УДК 004.94
[1]Ройко О.Ю., к.т.н., [2]Багнюк Н.В., к.т.н., доцент, [1]Грицан П.А., [1]Шепелюк Г.С.
[1]Волинський коледж Національного університету харчових технологій
[2]Луцький національний технічний університет

ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СТРУКТУР ДАНИХ ТА АЛГОРИТМІВ ДЛЯ МОДЕЛЮВАННЯ СКЛАДНИХ ПОВЕРХОНЬ З ВРАХУВАННЯМ ЇХ ДИФЕРЕНЦІАЛЬНИХ ХАРАКТЕРИСТИК

Ройко О.Ю., Багнюк Н.В., Грицан П.А., Шепелюк Г.С. Особливості програмної реалізації структур даних та алгоритмів для моделювання складних поверхонь з врахуванням їх диференціальних характеристик. У роботі розглянуто програмну реалізацію алгоритмів загущення та спрощення дискретних моделей поверхонь з врахуванням значень гаусової або середньої кривини. Моделі поверхонь представлені у вигляді сітки з трикутними комірками, елементи якої індексуються квадродеревом або BSP-деревом.
Ключові слова: дискретне моделювання поверхонь, алгоритм, квадродерево, BSP-дерево.

Ройко А.Ю., Багнюк Н.В., Грицан П.А., Шепелюк Г.С. Особенности программной реализации структур данных и алгоритмов для моделирования сложных поверхностей с учетом их дифференциальных характеристик. В работе рассмотрено программную реализацию алгоритмов сгущения и упрощения дискретных моделей поверхностей с учетом значений гауссовой или средней кривизны. Модели поверхностей представлены в виде сетки с треугольными ячейками, элементы которой индексируются квадродеревом или BSP-деревом.
Ключевые слова: дискретное моделирование поверхностей, алгоритм, квадродерево, BSP-дерево.

Roiko O.Y., Bahniuk N.V., Hrytsan P.A., Shepelyuk G.S. Software implementation of data structures and algorithms for modeling complex surfaces, taking into account their differential characteristics. In this paper considered the software implementation of algorithms for condensation and simplification of discrete surface models with allowance for Gaussian or mean curvature. Surface models are represented as a mesh with triangular cells, whose elements are indexed by a quadtree or BSP tree.
Keywords: discrete surface modeling, algorithm, quadtree, BSP-tree.

Постановка проблеми. Зростання складності інженерних завдань, яке викликане стрімким розвитком технічних рішень та нарощуванням темпів промислового виробництва, ставить перед дослідниками ряд невирішених проблем. До таких можна віднести раціональне використання обчислювальних ресурсів при вирішенні здач та створення адекватних моделей процесів, систем та окремих об'єктів. Незважаючи на стрімкий розвиток обчислювальних систем, потреба в розробці ефективних алгоритмів моделювання стала ще гострішою, оскільки зросли вимоги до точності та достовірності самих моделей. Якщо говорити конкретніше про геометричні моделі поверхонь та тривимірних тіл, то така модель повинна і візуально, і геометрично відтворювати параметри реального об'єкта з максимальною достовірністю, залишаючись при цьому достатньо швидкодіючою. Це важливо в таких областях як тривимірний друк, мобільне програмне забезпечення, доповнена реальність, та інших, де потрібно взаємодіяти в реальному часі з геометричним об'єктом.

Очевидно, що складність моделі напряму залежить від кількості елементів, з яких вона утворена, іншими словами від кількості трикутників, які апроксимують геометрію об'єкта. Тому важливо, щоб ефективний алгоритм геометричного моделювання враховував всі важливі особливості модельованої поверхні, і при цьому генерував мінімальну кількість трикутних елементів.

Одним із геометричних параметрів поверхні, відповідно до якого можна розраховувати кількість трикутних елементів (загущувати або спрощувати сітку) є кривина. Як відомо, в диференціальній геометрії виділяють гаусову та середню кривину [1], та відповідно розглядають їх диференціальні аналоги [2]. Тому розробка алгоритмів дискретного моделювання поверхонь із врахуванням їх кривини та відповідна програмна реалізація є актуальною інженерною задачею.

