

УДК 681.3.06 : 629.7.058.74

А. В. ГУСЕВ, С. Л. ИВАШИН, Э. А. ТАЛНЫКИН
(Новосибирск)

МАТЕМАТИЧЕСКИЕ МОДЕЛИ СЦЕН В СИНТЕЗИРУЮЩИХ СИСТЕМАХ ВИЗУАЛИЗАЦИИ РЕАЛЬНОГО ВРЕМЕНИ

Системы синтеза изображений [1—3] предполагают наличие математической (численной) модели реального или гипотетического сюжета. В процессе синтеза информация перерабатывается в соответствии с выбранными алгоритмами генерации, и результатом является растровое изображение в памяти видеокадра [1] (в некоторых системах изображение генерируется непосредственно на видеоконтрольном устройстве в темпе телевизионной развертки [2, 3]). В настоящей статье описан подход к организации баз данных визуальных сцен, применяемый в синтезирующих системах визуализации (ССВ), разрабатываемых в Институте автоматизации и электрометрии СО АН СССР [2]. Рассматриваемые системы предназначены для генерации нескольких связанных изображений (каналов наблюдения) в сценах с подвижными объектами. Одной из главных особенностей, влияющих на организацию баз данных, является фактор реального времени, т. е. способность системы генерировать изображения с частотой 25—50 Гц для всех активных каналов наблюдения.

В процессе функционирования ССВ в реальном времени база данных содержит информацию, достаточную для построения любых видов сцены, для всех положений наблюдателей, возможных в процессе выполнения задачи. Такую базу данных называем базой данных реального времени, отражая тот факт, что она используется непосредственно при генерации изображений и предназначена исключительно для этой цели. База данных реального времени есть результат адаптации описания модели сцены к конкретным особенностям архитектуры и программного обеспечения ССВ, к используемым алгоритмам генерации изображений, к специфике задачи и некоторым частным ее условиям. Например, если отрабатывается сюжет в условиях дневного освещения в течение небольшого промежутка времени, то можно с достаточной степенью достоверности считать источник освещения неподвижным и иметь в базе данных уже рассчитанными параметры освещенности для всех неподвижных объектов, а также тени от неподвижных объектов на неподвижные. Изменить условия освещенности в такой базе данных уже нельзя. Таким образом, база данных реального времени построена с учетом минимизации вычислительных затрат на поиск и обработку информации при генерации изображений в ущерб инвариантности и универсальности описания моделей, составляющих сцену.

При создании базы данных визуальные модели объектов являются предметом манипуляции для программ конструирования, сборки, коррекции и многих других видов обработки вне реального времени. Здесь требуется максимальная гибкость организации и представления, что противоречит жестким требованиям реального времени. С целью разрешения

этих противоречий вводим два совершенно различных представления для описания сцен. Одно из них используется на этапе подготовки данных, а другое — непосредственно в процессе генерации изображений (упоминаемая выше база данных реального времени).

Программный комплекс, обеспечивающий работу ССВ в реальном времени, представляет собой операционную систему специального назначения, которая, наряду с другими возможностями, поддерживает штатную организацию файлов. Базы данных реального времени размещаются на дисках в виде стандартных файлов, записываемых в процессе подготовки данных средствами штатной операционной системы.

Файл базы данных открывается заголовком, где хранятся информация о расположении в файле всех элементов базы данных, а также о характере освещенности, прозрачности атмосферы и другие данные, имеющие отношение к сцене в целом. Далее будут описаны основные элементы базы данных.

Структура сцены представляет информацию о взаимной подчиненности и расположении объектов сцены, их детализации и габаритах. Во время генерации изображения структура располагается в памяти в виде набора связанных деревьев, а в файле базы данных деревья представляются линейной синтаксической структурой, аналогично показанной в [4]. Среди деревьев выделено корневое дерево, описывающее непосредственно саму сцену. Остальные деревья представляют объекты сцены, которые могут быть вызваны в нескольких местах в структуре других объектов или сцены. Кроме информации, относящейся непосредственно к конфигурации и геометрии, деревья включают дополнительные конструкции, обеспечивающие эффективность алгоритмов отображения. Возможные типы узлов дерева показаны на рис. 1.

