

змінювати. Так само вони створені для спрощення налагодження, коли зміна коду є небажаною. Також корисно використовувати декоратори для розширення різних функцій одним і тим же кодом, без повторного його переписування кожен раз.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мова програмування Python 3, документація. Рівень доступу: <https://docs.python.org/3/>
2. Публікація Т. Popovic: «Advanced Python Techniques: Decorators», 2015. Рівень доступу: https://www.researchgate.net/publication/272833859_Advanced_Python_Techniques_Decorators
3. Реалізація шаблону «Декоратор» у Python, Implementing the Decorator Pattern in Python, 2020. Рівень доступу: <https://dev.to/erikwhiting88/implementing-the-decorator-pattern-in-python-1fdm>
4. M. Harrison, Guide To: Learning Python Decorators. - CreateSpace 2013. - 59 с. - ISBN 9781492325611.
5. E. Matthes, Python Crash Course: A hands-on, project-based introduction to programming. - 2015. - 446-453с. - ISBN 978-1-59327-603-4.

УДК 004.415.3:681.6

С.М. Мацюк¹, Я.І. Журавльов¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ГЕНЕРАЦІЯ МОДЕЛІ ЛАНДШАФТУ НА ОСНОВІ РЕГУЛЯРНОЇ СІТКИ ВИСОТ

Анотація. Проведено аналіз основних принципів представлення даних для зберігання інформації про ландшафти. Досліджено різні методи генерації тривимірних ландшафтів. Описано алгоритм для генерації моделі тривимірного ландшафту на основі регулярної сітки висот.

Ключові слова: ландшафт, триангуляція, карта висот, регулярна сітка висот, побудова поверхонь, тривимірна модель.

Вступ. В даний час 3D-моделювання є найважливішою областю машинної графіки, оскільки побудова тривимірного зображення, близького до реального, є досить складним завданням. Але, завдяки великому колу споживачів і неймовірно швидкому зростанню продуктивності обчислювальних систем, ця область активно розвивається.

Модель 3D поверхні є цифровим відображенням просторових об'єктів, як реальних, так і гіпотетичних, в тривимірному просторі. Простими прикладами 3D поверхонь є ландшафти, міські вулиці, підземні газові сховища або мережа

колодязів із зазначенням їх глибини, яку можна використовувати для визначення глибини водоносного шару. Всі ці приклади є реальними об'єктами, але поверхні також можуть бути обчисленими або уявними. Вигадані 3D поверхні – поверхні, які часто зустрічаються у відеоіграх чи комп'ютерних симуляціях.

3D поверхні зазвичай обчислюються з використанням спеціальних алгоритмів, які обробляють вихідні точкові, лінійні або полігональні дані і конвертують їх у цифрову 3D поверхню. Ці моделі поверхонь можуть бути створені з безлічі різних джерел даних.

Постановка задачі. Необхідно дослідити генерацію ландшафту різними методами та розробити програмне забезпечення, що відображає тривимірну модель ландшафту за вхідними растровими даними. Дані ландшафту задані картою висот у вигляді зображення формату .bmp, де висота визначається кольором.

Основний зміст роботи. Під час роботи були розглянуті наступні методи та генерації тривимірних ландшафтів:

- Алгоритм diamond-square для побудови фрактальних ландшафтів
- Алгоритм mid point displacement
- Алгоритм diamond-square
- Ландшафт на базі діаграми Вороного
- Генерація ландшафтів за допомогою пагорбового алгоритму (HillAlgorithm).

У результаті проведеного аналізу для генерації тривимірної моделі ландшафту було обрано пагорбовий алгоритм (HillAlgorithm). Це ітераційний алгоритм, заснований на декількох вхідних параметрах. Алгоритм викладено в наступних кроках:

- Створюється двомірний масив та ініціалізується нульовим рівнем;
- Береться випадкова точка на ландшафті або близько його меж (за межами), а також береться випадковий радіус в заздалегідь заданих межах. Вибір цих меж впливає на вид ландшафту – або він буде пологим, або скелястим;

- У вибраній точці "піднімається" пагорб заданого радіуса;

Фактично пагорб – це половина кулі, чим більше її радіус, тим більше пагорб (і вище). Математично це виражено наступним чином.

$$z = r^2 - ((x_2 - x_1)^2 + (y_2 - y_1)^2), \quad (1)$$

де (x_1, y_1) - задана точка, r - вибраний радіус, (x_2, y_2) - висота пагорба.

