

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО  
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РСФСР

КУЙБЫШЕВСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ  
АВИАЦИОННЫЙ ИНСТИТУТ имени академика С. П. КОРОЛЕВА

О. П. АЛЕКСАНДРОВ

ТЕХНИЧЕСКИЕ СРЕДСТВА  
И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ  
МАШИНОЙ ГРАФИКИ

Утверждено  
редакционно-издательским  
советом института  
в качестве учебного пособия

КУЙБЫШЕВ 1986

Александров О. П. **Технические средства и программное обеспечение машинной графики: Учебное пособие.** — Куйбышев: Куйбышевский авиационный институт, 1986. — 67 с.

В пособии излагаются основные сведения о технических и программных средствах машинной графики. Рассмотрены области применения машинной графики, используемые технические средства, виды преобразований графических данных. Общие положения иллюстрируются примерами программной реализации отдельных алгоритмов машинной графики.

Пособие предназначено для студентов, обучающихся по специальности «Прикладная математика»,  
Ил. 27. Табл. 3. Библиогр. — 15 назв.

Рецензенты: кафедра информационно - измерительных систем Куйбышевского ордена Трудового Красного Знамени политехнического института В. В. Куйбышева, канд. техн. наук доц. В. Б. Г у м е н н и к о в

# Введение

---

Машинная графика имеет дело с созданием и преобразованием графической информации. Машинная графика не имеет строго определенного содержания, и определяющим для нее является не предмет исследования, а метод. Сущность метода заключается в том, что геометрические объекты сопоставляются некоторым стандартным образом с информационными структурами данных, хранимыми в памяти ЭВМ. Структура графических данных является ограничением на взаимное расположение и свойства графических элементов изображения. Описание изображения, хранимое в памяти ЭВМ, соответствует не конкретному изображению, а целому классу изображений, элементы которых связаны определенными отношениями. Это позволяет получать различные изображения из одной структуры данных, задавая различные условия визуализации. Описание структур данных и условия визуализации могут сообщаться человеком в режиме диалога с ЭВМ или вычисляться машиной. Все это дает в руки человеку мощное средство создания и преобразования изображений, потенциальные возможности которого зависят только от его фантазии — чем она богаче, тем полнее раскрываются возможности машинной графики.

Обилие и разнообразие материала по машинной графике не позволяет охватить в рамках одного учебного пособия все вопросы в их полноте. Основная цель пособия состоит в том, чтобы дать исходные сведения по машинной графике для студентов младших курсов. Дальнейшее изучение более сложных вопросов машинной графики должно быть продолжено в рамках спецкурсов «Применение мо-

делирования в САПР», «Автоматизированные системы обработки изображений» и др.

Пособие состоит из четырех разделов. В первом даются общие сведения о машинной графике и областях ее применения. Во втором разделе описаны базисные средства, используемые при построении любой системы машинной графики. Здесь определяются стандартные системы координат и переходы от одной системы к другой, стандартные функции ввода/вывода графических данных; рассматриваются особенности построения и использования технических средств машинной графики. Третий раздел посвящен геометрическим преобразованиям графических данных. Особое внимание уделено использованию однородных координат, позволяющих заменить последовательность простых геометрических преобразований единым сложным преобразованием. В последнем разделе пособия рассмотрены основные позиционные задачи машинной графики, которые являются составными частями более сложных задач геометрического моделирования и получения реалистичных изображений.

В основу пособия положен курс лекций, читаемый автором студентам специальности 0647 «Прикладная математика».

Автор с благодарностью примет любые критические замечания и пожелания, касающиеся содержания и оформления этого пособия.

# I. Общие сведения о машинной графике

*Машинная графика* — новая область исследования, предмет ее — создание и преобразование графической информации. С помощью средств машинной графики (программных и технических) можно создавать графические объекты и манипулировать ими. Машинная графика тесно связана с двумя направлениями исследований, которые также имеют дело с графической информацией: обработка изображений и машинное зрение.

*Обработка изображений* имеет дело с проблемами, в которых как вход, так и выход являются картинками, например, система передачи изображений, в которой главными вопросами являются фильтрация помех и компактное представление изображений. Иногда необходимо более существенное преобразование, например, полутоновое изображение приводится к изображению с двумя уровнями. В другом случае мы можем создавать новое изображение из множества имеющихся изображений, например, воссоздавать трехмерную форму внутреннего органа человека из нескольких плоских изображений (проекций), полученных при зондировании этого органа рентгеновскими лучами с разных направлений.

*Машинное зрение* имеет дело с методами, дающими либо описание входной картинке, либо отнесение ее к одному из известных классов. Машинное зрение — обратная задача по отношению к машинной графике.

Связь трех направлений исследований, имеющих дело с обработкой графической информации, можно представить простым рисунком (рис. 1). Поскольку объект исследований у них общий (но разные цели и методы), естественно наличие общих задач для всех направлений. Такими являются задачи представления графических данных в вычислительной машине, структуры графических данных, хранение и поиск данных, сжатие данных и т. д.



Рис. 1

Менее очевидными являются другие проблемы, например, обработка изображений часто имеет дело с выделением контуров, в то время как машинная графика — с задачей заполнения контура. Так как одна задача противоположна другой, то неудивительно, что некоторые теоретические вопросы являются общими для обеих областей.

Полезность машинной графики объясняется двумя основными причинами.

1. В некоторых приложениях графическое представление информации не только более удобно, но и единственно разумно. Этот факт нашел выражение в пословице: «Одна картинка стоит тысячи слов». В число приложений машинной графики, где постоянно приходится решать графические задачи, входят, например:

подготовка программ для станков с ЧПУ (построение траекторий и их проверка);

получение фотошаблонов интегральных схем и печатных плат;

проектирование шоссеиных дорог;

конструирование автомобилей, кораблей, самолетов;

картография;

архитектура;

издательское дело (размещение материала на странице);

изготовление выкроек и лекал одежды (в швейной промышленности).

В этих и многих других областях совершенно неприемлемым оказывается словесное описание. Только графическое представление данных имеет смысл.

2. Машинная графика может быть ценным инструментом для отображения результатов моделирования различных физических, химических, биологических, а также абстрактных процессов, например:

инженер, проектирующий электронную схему с помощью ЭВМ, может получить график частотной зависимости для данной схемы;

врач, изучающий течение крови по артериям, получает рисунок (на основании расчетов), позволяющий выявить точки с высокой турбулентностью (именно эти точки чаще всего поражают атеросклерозом);

математик вводит уравнение и изучает влияние каждой переменной;

летчик практикуется на взлетах и посадках на моделируемом аэродроме, изображение которого на экране дисплея изменяется согласованно с «движением» управляемого летчиком «самолета», также моделируемого в ЭВМ.

Проектировщик робототехнического комплекса может изучить достоинства и недостатки модели робота, а пользователь обучаться работе с роботом — до изготовления или получения настоящего робота; разработчик программ для ЭВМ, используя специально подобранное преобразование данных, может по виду создаваемого программой графического образа судить о правильности или завершенности процесса вычислений. Это важно, если расчеты продолжительны, а результат заранее неизвестен.

Наиболее эффективно применение машинной графики в системах, где реализуется диалог человека и ЭВМ (интерактивный режим). Можно указать три основных задачи, в которых операции диалога выполняются естественно и просто средствами машинной графики: выбор; проектирование; моделирование процессов.

Типичными задачами выбора являются следующие.

Выбор альтернатив действий (использование графического «меню команд»). Список альтернатив в виде символов или условных графических обозначений воспроизводится на экране дисплея (рис. 2). Оператор, пользуясь световым пером, указывает на дальнейшее направление работы и тем самым запускает соответствующую программу.

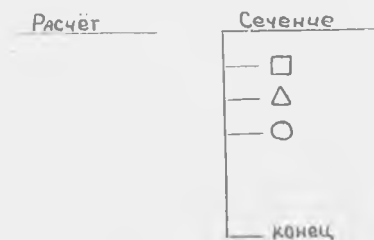


Рис. 2



Рис. 3

Выбор варианта при геометрических построениях. При конструировании геометрической формы оператор может указать световым пером один из вариантов геометрического решения, например, один из 4-х возможных способов проведения касательной к двум окружностям (рис. 3).

Выбор кривой наиболее подходящей формы. ЭВМ показывает проектировщику несколько кривых, а он выбирает ту из них, которая наиболее точно отвечает запросам с эстетической точки зрения (рис. 4). Такая задача возникает при конструировании внешних обводов автомобиля, корабля и самолета.

Выбор компонентов. Пользуясь световым пером, конструктор может указать на подходящую компоненту и тем самым быстро произвести выбор конкретной компоненты для ее использования в конструкции (рис. 5).

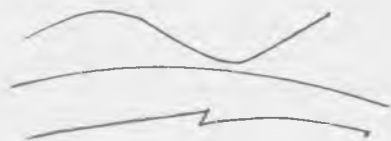


Рис. 4



Рис. 5

Укажем наиболее типичные задачи проектирования.

Подготовка графиков, схем, чертежей на графическом дисплее. Применяя графический «язык», оператор может намечать точки, соединять их отрезками прямых, опускать перпендикуляры, изображать углы и дуги окружностей и выполнять другие геометрические операции подготовки точного чертежа по предварительно выполненному эскизу. Хотя такие операции за пультом дисплея по стоимости не могут конкурировать с ручным вычерчиванием, они, тем не менее, имеют смысл, если чертеж впоследствии все равно должен быть переведен в цифровую («машинную») форму. Такие случаи неизбежны при изготовлении деталей на станках с числовым программным управлением (ЧПУ).

Построение трехмерных объектов из базовых элементов. Проектировщик, основываясь на своем опыте, может использовать геометрические формы базовых элементов, описание которых хранится в памяти ЭВМ. Например, он может сконструировать пролет моста из стоек и ферменных балок определенного типа (рис. 6). Как только такое построение заканчивается, геометрия структуры запоминается в математической форме, удобной для обработки. Этот метод может быть использован для синтеза трехмерных объектов путем объединения или вычитания шести базовых элементов-примитивов (рис. 7). Задаются тип элемента и значения его параметров. После этого осуществляется вызов и размещение элемента в требуемом месте трехмерного пространства. С помощью процедуры копи-



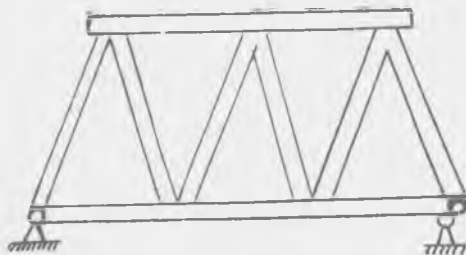


Рис. 6

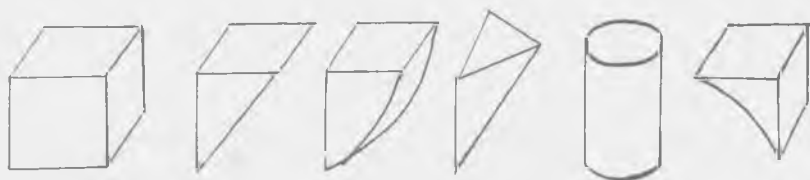


Рис. 7

рования базовые элементы или построенные из них изображения можно воспроизвести в других местах экрана. Процедура объединения дает возможность создавать новые элементы путем соединения элементов через поверхности. Полученные при построении объекты отображаются для пользователя в виде трех проекций или перспективного изображения.

**Размещение элементов.** С этой проблемой сталкиваются при проектировании печатных плат (задача компоновки), при раскрое материала в машиностроении, в швейной промышленности). Описание форм хранится в ЭВМ. Выбор и размещение элементов осуществляется в помощь человека-оператора.

**Описание связей.** Указываются элементы электронных, гидравлических или электрических систем, между которыми должны быть связи или соединения. ЭВМ воспринимает и запоминает указанные связи для последней обработки, например, при проектировании печатной платы на экране размещаются элементы схемы. Соединения между элементами схемы выполняются проводниками, изображаемыми линиями определенной толщины, расположенными параллельно осям координат (рис. 8). Выбор трассы осуществляется автоматически специальной программой или в режиме диалога.

Приведем примеры типовых задач моделирования процессов.

**Анализ движущихся деталей.** ЭВМ способна построить изображение движущихся деталей так, что проектировщик, глядя на это изображение, сможет уяснить вопросы точности зазоров, типов колебаний и другие сложные динамические факторы. Например, ЭВМ может показать в движении передаточный механизм шасси самолета (рис. 9). Когда движение конструкции воспроизводится в трехмерной проекции, проектировщик может легко проверить, не мешают ли некоторые детали друг другу. При этом он способен даже анализировать в динамике очень сложные ситуации, которые практически не смог бы себе представить, если бы изучал численные данные или статические чертежи.

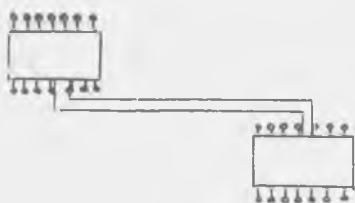


Рис. 8

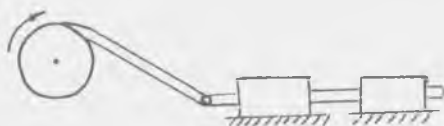


Рис. 9

В последнее время получили распространение системы геометрического моделирования действий роботов. В этих системах средствами машинной графики на экране дисплея воспроизводится рабочее место робота (вместе с ним самим). По командам, подаваемым с пульта ЭВМ и соответствующим различным действиям робота, координаты его частей в трехмерном пространстве пересчитываются, преобразуются в графическую форму, и пользователь видит меняющееся изображение робота, как если бы он наблюдал движение настоящего робота. Такое геометрическое моделирование робота позволяет проверить («отладить») управляющую программу, изучить характеристики робота, выбрать наиболее оптимальный режим работы, а также сбучить персонал без риска для их жизни или риска поломки робота.

**Обнаружение ошибок.** Средства машинной графики дают возможность увидеть и отметить ошибки, что уменьшает или вовсе устраняет необходимость операций контроля и число циклов исправлений. Например, оператор за пультом моделирует перемещение программно-управляемого инструмента (станок с ЧПУ), вырезающего из листа деталь. Ошибочный путь сразу замечается и может быть быстро исправлен (рис. 10).

Опыт использования машинной графики показывает, что ряд операций трудно или неудобно выполнять за графическим пультом или требует чрезмерно большого времени.

Некоторые трудности возникают из-за недостаточного совершенства технических и программных средств. Естественно, дальнейшее

развитие систем машинной графики должно улучшить это положение. Трудности иного характера возникают из-за того, что многие начальные фазы процесса трудно сформулировать или достаточно точно описать (например, при автоматизации процесса проектирования).

Назовем операции, которые трудно выполнять средствами машинной графики.

Изображение сложных пространственных объектов. Точное изображение на экране трехмерных деталей в их естественном виде является сложной задачей. В технике неизбежно приходится иметь дело с исключительно сложными узлами и сооружениями типа внутренних конструкций корабля или самолета с тысячами трубопроводов, каналов, перегородок, отверстий, проводов, заклепок и др. элементов оборудования. Такие сложные объекты со всеми их деталями в настоящее время еще не могут быть обработаны на ЭВМ. Известны два метода проектирования трехмерных объектов. Первый имитирует обычные чертежные приемы, подготавливая отдельные (плоские) проекции для воспроизведения их на экране. Затем световым пером указывается соответствие между линиями на этих проекциях. Второй заключается в формировании трехмерных объектов из некоторых базовых элементов путем их объединения и вычитания.

Каждый из методов имеет свои достоинства и недостатки. В любом случае для повышения естественности представления необходимо сложное программирование, а затраты на машинное время и необходимый объем памяти могут оказаться значительными. Когда приходится иметь дело со сложными узлами, любая из задач, касающихся определения линии пересечения двух поверхностей, определения невидимых линий, формирования перспективных проекций, создания тоновых изображений и т. п. сводится к солидной вычислительной работе.

Работа со сложными чертежами. Ввод сложных чертежей в ЭВМ занимает довольно много времени и требует

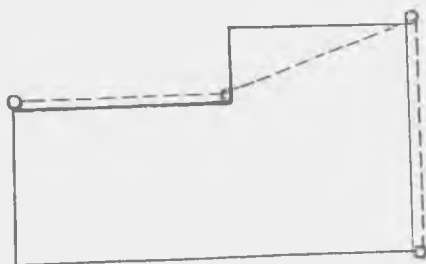


Рис. 10

высокой квалификации и чрезвычайной внимательности. Время ввода сложного чертежа в ЭВМ соизмеримо с временем ручного копирования чертежа на бумаге тушью. Такую работу целесообразно выполнять только в случае многократного использования чертежей ( типовые конструкции, на базе которых конструируются различные модификации).

