

**Korneliusz WARSZAWSKI, Tomasz ZAWADZKI**

UNIwersytet Zielonogórski, Instytut Sterowania i Systemów Informatycznych

## Wstęp do komputerowego generowania koryta rzeki z wykorzystaniem systemu cząstek na potrzeby modelowania wirtualnego środowiska

Mgr inż. Korneliusz WARSZAWSKI

Autor jest studentem studiów doktoranckich na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. W swojej pracy badawczej zajmuje się wirtualną rzeczywistością, metodami odwzorowania ukształtowania terenu oraz modelowaniem obiektów z wykorzystaniem systemu cząstek. Zawodowo pracuje jako Administrator Systemów Informatycznych Sądu Rejonowego w Nowej Soli.



e-mail: K.Warszawski@weit.zgora.pl

Mgr inż. Tomasz ZAWADZKI

Autor jest studentem studiów doktoranckich na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. W swojej pracy badawczej zajmuje się wirtualną rzeczywistością oraz grafiką trójwymiarową w wizualizacji, symulacji oraz rekonstrukcji budynków oraz obszarów miejskich.



e-mail: T.Zawadzki@weit.zgora.pl

### Streszczenie

Artykuł przybliży idee zastosowania systemu cząstek do automatycznego deformowania struktur terenowych. Za pomocą proponowanej metody można od podstaw ukształtować krajobraz jak i można jej użyć do deformacji wstępnie predefiniowanego ukształtowania terenu i wzbogacenia jego powierzchni o struktury przypominające formacje powstałe na skutek długotrwałego oddziaływania erozji hydrologicznej, tj. kanion czy koryto rzeki. Ze względu na krótki czas pracy algorytmu, który zbliżony jest do rzeczywistego, metoda pozwala na uzyskanie w rozsądnym czasie wybranych struktur terenowych również na typowych komputerach osobistych.

**Słowa kluczowe:** Wirtualna Rzeczywistość, System Cząstek, Modelowanie terenu, Generowanie koryta rzeki.

### Introduction to computer generation of a riverbed with use of the particle system for modelling the virtual environment

#### Abstract

The paper deals with a method that applies the particle system to modify height-field based landscapes with generation of hydrological erosion like canyons or riverbeds. The authors present a construction of the particle system, its main attributes and its influence on the final product of the modelling process. Performance of the particle-based algorithm is close to real-time, thus it offers the opportunity for fast terrain surface modifications, which enable selected landscapes to be rendered in real-time on typical desktop computers [6, 7]. The main data structure of the modelled terrain is typical height-field, based on a two dimensional array. Each cell represents the altitude at a coordination point defined by rows and columns of this array. This kind of landscape data structure enables fast and simple implementation of the terrain deformation and can be rendered in any virtual reality systems. In the paper there are discussed the basic problems of landscape hydrological erosion deformations and there is proposed a fast alternative to the algorithms based on Computational Fluids Dynamic (CFD) or rainfall erosion [9, 11]. The proposed method is not intended to simulate physically the erosion process but to focus on its results for exploitation in virtual environments in on-line visualisations.

**Keywords:** virtual reality, particle system, landscape modelling, riverbed generation.

## 1. Wstęp

Począwszy od zastosowań w symulatorach wojskowych, poprzez medycynę, systemy edukacyjne, a skończywszy na przemyśle rozrywkowym Wirtualna Rzeczywistość znajduje swoje zastosowanie w coraz szerszym zakresie życia człowieka. Jednym z elementów modelowania środowiska na potrzeby takich systemów są metody generowania powierzchni ukształtowania terenu. Obecnie najczęściej wykorzystuje się do tego celu algorytmy modelowania fraktalnego, rzadziej metody stricte iteracyjne [1, 2, 3, 4].

Zastosowanie systemu cząstek w modelowaniu ukształtowania powierzchni krajobrazu stanowi alternatywę dla obecnie stosowanych

do tego celu metod. Ponadto technika ta umożliwia zarówno kompleksowe generowanie powierzchni terenu jak również pozwala na modyfikowanie predefiniowanej struktury i wzbogacanie jej o kaniony lub koryta rzek znacznie poprawiając w ten sposób realizm komputerowo generowanych krajobrazów. Ponadto czas pracy algorytmu jest wystarczająco krótki, aby mógł być prezentowany przez wielo-ekranowe systemy rzeczywistości wirtualnej [5, 6, 7, 8].