Щоб згенерувати ландшафт повністю, необхідно побудувати безліч таких пагорбів. Але є ще дві речі на які, необхідно звернути увагу. Перше – необхідно ігнорувати негативні значення висоти пагорба. Друге – при генерації наступних пагорбів краще додавати отримане значення для даного пагорба до вже існуючих значень. Це дозволяє побудувати більш правдоподібний ландшафт, ніж правильно окреслені округлі пагорби.

- Повернення до другого кроку і повторення до вибраної кількості кроків;

- Нормалізація ландшафту;

При генерації значень для ландшафту не враховувались виходи цих значень за деякі межі (наприклад, якщо потім ландшафт буде зберігатися в монохромній картинці, то необхідно, щоб всі значення перебували в межі від 0 до 256). Для цього необхідно провести нормалізацію значень.

1) спершу пройти по всьому масиву і запам'ятати найбільше і найменше значення;

2) після того, як відомі ці значення, треба заново пройти по всьому ландшафту і зробити нормалізацію конкретних значень в межі від 0 до 1.

$$z_{norm} = \frac{z - \min}{\max - \min} \quad (2)$$

У результаті отримується готовий ландшафт, нормалізований і готовий до подальшого використання.

- "Долінізація" ландшафту.

Але якщо придивитися до отриманого ландшафту, то в ньому досить мало долин. Схили пагорбів надмірно круті, треба зробити їх більш пологими. У цьому допоможе попередній крок – нормалізація. Ідея "долінізації" полягає в наступному – взяти від кожного значення квадратний корінь. Це більшою мірою впливає на середні значення, практично не зачіпаючи мінімумів і максимумів.

Для реалізації алгоритму було розроблено програмне забезпечення з графічним інтерфейсом користувача. Вхідні дані представлені регулярною сіткою висот, яка зберігається у вигляді зображення .bmp формату. Тоді двома координатами x та y буде положення конкретного пікселя на картинці, а третя координата – висота z буде представлена кольором: чим вище значення, пряма залежність від яскравості пікселя – тим більше значення висоти для цієї точки. Зазвичай такі зображення містяться в монохромному варіанті, але можна використовувати і повноколірні версії. Другий варіант дає більше градацій висоти, ніж передбачувані 256 градацій в разі монохромного подання.

Тест 1

Для того, щоб побудувати модель ландшафту по карті висот, треба завантажити файл із розширенням .bmp.

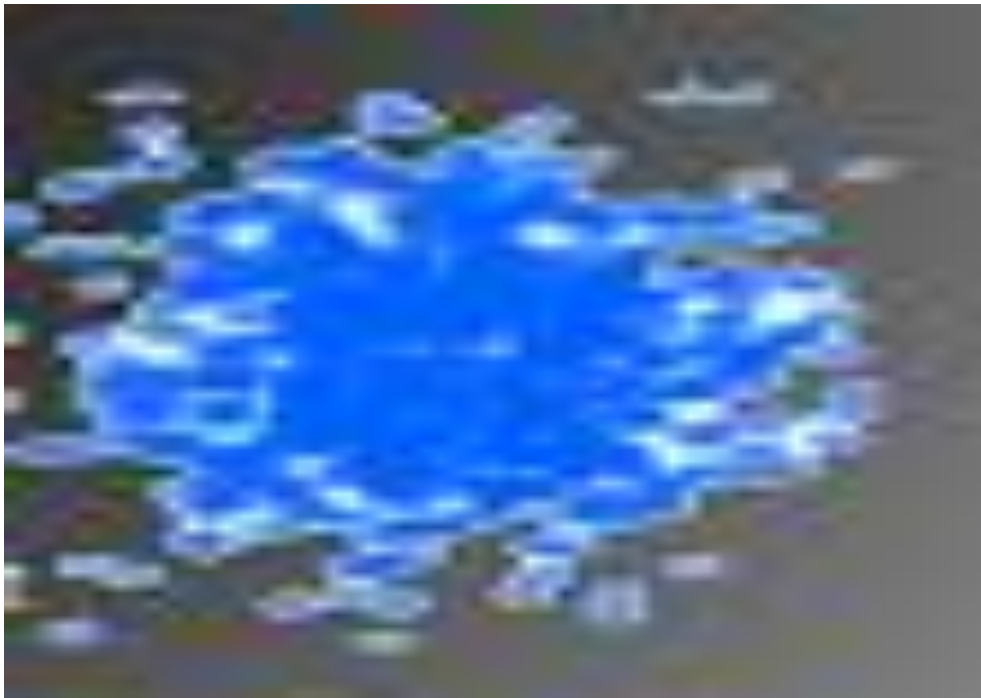


Рис. 1. Вихідний файл для побудови моделі

Після завантаження файлу з картою висот на екрані відображається тривимірна модель ландшафту.

