УДК 004.92

АНАЛІЗ ОСНОВНИХ КЛАСІВ АЛГОРИТМІВ РЕНДЕРИНГУ

Вяткін Сергій, канд. техн. наук, старший науковий співробітник
Інститут автоматики та електрометрії СВ РАН, Росія
Павлов Сергій, доктор техн. наук, професор кафедри проектування медико-біологічної апаратури
Романюк Сергій, аспірант,
Вінницький національний технічний університет, Україна

Виділяють два основні класи алгоритмів рендеринга [1-4]:

а) послідовний обхід бази даних геометричних примітивів з метою виділення об'єктів, що беруть участь у формуванні зображення (алгоритми полігональної растрової графіки [2]);

б) обхід модельного простору, в яке занурені об'єкти (наприклад, алгоритм трасування променів) [1].

Перший клас алгоритмів в основному застосовується при рендерингу в реальному часі. Це пов'язано можливістю спрощення геометричних примітивів. Як правило, це плоскі полігони, над якими не потрібно проводити складних обчислень при геометричних перетвореннях.

У другому випадку обхід простору не передбачає обчислення в реальному часі, так як сам по собі є трудомістким і, як правило, застосовується поряд з високоякісними алгоритмами рендеринга.

У свою чергу способи обходу простору поділяються на:

покроковий (step by step), коли з кожного пікселя випускається промінь, і алгоритм з деяким кроком реалізується по цьому променю;

ієрархічний обхід простору: в цьому випадку об’єм обходиться за деякими меншим об’ємом [3].

Існують так само гібридні алгоритми, в яких, з одного боку здійснюється ієрархічний обхід простору, з іншого боку база даних має так само ієрархічну структуру, і пошук здійснюється за елементами простору, використовуючи ієрархічну структури бази даних.

Проте, до всіх класів алгоритмів з метою їх оптимізації слід застосовувати стандартний спосіб пошуку вузького місця в обчисленнях (пляшкове горлечко).

У випадку покрокового (step by step) алгоритму найбільш дорогим з обчислювальних витрат є сам обхід модельного простору з метою пошуку елементів об’єму формуючих зображень і оптимізувати тут слід саме цей обхід.

Щодо випадку обходу бази даних можна сказати таке - в силу наявності архітектур апаратної підтримки алгоритмів рендеринга простих геометричних примітивів (полігонів), часто вузьким місцем є час передачі даних від бази даних до геометричного процесора. В цьому випадку оптимізація будується виходячи з апаратної реалізації алгоритму.

Оптимізації можливі на етапі обходу полігональної бази даних. База даних може мати ієрархію - дрібні примітиви, що розташовані близько один до одного в просторі, можуть бути об'єднані в більш великі об'єкти і т.д. Обхід

може здійснюватися спочатку над елементам ієрархії більш високого рівня, потім алгоритм спускається до більш дрібним примітивам.

Аналогічна ситуація з алгоритмами рендеринга, основаних на обході модельного простору. У цьому випадку алгоритм може модифікуватися способом доступу до бази даних, тобто база даних може будуватися за ієрархічним принципом. Але все ж суть алгоритмів залишається попередньою, для покрокових - це циклічний обхід простору. Тут можливі варіанти - розрізняють два види обходу модельно простору: рендеринг в просторі зображення, коли з кожного пікселя випускається промінь і алгоритм йде з деяким кроком по променю; і рендеринг в просторі об'єкта, коли обходиться вихідний простір, кожна отримана точка вибірки проектується на площину, паралельну екрану комп'ютера.

Як уже зазначалося, широкого розповсюдження набули бази даних, в яких об'єкти представлені у вигляді поверхонь, апроксимованих плоскими полігонами. Поширеність обумовлюється простотою геометричних перетворень (поворот, зсув, стиск-розтяг, перспективне перетворення) над такими геометричними примітивами. Слід зазначити деякі недоліки полігональних баз даних: розмір бази даних зростає квадратично по відношенню до збільшення деталізації об'єктів; при скануванні двовимірного простору не можна отримати повноцінного тривимірного зображення. Полігональна тривимірна графіка зі скануванням полігонів в площині зображення не є тривимірною в повному розумінні цього слова. Інформація, яка надається користувачеві в такій технології - неповна. Головне - це відсутність інформації про глибину об'єкта, мається на увазі не відсутність Z - координати точки поверхні, а відсутність інформації про промені, що проходить крізь об'єкт.