Узел типа *последовательность* (рис. 1, а) объединяет в указанном порядке несколько поддеревьев в одно дерево. Порядок поддеревьев определяет их приоритеты при отображении [2, 5], которые в данном случае фиксированы, т. е. объекты, стоящие выше по списку, всегда будут закрывать объекты, следующие за ними, независимо от положения наблюдателя. В таком отношении находятся, например, подвижные объекты на фоне Земли, наземные постройки и земная поверхность, если наблюдатель не будет опускаться «слишком низко». Если два объекта могут взаимно закрывать друг друга при различных положениях наблюдателя, их нельзя размещать в последовательность без нарушения правильности изображений тех видов сцены, где закрывающий объект расположен в списке за закрываемым.

Узел типа *разделяющая плоскость* (рис. 1, б) включает в качестве информации два поддерева и коэффициенты уравнения плоскости вида $Ax + By + Cz + D = 0$, разделяющей в пространстве геометрические образы этих поддеревьев. В процессе генерации на отображение обязательно оба поддерева, но сначала то, в полупространстве которого находится наблюдатель. Определение положения наблюдателя сводится к нахождению знака выражения $Ax_n + By_n + Cz_n + D$, где (x_n, y_n, z_n) —

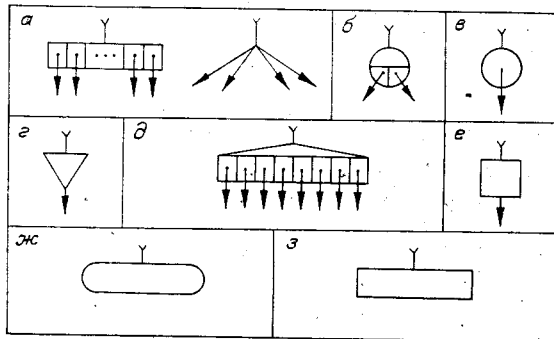


Рис. 1. Узлы дерева сцены:
а — последовательность; б — разделяющая плоскость; в — описанная сфера; г — габаритный шар; д — альтернатива детализации; е — погружение; жс — вызов; з — сегмент.

положение наблюдателя. Разделяющие плоскости в базе данных и описанный только что механизм перестановки двух поддеревьев узла *разделяющей плоскости* — это все, что имеет отношение к удалению невидимых поверхностей на этапе генерации изображений. Собственно удаление невидимых частей изображения проводится аппаратно на основе приоритетной схемы [2].

Узел типа *описанная сфера* (рис. 1, в) содержит координаты центра и диаметр некоторой сферы, а также включает поддерево, геометрический образ которого целиком заключен внутри сферы. Наличие в структуре объекта описанной сферы не отражается на получаемом изображении, а помогает сократить необходимые для генерации вычислительные ресурсы. Затраты на тест сферы относительно пирамиды видимости сравнимы с затратами на генерацию изображения одного четырехугольника. Если в результате тестирования обнаруживается, что сфера не попадает в пирамиду видимости, то поддерево вообще не рассматривается в процессе дальнейшей генерации изображения на данный канал наблюдения. Если сфера не пересекается какой-либо гранью пирамиды видимости, то в процессе генерации изображения поддерева можно не производить клиппирования (отсечения) относительно данной грани. При вырождении изображения сферы до размера менее элемента раstra можно также не генерировать образ поддерева. Последний прием позволяет не только сократить необходимые вычислительные ресурсы, но и разгрузить аппаратуру от генерации изображений вырождающихся деталей, фактически не влияющих на характер изображения.

