

**ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ ТА ОБЧИСЛЮВАЛЬНІ СИСТЕМИ
І КОМПЛЕКСИ В ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ****УДК 621****С. И. ВЯТКИН**

Институт автоматизации и электротехники СО РАН, Новосибирск, Россия

А.Н. РОМАНЮК., Д. Л. БЛАГОДЫРЬ

Винницкий национальный технический университет

**ТЕКСТУРНЫЙ ПРОЦЕССОР ДЛЯ СИСТЕМ
ВИЗУАЛИЗАЦИИ РЕАЛЬНОГО ВРЕМЕНИ**

Рассматривается архитектура текстурного процессора для компьютерных систем визуализации реального времени. Предложенный в данной работе подход позволил создать универсальную архитектуру текстурного процессора. Заложенные в архитектуру процессора возможности позволяют генерировать высокореалистичные текстурированные грани, работать с различными форматами текстур, а также проводить аппаратную декомпрессию сжатых текстур.

We consider the architecture of texture processors for computer systems visualization of real time. Proposed in this paper an approach to create a universal architecture texture processor. Incorporated into the architecture of the processor capabilities allow to generate highly realistic textured face with a variety of formats, textures, and conduct a hardware decompression of compressed textures.

Ключевые слова: текстура, архитектура процессора.

Введение

Общая производительность системы определяется быстродействием геометрического процессора (геометрические преобразования) и пиксельного процессора (растровые операции). Анализ показывает, что для получения фотореалистичных изображений в реальном времени следует поднять производительность растровых операций.

Решить проблему нехватки памяти может применение сжатых текстурных карт. Кроме того, это снижает интенсивность обмена данными с текстурной памятью и дает возможность использования видеотекстур, т. е. текстур получаемых непосредственно с видеокамеры.

Постановка задачи

Текстурный процессор должен выполнять две основные операции: отображение различного вида текстурных карт, в том числе и сжатых, на грани и загрузку текстурных карт в текстурную память. Тем самым с хост-процессора (host-processor) снимается часть вычислительной нагрузки и снижается объем данных, передаваемых по его шине за счет использования локальной памяти текстурного процессора. Характерными особенностями данной задачи являются интенсивный обмен данными с текстурной памятью (порядка 300 Мбайт/с) и трудоемкие вычисления (20-30 32-разрядных операций на 1 пиксель (pixel)).

Архитектура текстурного процессора

Для построения изображения используется так называемая виртуальная буферная методика [1]. Вся площадь кадра (экрана) разбивается на спаны (тайлы) – 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.

Процесс отображения текстурных карт на плоские грани включает два этапа. Первый – перспективное преобразование, т. е. вычисление координат текстурной карты (u, v), соответствующих координатам (Xs, Ys) пикселя на экране. Координаты связаны дробно-линейной функцией [2].

Второй этап отображения текстуры – фильтрация, необходимая для предотвращения элиазинга (aliasing). Применяются так называемые MIP-map (пирамидальные) текстурные карты, наиболее подходящие для аппаратной реализации, предложенные в работе [3].

Элементарным циклом работы текстурного процессора является текстурирование одной грани в пределах одного спана. Он может быть разделен на следующие этапы:

- загрузка необходимых данных: коэффициентов перспективного преобразования, адресов для выделения необходимого участка текстурной карты, атрибутов самой текстурной карты и т.п.
- загрузка соответствующего участка текстурной карты в локальную память;
- вычисление текстурных координат и текстурных адресов;

- вычисление значений цветовых компонент;
- выдача результата.

Для вычисления текстурных координат в пределах одного спана с учетом специфики виртуальной методики была предложена схема, в которой задаются коэффициенты в одном из углов (левом нижнем) спана и их приращения по вертикали и горизонтали. Затем инкрементным способом вычисляют текстурные координаты во всех остальных пикселях спана [4].

Адрес в текстурной памяти, а также коэффициенты билинейной интерполяции генерируются непосредственно из текстурных координат. Сначала масштабированием в соответствии с текущим уровнем детальности вычисляются смещения адресов в глобальной текстурной памяти и коэффициенты билинейной интерполяции. Затем смещения адресов в глобальной памяти преобразуются в адреса локальной памяти.

Задачи по отражению и периодическому повторению текстурных карт возлагаются на загрузчик локальной памяти.

Полученные описанным выше способом адреса в текстурных картах рассчитаны на плоскую модель памяти с непрерывным адресным пространством. Участок текстурной карты загружается в локальную память таким образом, чтобы локальные координаты левого нижнего угла минимального охватывающего проекцию спана прямоугольника получили значение (0,0).

