УДК 004.94
[1]Ройко О.Ю., [2]Бурчак І.Н., [2]Величко В.Л.
[1]Волинський технікум НУХТ.
[2]Луцький національний технічний університет.

ВИКОРИСТАННЯ БІНАРНОГО РОЗБИТТЯ ПРОСТОРУ В АЛГОРИТМІ СПРОЩЕННЯ ТРИВИМІРНИХ МОДЕЛЕЙ ДЛЯ ШВИДКОГО ПРОТОТИПУВАННЯ

Ройко О.Ю., Бурчак І.Н., Величко В.Л. Використання бінарного розбиття простору в алгоритмі спрощення тривимірних моделей для швидкого прототипування. В статті розглядаються особливості застосування бінарного розбиття простору для систематизації невпорядкованої множини трикутних комірок, якими утворена зовнішня поверхня тривимірного об’єкту. Це дає змогу побудувати алгоритм спрощення сітки з трикутними комірками, спираючись на значення дискретних аналогів гаусової та середньої кривини у вузлах сітки.
Ключові слова: BSP-дерева, сітка з трикутними комірками, алгоритм спрощення, швидке прототипування, алгоритм тріангуляції.

Ройко А.Ю., Бурчак И.Н., Величко В.Л. Использование бинарного разбиения пространства в алгоритме упрощения трехмерных моделей для быстрого прототипирования. В статье рассматриваются особенности применения бинарного разбиения пространства для систематизации неупорядоченого множества треугольных ячеек, которыми образована внешняя поверхность трехмерного объекта. Это дает возможность построить алгоритм упрощения сетки с треугольными ячеек, опираясь на значение дискретных аналогов гаусовой и средней кривзины в узлах сетки.
Ключевые слова: BSP-деревья, сетка с треугольными ячейками, алгоритм упрощения, быстрое прототипирование, алгоритм триангуляции.

Royko O., Burchak I., Velychko V. Using binary space partitioning in algorithm for simplification the three-dimensional models for rapid prototyping. In the article considered features of application of binary space partitioning for organizing disordered set of triangular cells that formed the outer surface of a three-dimensional object. This allows to build a mesh simplification algorithm of triangular mesh based on the values of discrete analogues Gaussian and mean curvatures in the nodes.
Keywords: BSP-trees, mesh with triangular cells, simplification algorithm, rapid prototyping, triangulation algorithm.

Постановка проблеми. На практиці широко застосовуються тривимірні моделі, одержані способом сканування реального фізичного об’єкту сканером, або моделі, створені за допомогою програмного забезпечення для тривимірного моделювання. В усіх цих випадках модель являє собою сукупність координат дискретних об’єктів — вузлів, комірок, зв’язків. Однією з особливостей готових геометричних моделей є невпорядкованість та неструктурованість даних про модель. В результаті цього пошук суміжних або інцидентних елементів, їх додавання та вилучення, може представляти деякі труднощі, особливо при значних обсягах даних. Для моделей, одержаних шляхом сканування, важливим питанням є спрощення сітки в областях, де вона загущена надлишково. Очевидно, що в першу чергу це стосується областей, які є відсіками площин. В інженерній практиці дуже часто доводиться стикатися зі складальними одиницями та деталями, поверхня яких частково або повністю складається саме з відсіків площин. Спрощення сітки на плоских ділянках і видалення надлишкових даних щодо них може суттєво зменшити обсяг інформації про геометричну модель та сприяти підвищенню швидкості її обробки. Для виявлення плоских ділянок доцільно застосовувати значення дискретних аналогів гаусової та середньої кривини у вузлах сітки. Можна підсумувати, що побудова алгоритму спрощення моделей тривимірних об’єктів є актуальною проблемою і має широке практичне застосування.

Аналіз останніх досліджень і публікацій. Застосування ієрархічних структур даних для побудови різноманітних алгоритмів комп’ютерного моделювання розглянуто в роботі [3]. Методики для визначення дискретних аналогів гаусової та середньої кривини у вузлах сітки з трикутними комірками, а також їх порівняння, наведені в [4].

Невирішені частини проблеми. Алгоритм спрощення дискретних моделей поверхонь на плоских ділянках повинен вирішувати наступні завдання: 1) впорядковувати та структурувати невпорядковану множину трикутних комірок, яка є моделлю тривимірного об’єкта; 2) виявляти області, які є відсіками площин; 3) на виявлених областях наносити сітку, яка утворена комірками більшого розміру, і таким чином зменшувати кількість комірок. Перспективним підходом для вирішення цих завдань є застосування ієрархічних структур даних для систематизації інформації про елементи сітки.

