



Визуализация сцены в C3D Vision

Основные характеристики и особенности работы

Эдуард Максименко

В настоящее время в качестве графической библиотеки очень широко используется программный компонент OpenGL, который фактически стал стандартом для визуализации сцены, в том числе и в трехмерном пространстве. Его применение и реализация доступны практически на всех графических платформах под различными операционными системами, в том числе UNIX и Windows. Производители графических адаптеров также занимаются аппаратной поддержкой и создают специализированные ускорители, которые работают в связке с OpenGL. Но при всех своих возможностях OpenGL не имеет структурного API для описания сцены, а его интерфейс предоставляет лишь базовые средства для трехмерного рендеринга. Это весьма затрудняет применение разработчиками программного обеспечения API OpenGL без использования вспомогательного кода, который обычно реализуется собственными силами и требует много времени и ресурсов. В связи с этим существует явная потребность в создании инструмента более высокого уровня, который бы обеспечивал разработчиков определенными средствами структурного описания сцены визуализации, а также обладал набором необходимых инструментов для интерактивного взаимодействия со сценой. Модуль визуализации C3D Vision — это один из компонентов от компании C3D Labs, предоставляющий такое решение.

Следует подчеркнуть, что стандарт OpenGL не связан с оконной системой, ввиду чего он получил весьма широкое распространение в открытых системах. Спецификация OpenGL использует расширение GLX, которое принадлежит протоколу ядра X Window System и обеспечивает взаимодействие OpenGL и X Window. Данное расширение организует прямой рендеринг в обход X-сервера, что позволяет реализовывать эффективные приложения, работающие в распределенной X-среде. Все эти возможности использованы в настоящем инструменте от компании C3D Labs. Применяя открытую модульную архитектуру, C3D Vision предлагает доступ к собственному API при наличии мощного и гибкого функционала и широкого перечня вспомогательных компонентов.

Визуализация данных в C3D Vision

C3D Vision представляет собой набор функциональных «кирпичей», из которых с минимальными трудозатратами строятся полноценные графические приложения. В модуле визуализации C3D заложена возможность масштабирования архитектуры, благодаря чему разработчик может создавать собственные классы объектов, наследуя их от уже

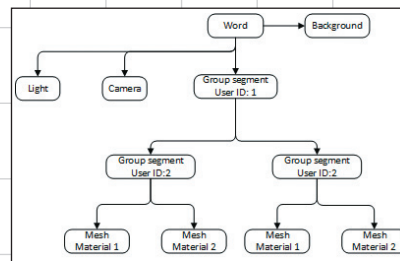


Рис. 1. Обобщенная схема представления сцены в C3D Vision

имеющихся, и тем самым закладывать в них свои свойства и правила. По мере необходимости можно задавать классы вплоть до отрисовочного представления объекта в сцене.

Графические приложения, созданные на базе C3D Vision, работают с геометрическими представлениями сцены, а именно с отрезками, поверхностями, ребрами и прочими объектами, которые отрисовываются в активном окне. Непосредственное взаимодействие пользователя с графическим приложением имеет некое представление отображения сцены. В случае если представление не имеет иерархической структуры, а, предположим, представлено простым линейным массивом объектов, то это затрудняет применение эффективных методов, например отсечения невидимых объектов или поиска объектов сцены. Очевидно, что применение представления сцены с иерархической структурой дает ряд преимуществ. В этом случае функциональные возможности объектов сцены очень похожи и их можно выделить в отдельные группы, которые объединяются в группы более высокого уровня. В свою очередь, применение иерархических принципов неизбежно приводит к известному графическому стандарту PHIGS (Programmer's Hierarchical Interactive Graphics System) — рис. 1.

Эдуард Максименко

К.т.н., руководитель разработки C3D Vision.

