

**СИСТЕМЫ И МЕТОДЫ АНАЛИЗА  
И СИНТЕЗА ИЗОБРАЖЕНИЙ**

УДК 681.3.06

С. Б. Белов, В. А. Бобков, В. П. Май, Ю. И. Роньшин

*(Владивосток)***СИСТЕМА КОНСТРУИРОВАНИЯ  
И РЕАЛИСТИЧНОЙ ВИЗУАЛИЗАЦИИ  
СЛОЖНЫХ ГЕОМЕТРИЧЕСКИХ ОБЪЕКТОВ**

В рамках конструктивного подхода к описанию геометрических 3D-объектов предложены метод и алгоритмы создания и реалистичной визуализации сложных пространственных сцен. Описана реализованная на этой основе система интерактивной графики, легко настраиваемая на требуемое приложение. Приведены оценки эффективности применения транспьютерной технологии для параллельной обработки графической информации, полученные в процессе разработки «параллельной» версии системы.

**Введение.** Задача создания и реалистичной визуализации пространственных геометрических объектов, являясь традиционной в области машинной графики и САПР, остается актуальной и сегодня. Во-первых, возникающие новые практические задачи требуют новых конструкторских и изобразительных возможностей, во-вторых, с учетом традиционно больших вычислительных затрат для этой задачи постоянно остается необходимость в повышении быстродействия систем за счет как разработки новых методов и алгоритмов, так и применения новейших информационных технологий. В настоящей работе акцент делается на двух аспектах:

— наращивание конструкторских возможностей за счет нетрудоемкого расширения класса используемых геометрических примитивов предлагаемым унифицированным методом;

— повышение быстродействия за счет модификации известных алгоритмов и применения параллельной обработки на базе транспьютерной вычислительной архитектуры.

Описываемая в статье система CG предназначена для конструирования сложных геометрических 3D-объектов и сцен из этих объектов и реалистичной растровой визуализации полученных сцен. Система обладает особенностями, отличающими ее от других работ аналогичного назначения. В основу разработки положены метод конструктивной геометрии, в рамках которого осуществляется конструирование объектов, и метод отображения объектов, основанный на Z-буферном подходе [1].

**Конструирование объектов.** Объект строится из примитивов геометрического типа, о которых будет сообщено далее, с помощью геометрических преобразований и операций объединения, пересечения, вычитания. Класс примитивов в рассматриваемой системе определяется как абстрактный тип данных со следующими операциями:

— нахождение точек пересечения примитива с заданным лучом;

- ортогональные геометрические преобразования примитива;
- нахождение минимальных и максимальных значений (возможно, приближительных) координат примитива по осям трехмерного пространства.

Одним из принципиальных требований к организации системы является обеспечение возможности нетрудоемкого наращивания поддерживаемого системой класса примитивов. Такая возможность реализована благодаря простым требованиям к классу примитивов, сформулированным выше, а также наличию в системе простого интерфейса для работы с примитивами. В настоящий момент в системе реализованы примитивы:

- полупространство,
- шар,
- многогранник,
- цилиндр,
- конус;
- обобщенный цилиндр:
- эллиптический цилиндр,
- цилиндр с образующей, задаваемой специализированной программой геометрических построений на плоскости;
- обобщенный тор:
- выпуклое тело вращения,
- тор,
- конический тор,
- выпуклый тор,
- спираль,
- коническая спираль,
- выпуклая спираль;
- участок обобщенной поверхности.

Примитивы описываются в параметрической форме с помощью программ-функций с фиксированным интерфейсом. Из приведенного списка требуют пояснения примитивы «обобщенный цилиндр» и «обобщенный тор».

«Обобщенный цилиндр» задается вектором своей оси и образующей линией (контур поперечного сечения). Эллиптический цилиндр представляет собой частный случай реализации примитива «обобщенный цилиндр». Для формирования образующей произвольной формы в системе используется специализированная программа «Аппарат конструктора» [2], с помощью которой можно построить в диалоге плоскую кривую произвольной формы.