Аналіз останніх досліджень і публікацій. У роботах [3–5] запропоновано алгоритми загущення та спрощення дискретних моделей поверхонь залежно від значень дискретних аналогів кривини. Розрахунок дискретних аналогів гаусової та середньої кривини проводиться за методиками, описаними в [2]. В даних роботах множина трикутних комірок, які утворюють модель, описується за допомогою ієрархічних структур даних — квадродерев та BSP-дерев [6]. Таке представлення сприяє подальшій програмній реалізації пропонованих алгоритмів. В цілому, вищеназвані алгоритми можуть бути адаптовані для перетворення тривимірних моделей з метою їх спрощення без втрати точності.

Виклад основного матеріалу. З метою практичної реалізацій алгоритмів загущення та спрощення дискретних моделей поверхонь, які були описані в [3, 4], було розроблено прикладне програмне забезпечення. Структура програми наведена на рис.1.

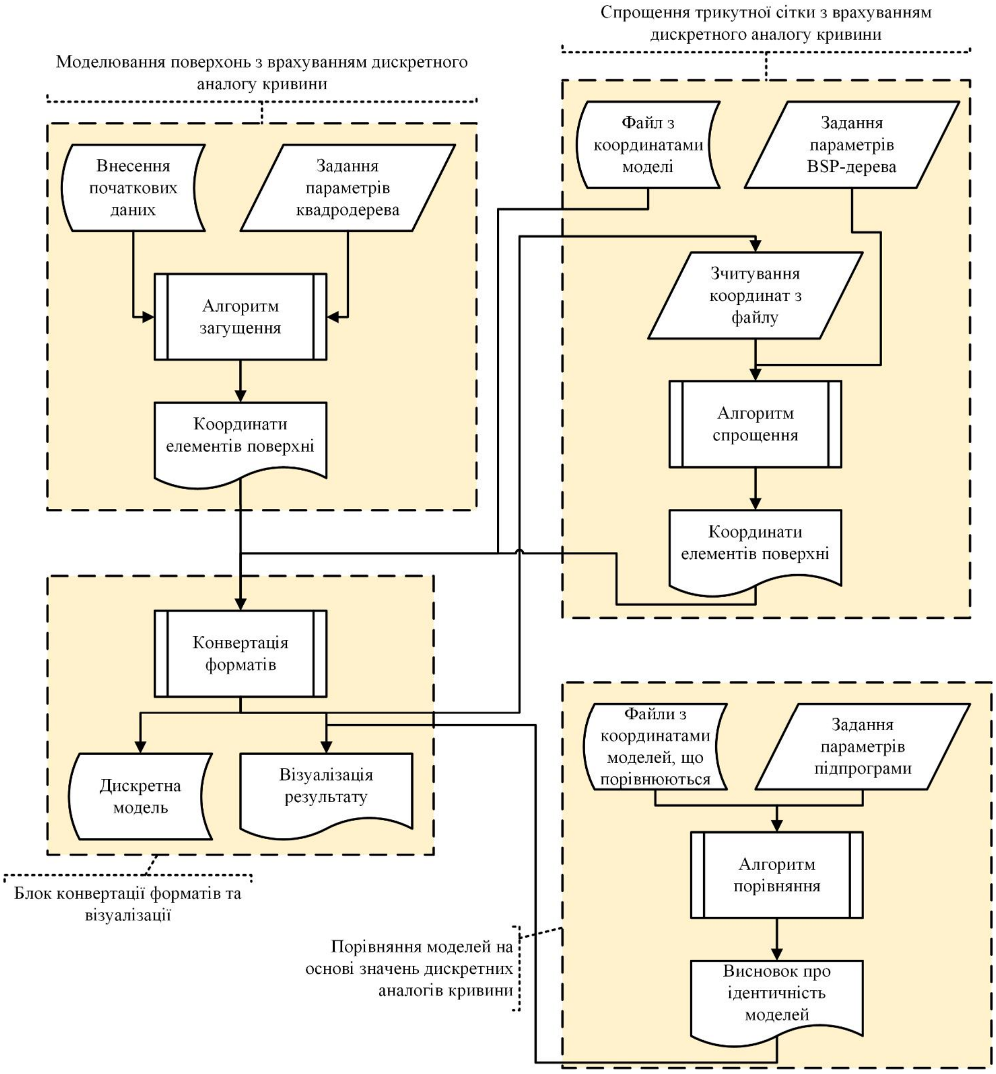


Рис. 1. Структура програмного забезпечення для моделювання поверхонь із врахуванням дискретних аналогів кривини

Для написання програмного забезпечення була використана високорівнева мова програмування Python [7]. Цей вибір пояснюється наявністю великої кількості функцій в стандартній бібліотеці, мінімалістичністю програмного коду, а також кросплатформенністю [8] як самої мови програмування, так і середовищ розробки та готового програмного забезпечення. Наявність спеціалізованих бібліотек, таких як numpy, орієнтованих на роботу з математичними функціями та об'єктами, дозволяє застосовувати Python у якості повноцінного математичного пакету.

Програма складається з чотирьох обчислювальних блоків, які реалізують виконання комплексних алгоритмів моделювання:

1. Блок моделювання поверхонь з врахуванням дискретного аналогу кривини — для моделювання поверхонь, що задані у формі замкнутої числової послідовності (функції від дискретних паремтрів). У якості алгоритму для загущення сітки використовується описаний у [3] алгоритм моделювання.