Таким образом, к подготовке текстурных карт предъявляются два основных требования:

- обеспечить соответствие содержимого локальной текстурной памяти указанной модели;
- обеспечить однородное соответствие текстурных координат и текстурных адресов во всех уровнях детальности пирамиды текстурных карт.

Используется алгоритм сжатия изображений CCC (Color Cell Compression) [5] и декомпрессии.

Для получения окончательного результата, т. е. цвета и прозрачности в каждом пикселе требуется произвести трilinearную интерполяцию [4].

Контурная текстура [6] обеспечивает возможность создания и отображения контуров произвольной формы в рамках текстурной карты, не размывающихся при приближении.

Задачи, возлагаемые на текстурный процессор, требуют большого количества однообразных вычислений и передачи массивов данных. Поэтому при разработке архитектуры процессора основное внимание уделялось тракту обработки и передачи данных. Повышение производительности вычислительной части потребовало максимального упрощения управляющих структур. Как следствие этого, в текстурном процессоре отсутствует какой-либо центральный управляющий автомат.

Внутренняя структура и потоки данных текстурного процессора показаны на рис. 1.

Необходимость согласования потоков внутри и вне процессора приводит к следующим особенностям архитектуры:

- все сложные вычисления производятся на конвейерных устройствах;
- вычисления и обмен данными с внешними устройствами происходят одновременно и, следовательно
- вся внутренняя память, используемая при вычислениях, удвоена;
- все потоки данных буферизованы.

Вычислитель текстурных координат состоит из собственно вычислителя координат и схемы коррекции [7]. Тракт вычислителя текстурных координат также используется для подготовки граничных значений текстурных координат, используемых в схеме коррекции.

На вычисление пары текстурных координат тратится один такт. Тракт вычислителя текстурных координат также используется для подготовки граничных значений текстурных координат, используемых в схеме коррекции. Результаты вычислений не нормализуются и передаются на схему коррекции.

Выходные данные поступают на генератор текстурного адреса.

На вычисление пары текстурных координат тратится один такт. Тракт вычислителя текстурных координат также используется для подготовки граничных значений текстурных координат, используемых в схеме коррекции. Результаты вычислений не нормализуются и передаются на схему коррекции.

Выходные данные поступают на генератор текстурного адреса.

Генератор текстурного адреса получает на входе текстурные координаты, уровень детальности спана, величину общего порядка вычислений из вычислителя текстурных координат и смещение локальных и глобальных текстурных координат друг относительно друга [4].

Полная схема трilinearной интерполяции состоит из 56 сумматоров и 28 умножителей. Отсутствие обратных связей позволяет построить интерполяторы по параллельно-конвейерной схеме, где четыре канала цветовых компонент обрабатываются независимо и одновременно. В канале прозрачности обрабатываются контурные текстуры.

Все четыре цветовые компоненты для одного пикселя вычисляются за один такт. При этом как в случае обычных, так и контурных текстур все цветовые компоненты поступают на выход одновременно.

Для вычисления цвета и прозрачности одного пикселя необходима выборка восьми значений из памяти – по четыре из каждого уровня детальности [7]. Для достижения желаемой пропускной способности – один пиксель за один такт – все восемь значений должны считываться одновременно.

Загрузчики текстуры отвечают за преобразование внешних форматов текстурных карт к внутреннему формату текстурного процессора, эффекты, связанные с отражением либо повторением

рисунка на границях текстурних карт, адресацію к внешней текстурной памяти. Вид работы определяется значением регистра команд.



Рис. 1. Поток данных в текстурном процессоре

Данный блок можно разделить на две части. Первая, генерирующая адреса внешней текстурной памяти, состоит из реверсивных счетчиков с предустановкой, сумматора и схем сравнения. Адресное пространство составляет 64 Мбайт, доступ осуществляется в страничном режиме. Вторая, отвечающая за разбор входных форматов, состоит из соответствующих форматам коммутаторов.

Текстурные карты менее чем с 4 битами на компоненту цвета не обеспечивают достаточного качества изображения. Поэтому поддерживаются только 16- и 32-битовые форматы:

- 32 бит RGBT, по 8 бит на компоненту;
- 16 бит RGBT, по 4 бита на компоненту;
- 16 бит RGB, R и B – по 5 бит, G – 6 бит;
- сжатая CCC.