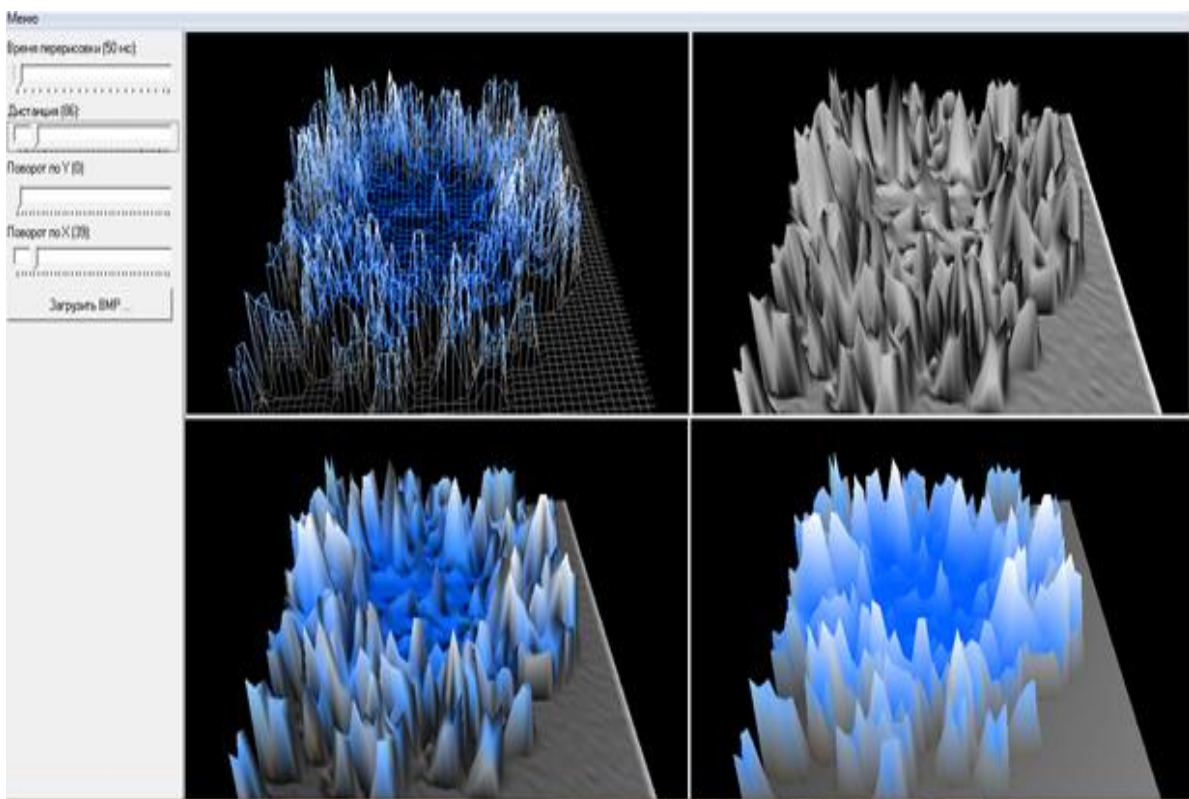


Рис. 2. Тривимірна модель ландшафту

Перед користувачем з'являється чотири різних моделі одного і того ж ландшафту, який був побудований з карти висот, яка зображена на рис. 2. Розглянемо кожне із зображень. У прямокутнику зображеному на рис. 3 (а) відображається каркасна модель ландшафту. На рис. 3 (б) відображається ландшафт візуалізований у суцільному вигляді.

