

Глава 1

Общие сведения о компьютерной графике

Данная глава знакомит с общей информацией о компьютерной графике: историей развития, способами получения изображений на экране компьютера, особенностями векторной и растровой графики. Также рассматриваются функции графических систем, с помощью которых получают графические изображения.

1.1. История развития компьютерной графики

При всем великолепии изображений, полученных с помощью компьютера, и впечатляющих эффектов, которые мы можем сегодня наблюдать, все начиналось совсем не так легко. Компьютерная графика пробивалась через тернии технических ограничений и зарождающегося нового мышления. Прошло немногим более полувека ее существования, а добилась она за это время высоких вершин. И продолжает покорять их с большой скоростью, неведомой многим другим ее собратьям. За время своей жизни компьютерная графика находит все больше и больше ценителей и друзей.

Рождение компьютерной графики можно отнести к середине XX века, когда появилась минимальная возможность получать незатейливые рисунки с помощью компьютера. Чаще всего это были графики и простые схемы. Простейшие графические программы были привязаны к конкретному компьютеру, что не позволяло передавать графические изображения и тем самым мешало развитию. Процесс получения компьютерных изображений был весьма дорогим, так как требовал достаточного объема памяти для хранения графических данных и регенерации для получения динамических изображений. Поэтому лишь немногие и особо богатые фирмы могли позволить себе эту роскошь. Первыми, кто использовал компьютерную графику, были состоятельные организации и, конечно, военная промышленность.

История развития методов вывода

Первые дисплейные устройства, которые были разработаны в 1960-х годах, назывались *векторными*.

Векторные дисплеи состоят из трех основных частей (рис. 1.1):

- дисплейный процессор (ДПц);
- буферная память;
- электронно-лучевая трубка (ЭЛТ).

вы хранятся в памяти для регенерации в виде совокупности образующих их точек, называемых *пикселями*. Значения пикселей хранятся в битовой карте, которая и является в данном случае дисплейной программой. Изображение, формируемое на растре, представляет собой совокупность горизонтальных растровых строк, каждая из которых состоит из отдельных пикселей.

Таким образом, *растр* — это матрица пикселей, покрывающая всю площадь экрана. Все изображение последовательно сканируется около 30 раз в секунду по отдельным строкам растра в направлении сверху вниз (рис. 1.2).

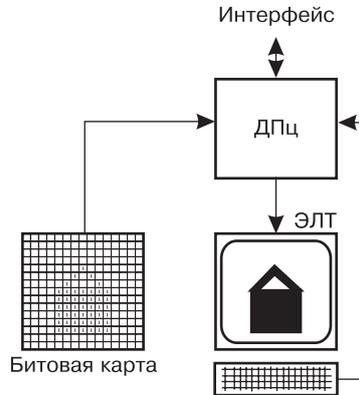


Рис. 1.2. Устройство растрового дисплея¹

Растровые дисплеи обладают следующими достоинствами:

- ❑ растровая графика по сравнению с векторной легче закрашивает изображения. В векторных дисплеях необходимость закраски области сильно увеличивает размер дисплейной программы, так как закраска происходит векторами. В растровом же дисплее эта процедура не влияет на размер дисплейной программы, так как меняются лишь значения внутренних пикселей;
- ❑ процесс регенерации не зависит от сложности рисунка. Векторные же дисплеи часто начинают мерцать, когда количество примитивов в буфере становится таким большим, что его нельзя считать и обработать за время, отведенное на считывание одного кадра, в результате чего требуется больше времени, а это может понизить частоту регенерации изображения;
- ❑ дешевизна.

При указанных выше достоинствах растровые дисплеи имеют и недостатки:

- ❑ необходимо больше памяти, так как полное изображение, состоящее из пикселей, должно храниться как битовая карта. В векторном дисплее вектор задается двумя точками, в растровом необходимо хранение и всех промежуточных;
- ❑ разрешающая способность растровых графических систем пока еще ниже, чем векторных (1280 × 1024 против 4096 × 4096 пикселей);

¹ Климов В. Е. Графические системы САПР. — М.: Высшая школа, 1990.

