

СПИСОК ЛИТЕРАТУРЫ

1. Математическое обеспечение графопостроителей. I уровень: СМОГ: Инструкция по программированию/Под ред. Ю. А. Кузнецова.— Новосибирск: ВЦ СО АН СССР, 1976.
2. Вильданов Р. Н., Дебелов В. А., Упольников С. А. СМОГ-85. Графический пакет для ЕС и СМ ЭВМ // Тез. докл. Всесоюз. конф. по проблемам машинной графики и цифровой обработки изображений.— Владивосток: ИАПУ ДВНЦ АН СССР, 1985.
3. Дебелов В. А., Мацокин А. М., Чубарев А. И. Подход к разработке распределенных систем для двухмашинных комплексов // Машинная графика и ее приложения.— Новосибирск: ВЦ СО АН СССР, 1983.

Поступила в редакцию 17 января 1990 г.

УДК 681.3.082.5

И. И. ЛИСИЦЫНА, Т. И. ПЕТРУШИНА, Н. Ф. ТРУБИНА

(Одесса)

МОДЕЛИРОВАНИЕ АРХИТЕКТУРЫ МНОГОПРОЦЕССОРНОЙ РАСТРОВОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ И ОЦЕНКА ЕЕ ПРОИЗВОДИТЕЛЬНОСТИ

Создание высокопроизводительной графической системы реального времени требует обоснования и предварительного исследования предполагаемой архитектуры [1]. Авторы используют для этой цели разработанную ими программную модель архитектуры многопроцессорной растровой графической системы получения трехмерных реалистических изображений. Программная модель позволяет охарактеризовать и проанализировать связи в графической системе, а на основе собираемой статистики получить оценки производительности системы и отдельных ее компонент. Анализ получаемой статистики дает также возможность выявить «узкие» места архитектуры, что является основанием для изменения конфигурации системы и (или) модификации набора команд единичного процессора. Модульная организация программной модели обеспечивает не только гибкость и удобство ее разработки, но и возможность подбирать и испытывать различные конфигурации, соответствующие вариантам архитектуры графической системы.

Графическая система предназначена для получения изображения сцен, содержащих трехмерные объекты, представленные в системе в виде визуализируемых оболочек (в-оболочек). В-оболочка получается в результате триангуляции реального объекта и содержит массив координат точек объекта (вершин), их яркостей и координат нормали в вершине, массив треугольных граней, заданных ссылками на вершины, и некоторые дополнительные характеристики. В-оболочки объектов строятся автономными средствами и накапливаются в базе данных объектов. Программная модель обрабатывает описание сцены, состоящее из набора в-оболочек, извлекаемых из базы данных.

Программная модель имитирует обработку описания сцены следующими компонентами графической системы.

Магистраль геометрических преобразований (МГП), отвечающая требованиям развитых графических стандартов (типа GKS и PHIGS) [2]. Предусматривается модельное преобразование координат вершин каждой в-оболочки из модельной системы координат, в которой задан объект, в мировую систему координат. Далее применяются видовое преобразование (геометрическое преобразование для приведения к норма-

лизованному видимому объему) и *преобразование образа* (преобразование в трехмерную область вывода для центральной проекции). Эти преобразования объединяются в единую обобщенную матрицу 4×4 , позволяющую преобразовывать координаты вершин объекта в пространство нормированных однородных координат. Вычисление обобщенной матрицы преобразования выполняется для каждого объекта в процессе построения описания сцены.

Исходной информацией для МГП является набор в-оболочек объектов сцены и их обобщенных матриц преобразований. Единичный процессор МГП выполняет следующие вычисления:

а) геометрические преобразования координат вершин (умножение вектора координат вершины на обобщенную матрицу преобразования);

б) «грубое» отсечение, предусматривающее исключение из дальнейшей обработки тех граней, ни одна точка которых не принадлежит видимому объему; отсечение выполняется в однородных координатах;

в) ортографическое проецирование и преобразование в пространство приборно-зависимых координат; вычисления сводятся к выполнению операции деления (при переходе от однородных к декартовым координатам) и целочисленному масштабированию (переход к физическим координатам);

г) удаление потенциально невидимых граней, т. е. исключение из дальнейшей обработки тех граней, нормаль к которым образует с нормалью к экрану угол, меньший или равный по абсолютной величине 90° ;

д) вычисление некоторых характеристик граней: яркости в вершинах, z -составляющей нормали грани, коэффициентов плоскости грани и других, которые требуются на последующих этапах обработки.