Метою дослідження є побудова методики та алгоритму спрощення сітки, що являє собою модель поверхні тривимірного об'єкта, в областях, які є відсіками площин.

Основні результати дослідження. Для структуризації та систематизації інформації про дискретні моделі об'єктів, що являють собою множину трикутних комірок, можна застосувати ієрархічні структури даних, зокрема BSP-дерева (BSP — Binary Space Partition, двійкове або бінарне розбиття простору). Вони переважно застосовуються для виявлення зіткнень між об'єктами або для впорядкування об'єктів тривимірної сцени в порядку віддалення від користувача [1].

З точки зору представлення інформації про дискретну модель поверхні важливою є властивість BSP-дерева впорядковувати об'єкти. З кожним не листовим вузлом дерева можна зв'язати деяку пряму на площині або площину у просторі, яка ділить множину об'єктів на дві частини. Для подальшого впорядкування необхідно рекурсивно повторити описані дії для кожної з частин (рис. 1).

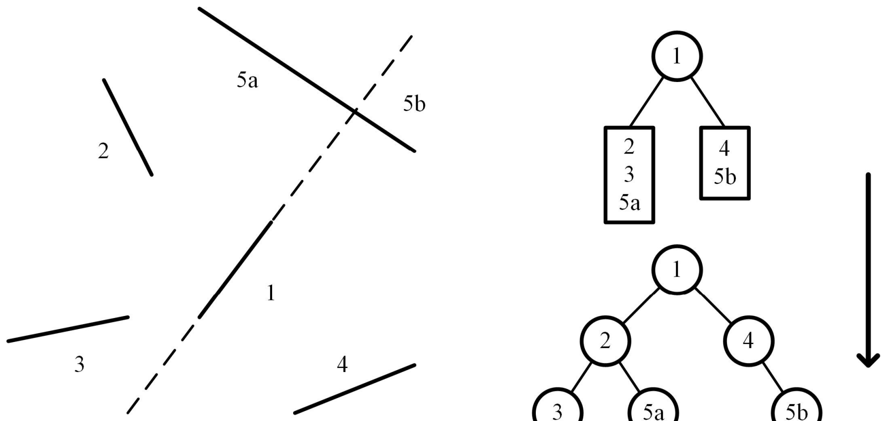

Рис. 1. **Впорядкування множини відрізків за допомогою BSP-дерева**
авторська розробка

Ця властивість дає змогу отримати систематизовану структуру даних із множини трикутників. Можна сформулювати алгоритм, який на вхід отримує дискретну модель поверхні і на виході повертає BSP-дерево, листовими вузлами якого є комірки сітки [2]:

1. Створити пусте дерево BSP.
2. Обрати два напрями координатних осей, вздовж яких модель має найбільшу протяжність.
3. Обрати січну площину Ω, таку, яка перетинає вісь, вздовж якої модель має найбільшу протяжність. Координати Ω занести в BSP.
4. Створити два піддерева BSP_1 та BSP_2.
5. Керуючись значеннями, отриманими при підстановці координат вузлів у рівняння площини Ω визначити орієнтацію комірок відносно січної площини та занести їх у відповідне дерево.
6. Якщо T_1 та T_2 містять по одній комірці, то занести її координати в листовий вузол та завершити алгоритм.
7. Повторити пункти 2-6 рекурсивно для BSP_1 та BSP_2 при іншій орієнтації січної площини.

Маючи алгоритм побудови BSP-дерева для множини трикутних комірок можна побудувати послідовність спрощення геометричної моделі. Сформулюємо наступну умову: якщо для вузла сітки $\mathbf{x}_i\left(x_i, y_i, z_i\right)$ значення дискретних аналогів кривини

$$H\left(\mathbf{x}_i\right)=0 \text{ та } K\left(\mathbf{x}_i\right)=0, \tag{1}$$

то вважатимемо його таким, що належить площині. Якщо для всіх вузлів комірки T справджується умова (1), то вважатимемо її такою, що апроксимує відсік деякої площини. Якщо для деякого піддерева BSP_i листові вузли відповідають трикутним коміркам, що апроксимують деяку площину, то BSP_i вважатимемо таким, що описує відсік площини.