- для отображения примитивов необходимо больше времени, так как в векторных дисплеях задаются две конечные точки (для отрезка), а в растровом нужно рассчитать еще и все промежуточные точки отрезка.

Развитие методов ввода

Параллельно с совершенствованием методов вывода улучшались и методы ввода. Громоздкое и хрупкое световое перо вытесняется тонкой указкой, которую перемещают по электронному планшету, или же смонтированной на экране прозрачной сенсорной панели, реагирующей на прикосновение. Кроме того, большие надежды возлагают на речевую связь, которая помогает вводить информацию без помощи рук и выводить ее в естественном виде.

Развитие программного обеспечения

Многие трудности в развитии графического программного обеспечения были связаны с его примитивностью. Процесс совершенствования программного обеспечения был длительным и медленным. Был пройден путь от аппаратно-зависимых пакетов низкого уровня, поставляемых изготовителями вместе с конкретными дисплеями, к аппаратно-независимым пакетам высокого уровня. Такие пакеты могут быть использованы для управления самыми разнообразными графическими устройствами. Основная цель аппаратно-независимого пакета — обеспечение мобильности прикладной программы при переходе от одного компьютера к другому.

1.2. Основные сведения о графических системах

Процессы проектирования в различных областях техники связаны с созданием и модификацией моделей объектов проектирования. Значительную часть этих моделей составляют данные о геометрических или графических характеристиках объектов.

Таким образом, *компьютерная графика* — это область деятельности, в которой компьютеры используются для создания, хранения и обработки моделей объектов и их изображений.

Графические системы служат для создания, поиска, хранения и модификации графических данных.

Графические системы могут быть пассивными и интерактивными.

- *Пассивные* — обеспечивают только вывод графических изображений, но человек при этом не имеет возможности прямого воздействия на графические преобразования.
- *Интерактивные* — дают возможность пользователю динамически управлять содержимым изображения. В таких системах используются интерактивные дисплеи, позволяющие работать в диалоге с графическим изображением.

Области применения графических систем представлены в табл. 1.1.

Таблица 1.1. Области применения графических систем

Область применения	Синтез изображения	Анализ изображения	Обработка изображения
Вход	Формальное описание	Визуальное представление	Визуальное представление
Выход	Визуальное представление	Формальное описание	Визуальное представление
Объект	Линии, пиксели, объемы, текст	Сгенерированное или сканируемое изображение	Сканируемое изображение
Задачи	Генерация, преобразование изображения	Распознавание образов, структурный анализ, анализ сцен	Повышение качества изображения

Синтез изображения позволяет получать на базе описания объекта пользователем его геометрическую модель с последующим ее отображением. Анализ изображения выполняет обратную задачу — на базе уже имеющегося графического изображения, сгенерированного ранее с помощью графической системы или отсканированного, дает формальное описание. Для этого необходимы специальные алгоритмы, выполняющие или распознавание образов, или структурный анализ, или анализ сцен в трехмерной графике.

1.3. Функции графических систем

Интерактивные графические системы выполняют следующие функции:

- ввод данных;
- вывод графических изображений;
- обработка запросов пользователя;
- поиск и хранение данных;
- реализация преобразований графической информации.

Функции ввода реализуются с помощью графических устройств ввода: клавиатуры, планшета, мыши, светового пера и т. д.

Функции вывода — с помощью графических устройств вывода: графопостроителя, дисплея, станка с ЧПУ.