Ограниченная производительность аппаратуры требует поддерживать сбалансированную детализацию объектов сцены не только на уровне вырождения мелких деталей. При удалении объекта от наблюдателя часто бывает целесообразно сменить детальное описание на более грубое, а при приближении — наоборот. Изменение детальности может быть произведено полной заменой модели. Например, если моделью является тело вращения, то при смене детализации можно менять число разбиений, применяемых при аппроксимации поверхности многоугольниками. Другой способ состоит в повышении детальности за счет включения в изображение дополнительных элементов объекта, таких, например, как оконные переплеты и структура кирпичной кладки на моделях строений. В общем случае может применяться смешанная стратегия. Подбор уровней детализации объектов, сбалансированный относительно имеющихся вычислительных ресурсов и производительности аппаратуры, — наиболее сложная задача в подготовке базы данных для конкретной сцены.

Механизм детализации сцены реализуется с помощью двух типов узлов (рис. 1, г, д) — *габаритный шар* и *альтернатива детализации*. В объект, который необходимо детализировать, вводится «пробный» шар, по размеру которого на изображении будет приниматься решение о степени детализации. Узел типа *габаритный шар* (см. рис. 1, г) содержит координаты центра и диаметр шара, а также поддерево, на которое распространяется детализация. Узел типа *альтернатива детализации* (см. рис. 1, д) включает восемь различных поддеревьев, соответствующих ситуациям, когда диаметр пробного шара на изображении составляет 1, 2, 4, 8, 16, 32, 64 и 128 элементов раstra. Детализация на рис. 1, д убывает слева направо.

Узел типа *погружение* (рис. 1, е) содержит матрицу движения в трехмерном пространстве и поддерево, на которое действует матрица. Если с течением времени изменять матрицу, то объект, описываемый поддеревом, становится подвижным. Погружение может использоваться также для размещения в разных местах сцены экземпляров объекта, описанного однажды в структуре данных.

Узел типа *вызов* (рис. 1, ж) означает включение в данном контексте в структуру описываемого объекта другого дерева, упоминаемого в узле вызова именем или отсылочным номером. *Вызов* — терминальный узел (лист) включающего дерева.

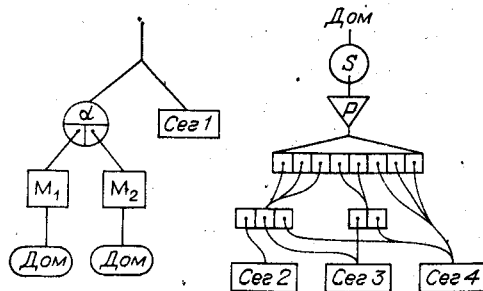


Рис. 2. Структура сцены, показанной на рис. 3.

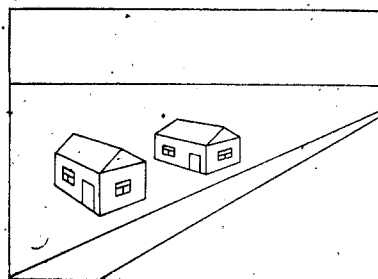


Рис. 3. Пример сцены.

Еще одним типом терминального узла является *сегмент* (рис. 1, з). *Сегмент* — это единственный элемент базы данных с информацией о реально видимых на экране объектах. Узел типа *сегмент* содержит спецификацию сегмента, позволяющую найти его расположение в файле базы данных.

В заключение обратим внимание на рис. 2, где показана структура сцены, изображенной на рис. 3. В схеме рис. 2 приняты следующие обозначения: *сег 1* — сегмент, описывающий «подстилающую» поверхность, т. е. землю, дороги и т. п.; α — плоскость, разделяющая два строения; M_1 и M_2 — матрицы погружения для размещения двух строений; *сег 2* — оконные переплеты; *сег 3* — окна; *сег 4* — стены и крыша. Строение заключено в сферу S , и для выбора альтернативы детализации введен габаритный шар P .