2. Блок спрощення сітки з трикутними комірками із врахуванням дискретного аналогу кривини — використовується для зменшення густоти сітки, отриманої шляхом сканування тривимірних об'єктів, на плоских ділянках моделі. Реалізує алгоритм, запропонований у [4].

3. Блок порівняння дискретних моделей — призначений для аналізу і порівняння дискретних моделей на основі значень дискретних аналогів кривини у відповідних областях. Частково базується на алгоритмі побудови BSP-дерева, описаному в [4].

4. Блок конвертації форматів та візуалізації результату — допоміжний блок, який призначений для взаємної конвертації форматів mesh, DXF та STL, із яких програма отримує координати елементів дискретної моделі та в які експортуються результати роботи програми. Також блок дозволяє візуально представити одержані результати.

З точки зору користувача робота з програмою складається із трьох етапів: 1) вибір алгоритму який відповідає поставленій задачі моделювання; 2) задання початкових параметрів роботи програми залежно від обраного алгоритму; 3) одержання готової дискретної моделі поверхні у вигляді файлу формату mesh, dxf та stl, візуалізація результату роботи програми.

За своєю суттю вихідні початкові параметри алгоритмів можна класифікувати на дві групи:

1. Геометричні параметри — містять інформацію на основі якої можуть бути отримані або обчислені дані про геометрію об'єкта моделювання: координати вузлів, топологія сітки, значення дискретних аналогів кривини тощо.

2. Параметри підпрограми — допоміжні параметри, які необхідні для коректної роботи відповідного обчислювального блоку. Залежать від алгоритму, який лежить в основі підпрограми.

Розглянемо детальніше специфіку роботи кожного обчислювального блоку програми.

Блок моделювання поверхонь із врахуванням дискретного аналогу кривини (рис. 20) на вхід приймає аналітичний опис поверхні у вигляді відповідної числової послідовності.

