

МЕТОД ПОБУДОВИ ВИПАДКОВИХ ПЕРЕХІДНИХ ШАРІВ ТА ЙОГО ЗАСТОСУВАННЯ У МУЛЬТИМАСШТАБНОМУ МОДЕЛЮВАННІ СТРУКТУР ГЕТЕРОГЕННИХ СЕРЕДОВИЩ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ OpenCL

© Яворський Н., Фармага І., Марікуца У., 2015

На основі методів побудови випадкових скалярних полів та методів побудови градуйованих перехідних шарів запропоновано метод побудови випадкових мікрорівневих перехідних шарів, що дає змогу моделювати, рекурентно у різних масштабах, структури гетерогенних середовищ, як комбінацію детермінованих елементів із заданими стохастичними перехідними шарами між ними. Описано реалізацію методу за допомогою технології високопродуктивних паралельних та розподілених обчислень OpenCL.

Ключові слова: гетерогенне середовище, перехідний шар, випадкове поле, гауссовий фільтр, мультимасштабне моделювання, OpenCL.

Basing on the random field construction methods and graded transition layers construction methods, the random micro level transition layer construction method is proposed. It gives the ability to do recurrent different scaled modelling of heterogeneous environment structures as the combination of deterministic elements with stochastic transition layers between them. Also described the method realization by technology of high performance parallel and distributed computing – OpenCL.

Key words: heterogeneous environment, transition layer, random field, Gaussian filter, multiscale modelling, OpenCL.

Вступ

Моделювання перехідних шарів є актуальним завданням, що часто виникає під час аналізу чи проектування фізичних середовищ, які складаються з двох чи більше компонент. Прикладом таких середовищ є композиційні матеріали. Під час побудови моделей структури гетерогенних середовищ, залежно від вибраного рівня деталізації [1, 2], перехідні шари можуть описуватися як: окремі одиниці (наприклад, полідисперсна модель Хашина [3, 4] чи модель ефективного інтерфейсу [5], [6]); спрощені елементи регулярних структурних одиниць (наприклад, модель елементарних комірок зі звуженнями [7]); та як випадковий розподіл з деякою концентрацією різних матеріалів елементів мікрорівневої моделі (наприклад, модель функціонально градуйованих матеріалів FGM [8, 9]). Моделі, що входять до останнього класу, є найперспективнішими у плані досліджень, автоматизації та подальшого практичного використання, оскільки дають змогу максимально адекватно описувати реальні фізичні та просторові структури гетерогенного середовища.

Випадкові мікрорівневі моделі, своєю чергою, поділяються на коміркові та моделі випадкових скалярних полів [10], яких стосується ця робота. Порівняно з комірковими, останні забезпечують значну гнучкість у формалізації класів реальних фізичних мікроструктур, що можуть бути ними змодельовані. Найбільше використовуються вони у моделюванні мікроемульсій, ґрунтових порід, аморфних сплавів та аерогелів. Проте, очевидно, що цим їх функціональні можливості не обмежуються.

З іншого боку, використання примітивніших, порівняно з мікрорівневими, структурних моделей гетерогенних середовищ значно спрощує обчислення та зменшує їх кількість, особливо у задачах проектування. Основний недолік моделей, що входять до цього класу, – використання методів усереднення, які ґрунтуються на низці аксіоматичних припущень, яких не завжди вдається дотримуватися на практиці (наприклад, у структурних моделях дуже важко, або й неможливо змоделювати явище перколяції), тому їх використання сильно обмежене.

У цій роботі наводиться варіант об'єднання основних переваг описаних класів моделей. Для цього пропонується метод побудови випадкових мікрорівневих перехідних шарів за різних масштабів моделювання. Це дає змогу будувати моделі структур гетерогенних середовищ як комбінацію детермінованих елементів із заданими стохастичними перехідними шарами між ними. Крім того, метод можна застосовувати рекурентно для безпосереднього мультимасштабного моделювання випадкових скалярних полів.

Враховуючи те, що моделювання випадкових скалярних полів потребує великої кількості однакових обчислень, його ефективно реалізовувати на SIMD (Single Instruction Multiple Data) архітектурах обчислювальних пристроїв, наприклад, на графічних картах (GPU). У цій роботі описано таку програмну реалізацію на основі технології OpenCL (Open Computing Language). Отже, можна значно збільшити швидкодію, за рахунок одночасного виконання програмного коду на великій кількості обчислювальних ядер.

Моделювання випадкових скалярних полів

Задача моделювання випадкових скалярних полів добре описана у великому колі робіт, зокрема в [10] та [11]. Вона складається таких основних етапів:

1. Побудова базового випадкового розподілу у деякому елементарному об'ємі (RVE) гетерогенного середовища, що переважно являє собою тривимірну матрицю, кожен елемент якої є випадковою скалярною величиною (інтенсивністю) $f(x, y, z)$ у діапазоні $[0;1]$.
2. Огрублення базового випадкового розподілу $f(x, y, z)$ деяким, переважно лінійним, фільтром K з обчисленням інтенсивності як середнього значення сусідніх інтенсивностей, зважених деяким коефіцієнтом:

$$F(x, y, z) = \iiint_{\Omega} f(x, y, z) K(\Omega) d\Omega \quad (1)$$

3. Застосування перерізу отриманого огрубленого поля $F(x, y, z)$ на деякому рівні F_0 вибраного діапазону або використання одночасно кількох перерізів:

$$I(x, y, z) = H(F(x, y, z) - F_0(x, y, z)), \quad H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (2)$$

Дотримуючись максимальної простоти, як генератор випадкових чисел використовують той, що поставляється з прикладною алгоритмічною мовою програмування, на якій реалізовується програма моделювання. Зазвичай використовується рівномірний (рис. 2, а), рідше гауссовий розподіли інтенсивностей.