Функции обработки запросов пользователя на входных и командных языках реализуются программой, называемой лингвистическим (диалоговым) процессором. Процессор преобразует описания геометрии объектов, заданные на входных языках, в формы, принятые в системе. В настоящее время наиболее эффективный метод работы пользователя с графической системой — диалог с использованием меню. Данные, получаемые системой через диалоговый процессор, делятся на два класса: параметры объекта и коды для управления графической системой. Первые поступают из входных языков, вторые — из командных. Параметры объекта направляются через СУБД в базу данных. Коды для управления графической системой поступают в монитор. Он управляет работой системы.

Организация базы данных графической системы определяется классами моделей объектов. Если объекты проектирования имеют графическое представление (схемы, планы, чертежи), в базе данных хранятся модели графических изображений этих объектов. Ориентация системы на объект определяет наличие в базе данных геометрических моделей объектов в трехмерном пространстве.

Формирование моделей и их модификаций, а также преобразование этих моделей выполняет геометрический процессор. В зависимости от сложности модели объекта в системе может исполняться несколько геометрических процессоров.

Геометрический процессор может выполнять и следующие функции:

- построение сечений и разрезов;
- проверка корректности геометрической компоновки узла конструкции;
- моделирование работы робота.

Для систем, работающих с двумерными геометрическими объектами, функции формирования, модификации и преобразования геометрической модели выполняет графический процессор.

1.4. Блок-схема графической системы

Блок-схема графической системы приведена на рис. 1.3. В памяти размещаются два важных информационных модуля:

- прикладная структура данных, содержащая описание объектов, изображения которых должны показываться на экране. Она же является моделью объектов;
- дисплейная программа, которая формируется графическим пакетом и читается дисплейным процессором во время регенерации изображения на экране.

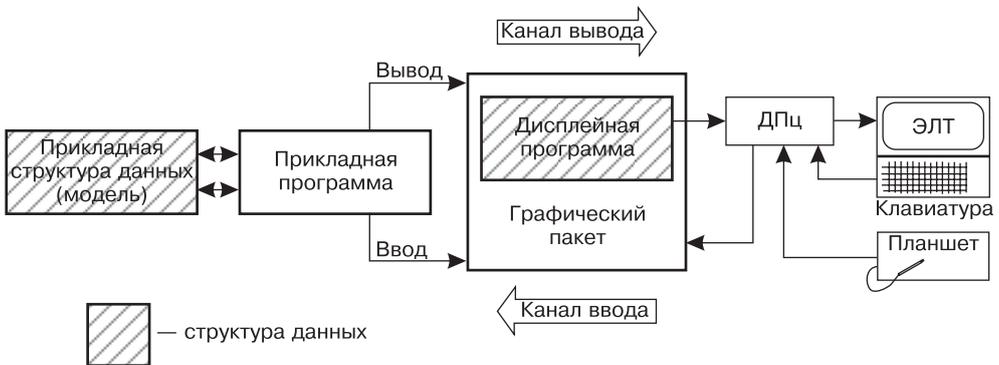


Рис. 1.3. Блок-схема графической системы¹

Канал вывода (от описания объекта к его изображению).

Прикладная программа извлекает информацию из прикладной структуры данных и записывает ее, а затем направляет графические команды, которые обраба-

¹ Александров В. В., Шнейдеров В. С. Рисунок, картина, чертеж на ЭВМ. — Ленинград: Машиностроение, 1988.

тываются графическим пакетом. Последний формирует дисплейную программу, используемую дисплейным процессором для получения изображения. Таким образом, канал вывода последовательно преобразует описание объекта в структуру представления, принятую в дисплейной программе.

Канал ввода (от устройств ввода к структуре данных и дисплейной программе).

Дисплейный процессор регистрирует факт использования устройства ввода и либо прерывает, либо передает данные по запросу. Ввод данных от дисплейного процессора осуществляет специальная программа ввода, которая передает их прикладной программе. Эти данные меняют состояние прикладной программы. Они могут также побудить прикладную программу модифицировать структуру данных, изменить параметры.

