УДК 004.92

Т.А. Матвейчук, О.О. Лисак [1]
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин,
[1] кафедра інформаційних технологій видавничої справи

АНАЛІЗ МЕТОДІВ СПРОЩЕННЯ ПОЛІГОНАЛЬНИХ МОДЕЛЕЙ 3D-ОБ’ЄКТІВ

Проаналізовано основні ітераційні алгоритми спрощення 3D-полігональних моделей об’єктів з метою скорочення об’єму даних, що приводить до підвищення швидкості процесів їх передачі та візуалізації.
Ключові слова: алгоритм, полігон, візуалізація.

The paper analyzes the basic iterative algorithms simplify 3D polygonal models of objects in order to reduce the amount of data which leads to increased speed processes of transmission and visualization.
Key words: algorithm, polygon, rendering.

Вступ

Використання тривимірної комп’ютерної графіки дає змогу за допомогою комп’ютера якісно змінити процес сприйняття інформації. Поява 3D комп’ютерної графіки була пов’язана передусім з процесом проектування, коли візуалізація ідей давала можливість бачити проблему і знаходити способи її вирішення. Подальше її застосування охоплює практично всі сфери людської діяльності. Особливо великі перспективи відкриваються в області 3D інтернет-технологій. Сучасні тривимірні інтернет-технології вже сьогодні застосовуються в багатьох сферах людської діяльності, таких як промисловість, САПР, різні галузі медицини, освіта, наукові дослідження, а також архітектура, транспорт, тренажери, системи віртуальної реальності, презентаційні системи, електронний туризм, розважальні системи тощо.

Однією з основних проблем передавання і відображення тривимірних даних є передача великих об’ємів даних, необхідних для опису складних тривимірних моделей, в умовах обмеження пропускної здатності каналів передачі даних, таких як Інтернет, космічний зв’язок тощо. Спрощення складних 3D-об’єктів є важливим напрямом досліджень в комп’ютерній графіці. Зниження рівнів деталізації Level of Detail (LOD) спрощує не тільки задачу передачі даних, але і задачу візуалізації.

Постановка проблеми

Побудова 3D-об’єктів у комп’ютерній графіці здійснюється в більшості систем на основі полігонів. Багатокутники, що утворюють поверхню об’єкта, є гранями, сторони граней – ребрами, точки перетину ребер – вершинами. Для ефективної роботи з полігональними об’єктами необхідна така структура даних, щоб вона відображала зв’язки між вершинами і багатокутниками в мережі. Запропоновано структуру 3D-об’єкта, утвореного полігонами, зберігати у вигляді списку, де півтора зв’язка.

Чим більша кількість полігонів використовується для побудови об’єкта, тим детальніше він буде представлений, і тим самим буде забезпечена більша реалістичність зображення. Для зниження обчислювальних витрат можна зменшити кількість полігонів, що утворюють полігональну поверхню. Цей процес прийнято називати спрощенням. Подібно до більшості завдань спрощення, тут також виникає проблема вибору між якістю представлення 3D-моделей і швидкодією. Критерієм ефективності прийнято швидкодію.

Отже, задачу спрощення полігональних моделей можна сформулювати так: необхідно зменшити кількість граней, що описують модель, зі збереженням (або з мінімальною зміною) топології. Це призводить до збільшення швидкодії алгоритмів прорахунку зображення.

Ітераційні методи спрощення

Проаналізуємо стан досліджень в області спрощення полігональних моделей.

Розв'язання цієї задачі можна віднести до класу оптимізаційних задач. У таких задачах може існувати множина різних рішень. Їх «якість» визначається значенням деякого параметра, і потрібно вибрати оптимальне рішення, за якого значення параметра буде мінімальним або максимальним (залежно від постановки задачі).

Більшість ітераційних алгоритмів спрощення полігональних моделей можна розділити на три категорії:

– проріджування вершин;

– згортання ребра;

– проріджування граней.