Использование уже хранящихся в памяти ЭВМ сложных чертежей затруднительно, т. к. экран дисплея имеет ограниченные размеры и низкую разрешающую способность. При обычных методах работы с чертежами большого формата конструктор может переключать свое внимание на различные участки чертежа, сохраняя в то же время ощущение восприятия его в целом. Эта особенность работы может быть только частично реализована на дисплее, если использовать способность ЭВМ быстро изменять масштаб изображения и перемещать увеличенное изображение в желаемую точку экрана.

## 2. Средства машинной графики

---

### 2.1. КООРДИНАТНЫЕ СИСТЕМЫ

В машинной графике для получения изображений используются различные координатные системы и стандартная система преобразований этих координат. В общем случае объекты, изображения которых хотят получить, представляются в трехмерном мировом пространстве. Только часть этого пространства, попадающего в трехмерное окно, может быть преобразовано в видимое изображение.

В машинной графике наиболее часто используются два типа проекций:

*ортогональная* (или параллельная) — видимым считается все, что находится внутри трехмерного окна в форме параллелепипеда видимости (рис. 11,а);

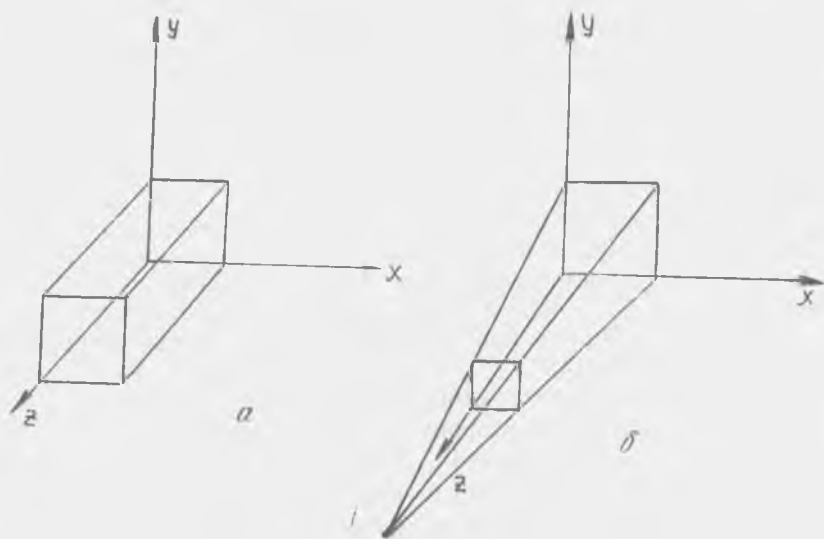


Рис. 11

центральная (перспективная) проекция, использующая пирамиду видимости, основание которой лежит в плоскости изображения, а вершина совпадает с точкой наблюдения (рис. 11,б).

Плоскость ( $XoY$ ), в которой формируется изображение, называется *картинной плоскостью*. Картинная плоскость служит плоскостью проецирования объектов из трехмерного мирового пространства, а также используется для задания окна отсечения по видимому объему.

Для задания окна на объекте или около него задается опорная точка. Начало системы координат картинной плоскости  $UV$  определяется проведением перпендикуляра из опорной точки на картинную плоскость. Расстояние до картинной плоскости может быть больше или меньше нуля, в зависимости от направления нормали картинной поверхности по отношению к мировой системе координат.

Ось  $V$  в картинной плоскости совпадает по направлению с проекцией вектора вертикали, задаваемого в мировой системе координат относительно опорной точки. Ось  $U$  проводится под  $90^\circ$  к оси  $V$  в направлении движения часовой стрелки, если смотреть по направлению нормали к картинной плоскости (т. е.  $U$ ,  $V$  и нормаль  $N$  образуют левостороннюю систему координат). Окно задается через минимальные и максимальные значения  $V$  и  $U$  относительно проекции опорной точки.

При заданном окне в картинной плоскости и типе проекции определяется видимый объем: для центральной проекции — полубесконечная пирамида, а для параллельной проекции — параллелепипед видимости. Оконное отсечение осуществляется по границам видимого объема, определяемого окном. Отсечение может быть не разрешено — тогда окно служит только для того, чтобы задать отображение из мировой системы координат в нормализованные координаты устройства (логическую картинную поверхность). Могут быть заданы передняя и задняя секущие плоскости, параллельные картинной поверхности. В этом случае осуществляется глубинное отсечение. Секущие плоскости определяются расстоянием вдоль нормали от опорной точки.

Условия умолчания при задании окна и системы координат картинной плоскости:

картинная плоскость перпендикулярна оси  $Z$ ;

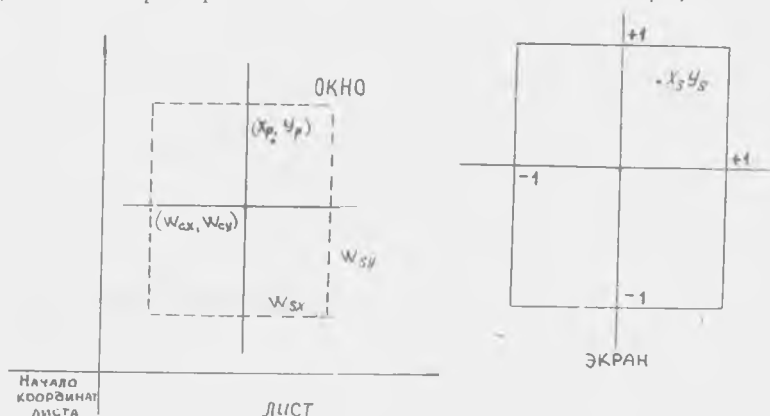
ось  $Z$  направлена в сторону от наблюдателя;

опорная точка находится в картинной плоскости в начале координат (т. е. расстояние до картинной плоскости равно 0);

вектор нормали картинной плоскости равен  $(0, 0, -1)$ ;

проецирование — параллельное, вектор направления проецирования равен  $(0, 0, 1)$ .

Таким образом, на первом этапе построения изображений выполняется отсечение по видимому объему. Видимый объем определяется окном, задаваемым в плоскости изображения и центром наблюдения (или направляющим вектором наблюдения). На следующем этапе трехмерные координаты точек объекта, попавшие в видимый объем, преобразуются в двумерные координаты листа (проецирование в окно на плоскости изображения). При ортогональном проецировании эта операция состоит просто в отбрасывании  $z$ -координат точек. При центральном проецировании преобразование более сложно, поэтому, несмотря на большую реалистичность получаемых изображений, используется реже, чем ортогональное. На третьем этапе производится преобразование из окна в рабочую область визуализации. Это преобразование называется также *кадрированием*.



Р и с. 12

Пусть центр окна (рис. 12) имеет координаты  $(w_{cx}, w_{cy})$  в системе координат картинной плоскости (будем для краткости называть ее листом). Размер окна, измеряемый от центра, равен  $(w_{sx}, w_{sy})$ . Тогда координаты точки  $(x_p, y_p)$  листа преобразуются в нормализованные координаты  $(x_N, y_N)$ :

$$x_N = \frac{x_p - w_{cx}}{w_{sx}}, \quad y_N = \frac{y_p - w_{cy}}{w_{sy}} \quad (1)$$

Эти координаты являются безразмерными величинами и изменяются в пределах  $-1 \dots +1$ . Это позволяет решать графические задачи без учета конкретных единиц измерения, зависящих от выбранных приложений. Программы, оперирующие нормализованными координатами, универсальны и легко переносятся. Часто в связи с этим картинную поверхность с нормализован-

ными координатами называют логической картинной поверхностью.

Последний этап преобразований координат связан с переходом от нормализованных координат к физическим координатам устройства, используемого для визуализации. Такое преобразование называется *преобразованием рабочего места* и связано с тем, что различные устройства вывода изображений имеют различные параметры, которые учитываются только при визуализации, а основная часть графической системы может быть выполнена независимой от физического устройства. Поэтому, если, например, информация выводится на экран дисплея в поле индикации с координатами центра ( $V_{cx}, V_{cy}$ ) и размерами ( $V_{sx}, V_{sy}$ ), то нормализованные координаты ( $x_N, y_N$ ) переходят в

$$x_s = x_N V_{sx} + V_{cx} \quad (2)$$

$$y_s = y_N V_{sy} + V_{cy}.$$

Последние два преобразования иногда совмещаются в одно преобразование (рис. 13):

$$x_s = \frac{x_p - w_{cx}}{w_{sx}} V_{sx} + V_{cx} \quad (3)$$

$$y_s = \frac{y_p - w_{cy}}{w_{sy}} V_{sy} + V_{cy}.$$

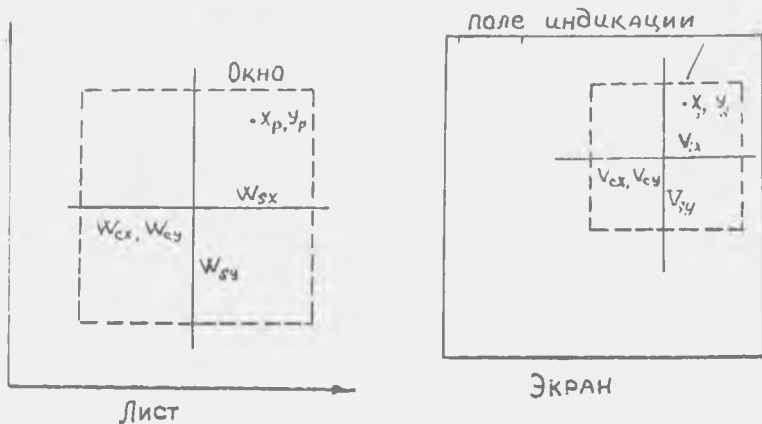


Рис. 13

## ВОПРОСЫ И УПРАЖНЕНИЯ

1. Изображение с листа чертежа, попадающее в окно, преобразуется в изображение на экране в прямоугольном поле индикации. Преобразуемая фигура — треугольник. Ширина



окна 200 мм, высота окна — 300 мм, координаты левого нижнего угла окна на поле чертежа — (400 мм, 300 мм). Ширина поля индикации — 400 единиц раstra, высота — 900 ед. раstra, координаты левого нижнего угла поля индикации — (100, 100). Для заданных координат вершин  $A$ ,  $B$ ,  $C$  треугольника на листе найти соответствующие координаты в поле индикации:

а)  $A = (440, 350)$ ,  $B = (490, 500)$ ,  $C = (540, 550)$

б)  $A = (440, 400)$ ,  $B = (490, 550)$ ,  $C = (540, 350)$

в)  $A = (440, 450)$ ,  $B = (440, 550)$ ,  $C = (490, 400)$

г)  $A = (490, 400)$ ,  $B = (540, 500)$ ,  $C = (590, 450)$

д)  $A = (490, 500)$ ,  $B = (540, 550)$ ,  $C = (590, 400)$

е)  $A = (490, 550)$ ,  $B = (540, 350)$ ,  $C = (590, 500)$

2. Напишите уравнения, для преобразования кадрирования, если заданы координаты левого нижнего угла окна на листе и левого нижнего угла поля индикации на экране, а также размеры сторон окна и поля индикации.

## 2.2. ОТСЕЧЕНИЕ

Если изображение на экране подвергается геометрическим преобразованиям, например повороту, то возможен выход части изображения за пределы экрана (рис. 14). В ранних устройствах для задания изображения резервировали пространство, намного большее площади экрана. При выводе изображения луч включался (или опускалось перо графопостроителя), если линия находилась в пределах экрана, и выключался (перо поднималось), если линия выходила за пределы экрана (рис. 15). Недостатком такого способа является непродуктивно затрачиваемое время на прочерчивание невидимых частей изображения (с нулевой яркостью). Кроме того, на хранение частей изображения расходуется память дисплея.

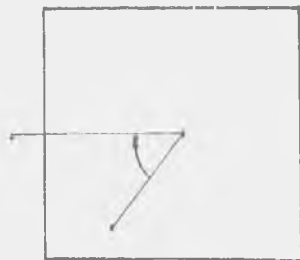


Рис. 14

В идеальном случае информация, подаваемая на дисплей, должна содержать лишь то, что может быть отображено на экране. Эта задача решается с помощью алгоритма отсечения и не так проста, как это кажется с первого взгляда, даже если изображение состоит только из отрезков прямых линий. Например, отрезок может быть частично видим, даже когда обе

концевые точки находятся за пределами экрана. На рис. 16 показан ряд различных положений, которые может занимать отрезок относительно экрана.

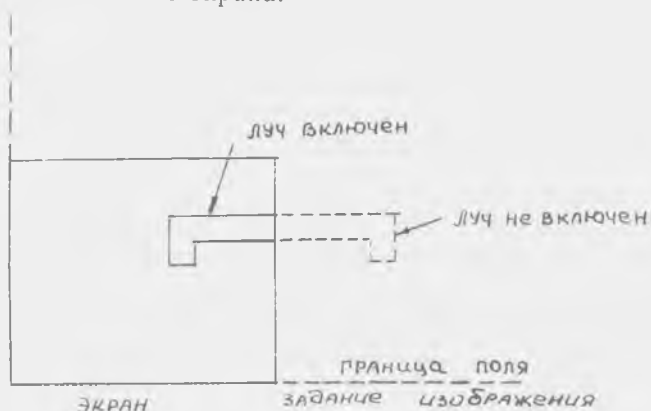


Рис. 15

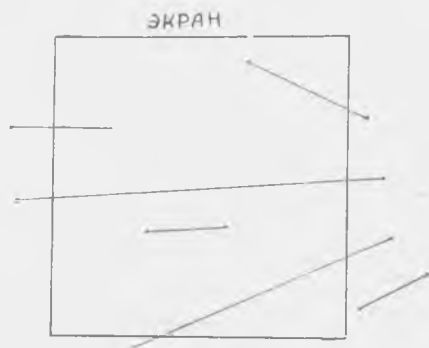


Рис. 16

Следует отметить, что поскольку экран имеет форму выпуклого многоугольника, то при отсечении отрезка не может появиться более одного видимого участка. Для быстрого поиска конечных точек видимой части отрезка и отбрасывания частей изображения, значительно выходящих за пределы экрана, Д. Коэном и А. Сазерлендом был предложен специальный алгоритм отсечения.

Работа алгоритма основана на том, что каждый отрезок либо полностью лежит в пределах экрана, либо его можно разделить так, чтобы одна из частей могла быть отброшена простейшим образом. Проверка на отбрасывание производится путем продолжения границ экрана так, чтобы они разделили поверхность, на которой находится полное изображение, на девять областей (рис. 17):

Каждой из этих областей присвоен четырехразрядный двоичный код. В каждом из четырех разрядов записывается единица, если выполняется следующее условие, и ноль, если оно не выполняется:

1-й разряд: точка над верхним краем экрана,

2-й разряд: точка под нижним краем экрана,

3-й разряд: точка справа от правого края экрана,

4-й разряд: точка слева от левого края экрана.

Две конечные точки отрезка получают коды  $A_1B_1C_1D_1$  и  $A_2B_2C_2D_2$ , соответствующие областям, в которых они находятся. Если коды для обоих концов отрезка равны нулю, то отрезок целиком лежит в поле экрана, в противном случае, если логическое пересечение кодов для концов отрезка

$$P = A_1A_2 \vee B_1B_2 \vee C_1C_2 \vee D_1D_2$$

не равно нулю, то отрезок полностью лежит вне экрана и может быть отброшен. Если отрезок нельзя отбросить при одной из этих проверок, то его следует разделить на части одним из 2-х способов:

находится точка пересечения отрезка с одним из краев экрана и часть, лежащая вне экрана, отбрасывается;

отрезок делится средней точкой пополам, что позволяет обойтись без вычисления точек пересечения отрезка с краями экрана.

Алгоритм отсечения позволяет устанавливать границы области отсечения в любом месте экрана — это дает возможность высвечивать изображение в прямоугольнике, размеры которого меньше экрана (в поле индикации), так что остается свободная площадь для вывода сообщений, команд и т. д. На экране может быть несколько полей индикации, например, для вывода различных проекций детали.

## 2.3. ТЕХНИЧЕСКИЕ СРЕДСТВА МАШИННОЙ ГРАФИКИ

### 2.3.1. СТРУКТУРА И ФУНКЦИИ ОБОБЩЕННОЙ СИСТЕМЫ МАШИННОЙ ГРАФИКИ

При построении любой системы машинной графики (МГ) должны учитываться следующие требования:

основная структура системы МГ должна минимально зависеть от конкретных физических устройств ввода/вывода графической информации;

1001	1000	1010
0001	ЭКРАН 0000	0010
0101	0100	0110

Рис. 17

функции ввода и вывода графических данных должны быть разделены;

функции построения описания объекта и формирования его изображения (визуализация) должны быть максимально разделены.

Прикладная программа создает описания объектов, которые должны быть изображены, в мировой системе координат. Операции визуализации, часто включающие также операцию отсечения, осуществляют переход от нормализованных координат к физическим координатам устройств вывода. Результирующая информация, подготовленная к выводу на устройство, называется *дисплейным файлом*.

