

УДК 629.7.058.74

А. А. Лубков, В. В. Полубинский, С. В. Храмов
(Новосибирск)

РАБОТА С БАЗОЙ ДАННЫХ В СИСТЕМЕ СИНТЕЗА ВИЗУАЛЬНОЙ ОБСТАНОВКИ ВЫСОКОЙ ПРОИЗВОДИТЕЛЬНОСТИ

Показана целесообразность применения специализированного процессора базы данных в системах синтеза визуальной обстановки высокой производительности (до 100 тыс. граней за кадр). Предложен принцип динамического обновления информации локальной базы данных, а также критерии для определения необходимости этого обновления.

В настоящее время в отечественной практике началось применение систем синтеза визуальной обстановки (ССВО) для авиационных тренажеров. Они позволяют имитировать разнообразные ситуации полета, обезопасить и удешевить подготовку пилотов.

Для синтеза визуальной обстановки в реальном времени (при обновлении кадра изображения каждые 20—40 мс) требуются значительные вычислительные ресурсы. С развитием БИС и СБИС появляется возможность постоянно увеличивать производительность ССВО, что прослеживается в разработках отечественных систем [1, 2]. Производительность этих систем составляет 200, 1000, 5000 граней за кадр.

По мере роста производительности ССВО и объемов используемых ими баз данных возникает потребность в качественно новом подходе к организации работы ССВО с базой данных.

В данной работе показана необходимость введения специализированного процессора, предложено решение, позволяющее обеспечить взаимодействие ССВО высокой производительности (до 100 тыс. граней/кадр) и базы данных окружающей обстановки. Рассматривается также способ уменьшения объема ОЗУ для хранения базы данных, основанный на динамическом процессе обновления локальных данных.

Математическая модель мира для ССВО. База данных содержит математическую модель окружающей обстановки. Модель включает описание объектов, их расположение в пространстве, связи между ними, а также свойства поверхностей — цвет, текстуру и некоторые другие параметры. Способ описания подобной модели представлен в [3].

Для статистической оценки баз данных можно воспользоваться моделями некоторых объектов. Статистика, собранная для ССВО производительностью 5000 граней/кадр, приведена в табл. 1. Из данных, помещенных в ней, можно вывести количественные соотношения между различными элементами моделей.

Исключив из рассмотрения крайние значения по количеству матриц — изображения взлетно-посадочной полосы и «Бурана 1» — и усреднив оставшиеся, получим соотношение

матрицы: плоскости: грани = 1 : 17 : 234

В ССВО высокой производительности должно увеличиваться количество разделяющих плоскостей, приходящихся на одну матрицу, поэтому для оценки удобнее взять соотношение

матрицы : плоскости : грани = 1 : 20 : 300,

плоскости : грани = 1 : 15

В базе данных для ССВО реального времени можно выделить две части: непосредственно данные, описывающие объекты, и структуру базы данных. Удобной структурой для описания базы данных является двоичное дерево. Данные и дерево могут располагаться в физически разных местах. Результатом обработки структуры является последовательность ссылок на описания объектов, попадающих в поле зрения наблюдателя, и результирующих матриц для преобразования объектов из мировой системы координат в систему координат наблюдателя. По ссылкам из памяти данных выбираются описания объектов, которые совместно с матрицами преобразований составляют файл для геометрического процессора. Процесс формирования этого файла возлагается на сценарный процессор. Место сценарного процессора в блок-схеме тренажера показано на рис. 1.

Т а б л и ц а 1

Название	Матрица	Разделяющая плоскость	Грань
Станция	5	200	2936
Корабль	17	321	3428
ВПП	72	283	2480
«Буран 1»	2	39	985
«Буран 2»	17	161	2060

Как описано в [3], каждый объект сцены представлен последовательностью сегментов. Все сегменты объекта фиксированы в пространстве относительно друг друга. Могут быть вложенные объекты, относительно положение которых задается матрицами. Матрицы движения объектов формируются моделирующим комплексом. Сценарный процессор определяет расположение наблюдателя относительно разделяющих плоскостей и формирует приоритетный список ссылок на потенциально видимые сегменты. При изменении положения наблюдателя соответственно изменяется порядок выдачи ссылок на сегменты. Возможна как программная, так и аппаратная реализация процессора. Ниже рассматриваются основные операторы, используемые для описания дерева базы данных.