Именно иерархические принципы были применены при разработке модуля визуализации C3D, поэтому описание сцены в нем представлено в виде графа, имеющего терминальные узлы или группы узлов в качестве объектов C3D Vision. Поскольку при таком представлении сцены в качестве узлов в графе выступают сегменты, было введено определение «сегментация сцены». Сегмент, не имеющий родителя, называется корневым (от англ. — root). Сегменты могут группироваться по свойствам: материалам, формам и т.д. Кроме того, имеется возможность создавать набор сегментов, в частности шаблонных групп, время жизни которых определяет пользователь. Это может быть набор манипуляторов, заранее созданных для редактирования определенных сегментов, или что-то еще. Надо заметить, что граф сцены принадлежит контейнеру, имеющему ряд необходимых функций для работы с графом, а также отрисовочный канал. Но, пожалуй, одной из самых важных характеристик C3D Vision является то, что пользователь никак не ограничен в создании множества подобных контейнеров. Это означает, что можно сформировать несколько независимых друг от друга графов, отличающихся по целому перечню значимых критериев.

Такой подход гарантирует высокую гибкость в работе визуализатора и дает возможность детально настраивать его для обеспечения максимальной производительности рендеринга. Однако применение сегментации сцены обусловлено не только удобством работы с разделенными на составляющие объектами, но также необходимостью реализации других задач, в том числе:

- изменение параметров сегментов — перемещение и вращение объектов, переключение и проверка видимости, настройка материалов и источников освещения и т.д.;
- перестроение дерева модели любым удобным образом — создание и удаление сегментов или объединение с другими сегментами графа;
- обход графа сцены с выполнением необходимых действий для каждого сегмента;



- отрисовка всей сцены при помощи OpenGL.

Сегментация сцены

Для того чтобы разработчик мог работать с объектами по определенному шаблону и имел возможность при необходимости создавать собственные типы объектов, в C3D Vision реализована сегментация сцены в виде направленного ациклического графа. Подобная организация графа позволяет создавать логическое представление сцены для более эффективного рендеринга (рис. 2).

Сегменты графа могут содержать геометрические данные либо ссылку на эти данные для последующей их отрисовки,

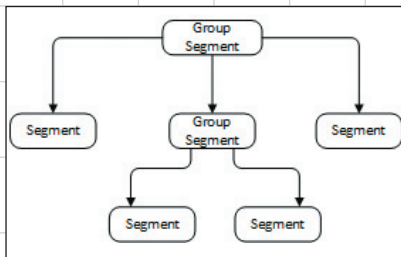


Рис. 2. Представление сцены в виде направленного ациклического графа

при этом применение ссылочной геометрии дает весьма значительную экономию оперативной и видеопамяти. Промежуточные сегменты графа позволяют группировать подсегменты и совершать над ними определенные действия. Надо сказать, что каждый

сегмент графа имеет собственную матрицу для преобразования, влияющую на все входящие подсегменты. Изменяя эту матрицу, можно менять местоположение геометрии в пространстве: осуществлять поворот, сдвиг и т.д. Каждый сегмент графа сцены имеет собственную систему координат, при этом матрица сегмента трансформирует полигональные модели, которые заданы в системах координат подсегментов, в собственную систему координат рассматриваемого сегмента. Поскольку произведение всех матриц от текущего сегмента до root-сегмента графа сцены образует матрицу преобразования из локальной системы координат текущего сегмента в мировую систему координат, то в рассматриваемом случае глобальная система координат будет связана с root-сегментом. В качестве примера сегментации сцены можно привести отрисовку 3D-модели экскаватора с поворотной башней и подъемным ротором (рис. 3).

На основе этой модели можно воспроизвести граф сцены. Рассмотрим только крупные механизмы 2-3-4-6 (рис. 4), то есть произведем сегментацию сцены по ключевым фрагментам.