Часто переходы от конкретных устройств ввода, а также вывод из дисплейного файла на конкретное физическое устройство вывода выполняются таким образом, что их фактическая работа скрыта от программиста. Программист работает с логическими устройствами ввода/вывода, не заботясь о фактическом выполнении функции ввода/вывода на конкретном оборудовании. Такая независимость программ от конкретных устройств достигается путем использования стандартных функций ввода и вывода, которые могут быть реализованы на различных имеющихся физических устройствах.

В стандартах машинной графики определено 6 выходных примитивов. Выходные примитивы являются абстрактными функциями, которые может выполнять устройство вывода:

*ломаная* — последовательность отрезков прямых, соединяющих упорядоченный набор точек (задается как параметр);

*полимаркер* — последовательность точек, помеченных маркерами одного типа;

*текст* — строка символов;

*заполненная область* — область внутри многоугольника, «залитая» определенным цветом или текстурой;

*поле* — прямоугольный массив точек, каждая из которых определенного цвета или яркости;

*сложный (обобщенный) примитив* (произвольная линия) — прочие примитивы, которые могут вычерчиваться (высвечиваться) устройством (например, окружности, дуги, эллипсы, сложные кривые и т. д.). Здесь устанавливается стандартный способ обращения к таким примитивам.

Каждый примитив описывается через его тип и атрибуты. Для визуализации примитивов должны быть заданы значения их атрибутов. Например, для визуализации ломаной задаются атрибуты типа линии (сплошная, короткий штрих, длинный штрих, штрихпунктирная и т. д.), ширина линии, цвет. Для визуализации маркера указывается тип (· + — × и т. д.), раз-

мер, цвет. Для визуализации текста должны быть указаны шрифт (стандартный, латинский, курсив, греческий, кириллица, арабский), качество начертания (низкое, среднее, высокое), цвет и т. д. Для заполненной области должен указываться тип заполнения (пусто, сплошное, по образцу, штриховка и т. д.), цвет и т. д.

Примитивы могут группироваться в *сегменты*, описывающие части изображения. При различных преобразованиях изображений сегмент рассматривается как целое. Сегменты могут включаться в состав других сегментов. При создании изображения прикладная программа открывает сегмент, вызывает примитивы вывода с конкретными значениями атрибутов, а затем закрывает сегмент. Существует два типа сегментов: постоянные и временные. *Постоянные* сегменты имеют имена, которые можно использовать для указания той части картинка, которую нужно изменить. *Временные* сегменты используются один раз в момент создания изображения и в дальнейшем не сохраняются.

Постоянные сегменты описываются своими атрибутами, которые могут динамически изменяться, поэтому они называются *динамическими атрибутами*. Примерами динамических атрибутов являются условия видимости, освещенности, чувствительности к световому перу и т. д. В отличие от атрибутов примитивов, значения динамических атрибутов постоянного сегмента могут изменяться программой после создания сегмента.

Для ввода графической информации определено 5 логических типов (или функций ввода):

*указатель* (селектор) — указание примитива вывода по имени сегмента или непосредственного указания элемента изображения (примитива);

*кнопка* (выбор) — выбор одной из нескольких возможных альтернатив;

*искатель* (локатор) — для указания положения (позиции) точки на картинной плоскости;

*ввод значений* (валюатор) — ввод числовых значений (констант, вещественных чисел);

*клавиатура* — ввод символьных строк.

Перед использованием устройство должно быть инициализировано и разрешено. При вводе графической информации может быть использован один из трех режимов:

по запросам пользователя (программы); подобно оператору READ в фортране система ждет ответа пользователя в виде ввода новой информации;

циклический опрос — система контролирует («следит») текущее вводимое значение, например, положение курсора на экране;

диалоговое прерывание.

Реализация виртуальных (логических) устройств ввода может быть выполнена различными техническими средствами. При этом в конкретном устройстве ввода могут быть реализованы не все возможности. Предусмотрено 9 уровней функциональных возможностей (табл. 1).

Таблица 1

Возможные структуры данных	Реализация режимов ввода		
	Не используются	Только по запросу из программы пользователя	Все режимы
Только примитивы 0	0 а	0 в	0 с
Простые сегменты 1	1 а	1 в	1 с
Составные сегменты 2	2 а	2 в	2 с

Примечание. Следует отметить, что поскольку на нулевом уровне не определено понятие сегмента, то указатель на уровне 0 в и 0 с не реализуем.

### 2.3.2. ОБЩИЕ СВЕДЕНИЯ О ТЕХНИЧЕСКИХ СРЕДСТВАХ МАШИННОЙ ГРАФИКИ

Технические средства машинной графики обычно объединяются в аппаратно-программный комплекс, называемый *графической интерактивной системой*. Характерная особенность таких комплексов — наличие довольно мощного логического ядра, комплекс в этом случае является средством решения задач, требующих большого объема вычислений, а также централизованным хранилищем информации. Такие комплексы предназначены для решения различных практических задач и у нас в стране получили название *автоматизированных рабочих мест (АРМ)*. Сейчас выпускается несколько моделей АРМ, проблемно-ориентированных на решение следующих задач:

проектирование радиоэлектронной и микроэлектронной аппаратуры (АРМ-Р);

проектирование механических и оптико-механических узлов и конструкций (АРМ-М);

картографирование (АРМ-К);

технологическая подготовка производства (АРМ-Т);

проектирование программ и программной документации (АРМ-Т).

Независимо от модели АРМ общая структура может быть представлена в виде схемы (рис. 18). АРМ строится в виде центрального вычислительного комплекса и набора выносных терминалов различного типа, связанных с центральным процессором комплекса по специальному каналу. В составе комплекса могут иметься устройства для изготовления чертежей, фотошаблонов, масок интегральных микросхем, вывода текстовых документов большого объема и ведения банка данных.

Терминальные станции имеют в своем составе дисплей различных типов и кодировщики, а также некоторые другие устройства. Управление этими устройствами и автономное решение задач на терминальных станциях осуществляется входящими в их состав микро- и мини-ЭВМ. Терминальная станция может иметь различное исполнение. Минимальная конфигурация терминала — рабочий стол с размещенным на нем графическим или алфавитно-цифровым дисплеем.

Обязательными устройствами графических комплексов являются средства графического взаимодействия пользователя и ЭВМ: графические дисплеи, графопостроители (координатографы), устройства полуавтоматического кодирования графической информации (устройства типа «сколки»). Различают два типа устройств МГ — *растровые* и *векторные*. В векторных устройствах графические примитивы кодируются командами, в которых используются координаты конечных векторов. Перо графопостроителя или электронный луч в графическом дисплее перемещается от одной конечной точки к другой. Такие системы также называются системами с *произвольным сканированием* (штриховыми). В растровых устройствах электронный луч или пишущий элемент смещается по строкам (или столбцам) всегда по одной траектории, не зависящей от графических данных.

Основное различие между векторным и растровым дисплеем состоит в организации памяти. В векторном дисплее точки, составляющие каждый примитив, хранятся в памяти последовательно и рисуются в том же порядке, примитив за примитивом. В растровом дисплее память организована в виде двумерного массива элементов. Каждый элемент хранит значение яркости и/или цвета соответствующей точки изображения. Между элементами памяти и изображения имеется взаимно однозначное соответствие. Точка изображения называется часто *пиксель* (от pixel-picture elements — элемент картины). Изображение формируется путем просмотра памяти по строкам (или столбцам), поэтому соседние на картинке элементы изображения не обязательно выводятся друг за другом.

Основной параметр графических устройств — *разрешающая способность*, т. е. число различных элементов в единице длины.





Так, для обычного алфавитно-цифрового печатающего устройства разрешающая способность равна:

4 точки / см по горизонтали

2,4 точки / см по вертикали

Разрешающая способность дисплея зависит от размера светового пятна и обычно составляет 0,25 мм (40 линий/мм).

В дисплее изображение формируется обычно на электронно-лучевой трубке (ЭЛТ). Для поддержания на экране устойчивого изображения необходимо обновлять его, т. е. регенерировать не менее 30 раз/с (светимость люминофора ЭЛТ быстро уменьшается и не может продолжаться более 10...60 мкс).

### **2.3.3. УСТРОЙСТВА ВВОДА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В ЭВМ**

Полуавтоматические устройства графического ввода информации (УГВ) позволяют в десятки раз увеличить производительность труда и повысить достоверность кодируемой информации. Кодирование с помощью полуавтоматических устройств осуществляет человек-оператор, который перемещая регистрирующий орган, обходит последовательно все графические элементы (точки, отрезки прямых, дуги окружностей и т. д.). Величина перемещения регистрирующего органа фиксируется и обрабатывается электронным устройством — преобразуется в цифровой код и заносится на носитель данных (перфоленду, перфокарты, магнитную ленту и т. д.).

Таким образом, при полуавтоматическом вводе графической информации в ЭВМ процесс распадается на два этапа: распознавание типа графических элементов во вводимом изображении (поиск и вычленение графических элементов определенного типа на изображении) и определение положения графических элементов на чертеже. Первая часть процесса ввода выполняется человеком. Тип графических элементов вводится в ЭВМ с помощью специальной клавиатуры или наборного поля. Считывание и кодирование положения графического элемента, на который специальным датчиком указывает человек-оператор, осуществляется кодировщиком. Существуют различные устройства полуавтоматического ввода графической информации, различающиеся по принципу действия, методам считывания и регистрации графической информации, степени автоматизации процесса считывания и т. д.

По методу считывания различают устройства непрерывного, дискретного и непрерывно-дискретного действия. УГВ непрерывного действия обеспечивает считывание координат точек, произвольно расположенных в плоскости изображения. Они

отличаются простотой конструкции, но имеют невысокую точность считывания и предназначены для графической информации нерегулярной структуры. С увеличением размеров изображения возрастает и погрешность отсчета координат, поэтому при жестких требованиях к точности считывания и больших размерах изображения используют устройства дискретного действия, рассчитанные на регулярную структуру изображения. В этом случае отсчет координат производится с заданным шагом в строго фиксированных точках рабочего поля (в узлах дискретной сетки), формат которого может достигать размеров  $1000 \times 1000$  мм при разрешающей способности 0,5...1 мм. Для увеличения разрешающей способности устройств применяется совмещение дискретного и непрерывного метода отсчета. При этом грубый отсчет производится с достаточно большим шагом дискретизации (2,5...5 мм), а координата точки в пределах дискрета определяется непрерывным методом.

В зависимости от способа отсчета текущих координат, УГВ подразделяются на электромеханические, оптико-механические, магнитные, контактные, звуковые, емкостные и др. Электромеханические устройства основаны на применении шаговых двигателей для перемещения координатной системы до совмещения искомой точки с неподвижным визиром отсчета. Основное достоинство устройств этого типа — простота системы управления и высокая надежность работы. К недостаткам нужно отнести низкую скорость считывания, сложность механических узлов устройства, значительные размеры и большую массу.

К оптико-механическому типу относятся устройства с оптическими фотодатчиками, жестко связанными с механической системой отсчета координат, которая состоит из двух неподвижных взаимно перпендикулярных зубчатых реек с перемещающейся зубчатой передачей, вращающей диск с определенным количеством прорезей. С одной стороны диска установлены два фотодиода, с другой — два светодиода. При вращении диска производится подсчет импульсов, возникающих на фотодиодах при возбуждении их потоком света от светодиодов, и одновременное определение направления движения логической схемой. Устройства этого типа удобны в эксплуатации, надежны и просты по конструкции.

Звуковые устройства основаны на определении координат путем измерения времени распространения звуковой волны от источника, которым является перемещаемый датчик, до специальных микрофонов, установленных на краях планшета. По такому же принципу работают и ультразвуковые устройства. Преимущество устройств этого типа — простота конструкции, удобство в эксплуатации, низкая стоимость. К недостаткам мож-

но отнести невысокую точность, низкую помехоустойчивость, чувствительность к климатическим изменениям.

В потенциометрических устройствах для определения координат используется метод моделирования потенциальных функций простого вида. Обычно устройства представляют собой планшет из проводящего материала с равномерной проводимостью, к сторонам которого поочередно подключается высокостабильное напряжение. Если теперь установить специальный датчик в любой точке планшета, то он окажется под напряжением, пропорциональным соответствующим координатам. Достоинством этих устройств является простота конструкции и эксплуатации, недостатками — низкая разрешающая способность, зависимость от внешних воздействий (температуры, влажности), необходимость при считывании прокалывать носитель изображения для обеспечения контакта датчика с планшетом.

Наиболее широкое распространение получили УГВ магнитного (электромагнитного) типа, в которых для определения координат используется индуктивная связь датчика со специальными координатными шинами (рис. 19), представляющими собой

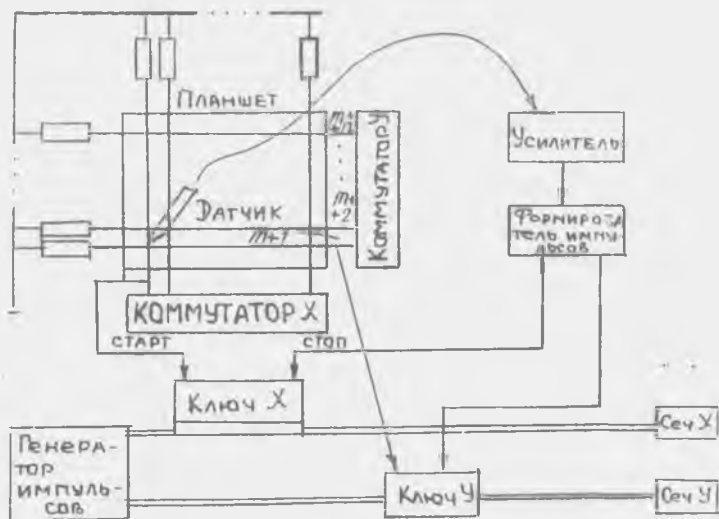


Рис. 19

тонкие проводники, уложенные на поверхности планшета (число проводников соответствует числу двоичных разрядов максимальной координаты). Каждый проводник из множества координатных шин можно рассматривать как первичную обмотку трансформатора, а датчик — как вторичную обмотку трансформатора.

Индуктивная связь между ними невелика, так что если через координатные шины проходит сигнал, то датчик «чувствует» сигнал только от ближайшей к нему шины. Сигнал на координатные шины подают поочередно, начиная с крайнего левого проводника, во все вертикальные шины, затем в самый нижний из горизонтальных проводников и по всем горизонтальным шинам. Одновременно с подачей сигнала в шину 1 открывается ключ  $X$ , и начинается подсчет импульсов в счетчике  $X$ . Аналогично одновременно с подачей сигнала в шину  $m + 1$  (самая нижняя шина по координате  $Y$ ) открывается ключ  $Y$  и начинается подсчет импульсов в счетчике  $Y$ . Окончание подсчета импульсов происходит в момент прихода сигнала с датчика. Поскольку интервалы времени между началом отсчета и приходом сигнала с датчика пропорциональны расстояниям от левого нижнего угла планшета по координатным осям  $X$  и  $Y$  до точки, в которой находится датчик, то результат подсчета импульсов счетчиками  $X$  и  $Y$  даст значения координат точки (с точностью до постоянного множителя — коэффициента преобразования системы). Достоинства УГВ магнитного типа — удобство в работе, высокая надежность и помехоустойчивость. Разрешающая способность у лучших конструкций достигает 0,1 мм, у худших ЭГВ такого типа 0,5 мм. Размеры поля кодирования обычно  $700 \times 1000$  мм или около этой величины.

### 2.3.4. ГРАФОПОСТРОИТЕЛИ

Важным средством машинной графики является графопостроитель, предназначенный для вывода графических данных на бумагу. Графопостроитель дает возможность получить чертежи, графики, схемы, диаграммы, контрольные прорисовки схем печатного монтажа и т. д. Выпускаются графопостроители двух типов — рулонные и планшетные (рис. 20). В рулонном

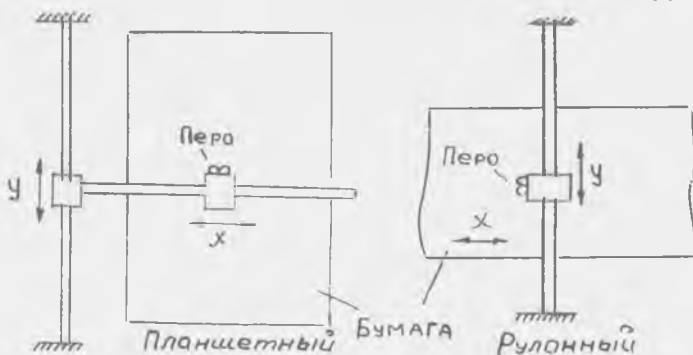


Рис. 20

графопостроителе движение пера относительно бумаги складывается из двух движений — движения бумаги (по координате  $X$ ) и движения каретки с пером по неподвижно закрепленной направляющей (координата  $Y$ ). В планшетном графопостроителе бумага неподвижно закрепляется на планшете, а вычерчивание реализуется за счет движения каретки с пером (координата  $X$ ) по направляющей, перемещающейся по второй, неподвижно закрепленной направляющей.

