

СИСТЕМА ТВЕРДОТЕЛЬНОГО МОДЕЛИРОВАНИЯ СГ

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

1. Общие положения

Для реалистического восприятия геометрических форм твердотельное моделирование представляет самый современный и наиболее мощный метод.

Неоспоримым преимуществом твердотельного моделирования является следующее:

- обеспечение автоматического удаления скрытых линий;
- автоматическое построение сечений объектов;
- распознавание линий пересечения и линий очерков объектов;
- наличие разнообразной палитры цветов в получении реалистических изображений объектов.

На рис. 2-9 показаны примеры твердотельного моделирования в системе СГ.

Геометрический объект в системе СГ строится из геометрических примитивов с помощью операций объединения, пересечения, вычитания и геометрических преобразований (рис. 1).

Для примитивов определены операции:

- нахождение точек пересечения с заданным лучом;
- ортогональные геометрические преобразования;
- нахождение минимальных и максимальных значений координат примитива по осям трехмерного пространства.

Важным моментом в твердотельном моделировании является то, что ряд объектов должен быть определен как замкнутый объем с полной геометрией о его внешних гранях.

В системе реализованы следующие примитивы:

- полупространство, шар, параллелепипед, шестигранник, цилиндр, конус, обобщенный цилиндр; эллиптический цилиндр; цилиндр с образующей, задаваемой "аппаратом конструктора"; обобщенный тор; выпуклое тело вращения, тор, конический тор, выпуклый тор, спираль, коническая спираль, выпуклая спираль, участок поверхности.

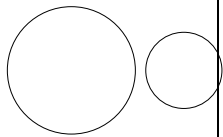
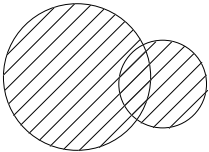
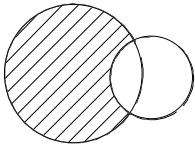
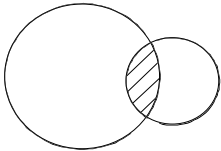

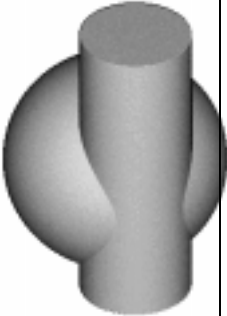


Теория множеств	Объединение (\cup)	Разность ($-$)	Пересечение (\cap)
 <p>A B</p>	 <p>$A \cup B$</p>	 <p>$A - B$</p>	 <p>$A \cap B$</p>
<p>Пример твердых тел</p>  <p>P Q</p>	 <p>$P \cup Q$</p>	 <p>$P - Q$</p>	 <p>$P \cap Q$</p>

Рис.1. Булевы операции "твердотельной геометрии"

Примитивы описываются в параметрической форме и задаются через их описывающие параметры. Обобщенный цилиндр задается вектором оси и образующей линией (контур поперечного сечения)

Для формирования образующей обобщенного цилиндра могут быть использованы линии из БДАК, или линии подготовленные системами "Вектор",

CorelDraw или AutoCAD, или линии снятые сканером с какого-либо рисунка.

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 тор, конический тор, выпуклый тор, спираль, коническая спираль, выпуклая спираль.