Из рис. 4 видно, что сегмент гусеничной платформы наследуется от root-сегмента и содержит матрицу перемещения. При трансформации перемещения изменение этой матрицы будет влиять на все подсегменты, поэтому экскаватор будет перемещаться по ходу движения. Соответственно, трансфор-

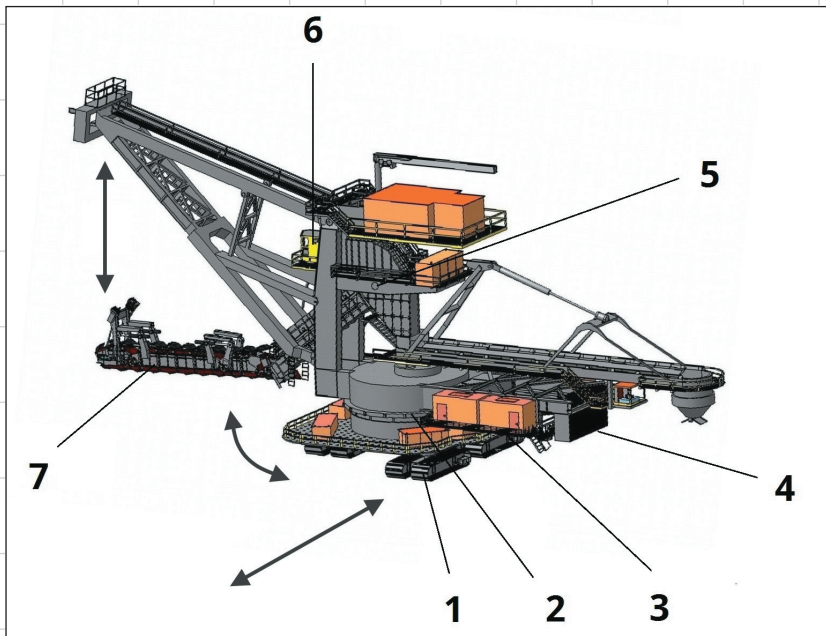


Рис. 3. Модель роторного экскаватора с поворотной башней и подъемным рабочим органом: 1 — гусеничная платформа; 2 — опорно-поворотный механизм; 3 — поворотная платформа; 4 — противовес; 5 — башня; 6 — кабина; 7 — роторный рабочий орган

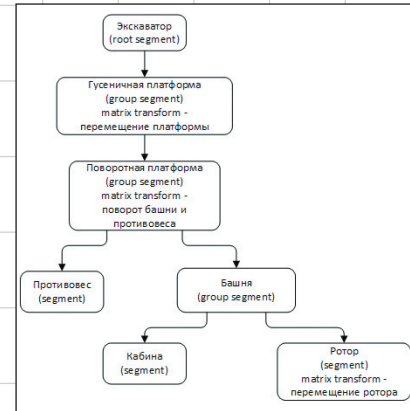


Рис. 4. Сегментация сцены роторного экскаватора

мация по матрице поворотной платформы будет поворачивать всю группу, включающую подсегменты. Группу сегментов, образующих подграф всего графа сцены, можно рассматривать как отдельный объект, состоящий из множества входящих объектов. Например, автомобиль с креслами может быть представлен неделимым объектом, хотя на самом деле состоит из множества геометрических объектов.

С помощью такого подхода можно решить ряд полезных задач по управлению местоположением объектов внутри сцены. Достаточно преобразовать матрицу одного промежуточного сегмента, чтобы изменить местоположение в сцене сразу у всех входящих сегментов, а следовательно, у всей геометрии. Таким образом, если решается задача перемещения автомобиля, то всё содержимое в нем — кресла, руль, педали, водитель с пассажирами — тоже перемещается без организации каких-либо препятствий для выполнения иных локальных задач. Здесь необходимо подчеркнуть, что любые действия над сегментом в основном распространяются на его подсегменты.

