

УДК 004.932

ИНТЕРАКТИВНЫЙ ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ДЛЯ СОЗДАНИЯ ФУНКЦИОНАЛЬНО ЗАДАНЫХ ОБЪЕКТОВ

С.И. Вяткин, А.Н. Романюк

Институт автоматизации и электрометрии СО РАН
Винницкий национальный технический университет

Разработан способ визуального представления свободных форм и аналитических возмущений для интерактивного предварительного моделирования. Создан интерактивный редактор сцен с возможностью сохранения результата, как в виде файла сцены, так и в виде растрового изображения.

Введение

Дальнейшие направления развития систем визуализации определяются как повышение качества изображений за счет производительности, как пиксельной, так и геометрической (геометрическая производительность выражается в количестве геометрических примитивов обрабатываемых за кадр). Сейчас, когда размеры полигонов стали сравнимы с размерами экранной точки – пикселя, невозможно существенно повысить производительность (а, следовательно, и качество изображения), используя лишь выработанные идеи. Поэтому качественного скачка следует ожидать в другом подходе. В использовании новых подходов и в комбинации с лучшими свойствами известных методик.

Алгоритмы на основе полигонов являются самыми распространенными на данный момент, получив широкое распространение в силу аппаратной поддержки. Они применяются для генерации изображений в системах реального времени, и используют полигональный способ задания объектов сцены, т.е. представление трехмерных объектов в виде набора многоугольников, совокупность которых образует желаемую поверхность, что даёт возможность отображать широкий класс произвольных моделей. В то же время такой способ задания объектов приводит большому расходу памяти и вычислительных ресурсов.

Алгоритмы отслеживания лучей используются для создания высококачественных изображений, близких по уровню реалистичности к фотографическим изображениям. В основе алгоритмов отслеживания лучей лежит попиксельное сканирование

екранної площини. Основне достоїнство такого підходу – висока реалістичність получаемих зображень, недолік – час генерації, який може займати від декількох хвилин до сотень годин.

Більш детально алгоритми першого і другого типу, їх достоїнства і недоліки описані в [1,2,3].

До основних достоїнств воксельно – базисованих алгоритмів [4, 5] можна віднести: ефективність методу растрингування, поєднуючого простоту обчислення з швидким пошуком і відбракуванням областей, не зайнятих об'єктами сцени; зменшення кількості поверхонь для опису криволінійних об'єктів. Задання об'єктів поверхнями вільних форм скорочує в 500 і більше раз опису баз даних порівняно з заданням їх полігонами. Знижується навантаження на геометричний процесор і зменшується потік даних від нього до відеопроцесора. Проста анімація, деформація і морфінг об'єктів. Можливість реалізації ефектів, пов'язаних з внутрішньою структурою об'єктів. Які отримуються в процесі побудови зображення природним шляхом.

При цьому ці алгоритми за критерієм якість/час генерації знаходяться посередині вищеописаних: за часом генерації порядку декількох секунд, якість наближується до фотореалістичних зображень, получаемих з допомогою алгоритмів відстеження променів. При використанні графічних акселераторів можна отримати околореальний час [6].

Постановка задачі

Процес створення об'єктів у вигляді неявних функцій, без засобів автоматизації моделювання, досить трудомісткий – в процесі моделювання, людина має справу не з об'єктом, який він може побачити, а з його функціональним представленням. В той же час недостатність наглядних базових примітивів ставить проблему пошуку методів їх додаткового конструювання і представлення. Як правило, для простих поверхонь, представлених квадратами і аналітичними вираженнями процес моделювання йде за схемою

- зміна об'єктів (розмір, положення, тип і т.д.)
- генерація зображення
- візуальна оцінка результату
- попередні кроки повторюються до отримання потрібного зображення

Данный процесс осложнен тем, что, даже при использовании алгоритма генерации изображения, оптимизированного под конкретную конфигурацию компьютера, время генерации составляет несколько секунд, что недостаточно для обеспечения интерактивности процесса моделирования.

В связи с вышесказанным, необходимо было решить следующие задачи:

А. Разработать способ визуального представления свободных форм и аналитических возмущений для интерактивного создания и редактирования сцен. Создать инструмент предварительного моделирования, (закрывающегося в указании их типа, положения, размера, и т.д.) с последующей передачи данных для непосредственной генерации изображения. Реализовать возможность сохранения результата, как в виде изображения, так и в виде файла сцены – для последующей доработки.

В. Расширить набор примитивов для построения сцен.

Предлагаемое решение (краткое изложение работы)

В данной работе предлагаются следующие решения поставленных задач А и В:

А. Как показывает опыт разработки редактора трёхмерных сцен [7, 8], для полноценной работы, до финального построения изображения, не нужно представлять точную картину пользователю, в ущерб скорости взаимодействия со сценой. Вместо этого объекты представляются приближёнными методами, позволяющими отразить все их основные параметры. С помощью предварительной генерации с использованием традиционных полигональных методов, (что позволяет использовать аппаратную поддержку) наглядно представить прототипы объектов пользователю, при этом используется тот же набор примитивов, что и в воксельной системе. Объекты, представляющие аналитические функции возмущения, рисуются каркасными моделями, чтобы отличать их от квадрик.