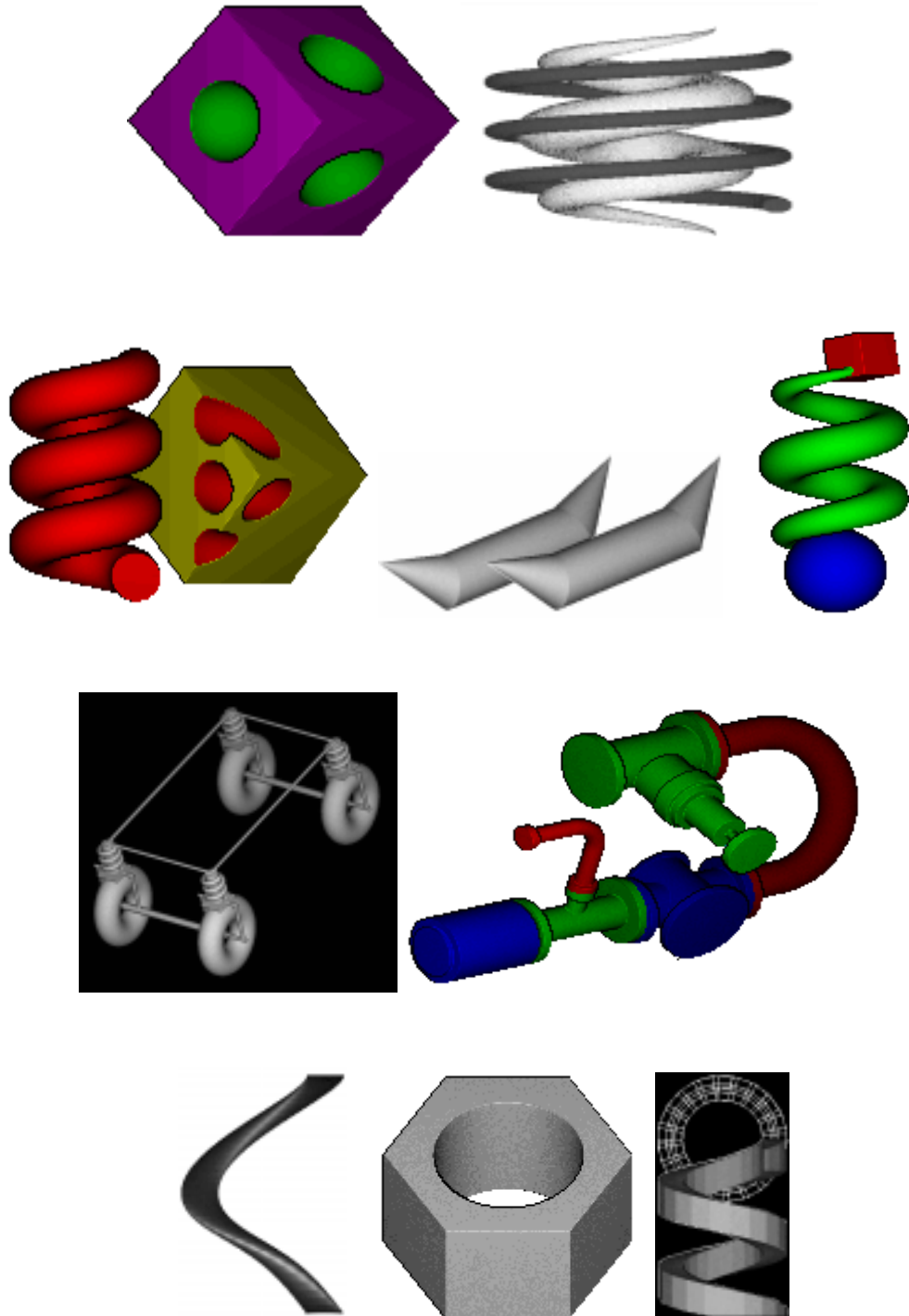


Рис.2. Примеры проектирования "твердотельных" фигур из стандартных примитивов системы CG

Участок поверхности генерируется в системе "Вектор" и передается в CG как объект с нулевой прозрачной/непрозрачной толщиной. С участком (или группой участков) поверхности можно выполнять все операции теорети-

ко-множественных операций. Возможность работы с трехмерными объектами произвольной формы, которые моделируются в системе "Вектор", решается передачей массива точек отдельно по граням и сечениям.

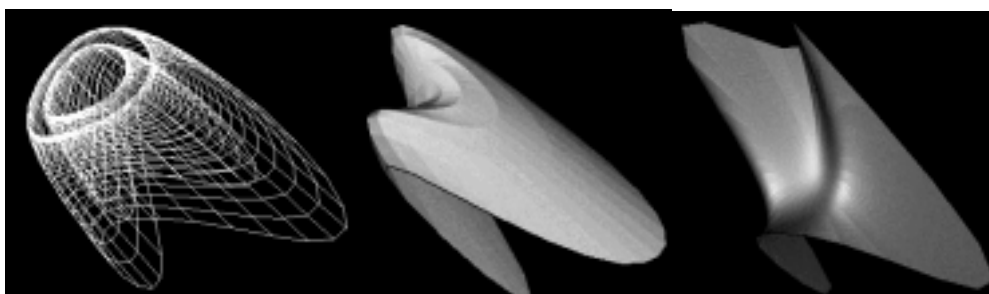
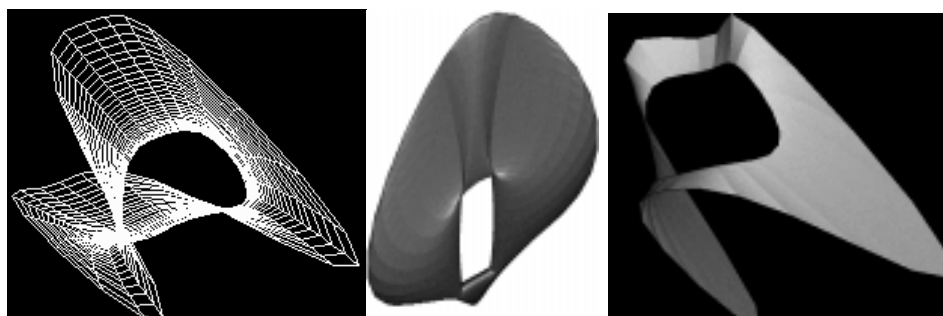
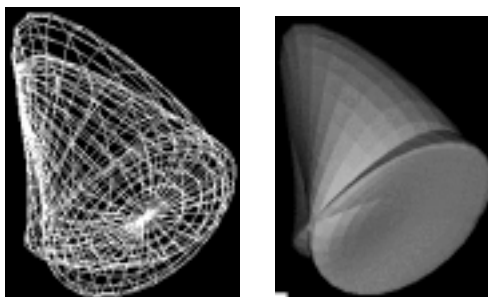
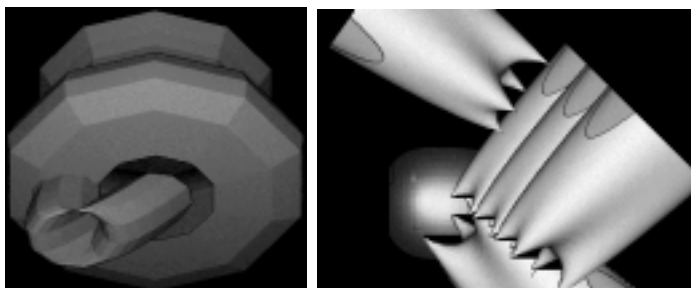