Узлами дерева являются операторы «Разделяющая плоскость», которые содержат один вход и два выхода (рис. 2). В этих узлах происходит анализ положения наблюдателя относительно плоскости, заданной коэффициентами уравнения в пространстве. Для определения положения наблюдателя относительно плоскости достаточно вычислить знак расстояния от наблюдателя до плоскости, который определяет приоритетное направление обхода. Поддерево, определяющее полупространство, в

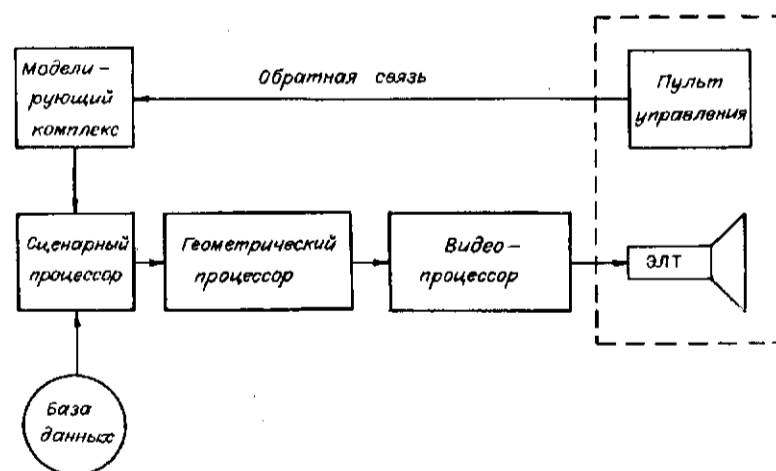


Рис. 1

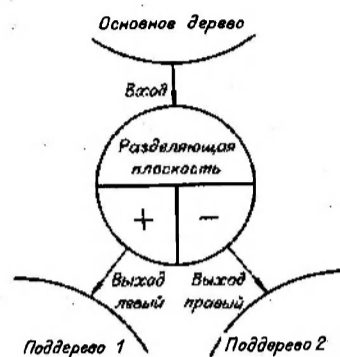


Рис. 2

котором находится наблюдатель, обрабатывается первым, после чего обрабатывается второе поддерево.

Наблюдатель задается четырехгранной пирамидой видимости, вершина которой располагается в начале его системы координат. При этом пирамида видимости отсекает в трехмерном пространстве некоторую часть, попадающую в поле зрения наблюдателя. В дерево базы данных, кроме разделяющих плоскостей, включены команды, позволяющие уменьшить объем вычислений: «Сфера», «Экран», «Детализация». Перечисленные команды позволяют исключить из процесса обхода дерева части, соответствующие объектам, заведомо

не попадающим в пирамиду видимости.

Линейный оператор «Сфера» позволяет определить входимость некоторой группы объектов в пирамиду видимости. В случае, если объекты не попадают в поле зрения, то обход соответствующего поддерева не производится. Поскольку пирамида видимости выхватывает из окружающего пространства небольшую часть, то применение данного оператора значительно экономит вычислительные ресурсы. Экономия основана на том, что вместо проверки входимости в пирамиду видимости группы объектов проверяется пересечение этой пирамиды со сферой, целиком включающей в себя тестируемые объекты. Если пересечение пусто, то дальнейший обход соответствующего поддерева не происходит.

В системах визуализации существует потребность введения подвижных и повторяющихся объектов. Описание каждого такого объекта можно задать единственным поддеревом. Для получения множества одинаковых объектов используется несколько обращений к соответствующему поддереву с различными матрицами преобразования в систему координат наблюдателя. Эта функция также возлагается на сценарный процессор. Для получения подвижных объектов достаточно менять матрицу перемещения от кадра к кадру. Формирование матриц перемещения осуществляется моделирующим комплексом.