Найшвидшим алгоритмом геометричного спрощення, що належить до категорії *проріджування вершин*, є алгоритм кластеризації вершин, що запропонував J.Rossinac [7]. Метод полягає в об'єднанні вершин, що містяться в одному кластері простору, в одну. Кластери являють собою тривимірну сітку, тобто координати вершин фактично заокруглюються із заданою точністю, а у разі збігу кількох вершин, після заокруглення, ці вершини замінюються однією (рис. 1).

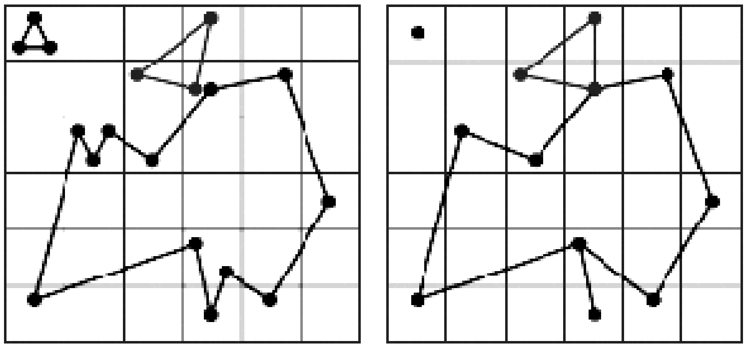

Рис.1. Етапи виконання алгоритму кластеризації вершин

Перевагою цього методу є його простота і швидкодія, що дає можливість використовувати його в режимі реального часу. Основними недоліками є перебудова масиву даних і втрата первісної топології об'єкта, що може призвести до серйозних помилок. З іншого боку, якщо відстані між вузлами решітки будуть співмірні з розмірами граней об'єктів, то можна отримати модель, що практично не відрізнятиметься від оригіналу. Основною областю застосування алгоритму кластеризації вершин є задачі, пов'язані з візуалізацією у режимі реального часу.

Алгоритми, побудовані на принципі *проріджування вершин*, описано в роботах [3, 8]. У цьому випадку видаляється одна з вершин полігональної сітки. Разом з видаленням вершини віддаляються грані, яким належить ця вершина, і утворюється отвір, який необхідно заповнити. Для цього виконується операція тріангуляції. Хоча алгоритм дає непогані результати, виконання операцій тріангуляції вимагає виконання додаткових операцій.

Точніше топологію моделі передає алгоритм проріджування вершин, що запропонував Schroeder [8]. Це багатопрохідний алгоритм, суть якого полягає у видаленні на кожному кроці вершин, розташованих від усередненої площини сусідніх вершин на відстані, меншій від заданої.

Виконуючи прохід алгоритму, знаходять вершини для видалення, при цьому критерієм на видалення є задана величина d (відстань від вершини до усередненої площини, максимально близької до сусідніх). Після видалення вершини утворюється отвір у тривимірному об'єкті, який заповнюється гранями за допомогою операції тріангуляції.

Алгоритми, основані на *згортанні ребер*, можна розглядати як окремий випадок видалення вершин, проте вони не потребують виконання додаткових операцій тріангуляції. Тому їх реалізація дає хороші результати з погляду якості та швидкодії.

Згортання ребра являє собою операцію злиття двох вершин, що утворюють ребро, в одну (рис. 2). При цьому, в загальному випадку, відбувається видалення двох трикутних комірок. Кількість ітераційних кроків визначається досягненням заданої кількості граней або деякого порогового значення.