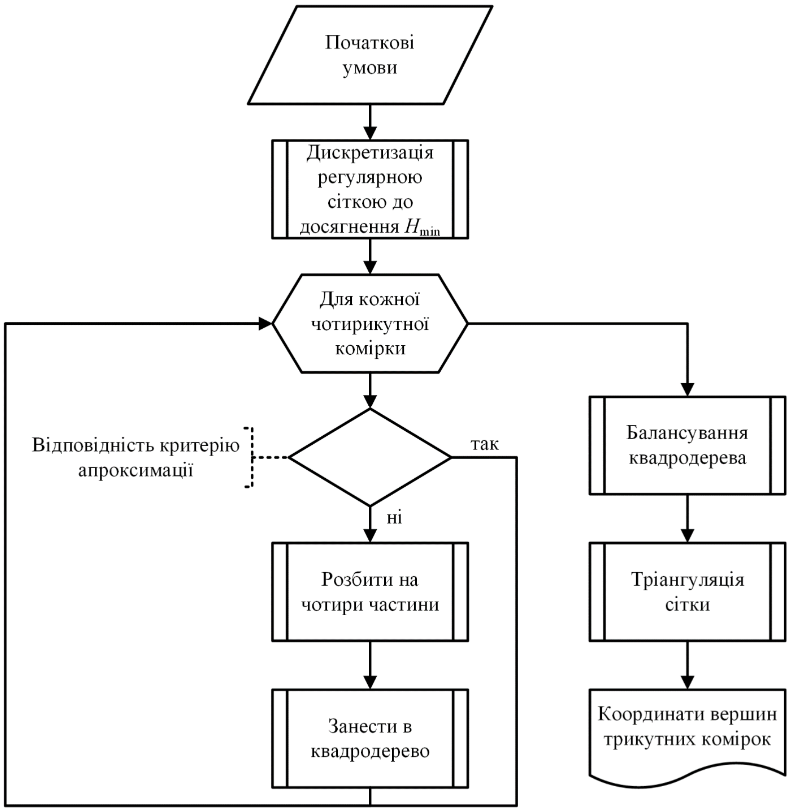


Рис. 2. Послідовність виконання операцій алгоритму моделювання поверхонь із врахуванням дискретного аналогу кривини

У ролі критерію апроксимації може виступати описаний у [9] перепад значення кривини в локальній області.

Залежно від типу поверхні можуть буди вказані наступні геометричні параметри:

1) поверхня, що задається як функція від двох аргументів:

— подвійна числова послідовність $z_{i,j}$, за якою обчислюється апліката вузла, $x = i$, $y = j$;

— області зміни дискретних параметрів i та j;

2) поверхня, що задається параметрично:

— три подвійні числові послідовності $x_{i,j}$, $y_{i,j}$ та $z_{i,j}$, за якими обчислюються координати вузлів;

— області зміни дискретних параметрів i та j;

3) поверхня Безьє [10]:

— матриця контрольних точок $P_{i,j}$;

4) NURBS-поверхня:

— матриця контрольних точок $P_{i,j}$ та вагових коефіцієнтів $\{w_{i,j}\}$;

— вузлові вектори $\{U\},\{V\}$.

Також необхідно задати ряд параметрів підпрограми:

1) $H_{\min}$ — мінімальна висота квадродерева. Задається з метою забезпечення алгоритму деякою множиною вихідних даних для розрахунку значень дискретних аналогів кривини. До досягнення висоти $H_{\min}$ дерево завжди буде повним, а це, в свою чергу, означає, що сітка є регулярною. Після перевищення висотою значення $H_{\min}$ буде відбуватися локальне загущення сітки з врахуванням значення кривини. За замовчуванням $H_{\min}=3$, що відповідає $4^3=64$ чотирикутним коміркам, або сітці розмірністю 8×8. Рекомендується не зменшувати дане значення для коректної роботи алгоритму.

2) $H_{\max}$ — максимально допустима висота дерева. Задається для обмеження процесу загущення сітки та уникнення ситуації зациклення алгоритму. За замовчуванням $H_{\max}=8$, це значення можна змінювати залежно від бажаної густоти сітки та продуктивності обчислювальної системи.

3) вид дискретного аналогу кривини (гаусова або середня), відповідно до якого здійснюється загущення сітки.

4) R — коефіцієнт чутливості алгоритму. Визначає наскільки чутливим до перепаду значень кривини буде алгоритм. За замовчуванням $R=0,3$. Було емпірично встановлено, що рекомендовані значення коефіцієнта лежать в діапазоні $R\in[0,2;0,5]$.

На виході алгоритму отримуємо множину координат вузлів трикутних комірок, яку можна конвертувати в один з підтримуваних форматів файлів.