Требуемые характеристики сценарного процессора. Для оценки необходимой вычислительной мощности сценарного процессора необходимо задаться качественным составом базы данных и количеством элементарных операций для выполнения операторов. Табл. 2 содержит список основных операторов и соответствующие им количества элементарных операций, а также объем памяти (в 32-разрядных словах), требуемый для их размещения. Можно предполагать, что количественное соотношение

Т а б л и ц а 2

Оператор	Количество сложений	Количество умножений	Количество обращений к памяти	Объем команды (32 р. слов)
«Матрица»	36	45	111	15
«Сегмент»	—	—	2	1
«Разделяющая плоскость»	3	3	7	5
«Габаритная сфера»	23	21	17	5

операторов в базе данных не изменяется с ростом производительности системы, хотя следует ожидать увеличения количества граней, попадающих на каждую матрицу. Оценка такого увеличения весьма затруднительна, поэтому расчет производительности процессора должен производиться с некоторым запасом. Предположим также, что команда «Сфера» существует для каждого из объектов, описанных в табл. 1. Допускаем округленное отношение числа сфер к числу граней 1 : 2000.

В среднем объеме данных для синтеза изображения объектов, попадающих в поле зрения наблюдателя, в несколько (8—10) раз меньше объема глобальной базы данных. На основании всех этих допущений, задав производительность системы и частоту повторения кадров, можно вычислить объем памяти команд сценарного процессора. Расчет приводится для ССВО с производительностью 100 тыс. граней за полукадр (20 мс).

В табл. 3 приведено количество основных команд сценарного процессора для базы данных объемом 800 тыс. граней. Из нее следует, что для хранения дерева базы данных необходимо приблизительно 364 тыс. слов по 32 разряда или 1,46 мВ. Такой объем памяти вполне доступен недорогой ЭВМ общего назначения. Гораздо сложнее с вопросом производительности.

Оценим объем вычислений, необходимых для обхода дерева базы данных производительностью 100 тыс. граней за 20 мс. В табл. 4 приведены результаты расчетов. Для операций с плавающей запятой (сложение и умножение) можно принять одинаковое время выполнения. В этом случае производительность сценарного процессора составляет приблизительно 3,5 Мфлопс и память — 4,1 млн обр. в с. Этот расчет учитывает только чистое время выполнения операций. В действительности возможны некоторые простои оборудования, что требует повышения производительности (в 1,5—2 раза). Следует также учесть, что нет возможности ограничить вычисления строго над теми поддеревьями, которые явно попадают в пирамиду видимости, так как команда «Сфера» не является идеальным «отсекателем» объектов вследствие различия в размерах объекта и охватываемой его сферы. Еще одним фактором, который требует ресурсы сценарного процессора, является необходимость взаимодействия с моделирующим комплексом (обновление матриц подвижных объектов, установка ключей), а также обработка команд, не учтенных в таблицах. Все это требует производительности порядка 6—9 Мфлопс и распараллеливания вычисления и обращений к памяти.

Для реализации сценарного процессора рассматривались три варианта:

1. Использование ЭВМ общего назначения и программная реализация необходимых функций.
2. Применение вычислительных ресурсов геометрического процессора.
3. Разработка специализированного процессора.

При использовании первого варианта обработка каждого оператора дерева базы данных требует вызова соответствующей подпрограммы, что нуждается в дополнительных вычислительных ресурсах сверх приведенных выше. Необходимо также дешифровать операторы и идентифицировать их с соответствующей подпрограммой. Несомненно, что можно найти ЭВМ, обеспечивающую такую производительность, однако ее стоимость и габариты могут быть неприемлемы. На верхнем пределе стоимости и габаритов находится мини-ЭВМ «Электроника 82»

Т а б л и ц а 3

Оператор	Количество команд	Занимаемый объем (32 р. слов)
«Матрица»	2670	40050
«Сегмент»	53330	53330
«Разделяющая плоскость»	53330	266650
«Габаритная сфера»	400	2000
Всего		362030

Т а б л и ц а 4

Оператор	Количество команд	Количество сложений	Количество умножений	Количество обращений к памяти
«Матрица»	330	11880	14850	36630
«Сегмент»	6660	—	—	13320
«Разделяющая плоскость»	6660	19980	19980	46620
«Габаритная сфера»	50	1150	1050	850
Всего		33010	35880	97420

(VAX-750). Ее производительность примерно на порядок ниже требуемой (1 Мфлопс, 200 нс — цикл памяти). Применение более мощных машин не оправдано из-за их больших габаритов и стоимости.