На рис. 21 приведена структурная схема управления меха-

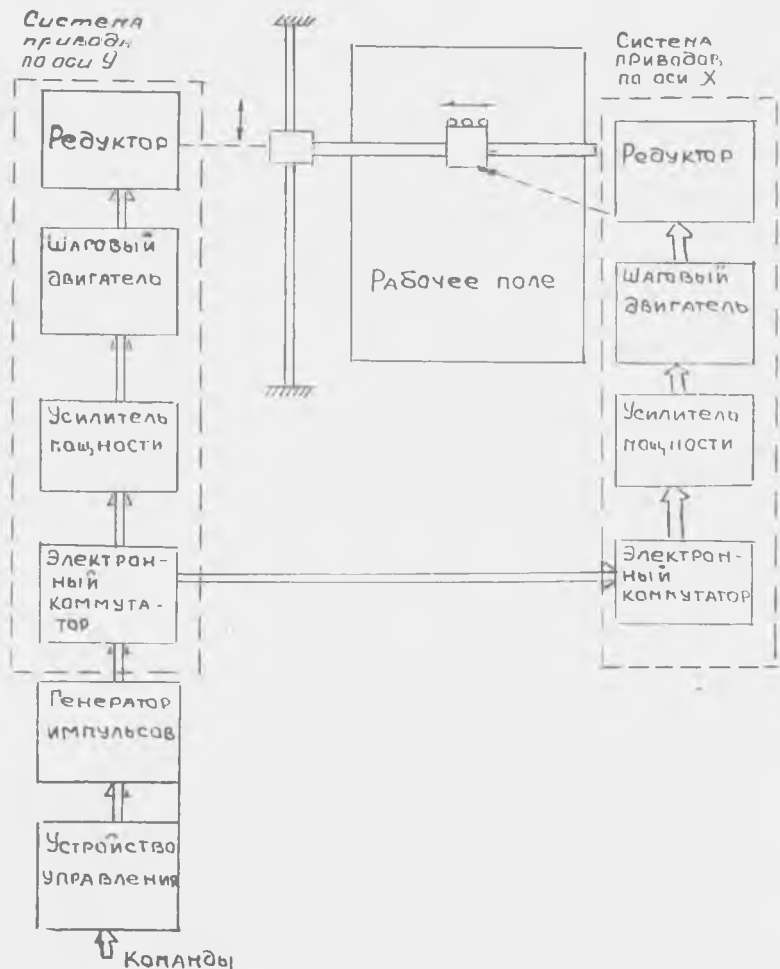


Рис. 21

пизмами графопостроителя. Команды на вычерчивание графических элементов расшифровываются в устройстве управления и преобразуются в соответствующую серию импульсов. Эти импульсы усиливаются и поступают на шаговый двигатель (по каждой координате перемещения), который вращает редуктор и вызывает соответствующее перемещение пера. Особенностью графопостроителя является дискретность перемещения пера и ограниченное число направлений перемещений. В большинстве современных графопостроителей имеется только восемь разрешенных направлений (рис. 22), они образуются путем комбинации направлений единичных шагов по координатным осям. Например, движение вправо — вверх осуществляется одновременным перемещением пера на один шаг вправо ( $+X$ ) и на один шаг вверх ( $+Y$ ). Величина шага перемещения в современных графопостроителях  $0,0125...0,1$  мм, поэтому вычерчиваемые линии для человека кажутся непрерывными. Дискретность перемещения обычно заметна только на

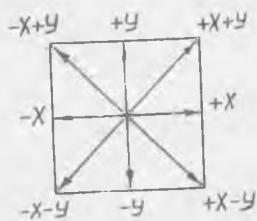


Рис. 22

линиях, расположенных под небольшими углами к вертикальной и горизонтальной осям координат.

Данные для вычерчивания на графопостроителе должны быть соответствующим образом подготовлены. Возможны 3 уровня подготовки данных: для работы в инкрементальном режиме; в кодах графопостроителя; для работы на уровне графических программ.

В инкрементальном режиме устройство управления дает команды перемещений на один элементарный шаг. Вся работа по вычислению направлений элементарных шагов для конкретной картинки полностью лежит на пользователе. Поскольку это очень трудоемкая работа, инкрементальный режим используется редко, несмотря на то, что в таком режиме может быть вычерчена практически любая фигура.

Использование кодов графопостроителя облегчает подготовку данных для вычерчивания. Так, например, отрезок прямой линии может быть вычерчен, если указать код линейной интерполяции и задать координаты начальной и конечной точек отрезка. Также по кодам вычерчиваются символы букв или цифр. К недостаткам режима работы в кодах графопостроителя следует отнести все еще большой объем работы по подготовке исходных данных для вычерчивания. В настоящее время этот режим используется редко.

Наиболее распространенным методом работы с графопостроителем сейчас является использование специального пакета графических подпрограмм. Обычно различают 3 вида графических подпрограмм: базисные, функциональные и прикладные. Базисные подпрограммы обеспечивают вычерчивание основных графических элементов (графических примитивов) — отрезков, дуг окружностей, символов, осей, прямоугольников, правильных многоугольников и т. п. Функциональные подпрограммы предназначены для вычерчивания элементов, ориентированных на определенные области науки и техники. Например, экономические графические подпрограммы дают возможность вычертить графики, диаграммы, использовать календарную разметку осей и т. д. Конечно, такое деление довольно условно и отражает только определенные тенденции в разработке графических подпрограмм. Прикладные программы предназначены для решения конкретных графических задач в узкой области, например, для вычерчивания компрессорных лопаток авиационных двигателей. Как правило, прикладные графические программы опираются на какой-либо пакет универсальных графических программ. Так, одним из наиболее широко используемых пакетов графических программ является ГРАФОР.

В последнее время все чаще начинают использовать графопостроители растрового типа, среди которых наиболее распространенными являются электростатические графопостроители (рис. 23,а). На те части бумаги, которые должны быть темными,

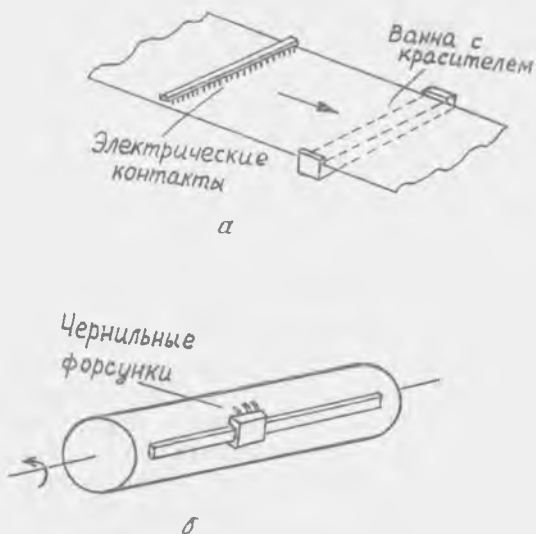


Рис. 23

наносится электрический заряд. Затем над бумагой распыляется противоположно заряженный черный краситель. Частицы красителя оседают на бумагу в тех местах, где был нанесен заряд. Заряд наносится на бумагу (ширина до 180 см) построчно. Плотность электрических контактов 40...100 контактов на см. Таким образом получается 40...100 строк/см. Каждое пятно в изображении будет либо черным, либо белым. Однако можно получить ограниченный набор уровней тона, определив более крупные конфигурации, состоящие из нескольких пятен.

Электростатический графопостроитель обычно в 10...20 раз быстрее векторного (перьевого), однако он требует предварительной развертки изображения подготовки раstra). Изображения на электростатическом ГП имеют меньшую контрастность, поскольку небольшое количество красителя прилипает и к незаряженным участкам бумаги.

Другой, часто использующийся тип растрового графопостроителя — струйно-чернильный, в котором вычерчивание осуществляется чернильной струей. Бумага закрепляется на вращающемся барабане, вдоль которого по направляющей медленно перемещаются чернильные форсунки (рис. 23,б).

### 2.3.5. ГРАФИЧЕСКИЙ ДИСПЛЕЙ

Наиболее интересным устройством машинной графики является графический дисплей. Обобщенная структурная схема графического дисплея векторного типа показана на рис. 24. Информация от внешней ЭВМ через блок сопряжения поступает в устройство управления дисплея. Здесь она разделяется на графические данные, которые поступают в оперативное запоминающее устройство (ОЗУ) дисплея, и управляющую информацию (команды), поступающую затем в распределитель. Устройство управления поочередно считывает элементы графических данных из ОЗУ и направляет их на знакогенераторы. Применение знакогенераторов упрощает отображение графических данных, однако ведет к усложнению аппаратуры управления. Наиболее часто применяются генераторы точек, символов (букв, цифр, специальных знаков), отрезков, окружностей. При этом каждому знаку может соответствовать своя яркость (или цвет) изображения. Кроме того каждый графический элемент должен быть высвечен в определенном, месте экрана, поэтому графические данные разделяются на координатные (они определяют положение конкретного элемента на экране), данные о яркости или цвете и данные о типе графического элемента. Эта информация преобразуется в аналоговые физические величины напряжения (или токи), которые управляют пучком электронов в ЭЛТ.



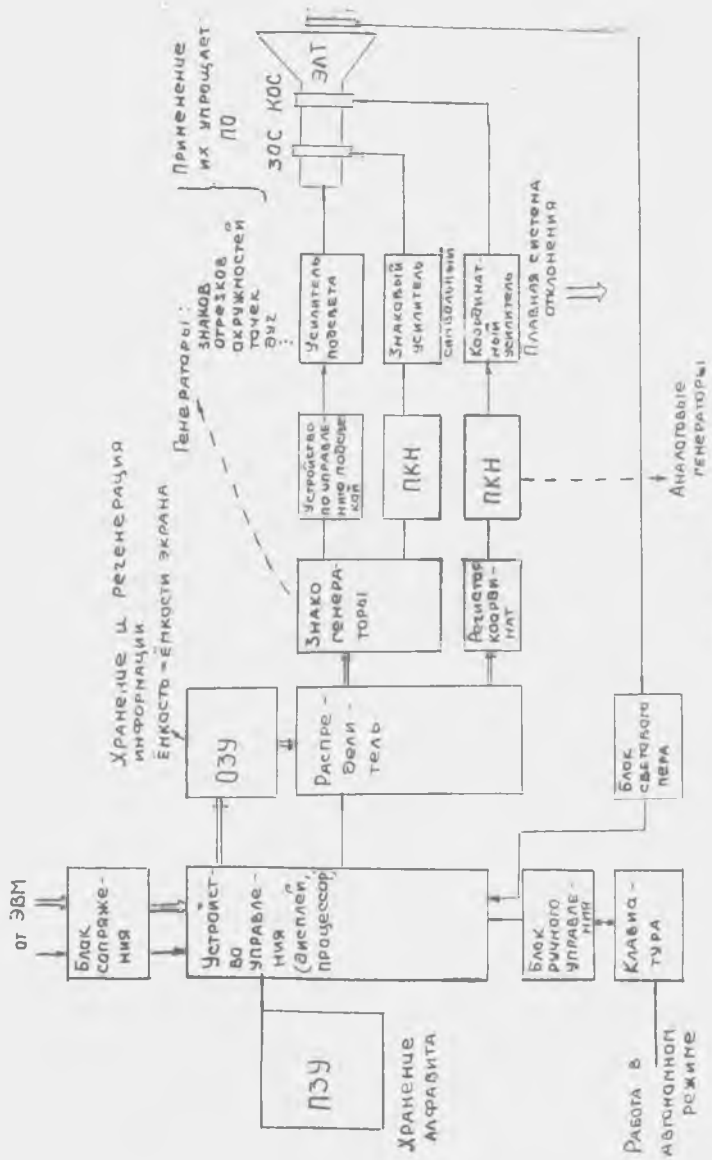


Рис. 24

Для оперативного обмена информацией между ЭВМ и человеком-оператором в состав дисплея включаются клавиатура и световое перо, которые позволяют вводить в ЭВМ и выводить из памяти ЭВМ нужную информацию, корректировать вызванную информацию последующей посылкой ее в ЭВМ для дальнейшей обработки. Клавиатура дисплея содержит алфавитно-цифровые клавиши (как на обычной пишущей машинке) и функциональные клавиши, которые соответствуют кодам операций режимов работы.

Развитие графических дисплеев идет по пути все большей автоматизации, большей автономности. В этих целях в состав дисплея сейчас обычно встраивается свой специализированный процессор, который берет на себя выполнение следующих функций:

- выбор команд из ОЗУ, управление работой дисплея,
- редактирование данных, формирование массивов данных при вводе с клавиатуры и светового пера,
- преобразование информации (сжатие—растяжение, линейные перемещения, вращения, режим «окна» и т. д.).

Следует отметить существенный недостаток графического дисплея векторного типа: поскольку обычная ЭЛТ не может сохранять информацию дольше 0,1...0,3 с, то для создания устойчивого изображения на экране требуется его многократное воспроизведение (считывание данных из ОЗУ). Этот процесс называется *регенерацией изображения*. Частота считывания определяется особенностями зрения человека (человек перестает замечать дискретный характер зрительной информации, если она предъявляется чаще 25—30 раз в секунду) и обычно составляет 25—50 Гц. Поскольку скорость считывания информации из ОЗУ ограничена физическими возможностями (обычно 1...5 мГц), то ограниченным оказывается и объем информации, который может быть отображен на экране. Для типичного графического дисплея этот объем выражается примерно 1000 адресуемых точек — концов отрезков, дуг, центров окружностей и т. д.

В последнее время все чаще применяют дисплеи растрового типа. Изображение в дисплеях растрового типа представляет собой матрицу  $N \times N$  элементов (пикселей), каждый из которых может принимать некоторое значение яркости или цвета из допустимого диапазона. Каждому элементу изображения соответствует ячейка буферной памяти дисплея, используемой для регенерации изображения. Таким образом, регенерация выполняется всегда для всего экрана независимо от того, что на нем изображено.

Причина растущей популярности графических элементов растрового типа — их низкая стоимость, т. к. они используют дешевую полупроводниковую память, сравнительно небольшое число логических схем, генератор развертки и телевизионный монитор. Нет необходимости в быстрых линейных (точных) генераторах векторов и связанных с ними векторных усилителях отклоняющей системы. Для дисплеев со средней разрешающей способностью ( $256 \times 256$ ) можно использовать хорошо отработанные и недорогие системы отклонения и управления лучом бытовых телевизионных приемников.

Графический дисплей любого типа позволяет реализовать графический диалог между оператором и ЭВМ. Это обеспечивается использованием определенных технических средств: алфавитно-цифровой клавиатуры, функциональной клавиатуры, устройств управления (курсора), светового пера. Световое перо может работать либо в режиме указания элемента изображения (идентификация) или некоторой точки на экране (указание координат), либо трассировки (или следящего перекрестья). В режиме трассировки или следящего перекрестья используется небольшой знак (+ или подобный символ), являющийся опорной точкой на экране. Координаты центра этого значка всегда известны ЭВМ. Оператор, пользуясь световым пером, может перемещать перекрестье в любую желаемую точку экрана — таким образом можно обозначать концы отрезков прямой, центры окружностей и другие элементы геометрических объектов. Как правило, следящее перекрестье состоит из серии точек, составляющих пересекающиеся горизонтальный и вертикальный отрезки. Размеры его сравнимы с входным полем зрения пера. Специальные цепи ЭВМ определяют, какие точки лежат в поле зрения светового пера при данном положении перекрестья на экране. Затем вычисляется положение центра тяжести этой части точек перекрестья, и данная точка используется в качестве центра для следующего положения перекрестья.

Список команд для дисплейного процессора называется *дисплейным файлом*.

Наиболее часто используются следующие дисплейные команды:

позиционирование — команда управляет лучом ЭЛТ — устанавливает луч в заданную точку (с координатами, задаваемыми 10-ю младшими разрядами команды);

вычерчивание векторов (абсолютные и относительные, длинные и короткие);

управление яркостью;

команды перехода — указывают адрес одной из ячеек памяти, к которой нужно перейти (позволяет дисплейному процессору работать независимо от ЭВМ);

установка режима индикации;

переход с возвратом (переход к подпрограмме) — одна из важнейших команд в дисплее, которая особенно важна при отображении символов или небольших частей чертежа, многократно повторяющихся в общем изображении на экране. При отсутствии такой команды все эти части пришлось бы записывать многократным повторением одних и тех же команд. Подпрограмма, как у обычной ЭВМ, заканчивается командой восстановления адресного регистра для продолжения вывода (следующая команда после обращения к подпрограмме). Подпрограммы полезны только в том случае, если определяемые ими изображения могут быть размещены в любом месте экрана, для чего изображение кодируется в относительных координатах; режим символов.