1.5. Получение изображения на векторном дисплее

В векторных дисплеях электронный луч, управляемый дисплейным процессором, создает изображение, двигаясь от точки к точке по отрезкам прямых, которые называются *векторами*. Для получения изображения необходимы хранение координат точек и генератор векторов. С помощью генератора векторов можно рисовать прямолинейные участки символов и осуществлять кусочно-линейную аппроксимацию кривых. При рисовании кривой необходимо учитывать следующую особенность: в областях, где кривая имеет малый радиус кривизны, отрезки прямых должны быть короче, чем в областях, где радиус кривизны относительно велик.

Максимальная скорость вычерчивания векторов — около 2 см/мкс. В лучших системах за один цикл регенерации выводится 6–10 тыс. векторов длиной 2–3 см.

Генератор векторов

Генератор векторов должен выполнять следующие действия.

1. Перемещать электронный луч из позиции $(x_{\text{пред}}, y_{\text{пред}})$ в позицию (x, y) по прямой линии или, по крайней мере, по линии, которая кажется наблюдателю прямой.
2. Рисовать все линии с одинаковой яркостью. Чем медленнее движется луч, тем ярче светится получаемая линия. Постоянная яркость может быть достигнута либо перемещением луча с постоянной скоростью для всех векторов, либо изменением интенсивности луча для разных векторов.
3. Включать луч точно в тот момент, когда он начинает перемещаться из начальной точки, и выключать в момент прихода в конечную точку.
4. Осуществлять все операции как можно быстрее.

Генератор векторов должен управлять тремя параметрами:

- отклонением по оси X ;
- отклонением по оси Y ;
- интенсивностью.

Если векторы рисуются с одной и той же скоростью, сигнал управления интенсивностью превращается просто в сигнал его включения и выключения. Для изображения прямой линии, проходящей от точки (x_1, y_1) к точке (x_2, y_2) , необходимо дополнить массивы X и Y координатами точек, лежащих приблизительно на этой прямой. Приблизительно, потому что для некоторого целого x соответствующее ему y может оказаться нецелым. Поэтому возникает необходимость в округлении или отсечении.

Относительные координаты

Рассмотрим дисплейную программу, изображающую треугольник (рис. 1.4).

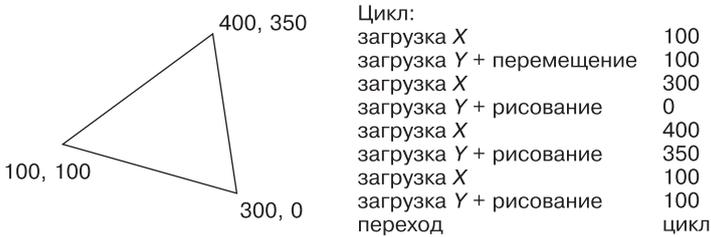


Рис. 1.4. Пример дисплейной программы, изображающей треугольник¹

Все идет хорошо до тех пор, пока в прикладной программе не возникнет необходимость переместить треугольник в другую точку экрана, например на $\Delta x = -50$, $\Delta y = 100$. Существуют два способа решения этой задачи.

Первый — прикладная программа находит описание треугольника в прикладной структуре данных, изменяет значения координат вершин и вызывает графический пакет для создания новой дисплейной программы.

Или второй, более эффективный. Прикладная программа выдает графическому пакету запрос на изменение сегмента дисплейной программы, который генерирует треугольник. Изменение заключается в вычитании 50 из всех x -координат и прибавлении 100 ко всем y -координатам. Для треугольника эта процедура будет выполнена быстро, но для сложных объектов она будет более трудоемкой.

Еще проще реализуется перемещение, когда координаты объекта можно задавать абсолютными и относительными величинами. Дисплейная программа имеет вид, представленный на рис. 1.5.