Do symulowania struktur tworzonych przez przepływający strumień cieczy najczęściej stosuje się metody oparte na dynamice płynów, jak metoda CFD (ang. *Computational Fluid Dynamics*) symulująca naturalny ruch wody na podstawie równań fizycznych oraz jej wpływ na deformowaną powierzchnię. Technika Fostera i Fedkiwa znacznie poprawia realizm wirtualnego krajobrazu, jednakże wymaga bardzo dużych mocy obliczeniowych [9, 10].

Nagashima zaproponował odmienne podejście do generowania takich struktur. Punktem startowym jego metody jest fraktalny algorytm przemieszczenia punktu środkowego. Za jego pomocą wyznaczona jest krawędź wzdłuż której postępuje hydrologiczna erozja terenu. W kolejnych krokach metoda poszerza powstający uskoki w krajobrazie. Za jej pomocą można zamodelować struktury geologiczne powstałe w wyniku długotrwałego oddziaływania erozji spowodowanej opadami deszczu czy poddawane oddziaływaniu termicznemu oraz na skutek trzęsień ziemi powstałych przez ruchy płyt tektonicznych [11, 12].

## 2. Budowa systemu cząstek

Typowym zastosowaniem systemów cząstek jest modelowanie w czasie rzeczywistym, obiektów o dynamicznie zmieniającym się i często trudnym do jednoznacznie zdefiniowania kształcie, tj.: strumieniem cieczy, eksplozje, dym czy ogień. Technika ta jest również wykorzystywana do modelowania roślinności w szczególności traw i zbóż [13, 14, 15].

Tak jak typowy system cząstek, również stosowany do modelowania terenu składa się z dwóch podstawowych elementów. Pierwszym jest kolekcja cząstek, posiadających określone parametry symulujące własności fizyczne, tj.: prędkość, lepkość, rozmiar, masa czy pozycja w przestrzeni. Wybór odpowiednich parametrów oraz dobór ich wartości ma duży wpływ na charakterystykę generowanej powierzchni krajobrazu. Drugim elementem jest zbiór emiterów, których zadaniem jest wprowadzanie nowych cząstek z kolekcji do systemu. Również i w ich przypadku określa się podstawowe atrybuty tj.: pozycja w przestrzeni oraz możliwość jej zmiany (zazwyczaj stosuje się nieruchome emitory) czy rozmiar okna emisji, określający granice przestrzeni wewnątrz której nowe cząstki mogą być wprowadzane do systemu. Ponieważ w tego typu systemie nie określa się czasu po jakim dana cząstka ulega zniszczeniu, konieczne staje się określenie dodatkowych warunków eliminacji cząstek. Podstawowym zdarzeniem uruchamiającym procedurę usuwania cząstki z systemu jest jej

kolizja z powierzchnią modelowanego terenu. Dodatkowo, aby zapobiec przemieszczaniu się cząstek w nieskończoność, określa się dozwoloną przestrzeń pracy algorytmu. Każda cząstka przekraczająca w ten sposób wyznaczoną granicę jest automatycznie eliminowana [5, 6, 7, 13, 14].

### 3. Przygotowanie mapy wysokościowej

Najprostszą strukturę danych, przechowującą w pamięci komputera ukształtowanie terenu jest mapa wysokościowa. Implementowana jest ona jako macierz dwuwymiarowa, składająca się z wartości punktów wysokościowych o współrzędnych określonych przez jej kolejne wiersze i kolumny. Bazową mapę wysokościową, która będzie poddawana modyfikacją, można uzyskać na podstawie danych DEM (ang. *Digital Elevation Map*) lub GIS (ang. *Geographic Information System*) rzeczywistych struktur geologicznych, bądź stosując jedną lub kilka metod automatycznego generowania terenu. Zastosowanie bazującego na samo podobieństwie algorytmu diamentowo-kwadratowego Millera, metody generowania uskoku Mandelbrota, systemów funkcji iterowanych (ang. *Iterated Function Systems*) czy wykorzystanie techniki cząsteczkowej umożliwi, w stosunkowo krótkim czasie uzyskanie odpowiedniej powierzchni krajobrazu [5, 6, 7, 16, 17, 18, 19].

