

Березин С.Б. berezin@microdin.ru

Боресков А.В. alex@garant.ru

Мишулин А.А. valaddin@dialup.ptt.ru

Первов П.В. mcfirst@mail.ru

Шикин Е.В. shikin@graphics.cs.msu.su

Московский государственный университет

КОМПЬЮТЕРНАЯ ГРАФИКА: ЭКСПЕРИМЕНТЫ И РЕЗУЛЬТАТЫ

Описывается опыт ведения с применением новых педагогических технологий «Компьютерная графика», обсуждаются полученные результаты творческой работы студентов.

Ключевые слова: *интерактивность, трехмерность, творчество.*

Дисциплины: компьютерная графика, педагогика

Курс «Компьютерная графика» читается студентам второго курса факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова в IV семестре в течение семи лет. Это немного. И еще свежи в памяти авторов размышления над тем, как именно построить курс, чем наполнить, какие формы контроля выбрать и как подвести к завершающему испытанию студентов, чтобы не погасить у них интереса к предмету и чтобы наилучшим образом могли проявиться затраченные ими усилия. Для воплощения этих целей были выбраны нестандартные формы, как подачи материала, так и итогового контроля. Сформулируем основные правила построения курса «Компьютерная графика».

1. Все лекции должны быть прочитаны за первую рабочую неделю семестра (в течение шести дней).
2. Каждый студент, прослушавший курс, выполняет задание, в выборе которого он совершенно свободен.
3. На выполнение задания отводится пять недель с предоставлением возможности работать над ним в компьютерном классе; подготовленная работа сдается (на дискете) в жестко определенный срок, который оговорен заранее.
4. Для анализа выполненных работ и контроля качества их исполнения привлекаются студенты старших курсов (в последние два года).

Лекционная часть курса разбита на две части - сравнительно легкую, включающую в себя математические основы компьютерной графики, основные методы закрашивания и удаления невидимых поверхностей, и более сложную, в которую входят геометрические сплайны и трассировка лучей. Вместо экзамена в традиционной его форме студентам предлагается написать программу, реализующую обсуждаемые в курсе алгоритмы. За эти годы требования к заданию претерпели некоторые изменения, однако одно условие сохранялось всегда - демонстрация в реальном времени движения трехмерных тел различной степени сложности. И если при первом прочтении курса в типичном задании предполагалось осуществление вращения вокруг осей симметрии простейших правильных многогранников, то сейчас требуется визуализировать движение двух-трех невыпуклых тел с взаимным перекрыванием при условии точечной освещенности. Важно отметить, что, несмотря на постоянное повышение требований к заданиям, подавляющее большинство студентов успешно справляется с их выполнением и в целом уровень студенческих работ год от года растет. Написание работы позволяет студенту подойти к ее выполнению творчески: он сам формирует себе задание и реализует его, исходя из своих возможностей. Указанный элемент творчества хорошо виден при рассмотрении большей части работ, которые сильно варьируются как по содержанию, так и по используемым алгоритмам.

Подобная схема чтения курса и приема экзамена является весьма удобной и несет в себе ряд преимуществ. Сжатое во времени чтение лекций повышает степень усвояемости материала и облегчает его понимание. За время, отведенное на выполнение работы (пять недель), студент получает возможность на практике применить полученные знания, довести их до готового продукта, удовлетворяющего определенному набору требований. В результате материал, воспринимаемый обычно как теоретический, приобретает конкретно-практическую форму работающей программы.

По описанной схеме каждый год выполняется около двухсот работ. Авторы постоянно интересуются мнением студентов относительно принятых форм чтения курса и приема экзамена. По нашим данным большинство студентов относится к этому положительно. Об интересе к предмету говорят постоянные пожелания об увеличении объема курса с целью освещения тех или иных аспектов компьютерной графики, не попавших в курс вследствие ограниченности отпущенного на него времени.

