УДК 681.3

К.т.н, доцент Соколова Н.А., магістрант Рихлюк О.В.

Національний технічний університету України
«Київський політехнічний інститут»

ВИБІР РІВНЯ ДЕТАЛІЗАЦІЇ ТА СПРОЩЕННЯ ПОЛІГОНАЛЬНИХ МОДЕЛЕЙ ОБ′ЄКТІВ

Abstract

Hadiya Sokolova, docent, PH.D; Olga Rykhlyuk, student
The Choosing of Detail Level and Simplifying of Polygonal Models
This article is about ways to choose level of detail, that connects geometrical complexity of the object and distance to observer. Different methods of simplifying object's polygonal surfaces are compared and analyzed. There a method of simplifying of polygonal surfaces, that uses consequential removing of vertexesis presented. There is also presented aquantitative dependence between the number of deleted vertexes and distance to observer, it allows to define level of detail uniquely.

Вступ

Тривимірні об'єкти в системах комп'ютерної графіки найбільш часто моделюються полігональною сіткою суміжних трикутників. Кожен трикутник є елементарним примітивом, що впливає на результуюче зображення. При спрощенні полігональної сітки відбувається зменшення числа трикутників і за рахунок цього збільшуються розміри полігонів що лишилися. У графічній системі доцільно відображати лише ті грані, що проектуються на область, співрозмірну з розмірами пікселя екрану. В ідеалі щонайменше одному трикутнику повинні відповідати 1-2 пікселі. Зменшення надлишкової деталізації дозволяє пришвидшити процес візуалізації, тобто зменшити навантаження на відеокарту, процесор або шину пам'яті [1]. Крім спрощення геометричної складності об'єкта також існує задача переходу від одного рівня деталізації до іншого та визначення конкретного геометричного представлення у даний момент часу. Наприклад, при віддаленні об'єкта від спостерігача доцільно замінити детальний опис об'єкта на більш грубе представлення, але так, щоб перехід між рівнями був непомітним, а якість візуалізації не погіршувалась.

Дана проблема виникає в системах архітектурного та ландшафтного проектування, різного роду симуляторах, системах візуалізації наукових (наприклад медичних) даних, об’ємних презентаціях та комп’ютерних іграх [2].

Проведено аналіз відомих алгоритмів спрощення, переходу та вибору рівня деталізації та на його основі запропоновано спосіб спрощення полігональної поверхні шляхом послідовного видалення вершин. Аналіз існуючих методів вибору рівня деталізації показав, що вони не встановлюють кількісної залежності між числом примітивів об’єкта і умовами візуалізації. Тому необхідно вивести числове співвідношення між кількістю вершин, що видаляються, та відстанню до спостерігача, що однозначно дозволить визначити рівень деталізації.

Постановка задачі

Метою роботи є модифікація стандартного підходу до оптимізації тривимірної сцени, аналіз алгоритмів спрощення полігональних поверхонь, модифікація одного із стандартних алгоритмів, який би дозволяв зберегти топологію і вигляд моделі, найбільш близький вихідному. Також необхідно сформулювати числове співвідношення між кількістю вершин, що видаляються, та відстанню від об’єкта до спостерігача.

Термінологія

LOD (Level Of Detail) - рівень деталізації полігональної моделі. Це один із варіантів представлення моделі, одержаний в результаті зменшення кількості вершин.

Проста вершина - це вершина повністю оточена трикутниками і кожне ребро, яке входить до неї, належить лише двом трикутникам.

Особливий кут - кут між нормалями двох суміжних або сусідніх поверхонь.

Особливе ребро - загальне ребро між двома сусідніми трикутниками, двогранний кут між якими більше особливого кута.

Гранична вершина - це вершина яка належить до границі полігональної моделі, т. е. не повністю оточена трикутниками.

Внутрішня вершина - це проста вершина яка належить двом особливим ребрам.

Зовнішня вершина - це проста вершина, яка належить одному або трьом особливим ребрам [3].

Алгоритм спрощення полігональної моделі

Алгоритми спрощення формують по вихідному представленню об’єкта його спрощене представлення. Вони класифікуються згідно основних принципів спрощення: кластеризація вершин, видалення вершин, стягування ребра, використання обмежуючих поверхонь.

На вхід алгоритмів *кластеризації* подається тривимірна решітка точок і кожна точка переноситься у найближчу точку решітки. Ті полігони, у яких після виконання операції виявляється менше трьох різних вершин відкидаються [10]. Цей метод легко реалізується та є швидким через свою лінійність, точність залежить від того, чи співрозмірні розміри решітки розмірам граней об’єкта. *Стягування ребра* є злиттям двох вершин, що утворюють ребро, у деяку точку. *Використання обмежуючих поверхонь* дозволяє максимально наблизити апроксимовану поверхню до вихідної шляхом введення двох обмежуючих поверхонь. Спрощення проводиться шляхом видалення вершин.

Алгоритми, що базуються на принципі видалення вершин, описані у роботах [8,9,10]. Їх було взято за основу у роботі, адже вони враховують локальну топологію об’єкта і на кожному кроці видаляють найменш важливу вершину.