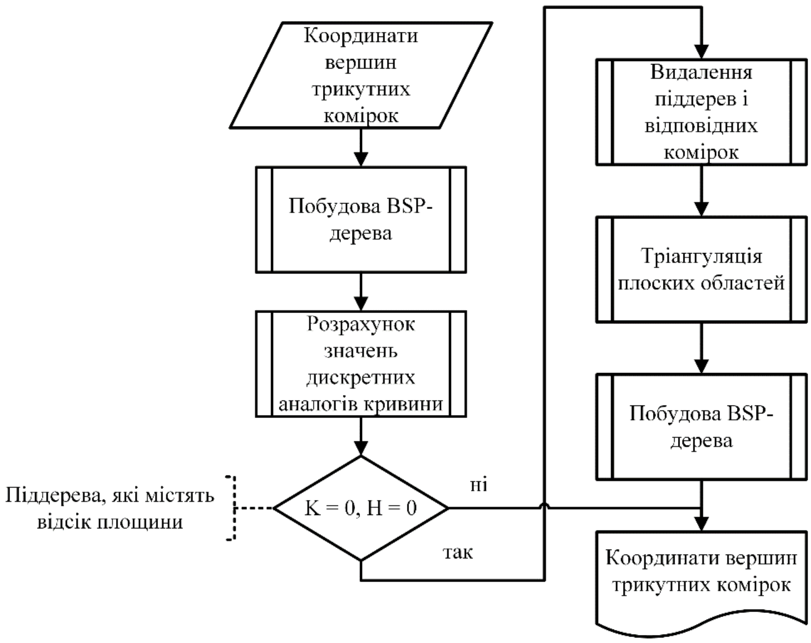


Рис. 3. Послідовність виконання операцій алгоритму спрощення дискретних моделей

Тепер розглянемо специфіку роботи алгоритму спрощення дискретних моделей (рис. 30). На вхід алгоритму надходить інформація про координати вузлів трикутних комірок. Кожна комірка представляється у вигляді двовимірного масиву $\left[[x_1,y_1,z_1],[x_2,y_2,z_2],[x_3,y_3,z_3],[n_x,n_y,n_z]\right]$, де $x_1,y_1,z_1,x_2,y_2,z_2,x_3,y_3,z_3$ — координати вузлів трикутної комірки, n_x,n_y,n_z — координати вектора нормалі.

Множина координат вершин трикутників отримується шляхом перетворення підтримуваного формату файлу блоком конвертації. Оскільки це єдине джерело інформації про геометрію моделі, то вона відповідно є єдиним геометричним параметром алгоритму. Що стосується параметрів підпрограми, то вони відсутні. Координати січних площин, значення кривини і топологія BSP-дерева встановлюються в ході роботи алгоритму.

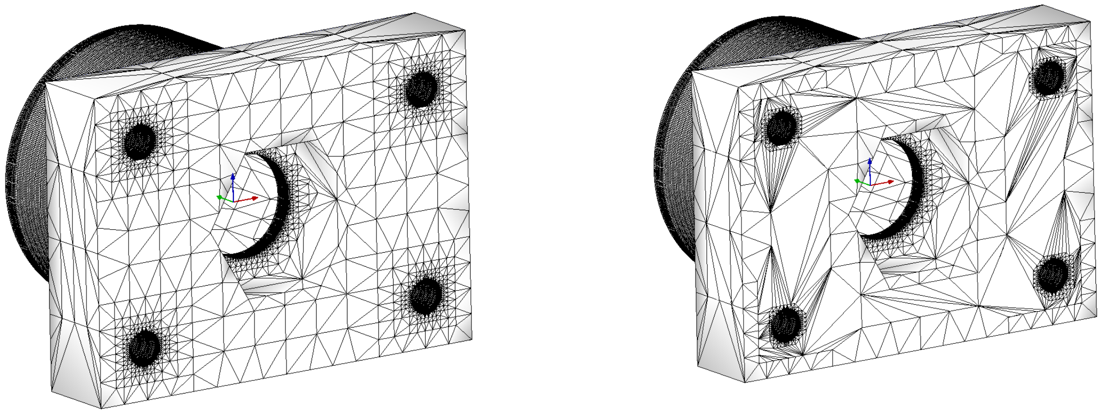
На виході обчислювального блоку одержуємо множину координат вершин трикутників, які за допомогою блоку конвертації форматів перетворюються на бажаний формат файлу.