Второй вариант используется в системе «Альбатрос», разработанной в Институте автоматики и электрометрии СО АН СССР [2]. Геометрический процессор состоит из нескольких взаимодействующих между собой процессоров с плавающей точкой общей производительностью 60 Мфлопс. Часть времени обработки кадра отводится для обхода дерева, а часть — на геометрические преобразования. Поскольку основные вычислительные ресурсы требуются для геометрических преобразований, то и структура геометрического процессора ориентирована на них. Распараллеливание вычислений ориентировано на операции умножения векторов и матриц. При обработке дерева базы данных производительность резко снижается (в 3—4 раза) и возникают простои оборудования. При этом время, затрачиваемое на обработку дерева, составляет 3—20 % от общего времени вычислений. Совмещение функций сценарного и геометрического процессоров приводит к усложнению схемы. Поочередность обработки дерева и геометрических преобразований создает дополнительную задержку реакции ССВО на воздействие моделирующего комплекса, требует значительный буфер для хранения промежуточных результатов. Указанное решение приемлемо при обработке 5000 граней за кадр, но потери быстродействия и дополнительный буфер делают этот вариант технически трудно реализуемым при повышении производительности системы до 100 тыс. граней за кадр.

Третий вариант — построение специализированного процессора с заданными выше параметрами — адекватен задаче. Поскольку основной операцией, выполняемой этим процессором, является скалярное умножение, то его вычислитель использует конвейер из соединенных последовательно умножителя и сумматора с плавающей точкой, управляемых от микропрограммного устройства. Такое решение позволяет достигнуть производительности 10 Мфлопс (серия 1843) на операциях умножения векторов и матриц, что соответствует поставленной задаче.

Спецпроцессор с микропрограммным управлением можно рассматривать как процессор с системой команд, являющихся операторами дерева базы данных. В этом случае можно говорить о программе обработки данных. Наличие языков описания моделей высокого уровня [4] позволяет с помощью компилятора для данного процессора использовать уже существующие модели объектов.

Динамическое изменение памяти данных. Еще одна проблема сценарного процессора — соотношение объема и быстродействия памяти данных.

С одной стороны, память данных должна содержать информацию, описывающую окружающую обстановку для всего процесса тренировки, с другой стороны, в поле зрения наблюдателя в каждом из кадров

изображения попадает лишь 10—15 % всей обстановки. При хранении базы данных на НМД полезным было бы производить подкачку в ОЗУ тех данных, которые требуются при текущем положении наблюдателя.

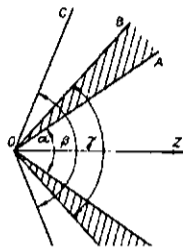
В случае системы [1] с производительностью 1000 граней за кадр требуемая производительность памяти составляет 50 тыс. выборок в с. В системе «Альбатрос» [2] соответствующая цифра составляет 250 тыс. выборок в с. Эти требования позволяют при выборе типа памяти ориентироваться только на ее объем, так как необходимое быстродействие достигается при использовании практически любых БИС ЗУ.

В системе синтеза высокой производительности требования к быстродействию памяти резко возрастают при постоянном росте ее объема. Так, при производительности ССВО 100 тыс. граней за кадр быстродействие должно составлять 200 нс и минимальный объем — 100 тыс. слов. Выше показано, что полный объем базы данных должен превышать количество граней, обрабатываемых в кадре в несколько (8—10) раз. Описание грани в зависимости от метода обработки составляет 30—100 байт. В него входят координаты вершин, нормали в вершинах, цвет, полупрозрачность, текстурные параметры и другая информация. Общий объем данных в этом случае составляет 24—160 мВ. Построение памяти такого объема и быстродействия при современном уровне развития СБИС представляется нецелесообразным, если учесть, что вся она одновременно не используется.

Таким образом, возникает потребность в организации локальной базы данных, содержащей данные, необходимые для отображения в текущем кадре. Необходим также механизм, обеспечивающий поддержание этой локальной базы данных при перемещении пирамиды видимости. Следует очищать память от неиспользуемой информации и по мере надобности подгружать новые необходимые данные. Назовем эти процессы чисткой и подкачкой, а их совокупность — обновлением базы данных.