Представление структуры сцены в файле базы данных имеет вид линейной последовательности записей, образующих вполне определенные синтаксические конструкции. Каждое дерево структуры сцены представляется отдельной последовательностью. Ниже описан синтаксис структуры дерева, где курсивом выделены терминальные элементы, т. е. конкретные записи. Для сопоставления с элементами дерева смотри рис. 1:

$\langle \text{структура дерева} \rangle ::= \langle \text{последовательность} \rangle \text{конец.}$
 $\langle \text{последовательность} \rangle ::= \text{ПУСТО} |$
 $\langle \text{последовательность} \rangle \langle \text{блок} \rangle.$
 $\langle \text{блок} \rangle ::= \text{сегмент} |$
 $\text{вызов} |$
 $\text{плоскость} \langle \text{последовательность} \rangle$
 $\text{раздел} \langle \text{последовательность} \rangle \text{конец} |$
 $\text{сфера} \langle \text{последовательность} \rangle \text{конец} |$
 $\text{шар} \langle \text{последовательность} \rangle \text{конец} |$
 $\text{погружение} \langle \text{последовательность} \rangle \text{конец} |$
 $\text{альтернатива} \langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{раздел}$
 $\langle \text{последовательность} \rangle \text{конец.}$

Записи типа *конец*, *раздел* и *альтернатива* играют роль синтаксических разделителей. Запись типа *плоскость* содержит коэффициенты уравнения плоскости, *сфера* — координаты центра и диаметр описанной сферы, *шар* — координаты центра и диаметр габаритного шара, *погружение* — матрицу погружения размера 4×3 , *сегмент* — информацию о расположении сегмента в файле базы данных, *вызов* — идентификацию одного из деревьев, представляющих структуру сцены. Для примера сцена, изображенная на рис. 2, 3, имеет следующее представление:

корень: плоскость A, B, C, D
 дом 1: погружение M_1
 вызов дом
 конец
 раздел
 дом 2: погружение M_2
 вызов дом
 конец
 конец
 дом: сфера X, Y, Z, D
 шар X, Y, Z, D
 альтернатива
 сегмент сег 2; сегмент сег 3; сегмент сег 4
 раздел
 сегмент сег 2; сегмент сег 3; сегмент сег 4
 раздел
 сегмент сег 2; сегмент сег 3; сегмент сег 4
 раздел
 сегмент сег 3; сегмент сег 4
 раздел
 сегмент сег 3; сегмент сег 4
 раздел
 сегмент сег 4
 раздел
 сегмент сег 4
 раздел
 сегмент сег 4
 конец
 конец
 конец

Таблица цветов содержит 256 записей типа (R, G, B) , описывающих соответственно красную, зеленую и синюю компоненты цвета ($0 \leq R, G, B \leq 1$). Для указания цвета некоторого объекта в базе данных используется пара (N, I) , где N — номер записи в таблице цветов ($0 \leq N \leq 255$), I — интенсивность выбранного цвета ($0 \leq I \leq 1$). При подготовке базы данных, наряду с табличными, могут использоваться непосредственные способы задания, а также нотации, отличные от R, G, B .

Сегмент представляет собой описание конкретного объекта, его части или нескольких объектов сцены. Одно из основных требований к сегментам состоит в возможности их передачи по каналам данных за одно обращение к каналу для того, чтобы свести к минимуму время, необходимое для передачи. В системе имеются два вида каналов, через которые производится передача сегментов: дисковый накопитель — память управляющей ЭВМ и память ЭВМ — память геометрического процессора [2]. В первом случае требуется, чтобы сегмент размещался в виде непрерывного участка дисковой памяти, а во втором — внешняя однородность передаваемых данных, так как канал имеет преобразователь форматов, настраиваемый на определенный режим передачи. В системе принят режим передачи нормализованными вещественными числами, поскольку данные такого типа преобладают в структуре сегмента. Информация других типов (целые числа, счетчики, признаки и т. п.) кодируется в отдельных полях представления вещественного числа.