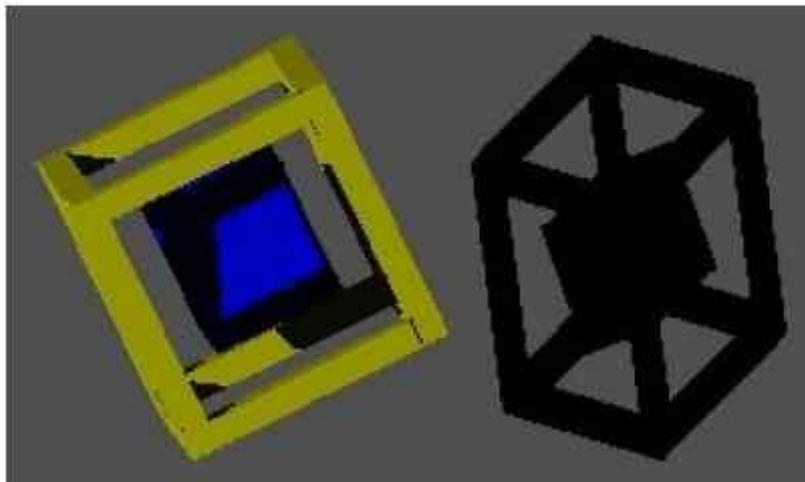
В требованиях на композицию не накладывается никаких ограничений, что дает студентам возможность создавать весьма нестандартные работы. Мы представляем здесь семь работ 1998 года, которые отобрали и прокомментировали студенты пятого курса факультета ВМК.



Лысенко А.

Работа Лысенко А. содержит в себе оригинальную идею - игра в «наперстки». В ней применены следующие алгоритмы:

- алгоритм буфера сегментов (S-буфера), являющийся модификацией хорошо известного алгоритма удаления невидимых линий Z-буфера и заключающийся в сохранении информации о глубине только для концов горизонтальных линий, которые возникают при растеризации многогранника; для каждого следующего многогранника значения глубины на концах линий при растеризации сравниваются с хранимыми в S-буфере;
- самостоятельно реализованный алгоритм закрашивания граней по Гуро.

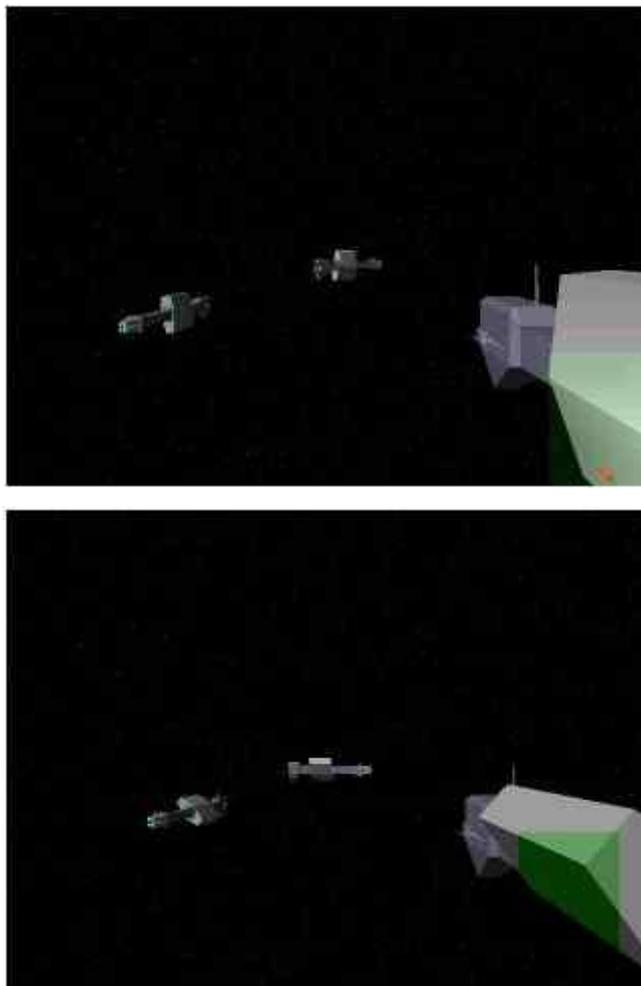


Камзолкин Д.

