УДК 378.147+004

РЕСУРСЫ ОТКРЫТОГО ОБРАЗОВАНИЯ 3D-ВИЗУАЛИЗАЦИИ И ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ ПО ТЕХНОЛОГИИ OpenGL

С.А. Тренин, студент

Новосибирский государственный университет, г. Новосибирск, Российская Федерация

Ключевые слова: ресурсы открытого образования, средства обработки графических данных, библиотека OpenGL, 3D-визуализация, параллельное программирование, графические примитивы, рендеринг, шейдеры.

Аннотация. Замыкание очередного витка технологической спирали связано с внедрением в повседневную практику искусственного интеллекта и средств генерации нового контента. Рассматриваются некоторые средства 3D-визуализации и параллельной обработки данных по технологии OpenGL.

Свершившаяся третья технологическая революция (по Д. Беллу, Л.Е. Гринину) определяет новые отношения в сфере практической инженерной деятельности. Наличие профессиональных компетенций, соответствующих успешно приобретенным знаниям, составлявшим основу инженерного образования в индустриальную эпоху, в настоящее время противостоит возможности установки необходимого комплекта дополнительных средств обработки в системе автоматизированного проектирования. Преподаватели дисциплин, предполагающих изучение фундаментальных основ инженерной графики, начертательной геометрии, черчения, находятся в ситуации антагонистического противостояния с гигантами компьютерной и программной индустрии, продвигающими очередные релизы средств автоматизированной обработки графических данных.

Одним из немногих механизмов реализации социальной ответственности IT-корпораций является предоставление «открытого» доступа к продвигаемым ими технологиям. Основным инструментом реализации такого доступа выступает поддержка на уровне производителя учебных программ и курсов по освоению их продуктов. По сути, лишь в этом контролируемом трансна-

циональным производителем окне возможностей реализуется творческая активность индивидуума.

Открытые образовательные ресурсы, тематически охватывающие задачи начертательной геометрии, инженерной графики и дизайна, вводят новые понятия в описании задачи построения изображения, не содержавшиеся в классических образовательных дисциплинах. Рассмотрим, к примеру, понятие «рендеринг». В большинстве случаев под рендерингом понимают процесс обработки исходных цифровых данных, составляющих описание модели, приводящий в итоге к построению визуального представления на устройстве отображения. В реализации интерфейса прикладного программирования OpenGL средствами объектно-ориентированного дизайна от NVIDIA рендеринг не выделяется как отдельный метод. Средствами получения изображения по данным модели здесь являются методы обработки событий, связанных с изменением положения и ориентации объекта на сцене, положения, интенсивности, спектра и параметров рассеяния света от источника освещения, представления текстуры объекта. Обработка визуального представления в интерфейсе прикладного программирования OpenGL делится на операции различного уровня детализации. Детальная обработка реализуется средствами программирования шейдеров, спецификация которых поддерживается группой Kronos Group [1]. Данным ресурсом поддерживается спецификация более 24 выпусков OpenGL API, реализованных с 1994 года по настоящее время.

Исходным тезисом в описании процесса обработки данных в библиотеке OpenGL является утверждение, что библиотекой реализуется лишь обработка данных в памяти графического процессора (GPU). Скромность заявленных амбиций разработчиков не должна вводить в заблуждение пользователей: на самом деле сформулированная таким образом цель предполагает перенос процесса обработки данных с центрального процессора вычислительной системы на многоядерное устройство параллельной обработки, которым является типичный современный GPU. Реализованная в формате 8-дюймового планшета систе-

ма на чипе NVIDIA Shield, например, содержит 192-ядерную графическую карту. Существенным компонентом поддержки параллельной обработки является создание адекватной парадигмы параллельного программирования, которая применительно к задачам визуализации должна быть также достаточно универсальной, гибкой и простой в освоении. Последнее требование представляется особенно важным для систем с открытым кодом: здесь успех технологии однозначно определяется объемами создаваемого на ее основе программного продукта.

Для понимания возможностей эффективного программирования устройств GPU представляется необходимым проанализировать основы технологии параллельного программирования этих устройств.

Графическими примитивами в OpenGL являются точка, отрезок, ломаная, прямоугольник и многоугольник. Графические примитивы применяются для определения вершин многоугольного тела в пространстве, представляющего моделируемый объект. При этом понятия вершины, ребра, грани тела являются естественными геометрическими понятиями, определения которых не отличаются от принятых в начертательной геометрии.

Визуализация объекта состоит в построении по данным моделирования (массиву координат вершин, дополненному описанием графа смежности) дополнительной информации об основном и вторичном цвете, глобальных координатах вершин объекта, направлении нормали к грани, координатах текстуры и других атрибутов изображения.

Процесс обработки разбивается на четыре основных этапа:

1. Переход от локальных координат вершины объекта и направления нормали к координатам и направлению в сценическом пространстве.

2. Расчет параметров освещенности отдельных точек в зависимости от взаимного положения объекта и источника света.

3. Наложение текстуры на поверхность объекта.

4. Определение и исключение невидимых объектов и частей объекта.

Повышение производительности OpenGL обеспечивается реализацией параллельной подготовки и обработки атрибутов изображения для всех вершин объекта.

Параллельная обработка предполагает реализацию перечисленных этапов с помощью программирования шейдеров средствами языка OpenGL Shader Language. Существенное увеличение гибкости процесса обработки с помощью шейдеров обеспечивается доступностью данных обо всех вершинах объекта в программном модуле шейдера. Благодаря этому жесткая последовательность прохождения первых трех этапов обработки может быть изменена произвольным образом в соответствии с задачами разработчика.

Применение OpenGL Shader Language сохраняет также возможности обработки высокого уровня программным интерфейсом OpenGL. Интеграция модулей OpenGL Shader Language в программном интерфейсе включает в себя процедуры загрузки и трансляции шейдеров, передачи в модули шейдеров данных о положении объекта, наблюдателя и источника света и высокоуровневую подготовку этих данных. С помощью объектно-ориентированного программирования реализуется взаимодействие между объектами высокого уровня и шейдерами, описывающее: создание вида объекта, описание текстуры, описание преобразования объекта, описание тесселяции, геометрических и вычислимых шейдеров.

Подводя итог вышесказанному, отметим, что возможности открытого образования в сфере 3D-визуализации и параллельной обработки данных, с учетом развития автоматической генерации параллельных программ, позволяют получить необходимые навыки практической инженерной деятельности, которые не только соответствуют требованиям времени, но и поддерживают применение перспективных технологий.

Список литературы

1. Портал Khronos Group [Электронный ресурс]. – Режим доступа: http://www.khronos.org/registry/OpenGL/index_gl.php