«Обобщенный тор» описывается осевой линией — пространственной кривой и радиусом кругового сечения, перпендикулярного оси. Радиус может изменяться вдоль осевой линии. Как частные случаи этого примитива реализованы «выпуклое тело вращения», «тор», «конический тор», «выпуклый тор», «спираль», «коническая спираль», «выпуклая спираль».

Сцена конструируется из экземпляров примитивов указанных типов. Из примитивов строятся конъюнкции (пересечения), которые объединяются в объекты. Как конъюнкции, так и объекты могут именоваться, и над поименованными объектами могут выполняться операции (геометрические преобразования, сложение, вычитание объектов). Поименованным объектом, в частности, может служить вся сцена.

В системе реализованы два режима работы — интерактивный и пакетный. В интерактивном режиме процесс конструирования и визуализации объектов поддерживается специально разработанным графическим диалоговым монитором [3] и Макроязыком, а в пакетном режиме — соответствующей библиотекой функций для вызова из программ на языке Си.

Визуализация сцены. Для визуализации объектов разработан алгоритм, основанный на Z-буферном подходе и обладающий некоторыми отличительными особенностями.

Первая особенность заключается в том, что пространственная задача по нахождению Z-расстояния сводится к плоской, что, с одной стороны, упрощает вычисления, а с другой — позволяет реализовать в системе более сложные геометрические примитивы, чем традиционные. Вычисления производятся

следующим образом. Рассчитывается плоское сечение  $ZX$  для строки, далее по сечению универсальным способом вычисляются значения  $Z$  для всех пикселей строки. В случаях простых примитивов для сечения может быть получено аналитическое выражение, в общем случае один или несколько плоских контуров, представленных ломаной. Для определения  $Z$ -координаты в общем случае достаточно одной базовой процедуры, вычисляющей точки пересечения плоского контура — ломаной с прямой.

По двум соседним сечениям  $Z$ -буфера строится строка элементов изображения. Соседнее сечение  $Z$ -буфера при этом используется для вычисления нормали к поверхности. Нормированная яркость в данной точке вычисляется по формуле

$$I = A \cos(Q) / (A + Z),$$

где  $A$  — коэффициент, определяемый из условий: при  $Z = 0$   $I = 1$ , при  $Z = Z_{\max}$   $I = 0$  ( $Z_{\max}$  задается пользователем);  $Q$  — угол между нормалью к поверхности и направлением на источник освещения. Предполагается, что источник освещения находится в бесконечно удаленной точке отрицательной части оси  $Z$  и излучает белый свет; среда абсолютно прозрачна; объекты имеют матовые поверхности с коэффициентом отражения 1.

Вторая особенность рассматриваемых алгоритма и системы — реализация нетрадиционных геометрических примитивов («обобщенный цилиндр», «обобщенный тор», «участок обобщенной поверхности») и возможность расширения набора примитивов стандартным способом, о чем сообщалось выше. Для этого в системе реализован эффективный механизм добавления новых примитивов. Для нового примитива требуется написать стандартную программу добавления в систему, поддерживающую работу с этим примитивом. Программа включает в себя три функции, основной из которых является функция вычисления плоского сечения примитива.

Третья особенность системы связана с заложенными в ней механизмами повышения быстродействия используемых алгоритмов. В процессе программной реализации системы исследовались разные алгоритмы обработки геометрических сцен. Кардинальным решением проблемы повышения быстродействия стала реализация системы с распараллеливанием алгоритма визуализации на основе использования транспьютерной вычислительной архитектуры. Эффективность транспьютерной версии системы характеризуется следующими цифрами. Время визуализации сцены типа «фрагмент архитектурной планировки города» (сложная сцена, состоящая примерно из 50 конъюнкций и занимающая всю площадь экрана  $640 \times 480$  пикселей на IBM PC 80286, 12 МГц) составило 352 с, а для сети из 16 транспьютеров T800 — 14 с.