### 4. Przygotowanie mapy twardości

Podobnie jak w przypadku mapy wysokościowej również parametry określające podatność danego obszaru na erozję można uzyskać na podstawie danych geologicznych. Alternatywnie, na potrzeby wirtualnego środowiska odpowiednie współczynniki twardości można zasymulować stosując zmodyfikowany algorytm uskoku Vossa. Podobnie jak w przypadku klasycznej metody powierzchnia mapy jest dzielona poprzez kolejno generowane proste, a wartości powstałych w ten sposób sfer są modyfikowane wg zależności:

$$h = \begin{cases} h + \varepsilon & \Leftrightarrow \Delta > 0 \\ h - \varepsilon & \Leftrightarrow \Delta \leq 0 \end{cases} \quad (1)$$

gdzie:

$$\Delta = (x_2 - x_1) * (y_i - y_1) - (y_2 - y_1) * (x_i - x_1), \quad (2)$$

$h$  – współczynnik twardości,  $\varepsilon$  – modyfikator wartości,  $\Delta$  – odległość węzła mapy wysokościowej od prostej,  $(x_1, y_1)$ ,  $(x_2, y_2)$  – punkty wyznaczające prostą,  $(x_i, y_i)$  – współrzędne węzła mapy wysokościowej.

Następnie otrzymane na tej podstawie współczynniki twardości poszczególnych węzłów na mapie wysokościowej powinny być poddane normalizacji na zakres [0, 1].

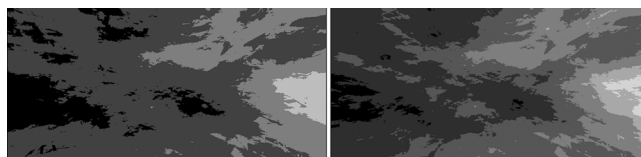
$$H = \frac{h - \min}{\max - \min}, \quad (3)$$

gdzie:  $H$  – znormalizowana wartość współczynnika twardości,  $h$  – bazowa wartość współczynnika twardości,  $\min$  – minimalna wartość na mapie twardości,  $\max$  – maksymalna wartość na mapie twardości.

Ostatnią czynnością jest określenie klas twardości i przypisanie do nich poszczególnych węzłów. Klasy twardości odpowiadają ogólnej liczbie minerałów występujących na obszarze modelowanego ukształtowania terenu.

$$C = \lfloor H * k \rfloor, \quad (3)$$

gdzie:  $C$  – wyznaczona klasa twardości,  $H$  – znormalizowana wartość na mapie twardości,  $k$  – liczba klas twardości.

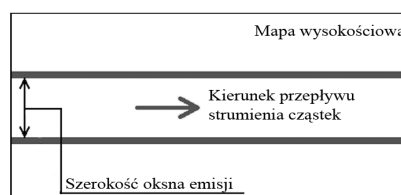


Rys. 1. Przykład mapy twardości (lewy: dla czterech klas, prawy: dla sześciu klas)  
Fig. 1. Example of hardness map (left: for four classes, right: for six classes)

### 5. Generowanie koryta rzeki

Podobieństwo ruchu cząstek w przestrzeni do strumienia cieczy pozwala na użycie tej metody jako algorytmu deformującego powierzchnię wirtualnego terenu w taki sposób, aby wygenerować struktury powstające pod wpływem erozji hydrologicznej. Wynikowy krajobraz jak i charakterystyka tych deformacji zależą wyłącznie od przyjętych parametrów systemu.

Odzwiedcieniem mapy wysokościowej w wirtualnym środowisku jest siatka wielokątów. Umieszczenie emitera (bądź grupy emitatorów) po jednej z jej stron, wyznacza punkt początkowy generowanego koryta rzeki, a jego szerokość wyznaczona jest przez rozmiar okna emisji emitera.



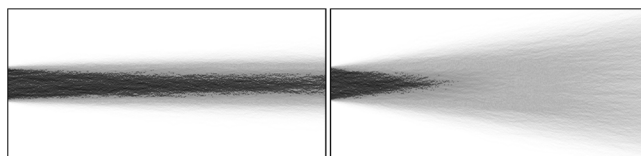
Rys. 2. Punkt początkowy generowanego kanionu  
Fig. 2. Initial point of the generated canyon

Ponieważ za pomocą mapy wysokościowej nie można przedstawić zaawansowanych struktur geologicznych, tj. klify, półki skalne czy jaskinie, to równanie wyznaczania trajektorii cząstki w przestrzeni można z rzutować do równania na płaszczyźnie. Zabieg ten znacznie skraca czas pracy algorytmu.

$$\begin{aligned} x_i &= x_{i-1} + v * \cos(\alpha), \\ y_i &= y_{i-1} - v * \sin(\alpha) \end{aligned} \quad (4)$$