Рис. 3. Примеры проектирования "твердотельных" фигур в CG из участков поверхности системы "Вектор"

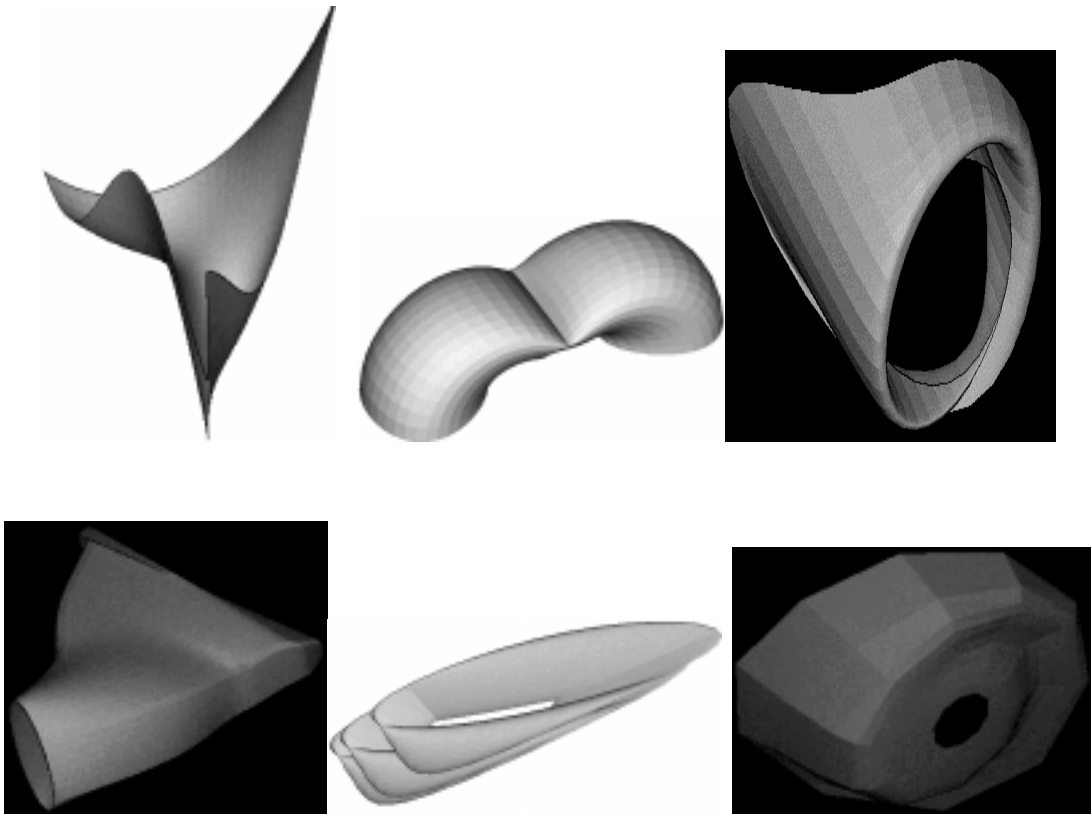


Рис. 4. Примеры проектирования "твердотельных" фигур в CG из участков поверхности системы "Вектор"

Сцена конструируется из примитивов указанных типов. Из примитивов строятся конъюнкции (пересечения), а конъюнкции объединяются в объекты. Как конъюнкции, так и объекты могут именоваться, и над поименованными объек-

тами могут выполняться операции (геометрические преобразования, сложение, вычитание объектов). В частности, поименованным объектом может служить вся сцена.