Как уже отмечалось ранее, ссылочная геометрия может быть применена для экономии памяти. Граф сцены обычно включает достаточно большое количество подграфов, а те, в свою очередь, несколько потомков, образующих дочерние подграфы. Наконец, последние могут иметь собственную геометрию. Если рассмотреть множество всех существующих моделей, то становится очевидным, что большинство из них не отличается по геометрии. Тот же самый автомобиль имеет четыре колеса одинаковой формы и размеров, поэтому рационально хранить такую геометрию в одном экземпляре, а для определения различий в местоположении колес использовать индивидуальные матрицы. С учетом изложенного в C3D Vision

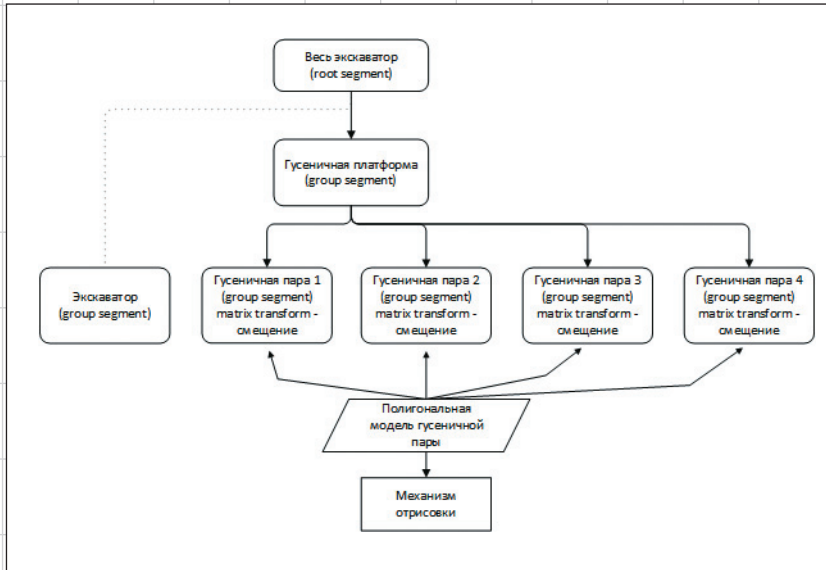


Рис. 5. Сегментация сцены с применением ссылочной геометрии

реализована возможность присвоения ссылок на геометрию тому или иному сегменту графа сцены (рис. 5).

Одно из наиболее важных свойств графа сцены — это хранение состояния отображения каждого сегмента. В данном случае речь идет о подготовке OpenGL перед отрисовкой по иным свойствам сегмента, таким как текстуры, шейдеры и материалы. Стоит отметить, что задание ряда свойств сегмента распространяется и на подсегменты, то есть на весь подграф.

При помощи сегментации сцены можно решить и некоторые глобальные задачи, имеющие отношение ко всему графу сцены в целом. В первую очередь это оптимизация рендеринга. Очевидно, что чем больше объектов необходимо отрисовать, тем больше машинного времени потребуются, чтобы рассчитать и отобразить изменения сцены на мониторе. Нетрудно заметить, что в течение длительного промежутка времени большое количество объектов находится за пределами области экрана или внутри других объектов. Поэтому вычислением таких объектов можно пренебречь, сделав их невидимыми и просто убрав из вывода на экран. Безусловно, это положительно скажется на производительности рендеринга сцены, ведь если объект не попадает в область видимости виртуальной камеры, то и нет необходимости отправлять его на отрисовку в видеоадаптер. С помощью графа сцены можно легко находить такие объекты. Для этого в каждом сегменте вычисляются ограниченный прямоугольник или сфера. Далее к нему добавляются все дочерние подграфы и сегменты, и уже непосредственно

перед отрисовкой сцены вычисляется пересечение ограничивающей сферы сегмента сцены с так называемой пирамидой видимости объектов. В результате строится сортированный список всех видимых сегментов графа сцены. Данный механизм отсечения незначительных элементов сцены характерен для режима оптимизации Frustum Culling.