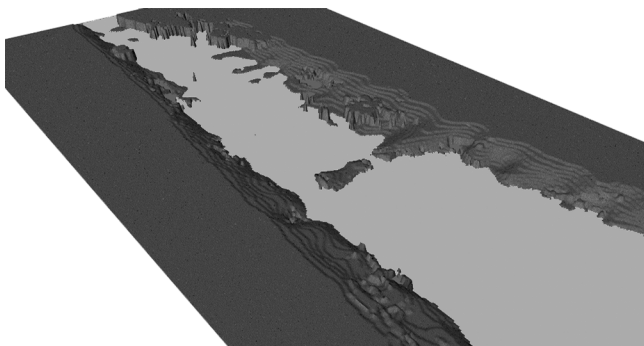
gdzie:  $(x_i, y_i)$  – nowa pozycja cząstki,  $(x_{i-1}, y_{i-1})$  – poprzednia pozycja cząstki,  $v$  – prędkość cząstki,  $\alpha$  – kąt kierunkowy cząstki.

Emitowany strumień cząstek w przypadku kolizji z powierzchnią terenu powoduje gradację wartości wysokości w komórce mapy wysokościowej wyznaczonej przez współrzędne miejsca kolizji. W przypadku, gdy cząstki mają określony atrybut rozmiaru to poza samym punktem kolizji, gradacji podlega również jego otoczenie w promieniu określonym przez tą właściwość. Im większa wartość tego parametru tym większy obszar mapy wysokościowej jest jednorazowo poddawany modyfikacją, co z kolei wpływa na chropowatość brzegów generowanego koryta rzeki w odwrotnie proporcjonalnej zależności.



Rys. 3. Wpływ modyfikatora kąta kierunkowego cząstki na chropowatość krawędzi kanionu (lewy: modyfikacja do  $\pm 1^\circ$ , prawy: modyfikacja do  $\pm 5^\circ$ )  
Fig. 3. Influence of the particle directional angle modifier on the canyon edge roughness (left: modification up to  $\pm 1^\circ$ , right: modification up to  $\pm 5^\circ$ )

Dodatkowo na chropowatość brzegów modelowanej struktury można wpłynąć poprzez dodanie losowego modyfikatora kąta kierunkowego cząstki. Współczynnik ten może być dodawany przy obliczaniu kolejnej pozycji cząstki. Z przeprowadzonych symulacji wynika, że określenie tej wartości powyżej  $\pm 2^\circ$  powoduje nadmierną chropowatość krawędzi, co obniża realizm generowanej struktury. Wprowadzenie dodatkowych modyfikatorów wyznaczania kolejnej pozycji cząstki w przestrzeni umożliwia generowanie załamów strumienia cząstek, co z kolei wpływa na zmianę kierunku koryta rzeki. Podstawowym modyfikatorem jest uzależnienie kąta kierunkowego cząstki od otaczających ją w danej chwili wartości na mapie wysokościowej. Zabieg ten umożliwia ukierunkowanie strumienia cząstek według spadku wysokości danej powierzchni terenu.



Rys. 4. Przykład wygenerowanego koryta rzeki  
Fig. 4. Example of the generated riverbed

## 6. Złożoność obliczeniowa

Tak jak w przypadku każdego złożonego algorytmu również w tym przypadku nie można jednoznacznie określić złożoności obliczeniowej metody. Jednakże określenie operacji dominujących pozwala na oszacowanie tej zależności. Dla każdego systemu cząstek czynnością dominującą jest przemieszczenie obiektów w wirtualnej przestrzeni. Z sumowanie operacji porównania i przypisania towarzyszącym zmianą pozycji cząstek pozwala na przybliżenie złożoności obliczeniowej przedstawianej metody. Przy założeniu, że ruch cząstek odbywa się na płaszczyźnie, algorytm można przedstawić za pomocą implementacji w C# i na tej podstawie oszacować jego złożoność.

<pre>for (int i = 0;     i &lt; n;     i++) {     part[i].pos.x += cos(alpha)     part[i].pos.y -= sin(alpha) }</pre>	<table border="0"> <tr><td style="border-left: 1px solid black; padding-left: 5px;">1</td><td>operacja</td></tr> <tr><td style="border-left: 1px solid black; padding-left: 5px;">n</td><td>operacji</td></tr> <tr><td style="border-left: 1px solid black; padding-left: 5px;">n-1</td><td>operacji</td></tr> <tr><td style="border-left: 1px solid black; padding-left: 5px;">2(n-1)</td><td>operacji</td></tr> <tr><td style="border-left: 1px solid black; padding-left: 5px;">2(n-1)</td><td>operacji</td></tr> </table>	1	operacja	n	operacji	n-1	operacji	2(n-1)	operacji	2(n-1)	operacji
1	operacja										
n	operacji										
n-1	operacji										
2(n-1)	operacji										
2(n-1)	operacji										

Oszacowana na podstawie operacji dominujących złożoność obliczeniowa metody kształtuje się według poniższej zależności:

$$F(n) = 6n + 4, \quad (5)$$

gdzie:  $n$  – liczba cząstek w kolekcji.