Цикл:	
загрузка X абс.	100
загрузка Y абс. + перемещение	100
загрузка X отн.	200
загрузка Y отн. + рисование	-100
загрузка X отн.	100
загрузка Y отн. + рисование	350
загрузка X отн.	-300
загрузка Y отн. + рисование	-250
переход	цикл

Рис. 1.5. Пример дисплейной программы (координаты объекта заданы абсолютными и относительными величинами)

¹ Александров В. В., Шнейдеров В. С. Рисунок, картина, чертеж на ЭВМ. — Ленинград: Машиностроение, 1988.

Таким образом, для перемещения треугольника достаточно модифицировать значения начальных координат x и y .

Генератор символов

Существуют четыре способа генерации символов:

- метод маски;
- метод Лиссажу;
- штриховой метод;
- метод точечной матрицы.

Метод маски

Луч ЭЛТ описывает фигуру, как на рис. 1.6 (24 штриха). Символ генерируется путем высвечивания одних штрихов и пропуска других. Символ кодируется 24-разрядным словом, в котором каждый разряд представляет один штрих.

Если в разряде 0, то штрих пропускается, если 1, то штрих высвечивается.

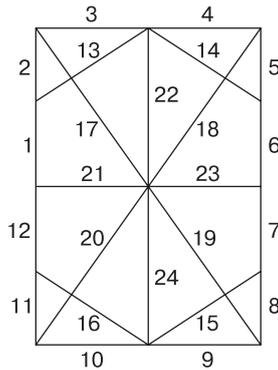


Рис. 1.6. Метод маски

Недостатки метода:

- низкое качество символов;
- низкая скорость, так как луч должен пойти по всей маске для любого символа;
- необходима перекодировка из ASCII в специальный 24-разрядный код.

Метод Лиссажу

Для формирования символов используются фигуры Лиссажу. Этот способ получения символов является аналоговым. Различные возможности генерирования примитивов (отрезков и дуг), из которых формируются символы, показаны на рис. 1.7.

Достоинства метода:

- хорошее качество, плавность линий, что особенно важно для строчных букв;
- любые формы символов.

Недостатки метода:

- низкая скорость;
- высокая стоимость аппаратной части.

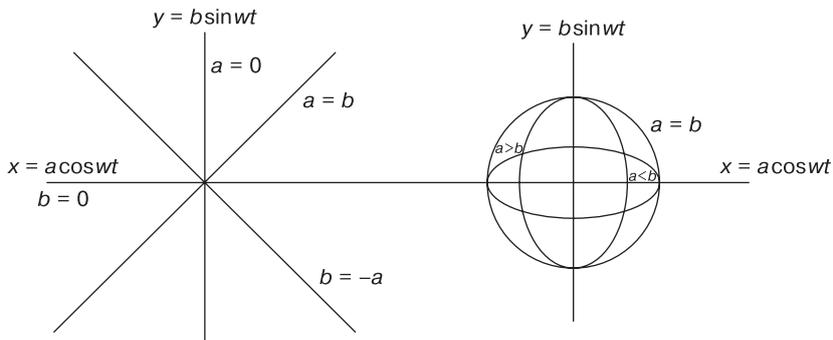


Рис. 1.7. Метод Лиссажу

Штриховой метод

Штриховой генератор символа представляет собой аналоговое устройство, которое выдает волны различной формы отдельно для отклонения луча ЭЛТ по направлениям X и Y , а также сигнал яркости в виде «вкл./выкл.» (рис. 1.8).