Для огрублення побудованого поля використовується лінійний фільтр деякого, наперед визначеного розміру. Що більший розмір фільтра, то гладші поверхні отримують у кінцевому результаті. Дуже часто на цьому етапі використовується добре відомий гауссовий фільтр:

$$K(x, y, z, a, b, c, r) = \exp \left(- \frac{\left(\frac{x}{a} \right)^2 + \left(\frac{y}{b} \right)^2 + \left(\frac{z}{c} \right)^2}{\left(\frac{r}{2} \right)^2} \right) \quad (3)$$

де x, y, z – координати відносно центра фільтра; a, b, c – діаметрально протилежні точки осей еліпсоїда; r – радіус еліпсоїда. Описаний фільтр враховує інтенсивності сусідніх точок експоненційно, від одиниці в центрі його координат, і приблизно рівно нулю на еліпсоїді з заданим радіусом. Двовимірні варіанти таких еліпсоїдів зображено на *рис.1*. Їх використання дає змогу моделювати анізотропні поля, “видовжені” по більших головних осях.

Особливістю гауссового фільтра є те, що він може бути незалежно розділений окремо по кожній з осей координат, що на порядки зменшує складність обчислень. У цьому випадку з $O(h_x h_y h_z r^3)$, де h_x, h_y, h_z – висота ширина та глибина матриці, до $O(h_x h_y h_z r)$. Слід також зазначити, що після застосування фільтра отримані значення можуть бути досить великими, і не попадати в діапазоні $[0;1]$. Щоб обійти цю проблему, достатньо виконати нормалізацію. На *рис. 2, b* зображено результат застосування гауссового фільтра до базового випадкового розподілу інтенсивностей в елементарному об’ємі гетерогенного середовища.

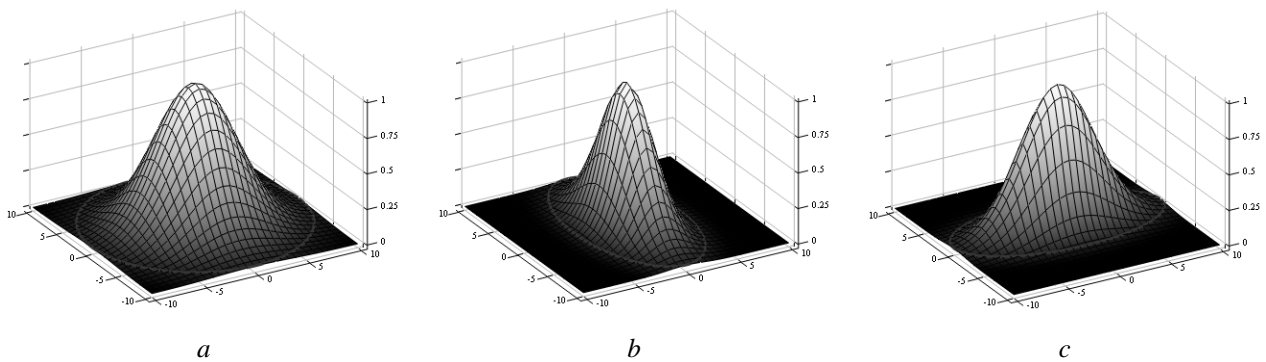


Рис. 1. Приклади двовимірних гауссових фільтрів: a – у формі кола з радіусом 10 одиниць; b – у формі еліпса з головними осями 5 та 10 одиниць; c – у формі еліпса з головними осями 10 та 5 одиниць

Останнім етапом моделювання є застосування до огрублених фільтром випадкових скалярних полів одного чи кількох перерізів на деякому рівні. Значення інтенсивностей, що не долають вибраний рівень, просто відкидаються. Приклад застосування єдиного перерізу до огрубленого фільтром випадкового скалярного поля зображено на *рис. 2, c*.