В работе Камзолкина Д. представлены две полигональные сцены: невыпуклый многогранник и куб в трехмерной рамке. В обеих сценах фигуры вращаются вокруг собственных центров

симметрии. Несмотря на кажущуюся простоту, автором реализован целый ряд алгоритмов трехмерной графики:

- для удаления невидимых линий используется тот же алгоритм S-буфера с небольшим изменением - для каждой линии развертки S-буфер строится отдельно с извлечением из всех многогранников частей, попадающих на данную линию развертки; такой подход несколько замедляет программу, но упрощает реализацию данного алгоритма;
- алгоритм растеризации многогранников сцены с плоским закрашиванием;
- алгоритм расчета освещенности многогранников сцены в зависимости от их положения по отношению к источнику света;
- алгоритм расчета теней от бесконечно удаленного источника не только на некоторую плоскость, расположенную за объектом (что является необходимым требованием к работе), а также всех тел сцены друг на друга (с единственным упрощением - цвет тени не рассчитывается и тень считается черной).



Гогин Г.: динамическая освещенность (подсвечена зеленым).

Гогин Г.: объект без источника света (зеленым выделена ранее освещенная область)

В работе Гогина Г. присутствуют три одинаковых объекта сложной формы с вращающейся средней частью. Образы объектов - космических кораблей - были взяты из известного сериала «Вавилон-5»; несмотря на относительную простоту модели (около 150 граней), сходство с оригиналом очевидно.

Грани объектов сортируются с помощью BSP-дерева, выводятся в дублирующий буфер и впоследствии отображаются на экран. Данный способ вывода позволяет избежать неприятного мерцания изображения на экране монитора. В работе реализована динамическая освещенность (то есть освещенность рассчитывается от движущихся источников света). Выбран достаточно простой метод освещенности (освещенность по Гуро), и одновременно на экране присутствуют не более двух источников света. Кроме движущихся источников света присутствует основной - бесконечно удаленный источник света (он находится слева от сцены). В целях целостности композиции тени не реализованы.

Программа работает в разрешении 640 на 480 в 256 цветах, написана на языке С с использованием ассемблера, собрана на компиляторе Watcom C++ 10.0.



Гирлин В.

В работе Гирлина В. реализованы такие аспекты компьютерной графики, как текстурирование, наложение освещенности и расчет теней.

Объекты в работе динамически изменяются (топология объекта при этом сохраняется) от сильно невыпуклого объекта - многолучевой звезды - до практически правильного многогранника.

В работе создана иллюзия полета над поверхностью, для чего используются текстурированные многоугольники, из которых сложены «пол» и «потолок». Правильно

просчитана освещенность от бесконечно удаленного источника света и тени, отбрасываемые на горизонтальную плоскость («пол»). Программа работает в разрешении 320 на 200 в 256 цветах, перед стартом программа проводит специальные расчеты, которые позволяют изменить параметры движения для того, чтобы ускорить работу на медленных компьютерах и замедлить - на быстрых. Вообще, данная программа весьма нетребовательна к ресурсам компьютера. Программа написана на языке С с использованием ассемблера.



Подбельская Е.

Работа Подбельской Е. демонстрирует реализацию алгоритма построения изображения трассировкой лучей. Программа рассчитывает изображение и сохраняет его по кадрам на диске. Расчет изображения идет весьма долго, но качество получаемого изображения во много раз выше, чем при обычных алгоритмах расчета и вывода изображения.

Построенная сцена весьма сложна, тем не менее, все объекты текстурированы, освещенность рассчитана правильно. Для создания динамики кадры, полученные после работы программы, собраны вместе и обработаны программой Autodesk Animator Pro. Полученный мультфильм показывает сцену в динамике. По заявлению автора программы, при ее создании использовались программы, представленные в книге [1]. В них были внесены необходимые изменения и дополнения, была создана сцена, которая и демонстрируется в программе. Программа создает кадры в режиме 640 на 480 в 256 цветах и написана на языке С.



Киселев П.

Работа Киселева П. состоит из трех исполнимых модулей. Модуль `b_spline.exe` по заданным в файле координатам вершин замкнутой ломаной и целочисленной константе N строит B-сплайн, аппроксимирующий данную ломаную, и формирует новый файл, содержащий координаты точек на сплайновой кривой. При этом каждому отрезку исходной ломаной соответствует N отрезков результирующей ломаной, т.е. шаг между точками уменьшается в N раз.

