УДК 004.027

УДОСКОНАЛЕННЯ АЛГОРИТМУ Z-БУФЕРИЗАЦІЇ

Савчук Тамара, Чернобай Олексій

Вінницький національний технічний університет

Анотація

У доповіді запропоновано удосконалення алгоритму z–буферізації з використанням ієрархічного розбиття полігона по середній точці та порівняння значень вершин та середньої точки полігона зі значеннями z-буфера, що забезпечує зменшення кількості порівнянь z-координат, і як наслідок, зменшення часу виконання рендерингу тривимірного зображення. Перевагою даного удосконалення є можливість регулювання швидкості роботи z-буферизації шляхом зменшення або збільшення кількості ітерацій розбиття полігона.

Abstract

The report proposes an improvement of the z-buffering algorithm using the hierarchical breakdown of the polygon along the midpoint and compares the values of the vertices and the mean point of the polygon with z-buffer values, which reduces the quantity of z-coordinates comparisons, and as a consequence, reduces the rendering time of the three-dimensional image. The advantage of this improvement is the ability to control the speed of z-buffering by reducing or increasing the quantity of iterations of the polygon breakdown.

Алгоритм, що використовує z-буфер - це один з найпростіших алгоритмів видалення невидимих поверхонь у просторі зображення [1]. Ідея z-буфера є простим узагальненням ідеї про буфер кадру, що використовується для запам'ятовування координати z чи глибини кожного видимого пікселя в просторі зображення . В процесі роботи алгоритму глибина чи значення z кожного нового пікселя, який потрібно занести в буфер кадру, порівнюється з глибиною того пікселя, який вже занесений в z-буфер. Якщо це порівняння показує, що новий піксель розташований попереду пікселя, що знаходиться в буфері кадру, то новий піксель заноситься в цей буфер і, крім того, проводиться коригування z-буфера новим значенням z. Якщо ж порівняння дає протилежний результат, то ніяких дій не проводиться.

Алгоритм z-буферизації практично не має недоліків, якщо реалізується апаратно. Програмно ж існують інші методи, здатні конкурувати з ним: Z-сортування («алгоритм художника») і двійкове розбиття простору (BSP). Основний недолік z-буферизації полягає в споживанні великого обсягу пам'яті: в роботі використовується так званий буфер глибини або Z-буфер [2].

Існуючі рішення по удосконаленню z-буферизації можна поділити на такі, що покращують швидкість роботи алгоритму, і такі, що зменшують витрати пам’яті.

До рішень, покращуючих швидкість роботи алгоритму, відносять такі:

— використання половини розрядності z-буфера для уникнення щокадрового очищення даних [2];

— попереднє сортування полігонів за дальністю [2];

— рекурсивне розбиття об’єктного простору та простору зображення на тайли (зазвичай 8x8) та порівняння даних z-буфера з середнім значенням глибини тайлу (ієрархічний z-буфер) [3];

— використання перехідної когерентності (застосовується у анімації тривимірних зображень) [3];

— обчислення потенційно видимих поверхонь (PVS) за даними розміщення та повороту камери та полігонів [3];

— використання z-піраміди (декількох буферів з різними рівнями точності) [3].

Рішення, які зменшують витрати пам’яті:

— використання z-буфера з фіксованою точністю [1];

— зменшення розміру z-буфера та розбиття простору зображення на ділянки розміру z-буфера [1];

— стиснення без втрат та відновлення z-буфера для економії пропускної можливості оперативної пам'яті [2].

Більшість вищеозначених рішень сумісні між собою і можуть застосовуватися разом. Узагальненням багатьох рішень є ієрархічний z-буфер. Його використання дозволяє зменшити розмір z-буфера до розміру тайлу (зазвичай 8x8), а також зменшити кількість порівнянь z-значень полігона з даними z-буфера. При цьому витрачається додаткова пам'ять на зберігання даних z-піраміди (багатошарового z-буфера з різними рівнями точності) та ієрархічно поділеного об'єктного простору (причому збереження даних об'єктного простору витрачає на 33,3% більше пам'яті, ніж збереження даних простору зображення), з чого можна зробити висновок, що витрати пам'яті в даному рішенні набагато вищі за допустимі для середньостатистичних відеокарт [3]. Тому необхідно розробити рішення, яке б прискорювало роботу алгоритму z-буферизації, але не витрачало великих об'ємів відеопам'яті.

