

IBM PC图形编程技术

中国科学院成都计算机应用研究所

TP31
1936

IBM PC 图形编程技术

Hyman michael 著

汪永宏 译

中国科学院成都计算机应用研究所情报室

1988年10月

引 言

当我在写图形程序时，总是想为自己开发一些子程序，如画线，三维透视，以及旋转等等。我想寻找一些比较好的计算机图形学书籍，但没有找到一本满意的。要么这些书太简单——只提到一些最基本的论题——和通常只用BASIC，或者是一些数学家为计算机科学所写的，只涉及用于大型计算机的疑难课题。

本书却不是这样。它涉及的论题很广泛，从线图、块图到随手画。由简单到复杂，从容易使用的BASIC到有效的汇编语言。当你是一个初学者时，你可以从中学到实现计算机图形的每个步骤，而当你欲向更复杂的论题进军时，你也可从中获得启迪。

第一章给出了使用IBMPC图形的基础。

第二章讨论二维图形，包括建立线形图的数据；二维图形绘制；平移，缩放和动画；以及用图形定义语言绘图。

在第二章的基础上，第三章详细介绍了三维图形，包括透视，旋转，矩阵，键盘交互，映射，剪切，可变透视和剪辑。

第四章深入研究汇编语言图形程序设计。从最基本的模式建立，清屏和直线绘制这些技术开始，逐步进入开发能以闪电速度运行的完整的三维图形系统。

第五章包括建立真实的三维图形所必需的知识，这中间有视觉生理学概述，立体透视，色彩过滤系统，以及打印三维映象。这里的程序都是同时用BASIC和汇编语言开发的。

第六章通过讨论目标的高级数据结构来表达线性图形。

第七章讨论高级图形技术，例如防止闪烁，有效地存贮和加载图形屏幕，以及直接编程IBM彩色图形适配器。

第八章处理块图。包括变换目标及块数据，显示和活动BASIC或汇编语言的块数字，旋转，缩放和隐藏等这些论题。

第九章告诉你怎样建立一个图形编辑器——用户交互式绘图程序。内容包括交互，建立绘笔，画线、方框，圆和椭圆，区域涂色，喷涂笔，剪贴以及建立HELP菜单。

第十章讨论在文本和单色屏幕上的线和块图。

使用本书

如果你是一个初学者，你应当从面向BASIC的章节开始——第一、二、三、五、六、八和九章。如果你碰到有些部分不太懂（例如，第八章可能有点困难），跳过去或只试用一下程序。然而当你觉得对图形程序设计比较自如时，再回过头来看那些比较难的部分。

如果你是一个中等水平的用户，可以考虑读一读汇编语言部分，以及第七章。

如果你是一个高级用户，你可能希望跳过BASIC部分，集中精力于汇编语言程序设计和高级技术，那么重要的部分是第四、五、六、七、八和十章。

当然，从一般意义上来说，最好还是通读全书。当你遇到不懂的部分，不用耽心——跳过它或试试程序。以后可以再回头来读。若你对汇编语言不感兴趣，那也无妨。以后你总是可以回过头来重读的。

所需设备

IBM PC或兼容机

DOS盘

大多数章节要求：

IBM彩色/图形控制器

彩色监视器

彩色监视器可以是电视，综合的或红绿兰（RGB）监视器。最好用RGB监视器，因为能给出明快清晰的图象，充分利用IBM分辨率。

有几种使用了IBM的宏汇编。

目 录

引言

第一章：图形准备	1
1.1 图形屏幕.....	1
1.2 使用图形屏幕.....	1
第二章：二维绘图	5
2.1 数据结构.....	5
2.2 建立点线数据库.....	17
2.3 画外形.....	19
2.4 平移.....	23
2.5 缩放.....	26
2.6 动画目标速度.....	28
2.7 色彩.....	29
2.8 图形定义语言.....	31
第三章：三维绘图	37
3.1 数据结构.....	37
3.2 平移.....	41
3.3 透视.....	41
3.4 缩放.....	46
3.5 旋转.....	46
3.6 矩阵.....	54
3.7 再访矩阵.....	61
3.8 键盘交互.....	65
3.9 旋转复杂性.....	69
3.10 映射.....	70
3.11 剪切.....	72