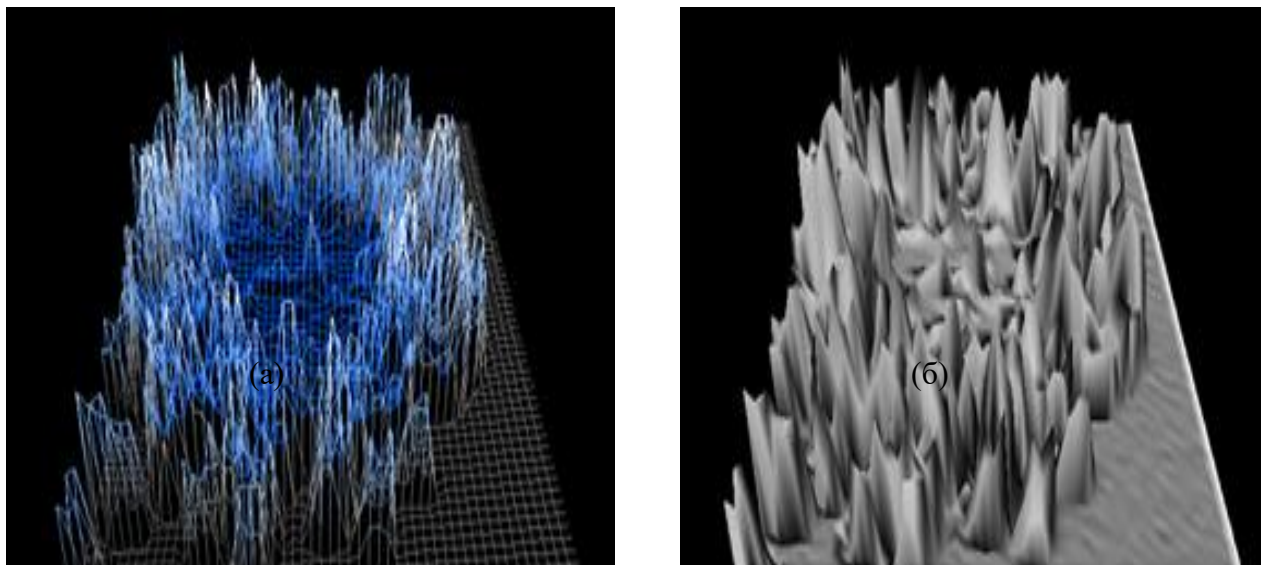


Рис. 3. (а) – ландшафт, візуалізований в каркасному режимі;
(б) – ландшафт, візуалізований у суцільному режимі

Для того, щоб правильно освітлювати та текстурувати ландшафт для кожної вершини треба обчислювати нормаль і координати текстури, як це зроблено на рис. 4 (а). А ось у випадку на рис. 4 (б) не було обчислено нормаль до кожної вершини, що призвело до неправильного текстурування ландшафту.

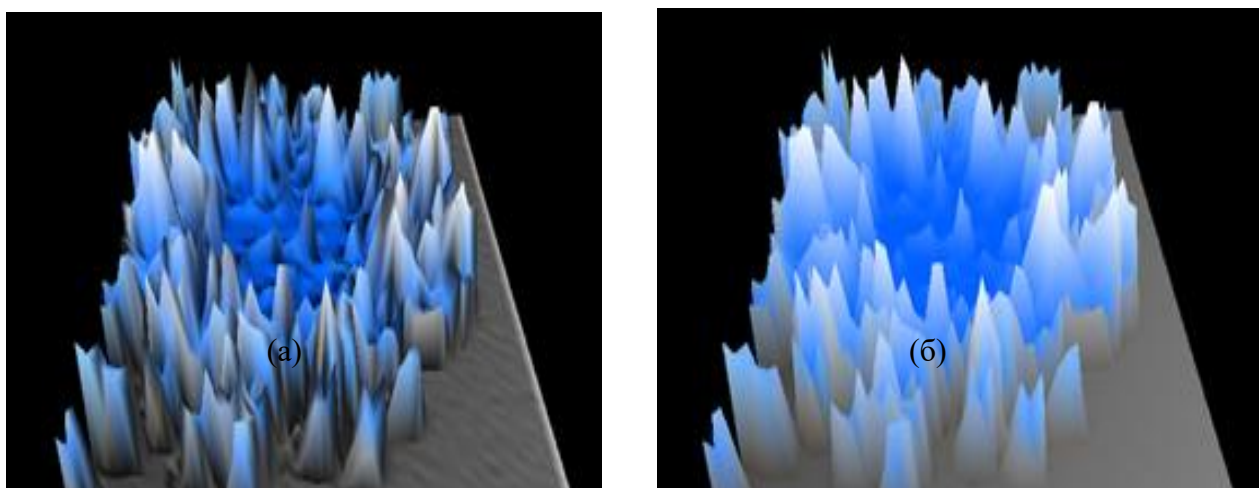


Рис. 4 (а) – тривимірна модель ландшафту з обчисленими нормальями та координатами текстури до кожної вершини;(б) – тривимірна модель ландшафту без обчислення нормалей до кожної вершини

Тест 2

Завантажимо до програми кольорове зображення ландшафту поверхні Землі (рис. 5).