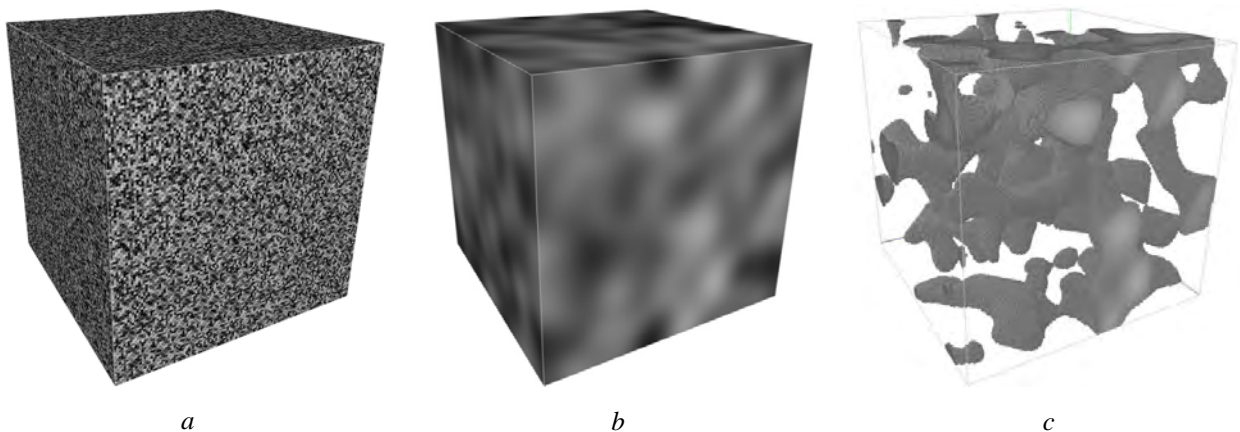


Рис. 2. Приклад моделювання структури гетерогенного середовища за допомогою випадкових скалярних полів. Елементарний об’єм є матрицею $128 \times 128 \times 128$ елементів: a – базовий рівномірний випадковий розподіл інтенсивностей в матриці; b – результат застосування гауссового фільтра у формі сфери радіусом 20 елементів; c – результат застосування єдиного перерізу, показано інтенсивності $\geq 0,63$

Моделювання випадкових градуйованих перехідних шарів

Градування, тобто моделювання деяких функціонально заданих переходів між фазами гетерогенних середовищ, широко застосовується у моделях функціонально градуйованих матеріалів

(FGM) [8], [9], [11]. Для градування моделей випадкових скалярних полів, у фільтр огрублення K , базового розподілу інтенсивностей, вводять спеціальну функцію інтенсивностей $I(x, y, z)$, що описує бажану форму випадкових полів [11], [12]. У найпростішому випадку функція інтенсивностей $I(x, y, z)$ описує лінійний спад, між фазами середовища, що умовно дорівнюють 1 та 0 (рис. 3, а). Після цього повторюються попередньо описані кроки моделювання випадкових скалярних полів, з тією відмінністю, що під час фільтрування до знайденого огрубленого значення поля треба додати значення $I(x, y, z)$ (рис. 3, б). Геометрично процедура додавання функції є еквівалентом обертання чи нахилу утвореного поля у додатковому вимірі, так що подальший переріз вже не буде паралельний до площини нульових інтенсивностей, а перетинатиме її під деяким кутом. Переріз повторюватиме форми функції $I(x, y, z)$. Цей підхід можна використовувати ітераційно. Якщо зменшувати на кожній ітерації розміри фільтра, з'являється можливість моделювати гетерогенні структури з різними масштабами градування (рис. 3, с).

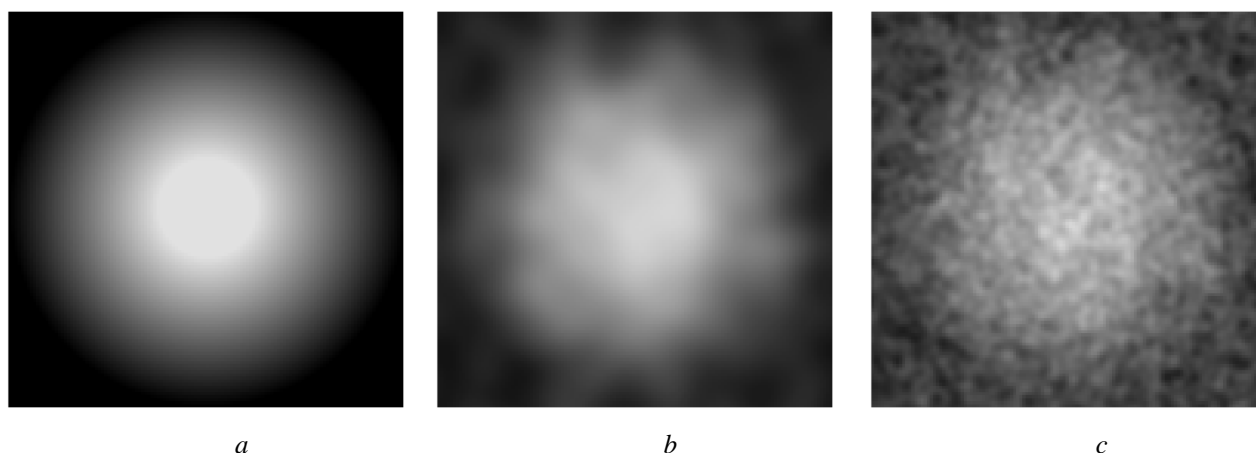


Рис. 3. Приклад моделювання випадкових градуйованих перехідних шарів. Елементарний об'єм є матрицею $128 \times 128 \times 128$ елементів. У центрі розміщено сферу радіусом 64 елементи. Перехідний шар становить 75 % = 48 елементів. Зображено переріз по екватору: а – функція інтенсивності описує лінійний спад від 1 до 0; б – результат застосування гауссового фільтра у формі сфери радіусом 16 елементів та з заданою функцією інтенсивності; с – результат повторного застосування гауссового фільтра у формі сфери радіусом чотири елементи та з функцією інтенсивності з попереднього кроку, що відповідає збільшенню деталізації під час мультимасштабного моделювання

Описаний метод побудови випадкових перехідних шарів має недоліки. Зокрема, він деформує базові примітиви (сфери, циліндри, тетраедри, еліпсоїди,...), що можуть бути попередньо задані простішими структурними моделями гетерогенних середовищ. Недолік можна виправити, використавши деяке маскування ділянок елементарного об'єму.