3.12	二维图形的矩阵.....	73
3.13	图形定义语言中的旋转.....	74
3.14	可变透视.....	74
3.15	剪辑.....	77
3.16	专用BASIC2.0命令.....	86
第四章: 汇编语言绘图.....		89
4.1	建立图形模式.....	89
4.2	画点.....	91
4.3	清屏.....	99
4.4	画直线.....	100
4.5	三维线图包.....	105
4.6	读取键盘命令.....	106
4.7	获取移动值.....	108
4.8	获取三角值.....	110
4.9	加载转换矩阵.....	113
4.10	点转换.....	114
4.11	考虑透视和点宽及转换成屏幕座标.....	118
4.12	画线.....	120
4.13	剪辑.....	122
4.14	扫尾.....	123
第五章: 真实三维图形.....		125
5.1	简述深度的觉察.....	125
5.2	做真实三维图象.....	126
5.3	彩色过滤.....	127
5.4	BASIC真实三维图象——彩色过滤方法.....	127
5.5	汇编语言程序中的真实三维图形.....	136
5.6	打印三维图象.....	142
5.7	使用偏光过滤系统.....	145
第六章: 多目标和结构化目标.....		148
6.1	多目标.....	148
6.2	汇编语言中的多目标.....	150
6.3	结构化目标.....	152
6.4	磁盘存贮和检索目标.....	153
第七章: 高级图形技术.....		154

7.1 防止闪烁	154
7.2 存贮和加载图形屏幕	155
7.3 彩色图形控制器程序设计	161
第八章: 块图	171
8.1 转换目标成块数据	171
8.2 BASIC的块图	173
8.3 PUT试验	180
8.4 活动	181
8.5 多目标	182
8.6 目标维护	183
8.7 旋转像素点矩阵	183
8.8 图形缩放	186
8.9 隐藏	188
8.10 加速动画	193
8.11 汇编语言的块图	194
第九章: 图形编辑器	203
9.1 基础	203
9.2 手画略图	205
9.3 色彩选择	210
9.4 改变色调板	210
9.5 改变背景色	211
9.6 清屏	211
9.7 画直线	212
9.8 画矩形	215
9.9 LINE的style 选择	216
9.10 圆和椭圆	218
9.11 涂色	221
9.12 书法笔	227
9.13 喷涂笔	229
9.14 剪贴	235
9.15 可用命令菜单显示	239
9.16 其他特点的建议	241
第十章: 文本屏幕上的图形	242
10.1 文本屏幕存贮映象	242
10.2 模式建立	243

10.3	清屏.....	246
10.4	线图.....	246
10.5	块图.....	249
10.6	多页.....	252
10.7	非标准文本模式.....	253
10.8	单色显示适配器的文本图形.....	253

附录

第一章 图形准备

本章告诉你怎样准备计算机显示图形。

1.1 图形屏幕

在图形屏幕上绘制计算机图，我们是用分辨率和能够显示的色彩值来描绘图形屏幕的。

分辨率是告诉我们出现在屏幕上象素的行列数。第一个数是行分辨率，第二个数是列分辨率。例如640×200分辨率意味着每行640个象素，每屏有200行——总共128,000个象素。

色彩值是每个象素可能显示的颜色号。屏幕的每个象素能是红、兰或黑的屏幕是一个三色屏幕。

IBM PC有四种显示屏幕——高分辨率图形屏幕，中分辨率图形屏幕，40列文本屏幕和80列文本屏幕。高分辨率图形屏幕有640×200个象素。每个象素可以是黑或白。中分辨率图形屏幕有320×200个象素，每个象素可取四种颜色。文本屏幕有40×25或80×25两种，可以有十六种颜色。我们几乎总是用图形屏幕。第七章和第十章我们才讨论文本屏幕。