Вычисления «а»—«в» выполняются в области вещественных чисел, все последующие вычисления в системе — только в рамках целочисленной арифметики.

Все вычисления реализуются с помощью операций сложения, вычитания и умножения, за исключением вычисления на этапе «в», где используется вещественное деление при переходе от однородных нормированных координат к приборно-зависимым, и на этапе «д», где применяется целочисленное деление для вычисления коэффициентов граней и их ребер.

МГП состоит из нескольких единичных процессоров, каждый из которых выполняет описанные выше действия над отдельной в-оболочкой. Число единичных процессоров есть параметр программной модели, который варьируется при испытании различных моделей архитектуры. Программная модель каждого процессора предусматривает однократный последовательный просмотр вершин и граней в-оболочки и их синхронную обработку. Этим уменьшается объем вычислений и обеспечивается распараллеливание дальнейшей обработки по граням.

Процессор визуализации, выполняющий удаление заслоняемых поверхностей, отсечение по границе окна, растровую развертку изображения и раскраску. Архитектура такого процессора, а также его программная модель существенно зависят от алгоритмов, выбранных для его реализации. Авторы разработали три различные программные модели, основанные на алгоритмах [3]: z -буфер, разбиение области и вариант алгоритма построчного сканирования, учитывающего свойство когерентности по видимой части связанной поверхности (предложен авторами).

Программная модель на базе z -буфера предполагает распараллеливание обработки по в-оболочкам, причем эта обработка синхронизирована с работой МГП. Каждый единичный процессор МГП завершает обработку грани в-оболочки выполнением растровой развертки грани и ее раскраску с заполнением z -буфера и буфера раstra. Доступ к z -буферу и буферу раstra осуществляется через видеоконтроллер большой производительности по m каналам, где m — параметр модели. Модель единичного процессора формирует для передачи и обработки видеоконтроллером блоки данных из 16 слов: 4 четверки слов, каждая из которых со-

держит x - и y -растровые координаты точки, ее глубину (z -координата) и яркость.

Отсечение по границе окна для соответствующих граней выполняется в процессе растровой развертки.

На основе программной модели с использованием алгоритма z -буфера разработан вариант архитектуры графической системы, который исследуется в настоящее время. Для получения дополнительных оценок разрабатывается имитационная модель этой архитектуры. Анализ первых полученных оценок по набранной статистике выполнения арифметических операций показывает, что по числу операций умножения и деления модель на базе z -буфера сравнима с моделью на базе алгоритма построчного сканирования, а по числу операций сложения (вычитания) и сравнения на порядок хуже.

Программная модель на базе алгоритма *разбиения области* реализует традиционный алгоритм с делением области рисунка на четыре квадратные подобласти [4] с небольшой модификацией.

Для каждой исследуемой области поддерживается список пересекающих ее граней, также хранится *единственная* (если таковая есть) покрывающая область грань. В случае, если покрывающих граней несколько, запоминается грань с минимальной z -координатой. Вопрос о взаимном расположении грани и плоскости решается с помощью достаточно простого алгоритма, требующего, однако, большого количества арифметических операций. Как показал анализ набранной статистики, число выполняемых операций сложения (вычитания), сравнения и умножения выше на порядок по сравнению с z -буфером и на два порядка по сравнению с алгоритмом построчного сканирования. Ограниченные возможности авторов в выборе аппаратных средств заставили их отказаться от разработки архитектуры по этой программной модели. По мнению авторов, решающим здесь мог бы быть выбор для реализации нетрадиционных средств.