З метою усунення означених недоліків доцільно в алгоритмі z-буферизації передбачити розбиття полігонів. Такий підхід необхідно реалізувати з врахуванням когерентності у просторі зображення, що базується на припущенні, що однократне обчислення загальної z-глибини певного елемента (в даному випадку, полігона) дозволить визначити видимість об'єкта, який покриває групу пікселів на екрані [3]. Розбиття полігонів може призводити до більших витрат обсягів пам'яті, але використання когерентності у просторі зображення дозволить мінімізувати ці витрати. Об'єм витрат пам'яті також залежить від способу розбиття полігонів. Їх можна розбивати на тайли [3] або через обчислення середньої точки полігона та його розбиття по ній. Необхідно, щоб кількість отримуваних полігонів після розбиття була завжди однаковою. Розглянемо можливі ситуації розбиття полігона, що представлені на рисунку 1:

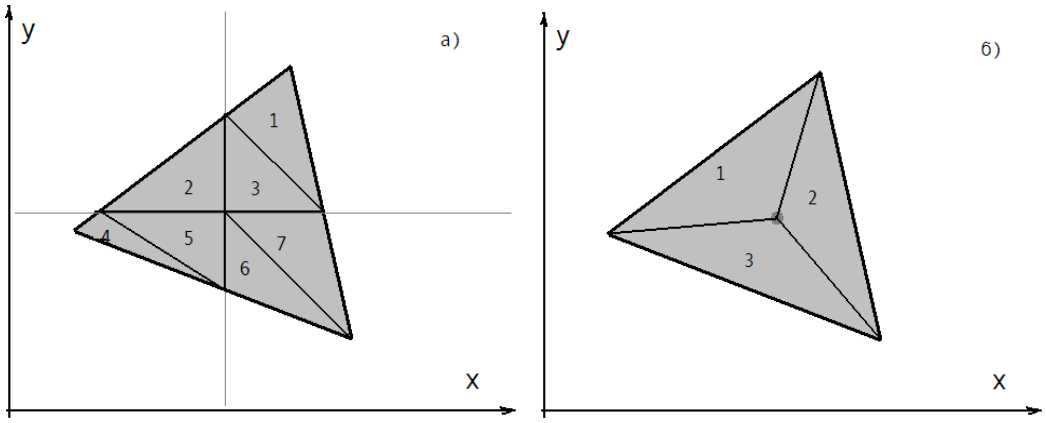

Рисунок 1 – Порівняння методів розбиття полігона тайлами (а) та середньою точкою (б)

Як видно, другий метод є більш прийнятним, оскільки при кожному розбитті отримується завжди 3 полігона. Для реалізації даного метода необхідна структура даних, яка б забезпечувала зберігання дочірніх полігонів протягом визначення видимості обчислюваного полігона. Отже, алгоритм працюватиме наступним чином:

1) обчислюється середня точка приведеного до екранних координат полігона за координатами його вершин;
2) z-значення вершин та середньої точки порівнюються з відповідними z-значеннями буфера глибини (z-буфера);

3) якщо всі z-значення вершин та середньої точки більші за z-значення буфера глибини – полігон вважається невидимим, і алгоритм починає обробку наступного полігона. Якщо всі z-значення менші за відповідні значення буфера глибини – полігон вважається повністю видимим і рендериться на екрані, після чого алгоритм починає обробку наступного полігона. Інакше – перехід до етапу 4;
4) у спеціальну структуру даних записуються 3 нових полігона.
   Перехід до етапу 1.

Даний алгоритм може спрацювати неточно, якщо край більш близького полігона або зазір між більш дальніми полігонами не буде співпадати з вершинами та середньою точкою обчислюваного полігона. В такому випадку необхідно проводити більше однієї ітерації розбиття полігона перед першим порівнянням його z-значень з відповідними значеннями буфера глибини. Також необхідно обмежувати кількість рекурсій для уникнення зациклювання роботи алгоритму.

Для перевірки роботи алгоритму проведемо рендеринг однієї сцени на екрані розміром 800x600 пікселів з використанням ієрархічної z-буферизації та z-буферизації з використанням розбиття полігона по середній точці (на початку роботи – 2 ітерації розбиття). Результати дослідження представлені в таблиці 1.

Таблиця 1 – Результати дослідження

Алгоритми	Кількість порівнянь z-координат	Час рендерингу, мс
Ієрархічна z-буферизація	20216	57
Z-буферизація з використанням розбиття полігона по середній точці	12635	51

Як видно із таблиці 1, кількість порівнянь z-координат та час рендерингу при виконанні z-буферизації з розбиттям полігонів зменшилась, що свідчить про ефективність даного алгоритму.

Отже, проведені дослідження показали, що z-буферизація з використанням ієрархічного розбиття полігона по середній точці та порівнянням його вершин та середньої точки зі значеннями z-буфера може забезпечити зменшення кількості порівнянь z-координат на 7581, що складає 40%, та зменшення часу виконання рендерингу тривимірного зображення на 6 мс, що складає 10%. Перевагою даного удосконалення є можливість регулювання швидкості роботи z-буферизації шляхом зменшення або збільшення кількості ітерацій розбиття полігона. Запропоноване удосконалення алгоритму z-буферизації може бути використано у мобільних додатках, які використовують тривимірну графіку.

Список використаних джерел:

1. Алгоритм, що використовує z-буфер [Електронний ресурс] – Режим доступу: http://compgraph.tpu.ru/zbuffer.htm

2. Z-буферизація [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/Z-буферизация

3. Ієрархічний z-буфер [Електронний ресурс] – Режим доступу: https://www.ixbt.com/video/hierar-z.html