Слово CCC (64 бит) включает 16-разрядную битовую маску и два 24-разрядных слова RGBT (RGB). Интерфейс с ведущим процессором строится из двух основных принципов. Во-первых, текстурный процессор является промежуточным устройством между хост-процессором и текстурной памятью. Фактически, хост-процессор адресуется в текстурную память и получает из нее предобработанные данные. Во-вторых, текстурный процессор должен работать в паре с любым универсальным процессором с 64-разрядной шиной.

Ведущему процессору доступны два набора регистров. Для записи: регистр команд и банк входных данных. Для чтения: регистры состояния и выходного буфера. Все регистры 64-разрядные. Номер регистра передается по 4-разрядной адресной шине.

Для инициирования прерываний хост-процессора через внешние выходы доступны биты регистра состояния, указывающие заполнение буферов и сообщающие об окончании рабочего цикла.

Заключение

На основе описанной выше архитектуры была построена программная модель СБИС, генерирующая последовательность кадров для анимации.

По результатам моделирования была скорректирована модель вычислений в текстурном процессоре 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 для внешнего оборудования:

- по вычислению уровня детальности;

- по предварительной фильтрации текстурных карт.
Высокоуровневое моделирование показало правильность выбранного подхода.
Тестовая генерация топологии показала, то требуемая площадь кристалла для технологии КМОП 0.7 мкм с двумя слоями металлизации не превысит 64 ммІ. Сложность процессора – порядка 300'000 вентиляей.

Литература

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 текстелей // Сборник научных трудов по материалам международной научно-практической конференции «Современные направления теоретических и прикладных исследований», Том 3, Технические науки, Одесса, изд-во: Черноморье, С. 65-69, 2011.
5. G. Campbell, T. DeFanti, J. Frederiksen, S. Joyce, and L. Leske, "Two Bit / Pixel Full Color Encoding," in Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, pages 215-223, 1986
6. 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.
7. Вяткин С.И., Романюк О.В. Вычисление текстуры с предфильтрацией для рендеринга полигонов в реальном времени // Четвертая международная научно-техническая конференция "Моделирование и компьютерная графика" 5-8 октября 2011 года, Донецк, ДонНТУ, Украина, С. 47-50.

Надійшла до редакції
10.5.2012 р.

УДК 621.384.3

О.О. МУХА, В.Г. КОЛОБРОДОВ

Національний технічний університет України «Київський політехнічний інститут»

КРИТЕРІЙ КОРЕКЦІЇ ЗОБРАЖЕННЯ ПІДВОДНОГО ОБ'ЄКТУ В ОПТИКО-ЕЛЕКТРОННИХ СИСТЕМАХ СПОСТЕРЕЖЕННЯ

В даній статті досліджується вплив дальності до об'єкта, що визначається на основі використання метода стробіювання по дальності, на ефективність роботи методу корекції зображення, сформованого ОЕСПС через схвильовану межу розподілу «повітря-вода».

In this paper investigated the influence of approximate determine distance of object, that implemented through the use of the Gated Viewing method, the efficiency of the method of correcting the image that formed by excited surface «air-water».

Ключові слова: оптико-електронна система підводного бачення, корекція зображення.

Вступ

Процес дослідження Світового океану, в останній час, обумовив стрімкий розвиток оптико-електронних систем підводного спостереження (ОЕСПС). Такі системи можуть бути встановлені на підводних носіях, кораблях, літаючих апаратах. В останніх двох випадках до проблем інтенсивного поглинання та розсіювання випромінювання у водному середовищі, що обмежують ефективність роботи таких систем, додається проблема пов'язана з необхідністю ведення спостереження через поверхню розділу «повітря-вода», майже завжди схвильовану. Як правило, ОЕСПС виконуються на основі активно-імпульсних лазерних систем з застосуванням методу стробіювання по дальності. Такий підхід дозволяє зменшити вплив поглинання і інтенсивного розсіювання випромінювання у зворотному напрямку, а також надає можливість оцінки дальності до об'єкта спостереження. Вирішити проблему схвильованої водної поверхні намагаються шляхом застосування різних видів цифрової обробки. В останній час з'явилась низка теоретичних робіт, присвячених задачам корекції зображення, спотворених поверхневим хвилюванням, в припущенні, що відома повна або часткова інформація про розподіл нахилів елементарних частин поверхні розділу. А в роботі [1] приведені результати експериментальної реалізації методу корекції такого типу. Розрахунки на основі моделі визначення граничної дальності дії ОЕСПС, запропонованої в [2], показують що для відносно невеликих значень швидкості надводного вітру, ефективність роботи зменшується в рази по відношенню до випадку спокійної водної поверхні. Тому дуже гостро постає питання корекції