Создание редактора сцен, в котором представление структуры объектов сцены в виде дерева позволяет удобным образом создавать и удалять объекты, меняя динамически их свойства.

Для возможности последующего возвращения к работе разработан формат сохранения данных сцены, которые можно сохранить в виде файла на диске, для последующего возвращения к работе.

В. Добавление в качестве базовых примитивов (к имеющимся примитивам: эллипсоиду и цилиндру) куба и тора, с сохранением основных свойств функционально заданных объектов.

Иерархия объектов сцены посредством дерева

Модель представления объектов сцены реализована в виде дерева. Программному представлению такой модели на языке C++, так как он является объектно-ориентированным, соответствует наличие общего – базового для всех объектов класса CTreeElement, единственным экземпляром которого является объект Scene –который не отображается, а служит корнем дерева объектов. Объект Scene имеет статический член типа CRenderOptions, хранящий данные для генератора изображения (уровни детальности, цвет фона, параметры OpenGL: толщину линий для рисования возмущений, число граней для рисования свободных форм).

От класса CTreeElement, с помощью наследования, производится класс CGeomObj, свойством такого представления являются инкапсуляция данных, когда каждый объект имеет свои матрицы свойств, в тоже время полиморфизм позволяет растрирующему алгоритму работать с отображаемыми объектами, не привязываясь к их внутреннему устройству - фактически растеризатор работает только с одним базовым классом, объявляющим виртуальные методы. Эти методы переопределяются для каждого класса объектов системы, которые передают свои параметры (такие как размер, положение, цвет, прозрачность, способ генерации текстурных координат, и массив данных для них) для генерации свободных форм при построении конечного изображения растеризатором.

Для представления деревом так же удобны в работе, реализованные в программе, возможности копировать-вставлять, удалять и добавлять в нужное место выбранные элементы.

Способ сохранения и передачи данных

Независимый от структуры и формата представления данных, модуль программы, сохраняет файл сцены в текстовом виде, что оставляет возможность контроля и ручной отладки параметров.

Текущая реализация сохраняет данные каждого объекта как блок, имеющий заголовок, указывающий класс объекта и, далее, в виде пар «параметр = значение», все его переменные. При считывании файла, по имени класса, динамически создаются экземпляры объектов с некоторыми разумными параметрами по умолчанию, которые

изменяются на данные из файла при продолжении считывания. Параметры по умолчанию необходимы для дальнейшей совместимости версий – отсутствие в файле ожидаемых значений не приведёт к нарушению работы программы. Указание имени класса позволяет отделить свойства материала от свойств объекта в отдельный класс, экземпляром которого обладает каждый объект – вместо хранения этих данных непосредственно переменными класса.

В конце, блоком, общим для всей сцены, сохраняются опции и параметры сцены.

Заключение

Разработан способ визуального представления свободных форм и аналитических возмущений для интерактивного предварительного моделирования. Создан интерактивный редактор сцен с возможностью сохранения результата, как в виде файла сцены, так и в виде растрового изображения. Также расширен набор примитивов для построения сцен; исследованы свойства новых примитивов. При создании редактора была проделана работа по оптимизации алгоритма растривания. В дальнейших планах предполагается продолжить разрабатывать способы интерактивного взаимодействия с построенной сценой - удобный аппарат визуального воздействия.

Литература

1. Д. Роджерс. Алгоритмические основы машинной графики. М.: Мир, 1989.
2. Тихомиров Ю. Программирование трехмерной графики. СПб.: ВHV - Санкт-Петербург, 1998.
3. Д. Фоли, А. Ван Дам. Основы интерактивной машинной графики. М.: Мир, 1985.
4. Sergey I. Vyatkin, Boris S. Dolgovesov, Mikhail A. Gorodilov. Perturbation Functions in Computer Graphics// Modern Instrumentation (MI, ISSN: 2165-9273), Volume 2, Number 2, 2013, pp. 26-32.
5. S.I. Vyatkin, "Complex Surface Modeling Using Perturbation Functions", Optoelectronics, Instrumentation and Data Processing, Vol. 43, No. 3, 2007. pp. 40-47.
6. S.I. Vyatkin. Method of binary search for image elements of functionally defined objects using graphics processing units. Optoelectronics, Instrumentation and Data Processing November 2014, Volume 50, Issue 6, pp 606-612.
7. Sergey I. Vyatkin. An Interactive System for Modeling, Animating and Rendering of Functionally Defined Objects// American Journal of Computer Science and Engineering Survey, Volume 2, Issue 3 Nov-Dec. 2014 AJCSES
8. Веб-дизайн і комп'ютерна графіка. Навчальний посібник. / О. Н. Романюк, Д. І. Кательніков, О. П. Косовець. — Вінниця : ВНТУ, 2007. — 103 с.

Надійшла до редакції 10.04.2015