Іншим недоліком методу є те, що на великих масштабах, тобто за великого розміру фільтра, початкова функція інтенсивностей сильно деформує випадкове скалярне поле, нівелюючи його наявність. Через це, застосовуючи перерізи, неможливо побудувати окремі кластери великих розмірів, які б простягалися через весь перехідний шар. З іншого боку, якщо відкинути початкову функцію інтенсивностей і безпосередньо використовувати випадкове скалярне поле, отримані гетерогенні структури матимуть велику кількість розривів між кластерами та базовими фазами середовища. Щоб вийти з цієї ситуації, додаючи до знайденого огрубленого поля значення функції інтенсивності, достатньо помножити останнє на деякий коефіцієнт.

Враховуючи доволі велику кількість елементів матриці елементарного об'єму, високу латентність доступу до пам'яті на SIMD архітектурах (порівняно з обчислювальними можливостями) для побудови випадкових градуйованих перехідних шарів, пропонується така процедура:

1. Застосувати до початкової функції інтенсивностей $f(x, y, z) = I(x, y, z)$, збереженої у матриці елементарного об'єму F , подвійний переріз на рівнях $F_0 = 0 + \epsilon$ та $F_1 = 1 - \epsilon$, де $0 \leq \epsilon \leq 1$.

Прийняти значення, що не попадають в переріз, як замасковані, та присвоїти їм протилежні за знаком значення, зменшені на деяку константу C :

$$F(x, y, z) = \begin{cases} -F - C, & F < F_0 \vee F > F_1 \\ F, & F_0 \leq F \leq F_1 \end{cases} \quad (4)$$

При цьому вважаємо, що інтенсивність однієї фази середовища дорівнює нулю, а іншої – одиниці. Константа C необхідна для випадків, коли необхідно розглядати інтенсивності, що дорівнюють нулю чи одиниці. Для уникнення помилок заокруглення в числах з плаваючою комою, рекомендується приймати $1 < C \leq 2$. Зауважте, що немає необхідності створювати окрему матрицю маскування.

2. Створити нову матрицю G такого самого розміру, та побудувати в ній випадкове поле з застосуванням необхідного фільтра, за допомогою попередньо описаних кроків.

3. Помножити не замасковані значення функції інтенсивності, збереженої в F , на необхідний коефіцієнт a , та додати до них значення побудованого випадкового поля G .

4. Нормалізувати не замасковані значення F .

5. Зняти маску з F :

$$F(x, y, z) = \begin{cases} -F - C, & F < 0 \\ F, & F \geq 0 \end{cases} \quad (5)$$

Приклад роботи описаної процедури наведено на рис. 4.

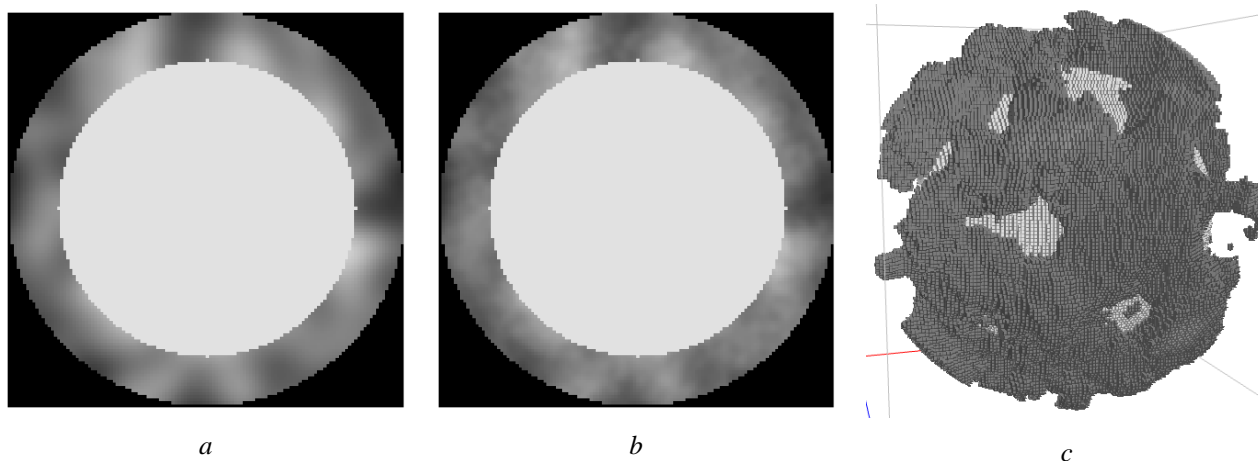


Рис. 4. Приклад моделювання випадкових градуйованих перехідних шарів з маскуванням.

Елементарний об'єм є матрицею $128 \times 128 \times 128$ елементів. У центрі розміщено сферу радіусом 64 елементи.

Перехідний шар становить 25% = 16 елементів. Зображено переріз по екватору: *a* – результат застосування гауссового фільтра у формі сфери радіусом 16 елементів та з заданою функцією інтенсивності, помноженою на 0,25; *b* – результат повторного застосування гауссового фільтра у формі сфери радіусом чотири елементи та з функцією інтенсивності з попереднього кроку, помноженою на 5,0; *c* – результат застосування єдиного перерізу, показано інтенсивності $\geq 0,55$

Реалізація методу побудови випадкових перехідних шарів за допомогою OpenCL

Технологія OpenCL (Open Computing Language) є однією з технологій високопродуктивних паралельних та розподілених обчислень. Вона дає змогу реалізовувати концепцію паралелізму даних на широкому колі обчислювальних пристроїв, зокрема на центральних процесорах (CPU), на графічних картах, що підтримують інтерфейси загального використання (GPGPU), на програмованих користувачем вентильних матрицях (FPGA) та на процесорах цифрової обробки сигналів (DSP). У цьому випадку основну увагу звернуто саме на графічні карти, а також на багатоядерні центральні процесори. Ці класи обчислювальних пристроїв мають SIMD (Single Instruction Multiple Data) архітектуру і дають змогу максимально ефективно виконувати запрограмовані алгоритми, що використовують декомпозицію даних великих задач.

