functions. Computer Graphics Forum. 2010. Vol. 29, No. 8. P. 2579–2600.
2. Wolfram S. A New Kind of Science. Wolfram Media, 2002. 1197 p.
3. Ilachinski A. Cellular Automata: A Discrete Universe. World Scientific, 2001. 808 p.
4. Johnson L., Yannakakis G. N., Togelius J. Cellular Automata for Real-time Generation of Infinite Cave Levels. Proceedings of the 2010 Workshop on Procedural Content Generation in Games. 2010. P. 1–4.

Савелко Ростислав Олегович – студент групи 1ПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e–mail: rostiksavelko@gmail.com.

Романюк Олександр Никифорович – доктор технічних наук, професор, професор кафедри програмного забезпечення, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e–mail: rom8591@gmail.com.

Savelko Rostyslav O. – student of group 1PI-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: rostiksavelko@gmail.com.

Romanyuk Oleksandr N. – Doctor of Technical Sciences, Professor, Professor of the Software Engineering Department, Head of the Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, e–mail: rom8591@gmail.com.