Достижение реалистичности получаемых изображений также потребовало специальных усилий. Для устройств с ограниченным цветовым разрешением непрерывные значения яркости отображаются в индексы палитры случайной функцией с вероятностью, зависящей от того, насколько близко данное значение яркости находится к дискретному значению яркости для данного индекса палитры. Таким образом моделируется непрерывная шкала яркости. Качество изображения повышается применением алгоритма «размывания» границ интенсивности цвета [4]. Реализованы два алгоритма имитации интенсивности. Первый имитирует 12-уровневую шкалу интенсивности, второй — непрерывную.

Алгоритм визуализации с параллельными вычислениями. Для оценки эффективности применения параллельных вычислений в рамках настоящей работы и в целях повышения быстродействия системы реализована «параллельная» версия системы на базе транспьютерной вычислительной архитектуры. Параллельный алгоритм построен на базе вычислительной схемы алгоритма визуализации последовательного прототипа системы путем применения метода фиксированного разбиения экрана на отдельные области, обработка каждой из которых организуется как отдельный процесс. Такой способ распа-

раллеливания привлекает своей универсальностью и сравнительно небольшими затратами на введение параллелизма [5]. Испытания проводились на транспьютерной сети, связанной с ПЭВМ типа IBM PC/AT с числом транспьютерных модулей до 18. В процессе испытаний варьировались сложность и неоднородность визуализируемой сцены, площадь окна визуализации, способ разбиения экрана на области и количество используемых транспьютеров. Следует отметить, что при такой схеме распараллеливания для каждого отдельного процессора существует некоторая постоянная составляющая времени работы на нем, не зависящая от объема вычислений на этом процессоре. Она складывается из накладных расходов на транспортировку сообщений между процессорами и времени выполнения установочных операций, которые дублируются на каждом процессоре. К установочным операциям относятся геометрические преобразования сцены и вычисление глобальных характеристик сцены. Эта величина становится критической при повышении степени распараллеливания за счет увеличения числа процессоров.

В результате испытаний параллельной версии, реализованной вышеописанным способом на сети транспьютеров, получены следующие общие выводы:

1. Для достаточно простых сцен можно реализовать режим реального времени визуализации сцены путем увеличения числа используемых транспьютеров.

2. Для повышения эффективности распараллеливания, особенно в случае сложных сцен, необходимо уменьшать постоянную составляющую за счет распараллеливания установочных операций или выполнения их на высокоскоростном геометрическом процессоре.

3. При распараллеливании разбиение экрана на горизонтальные полосы более эффективно, чем на вертикальные.

4. Общее время вычислений, как уже было отмечено выше, складывается из двух составляющих — постоянной, не зависящей от объема изображения (предварительные установочные операции), и переменной, непосредственно зависящей от объема изображения. Поэтому с уменьшением площади окна визуализации эффективность распараллеливания уменьшается, поскольку доля постоянной составляющей возрастает.

5. Поскольку неоднородность сцены очевидным образом понижает эффективность распараллеливания, предлагается использовать метод адаптивного разбиения экрана на горизонтальные полосы, что существенно уменьшит неоднородность сцены.

**Функциональные возможности системы.** Основной режим работы системы диалоговый, вместе с тем в ней реализована библиотека функций, доступная для вызова из программ на языке Си, написанных пользователем. Диалог организован по принципу диаграммы состояний, переход между которыми осуществляется с помощью команд текущего состояния. С помощью макро-средств можно описывать и хранить описание сценария диалога. В исходном диалоговом состоянии доступны состояния :

- «Калькулятор»,
- «Задание сцены»,
- «Задание экранной системы координат»,
- «Визуализация».

«Калькулятор» является компонентой диалогового монитора, которая существенно повышает уровень организации диалога и обеспечивает:

а) запоминание значений параметров диалога во внутренних переменных. Кроме обычных типов целого и вещественного числа, параметры могут принимать значения двумерных и трехмерных точек, а в множество операций включены операции векторной алгебры;

б) вычисление произвольных математических выражений над константами и внутренними переменными;

в) описание отдельных фрагментов диалога в виде параметрических макроопераций, состоящих из команд диалога, операций присваивания для внутренних переменных, условных команд и вызовов произвольных Си-функций. Внутри макрооперации также могут вызываться другие макрооперации. В

системе автоматически сохраняется «история» диалога, что обеспечивает возможность автоматизировать процесс подготовки новых макроопераций.