Основними особливостями роботи з графічними картами є висока латентність використання пам'яті, особливо у випадках пересилання даних з чи на пристрій. Тому максимально ефективного результату можна досягти у разі локальної обробки доволі великих порцій даних. Більшість таких пристроїв не реалізують кешування даних. Зазвичай не підтримується динамічне виділення пам'яті. Графічні карти містять велику кількість (порівняно з центральними процесорами) робочих вузлів, що одночасно виконують однакові інструкції. Якщо в інструкціях наявні програмні розгалуження чи умовні оператори, залежно від результатів виконання, деякі робочі вузли можуть простоювати, знижуючи цим максимальне ефективне навантаження на пристрій. Детальніше з цими особливостями можна ознайомитися у специфікації OpenCL або в [13].

Розробники пристроїв, що підтримують технологію OpenCL, зазвичай поставляють власний відповідний набір із засобів розроблення, утиліт і документації програмних продуктів (SDK). Кожен такий набір містить прикладні програмні інтерфейси (API) для роботи з пристроєм, API для взаємодії CPU і пристрою, та компілятор+компонувальник алгоритмічної мови OpenCL. Разом ці елементи утворюють так звану OpenCL платформу. Алгоритмічна мова технології OpenCL є C-подібною мовою програмування. Класично типова OpenCL програма складається з таких етапів:

- Визначити наявні в системі OpenCL платформи.
- Визначити наявні пристрої на платформах.
- Створити програмний OpenCL контекст для платформи. Контекст відповідає за взаємодію CPU і пристрою.
- Скомпілювати під створений контекст написані C-подібні OpenCL коди програм. Отримані програмні об'єкти аналогічні до динамічно компонованих бібліотек, що будуть розміщуватися на пристрої.
- Створити ядра виконання (kernel-об'єкти). Ядра є точками початку виконання інструкцій, які виконують робочі вузли.
- Створити чергу команд.
- Створити необхідні буфери пам'яті.
- Запустити в черзі команд на виконання потрібну кількість ядер.
- Скопіювати з пристрою отриману необхідну інформацію.
- Знищити всі створені об'єкти.

Враховуючи описані кроки, пропонуємо алгоритм побудови випадкових скалярних полів:

1. Створити на пристрої два буфери пам'яті “_data” та “_buffer” розміру “_size“x”_size“x”_size”. У перший скопіювати попередньо створений базовий випадковий розподіл інтенсивностей.
2. Створити програмні ядра гауссового фільтра, що запускатимуться на виконання, по одному ядру на кожен вимір задачі.
3. Створити чергу команд.
4. Визначити кількість програмних ядер, що будуть одночасно запуснені на виконання, відповідно до наявної на пристрої кількості робочих вузлів.
5. Застосування фільтра по одному на вимір задачі:
 - a. Заповнити буфер “_buffer” нулями.
 - b. Запустити на виконання знайдену кількість відповідних цьому виміру програмних ядер. Дані, що фільтруються, розміщені в буфері “_data”, результат розміщується в буфері “_buffer”.
 - c. Скопіювати отримані відфільтровані результати з буфера “_buffer” в буфер “_data”.
6. Скопіювати з пристрою дані, розміщені в будь-якому з буферів (після останнього виконання кроку 5.с вони будуть однаковими).
7. Нормалізувати отримані дані.
8. Знищити всі створені програмні об'єкти.

Оскільки далеко не всі OpenCL пристрої підтримують обчислення з плаваючою комою подвійної точності, у цій роботі всі обчислення виконують в одинарній точності, тобто з використанням типу даних float. З метою мінімізації кількості звернень до пам'яті гауссові фільтри реалізуються у вигляді простої функції:

```
float _GaussianFilter(
    float r,
    float x, float y, float z,
    float fx, float fy, float fz){
    return exp(-(x*x/fx/fx + y*y/fy/fy + z*z/fz/fz) / ((r/2.0) * (r/2.0)));}
```

Наступним кроком є створення трьох програмних ядер, що запускатимуться на виконання, по одному ядру на кожен вимір. Особливістю роботи з технологією OpenCL є те, що кількість запусків на виконання програмних ядер повинна бути кратною до кількості груп робочих вузлів, що є на пристрої. Разом ці групи утворюють окремі обчислювальні вузли. Їх кількість зазвичай дорівнює деякому степеню двійки, тому, для спрощення, розміри матриці елементарного об'єму гетерогенного середовища також дорівнюють деякому степеню двійки. Це також дає змогу обійти проблему фільтрації на границях матриці, перенесенням індексів на протилежну сторону елементарного об'єму, за допомогою простих побітових операцій. Утворені структури будуть ідентичними на протилежних сторонах елементарного об'єму. Нижче наведено одне з трьох програмних ядер, що реалізують гауссовий фільтр:

```
__kernel void applyGaussianFilterX(
    int discreteRadius,
    float ellipsoidScaleFactorX,
    float ellipsoidScaleFactorY,
    float ellipsoidScaleFactorZ,
    __global float *_data,
    __global float *_buffer,
    int _size)\
{
    long i = get_global_id(0);
    long j = get_global_id(1);
    long k = get_global_id(2);

    for( int p = -discreteRadius; p <= discreteRadius; ++p)
        _buffer[(i * _size * _size) + (j * _size) + k] +=
            _data[(((i+p)&(_size-1)) * _size * _size) +
                (j * _size) + k] *
            _GaussianFilter(
                discreteRadius,
                p, 0, 0,
                ellipsoidScaleFactorX,
                ellipsoidScaleFactorY,
                ellipsoidScaleFactorZ);
}
```