## 7. Wnioski

Przemieszczanie się strumienia cząstek w wirtualnym środowisku przypomina w swoim zachowaniu przepływ strumienia wody. Odpowiednie zasymulowanie oddziaływania takiego strumienia na wirtualny krajobraz umożliwia zastosowanie systemu cząstek do modelowania struktur terenowych poddanych długotrwałemu oddziaływaniu erozji hydrologicznej.

Podatność algorytmu na parametryzację ułatwia dostosowanie wyników działania metody dożądanego poziomu modelowanego krajobrazu. Dobór odpowiednich parametrów systemu pozwala na symulowanie wpływu płynów o różnej dynamice na wejściowe ukształtowanie powierzchni terenu, umożliwiając zastosowanie metody do wyznaczania tras przepływu fali powodziowych lub optymalnego uregulowania koryt rzek.

Czas pracy proponowanej metody w głównej mierze zależy wprost proporcjonalnie od liczby cząstek modelujących daną strukturę powierzchni. Taka właściwość algorytmu umożliwia zastosowanie go w symulacjach czasu rzeczywistego i otrzymanie żądanej konfiguracji ukształtowania terenu w stosunkowo krótkim czasie.

## 8. Literatura

- [1] R.M. Satava: Virtual reality and telepresence for military medicine. *Computers in Biology and Medicine*, tom 25, wyd. 2, 1995.
- [2] G. Ferrington, K. Loge: Virtual Reality: A new learning environment. *Computing Teacher*, tom 19, nr 7, 1992.
- [3] J. Rickel, W.L. Johnson: Virtual humans for team training in Virtual Reality. In the proceedings of the 9th World Conference on AI in Education, 1999.
- [4] C.J. Bonk, V.P. Dennen: Massive multiplayer online gaming: a research framework for military training and education. Technical Report for Office of the Under Secretary of Defense for Personnel and Readiness, 2005.
- [5] K. Warszawski: Cząsteczkowe generowanie ukształtowania terenu, *Software Developer Journal*, 2007.
- [6] K. Warszawski, T. Zawadzki: Building a Particle System for Virtual Landscape Generation. In 1st International Interdisciplinary Technical Conference of Young Scientists – Intertech, 2008.
- [7] K. Warszawski, S. Nikiel, T. Zawadzki: Particle system for generation of terrain structures. *Przegląd Telekomunikacyjny i Wiadomości Telekomunikacyjne*, nr 6, 2008.
- [8] L. Teo, J. Byrne, D. Ngo: A method for determining the properties of multi-screen interfaces. *International Journal of Applied Mathematics and Computer Science*, tom 10, nr 2, 2000.
- [9] N. Foster, R. Fedkiw: Practical animation of liquids. In proceedings of Siggraph '01, 2001.
- [10] D.J. Chae, J.H. Park: An active model of water movement by activity-based method for simulation of the virtual environment. *International Journal of Information Technology*, 2006.
- [11] K. Nagashima: Computer generation of eroded valley and mountain terrains. *The Visual Computer*, nr 13, 1997.
- [12] S.T. Teoh: River and coastal action in automatic terrain generation. In proceedings of the International Conference on Computer Graphics and Virtual Reality '08, 2008.
- [13] W.T. Reeves: Particle Systems – A technique for modelling a class of fuzzy objects. *Computer Graphics*, tom 17, nr 3, 1983.
- [14] J. Lander: The ocean spray in your face. *Game Developer*, nr 6, 1998.
- [15] J. Van Der Burg: Building an advanced particle system. *Game Developer*, nr 3, 2000.
- [16] F.K. Musgrave, C.E. Kolb, R.S. Mace: The synthesis and rendering of eroded fractal terrains. *Computer Graphics*, tom 23, nr 3, 1989.
- [17] E. Koh, D.D. Hearn: Fast generation and surface structuring methods for terrain and other natural phenomena. tom 11, nr 3, *Eurographics*, 1992.
- [18] J. Olsen: Realtime procedural terrain generation. Department of Mathematics and Computer Science University of Southern Denmark, 2004.
- [19] S. Nikiel, G.H. Kirby: Iterated function systems for terrain synthesis. *Machine Graphics and Vision*, tom 9, nr 1/2, 2000.