В состоянии «Задание сцены» осуществляется непосредственное ее описание. Элементы (примитивы) и конъюнкции, в которые элементы включаются, описываются с помощью параметров: имя, тип элемента (один из перечисленных выше), знак, цвет. Для работы с объектами имеются операции:

«Добавить объект» — объект добавляется к уже определенному объекту или определяется новый выходной объект;

«Вычесть объект» — указанный объект вычитается из другого объекта.

В перечисленных операциях действие производится над множествами конъюнкций объектов, участвующих в операции. В качестве объекта может фигурировать конъюнкция объектов. Геометрические преобразования выполняются с помощью операций:

«Перенос объекта» — объект перемещается на заданный вектор переноса;

«Поворот объекта» — объект поворачивается на заданный угол относительно указанной оси;

«Растяжение объекта» — растяжение (сжатие) объекта с заданным коэффициентом.

Имеются также операции удаления элемента и конъюнкции объектов. Для задания параметров элемента пользователь переходит из состояния «Задание сцены» в состояние «Задание значений параметров выбранного типа примитива».

Для наблюдения сцены из разных точек пространства предусмотрены операции в состоянии «Задание экранной системы координат».

В состоянии «Визуализация» устанавливаются режимы визуализации. Делается это с помощью функций:

— выбор режима цветообразования: «VGA цветной», «VGA серый» и два варианта режима для адаптера EGA;

— задание «коэффициента сжатия». Коэффициент определяет условный размер логического пиксела, для которого выполняется расчет интенсивности. Изменяя этот коэффициент, можно варьировать степень соотношения между скоростью расчета и качеством изображения, другими словами, обеспечивается возможность в целях отладки сцены быстро получать ее «грубое» изображение;

— выбор варианта визуализации: заданный объект или вся сцена;

— включение/выключение показа на экране меню;

— сохранение/восстановление изображения на ВЗУ.

**Заключение.** В целом можно резюмировать, что в настоящей версии описываемой системы, во-первых, удалось достигнуть совмещения универсальности и, благодаря встроенному механизму добавления новых примитивов, способности эффективно настраиваться на требуемое приложение. Первый практический опыт применения системы CG показал ее эффективность для решения задач компоновки объектов в ограниченном пространстве в задачах САПР и для архитектурных приложений. Во-вторых, были получены практические результаты по организации параллельных вычислений в алгоритме визуализации и оценки их эффективности на базе использования транспьютерной архитектуры. Дальнейшее развитие работ направлено на создание более совершенных средств графического редактирования пространственных сцен и разработку параллельных алгоритмов и программных средств, обеспечивающих режим динамики для реалистичных изображений.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бобков В. А., Кислюк О. С., Хамидулин А. В. Формирование растровых графических изображений пространственных объектов, построенных методом конструктивной геометрии // Программирование. 1989. № 3.
2. Роньшин Ю. И., Белов С. Б., Болотов В. П. Подсистема плоской геометрии для САПР судов // Мат-лы III Всесоюз. конф. по проблемам машинной графики и цифровой обработки изображений. Владивосток: ДВНЦ АН СССР, 1988.

3. Роньшин Ю. И. Монитор графического диалога // V Всесоюз. конф. «Машинная графика 89»: Тез. докл. Новосибирск: ВЦ СО АН СССР, 1989.
4. Bayer В. Е. An optimum method for two-level rendition of continuous — tone pictures // Int. Conf. Commun., Conf. Rec., 1973.
5. Ван Рет Ф., Ламоте В., Флеракерс Э. Методы ускорения трассировки лучей с использованием MIMD-архитектуры // Программирование. 1992. № 4.

*Поступила в редакцию 7 мая 1993 г.*

---

---

---

**Реклама продукции в нашем журнале — залог Вашего успеха!**