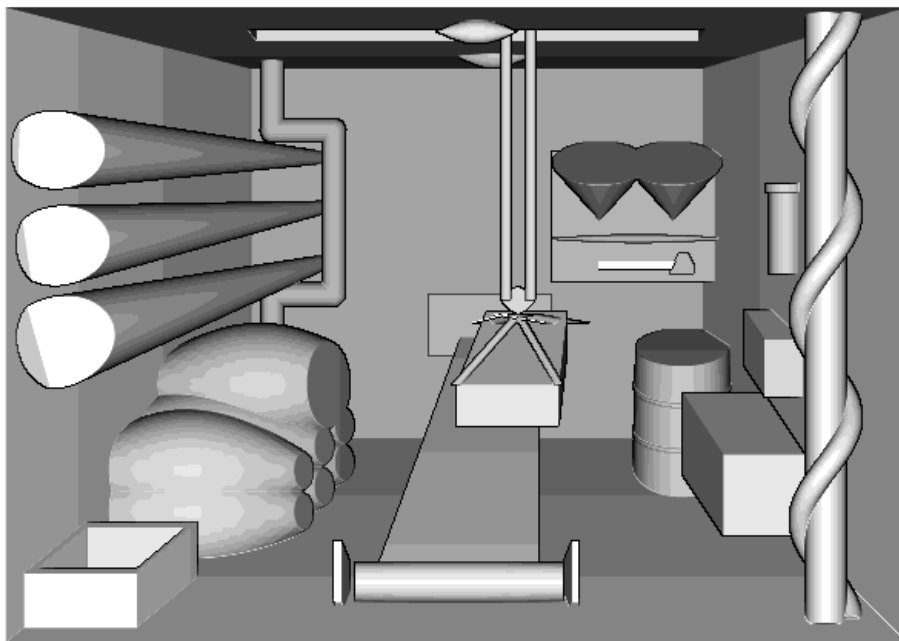


Рис. 5. Пример формирования сцены, полученной в системе CG

2. 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 расстояния z сводится к плоской, что, с одной стороны, упрощает вычисления, а с другой - позволяет реализовывать в системе более сложные геометрические примитивы, чем традиционные. Делается это просто - вначале рассчитывается плоское сечение zx для строки, а затем по сечению универсальным способом вычисляются значения z для всех пикселей строки. В случаях простых примитивов для сечения может быть получено аналитическое выражение, а в общем случае - это один или несколько плоских контуров, представленных ломаной. Для определения z -координаты в общем случае достаточно однобазовой процедуры, вычисляющей точки пересечения плоского контура ломаной с прямой.

Вторая особенность предлагаемых алгоритма и системы в целом - реализация сравнительно сложных, нетрадиционных геометрических примитивов (обобщенный цилиндр, обобщенный тор) и предусмотренная возможность расширения набора примитивов стандартным способом. Для этого в системе реализован эффективный механизм "подключения" новых примитивов.

Третья особенность системы связана с усилиями разработчиков, направленными на повышение быстродействия реализуемых алгоритмов. С этой целью в процессе программной реализации рассматривались и исследовались альтернативные алгоритмы.

Достижение реалистичности получаемых изображений также потребовало специальных усилий. В частности, качество изображения было повышено алгоритмом "размывания" границ интенсивности цвета. Для видеоадаптеров ПЭВМ типа EGA реализованы два алгоритма имитации интенсивности. Первый имитирует двенадцатиуровневую шкалу интенсивности, второй - непрерывную.

3. 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 макросредств можно описывать и хранить описание сценария диалога.

В исходном диалоговом состоянии доступны четыре состояния:

- работа с "калькулятором",
- задание сцены,
- задание экранной системы координат,
- визуализация.

"Калькулятор" является компонентой диалогового монитора, которая существенно повышает уровень организации диалога и обеспечивает пользователю:

- запоминание значений параметров диалога во внутренних переменных. Кроме обычных типов целого и вещественного числа, параметры могут принимать значения двумерных и трехмерных точек, а во множество операций включены операции векторной алгебры;
- вычисление произвольных математических выражений над константами и внутренними переменными;
- описание отдельных фрагментов диалога в виде параметрических макроопераций, состоящих из команд диалога, присваиваний для внутренних переменных, условных команд и вызовов произвольных "СИ"-функций. Внутри макроопераций 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, куда элементы включаются, описываются с помощью параметров: имя, тип элемента, знак (-, +), цвет.

Для работы с объектами имеются операции:

- "Добавить объект" - некоторый объект А добавляется к уже определенному объекту В либо определяется новый выходной объект;

"Вычесть объект" - производится вычитание указанного объекта А из объекта В.

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" - поворот объекта 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 задания значений параметров выбранного типа примитива. Имеется семь операций для выполнения этой функции.

Получить полную информацию о состоянии конструируемой сцены пользователь может с помощью трех операций в состоянии "Справка".

Для наблюдения сцены из разных точек пространства предусмотрены операции в состоянии "Режим задания ЭСК (экранной системы координат)".

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

- выбор режима цветообразования. Возможные режимы: "VGA цветной", "VGA серый" и два варианта режима для адаптера EGA;

- задание "коэффициента сжатия". Коэффициент определяет условный размер логического пикселя, 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 "быстро" визуализировать "грубое" изображение;

- выбор варианта визуализации: заданный объект или вся сцена;

- включение/выключение показа на экране меню;

- сохранение изображения в файле;

- загрузка изображения из файла.

При создании систем САПР судов возникает задача описания и реалистиче-

ской визуализации в интерактивном режиме пространственных геометрических объектов. В конкретных прикладных задачах такими геометрическими объектами являются как реальные объекты проектирования (деталь конструкции, корпус судна и т.п.), так и модельные объекты, с помощью которых осуществляется моделирование реальных объектов и процессов. В первом случае наряду с визуализацией необходимо также уметь получать точные чертежи проектируемых объектов.

Для второго типа задач достаточно иметь средства описания и визуализации геометрических объектов и возможности вычислений различных геометрических характеристик для обеспечения управления процессом конструирования. Примером этого типа задач является актуальная в машино-, авиа- и судостроении задача компоновки, т.е. оптимального размещения в замкнутом пространстве узлов и механизмов. Она может решаться путем замены реальных объектов упрощенными геометрическими, созданием и визуализацией соответствующей сцены модельных объектов на экране дисплея и последующей оценкой в интерактивном режиме с участием оператора. В результате последовательности таких итераций может быть получено оптимальное решение. Из такой постановки очевидным образом следует целесообразность разработки универсального (по отношению к специфике приложений) математического и программного инструмента для конструирования и визуализации сложных пространственных объектов. Реализация такого инструмента в виде автономной программной системы, которая может настраиваться на разные входные структуры данных, позволит использовать эту систему как самостоятельно, так и в составе других прикладных систем типа САПР и АСНИ для визуализации сложных пространственных сцен.