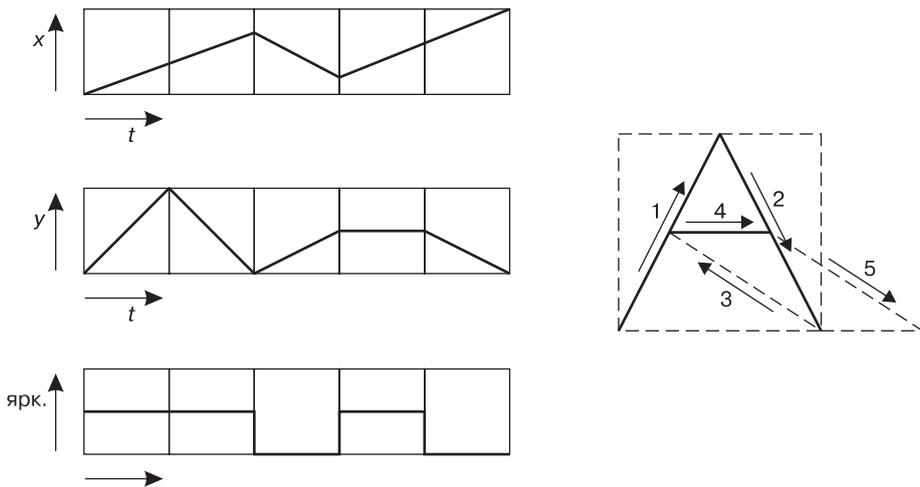


Рис. 1.8. Штриховой метод

Достоинства метода:

- высокая скорость (1 символ \approx 10 мкс);
- хорошее качество;
- можно задавать любые размеры символов.

Недостатком указанного метода является его высокая стоимость.

Метод точечной матрицы

Генератор символов с точечной матрицей является чисто цифровым устройством, поэтому он надежный и недорогой. Символы будут удовлетворительного

качества при условии, что они малы. Поэтому они часто используются в алфавитно-цифровых дисплеях.

Символ генерируется путем высвечивания последовательности точек. Луч ЭЛТ последовательно проходит через поля матрицы символа. В каждом поле матрицы луч может быть либо включен, либо выключен (рис. 1.9).

Для символа 5×7 необходимо 35 бит.

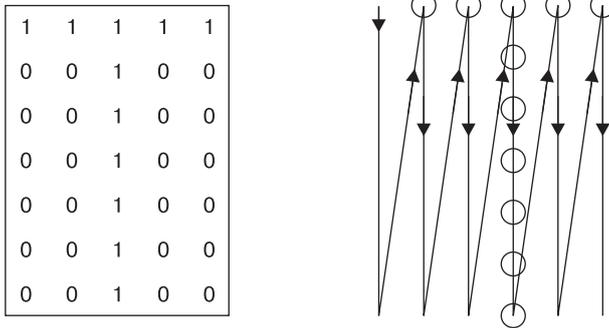


Рис. 1.9. Метод точечной матрицы

Достоинством данного метода является низкая стоимость, а недостатком — плохое качество при увеличении размера букв.

1.6. Получение изображения на растровом дисплее

В системах с растровым сканированием кодирование изображения проще, чем в рассмотренных выше. Рисуемые на экране примитивы разбиваются на составляющие их точки. Основное различие между векторным и растровым дисплеями состоит в организации памяти, хранящей точки. В векторном дисплее точки, составляющие каждый примитив, хранятся в памяти последовательно и рисуются в этом же порядке, примитив за примитивом, так как луч может двигаться по экрану произвольным образом. В растровом дисплее память регенерации организована в виде двумерного массива. Элемент, находящийся на пересечении строки и столбца, хранит значение яркости или цвета соответствующей точки экрана. Верхняя строка массива соответствует верхней строке раstra. Регенерация изображения осуществляется последовательным сканированием буфера по строкам раstra.

Простой буфер регенерации содержит по 1 биту на пиксел и, таким образом, определяет двухцветное (черно-белое) изображение. Задача системы вывода изображения состоит в циклическом просмотре буфера регенерации по строкам, обычно от 30 до 100 раз в секунду. Адреса памяти генерируются синхронно с координатами раstra, и содержимое выбранных элементов памяти используется для управления интенсивностью электронного луча (рис. 1.10).

Генератор растровой развертки формирует сигналы отклонения и управляет адресными X- и Y-регистрами, определяющими следующий элемент буфера регенерации.