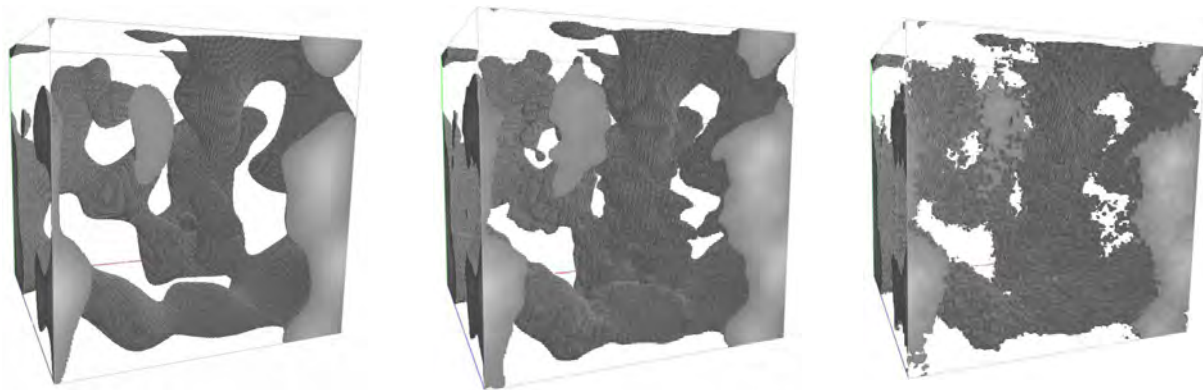
Кількість програмних ядер, що будуть одночасно запуснені на виконання, відповідно до наявної на пристрої кількості робочих вузлів, за можливості беруть максимальною. Для цього перевіряється інформація пристрою за OpenCL параметром "CL_DEVICE_MAX_WORK_GROUP_SIZE". Для побудови випадкових градуєваних перехідних шарів описаний алгоритм та його програмну реалізацію треба підставити в процедуру, описану в попередньому розділі.

Результати мультимасштабного моделювання випадкових перехідних шарів

Описаний у цій роботі метод програмно реалізований на алгоритмічній мові C++11 з використанням OpenCL версії 1.2, та Qt SDK версії 5.4.1. Робоча операційна система Windows 7 Ultimate x64. Програма зібрана під x64 за допомогою компілятора MinGW версії 4.9.2. Моделювання проводилися на персональному комп'ютері пересічної комплектації. Зокрема:

- OpenCL GPU: "Cedar (AMD Radeon HD 6300M Series)", AMD OpenCL SDK 2.9.1 driver 1445.5 (VM), 2x12 computing units 0.750 GHz, (робоча група 4x4x4).
- OpenCL CPU: "Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz", AMD OpenCL SDK 2.9.1 driver 1445.5 (sse2,avx), 2x2 computing units 2.294 GHz, (робоча група 8x8x8).

Також моделювання виконано в однопотоківій версії програми, без використання OpenCL. Отримані результати наведено на рис. 5–9.



Масштаб 1

Масштаб 2

Масштаб 3

Рис. 5. Результати застосування єдиного перерізу інтенсивностей $\geq 0,63$ у різних масштабах моделювання випадкових полів

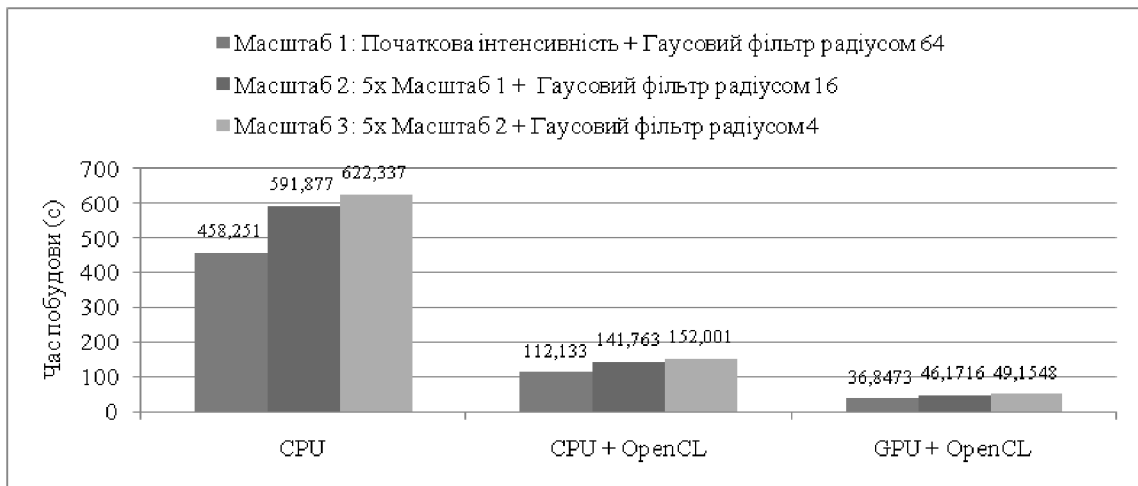
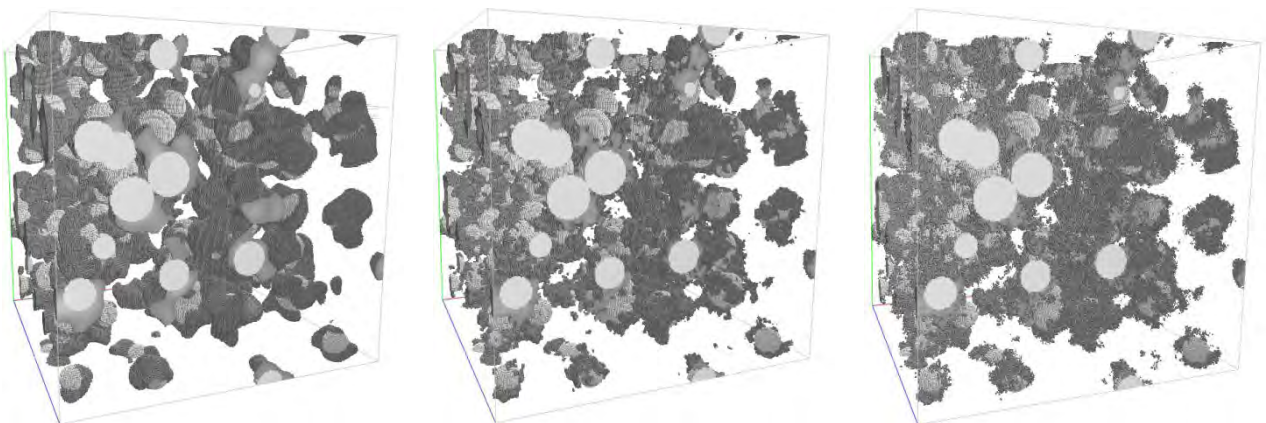


