Секция математика и информатика

РАЗРАБОТКА СИТЕМЫ АНИМАЦИИ ТРЕХМЕРНЫХ СЦЕН НА GPU

GPU BASED 3D SCENES ANIMATION SYSTEM DEVELOPMENT

Иванов Тимофей (8-1)

Амбросовская Дарья (8-2)

Фокеев Борис (8-2)

Подкопаев Александр (8-4)

Уросова Софья (8-4)

Григорович Вячеслав (9-5)

Крейнин Матвей (9-5)

Писарев Евгений (9-5)

Синяков Степан (9-5)

Файзуллин Музаффар (10-1)

Шабанов Никита (10-3)

Мосягин Олег (10-5)

Сорин Алексей (10-5)

Сорин Николай (10-5)

Филиппов Денис (10-5)

Кожухаров Никита (10-6)

Пономаренко Ульяна (10-6)

Тарасов Денис (10-6)

(ГБОУ Санкт-Петербургский губернаторский физико-математический лицей № 30)

Научный руководитель:
Галинский Виталий Александрович,
заместитель директора СПб ГФМЛ № 30 по ИТ,
руководитель группы компьютерной графики
СПб ГФМЛ № 30

The project is devoted to development and implementation of 3D scenes' visualization based on the usage of computer graphics kernel (GPU — graphics processing unit) for computation and animation. Research area is covered shader program developing for visualization and object control. Authors use vertex, tessellation, geometry and fragment shader programs and GLSL shading language.

Доклад посвящен разработке и реализации системы визуализации трехмерных сцен на основе переноса вычислительных процессов анимации и отображения на графическое ядро компьютера (GPU — graphics processing unit). В работе рассматриваются разработанные приемы написания шейдерных программ для графических процессоров для управления визуализацией и взаимодействием объектов в сцене. Созданная система использует вершинные, тесселяционные, геометрические и фрагментные шейдерные программы и язык GLSL.

Перед авторами была поставлена задача: разработать и реализовать систему анимации, включающую в себя возможность моделирования и отображения трехмерных объектов. При этом основной объем вычислений должен производиться на графическом ядре компьютера, посредством использования шейдеров.

Проект условно разбит на несколько взаимосвязанных частей.

Анимация — основной компонент системы — выполняет роль диспетчера построения кадра. Активные объекты анимации хранятся в отдельной очереди обработки и обслуживаются на каждом кадре путем вызова методов реакции на межкадровое взаимодействие и отображение. При этом для обеспечения вывода полупрозрачных объектов и объектов с постобработкой разработана система отложенного вывода, в которой объекты регистрируются и активируются после основного построения геометрических элементов.

Система построения — рендеринга — является основным ядром отображения. Для ее функционирования разработан специальный конвейер вывода, суть которого — обеспечить правильное отображение трехмерных объектов с учетом их освещения, затенения другими элементами сцены, удалением невидимых поверхностей. В конвейере предусмотрено использование всех типов современных шейдеров (микропрограмм для видеокарты) отображения: вершинных (на них в проекте выполняются базовые преобразования координат), тесселяционных (на этом шаге поверхности при необходимости адаптивно подразбиваются на более мелкие фрагменты), геометрические (в работе они используются для порождения новой геометрии построения, "выращивани" элементов сцены и т.п.).

Последняя фаза — фрагментые шейдера. Нами разработан механизм, позволяющий выполнять на фрагментных шейдерах основную часть вычислений, связанных с освещением сцен и их постобработкой. Данный подход обеспечивает многократное увеличение производительности за счет параллельного выполнения шейдеров для каждой точки экрана. Все построения происходит в текстурные плоскости, позволяющие хранить разнообразные параметры и атрибуды геометрических объектов. Это дает возможность разбить конвейер вывода на несколько фаз: фаза отображения геометрии с занесением ее параметров в текстурные плоскости, фаза освещения с применением разнообразных источников света к тем точкам экрана, атрибуты которых удовлетворяют условию попадания в зону их действия, фаза сборки и постобработки для объединения вместе текстурных плоскостей, выполнения отложенного построения, вывода разнообразной информации, постобработке всего кадра и т.п. Для моделирования затенения от источников света используются карты теней, которые строятся на отдельном "проходе" по сцене с камерой, "перенесенной" в позицию источника. Карта теней (*shadow map*) используется в дальнейшем при учете финального освещения.

В системе построения разработана подсистема создания геометрических объектов на базе опорных структур — топологий. Топология служит для моделирования и контроля форм объектов, однако, вся отображаемая информация передается на хранение на видеокарту. Посредством применения тесселяционных шейдеров топологические структуры подразбиваются на детали на уровне их построения на видеокарте.

Помимо перечисленных компонентов авторами разработаны различные подсистемы взаимодействия (система ввода, синхронизации по времени, звука и др.)

Для оптимизации системы хранения данных был разработан специальный механизм распределения памяти. Его необходимость обусловлена наличием в системе анимации отложенного вывода, при котором объекты регистрируются на построения на разных фазах графического конвейера. Такие объекты требуется удалять в конце построения кадра, когда вывод всех фаз конвейера закончен. Множественные аллокации памяти, приводящие к ее фрагментации и увеличению времени построения сцен, были заменены на единичные выделения массивов данных. Размер памяти, выделяемый на каждый последующий блок, вдвое больше размера предыдущего. Тем самым мы добились уменьшения временных затрат во время построения каждого кадра, так как процесс выделения и освобождения выполняется за константное время.

Разработанный подход построения анимационной системы позволяет реализовывать проекты, ориентированные на визуализацию сложных статических и динамических трехмерных сцен в реальном времени. Благодаря использованию графического ядра компьютера, процесс создания и отображения анимации был оптимизирован за счет переноса основных вычислений и построений на уровень графической карты, обладающей мультипотоковой архитектурой, в несколько раз превосходящей по количеству ядер архитектуру центрального процессора.

Литература.

1. Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 (8th Edition). Addison-Wesley Professional; 8 edition, 2013.

2. David Wolff. OpenGL 4 Shading Language Cookbook - Second Edition. Packt Publishing; Revised ed. edition, 2013.

3. Muhammad Mobeen Movania. OpenGL Development Cookbook. Packt Publishing, 2013.

4. Elmar Eisemann, Michael Schwarz, Ulf Assarsson, Michael Wimmer, "Real-Time Shadows First Edition", A K Peters/CRC Press, 2011.