

УДК 681.3.11

ВЫЧИСЛЕНИЕ ТЕКСТУРЫ С ПРЕДФИЛЬТРАЦИЕЙ ДЛЯ РЕНДЕРИНГА ПОЛИГОНОВ В РЕАЛЬНОМ ВРЕМЕНИ

Вяткин С.И., Романюк О.В.*

Институт автоматизации и электрометрии СО РАН, Новосибирск,
Россия

*Винницкий национальный технический университет, Украина
ran12345@mail.ru

В данной работе рассматривается организация вычислений текстуры для компьютерных систем визуализации реального времени. Заложенные в метод возможности позволяют генерировать высокореалистичные текстурированные полигоны, работать с различными форматами текстур, а также проводить декомпрессию сжатых текстур.

Введение

Для построения изображения используется, так называемая виртуальная методика [1]. Вся площадь кадра (экрана) разбивается на спаны - прямоугольники меньших размеров, которые затем обрабатываются независимо друг от друга. Таким образом, обработка одного спана является законченным элементарным циклом работы.

Независимость обработки спанов позволяет эффективно распараллеливать процесс построения изображения. Несомненным достоинством такого метода является также и то, что разбиение задачи значительно снижает требования к ресурсам вычислительных компонент. В частности, это позволяет применить локальную высокоскоростную память и упростить некоторые вычисления.

Вычислительная сложность и потоки данных при обработке спана определяются его размерами. Для задачи текстурирования граней снизу размер спана ограничивает величина потоков данных. Сверху - объем локальной памяти, качество изображения и средние размеры граней. После анализа были выбраны спаны размером 8x8, удовлетворяющие всем перечисленным условиям.

Постановка задачі

В пропонуваному методі повинні виконуватися дві основні операції: зображення різного виду текстурних карт, в тому числі і стиснутих, на полігони і завантаження текстурних карт в текстурну пам’ять.

Описание метода

Процес зображення текстурних карт на плоскі полігони починається з обчислення дробно-лінійної функції [2]. Далі виконується фільтрація, необхідна для запобігання еліазингу (aliasing). Порівняльний аналіз методів фільтрації дано в [2]. В цій роботі застосовуються так звані MIP-тар (пірамідальні) текстурні карти, запропоновані в [3]. За допомогою попередньої фільтрації отримують набір квадратних текстурних карт з різною роздільною здатністю для кожного об’єкта.

Кожній текстурній карті присвоюють відповідне цілочисленне значення так званого рівня деталізації. Далі в відповідності з текстурними координатами з кожної текстурної карти читаються по чотири тексели. Трилінійна інтерполяція по цим восьми значенням завершує процес фільтрації [4]. З допомогою видозмінення процесу фільтрації отримують деякі спеціальні ефекти. Наприклад, можливо отримати текстурний малюнок з постійною шириною контуру [5].

Коефіцієнтами трилінійної інтерполяції є дробні частини текстурних координат і рівня деталізації.

Таким чином, адреса в текстурній пам’яті, а також коефіцієнти білінійної інтерполяції генеруються безпосередньо з текстурних координат. Спочатку масштабуванням в відповідності з поточним рівнем деталізації обчислюються зміщення адресів в глобальній текстурній пам’яті і коефіцієнти білінійної інтерполяції. Далі зміщення адресів в глобальній пам’яті перетворюються в адреси локальної пам’яті.

Обчислювач текстурних координат складається з власне обчислювача координат і схеми корекції. Структура обчислювача текстурних координат також використовується для підготовки граничних значень текстурних координат, використовуваних в схемі корекції. Результати обчислень не нормалізуються і передаються на схему корекції. Схема корекції порівнює отримані координати з мінімальними і

максимальними координатами участка текстурной карты, загруженной в локальную память. При необходимости значения текстурных координат, превышающие допустимые значения, заменяются граничными.

Для рассинхронизации вычисления текстурных координат и загрузки коэффициентов необходимо использовать два экземпляра памяти под коэффициенты.

Все четыре цветовой компоненты для одного пиксела вычисляются за один такт. При этом как в случае обычных, так и контурных текстур все цветовой компоненты поступают на выход одновременно. Для вычисления цвета и прозрачности одного пиксела необходима выборка восьми значений из памяти - по четыре из каждого уровня детальности. Для достижения желаемой пропускной способности - один пиксел за один такт - все восемь значений должны считываться одновременно. Так как в каждом из уровней детальности всегда используются соседние текселы, возможно расслоение памяти по банкам для параллельного чтения всех восьми значений. При этом старшие разряды текстурного адреса используются для выборки значений, а два младших - для управления выходным коммутатором. Собственно, старшие разряды выбирают 64-разрядное слово, а младшие переставляют в нем байты (рис. 1.).

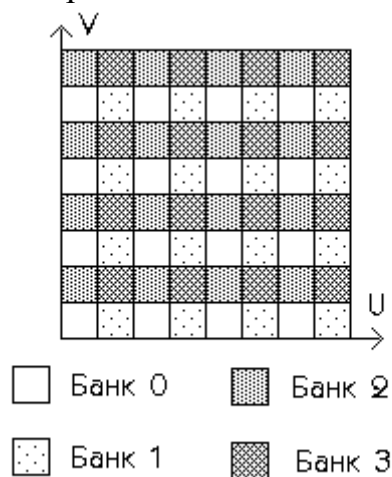


Рис. 1 Расслоение локальной текстурной памяти по банкам.

Для каждого из четырех каналов потребуется по два банка памяти: 8x8 слов для лучшего и 4x4 слова для худшего уровня детальности, в свою очередь, разделенные на четыре части.

Поскольку проекция спана в локальной памяти может быть ориентирована произвольно, нельзя начинать вычисления до окончания загрузки. Для распараллеливания процессов загрузки и вычислений необходимо иметь два экземпляра локальной памяти. Размер одного экземпляра памяти 2560 бит.

Загрузка текстурной карты в локальную память происходит последовательно. Поэтому, в отличие от выхода, вход локальной памяти не требует коммутатора.

Заключение

В процессе выполнения данной работы была построена программная модель, генерирующая последовательность кадров для анимации. По результатам моделирования была скорректирована модель вычислений и найден оптимальный путь минимизации объема данных, передаваемых из текстурной памяти. Исследовано влияние точности вычислений на качество изображения. Определены критерии точности вычислений. Также выработаны рекомендации по вычислению уровня детальности и по предварительной фильтрации текстурных карт.

Список литературы

1. R.F. Sproull, I.E. Sutherland et al. Characterization of ten rasterization techniques // *Comput. Graph.*-1989.-23, N3.
2. Paul S. Heckbert. Survey of Texture Mapping// *IEEE Comput. Graph. and Applicat.*-1986.-6, N11, pp. 56-67.
3. Lance Williams, Pyramidal Parametrics, *Computer Graphics (SIGGRAPH '83 Proceedings)*, pp. 1-11, July, 1983.
4. Вяткин С.И., Романюк О.В., Стахов А.А. Формирование реалистических изображений с использованием пошагового вычисления текселей// Сборник научных трудов по материалам международной научно-практической конференции «Современные направления теоретических и прикладных исследований», Том 3, Технические науки, Одесса, изд-во: Черноморье, С. 65-68, 2011.
5. Вяткин С.И., Романюк О.В., Костюкова Н.С. Отображение контурной текстуры на произвольно ориентированные плоские и криволинейные поверхности // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника», 2010, Донецк: ДонНТУ, вып. 11 (164). С. 118–121.

Получено 12.06.2011