Следует при этом отметить, что безотносительно к выбранному математическому методу описания геометрических объектов и алгоритмам растровой визуализации особенностью любой реализованной системы будет потребление больших вычислительных ресурсов. Например, при реализации наиболее распространенного z-буферного алгоритма на ЭВМ типа ЕС-1066 или ПЭВМ с микропроцессором 80386/387 реализация практически реальных сцен требует от нескольких минут до десятков минут. Такие

времена неприемлемы для интерактивного режима работы. Реальным выходом из этой ситуации может быть разработка параллельных алгоритмов и реализация их на многопроцессорных вычислительных системах. Задача может быть решена в нижеследующей постановке.

Необходимо разработать математические и программные средства интерактивного конструирования и растровой визуализации сложных пространственных сцен, отвечающие следующим требованиям:

- описание сложных геометрических объектов и сцен, составленных из объектов, осуществляется методом конструктивной геометрии;

- набор поддерживаемых геометрических примитивов, из которых конструируются объекты, должен быть достаточно широким (для целей приложения) и легко расширяемым;

- обеспечивается возможность геометрических преобразований объектов сцены;

- вышеперечисленные возможности конструирования объектов и их преобразования должны поддерживаться языком;

- алгоритмы растровой визуализации основываются на использовании z-буферов; при разработке и реализации алгоритмов преследуется цель минимизации вычислительных ресурсов;

- для повышения реалистичности изображений реализуются соответствующие механизмы (освещенность сцены источником света, зависимость освещенности от глубины сцены и т.д.);

- программная реализация выполняется на языке "С" на ПЭВМ в виде автономной интерактивной системы с настраиваемым входным интерфейсом;

- для повышения быстродействия системы делается распараллеливание алгоритмов и адаптация программного обеспечения к многопроцессорной вычислительной системе.

4. Метод и алгоритмы конструирования и визуализации геометрических объектов

В основу разработанной системы в качестве принципиальной математической

основы положен метод конструктивной геометрии, который предполагает, что описание геометрического объекта строится из ограниченного набора типов примитивов с помощью теоретико-множественных операций и операций геометрического преобразования. При этом внутренним представлением объекта при описании метода конструктивной геометрии будет дерево (рис. 21) с терминальными вершинами, являющимися примитивами (шары, цилиндры и т.д.), и нетерминальными, являющимися операциями объединения, пересечения, вычитания и геометрических преобразований. При приведении дерева к дизъюнктивной нормальной форме (ДНФ), операциям ставятся в соответствие дизъюнкция и конъюнкция. Достоинствами такого подхода к описанию объектов является легкость определения семантики описаний (иначе говоря, легко построить предикат, который определяет, принадлежит точка объекту или нет) и тривиальность алгоритмов теоретико-множественных операций. Однако подход обладает и существенными недостатками. Одним из основных недостатков является сложность получения изображений пространственных объектов, построенных методом конструктивной геометрии.

В работе предлагается алгоритм и реализованная на его основе программная система для построения растровых цветных изображений пространственных объектов, построенных методом конструктивной геометрии.

Достоинства предлагаемого алгоритма следующие. В процессе построения изображения не происходит аппроксимации построенного объекта участками поверхности (например, многоугольниками). Изображение строится непосредственно на основе представления в виде дерева, терминальными вершинами которого являются примитивы, а нетерминальными - операции.

Класс примитивов заранее не ограничен. Примитивы определены как абстрактные типы данных, над которыми достаточно уметь осуществлять следующие операции:

1. ортогональные геометрические преобразования;

2. нахождение точек пересечения с заданным лучом;

3. нахождение минимальных и максимальных значений (необязательно точ-

ных) координат примитива по осям трехмерного пространства.

Примитив может быть и составным объектом, например, многогранником, описанным через границу. Необходимо

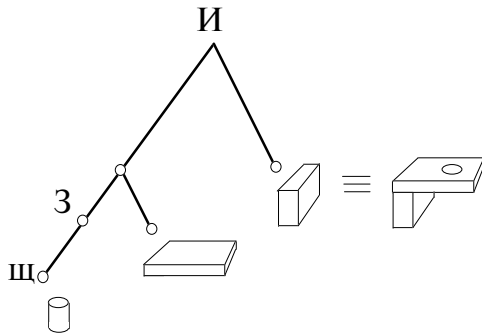


Рис. 6

Пусть P - любой примитив, над которым определены операции 1 - 3, либо объект, который является его отрицанием. Тогда описание пространственного объекта будет иметь вид:

$$\bigcup_{i=1..n} \bigcap_{j=1..m} P_{ij}$$

Теоретико-множественные операции над объектами, представленными в такой форме, задаются достаточно просто, хотя и несколько сложнее, чем в том случае, когда объект можно представить в виде произвольного булевого выражения.