Запропонований у роботі алгоритм спрощення полігональної поверхні складається з трьох основних кроків:

1. Аналіз локальної топології та геометрії вершин (пошук простих вершин та інше).
2. Обчислення критерію M(V), який показує яку вершину краще всього видалити на даному кроці.
3. Вилучення та триангуляція місця вилученої вершини.

Більш докладно розглянемо кожну з трьох частин алгоритму:

1. Знаходимо вершини , які є кандидатами на вилучення - це прості, граничні або внутрішні вершини. Складних та зовнішніх вершин ми не торкаємось.
2. Обчислення критерію.

Стандартним підходом є пошук дистанції від вершини до площини, якій належать суміжні вершини. Можливий випадок, коли вони не належать до однієї площини – тоді обираємо усереднену найближчу площину. Цей підхід не є найкращим, оскільки враховує тільки один фактор. Було вирішено додати у якості ще одного фактору дистанцію до найближчого ребра з тих, які з'єднують суміжні точки з вершиною-кандидатом на видалення. Цей критерій краще відображає локальні топологічні та геометричні властивості моделі.

Тобто замість:

$$M(v) = D$$

де D - дистанція до найближчої площини, покладемо:

$$M(v) = a_1 * D_1 + a_2 * D_2;$$

Де D_1 - дистанція до найближчої площини.D_2- дистанція до найближчого ребра, a_1 та a_2 - коефіцієнти критерію. Їх можна обрати, наприклад, таким чином, щоб ці дві відстані робили однаковий внесок у деякому тривіальному випадку. Наприклад, видалення вершини у тетраедрі.

Чим меншого значення набуває $M(v)$, тим кращим буде кандидат на видалення. Зазначимо те, що зміна критерію жодним чином не змінює асимптотичну складність всього алгоритму. Також обчислення дистанції від вершини до ребра відбувається досить просто, порівняно з іншими частинами цього алгоритму, тому зміни невідчутно впливають на швидкість його роботи [4].

3. Вилучення та триангуляція

Якщо ми збираємось вилучити внутрішню вершину, то дірка, яка з'явилась, повинна бути розбита на дві частини особливим ребром. При вилученні простої або граничної вершини ми отримуємо одну дірку, при вилученні внутрішньої - дві. Тобто дірка - це в загальному випадку невипуклий полігон, який необхідно розбити на трикутники. У випадку, коли не можемо розбити полігон – вершина не видаляється.

Після видалення чергової вершини переходимо до пункту 1. Алгоритм працює доти, поки не отримаємо необхідну точність або кількість полігонів[4,5].

За запропонованим алгоритмом можна формувати об'єкти різного рівня деталізації , однак для використання різних представлень необхідно заміняти одне представлення іншим. При цьому необхідно зробити процес переходу найменш помітним користувачеві.

Вибір рівня деталізації

Алгоритми цієї групи визначають, який рівень деталізації використовується у даний момент, і коли необхідно здійснювати перемикання між рівнями деталізації. Для реалізації поставленої задачі необхідно визначити співвідношення між кількістю точок, що видаляються, і відстанню до об'єкта[3].

Розглянемо перспективну зміну розмірів об'єкта у залежності від відстані до точки спостереження. Скористаємось співвідношенням

$$h_p = \frac{h}{z/d} \qquad (1)$$

де – h висота об’єкта , h_p – розмір об’єкта по осі Y на площині проекції, z – відстань від точки спостереження до об’єкта, d – відстань від точки спостереження до картинної площини.

Таким чином, на певній відстані розмір проекції грані на картинну площину менший за розмір пікселя екрану, і обробка даної грані не має сенсу. Пропонується за рахунок зменшення кількості граней збільшувати їх розміри, залишаючи практично незмінним площу проекції грані на площину. Припустимо, що усі грані мають приблизно однакові розміри. Тоді довжина ребра грані рівна h, та з площі трикутника її значення можна визначити як

$$h = \sqrt{2S_F} \qquad (2)$$

де S_F середня площина однієї грані , що

$$S_F \approx \frac{S}{F} \qquad (3)$$

Де S – загальна площина поверхні об’єкта, F – кількість граней. Враховуючи вирази (1), (2) і (3) , середній розмір проекції грані, розміщеної перпендикулярно осі Z, має вигляд:

$$h_p \approx \frac{\sqrt{2S/F}}{z/d} \qquad (4)$$

Розглянемо два рівня деталізації одного і того ж об’єкта, розміщеного на відстані z_1 і z_2 до спостерігача . На різних відстанях розміри проекцій ребер повинні бути однаковими, звідси отримуємо співвідношення між числом граней і відстанню до користувача для двох рівнів деталізації:

$$F_2 \approx F_1 \left(\frac{z_1}{z_2}\right)^2 \qquad (5)$$

Це дозволяє знайти залежність зміни кількості граней в залежності від відстані. Після декількох математичних перетворень отримуємо співвідношення

$$T \approx (V_1 - 2)\left(1 - \left(\frac{z_1}{z_2}\right)^2\right).$$

Висновки

У роботі розглянуло проблему оптимізації тривимірної сцени для адаптації її до відображення на різного роду електронних пристроях з метою зменшення використання ресурсів системи та прискорення візуалізації. Запропоновано поетапний алгоритм, що складається з визначення необхідних рівнів деталізації об’єктів на сцені, спрощення полігональних об’єктів до необхідних рівнів деталізації та адаптація процесу перемикання між рівнями деталізації (рис. 1).