Рис. 6. Результати мультимасштабного моделювання випадкових скалярних полів для елементарних об'ємів у вигляді матриці розміром $256 \times 256 \times 256$ елементів. Початковим є рівномірний розподіл інтенсивностей

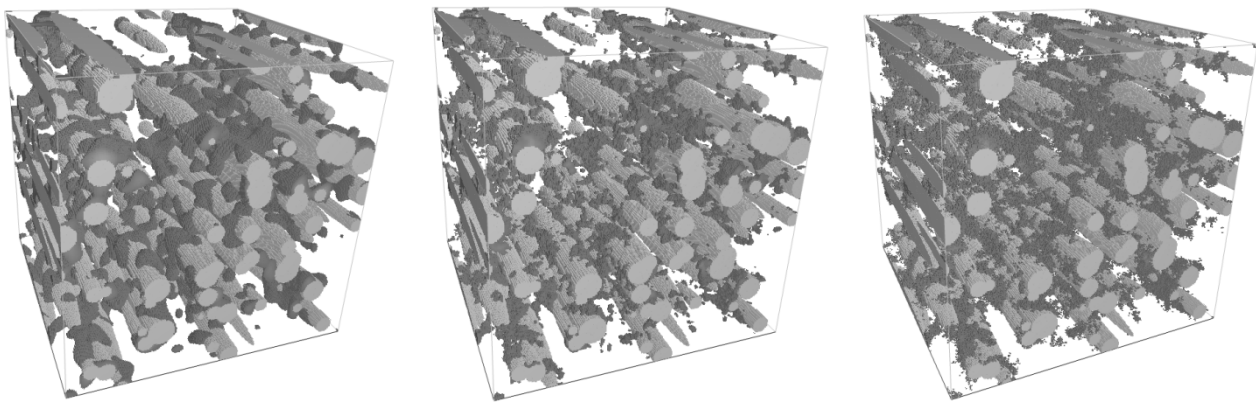


Масштаб 1

Масштаб 2

Масштаб 3

Рис. 7. Результати мультимасштабного моделювання випадкових градуїюваних перехідних шарів для елементарних об'ємів у вигляді матриці розміром $256 \times 256 \times 256$ елементів. Початковим є заповнення елементарного об'єму 150 сферами, що перетинаються, з мінімальним радіусом 16 елементів та максимальним радіусом 32 елементи. Перехідний шар становить половину радіуса кожної сфери. Початкова інтенсивність перехідного шару змінюється лінійно між фазами середовища. Показано результат застосування єдиного перерізу інтенсивностей $\geq 0,63$ у різних масштабах



Масштаб 1

Масштаб 2

Масштаб 3

Рис. 8. Результати мультимасштабного моделювання випадкових градуйованих перехідних шарів для елементарних об'ємів у вигляді матриці розміром $256 \times 256 \times 256$ елементів. Початковим є заповнення елементарного об'єму 75 еліпсоїдами (сфера 100 % по осі X, 10 % по осі Y, 10 % по осі Z), що перетинаються, з мінімальним радіусом 128 елементів та максимальним радіусом 256 елементів. Перехідний шар становить половину радіуса кожного еліпсоїда. Початкова інтенсивність перехідного шару змінюється лінійно між фазами середовища. Показано результат застосування єдиного перерізу інтенсивностей $\geq 0,63$ у різних масштабах