Операция объединения задается тривиально. Операции пересечения и вычитания задаются по известным формулам булевой алгебры:

$$\begin{aligned} & \left(\bigcup_{i=1..n} \bigcap_{j=1..m_i} P_{ij} \right) \cap \left(\bigcup_{i'=1..n'} \bigcap_{j'=1..m'_i} Q_{i'j'} \right) = \\ & = \bigcup_{i=1..4} \bigcup_{i'=1..n'} \left(\bigcap_{j=1..m_i} P_{ij} \cap \bigcap_{j'=1..m'_i} Q_{i'j'} \right) \setminus \left(\bigcup_{i=1..n} \bigcap_{j=1..m_i} P_{ij} \right) = \\ & \left(\bigcup_{i=1..n} \bigcap_{j=1..m_i} P_{ij} \right) \cap \left(\bigcup_{i'=1..n'} \bigcap_{j'=1..m'_i} Q_{i'j'} \right) = \\ & = \bigcup_{i=1..4} \bigcup_{i'=1..n'} \left(\bigcap_{j=1..m_i} P_{ij} \cap \bigcap_{j'=1..m'_i} Q_{i'j'} \right) \setminus \left(\bigcup_{i=1..n} \bigcap_{j=1..m_i} P_{ij} \right) = \\ & = \bigcap_{i=1..n} \bigcup_{j=1..m_i} \text{и } \exists P_{ij} = \end{aligned}$$

только, чтобы над ним были определены операции 1 - 3.

Объект описывается в виде дерева специального вида, представленного в ДНФ.

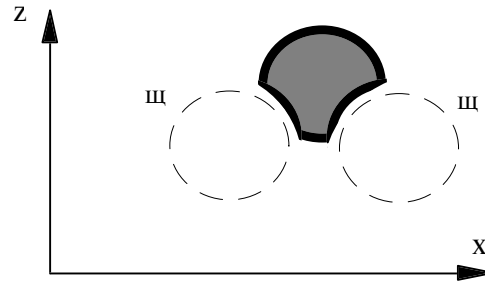


Рис. 7

$$= \bigcup_{j_2=1..m} \bigcup_{j_2=1..m} \dots \bigcup_{j_n=1..m_n} (\text{и } P_{1j_1} \cap \text{и } P_{2j_2} \cap \dots \cap \text{и } P_{nj_n})$$

При построении результирующих дизъюнкций происходит грубая проверка, содержит ли данная конъюнкция хотя бы одну точку. Для этого строится пересечение параллелепипедов, аппроксимирующих примитивы (см. операцию 3 над примитивами), не имеющие знака "-".

Если пересечение пусто, конъюнкция исключается, если нет - проверяется, нет ли в конъюнкции примитивов со знаком таких, что параллелепипед, аппроксимирующий примитив, не пересекается с построенным пересечением. Если такие примитивы есть, они исключаются из конъюнкции.

Операции геометрического преобразования объекта сводятся к преобразованию примитивов с использованием операции 1. Рассмотрим, как работает алгоритм построения растровых изображений для пространственных объектов, представленных в виде ДНФ. На первом этапе проводится геометрическое преобразование объекта, переводящее его в экранную систему координат. В экранной системе координат плоскость $Z=0$ есть плоскость экрана, проецирование осуществляется вдоль оси z . Ориентация строки пикселей экрана совпадает с ориентацией оси x . Затем осуществляется сортировка примитивов, из которых состоит пространственный объект. Каждой строке пикселей экрана k ставится в соответствие список S -примитивов таких, что минимальное значение по оси y (найденное

с помощью процедуры 3) примитива попадает в данную строку.

На следующем этапе работы алгоритма происходит последовательный анализ строк с экрана. Для каждой строки k строится сечение z -буфера $Z=Z(x,y)$, где Z - расстояние до ближайшего участка поверхности выводимого объекта вдоль оси наблюдения, y - y -координата k -й строки пикселей, x - координата пикселей в строке. По двум соседним сечениям z -буфера (для строк k и $k-1$) строится строка изображения, которая выводится на растровое графическое устройство. Соседнее сечение z -буфера при этом используется для вычисления нормали к поверхности. Освещенность в данной точке вычисляется по формуле:

$$I = I_0 \cos(\theta) f(r),$$

где I - интенсивность источника света; θ - угол между нормалью к поверхности и направлением на источник света; r - расстояние от точки поверхности до источника света; f - функция убывания освещенности с глубиной (обычно $a/a + r$). При последовательном анализе строк алгоритм использует текущий список S -примитивов, проекции которых пересекаются с данной строкой. Первоначально этот список пуст. При переходе к анализу новой строки в него добавляется список примитивов S (см. выше), проекции которых пересекаются с данной строкой и не пересекаются с предыдущей, и из него изымаются примитивы, для которых максимальное значение по оси u меньше значения u для данной строки k .

Таким образом, резко сокращаем количество примитивов, которое надо рассмотреть при построении сечения z -буфера, а если процедура 3 дает точные значения минимального и максимального значения по оси u , то при построении сечения z -буфера будем рассматривать только те примитивы, проекции которых заведомо пересекаются с данной строкой пикселей.