Дисплейный файл практически представляет собой *программу*, предназначенную для исполнения дисплейным процессом. Многие особенности обычной программы для ЭВМ имеются и в дисплейном файле: команды передачи управления, подпрограммы, команды непосредственной загрузки регистров данными, содержащимися в команде и т. д. Такая программа исполняется как обычная программа для ЭВМ с использованием программного счетчика. Дисплейные файлы можно даже формировать тем же способом, что и программы для ЭВМ, т. е. писать команды для вывода необходимого изображения и затем переводить их в двоичный код. Однако дисплейные файлы редко формируют таким способом, т. к. в этом случае программы можно было бы использовать только для представления заранее определенных изображений. Вместо этого пишут программы для ЭВМ, одна из функций которых состоит в генерировании дисплейного файла. Этот процесс — процесс генерирования программы с помощью другой программы — очень похож на процесс трансляции (компиляции), поэтому называется *компиляцией дисплейного файла*.

*Компиляция* — преобразование графических макрокоманд в последовательность дисплейных команд, которая добавляется при этом к дисплейному файлу (добавляется некоторая графическая единица, например отрезок прямой). Координаты отрезка прямой в этом случае являются параметрами обращения к графической макрокоманде. Для иллюстрации этого процесса рассмотрим подробно работу типичной графической макрокоманды:

*LINE* (100, 150, 500, 600).

Данное обращение — команда на добавление к дисплейному файлу описания отрезка прямой, идущего от точки (100, 150) к точке (500, 600). К дисплейному файлу должны быть добавлены четыре дополнительные команды и указатель конца дисплейного файла переместится вперед на 4 позиции (рис. 25):

Установка $x$		100	Старое положение указателя конца дисплейного файла
Установка $y$		150	
Установка $\Delta x$		400	
Установка $\Delta y$	Чертить	450	

Новое  
положение  
указателя

Рис. 25

Входящие в состав компилятора подпрограммы могут быть сгруппированы в пакеты подпрограмм, которые загружаются совместно с основной программой.

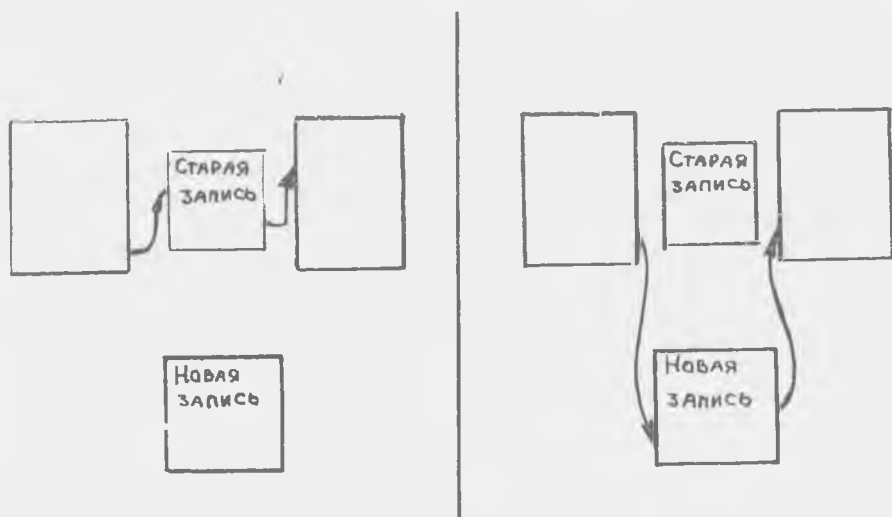
Параметры, входящие в состав обращений к макрокомандам либо вводятся пользователем, например, с пультовой пишущей машинки (или с алфавитно-цифровой клавиатуры дисплея) в виде значений двух пар координат (прикладная программа обращается к макрокоманде *LINE* с этими координатами в качестве параметров), либо вводятся с пульта устройства ввода графической информации («сколки»), либо вычисляются основной программой и подставляются как конкретные значения переменных, описанных в обращении к макрокоманде (т. е. графические данные должны образовывать некоторую структуру данных, переход которой к выводу изображения производится с помощью специального алгоритма визуализации).

Важной характеристикой компилятора дисплейного файла является скорость компиляции. Например, если необходимо поворачивать на экране изображение трехмерного объекта (машинностроительной детали), то при каждой команде на поворот программа должна компилировать новый дисплейный файл. Если новые дисплейные файлы нельзя получать чаще, чем 10 раз/с, то движение детали на экране станет прерывистым, а если частота меньше одного кадра в секунду, то оператор может потерять всякий контроль над программой. В большинстве случаев динамически изменяется только часть изображения, поэтому можно избежать полной повторной компиляции дисплейного файла. Но для этого изображение нужно представлять в виде отдельных сегментов, которые должны компилироваться в от-

дельные сегменты дисплейного файла. При этом для программиста необходимо добавить средства формирования отдельных записей (генерирование новых записей, переформирование и стирание их) и идентификации (именования) записей, которые нужно изменить или стереть.

Как же происходит работа с сегментированными дисплейными файлами? Пусть построен сегментированный дисплейный файл, содержащий записи  $R_1, \dots, R_n$ . Предположим, что нужно изменить содержание записи  $R_2$ . Например, эта запись может списывать объект, который непрерывно поворачивается, в связи с этим невозможно просто заменить содержимое  $R_2$  новыми дисплейными командами, поскольку существует значительная вероятность того, что дисплейный процессор попытается исполнить  $R_2$  в процессе замены. Такая ситуация обычно приводит к неопределенным результатам, если не приняты особые меры предосторожности при формировании записи, которые усложняют компилятор и уменьшают скорость компиляции.

Правильный путь состоит в использовании *двойной буферизации*. Новый вид записи формируется в некоторой неиспользуемой части памяти, а ее существующий вариант сохраняется в дисплейном файле. Когда новая запись полностью сформирована, она начинает использоваться вместо старого варианта, а соответственное место памяти освобождается (рис. 26).



Р и с. 26

## ВОПРОСЫ И УПРАЖНЕНИЯ

1. Чем отличаются графопостроители планшетного и рулонного типов?
2. Зачем в дисплее необходимо запоминающее устройство?
3. Что такое инкрементальный код графопостроителя?
4. Каков принцип действия светового пера? Какие режимы работы светового пера вы знаете?
5. Чем определяется координата вводимой точки чертежа с помощью устройств электромагнитного типа?
6. Как определяются опорные точки чертежа при его вводе в ЭВМ? Поясните на примере.
7. Назовите формы представления (методы кодирования) информации перед ее вычерчиванием на графопостроителе.
8. На чем основан метод следящего перекрестья?
9. Какое устройство обеспечивает графическое взаимодействие проектировщика и ЭВМ в автоматизированном рабочем месте?
10. Какие из чертежных автоматов — электромеханические графопостроители, электронные на запоминающих ЭЛТ, растровые электрохимические, электротермические имеют следующие преимущества перед другими: высокая точность, большая разрешающая способность, высокое качество линий и символов?
11. Какой из недостатков свойственен графопостроителям электромеханического типа (по сравнению с другими типами графопостроителей): а) невысокая скорость воспроизведения чертежа, б) невысокое качество линий и символов, в) невысокая разрешающая способность, г) невозможность получения чертежа на обычной бумаге.

## 3. Геометрические преобразования

---

### 3.1. ОДНОРОДНЫЕ КООРДИНАТЫ

#### 3.1.1. ОПРЕДЕЛЕНИЕ ОДНОРОДНЫХ КООРДИНАТ

Однородные координаты играют основную роль в проективной геометрии. Хотя в машинной графике используется евклидова геометрия, использование однородных координат дает много преимуществ.

Рассмотрим уравнение прямой

$$ax + by + c = 0, \quad (5)$$

если  $a, b, c$  — константы, а  $x$  и  $y$  — переменные, то выражение (5) содержит 2 члена первой степени и один член нулевой степени (константа  $c$ ), поэтому называется неоднородным (уравнение для точек данной прямой). Если же наоборот,  $a, b$  и  $c$  рассматриваются как переменные, а  $x, y$  — как константы, то все члены имеют одну и ту же степень и выражение называется *однородным* (уравнение для прямых, проходящих через точку  $(x, y)$ ). Таким образом, при обычном задании координат точки определяются неоднородным уравнением, а прямые — однородным. В проективной геометрии различие множества точек и множества прямых устраняется введением дополнительных, реально не существующих точек, которые называются несобственными точками («бесконечно удаленные точки»). Тогда оказывается справедливым принцип двойственности: каждому свойству точек соответствует определенное свойство прямых (например, для утверждения «существует только одна прямая, проходящая через 2 различные точки» справедливо двойственное утверждение «существует единственная точка пересечения двух прямых»). В машинной графике принцип двойственности может быть использован для более экономного представления программного обеспечения. Например, одна и та же процедура может быть использована для нахождения либо пересечения двух прямых, либо нахождения прямой, проходящей через две точки (в обоих случаях это связано с вычислением миноров определителя  $3 \times 3$ ).



Однородное представление двумерной точки  $(x, y)$  в общем случае имеет вид  $[\omega x, \omega y, \omega]$ , где  $\omega$  — любой ненулевой скаляр (скалярный множитель). Для преобразования однородных координат точки  $[a, b, c]$  обратно к ее двумерному представлению надо поделить первые две координаты на скалярный множитель:  $[a/c, b/c]$ .

Уравнение прямой в однородных координатах имеет вид

$$au + bv + c\omega = 0, \quad (6)$$

где  $u = x\omega$ ,  $v = y\omega$ .

В матричной форме уравнение запишется как

$$[u \ v \ \omega] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \omega [x \ y \ 1] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0. \quad (7)$$

Выбор скалярного множителя производится из соображений удобства вычислений. Часто берут  $\omega = 1$ , или используют  $\omega$  как масштабный множитель. Например, если можно работать только с целыми числами  $-2^{12} \leq x \leq 2^{12} - 1$ , то точка  $x = 0,25$ ,  $y = 0,1$  не может быть представлена, но если взять  $\omega = 20$ , то

$$[\omega x \ \omega y \ \omega] = [5 \ 2 \ 20].$$

Таким образом, второе преимущество применения однородных координат в машинной графике состоит в том, что легко могут быть обработаны точки с координатами вне допустимой области (выше или ниже) путем изменения величины  $\omega$ . Координаты точек, определяющих фрагмент изображения, могут быть преобразованы к допустимым границам. После этого, чтобы найти видимую часть фрагмента, можно использовать алгоритм отсечения.

Рассмотрим некоторые вопросы элементарной геометрии с использованием однородных координат.

### 3.1.2. ИСПОЛЬЗОВАНИЕ ОДНОРОДНЫХ КООРДИНАТ НА ПЛОСКОСТИ

Представление точки и прямой в однородных координатах. Точка плоскости в однородных координатах есть вектор-строка:

$$\bar{p} = [\omega x \ \omega y \ \omega],$$

прямая есть вектор-столбец:  $\lambda^T = [a \ b \ c]$ .

Уравнение прямой имеет вид  $\bar{p}\lambda = 0$ . Взаимное положение точки  $\bar{p}$  относительно прямой определяется скалярным произведением  $\bar{p}\bar{\lambda}$ :

$\bar{p}\bar{\lambda} = 0 \rightarrow$  точка на прямой,

$\bar{p}\bar{\lambda} > 0 \rightarrow$  точка в положительной полуплоскости,

$\bar{p}\bar{\lambda} < 0 \rightarrow$  точка в отрицательной полуплоскости.

Расстояние от точки до прямой (по перпендикуляру)

$$(\bar{p}\bar{\lambda}) \frac{1}{w} \frac{1}{\sqrt{c^2 + b^2}}$$

Уравнение прямой, проходящей через 2 заданные точки. Пусть заданы точки  $p_1 = (x_1 y_1 \omega_1)$  и  $p_2 = (x_2 y_2 \omega_2)$ . Произвольная точка  $p$  с координатами  $(x, y, \omega)$  коллинеарна точкам  $p_1$  и  $p_2$ , если ее координаты линейно зависят от координат точек  $p_1$  и  $p_2$ . Это означает, что определитель матрицы, в которой столбцы составлены из координат трех точек, должен равняться нулю, т. е.

$$\det \begin{bmatrix} x & x_1 & x_2 \\ y & y_1 & y_2 \\ \omega & \omega_1 & \omega_2 \end{bmatrix} = 0, \quad (8)$$

Это и есть уравнение прямой, проходящей через две точки  $p_1$  и  $p_2$ . В обычной форме уравнение прямой запишется как

$$x(y_1\omega_2 - \omega_1 y_2) + y(\omega_1 x_2 - x_1 \omega_2) + \omega(x_1 y_2 - y_1 x_2) = 0 \quad (9)$$

или в матричной форме

$$\bar{L} = \bar{p}_1 \times \bar{p}_2 = \begin{bmatrix} y_1 \omega_2 - \omega_1 y_2 \\ \omega_1 x_2 - x_1 \omega_2 \\ x_1 y_2 - y_1 x_2 \end{bmatrix}$$

(векторное произведение)

Точка пересечения двух заданных прямых. Пусть заданы две прямые  $L_1^T = [a_1 b_1 c_1]$  и  $L_2^T = [a_2 b_2 c_2]$ . Третья прямая с коэффициентами  $(a, b, c)$ , проходящая через точку пересечения прямых  $L_1$  и  $L_2$ , должна линейно зависеть от них, поэтому определитель матрицы, составленной из коэффициентов прямых, должен равняться нулю,

Имеем

$$\det \begin{bmatrix} a & a_1 & a_2 \\ b & b_1 & b_2 \\ c & c_1 & c_2 \end{bmatrix} = 0 \quad (10)$$

Раскрытие определителя дает

$$a(b_1c_2 - c_1b_2) + b(c_1a_2 - a_1c_2) + c(a_1b_2 - b_1a_2) = 0, \quad (11)$$

т. е. точка пересечения  $p$  является векторным произведением:

$$\bar{p} = \bar{L}_1 \times \bar{L}_2 = [(b_1c_2 - c_1b_2) \ (c_1a_2 - a_1c_2) \ (a_1b_2 - b_1a_2)]$$

### 3.1.3. ОДНОРОДНЫЕ КООРДИНАТЫ В ТРЕХМЕРНОМ ПРОСТРАНСТВЕ

Представление точки и плоскости в однородных координатах. Точка описывается вектором  $[wx \ wy \ wz \ w]$ . Плоскость представляет вектор-столбец

$$\bar{\alpha} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$

Уравнение плоскости  $\bar{p} \cdot \bar{\alpha} = 0$ . Скалярное произведение  $\bar{p} \cdot \bar{\alpha}$  определяет расстояние от  $p$  до плоскости  $\bar{\alpha}$ . Оно положительно, если точка  $p$  лежит по одну сторону плоскости, и отрицательно, если она лежит по другую сторону. Действительное расстояние равно:

$$(\bar{p} \cdot \bar{\alpha}) \frac{1}{w} \frac{1}{\sqrt{a^2 + b^2 + c^2}}$$

Пересечение плоскостей. Три плоскости  $\gamma_1, \gamma_2, \gamma_3$  пересекаются в одной точке  $p$  так, что  $\bar{p} \gamma_1 = 0, \bar{p} \gamma_2 = 0, \bar{p} \gamma_3 = 0$ . Если через  $Q$  обозначить матрицу

$$Q = \begin{bmatrix} A_1 & A_2 & A_3 & 0 \\ B_1 & B_2 & B_3 & 0 \\ C_1 & C_2 & C_3 & 0 \\ D_1 & D_2 & D_3 & 1 \end{bmatrix},$$

то точка  $p$  является нижней строкой обратной матрицы  $Q^{-1}$ . Если  $Q^{-1}$  не существует, то три плоскости не пересекаются в одной точке.

Проведение плоскости через 3 заданные точки. Три точки определяют плоскость, когда они не коллинеарны, т. е. не лежат на одной прямой. Три точки  $\vec{p} = [p_1 p_2 p_3 p_4]$ ,  $\vec{q} = [q_1 q_2 q_3 q_4]$ ,  $\vec{r} = [r_1 r_2 r_3 r_4]$  не являются коллинеарными, если матрица

$$Q = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ q_1 & q_2 & q_3 & q_4 \\ r_1 & r_2 & r_3 & r_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

невырождена (несингулярна), т. е. имеет обратную матрицу, тогда

$$\vec{a} = Q^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ d \end{bmatrix}$$

$a$  может быть любым числом  $\neq 0$ , например, равным  $\det Q$ .

## 3.2. ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

### 3.2.1. ДВУМЕРНЫЕ ПРЕОБРАЗОВАНИЯ

В машинной графике часто возникает необходимость в преобразовании сформированных изображений. Например, может потребоваться увеличить изображение, чтобы лучше рассмотреть детали, или уменьшить его, чтобы разместить в пределах экрана. Часто требуется разместить на экране отдельные фрагменты изображения (например, изображения микросхем при размещении их на печатной плате). Любое преобразование может быть представлено в виде комбинации трех простых преобразований — сдвига, масштабирования и поворота. Все эти преобразования используются для вычисления координат новой точки  $(x', y')$  по координатам точки  $(x, y)$  в исходном описании изображения.