Сегмент содержит упорядоченный набор плоских многоугольников (граней) в трехмерном пространстве. В процессе генерации изображения грани обрабатываются в том порядке, в каком они представлены в структуре сегмента, т. е. грани, стоящие впереди, не могут закрыться на изображении гранями, стоящими за ними по списку. При этом грани заднего плана (невидимые в силу ориентации относительно наблюдателя) отбраковываются на одном из первых этапов обработки. Правильно

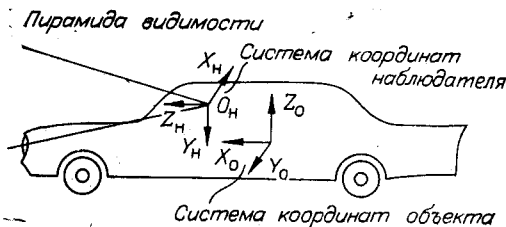
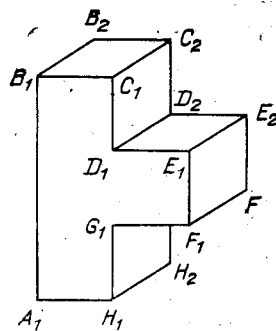


Рис. 5. Системы координат.

Рис. 4. Геометрический образ сегмента.

упорядоченный сегмент будет иметь достоверное изображение со всех точек наблюдения. Алгоритм сортировки граней и получения такого порядка приведен в [5]. На рис. 4 изображен непрозрачный объект. Легко убедиться, что следующий порядок его граней будет правильным:

$$\begin{array}{lll}
 H_1 G_1 F_1 E_1 D_1 C_1 B_1 A_1, & A_2 B_2 C_2 D_2 E_2 F_2 G_2 H_2, & B_1 B_2 A_2 A_1, \\
 B_1 C_1 C_2 B_2, & A_1 A_2 H_2 H_1, & E_1 F_1 F_2 E_2, \\
 G_1 G_2 F_2 F_1, & C_1 D_1 D_2 C_2, & G_1 H_1 H_2 G_2.
 \end{array}$$

Описание грани сегмента содержит цвет (номер в таблице цветов), нормаль, признак детали и список вершин. Признак детали разрешает отбраковывать грань в случае аппаратных перегрузок. Если в процессе построения изображения оказывается невозможным отобразить в одном кадре все поступившие грани, то отбрасываются грани, имеющие малую площадь и снабженные признаком детали. Описание вершины в общем случае содержит координаты, индивидуальную интенсивность и вектор псевдонормали, который используется в алгоритме расчета освещенности данной точки. Если грань представляет элемент плоской поверхности объекта, то нет необходимости хранить вектор нормали в каждой вершине, так как он совпадает с нормалью грани. Индивидуальные нормали необходимы в случае полигональной аппроксимации криволинейных поверхностей [6].

При описании поверхностей твердого тела, а особенно при аппроксимации криволинейных поверхностей часто бывает, что одна вершина участвует в нескольких гранях. Например, на рис. 4 каждая вершина встречается в трех гранях. В каждом из сегментов может присутствовать таблица вершин и таблица нормалей, тогда при перечислении вершин и их индивидуальных нормалей достаточно разместить только индексы в таблицах. Такой прием позволяет сократить не только объем памяти для сегмента, но также и затраты на геометрические преобразования. Перед обработкой сегмента таблицы вершин и нормалей преобразуются один раз, после чего их преобразованные значения участвуют в построении изображения без всякой обработки. Каждая из таблиц сегмента может быть пустой. Способ задания граней (непосредственно или через таблицу) выбирается отдельно для каждой грани при подготовке базы данных.