Рассмотрим теперь, как на основе текущего списка примитивов осуществляется построение сечения z -буфера для строки пикселей k . Первоначально $z = +\infty$ для любого x . Затем для конъюнкции j исходного пространственного объекта, все примитивы (без знака) которого находятся в текущем списке примитивов S , строятся сечения z -буфера $Z(x,y)$. Сечение z -буфера для всего объекта последо-

вательно модифицируется этими z -буферами по формуле:

$$Z(x,y) := \min(Z(x,y_n), z_j(x,y_n)).$$

Сечение z -буфера $Z(x,y)$ для конъюнкции j строится следующим образом.

Из текущего списка примитивов S выбирается очередной примитив P . Если все примитивы, входящие в конъюнкцию j и содержащие P без знака, входят одновременно и в текущий список примитивов S , и если сечение $Z(x,y)$ для конъюнкции j еще не строилось, то строится отрезок $[XMIN, XMAX]$, являющийся пересечением всех отрезков $[XMIN, XMAX]$, где $XMIN, XMAX$ - минимальное и максимальное значение x для примитивов без знака, входящих в конъюнкцию j . Если отрезок $[XMIN, XMAX]$ не пуст, строится таблица, в которой каждому пикселю x,y z -буфера $Z(k,y)$ (где x лежит в интервале от $XMIN$ до $XMAX$) ставится в соответствие список S . В список S входят координаты точек пересечения луча, проведенного через точку x,y параллельно оси наблюдения z с поверхности всех примитивов, входящих в конъюнкцию j . Точки пересечения луча с примитивом можно найти в силу существования процедуры 2 (см. выше). Точки пересечения, входящие в список S , упорядочиваются по возрастанию координаты z . Затем для каждого пикселя x,y ищется расстояние $Z(x,y)$ вдоль оси наблюдения до ближайшей точки поверхности конъюнкции j . Для этого используется список S точек пересечения луча наблюдения, проходящего через пиксель x,y с примитивами из конъюнкции j .

Рассмотрим процесс нахождения $Z(x,y)$. Будем двигаться по лучу из точки $Z=0, x=x, y=y$ вдоль оси z . Обозначим количество примитивов без знака из конъюнкции j , внутри которых будем находиться, через N , а количество примитивов со знаком из конъюнкции j , внутри которых будем находиться, как N . Тогда, если при движении вдоль оси z пересекаем: первую граничную точку примитива со знаком из конъюнкции j , то $N:=N-1$, первую граничную точку примитива без знака из конъюнкции j , то $N:=N+1$; вторую граничную точку примитива без знака из конъюнкции j , то $N:=N-1$.

Если после пересечения очередной граничной точки $N=0$ и $N=N$ (общему количеству примитивов без знака из конъюнкции j), то z координата данной

граничной точки дает нам искомое значение $Z(x,y)$. Действительно, $N=0$ и $N=N$ означает, что мы находимся внутри всех примитивов конъюнкции j без знака и вне всех примитивов конъюнкции j со знаком "+".

В качестве геометрических примитивов на текущий момент в системе выбраны и программно поддерживаются следующие примитивы: полупространство, шар, параллелепипед, цилиндр, конус, обобщенный цилиндр (образующая - произвольный контур), обобщенный тор (осевая - произвольная кривая в пространстве, образующая - окружность переменного радиуса $r=f(t)$, где t - параметр, зависящий от длины осевой).

5. Распараллеливание алгоритмов

Для анализа возможных методов распараллеливания общего алгоритма работы системы представим его в виде укрупненного блок-схем, изображенных на рис. 6 и рис. 7. На рис. 6 выделены три этапа преобразований графических данных. Вначале задается описание сцены в мировой системе координат в диалоговом режиме. Затем делается преобразование данных в экранную систему координат. Эти действия выполняются графическим монитором при участии оператора-пользователя.

Третий основной этап работы системы - строчное формирование растрового изображения, что обеспечивается работой всех алгоритмов визуализации. После формирования порции строк или всего изображения осуществляется непосредственное отображение сформированного изображения на мониторе.

Для упрощения работы по адаптации ПО на многопроцессорной системе вначале была реализована простая схема распределения ПО системы между ПЭВМ и транспьютерным набором. Согласно этой схеме первые два этапа блок-схемы рис. 8, а также последний - визуализация, выполнялись на ПЭВМ. На транспьютерном наборе выполнялся этап формирования изображения, который был организован в виде одной задачи. Работа всех алгоритмов этого этапа построена на операциях над числами с плавающей запятой. В результате тестирования было выявлено, что при такой схеме распределения ПО на сравнительно сложных геометрических сценах достига-

ется ускорение в сравнении с функционированием на ПЭВМ в 10 - 12 раз.