Рис. 5 Карта висот частини поверхні Землі

Подивимось на отриману тривимірну модель ландшафту частини земної поверхні на рис. 6.

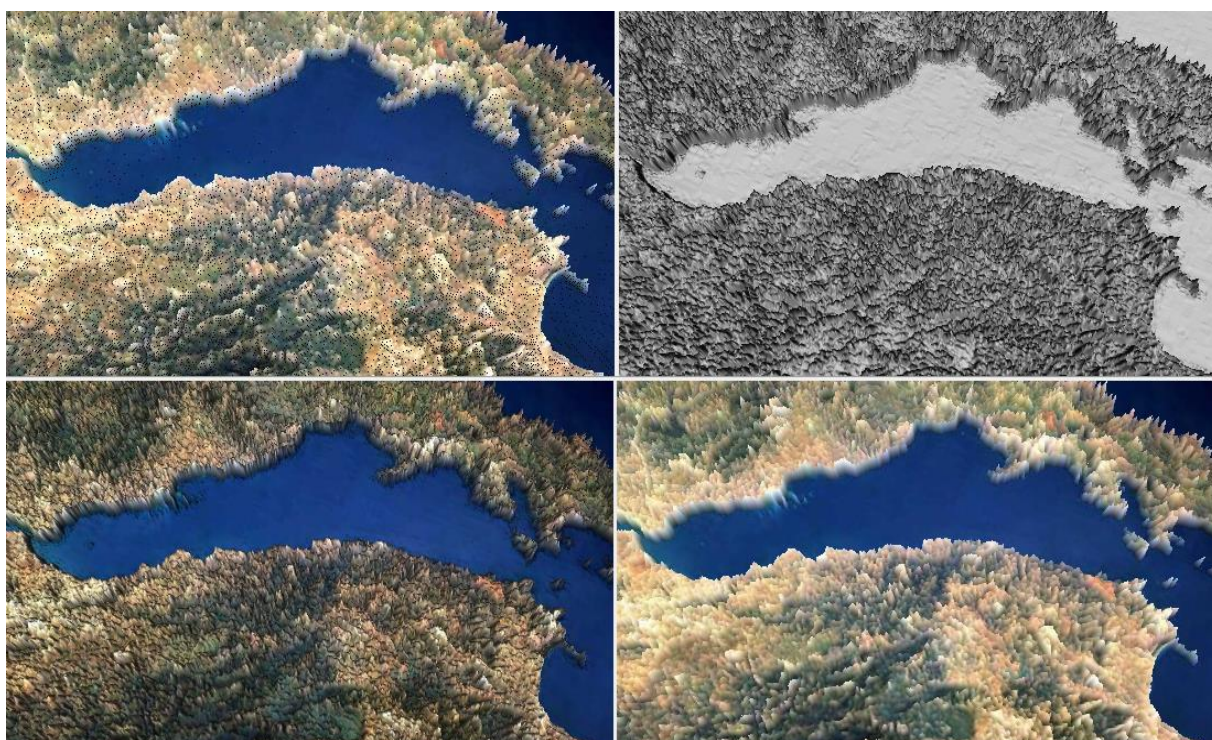


Рис. 6 Тривимірна модель ландшафту частини земної поверхні

Якщо приблизити дану модель, можна побачити різкі перепади між висотами вершин (рис. 7).