Таким образом, оценивая преимущества, предоставляемые пользователю базовым функционалом сегментации сцены, можно отметить следующее:

- представление пользовательских форм сводится к определенному шаблону, что упрощает создание пользователем собственных форм, поскольку исключает использование более низкого уровня представления;
- за счет использования концепции графа пользователь может создавать сегменты сцены в виде слоев с их последующей настройкой. Фактически граф сцены представляет собой набор слоев, каждый из которых может быть задан невидимым, полупрозрачным или заблокированным (доступным только для чтения). Надо отметить, что между слоями и группами может не быть никакого внутреннего различия в структурном представлении, так как и слои и группы представлены сегментами сцены;
- логическая связь между объектами модели (кресла в машине) представляется как расширение модели (машины), при этом сегментация сцены может описывать и пространственные отношения различных объектов;
- в больших приложениях при проектировании модели предъявляются повышенные

требования к использованию оперативной и видеопамати, так как их экономное расходование является определяющим в вопросах оптимизации вычислений. Для этих целей в C3D Vision был реализован механизм работы со сценой при помощи ссылочной геометрии;

- применение иерархического подхода в построении графа сцены открывает возможности для решения глобальных задач, например таких, как пространственное разбиение геометрии BVHs, включая эффективное отсечение и быстрое определение столкновений объектов сцены.

Освещение сцены

Реализация света в модуле C3D Vision выполняет ту же роль, что и реальные источники света, поскольку она делает объекты видимыми. Именно источники освещения определяют, какая часть модели будет участвовать в проекции сцены на плоскость, будь то экран компьютера или дисплей мобильного телефона. Самых источников освещения может быть несколько, а соответствующие им объекты C3D Vision способны имитировать различные эффекты освещения. Все объекты смоделированы на основе поведения реальных источников света. Важная особенность — сцена должна иметь хотя бы один источник света, чтобы объекты виртуальной сцены стали видимыми.

Сегодня в C3D Vision реализовано несколько типов источников освещения:

- **Point Light** — реализует освещение сцены, аналогичное ближнему источнику света, при этом источник освещения занимает определенное местоположение и испускает свет из этого положения (функция SetPosition). Объекты на сцене также освещаются в зависимости от их положения и расстояния относительно источника света. Можно не только задать параметры затухания, определяющие интенсивность ослабления источника света в зависимости от расстояния, но и установить значение константы или линейной/квадратичной интерполяции для затухания источника света;
- **Direction Light** — реализует освещение сцены, аналогичное удаленному источнику света. Направление источника света, как и в случае с ближним источником света, определяется с помощью функции SetPosition, но уже без указания конкретного местоположения;
- **Spot Light** — имеет местоположение и направление и реализует освещение сцены, подобное источнику Point Light. В данном случае свет проецируется в конусо-



бразную область (функция SetSpotCutoffAngle), а ее значение устанавливается в радианах.

Виртуальная камера

Вследствие большого распространения телевидения и различных видеоигр, в наше время всё больше людей воспринимают изображение как пространство, отраженное в видеоскителе виртуальной камеры. Но в OpenGL реализации подобного рода камеры не существует. Поэтому, чтобы сформировать иллюзию движения или вращения в сцене относительно зрителя, такую реализацию необходимо создать.

С помощью камеры в C3D Vision можно получить практически любое изображение. Камера является устройством, которое захватывает и отображает мир наблюдателю. Путем манипуляции через представленный функционал камеры можно задать уникальные настройки и действия, в частности вращение

вокруг всей сцены по орбите или относительно начала координат, панорамирование, масштабирование, вращение вокруг оси взгляда и т.д.

Модуль визуализации C3D имеет ряд инструментов, которые могут управлять местоположением камеры. Все эти инструменты являются наследниками от библиотечного класса Process, который принимает события от мыши и клавиатуры и интерпретирует действия пользователя при работе с данными устройствами как действия непосредственно с камерой. Помимо реализации уже готовых инструментов, разработчику программного обеспечения на базе C3D Vision предоставлена возможность создавать на основе класса Process собственные программные инструменты, необходимые для управления камерой.