*Сдвиг*  $x' = x + T_x$ ,  $y' = y + T_y$

или в матричной форме:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \quad (12)$$

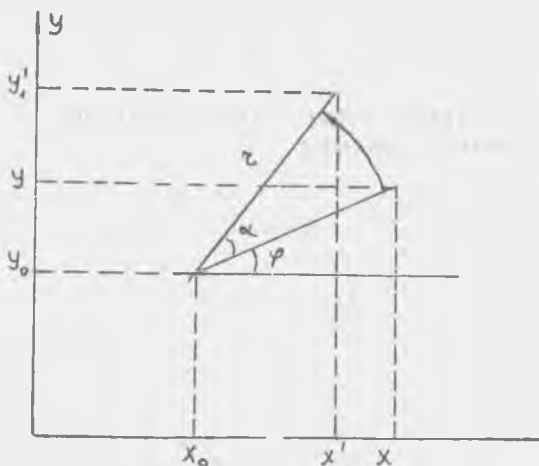
Масштабирование :

$$x' = x \cdot S_x, \quad y' = y \cdot S_y$$

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Масштабирование выполняется относительно начала координат. Если  $S_x$  или  $S_y$  меньше нуля, получим зеркальное изображение относительно соответствующей оси. Если  $S_x \neq S_y$ , то изображение искажается из-за неодинакового изменения формы вдоль осей  $x$  и  $y$ .

Поворот радиус-вектора точки  $(x, y)$  на угол  $\alpha$  против часовой стрелки относительно точки с координатами  $(x_0, y_0)$  показан на рис. 27:



Р и с. 27

$$\begin{cases} x - x_0 = r \cos \varphi, \\ y - y_0 = r \sin \varphi, \end{cases} \quad (14)$$

$$\begin{cases} x' - x_0 = r \cos (\varphi + \alpha), \\ y' - y_0 = r \sin (\varphi + \alpha). \end{cases} \quad (15)$$

Заменяя синус на косинус в системе (15) и затем подставляя значения из системы (14), получим

$$\begin{cases} x' = x_0 + (x - x_0) \cos \alpha - (y - y_0) \sin \alpha, \\ y' = y_0 + (x - x_0) \sin \alpha - (y - y_0) \cos \alpha. \end{cases} \quad (16)$$

Чаще поворот относительно произвольной точки представляется как последовательность трех преобразований: сдвиг точки, являющейся центром поворота, в начало системы координат, поворот относительно начала координат, обратный сдвиг центра поворота в исходное положение. При этом формулы для поворота упрощаются:

$$\begin{cases} x' = x \cos \alpha - y \sin \alpha, \\ y' = x \sin \alpha + y \cos \alpha. \end{cases} \quad (17)$$

Такое преобразование эквивалентно повороту системы координат по часовой стрелке на угол  $\alpha$ .

В матричной форме преобразование поворота (в однородных координатах) запишется следующим образом:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

### 3.2.2. ТРЕХМЕРНЫЕ ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

Поворот плоскости  $XU$  можно представить как частный случай поворота в трехмерном пространстве вокруг оси  $z$  (рис. 28):

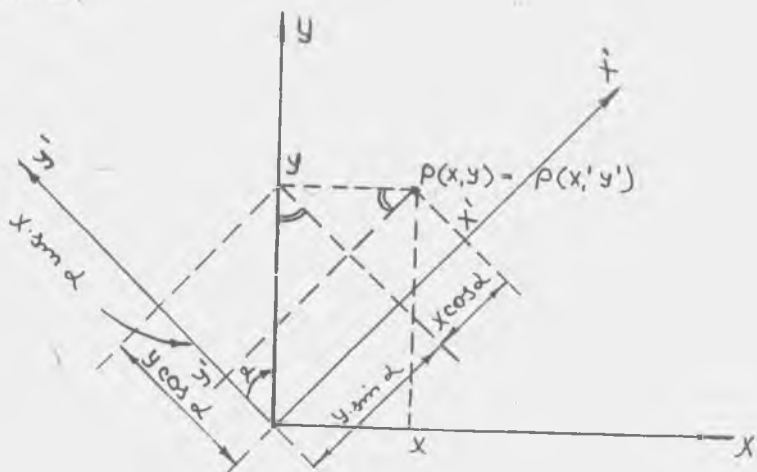


Рис. 28

$$\begin{aligned}x' &= x \cos \alpha + y \sin \alpha + z \cdot 0, \\y' &= -x \sin \alpha + y \cos \alpha + z \cdot 0, \\z' &= x \cdot 0 + y \cdot 0 + z \cdot 1\end{aligned}\quad (19)$$

В матричной форме преобразование (19) можно записать так:

$$[x' \ y' \ z'] = [x \ y \ z] \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} = [x \ y \ z] \cdot R_z(\alpha) \quad (20)$$

Аналогично поворот вокруг оси  $y$  на угол  $\beta$

$$[x' \ y' \ z'] = [x \ y \ z] \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} = [x \ y \ z] \cdot R_y(\beta) \quad (21)$$

Поворот вокруг оси  $x$  на угол  $\gamma$

$$[x' \ y' \ z'] = [x \ y \ z] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & \sin \gamma \\ 0 & -\sin \gamma & \cos \gamma \end{bmatrix} = [x \ y \ z] \cdot R_x(\gamma) \quad (22)$$

Преобразование поворота общего вида относительно трех координатных осей (центр вращения в начале координат) можно получить как суперпозицию трех плоских поворотов (перемножение трех матриц). Матричное умножение, однако, не является коммутативной операцией, поэтому необходимо задать некоторый порядок выполнения поворотов. Обычный порядок — поворот относительно оси  $x$ , затем  $y$  и затем  $z$ ; тогда

$$\begin{aligned}[x' \ y' \ z'] &= [x \ y \ z] \cdot R_x(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) = \\ &= [x \ y \ z] \cdot R_{xyz}(\alpha, \beta, \gamma)\end{aligned}\quad (23)$$

*Перенос:* перенос точки производится путем добавления к каждой координате точки положительной или отрицательной константы:

$$[x' \ y' \ z'] = [x \ y \ z] + [T_x \ T_y \ T_z] \quad (24)$$

*Масштабирование:* в этом преобразовании каждая координата умножается на некоторый коэффициент:

$$[x' \ y' \ z'] = [x \ y \ z] \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \quad (25)$$

$S_x, S_y, S_z$  — масштабные множители по осям.

Если  $S_x = S_y = S_z = c$ , то матрицу можно заменить скаляром  $c$ .

### 3.3. ПЕРСПЕКТИВНОЕ ИЗОБРАЖЕНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ

Так как экран дисплея или лист чертежа представляют собой двумерное пространство, мы можем изобразить объекты только в виде проекций. Проецирование требует большого объема вычислений, поэтому чаще всего ограничиваются использованием параллельной (ортогональной) проекции, для которой трехмерная точка  $P(x, y, z)$  переводится в проекцию  $P_s(x_s, y_s)$  этой точки простым преобразованием

$$x_s = x, \quad y_s = y, \quad z_s = 0. \quad (26)$$

Для центрального (перспективного) преобразования задача имеет более сложное решение. Рассмотрим схему получения центральной проекции точки  $P(x, y, z)$ , в которой точка наблюдения расположена на расстоянии  $f$  от картинной плоскости ( $f$  — фокусное расстояние), начало трехмерной системы координат лежит в картинной плоскости ось  $z$  перпендикулярна картинной плоскости и направлена в сторону от наблюдателя (рис. 29,а). Тогда из подобия треугольников (рис. 29,б) получаем.

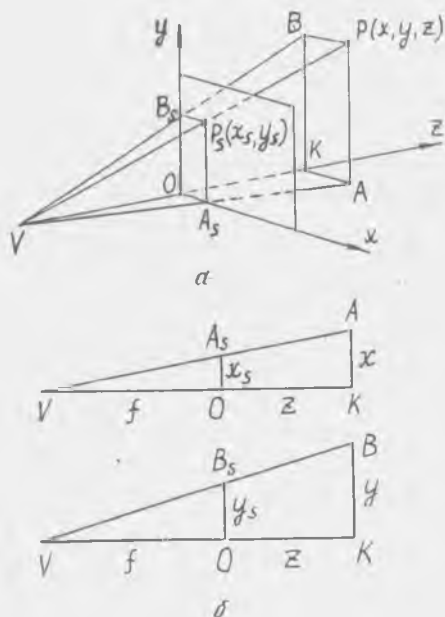


Рис. 29



$$\frac{x_s}{f} = \frac{x}{f+z}, \quad \frac{y_s}{f} = \frac{y}{f+z}, \quad (27)$$

откуда

$$x_s = \frac{f}{f+z} x = \frac{1}{1+(z/f)} x; \quad y_s = \frac{f}{f+z} y = \frac{1}{1+(z/f)} y. \quad (28)$$

Это преобразование принципиально отличается от геометрических преобразований поворота, сдвига и масштабирования, так как является нелинейным. Нелинейность связана с необходимостью выполнять деление на фокусное расстояние  $f$ , а для других преобразований достаточно операций умножения и сложения.

### 3.4. ИСПОЛЬЗОВАНИЕ ОДНОРОДНЫХ КООРДИНАТ ПРИ ВЫПОЛНЕНИИ ПРЕОБРАЗОВАНИЙ

Использование однородных координат позволяет совместить последовательно выполняемые преобразования в одно сложное преобразование графической информации. Для этого следует перемножить матрицы, соответствующие простым преобразованиям, и результирующую матрицу использовать для получения новых координат точки. Совмещение преобразований особенно эффективно, если одному и тому же преобразованию подвергаются многие графические элементы.

Для трехмерных геометрических преобразований в однородных координатах справедливы следующие соотношения:  
*сдвиг*

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} = \bar{X} \cdot \bar{T}; \quad (29)$$

*масштабирование*

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \bar{X} \cdot \bar{S}; \quad (30)$$

*поворот на угол  $\alpha$  вокруг оси  $x$ ;*

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \bar{X} \cdot \bar{R}_x(\alpha); \quad (31)$$

поворот на угол  $\beta$  вокруг оси  $y$ :

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \bar{X} \cdot \bar{R}_y(\beta); \quad (32)$$

поворот на угол  $\gamma$  вокруг оси  $z$ :

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \bar{X} \cdot \bar{R}_z(\gamma). \quad (33)$$

Центральное проецирование выполняется с помощью следующего преобразования:

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/f \\ 0 & 0 & 0 & 1 \end{bmatrix} = \bar{X} \bar{P}. \quad (34)$$

Совмещение преобразований выполняется путем последовательного перемножения (в нашем случае — справа) матриц отдельных преобразований:

$$\bar{X}' = \bar{X} \cdot \bar{Q}_1 \cdot \bar{Q}_2 \cdot \dots \cdot \bar{Q}_n = \bar{X} \cdot \bar{Q} \quad (35)$$

В связи с тем, что в литературе по машинной графике встречаются различные подходы к выполнению преобразований графической информации, следует сделать несколько замечаний.

1. Хотя стандартное расположение осей координат таково, что плоскость  $XOY$  параллельна картинной плоскости и ось  $z$  направлена от наблюдателя (это соответствует так называемой «левой системе координат»), используется также и система координат, в которой ось  $z$  направлена к наблюдателю («правая система координат»). В левой системе координат положительное направление для углов соответствует движению по часовой стрелке, в правой системе координат — движению против часовой стрелки.

2. В аналитической геометрии преобразования выполняются при помощи поворота системы координат, а объект (точки, геометрические фигуры) остается «на месте». В машинной графике система координат зафиксирована, а объект поворачивается, сдвигается, масштабируется относительно данной системы координат. Очевидно, что поворот объекта на угол  $+\Theta$  соответствует повороту системы координат на угол  $-\Theta$ .

3. Иногда координаты точки представляются вектором-столбцом. В этом случае для выполнения аналогичных геометрических преобразований матрица преобразования должна быть транспонирована и записана слева, а вектор-столбец — справа:

$$\bar{X}' = \bar{P} \bar{X}, \quad \bar{P} = \bar{Q}^T.$$

При совмещении преобразований матрица для последнего выполняемого преобразования должна быть самой левой, а для первого преобразования — самой правой:

$$\bar{X}' = \bar{P}_n \bar{P}_{n-1} \dots \bar{P}_2 \bar{P}_1 \bar{X}.$$

4. При выполнении серии поворотов вокруг различных координатных осей следует придерживаться такого порядка: ось  $x$ , ось  $y$ , ось  $z$ .

5. При выполнении перспективной проекции результирующие координаты получаются в форме

$$[u \quad v \quad 0 \quad \omega]$$

Чтобы получить координаты проекции точки в системе координат картинной плоскости, нужно две первые компоненты результирующего вектора разделить на четвертую компоненту:

$$x_s = u/\omega, \quad y_s = v/\omega. \quad (36)$$

Для ортогональной проекции такое деление выполнять не нужно.

При оформлении преобразования в виде программного модуля необходимо выделить:

входные параметры — координаты точки  $x, y, z$ ;

выходные параметры — координаты проекции точки  $x_s, y_s$ ;

управляющие параметры: углы поворота  $\alpha, \beta, \gamma$ ; масштабные множители  $S_x, S_y, S_z$ ; сдвиги по осям:  $T_x, T_y, T_z$ ; центр проекции  $f_0$  (точка зрения).

По умолчанию выбираются следующие значения параметров:

$$T_x = T_y = T_z = 0, \quad S_x = S_y = S_z = 1, \quad \alpha = \beta = \gamma = 0.$$

## ВОПРОСЫ И УПРАЖНЕНИЯ

1. Укажите матрицу преобразования фигуры  $A_i$  в фигуру  $B_i$ ,  $i = 1, 2, 3$  (рис. 30).

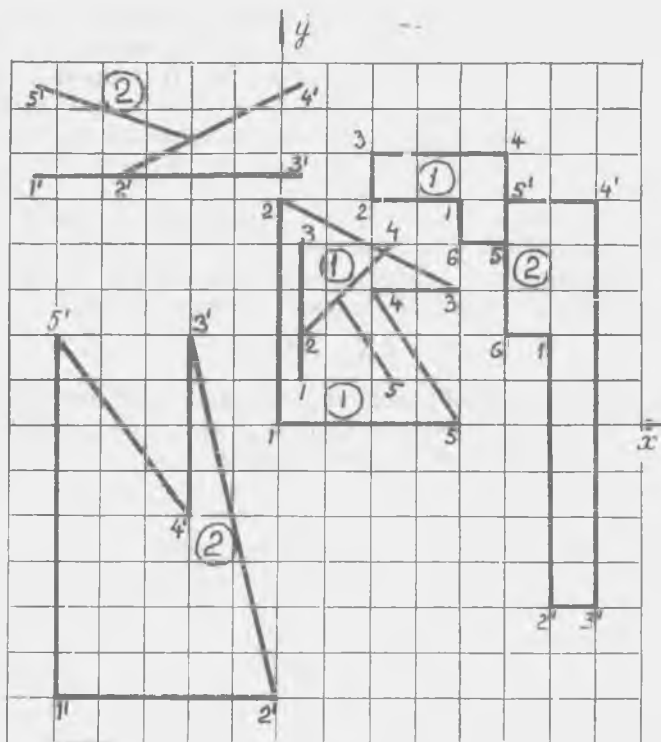


Рис. 30

2. Найдите координаты точки  $P_s$ , которая получается из точки  $P(10, 20, 30)$  с помощью преобразования масштабирования с множителями  $S_x = 15$ ,  $S_y = 0,75$ ,  $S_z = 0,5$ .

3. Найдите координаты точки  $P_{pp}$ , которая получается из точки  $P(10, 20, 30)$  с помощью ортогонального проецирования.

4. Найдите координаты точки  $P_{cpr}$ , которая получается из точки  $P(10, 20, 30)$  с помощью центрального проецирования с фокусным расстоянием  $f = 200$ .

5. Найдите координаты точки  $P_{xz}$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно оси  $x$  на угол  $\alpha = 90^\circ$ , затем относительно оси  $z$  на угол  $\beta = 30^\circ$ .

6. Найдите координаты точки  $P_{yz}$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно оси  $y$  на угол  $\alpha = 60^\circ$ , а затем относительно оси  $z$  на угол  $\beta = 30^\circ$ .

7. Найдите координаты точки  $P_{xyz}$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно осей  $x$ ,  $y$ ,  $z$  соответственно на углы  $\alpha = 90^\circ$ ,  $\beta = 60^\circ$ ,  $\gamma = 30^\circ$ .

8. Найдите координаты точки  $P_T$ , которая получается из точки  $P(10, 20, 30)$  с помощью преобразования переноса с параметрами  $T_x = 5$ ,  $T_y = -5$ ,  $T_z = -15$ .