Встановлено залежність між відстанню від об’єкта до користувача та кількістю граней , що формують поверхню. Запропонований алгоритм зменшення кількості полігонів показує візуально добрі результати після вилучення до 90% усіх вершин на складних об'єктах Також алгоритм дозволяє керувати кількістю видалених трикутників. Поєднання даного алгоритму з алгоритмом вибору рівня деталізації істотно зменшує навантаження на відеокарту, але кількість обчислень у процесорі зростає. Однак слід відзначити, що він працює з об'єктами, які значно менше розмірів сцени, де вони відображаються. Слід вносити деякі зміни для обробки, наприклад, великих ландшафтів, що може стати предметом подальших досліджень.

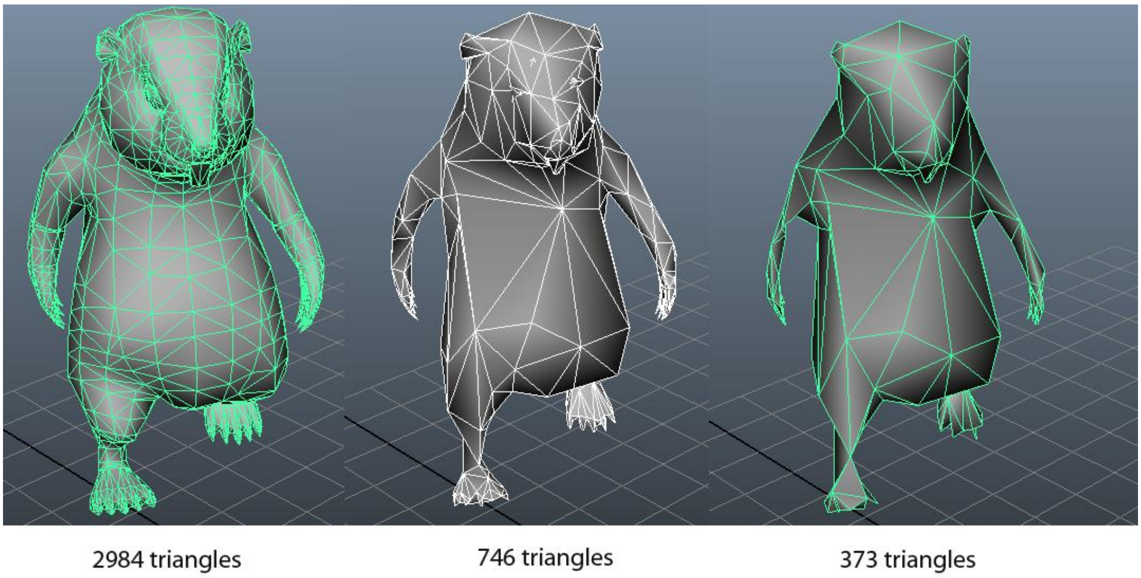

Рис.1. Спрощення полігональної моделі об’єкта.

Література

1. *Гусев А.В., Ивашин С.Л., Талныкин Э.А.* Математические модели сцен в синтезирующих системах визуализации реального времени // Автометрия. 1985. № 4. 3-9.
2. *Захаров А.А. Брагин П.А.* Разработка методов изменения геометрической сложности графических объектов для систем генерации визуальной обстановки // Методы и устройства передачи и обработки информации. Вып. 2. СПб.: Гидрометеоиздат, 2002. 88-92.
3. *Препарата Ф., Шеймос М.* Вычислительная геометрия. М.: Мир, 1989.

4. *Скворцов А.В.* Алгоритмы построеният риангуляции с ограничениями // Вычислительные методы и программирование. 2002. 3, № 1. 86-89

5. *Эйнджел Э.* Интерактивная комп'ютерная графика. Вводный курс на базе OpenGL. М.: Издательскийдом"Вилиамс", 2001.

6. *Akenine-Miller T., Haines E.* Real-time rendering. Natick: A.K. Peters Ltd., 2002.

7. *Coorg S., Teller S.* Real-time occlusion culling for models with large occluders // Proc. 1997 Symposium on Interactive 3D-Graphics. New York: ACM SIGGRAPH, 1997. 83-90.

8. *Luebke D.P.* A developer's surveu of polygonal simplification algorithms // IEEE Computer Graphics and Applications. 2001. 21, N 3.24-35.

9. *Melax S.* A Simple, fast and effective polygon reduction algorithm // Game Developer. 1988. 5, N 11. 44-49.

10. *Ronfard R., Rossingac J.* Full-range approximation of triangulated polyhedral// Computer Graphics Forum, 1966. 15, N3 67-76.