Блок порівняння дискретних моделей базується на алгоритмі побудови BSP-дерева з рядом суттєвих відмінностей. Його призначення та принцип роботи є предметом окремої публікації.

Блок конвертації форматів та візуалізації підтримує багатосторонню конвертацію форматів файлів mesh, dxf, stl, та здійснює підготовку координат до вигляду, зручного для роботи програми. Оскільки він виконується в ході роботи інших обчислювальних блоків, то, відповідно, для кожного з них необхідно вказати формат вхідного файлу та бажаний формат вихідного файлу з координатами моделі.

Хоча блок конвертації підтримує роботу з трьома типами файлів моделей, але основним із них є STL. Це пояснюється його поширеністю у якості способу представлення, передачі та обміну інформацією про геометрію моделі.

Приклад спрощення тривимірної моделі в результаті виконання алгоритму представлений на рис.4.



До застосування алгоритму *Після застосування алгоритму*

Рис. 4. Приклад спрощення сітки в результаті роботи алгоритму

Висновки та перспективи подальших досліджень. Подальші дослідження більшою мірою можуть стосуватися підвищення швидкодії програмного забезпечення. Це може стосуватися як оптимізації обчислювальних алгоритмів, так і застосування продуктивніших програмних рішень. Так доцільно деякі фрагменти програмного коду переписати з використанням C++, що може суттєво покращити швидкодію кінцевого програмного забезпечення.

1. Александров А.Д. Геометрия: Учебное пособие / А.Д. Александров, Н.Ю. Нецветаев. – М.: Наука, 1990. – 671 с.

2. Meyer M. Discrete Differential Geometry Operators for Triangulated 2-Manifolds / M. Meyer, M. Desbrun, P. Schröder, A. Barr // Visualization and Mathematics III. – 2003. – Вип. 3. – с. 35–57.

3. Ройко О.Ю. Використання квадродерев при загущенні сітки з трикутними комірками / О.Ю. Ройко // Сучасні технології в машинобудуванні та транспорті. Науковий журнал. – 2014.

4. Ройко О.Ю. Використання бінарного розбиття простору в алгоритмі спрощення тривимірних моделей для швидкого прототипування / О.Ю. Ройко, І.Н. Бурчак, В.Л. Величко // Комп’ютерно-інтегровані технології: освіта, наука, виробництво. – Луцьк: ЛНТУ, 2015. – Вип. 19. – с. 142–145.

5. Ройко А.Ю. Методика обеспечения сбалансированости квадродерева / А.Ю. Ройко // Сборник статей Международной научно-практической конференции “Наука и образование XXI века.” – Уфа: 2013. – с. 184–187.

6. Zachmann G. Geometric data structures for computer graphics / G. Zachmann, E. Langetepe. – Eurographics Assoc., 2003. – 36 с.

7. Лутц М. Изучаем Python / М. Лутц. – М.: Символ-Плюс, 2011. – 1272 с.

8. Слива М.В. Кроссплатформенный подход как средство унификации обучения программированию в различных операционных системах / М.В. Слива // Прикладная информатика / М., ООО “Синергия ПРЕСС.” – Litres, 2012. – Вип. 2. – с. 38–45.

9. Ройко О.Ю. Гаусова кривина дискретно заданих поверхонь як критерій для загущення сітки / О.Ю. Ройко // Комп’ютерно-інтегровані технології: освіта, наука, виробництво. – Луцьк: ЛНТУ, 2011. – Вип. 6. – с. 118–122.

10. Роджерс Д. Математические основы машинной графики / Д. Роджерс, Д. Адамс. – М.: Мир, 2001. – 604 с.