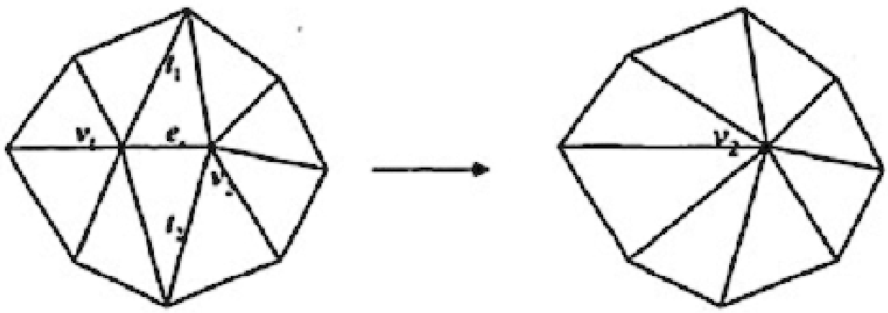

Рис. 2. Процедура згортання ребра e_r. Вершина V_1 переноситься у вершину V_2

На кожному кроці реалізації алгоритму вирішуються дві проблеми:

1. Які пріоритети вибрати, щоб отримати оптимальне рішення.

2. Де розташовуватиметься вершина, в яку будуть перенесені вершини ребра, що стягується.

Найпростіша стратегія – використовувати одну з вершин ребра, що згортається, як вершину заміни. Щоб отримати точніше рішення потрібно, щоб вершина заміни не належала до вершин вихідної сітки.

Один з варіантів – організувати згортання ребра в деяку точку, що можна знайти як середнє арифметичне координат цих вершин (рис. 3). Тим самим можна мінімізувати помилку. Однак у цьому випадку відбуватиметься перестроювання сітки щодо вихідної, що ускладнить відновлення первісної форми моделі. У процесі передачі даних, що підвищують рівень деталізації, доведеться виконувати пересортовування масиву даних.

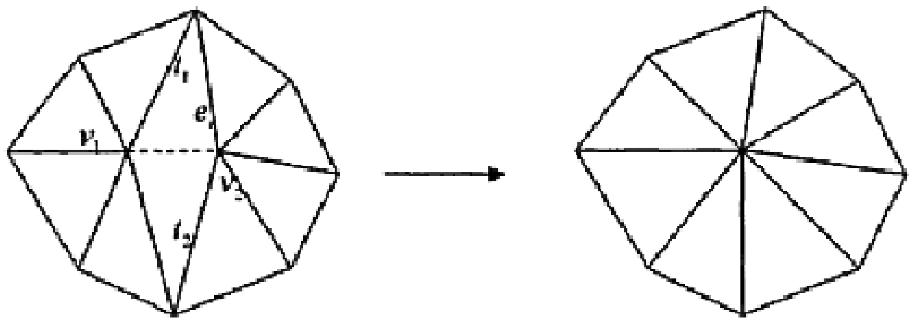

Рис. 3. Точка згортання, знайдена як середнє арифметичне координат вершин v_1 і v_2.

Послідовність кроків, що виконуються при операціях згортання ребер, визначається деякою мірою помилки, що відображає локальне геометричне відхилення комірок. Вибір шляху визначення цієї помилки визначає різні алгоритми цього класу.

Алгоритм H. Hoppe’a [5] – один з перших у цьому класі. У ньому використовується міра помилки, що визначається як середня відстань між новими трикутниками і типовими точками на вихідній моделі. Алгоритм забезпечує спрощення високої якості, проте багаторазові виміри відстаней до поверхні уповільнюють його роботу.

В алгоритмі, що розробив A. Gueziec [4], визначається обсяг допуску, як опукла комбінація сфер, розташованих в кожній вершині спрощення. Вибір граней ґрунтується на найменшій довжині, а нова вершина позиціонується так, що первісна поверхня гарантовано лежатиме в межах цього обсягу. Цей алгоритм також дає результати хорошої якості, і, за оцінками [9], працює швидше від попереднього.

В алгоритмі Kobbelt’a [6] підтримується зв’язок між точками вихідної моделі і відповідними околицями на спрощеній моделі. Це вимагає додаткових обчислювальних ресурсів і великого об’єму пам’яті.

Описані вище методи спрощення приводять до такого скорочення вихідної множини V_0, що множина V_c є її підмножиною V_c є V_0. Такі методи мають високу швидкість, оскільки, незалежно від критеріїв вибору вершин, що видаляються, проводять тільки дві операції: видалення вершини і тріангуляцію отриманого отвору.