Программная модель на базе *алгоритма построчного сканирования* предусматривает его существенную модификацию. Идея принадлежит С. В. Македону — одному из разработчиков архитектуры графической системы. Алгоритм построчного сканирования позволяет удалять заслоняемые части поверхности в процессе анализа взаимного расположения тех ребер всех объектов видимого объема, которые пересекают плоскость, параллельную плоскости xz и соответствующую некоторой строке раstra. Идея модифицированного алгоритма заключается в том, что для определения заслоняемых частей поверхностей реальных тел достаточно проанализировать взаимное расположение только концов так называемых «цепочек» — наборов ребер, принадлежащих видимой части связной поверхности тела, расположенной вдоль строки раstra. Более того, взаимное расположение цепочек мало меняется при переходе от одной строки раstra к другой, что позволяет следить только за относительными перемещениями цепочек.

Авторам удалось использовать эти свойства и на их основе разработать модифицированный алгоритм построчного сканирования, требующий минимального объема вычислений по сравнению с другими алгоритмами.

Программная модель на базе этого алгоритма предполагает наличие *магистральной растрезации и раскраски* (МРР), состоящей из нескольких (их число — параметр модели) единичных процессоров. Каждый единичный процессор этой магистральной растрезации должен генерировать изображение вдоль некоторого конкретного множества строк раstra. Поэтому такой процессор будет получать для обработки только те грани из всех в-облочек, которые отображаются на это множество. МРР в программной модели организована так, что каждый единичный процессор, завершив обработку очередной грани, определяет, какому единичному процессору МРР она должна быть передана для дальнейшей обработки. Грань передается на обработку процессору МРР, обрабатывающему ту строку

растра, на которую отображается вершина грани, имеющая максимальную в грани y -координату. Взаимодействие между процессорами МГП и МРР осуществляется через контроллер передачи данных (КПД), реализующий, по сути, распределяющую сортировку граней. Когда МГП заканчивает обработку всех в-оболочек сцены, она готова начать обработку описания следующей сцены, тем самым предусматривается конвейерный режим взаимодействия двух магистралей.

Единичный процессор МРР работает в двух режимах.

1. Пока МГП не закончила обработку текущей сцены, обрабатываются поступающие через КПД грани. Для каждой обрабатываемой строки растра строится список ребер тех граней, которые будут разворачиваться в этой строке.

2. Когда МГП заканчивает обработку текущей сцены, единичные процессоры МРР образуют очередь в порядке следования обрабатываемых ими строк растра. В этом режиме единичный процессор выполняет следующие действия. Для текущей строки растра выполняется итеративный пересчет всех цепочек, расположенных вдоль этой строки. При этом если некоторое ребро цепочки, расположенной вдоль предыдущей строки растра, принадлежит цепочке, расположенной вдоль текущей строки, то просто перевычисляется x -координата этого ребра в текущей строке. Если ребро заканчивается в предыдущей строке, то в последующей оно заменяется на список ребер, исходящих из конца ребра (если таковые имеются), построенный в первом режиме работы единичного процессора. На этом этапе могут быть добавлены новые цепочки. После обработки цепочек происходит переупорядочивание их концов по x -координатам. Если в процессе переупорядочивания изменяется взаимное расположение некоторых цепочек, то может измениться взаимная заслоняемость поверхностей объектов, им соответствующих. Поэтому формируется специальный список видимости цепочек строки растра, в котором отмечаются порядок и участки видимости цепочек. При определении видимости цепочки учитывается возможность выполнения отсечения по границе окна.

После корректировки список видимости используется для растровой развертки и раскраски вдоль соответствующей строки растра. Сформированная единичным процессором строка растра записывается в буфер растра.

На основе программной модели с использованием модифицированного алгоритма построчного сканирования разработан вариант архитектуры графической системы, исследования которого продолжаются.