9. В трехмерном пространстве задана точка  $P(10, 20, 30)$ . Найдите координаты точки  $P_z$ , которая получается в результате поворота точки  $P$  относительно оси  $z$  на угол  $\alpha = 30^\circ$ .

10. Найдите координаты точки  $P_y$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно оси  $y$  на угол  $\beta = 60^\circ$ .

11. Найдите координаты точки  $P_x$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно оси  $x$  на угол  $\gamma = 90^\circ$ .

12. Найдите координаты точки  $P_{xy}$ , которая получается в результате поворота точки  $P(10, 20, 30)$  относительно оси  $x$  на угол  $\alpha = 90^\circ$ , затем относительно оси  $y$  на угол  $\beta = 60^\circ$ .

## 4. Позиционные задачи машинной графики

---

### 4.1. ДОПУЩЕНИЯ И ОПРЕДЕЛЕНИЯ

Решение многих проблем машинной графики, например, таких как анализ видимости объектов или их геометрическое моделирование, требует вычисления относительного положения графических элементов. К таким задачам относятся определение взаимного положения прямых и точек, определение принадлежности точки замкнутой области, определение части отрезка, попадающего внутрь многоугольника и т. д. Решая такие задачи, будем предполагать, что отрезки прямых определяются их концевыми точками, и что такие точки упорядочены. Будем считать, что все отрезки прямых направлены снизу вверх для негоризонтальных отрезков и слева направо для горизонтальных отрезков. Будем говорить, что точка лежит справа от отрезка, если она лежит справа от наблюдателя, движущегося вдоль прямой от начальной концевой точки к конечной концевой точке отрезка. Будем также говорить, что точка загорожена отрезком, если горизонтальная прямая, проведенная через эту точку, пересекает отрезок в точке, координата  $x$  которой меньше, чем у данной точки. Отношение невидимости не определено, если точка лежит выше верхней или ниже нижней концевой точки, или принадлежит отрезку, который горизонтален. Когда отношение «невидимости» определено, то оно эквивалентно отношению «лежит справа от» (отрезка).

### 4.2. ПОЛОЖЕНИЕ ТОЧКИ ОТНОСИТЕЛЬНО ПРЯМОЙ

Пусть  $x, y, z$  являются координатами точки  $P$ , а  $P_1(x_1, y_1, \omega_1)$  и  $P_2(x_2, y_2, \omega_2)$  — концевые точки отрезка  $L$ . ( $P_1$  — начальная точка отрезка  $L$ ,  $P_2$  — конечная точка отрезка  $L$ ). Если точка  $P$  лежит на прямой, включающей отрезок  $L$ , то ее координаты линейно зависят от координат точек  $P_1$  и  $P_2$ .

Это означает, что определитель матрицы, в которой столбцы составлены из координат трех точек  $P, P_1, P_2$ , должен быть равен нулю, т. е.

$$\det \begin{bmatrix} x & x_1 & x_2 \\ y & y_1 & y_2 \\ \omega & \omega_1 & \omega_2 \end{bmatrix} = 0. \quad (37)$$

Раскрывая определитель, можно переписать уравнение в более удобной форме:

$$x(y_1\omega_2 - \omega_1y_2) + y(\omega_1x_2 - x_1\omega_2) + \omega(x_1y_2 - y_1x_2) = 0 \quad (38)$$

Если определитель меньше нуля, то точка  $P$  лежит справа от прямой, если больше нуля — то слева от прямой, включающей отрезок  $L$ .

### 4.3. ТОЧКА ПЕРЕСЕЧЕНИЯ ДВУХ ПРЯМЫХ

Точка пересечения двух прямых  $L_1$  и  $L_2$  с коэффициентами  $(a_1, b_1, c_1)$  и  $(a_2, b_2, c_2)$  может быть найдена аналогичным образом. Третья прямая  $L$  с коэффициентами  $(a, b, c)$ , проходящая через эту точку, должна линейно зависеть от первых двух и поэтому определитель матрицы, составленной из коэффициентов прямых, должен быть равен нулю, т. е.

$$\det \begin{bmatrix} a & a_1 & a_2 \\ b & b_1 & b_2 \\ c & c_1 & c_2 \end{bmatrix} = 0. \quad (39)$$

Раскрытие определителя дает

$$a(b_1c_2 - c_1b_2) + b(c_1a_2 - a_1c_2) + c(a_1b_2 - b_1a_2) = 0. \quad (40)$$

Так как  $(a, b, c)$  являются коэффициентами произвольной прямой, проходящей через точку пересечения  $L_1$  и  $L_2$ , то выражения в скобках являются координатами их пересечения:

$$x = \frac{b_1c_2 - c_1b_2}{a_1b_2 - b_1a_2}, \quad y = \frac{c_1a_2 - a_1c_2}{a_1b_2 - b_1a_2} \quad (41)$$

### 4.4. ПЕРЕСЕЧЕНИЕ ОТРЕЗКОВ

Используем полученный результат, чтобы проверить, пересекаются ли два отрезка прямых, и если да, найдем это пересечение.

Если два отрезка определяются точками  $P_1$  и  $P_2$ ,  $P_3$  и  $P_4$ , то они пересекаются, если координаты  $P_1$  и  $P_2$  при подстановке в уравнение для прямой, соединяющей  $P_3$  и  $P_4$ , дают различные знаки. Аналогичное отношение имеет место для точек  $P_3$  и  $P_4$  относительно отрезка  $P_1 P_2$  (рис. 31). Поэтому необходимо вычислить знаки следующих четырех величин:

$$\begin{aligned} S_1 &= x_1(y_3w_4 - w_3y_4) + y_1(w_3x_4 - x_3w_4) + w_1(x_3y_4 - y_3x_4) \\ S_2 &= x_2(y_3w_4 - w_3y_4) + y_2(w_3x_4 - x_3w_4) + w_2(x_3y_4 - y_3x_4), \\ S_3 &= x_3(y_1w_2 - w_1y_2) + y_3(w_1x_2 - x_1w_2) + w_3(x_1y_2 - y_1x_2), \\ S_4 &= x_4(y_1w_2 - w_1y_2) + y_4(w_1x_2 - x_1w_2) + w_4(x_1y_2 - y_1x_2). \end{aligned} \quad (42)$$

Условие пересечения состоит в том, что  $S_1$  и  $S_2$ , — аналогично  $S_3$  и  $S_4$ , имеют разные знаки, т. е. произведения  $S_1 S_2$  и  $S_3 S_4$  отрицательны.

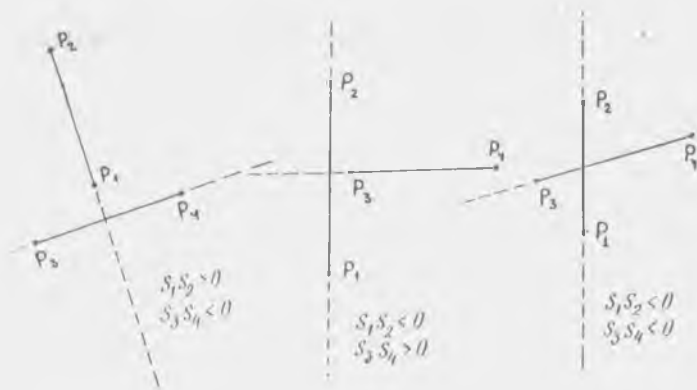


Рис. 31

Если условия для пересечения удовлетворены, то координаты общей точки могут быть найдены решением следующей пары линейных уравнений:

$$\begin{aligned} x(y_3w_4 - w_3y_4) + y(w_3x_4 - x_3w_4) + w(x_3y_4 - y_3x_4) &= 0, \\ x(y_1w_2 - w_1y_2) + y(w_1x_2 - x_1w_2) + w(x_1y_2 - y_1x_2) &= 0. \end{aligned}$$

Отметим, что число неизвестных равно числу уравнений, потому что значение  $w$  произвольно.

Условия пересечения могут быть записаны в краткой форме:

$$\begin{aligned} S_1 &= \det(\bar{P}_1, \bar{P}_3, \bar{P}_4), \quad S_2 = \det(\bar{P}_2, \bar{P}_3, \bar{P}_4), \quad S_1 S_2 < 0, \\ S_3 &= \det(\bar{P}_3, \bar{P}_1, \bar{P}_2), \quad S_4 = \det(\bar{P}_4, \bar{P}_1, \bar{P}_2), \quad S_3 S_4 < 0. \end{aligned} \quad (43)$$



Тогда уравнениями для точки пересечения будут

$$\begin{cases} \det(\bar{P}, \bar{P}_1, \bar{P}_2) = 0, \\ \det(\bar{P}, \bar{P}_3, \bar{P}_4) = 0. \end{cases} \quad (44)$$

Если одно или больше  $S_i$  равны нулю, то имеем особый случай, и сравнение знаков не может быть выполнено.

Здесь нужно отметить, что если выбираются случайным образом четыре точки на евклидовой плоскости, то маловероятно, чтобы любые три из них были коллинеарными. Однако, когда переходим к дискретной плоскости, где координаты изменяются в небольших диапазонах (например, между 0 и 511), ситуация изменяется. Поэтому при ограниченной точности, используемой в дисплеях, такие случаи часто встречаются на практике.

Если  $S_1$  равно нулю, то  $P_1$  лежит на прямой, определенной точками  $P_3$  и  $P_4$ . Чтобы найти, лежит ли она также на отрезке, определяемом этими точками, необходимо проверить только знаки  $S_3$  и  $S_4$ . Если один из них равен нулю, то это значит, что две точки совпадают. Если обе равны нулю, то значит четыре точки коллинеарны. Частичное совпадение отрезков может быть проверено сравнением либо  $x$ , либо  $y$  координат.

#### 4.5. ПОЛОЖЕНИЕ ТОЧКИ ОТНОСИТЕЛЬНО МНОГОУГОЛЬНИКА

Если вершины многоугольника  $A_i(x_i, y_i, \omega_i)$  упорядочены по часовой стрелке, то точка  $P(x, y, \omega)$  лежит внутри многоугольника, если она лежит всегда справа для наблюдателя,двигающегося по сторонам. Если многоугольник выпуклый, то величина  $\det(P, A_i, A_{i+1})$  отрицательна для всех сторон. В этом случае решение проблемы ясно.

Пример. Пусть вершинами выпуклого многоугольника являются точки  $A_1(0, 0)$ ,  $A_2(3, 3)$ ,  $A_3(6, 3)$ ,  $A_4(9, 0)$ ,  $A_5(5, -4)$  (рис. 32). Пусть  $x$  и  $y$  обозначают координаты точки  $P$ , в этом случае имеют место пять неравенств:

$$\begin{array}{ll} -3x + 3y < 0 & \text{или } x > y \\ 3y - 9 < 0 & \text{или } y < 3 \\ 3x + 3y - 27 < 0 & \text{или } x + y < 9 \\ 4x - 4y - 36 < 0 & \text{или } x - y < 9 \\ -4x - 5y < 0 & \text{или } 4x > -5y \end{array}$$

Для точки  $A$  с координатами (3, 1) левые части первого множества неравенств имеют следующие значения:  $-6$ ,  $-6$ ,  $-15$ ,  $-28$ ,  $-17$ . Для точки  $B$  с координатами (1, 2) левая часть первого неравенства равна 3. Если многоугольник не выпуклый,

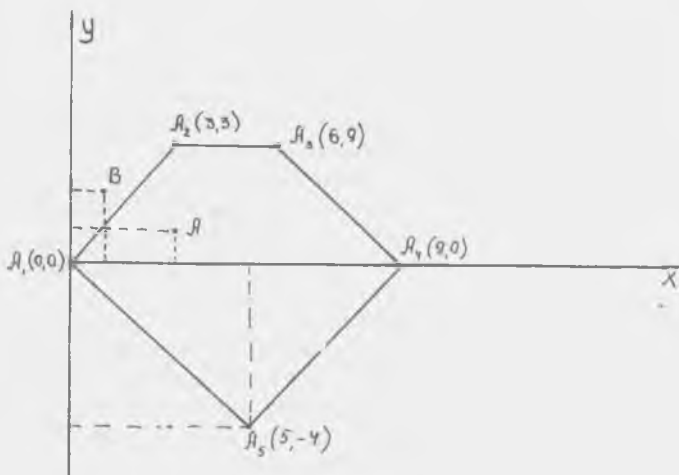


Рис. 32

то такого простого решения нет. Можно показать, что любая точка внутри невыпуклого многоугольника  $\pi$  принадлежит выпуклому многоугольнику, образованному некоторыми из сторон  $\pi$  и их продолжениями. Однако поиск таких многоугольников является трудной задачей. В некоторых приложениях они могут быть заданы заранее, например, если невыпуклый многоугольник проектируется как часть выпуклого многоугольника. Усилия по поиску таких многоугольников могут быть оправданы, если планируется проверка положения большого числа точек относительно фиксированного (единственного) невыпуклого многоугольника.

Другое решение, справедливое для любого многоугольника, заключается в проведении луча  $R$  через  $P$  и нахождении числа пересечений его со всеми сторонами  $\pi$ .

Луч не должен проходить через вершины многоугольника. Тогда точка  $P$  находится внутри многоугольника  $\pi$ , если число пересечений  $R$  и  $\pi$  нечетно, и  $P$  находится вне  $\pi$ , если число пересечений четно. Недостатком метода является необходимость решения множества систем уравнений, что требует значительно большей работы, чем проверка знаков неравенств. Кроме того, необходимо учесть усилия по сортировке точек пересечения. Таким образом, метод полезен только для невыпуклых многоугольников.

Другой способ заключается в вычислении суммы углов. Точка  $P$  соединяется со всеми вершинами  $A_i$  многоугольника  $\pi$ . Определяются углы между отрезками  $PA_i$  и  $PA_{i+1}$  (выбирается меньший из двух углов между  $PA_i$  и  $PA_{i+1}$ , углы против часовой

стрелки положительны, по часовой стрелке — отрицательны). Точка  $P$  находится вне многоугольника  $\pi$ , если сумма углов равна нулю, и внутри  $\pi$ , если сумма углов равна  $2\pi$  (рис. 33).

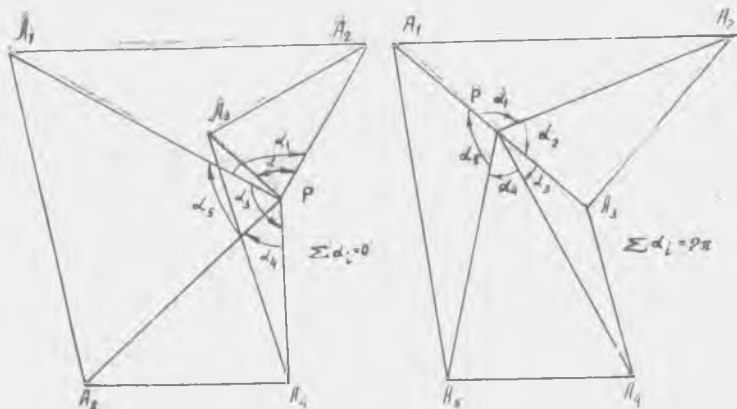


Рис. 33

#### 4.6. ОТСЕЧЕНИЕ ОТРЕЗКА ВЫПУКЛЫМ МНОГОУГОЛЬНИКОМ

**Задача.** Если  $A_1, A_2, \dots, A_n$  являются вершинами выпуклого многоугольника  $\pi$  на плоскости, найти часть отрезка прямой  $L$  (определенного своими концевыми точками  $P_1$  и  $P_2$ ), которая лежит внутри сторон многоугольника.

Отметим, что наивное решение, заключающееся в проверке, лежат ли точки  $P_1$  и  $P_2$  внутри многоугольника, является неточным, т. к. отрезок может пересекать выпуклый многоугольник, даже если концевые точки находятся вне многоугольника. Возможно следующее решение, основанное на материале п. 4.4. Сначала вычислим  $n$  величин:

$$S_i = \det(\bar{A}_i, \bar{P}_1, \bar{P}_2) = x_i (y_1 \omega_2 - \omega_1 y_2) + y_i (\omega_1 x_2 - x_1 \omega_2) + \omega_i (x_1 y_2 - y_1 x_2), \quad i = 1, 2, \dots, n.$$

При этом возможны варианты, когда:

1. Все  $S_i$  не равны нулю и имеют равные знаки. Тогда все вершины многоугольника лежат по одну сторону от  $L$  и никакая часть прямой не лежит внутри многоугольника.

2. Одна из величин равна нулю, а все оставшиеся имеют одинаковые знаки. Тогда прямая проходит через одну из вер-

шин, но все другие вершины находятся по одну сторону от прямой. Этот случай эквивалентен первому случаю.

3. Две величины равны нулю, а оставшиеся имеют одинаковые знаки. Поскольку  $\pi$  является выпуклым, это может быть только тогда, когда две вершины лежат на  $L$ . Тогда имеем частный случай, когда одна из сторон многоугольника лежит на той же прямой, на которой находится отрезок  $L$ .