Алгоритми, що повністю перебудовують V_0, ресурсоємніші, але дають точніші результати. Метод перебудови полігональної сітки, що запропонував G.Turk [10], спочатку розроблявся для медичних цілей для спрощення гладких поверхонь, які не мають розривів першої похідної. Суть методу полягає в повній перебудові моделі M_0 на модель M_c з меншою кількістю трикутних граней. Для цього на поверхні моделі M_0 довільно розміщують m_c вершин (де m_c – кількість вершин оптимізованої моделі, що задає користувач). Потім на базі множини V_c і первісної геометрії будується множина K_c.

Завдяки такому підходу до отримання нових граней топологія спрощеної моделі достатньо точно повторює топологію первісного об’єкта, але якщо в первісній поверхні були розриви першої похідної (гострі кути і грані), то ці фрагменти при оптимізації будуть передані з помилкою, що робить метод перебудови моделі застосовним тільки для гладких поверхонь. Хоч цей алгоритм досить складний і комплексний, розглянутий метод не є швидкодіючим, і алгоритм досить вимогливий до ресурсів пам’яті. Метод не рекомендується застосовувати для оптимізації в режимі реального часу.

Програмні реалізації розглянутих методів можна знайти в різних системах САПР і тривимірного моделювання [11]. У програмі тривимірного моделювання 3D MAX є вбудовані модулі Optimize і MultiRez [2]. Також для цього пакета існують модулі сторонніх виробників – PolygonCruncher компанії MooToolsSoftware і PolyChop компанії BioWare. Крім цього, існують самостійні програми, що виконують оптимізацію та спрощення полігональних моделей, наприклад Internet Model Optimizer компанії ParallelGraphics.

Проблеми, які можуть виникнути у разі використання ітераційних спрощень, можна проілюструвати прикладом (рис. 4).

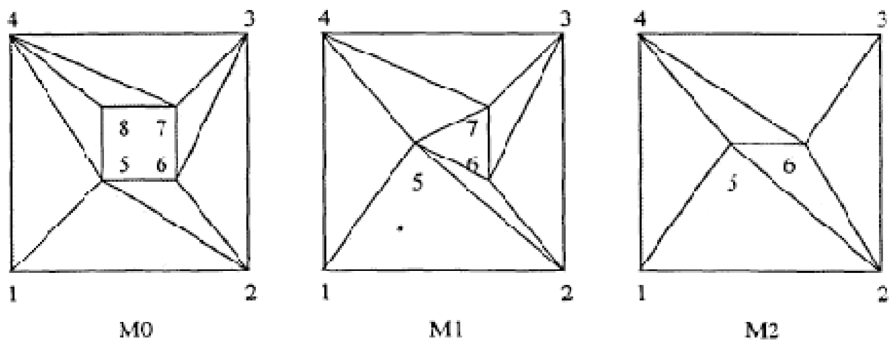

Рис. 4. Спрощення простого плоского об’єкта з порушенням топології

Початкова модель МО – прямокутна поверхня з прямокутним отвором. Вихідна модель містить вісім трикутних граней. Виконуючи згортання ребра між вершинами (V5; V8), отримуємо перше наближення Ml. Після видалення однієї вершини і однієї грані одержимо вже трикутний отвір. Виконавши згортання ребра між вершинами (V6; V7), отримуємо наближення М2, яке має тепер шість граней, але без отвору. На рис. 5 представлена полігональна модель деталі, в якій відбулося порушення топології у результаті застосування операції спрощення (модуль Optimize [2]).