Во всех предлагаемых моделях используется единый метод раскраски — метод Гуро [3] с некоторыми дополнениями, а именно: интенсивность в вершине грани вычисляется в зависимости от косинуса угла между нормалью к поверхности объекта в вершине и направлением на источник света. Нормаль к поверхности объекта в вершине вычисляется при формировании в-оболочки объекта в базе данных объектов усреднением нормалей по всем сходящимся в этой вершине граням объекта. Чтобы не вычислять эту нормаль, поскольку она задана в модельной системе координат, при формировании описания сцены вычисляются координаты вектора направления на источник света в модельной системе координат и передаются вместе с в-оболочкой и обобщенной матрицей преобразования на обработку МГП.

Единичный процессор МГП вычисляет яркость P_i^t в вершине по формуле

$$P_i^t = P_\Phi + (P_i \cos \alpha - P_\Phi) e^{-\beta z_i},$$

где $\cos \alpha$ — скалярное произведение вектора нормали в вершине и вектора направления на источник света; P_i — исходная яркость вершины; z_i — z -координата вершины; P_Φ — яркость фона; β — коэффициент прозрачности среды.

Процессоры визуализации всех трех рассмотренных моделей реализуются на базе только целочисленной арифметики. Для обеспечения требуемой точности вычислений при растровой развертке и раскраске используется алгоритм Брезенхэма [5], позволяющий также ограничиться при вычислениях только операциями сложения и сравнения.

Система имеет гибкую структуру и допускает модификации и расширения. Для поддержки программной модели разработан обширный инструментарий. Имеется система автоматической триангуляции поверхностей канонических объектов: сферы, тора, конуса, цилиндра и т. д.; система построения сложных объектов из канонических; программа построения в-оболочек и ведения базы данных объектов и т. д.

СПИСОК ЛИТЕРАТУРЫ

1. Arnold D., Reynolds G. The use of modularity and configurability in the comparison of graphics systems design // NATO ASI Series, 1988.— V. F40.
2. ISO Information Processing Systems — Computer Graphics — Programmer's Hierarchical Interface to Graphics (PHIGS) // Functional Description, ISO DP9592, 1986.
3. Фоли Дж., ван Дэм А. Основы интерактивной машинной графики.— М.: Мир, 1985.— Кн. 1.
4. Павлидис Т. Алгоритмы машинной графики и обработки изображений.— М.: Радио и связь, 1986.
5. Bresenham T. E. A linear algorithm for incremental digital display of circular arcs // Communs of the ACM.— 1977.— 20, N 2.— P. 100.

Поступила в редакцию 16 января 1990 г.

УДК 629.7.058.74 : 681.3.06

А. М. КОВАЛЕВ, А. С. ТОКАРЕВ

(Новосибирск)

СИНТЕЗ ИЗОБРАЖЕНИЙ БЕЗ ЭЛАЙСИНГА И ОКОНТУРИВАНИЯ

В системах синтеза визуальной обстановки [1] качество изображений во многом определяется средствами антиэлайсинга, устраняющими также визуальные аномалии, как ступеньки на контурах объектов, разрывы тонких граней, мерцания малых объектов, муары и пр., в сочетании с правильным выбором разрядности цифрового кодирования сигналов для плавного, без видимых скачков изменения яркости или цвета на гладких поверхностях объектов.

Франклин Кроу [2, 3] впервые показал, что эффективным средством антиэлайсинга является предварительная фильтрация синтезированных изображений. Джонсон Ян [4] со ссылкой на [5, 6] обратил внимание на то, что фильтр должен обладать свойством равноэнергетичности. И наконец, Эдвин Катмулл [7] показал, что несмотря на фильтрацию, изображение будет иметь элайсинг, если не компенсировать нелинейность модуляционной характеристики воспроизводящих устройств, например электронно-лучевых трубок (ЭЛТ). В то же время в [7] отмечается, что при использовании общепринятого в машинной графике 8-разрядного двоичного кода интенсивности возникают проблемы квантования («оконтуривания») в области малых яркостей.

В книге по интерактивной машинной графике [8] таблицу γ -коррекции для компенсации нелинейности модуляционной характеристики ЭЛТ рекомендуют составлять с учетом логарифмического распределения уровней интенсивности. Такой подход решает проблему «оконтурива-