Таким чином методика виявлення плоских ділянок на поверхні моделі наступна:

1. У сформованому BSP-дереві для всіх піддерев BSP_i нижнього рівня висотою $h \geq 3$, які містять принаймні 6 листових вузлів визначити значення дискретних аналогів кривини.

2. Якщо BSP_i і всі його елементи відповідають умові (1), то помітити його корінь, як такий, що апроксимує відсік площини. Інакше помітити його корінь і всі батьківські вузли як такі, що не апроксимують площини.

Останнім етапом комплексного алгоритму спрощення тривимірних моделей об'єктів з врахуванням дискретного аналогу кривини є вилучення сітки на плоских ділянках поверхні моделі та нанесення нової сітки, утвореної більшими елементами. Таким чином загальна кількість трикутних комірок зменшиться, а отже і зменшиться інформаційний обсяг моделі.

Задача нанесення трикутної сітки на областях, де вона була вилучена, зводиться до відомої задачі тріангуляції багатокутника (рис. 3). Зокрема, це завдання можна вирішити, застосувавши вушний або монотонний методи тріангуляції [3]. Після виконання тріангуляції плоскої багатокутної області для нової множини трикутних комірок необхідно побудувати нове BSP-піддерево за описаною вище методикою, і результат приєднати до того вузла дерева, який відповідає даному багатокутнику. Таким чином, можна сформулювати остаточний комплексний алгоритм спрощення заданої сітки з трикутними комірками із врахуванням дискретного аналогу кривини:

1. Для заданої множини трикутних комірок будується BSP-дерево, яким індексуються елементи моделі.

2. Для комірок, що містяться у піддеревах побудованого BSP-дерева розраховуються значення дискретних аналогів гаусової та середньої кривини. Якщо обидва значення рівні нулю для деякої множини суміжних комірок, то вони вважаються такими, що апроксимують відсік площини, а піддерево у якому вони містяться, помічається відповідним чином.

3. Піддерева, які помічені як такі, що містять відсік площини видаляються з дерева разом з відповідними трикутними комірками.

4. Виконується тріангуляція плоских багатокутних областей, утворених в результаті видалення піддерев.

5. Для новоутворених трикутних комірок повторюється пункт 1 і результат заноситься в існуюче дерево. Утворене BSP-піддерево помічається як таке, що апроксимує відсік площини.

Результат виконання алгоритму спрощення сітки із врахуванням дискретного аналогу кривини представлений на рис. 2. Як видно з даного прикладу спрощення сітки відбувається лише на областях, що являють собою плоскі ділянки. Області, які представляють собою циліндричні отвори, пази тощо ігноруються.

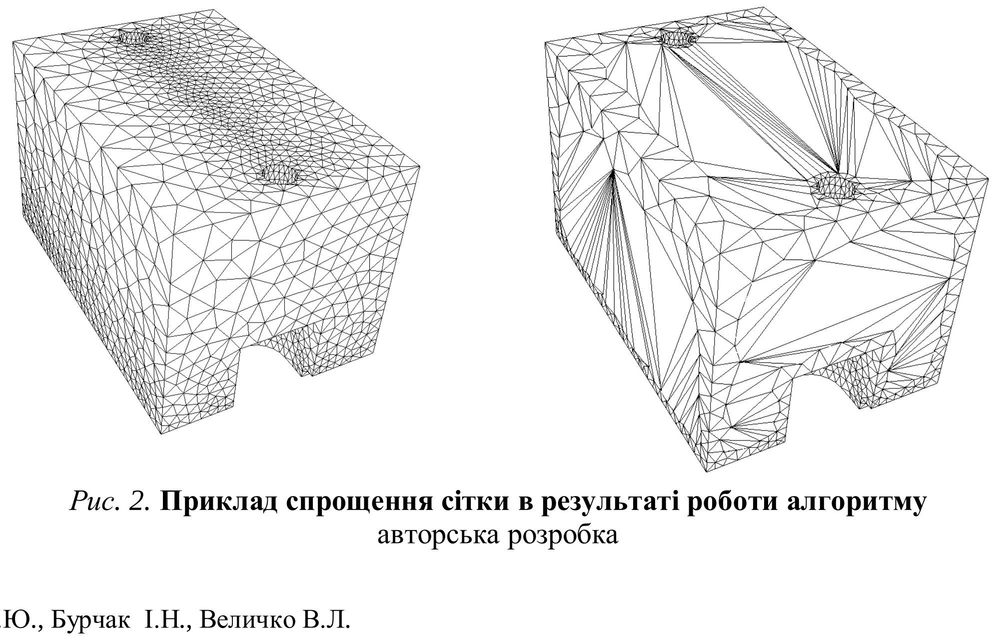

Рис. 2. **Приклад спрощення сітки в результаті роботи алгоритму**
авторська розробка

Очевидно, що ефективність алгоритму, а саме співвідношення кількості комірок в моделі після і до роботи алгоритму, залежить в першу чергу від кількості і розміру плоских ділянок на поверхні. Тому для різних моделей ефективність алгоритму спрощення може суттєво відрізнятися.