Критерием осуществления чистки/подкачки объекта предлагается считать его угловое расстояние от оси пирамиды видимости либо от плоскостей пирамиды видимости. При перемещении пирамиды видимости в нее попадают новые объекты. Для того чтобы отобразить их, необходимо наличие соответствующих данных в локальной базе данных. Из-за значительной инерционности НМД по сравнению с ССВО необходима предварительная загрузка данных. Для этого предлагается ввести дополнительную пирамиду — пирамиду подкачки. Пирамида подкачки должна иметь большие угловые размеры, чем пирамида видимости, чтобы объекты, находящиеся вблизи пирамиды видимости, заранее подкачивались в память. Поскольку одновременно может подкачиваться только один объект, то для нормальной работы механизма подкачки требуется такая регулировка пирамиды подкачки, при которой в нее попадает не более одного объекта в дополнение к существующим в пирамиде видимости. Такой критерий регулировки дает возможность подкачивать первыми те объекты, которые вероятнее всего попадут в ближайшее время в пирамиду видимости.

В процессе перемещения пирамиды видимости часть объектов выходит из поля зрения наблюдателя. Для освобождения локальной памяти данных от ненужной информации служит процесс чистки. Критерием отбраковки описания объекта также служит его угловое расстояние от оси или плоскостей пирамиды видимости. Для оценки этого расстояния вводится еще одна пирамида — пирамида чистки. Описания всех объектов, не попадающих в нее, выбрасываются из локальной базы данных. Критерием для установки границ пирамиды чистки является количество свободного места в памяти. Если места в локальной базе данных много, то можно расширить границы пирамиды чистки и таким образом уменьшить количество отбраковываемых объектов. В случае дефицита свободного пространства в памяти и при необходимости загрузить какой-либо объект необходимо сузить пирамиду чистки и таким образом спровоцировать чистку наиболее отдаленного от оси зрения объекта. Продольное сечение



пирамид видимости, подкачки, чистки показано на рис. 3 (AOA' — пирамида видимости, BOB' — пирамида подкачки, COC' — пирамида чистки).

Рассмотренный принцип динамического регулирования позволяет обеспечить загрузку локальной памяти данных необходимой в данный момент информацией, реализовать наиболее полную загрузку памяти, уменьшить поток данных от НМД. В частном случае, если объем локальной памяти данных превышает объем всей базы

При аппаратной реализации принципа возникает вопрос о динамическом распределении памяти данных. Однако рассмотрение этого вопроса выходит за рамки статьи.

ВЫВОД

Повышение производительности ССВО требует качественно нового подхода к взаимодействию с базами данных, а именно: применения специализированного сценарного процессора и введения локальной базы данных с эффективным механизмом ее поддержки.

СПИСОК ЛИТЕРАТУРЫ

1. Ковалев А. М., Талныкин Э. А. Машинный синтез визуальной обстановки // Автометрия.—1984.—№ 4.
2. Долговесов Б. С. Архитектура системы отображения трехмерных объектов в реальном времени широкого назначения // Программа и тез. докл. V всесоюз. конф. «Машинная графика 89».— Новосибирск: ВЦ СО АН СССР, 1989.
3. Гусев А. В., Ивашин С. Л., Талныкин Э. А. Математические модели сцен в синтезирующих системах визуализации реального времени // Автометрия.—1985.—№ 4.
4. Гусев А. В., Талныкин Э. А. SDL — язык описания трехмерных сцен в системах динамической машинной графики // Автометрия.—1986.—№ 4.

Поступила в редакцию 15 ноября 1990 г.

УДК 519.219 : 519.237.5

О. А. Степанов

(Ленинград)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СТРУКТУРЫ РАЗЛИЧНЫХ АЛГОРИТМОВ СОВМЕЩЕНИЯ ГАУССОВЫХ ПРОЦЕССОВ И ПОЛЕЙ

Рассмотрена структура алгоритма решения общей задачи совмещения зашумленных реализаций гауссовых процессов и полей, основанная на методе максимума функции правдоподобия. Установлена взаимосвязь этого алгоритма с алгоритмами, полученными в предположении наличия точного эталона, и алгоритмами, вытекающими из так называемого метода обобщенной корреляции. Предложен механизм адекватного учета неидеальности эталона при решении задачи совмещения.