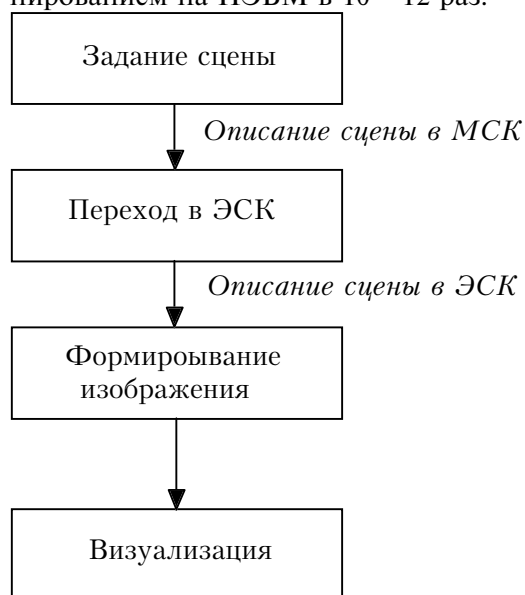


Рис.8. Алгоритм обработки сцены

После проведения описанного эксперимента был проведен анализ с целью реализовать оптимальную схему распараллеливания применительно к этапу формирования изображения, на котором работают все трудоемкие алгоритмы. Возможны два уровня распараллеливания работы системы. Первый уровень соответствует различным способам разделения экранной матрицы изображения на составляющие области, для каждой из которых организуется свой процесс полного цикла обработки данных до формирования изображения. В этом случае каждый процесс заполняет свою область в результирующей матрице изображения, которая по завершению всех процессов передается на визуализацию. Плюсом такого способа распараллеливания является то, что не требуется модификация последовательно работающего алгоритма системы. Издержки здесь возникают за счет того, что при отдельной обработке отдельных частей изображения-экрана появляются граничные эффекты, для устранения которых необходимо предусмотреть грамотную "сшивку" смежных частей. Более существенный минус заключается в том, что при таком способе априори нельзя получить оптимального использования имеющихся процессов без дополнительного механизма адаптивного разделения. Причина состоит в том, что одинаковые по площади части экрана могут содержать различные по насыщен-

ности изображения и, соответственно, разные объемы вычислений.

Второй уровень распараллеливания соответствует реализации параллельных алгоритмов, т.е. алгоритмов с параллельной обработкой промежуточных данных. В этом случае глубина параллелизма определяется свойствами самого алгоритма.

Из всех возможных разбиений - на квадраты (размером в пиксель и больше), на горизонтальные полосы (от одной строки до нескольких), вертикальные полосы и возможные их комбинации - выбран способ разбиения матрицы-экрана на вертикальные полосы. Применительно к блок-схеме на рис. 24 распараллеливание в этом случае делается в точке А. Достоинствами этого способа в сравнении с другими перечисленными являются следующие:

- каждая вертикальная полоса рассматривается как самостоятельный экран, применительно к которому работает весь последовательный алгоритм (от точки А), а модификация ПО в этом случае минимальна;

- трудоемкий этап "оптимизация для экрана" выполняется меньшее число раз в сравнении с другими способами разбиения;

- минимальные издержки на устранение граничных эффектов в сравнении с другими способами.

В результате реализации предложенного способа распараллеливания работы системы на сложных геометрических сценах получено ускорение в 8-9 раз.

6. Технические характеристики

Система CG требует при работе 200 кб оперативной памяти без учета динамической памяти под объекты визуализируемой сцены. Время построения и визуализации сцены зависит от количества объектов в сцене, занятой ими на экране площади, от типов используемых прими-

тивов и от режима визуализации. Наиболее трудоемким в вычислительном отношении является примитив "обобщенный тор", остальные примитивы требуют существенно меньших вычислительных затрат на обработку.

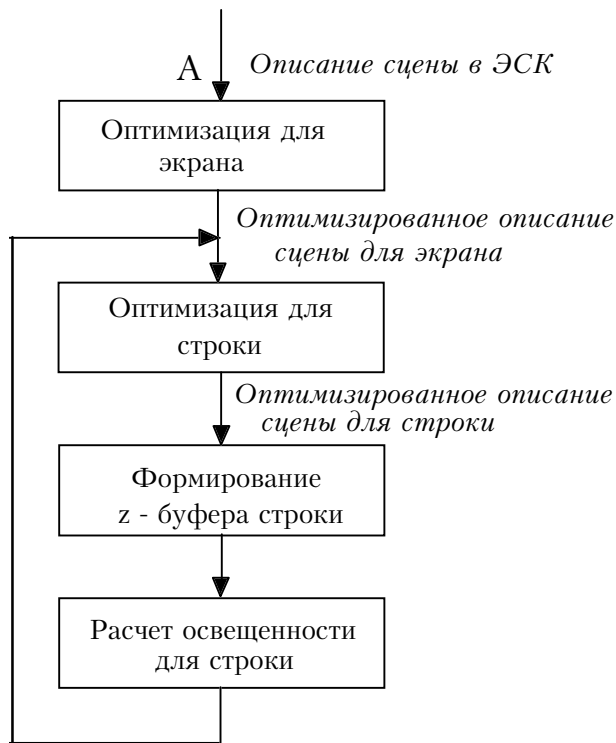


Рис. 9. Формирование растрового изображения

Для всех изображений, приведенных в качестве примеров, на рисунках указано время визуализации для ПЭВМ типа IBM PC AT 386/387 (16 МГц). Режим VGA для указанной модели ПЭВМ реализован со следующими характеристиками:

- в цветном варианте: палитра 256 цветов, разрешение 320*200, 640*480, 800*600;

- в черно-белом варианте: яркость 16 уровней, разрешение 640*480, 800*600, 1024*800. Язык реализации системы - "СИ".