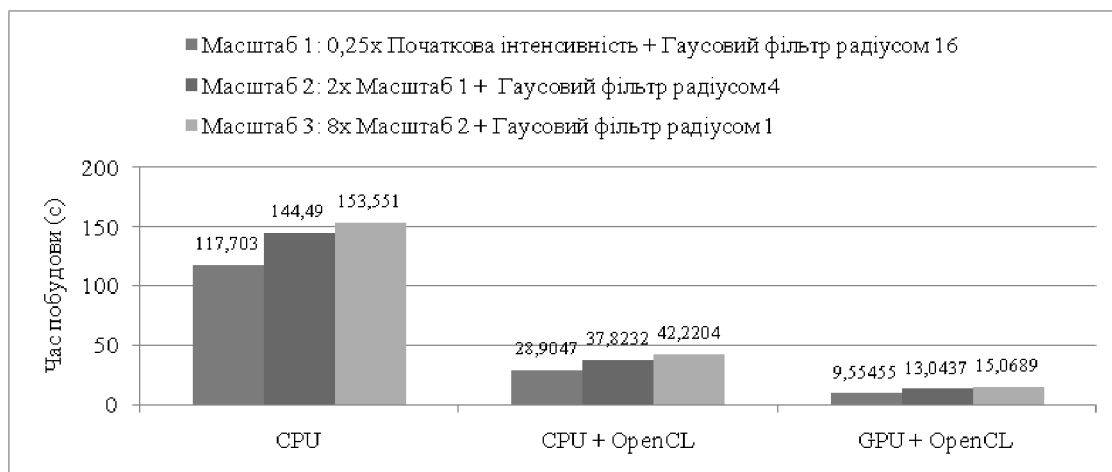


Рис. 9. Результати мультимасштабного моделювання випадкових градуйованих перехідних шарів для елементарних об'ємів у вигляді матриці розміром $256 \times 256 \times 256$ елементів

Висновки

У цій роботі, на основі методів побудови випадкових скалярних полів та методів побудови градуйованих перехідних шарів, запропоновано метод побудови випадкових мікрорівневих перехідних шарів, що дає змогу моделювати, рекурентно у різних масштабах, структури гетерогенних середовищ як комбінацію детермінованих елементів із заданими стохастичними перехідними шарами між ними.

Крім того, наводиться приклад реалізації запропонованого методу на основі технології високопродуктивних паралельних і розподілених обчислень OpenCL. Це дає змогу ефективно виконувати моделювання на обчислювальних пристроях з SIMD архітектурою, наприклад, на графічних картах чи багатоядерних центральних процесорах. Зокрема, наведені результати моделювання, з яких видно, що навіть з використанням персональних комп'ютерів пересічної

комплектації, реалізація за допомогою технології OpenCL більш ніж у десять разів пришвидшує обчислення таких задач.

1. Jaworski N., Farmaga I., Lobur M., Spiewak P. – *Research of composite materials optimal design task based on numerical simulation // Proc. of the 8th International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT’2013).* – P. 46–48, Lviv, 2013.
2. Нарусберг В., Тетерс Г. *Устойчивость и оптимизация оболочек из композитов.* – Рига: Зинатне, 1988.
3. Hashin Z. *The elastic moduli of heteroheneous materials // Journal of Applied Mechanics*, 29(1): 143-150, 1962.
4. Christinsen R. *Mechanics of composite materials // New-York: Wiley*, 1979.
5. Dunn M., Ledbetter H. *Elastic moduli of composites reinforced by multiphase particles // Journal of Applied Mechanics*, 62(4):1023–1028, 1995.
6. Odegard G., Clancy T., Gates T. *Modeling of the mechanical properties of nanoparticle/polymer composites // Polymer*, 46(2):553–562, 2005.
7. Дульнев Г., Заричняк Ю. *Теплопроводность смесей и композиционных материалов: справочная книга.* – Ленинград: Энергия, 1974.
8. Reiter T., Dvorak G., Tvergaard V. *Micromechanical Models for Graded Composite Materials // Journal of the Mechanics and Physics of Solids*, 45(8):1281–1302, 1997.
9. Ostoja-Starzewski M. *Random Field Models of Heterogeneous Materials // International Journal of Solids and Structures*, 35(19): 2429–2455, 1998.
10. Torquato S. *Random Heterogeneous Materials. Microstructure and Macroscopic Properties // New-York: Springer*, 2002.
11. Rahman S. *A random field model for generating synthetic microstructures of functionally graded materials // International Journal for Numerical Methods in Engineering*, 76(1):972–993, 2008.
12. Rahman S. *Multi-scale fracture of random heterogeneous materials // Ships and Offshore Structures*, 4(3):261–274, 2009.
13. Munshi A., Gaster B., Mattson T., Fung J., Ginsburg D. *OpenCL Programming Guide.* – New-York: Addison-Wesley, 2011.

O. Troyan

Lviv Polytechnic National University
Department of Information Technology Publishing Affair

METHOD OF FORMING LATENT IMAGE TO PROTECT DOCUMENTS BASED ON THE EFFECT MOIRE

© Troyan O., 2015

Analysis of modern methods of information protection based on printed documents. It is shown that methods of protection from moiré effect provide reliable and effective protection by gaining new protection technology that is displayed in the optical acceleration motion layers and causes moire in fraud. Latent images can securely protect paper documents. Introduce a system of equations to calculate curvilinear patterns, where the optical formula of acceleration and periods moire stored in its simplest form. There is a need to create new types of protection of printed documents as the means and methods of fraud are becoming more common.

Key words: method of protection, securities, security printing.

Accession

With the development of computer technology forgery of documents is common. Manufacturing documents are easier to falsification consequence of that is becoming more development. In this regard, there is need for protection of electronic and paper documents in new ways. Computer industry and modern copiers developing rapidly. Modern computer systems and networks face different attacks, so now becoming urgent question of security of electronic documents. Given the variety of potential threats to information protection of electronic documents should be as high. To be effective protection it must meet