В рассматриваемой работе в исходном файле `asher.pnt` задана ломаная, грубо аппроксимирующая узел Эшера (Escher). После обработки данного файла программой `b_spline.exe`, был получен файл `asher.crv`, содержащий гораздо более точную аппроксимацию.

Второй модуль программы, `out.exe`, выполняет основную часть работы, а именно строит изображение трехмерного тела, полученного протягиванием окружности заданного радиуса вдоль отрезков ломаной, причем центр окружности лежит на отрезке ломаной, а плоскость окружности перпендикулярна направлению отрезка. Другими словами, происходит формирование цилиндрической поверхности, осью которой является заданная ломаная. Изображение полученного таким образом трехмерного объекта строится методом трассировки лучей с учетом цветных источников освещения и отбрасываемых частями объекта друг на друга теней.

В качестве исходных данных для программы `out.exe` использовался файл `asher.crv`, дополненный информацией о расположении и цвете источников освещения. Радиус протягиваемой окружности жестко задан в исходном тексте программы. Было рассчитано

изображение для 100 кадров, причем в каждом кадре объект смещался в некотором направлении и поворачивался на некоторый угол.

Одно из обязательных требований к работе - наличие динамики. Но метод трассировки лучей, использованный в данной работе, требует очень больших затрат машинного времени, поэтому результаты работы программы out.exe были записаны в файл result.fml, содержащий образы экрана для каждого из ста кадров. Формирование этого файла занимает достаточно много времени.

Третий модуль, film.exe, последовательно отображает считанные из файла кадры на экране. Программа использует режим экрана с разрешением 320 на 200 пикселей и 256 цветами. Вследствие того, что затраты на чтение кадра из файла и отображение его с использованием прямого доступа в видеопамять невелики, то в результате достигается вполне достаточная для динамического изображения частота смены кадров.



Балакирев А.

В работе Балакирева А. решается задача отображения поверхности планеты с возможностью интерактивного перемещения по ней. Программа написана на языке Turbo Pascal. Поверхность задана в виде прямоугольной сетки размером 256 на 256 элементов. Каждому квадрату поверхности соответствуют его цвет и высота. Таким образом, поверхность задается как набор элементарных кубических элементов или вокселей. Такой подход находит применение в программах-имитаторах летательных аппаратов и в компьютерных играх.

Алгоритм построения изображения основан на модифицированном алгоритме трассировки лучей на плоскости, называемом также ray-casting. Для каждого столбца изображения рассчитываются точки пересечения луча, выходящего из точки расположения наблюдателя и проходящего через проекцию данного столбца на горизонтальную плоскость, с проекциями на эту же плоскость элементов поверхности. Перебирая точки пересечения в порядке

удаления от наблюдателя, можно определить цвета пикселей в формируемом столбце изображения.

Возможность интерактивного перемещения по поверхности накладывает на производительность алгоритма достаточно жесткие требования. Поэтому автор программы применяет несколько повышающих производительность приемов. Например, для вычисления тригонометрических функций используются заранее рассчитанные таблицы, а вместо чисел с плавающей запятой - числа с фиксированной запятой, операции над которыми сводятся к операциям над целыми числами.

Исходные данные для алгоритма находятся в файлах `ground.bmp` и `height.bmp`, представляющих собой восьмибитные изображения в распространенном формате BMP размером 256 на 256 пикселей. Каждый пиксель этих изображений определяет характеристики соответствующего элемента поверхности. Файл `ground.bmp` определяет цвет элементов поверхности. В файле `height.bmp` заданы высоты элементов поверхности, причем цвет с нулевым номером задает минимальную высоту, а цвет с номером 255 - максимальную. При просмотре файла `height.bmp` в палитре градаций серого цвета с возрастающей интенсивностью более высоким участкам поверхности соответствуют более светлые участки изображения.

Использование стандартных форматов для файлов с данными позволяет применять существующие программы редактирования изображений для создания новых ландшафтов.

Список литературы

[1] Шикин Е.В., Боресков А.В. Компьютерная графика. Динамика, реалистические изображения. - М.: ДИАЛОГ-МИФИ, 1995.