Існує інше завдання - об'єкти представлені у вигляді аналітичних функцій [4].

Основні вимоги до такого подання: визначення належності елемента простору об'єкту і навпаки; знаходження нормалі до поверхні для обчислення моделі освітлення

Функціональне завдання вигідно відрізняється від дискретного полігонального опису в тому сенсі, що не існує витрат на підвищення деталізації об'єктів.

Однак алгоритм рендеринга виявляється досить трудомістким - по-перше, це, як правило, покрокові алгоритми обходу модельного простору (наприклад, трасування променів), по-друге, обчислення функцій в певній точці простору може виявитися трудомістким через складність самих функції. У цьому випадку потрібно шукати інші способи оптимізації: використовувати обмежуючі оболонки, за допомогою яких можна виділити ділянку, а не сканувати весь обєм; спрощене обчислення функцій.

Слід згадати про існування ще одного способу подання даних - це так зване воксельне завдання [3]. Як правило, таке завдання потрібно для специфічних додатків - наукової візуалізації, дефектоскопія в технічних додатках, медичні програми.

Тут достовірність зображення має першорядне значення, і спрощення моделі неприйнятно. В цьому випадку відповідним способом відображення є покрокові алгоритми обходу простору. При цьому буде складно використовувати методику обмежуючих оболонок, так як заздалегідь неможливо визначити необхідну ділянку. У цьому випадку швидкість відображення наближено постійна і пропорційна частоті сітки. Слід зазначити, що і в цьому випадку база даних має великий розмір при апаратної реалізації алгоритму, існує проблема передачі даних від бази даних до геометричного процесора. Таким чином, тут слід оптимізувати передаючі канали, пристрій пам'яті і т.д.

Резюмуючи вищевикладене, хочеться відзначити, що кожен клас алгоритмів суворо ставиться до певних способів завдання і кожному способу завдання зіставляються додатки, для яких вони застосовуються.

Полігональна растрова графіка за останній час отримала широка поширення в основному за рахунок індустрії розваг, CAD / CAM, військових і цивільних симуляторів. Подання об'ємних даних у вигляді завдань на регулярній тривимірної сітці, застосовується в наукових і технічних додатках.

За останні кілька років розвиток алгоритмів растрування плоских полігонів в тривимірному просторі не зазнавали істотних змін. Розробники апаратних засобів підтримки таких алгоритмів, як правило, покладаються на грубу силу (brute force). Нарощується частота геометричного процесора, збільшується пропускна здатність каналів і т.д.

Таким чином, алгоритми тривимірної комп'ютерної графіки класифіковані, і можуть відрізнятися деталями в межах одного класу.

З іншого боку розвиток апаратно реалізованих алгоритмів відрізняється від розвитку програмно реалізованих. У першому випадку розробник повинен бути обізнаний про архітектуру машини, окремих її частин, вузлів, і проектування алгоритму має відбуватися в цьому напрямку.

У другому випадку, коли алгоритм розробляється для універсальної обчислювальної машини, зазвичай на мові програмування високого рівня, розробнику не потрібно замислюватися над технічними аспектами.

Список використаної літератури

1. Херн Д., Бейкер М. Компьютерная графика и стандарт OpenGL / Д. Херн, М. Бейкер. — М. : Издательский дом "Вильямс", 2005. — 1168 с.
2. Романюк О. Н. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об’єктів. Монографія. / О. Н. Романюк, А. В. Чорний. - Вінниця : УНІВЕСУМ-Вінниця, 2006. — 190 с.
3. Akenine-Möller T., Hoffman N., Haines E. Real-Time Rendering / T. Akenine-Möller, N. Hoffman., E. Haines. — Wellesley: A. Peters, 2007. — 1045 p.
   Computer Graphics: Theory Into Practice/ McConnell Jeffrey J. — Sudbury : Jones and Bartlett Publishers, 2006. — 500 p.
4. Вяткин С.И. Моделирование сложных поверхностей с применением функций возмущения / С.И. Вяткин // Автометрия. — 2007.—Т. 43, № 3. — С.40–47.