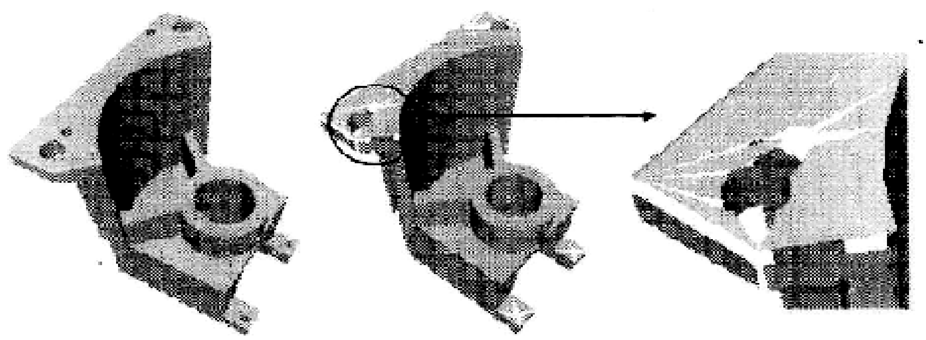

Рис. 5. Порушення топології

Ці приклади ілюструють спрощення, основані на геометричних критеріях, які призвели до зміни топології вихідної моделі. Для прикладних областей, в яких топологічні зміни небажані, алгоритм повинен містити додаткові механізми, що гарантують збереження топології.

Крім розглянутих методів спрощення, можливий принципово інший підхід до реалізації процесу прогресивної передачі даних і, зокрема, тривимірних моделей. Це використання математичного апарату, основаного на використанні вейвлет-перетворень. Один з найяскравіших прикладів застосування вейвлетів у комп'ютерній графіці – це використання їх для стиснення растрових зображень [1].

Висновки

Аналіз технологій передачі тривимірної інформації в Інтернеті показав, що вони не цілком задовольняють вимоги щодо швидкодії, якості та ефективності. Крім цього, процес передачі тривимірних об'єктів стикається з головною проблемою для розробників, що працюють у реальному режимі часу, – обмеженням на пропускну здатність ресурсів Інтернету.

Отже, виділення проблеми тривимірної візуалізації віртуальних об'єктів, що передаються через канали зв'язку, які мають обмеження на пропускну здатність, в окрему теоретичну задачу, має не тільки наукове, але й велике прикладне значення.

1. Сэломон Д. Сжатие данных, изображений и звука. – М.: Техносфера, 2007. – 368 с. 2. Тозик В.Т., Меженин А.В. 3ds max 7: трехмерное моделирование и анимация. – СПб.: БХВ-Петербург, 2005. – 992 с: ил. 3. Cohen J., Varshney A., Manocha D., Turk G., Weber H., Agarwal P., Brooks F. and Wright W. Simplification Envelopes. // In ACM SIGGRAPH 96 Conference Proceedings, – 2001 – P. 119–128. 4. Gueziec A. Surface Simplification with Variable Tolerance // Proc. Second Annual Symp. Medical Robotics and Computer-Assisted Surgery, Wiley and Sons, New York, 1999. – P. 132–139. 5. Hoppe. H. Progressive Meshes. In ACM SIGGRAPH 96 Conference Proceedings, 1996. – P. 99–108. 6. Kobbelt L., Campagna S., and Seidel H.P. A general framework for mesh decimation // in Proc. Graphics Interface '98, 2006. – P. 311–318. 7. Rossignac J., Borrel P. Multi-resolution 3D approximations for rendering complex scenes // Modeling in Computer Graphics. Berlin Springer-Verlag, 2003. – P. 455–465. 8. Schroeder W.J., Zarge J. A. and Lorensen W.E. Decimation of Triangle Meshes. In ACM Computer Graphics (SIGGRAPH 92 Conference Proceedings), Vol. 26, No. 2, 1992. – P. 65–70 9. To D., Lau R. and Green M. An Adaptive Multi-Resolution Method for Progressive Model Transmission. In Presence: Teleoperators and Virtual Environments, Vol. 10, No. 1, 2001. – P. 62–74. 10. Turk G. Re-tiling Polygonal Surfaces. In ACM Computer Graphics (SIGGRAPH92 Conference Proceedings), Vol. 26, No. 2, 2002. – P. 55–64. 11. Watt Alan. 3D Computer Graphics. Addison-Wesley, 3rd. edition, 1999.