4. Одна или две из величин равны нулю, а оставшиеся имеют различные знаки. Это означает, что  $L$  пересекает  $\pi$ , одновременно проходя через одну или две его вершины. Поскольку  $\pi$  является выпуклым, имеется только две смены знаков, и можно рассмотреть этот случай совместно со следующим.

5. Все  $S_i$  не равны нулю, но имеют разные знаки. Тогда будет две смены знака по периметру многоугольника.

Используем обозначения  $A_j, A_{j+1}$  и  $A_k, A_{k+1}$ , чтобы отметить пару вершин многоугольника, между которыми происходит эта смена знаков. Переменная  $S_i$  может быть равна нулю в одной из вершин одной или двух пар, если имеет место случай (4). Отрезок (сторону многоугольника), соединяющий точки первой пары, обозначим как  $L_1$ , а отрезок, соединяющий точки второй пары, как  $L_2$ . Заметим, что безразлично, какую пару считать первой, а какую второй.

Чтобы определить, пересекает ли прямая линия, включающая отрезок, многоугольник (последние 2 случая), мы должны проверить положение точек  $P_1$  и  $P_2$  относительно отрезков  $L_1$  и  $L_2$ , т. е. найти знаки четырех величин:

$$U_1 = \det (P_1, A_j, A_{j+1}) = x_1 (y_j \omega_{j+1} - \omega_j y_{j+1}) + \\ + y_1 (\omega_j x_{j+1} - x_j \omega_{j+1}) + \omega_1 (x_j y_{j+1} - y_j x_{j+1})$$

(положение точки  $P_1$  относительно отрезка  $L_1$ );

$$U_2 = \det (\bar{P}_2, \bar{A}_j, \bar{A}_{j+1}) = x_2 (y_j \omega_{j+1} - \omega_j y_{j+1}) + \\ + y_2 (\omega_j x_{j+1} - x_j \omega_{j+1}) + \omega_2 (x_j y_{j+1} - y_j x_{j+1}),$$

(положение точки  $P_2$  относительно отрезка  $L_1$ );

$$U_3 = \det (\bar{P}_1, \bar{A}_k, \bar{A}_{k+1}) = x_1 (y_k \omega_{k+1} - \omega_k y_{k+1}) + \\ + y_1 (\omega_k x_{k+1} - x_k \omega_{k+1}) + \omega_1 (x_k y_{k+1} - y_k x_{k+1})$$

(положение точки  $P_1$  относительно отрезка  $L_2$ );

$$U_4 = \det (\bar{P}_2, \bar{A}_k, \bar{A}_{k+1}) = x_2 (y_k \omega_{k+1} - \omega_k y_{k+1}) + \\ + y_2 (\omega_k x_{k+1} - x_k \omega_{k+1}) + \omega_2 (x_k y_{k+1} - y_k x_{k+1})$$

(положение точки  $P_2$  относительно отрезка  $L_2$ ).

Если направления сторон многоугольника упорядочены по часовой стрелке, то знак  $U_i$  в этих уравнениях будет отрица-

Рис	$U_1$	$U_2$	$U_3$	$U_4$	$Q_1$	$Q_2$
(а)	+	+	-	-	невидим	
(б)	+	-	=	-	$L \cap L_1$	$P_2$
(в)	-	+	-	-	$P_1$	$L \cap L_1$
(г)	+	-	-	+	$L \cap L_1$	$L \cap L_2$
(д)	-	+	+	-	$L \cap L_2$	$L \cap L_1$
(е)	-	-	-	+	$P_1$	$L \cap L_2$
(ж)	-	-	+	-	$L \cap L_2$	$P_2$
(з)	-	-	+	+	невидим	
(и)	-	-	-	-	$P_1$	$P_2$

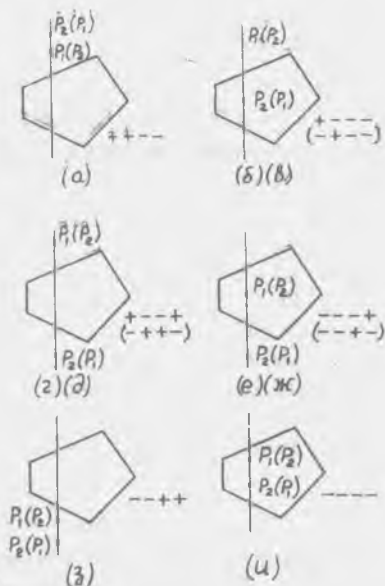


Рис. 34

тельными, когда точка лежит внутри многоугольника. На рис. 34 показаны различные положения точек  $P_1$  и  $P_2$  относительно двух отрезков, а в табл. 2—знаки величин  $U_i$  и схема вычисления концевых точек отображаемого отрезка. Данная таблица и алгоритм используют следующие обозначения:  $Q_1$  и  $Q_2$  являются концевыми точками отрезка, который должен быть вычерчен;  $L \cap L_i$  ( $i = 1, 2$ ) обозначает точку пересечения отрезков  $L$  и  $L_i$ .

Теоретически возможно 16 случаев сочетаний знаков 4-х переменных, но в силу геометрических ограничений имеем только 9 случаев. В самом деле, невозможно, чтобы все  $U_i$  были положительными, т. к. положительные полуплоскости для двух сторон являются внешними для многоугольника. Аналогично, невозможно получить 3 положительных знака. Остается 11 случаев. Конфигурация « $- + - +$ » означает, что  $P_1$  лежит внутри многоугольника, а  $P_2$  находится выше прямой  $A_j A_{j+1}$  и ниже прямой  $A_k A_{k+1}$ , что невозможно. Аналогичный вывод можно сделать для конфигурации « $+ - + -$ », что оставляет 9 случаев, показанных в табл. 2. Отметим, что с геометрической точки зрения имеется только 6 случаев, которые показаны на рис. 34. Если предположить, что  $P_1$  всегда ниже  $P_2$ , то можно ограничиться только 6-ю случаями, но желательно избегать такого предположения. В случае, когда  $P_1 P_2$  является стороной много-

угольника, порядок концевых точек не определяется их координатами.

#### 4.7. ОТСЕЧЕНИЕ ПРОИЗВОЛЬНОГО МНОГУГОЛЬНИКА ПРЯМОЙ ЛИНИЕЙ

Для того, чтобы отобразить многоугольник, не обязательно являющийся выпуклым, в прямоугольном окне, можно либо отсечь каждую из сторон многоугольника окном (см. разд. 4), либо отсечь весь многоугольник каждой из сторон окна. Здесь решается вторая задача. Алгоритм может также использоваться как основа для нахождения пересечения двух произвольных многоугольников.

Сначала для каждой пары соседних вершин многоугольника вычисляются величины  $U_1$  и  $U_2$ :

$$U_1 = \det (\bar{A}_i, \bar{P}_1, \bar{P}_2),$$

$$U_2 = \det (\bar{A}_{i+1}, \bar{P}_1, \bar{P}_2).$$

По мере продвижения от одной пары вершин к следующей новое значение  $U_1$  равно старому значению  $U_2$ , так что необходимо одно вычисление в каждой вершине. С учетом допущений, сделанных в п. 4.1, вершина считается видимой, если  $U_i$  положительно (рис. 35). Анализ знаков величин  $U_i$  в соседних вершинах многоугольника позволяет определить видимость его сторон. Возможные случаи перечислены в табл. 3.

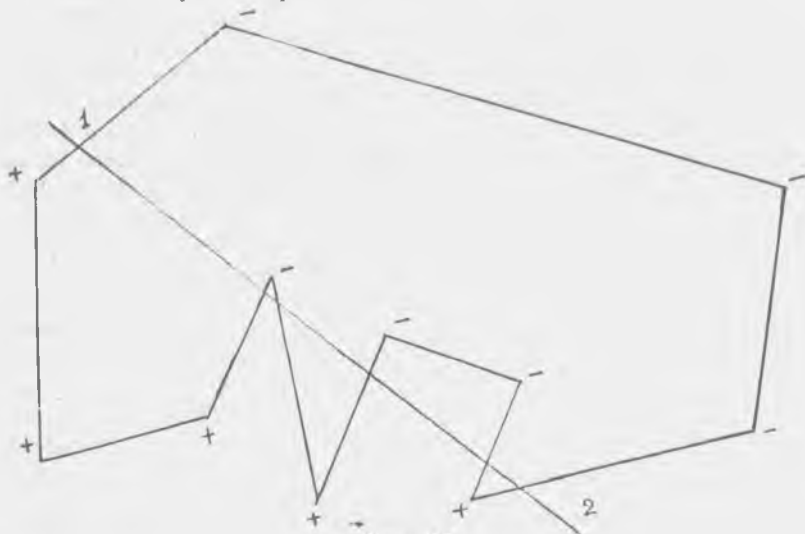


Рис. 35

$U_1$	$U_2$	Статус (видимость)
+	+	Весь видимый
+	0	Весь видимый
+	-	Частично видимый
0	+	Весь видимый
0	0	Невидимый
0	-	Частично видимый
-	+	Частично видимый
-	0	Невидимый
-	-	Невидимый

#### 4.8. ПЕРЕСЕЧЕНИЕ ДВУХ МНОГОУГОЛЬНИКОВ

Необходимость нахождения пересечения двух многоугольников часто возникает в машинной графике, например, при анализе видимости объектов. Один из методов решения данной задачи — повторное выполнение отсечения многоугольника прямой линией (см. п. 4.7). Пусть  $\pi_1$  и  $\pi_2$  — два заданных многоугольника,  $q_1, q_2, \dots, q_n$  — стороны  $\pi_2$ ;  $p_1, p_2, \dots, p_m$  — вершины многоугольника  $\pi_1$ . Тогда можно выполнить отсечение многоугольника  $\pi_1$  каждой из сторон  $q_i$ . Для ускорения нахож-

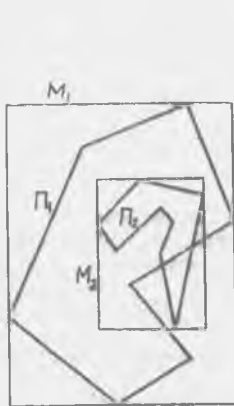


Рис. 36

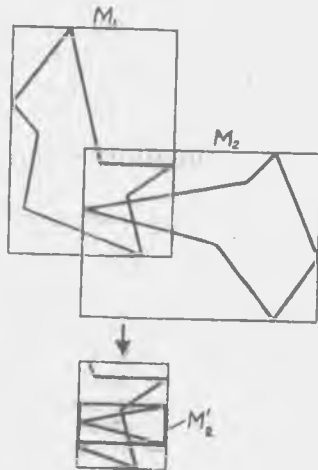


Рис. 37

дения пересечений можно использовать огибающие прямоугольники  $M_1$  и  $M_2$  вместо исходных многоугольников  $\pi_1$  и  $\pi_2$ . Если прямоугольники не пересекаются, то значит многоугольники также не пересекаются. Если один из прямоугольников содержится внутри другого, например,  $M_2$  внутри  $M_1$ , то можно рассмотреть пересечение  $M_2$  и  $\pi_1$ , а затем пересечение полученного многоугольника и  $\pi_2$  (рис. 36). Если  $M_1$  и  $M_2$  имеют непустое пересечение  $M_{12}$ , можно рассмотреть пересечение каждого многоугольника с прямоугольником  $M_{12}$ , затем определить два новых многоугольника и продолжить поиск пересечения для них (рис. 37). Такой подход имеет преимущество в том плане, что из рассмотрения быстро исключаются вершины, далеко отстоящие друг от друга.

### ВОПРОСЫ И УПРАЖНЕНИЯ

1. Определить знаки величин  $S_1, S_2, S_3, S_4$  для всех пар следующих отрезков:

A: (20, 220), (150, 220)

B: (35, 200), (130, 250)

C: (80, 210), (100, 200)

D: (110, 200), (110, 250)

E: (70, 170), (130, 200)

F: (80, 250), (150, 170)

G: (90, 170), (90, 250)

Какие отрезки пересекаются?

2. Определить координаты точек пересечения отрезков из задачи 1.

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- Автоматизированное проектирование. Геометрические и графические задачи / В. С. Полозов и др. — М.: Машиностроение, 1983. — 280 с.
- Басов Е. П., Абрамов В. Н. Графические регистрирующие устройства ЕС ЭВМ. — М.: Статистика, 1977. — 168 с.
- Баяковский Ю. М., Галактионов В. А., Михайлова Т. П. Графор. Графическое расширение фортрана. — М.: Наука, 1985.
- Верлер К. Обработка графической информации с помощью вычислительной техники. — М.: Машиностроение, 1979. — 254 с.
- Гилой В. Интерактивная машинная графика. — М.: Мир, 1981. — 384 с.
- Графические средства автоматизации проектирования РЭА. / Томашевский Д. Н. и др. — М.: Сов. радио, 1980. — 224 с.
- Елшин Ю. М. Автоматизированные рабочие места при проектировании РЭА. — М.: Радио и связь, 1983. — 128 с.



- Зозулевич Д. М.* Машинная графика в автоматизированном проектировании. — М.: Машиностроение, 1976. — 240 с.
- Кобелев В. В.* Машинная графика для системы БЭСМ — алгол. — М.: Наука, 1978. — 176 с.
- Котов И. И.* и др. Алгоритмы машинной графики. — М.: Машиностроение, 1977. — 232 с.
- Ньюмен У., Спрулл Р.* Основы интерактивной машинной графики. — М.: Мир, 1976. — 576 с.
- Принс М. Д.* Машинная графика и автоматизация проектирования. — М.: Сов. радио, 1975. — 232 с.
- Уокер Б. С.* и др. Интерактивная машинная графика. — М.: Мир, 1980.
- Фоллс Дж. ван Дэм А.* Основы интерактивной машинной графики. В 2-х т. Пер. с англ. — М.: Мир, 1985.
- Энджел Я.* Практическое введение в машинную графику. — М.: Радио и связь, 1984.

# Содержание

---

Введение . . . . .	3
1. ОБЩИЕ СВЕДЕНИЯ О МАШИННОЙ ГРАФИКЕ . . . . .	5
2. СРЕДСТВА МАШИННОЙ ГРАФИКИ . . . . .	13
2.1. Координатные системы . . . . .	13
2.2. Отсечение . . . . .	17
2.3. Технические средства машинной графики . . . . .	19
2.3.1. Структура и функции обобщенной системы машинной графики . . . . .	19
2.3.2. Общие сведения о технических средствах машинной графики . . . . .	22
2.3.3. Устройства ввода графической информации в ЭВМ . . . . .	25
2.3.4. Графопостроители . . . . .	28
2.3.5. Графический дисплей . . . . .	32
3. ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ . . . . .	40
3.1. Однородные координаты . . . . .	40
3.1.1. Определение однородных координат . . . . .	40
3.1.2. Использование однородных координат на плоскости . . . . .	41
3.1.3. Однородные координаты в трехмерном пространстве . . . . .	43
3.2. Геометрические преобразования . . . . .	44
3.2.1. Двумерные преобразования . . . . .	44
3.2.2. Трехмерные геометрические преобразования . . . . .	46
3.3. Перспективное изображение трехмерных объектов . . . . .	48
3.4. Использование однородных координат при выполнении преоб- разований . . . . .	49
4. ПОЗИЦИОННЫЕ ЗАДАЧИ МАШИННОЙ ГРАФИКИ . . . . .	54
4.1. Допущения и определения . . . . .	54
4.2. Положения точки относительно прямой . . . . .	54
4.3. Точка пересечения двух прямых . . . . .	55
4.4. Пересечение отрезков . . . . .	55
4.5. Положение точки относительно многоугольника . . . . .	57
4.6. Отсечение отрезка выпуклым многоугольником . . . . .	59
4.7. Отсечение произвольного многоугольника прямой линией . . . . .	62
4.8. Пересечение двух многоугольников . . . . .	63
БИБЛИОГРАФИЧЕСКИЙ СПИСОК . . . . .	64

Св. план 1986 г. поз. 139

*Олег Петрович Александров*

**ТЕХНИЧЕСКИЕ СРЕДСТВА И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ МАШИННОЙ ГРАФИКИ**

Редактор Е. Д. Антипова  
Техн. редактор Н. М. Каленюк  
Корректор В. П. Петрова

Сдано в набор 26.05.86 г. ЕО 00309.  
Подписано в печать 25.09.86 г. Формат 60×84 1/16.  
Бумага оберточная. Высокая печать.  
Литературная гарнитура.  
Усл. п. л. 4,0. Уч.-изд. л. 3,9. Т. 500 экз.  
Заказ 615. Цена 20 к.

Куйбышевский ордена Трудового Красного Знамени  
академический институт имени академика С. П. Королева,  
г. Куйбышев, ул. Молодогвардейская, 151.  
Типография ЭОЗ КуАИ, ул. Ульяновская, 18,