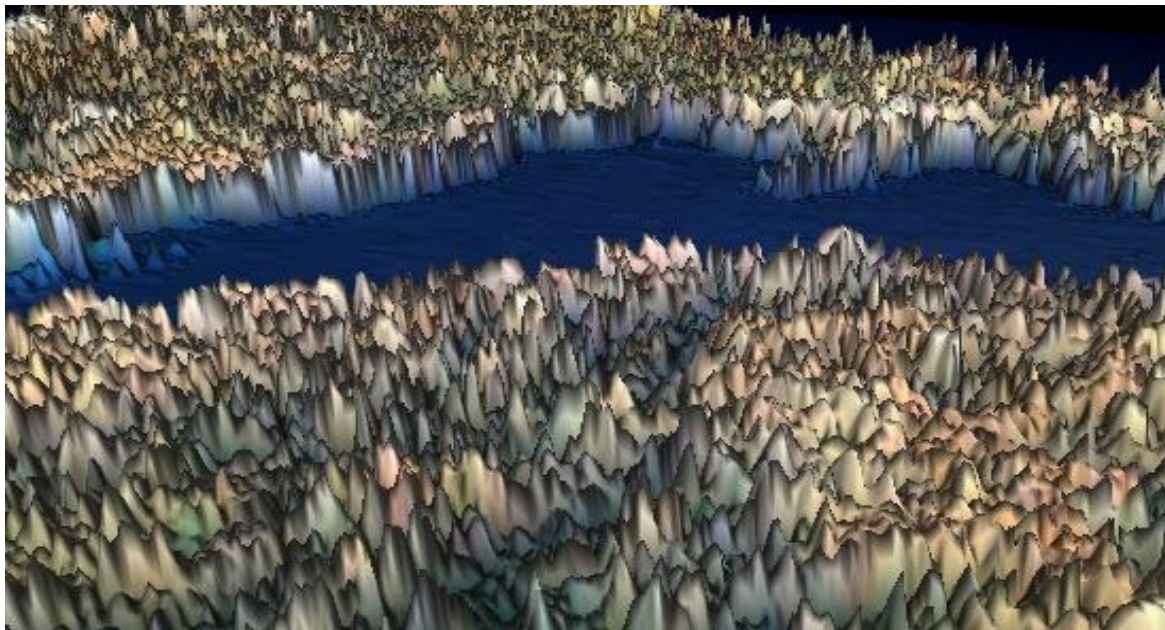


Рис. 7. Різкі перепади між висотами вершин

Це відбувається, тому що менша відстань між вершинами дозволяє згладити перепади між висотами вершин, але зменшує розмір сітки, в той час як велика відстань між вершинами збільшує розмір сітки, але може привести до різких перепадів між висотами вершин.

Наукова новизна: досліджено та удосконалено алгоритм для генерації моделі ландшафту на основі регулярної сітки висот у вигляді програмного забезпечення, яке містить в собі функції завантаження карти висот із растрового зображення, обертання отриманої моделі під різним кутом, віддалення та приближення тривимірної моделі.

Висновки. Поставленої цілі було досягнуто за рахунок впровадження покращення в структуру вже існуючого пагорбового алгоритму за рахунок використання регулярної сітки висот, яка зберігається у вигляді растрового зображення. У ході експерименту була також виявлена залежність між відстанню між вершинами ландшафту, та розміром сітки висот, яка може чинити безпосередній вплив на якість отриманої тривимірної моделі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Jhat Z. A., Mir A. H., Rubab S. 2011. Fingerprint Texture Feature for Discrimination and Personal Verification. International Journal of Security and its Applications, Vol. 5, No. 3
2. V. Dixit, Deepti Singh, Parul Raj, M. Swathi, P. Gupta, kd-tree based fingerprint identification system 01/2008; DOI:10.1109/IWASID.2008.4688340

3. Zhengu O., Feng J., Su F., Cai A., 2006. Fingerprint Matching with Rotation-Descriptor Texture Features. The 18th International Conference on Pattern Recognition, pp. 417-420.
4. Johan De Boer, Asker M Bazen, Sabih H Gerez, Indexing fingerprint databases based on multiple features Journal of The Acoustical Society of America- J ACOUST SOC AMER. 12/2001;
5. C. Wilson, M. Garris, C. Watson, A. Hicklin, Studies of Fingerprint Matching Using the NIST Verification Test Bed (VTB). Technical Report NISTIR 7020, July 2003.
6. C. Watson, C. Wilson, M. Indovina, R. Snelick, K. Marshall, Studies of One-to-One Matching with Vendor SDK Matchers. Technical Report NISTIR 7119, July 2004.
7. Орещенко А.В. Способы програмної реалізації тривимірних реалістичних картографічних моделей, 2009.
8. Вурста С.Ю., Літнарівч Р.М. Побудова фрактальних поверхонь в комп'ютерній графіці – м. Рівне, 2010.
9. Ensemble of Clustering Algorithms for Large Datasets / I. A. Pestunov, V. B. Berikov, E. A. Kulikova, S. A. Rylov // Optoelectronics, Instrumentation and Data Processing. – 2011. – Vol. 47, No. 3. – P. 245–252.
10. Fern X. Z. Solving cluster ensemble problems by bipartite graph partitioning / X. Z. Fern, C. E. Brodley // Proceedings of the 21 st International Conference on Machine Learning, Canada, 2004. – P. 47.
11. Ackerman M. Measures of Clustering Quality: A Working Set of Axioms for Clustering / B.-D. Shai, M. Ackerman // Proceedings of NIPS Conference, 2008. – P. 121–128.