Заключение

В данной статье были рассмотрены основные принципы по-

строения сцены, а также ключевые особенности функционала программной библиотеки C3D Vision. Тем не менее значительная часть предоставляемых возможностей осталась за пределами анализа: поддержка шейдеров, выбор объектов сцены и их детализация с применением технологии LOD, позволяющей на большом удалении от камеры использовать более простые полигональные модели, а также поддержка полупрозрачных объектов, pixel culling и многое другое. Освещение данных вопросов требует особого внимания, поэтому они будут рассмотрены в последующих статьях.

Стоит отметить, что одной из наиболее важных особенностей C3D Vision является возможность масштабирования программного кода, то есть создание пользователем собственных объектов и интерактивных процессов, позволяющих расширить инструментальные возможности исходной библио-

теки. Модуль визуализации C3D Vision базируется на геометрическом ядре C3D Modeler. Это означает, что при использовании в проекте математической части из набора C3D Toolkit для разработчиков инженерного ПО ее сопряжение с объектами C3D Vision поддерживается напрямую, что существенно облегчает создание приложений.

Модуль визуализации C3D Vision постоянно развивается, совершенствуется его базовый функционал и разрабатываются новые фичи, исходя из реальных запросов и потребностей отрасли САПР. Это дает основания полагать, что компонент C3D Vision будет оставаться востребованным для решения текущих задач разработчиков программного обеспечения и не потеряет своей актуальности в будущем.

Более подробно ознакомиться с C3D Vision и уточнить условия его использования можно на сайте www.c3dlabs.com.

НОВОСТИ

В новый год — с новыми знаниями по геометрическому моделированию! C3D Labs представляет букварь для будущих разработчиков инженерного софта



C3D Labs, дочернее подразделение группы АСКОН и резидент ИТ-кластера «Сколково», поздравляет преподавателей и студентов учебных заведений с наступающим новым годом и объявляет о выходе обновленного учебника по геометрическому моделированию от руководителя разработки C3D Toolkit, к.т.н. Николая Николаевича Голованова.

В учебном пособии, вышедшем в издательстве «КУРС», широко представлен математический аппарат современных САД-систем, а также подробно описана работа с численными моделями геометрических 2D- и 3D-объектов. В книге изложены методы создания кривых, поверхностей и тел, приведены алгоритмы и структуры данных геометрических объектов, изложены принципы, обеспечивающие взаимосвязь элементов 3D-модели, рассмотрен состав геометрической модели и области ее применения и многое другое.

Издание представляет собой значительно переработанную и дополненную версию предыдущих двух книг Николая Николаевича и является отличным дополнением к уже реализуемой компанией C3D Labs программе поддержки инженерного образования C3D ELP, которая включает льготную лицензию на C3D и доступ к технической документации.

Н.Н. Голованов, руководитель разработки C3D: «Основной моей деятельностью является работа над геометрическим ядром C3D. Начав заниматься разработкой, я столкнулся с дефицитом книг о математическом аппарате САД-систем. Тогда я решил написать такую книгу сам. Так появилась книга о кривых, поверхностях, телах, топологии, геометрических ограничениях и геометрических моделях. Со временем книга стала учебником, помогающим мне и моим коллегам в повседневной работе. Надеюсь, она будет полезна студентам и другим разработчикам САПР».

По вопросам приобретения обращайтесь в издательство «КУРС» или книжные магазины вашего города.

Учебник предназначен для студентов высших учебных заведений, обучающихся по специальности «Информатика и вычислительная техника», и специалистов по системам автоматизированного проектирования. Допущено в качестве учебного пособия Учебно-методическим объединением вузов Российской Федерации по университетскому образованию.