Таблица наблюдателей содержит описание возможных каналов наблюдения или наблюдателей, каждый из которых приписан к определенному объекту сцены. В системе имеются средства движения объектов (например, метки дом 1 и дом 2 в представленной выше сцене определяют два потенциально подвижных объекта), и наблюдатель движется вместе с объектом, к которому он приписан. Наблюдатель определяется пирамидой видимости и ее положением в системе координат объекта (X_o, Y_o, Z_o) (рис. 5). Описание наблюдателя в таблице содержит положение точки наблюдения (вершина пирамиды O_n) в системе координат объекта, матрицу преобразования из системы координат объекта в систему наблюдателя (X_n, Y_n, Z_n) и уравнения четырех плоскостей пирамиды (левой, правой, верхней и нижней) в системе координат объекта.

Заключение. Предложенный подход применяется при разработке программного обеспечения конкретных систем синтеза визуальной обстановки [2], реальная эксплуатация которых показала работоспособность изложенной техники организации баз данных в системах реального времени.

ЛИТЕРАТУРА

1. Demos G., Brown M. D., Weinberg R. A. Digital scene simulation: The synergy of computer technology and human creativity.— Proc. IEEE, 1984, N 1.
2. Ковалев А. М., Талныкин Э. А. Машинный синтез визуальной обстановки.— Автометрия, 1984, № 4.
3. Башков Е. А., Казак А. В. Генераторы изображения для авиатренажеров.— За рубежом радиоэлектроника, 1984, № 8.
4. Талныкин Э. А. Внутренний язык для описания визуальных моделей.— Автометрия, 1985, № 4.
5. Зингер Б. Х., Талныкин Э. А. Предварительная пространственная сортировка — основа алгоритма удаления невидимых поверхностей для систем отображения приоритетного типа.— Автометрия, 1983, № 6.
6. Gouraud N. Computer display of curved surfaces.— IEEE Trans., 1974, v. C-20, June.

Поступила в редакцию
13 февраля 1985 г.

УДК 519.219 : 519.237.5

А. Г. БУЙМОВ
(Томск)

АНАЛИЗ ВЛИЯНИЯ КОРРЕЛЯЦИОННЫХ СВОЙСТВ НЕОДНОРОДНОГО ЯРКОСТНОГО ШУМА НА КОВАРИАЦИОННУЮ МАТРИЦУ ОШИБОК СОВМЕЩЕНИЯ ИЗОБРАЖЕНИЙ

Введение. Предельная точность корреляционно-экстремальных систем, работающих по принципу совмещения изображений [1] и использующих фильтры Калмана [2], метод минимума квадратичного рассогласования [3] или максимума функции взаимной корреляции [4], соответствует точности метода наименьших квадратов. Это обусловлено известным из теории корреляционно-экстремальных систем фактом: установившиеся ошибки совмещения не превышают соответствующих дифференциальных радиусов корреляции изображений и допускают линеаризацию нелинейных зависимостей [1, 2]. При малом сдвиге λ прямоугольного $n = T_1 \times T_2$ раstra относительно исходного положения $\lambda = 0$ для элементов матричного изображения x_{ij} можно записать

$$x_{ij}(\lambda) = x_{ij}(0) + \lambda^T \nabla x_{ij}, \quad i = \overline{1, T_1}; j = \overline{1, T_2},$$

где ∇x_{ij} — градиент яркости изображения в точке ij при $\lambda = 0$; t — знак транспонирования. Совмещение изображений $x_{ij}(\lambda)$ и $x_{ij}(0)$ методом наименьших квадратов в условиях аддитивного шума ε_{ij} представляет собой задачу $\sum_{ij} (\lambda^T \nabla x_{ij} - \varepsilon_{ij})^2 = \min_{\lambda}$, которая при математических ожиданиях $M(x_{ij}) = 0$ приводит к несмещенным оценкам (случайным ошибкам)

$$\hat{\lambda} = A_0^{-1} n^{-1} \sum_{ij} \varepsilon_{ij} \nabla x_{ij} \quad (1)$$

с минимальными дисперсиями. Предполагается, что обратная матрица