В результаті аналізу обчислювальної складності алгоритму було встановлено, що він має порядок $O\left(N\log_2 N\right)$, де N - кількість трикутних комірок, поданих на вхід алгоритму. При цьому найбільш затратним етапом виконання алгоритму спрощення з точки зору обчислювальних ресурсів є побудова BSP-дерева для всієї моделі. Очевидно, що суттєве збільшення складності моделей, а отже і зростання кількості вхідних даних, може негативно позначитись на часові виконання програми.

Порядок росту $O\left(N\log_2 N\right)$ означає, що існують додатні константи c та N_0, такі, що для всіх $N \geq N_0$ виконується нерівність

$$T\left(N\right) < cN\log_2 N\,, \qquad (2)$$

де $T\left(N\right)$ — час виконання алгоритму. Графічно значення функції $T\left(N\right)$ лежать під кривими (рис. 3). На даному графіку криві побудовані для значень $c = \{1, 2, 4, 8\}$. Із рисунку видно, що при великих значеннях N функція $T\left(N\right)$, має характер близький до лінійного.

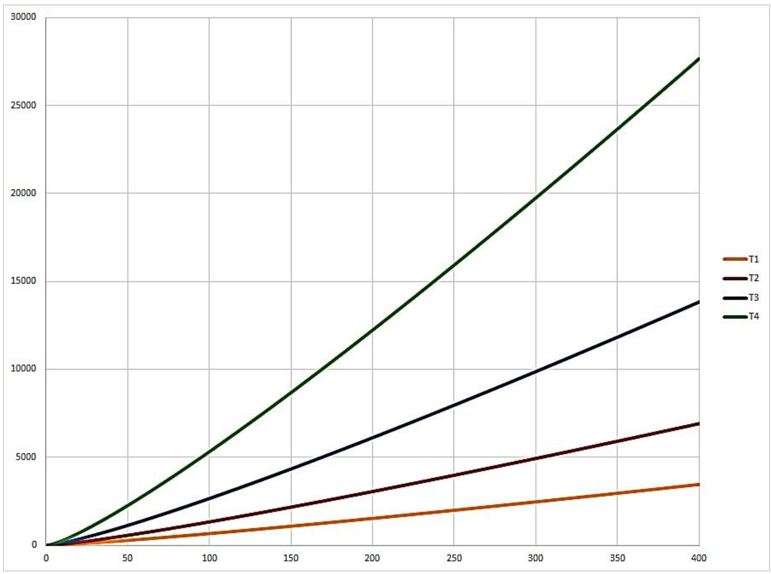

Рис. 3. **Графічна залежність часу виконання алгоритму спрощення від кількості вхідних даних**
авторська розробка

Висновки. Розроблений алгоритм спрощення з використанням бінарного розбиття простору має широкі перспективи застосування, враховуючи появу та поширення недорогих пристроїв для тривимірного друку. В подальшому варто розглянути можливість застосування інших структур даних для впорядкування множини комірок, оскільки використання BSP-дерев ставить значні вимоги до продуктивності обчислювального пристрою при спрощенні складних моделей.

1. Ласло М. Вычислительная геометрия и компьютерная графика на С++ / М. Ласло. – М: Бином, 1997. – 301 с.
2. Ройко О.Ю. Використання BSP-дерев для представлення інформації про дискретну модель поверхні / О.Ю. Ройко // Актуальні задачі сучасних технологій: зб. тез доповідей міжнар. наук.-техн. конф. – Тернопіль: ТНТУ, 2014. – с. 215–216.
3. Goodman J. та ін. Handbook of discrete and computational geometry / J. Goodman, J. O’Rourke. – New York: CRC press, 2010. – 1558 с.
4. Meyer M. та ін. Discrete differential-geometry operators for triangulated 2-manifolds / M. Meyer, M. Desbrun, P. Schröder, A. Barr // Visualization and mathematics. – 2003. – Вип. 3. – с. 35–57.