1.2 使用图形屏幕

现在我们已经可以开始使用图形屏幕了。我们将使用BASIC，所以请将DOS盘插入A磁盘驱动器，并键入：

BASICA

现在我们将学习四个BASIC命令：SCREEN，COLOR，CLS和KEY OFF。在我们介绍命令的格式时，命令本身将用大写字母表示，而值（或者变量）将用小写表示，可选项将用方括号括起来表示。

SCREEN

现在开始。我们用SCREEN命令选择图形屏幕。其格式是：

SCREEN n

其中 n=2 是高分辨率屏幕

n=1 是中分辨率屏幕

n=0 返回到文本屏幕

让我们试一试。键入：

SCREEN 1

然后：

SCREEN 2

最后：

SCREEN 0

到这里，我们可以看到文本和两种图形屏幕之间唯一的差别是中分辨率图形屏幕使用大号文字，两种图形屏幕都没有闪烁光标。

COLOR

在中分辨率屏幕，我们可以选择背景色和色调板。我们用COLOR命令来选择：

COLOR [背景色] [, 色调板]

背景色是0到15之间的一个数字（见图1.1）。该数字指出选用什么颜色作为全屏的背景色（或称为底色）。

颜色表

0	黑	8	灰
1	兰	9	浅兰
2	绿	10	浅绿
3	青	11	浅青
4	红	12	浅红
5	紫	13	浅紫
6	棕	14	黄
7	白	15	高强度白

图1.1 十六种背景色

让我们先建立一个兰色背景：

SCREEN 1

COLOR 1

现在，使它变成

COLOR 9

下面再变成红色

COLOR 4

你或许想试试所有十六种背景色吧！

IBM 的色彩是由红、绿和兰三色组成的。这三种颜色可以组合成八种（ 2^3 ）基本颜色。每一种颜色可以高亮度显示，这就是颜色8到15（图1.2）。

颜色成分

0		6	绿+红	12	红+亮度
1	兰	7	兰+绿+红	13	兰+红+亮度
2	绿	8	亮度	14	绿+红+亮度
3	兰+绿	9	兰+亮度	15	兰+绿+红+亮度
4	红	10	绿+亮度		
5	兰+红	11	兰+绿+亮度		

图1.2 所有十六种颜色是红、兰、绿三色的组合，颜色8到15与颜色0到7具有相同的组成成分，只是高亮度显示。

在中分辨率下，每个象素可以是四色之一，这取决于所选择的色调板：

颜色	色调板 0	色调板 1
0	背景	背景
1	绿	青
2	红	洋红
3	棕	白

色调板是偶数时，总是用色调板 0，否则用色调板 1。色调板 1 所用的颜色与色调板 0 是一样的，只是加进了兰色。

等一会我们将作色调板的试验。

CLS 和 KEY OFF

在画图之前，我们需要有一个办法来清除文本屏幕或原有散布在屏幕上的图形。我们可以用 CLS 来做到这一点。试键入：

CLS

除了屏幕的最后一行显示功能键的功能而外，整个屏都清除干净了。若要消除这一行，可以键入：

KEY OFF

（要重新显示功能键的话，可以键入 KEY ON）

我们的第一个图形程序

我们将用下面的程序来测试这四个命令并显示 IBMPC 图形的特性：（不用担心 P SET 和 LINE 命令以及有关的数学计算；后面我们将学到。）

```
10 PI=3.14/180
20 KEY OFF
30 CLS
40 FOR BOX=1 TO 180 STEP 5
50   PSET (160 + BOX/2.5 * (COS (BOX*PI) + SIN (BOX*PI)) ,
        100-BOX/2.5 * (-SIN (BOX*PI) + COS (BOX*PI)) )
60   LINE- (160 + BOX/2.5 * (-COS (BOX*PI) +
        SIN (BOX*PI)) , 100-BOX/2.5 * (SIN (BOX*PI) *
        COS (BOX*PI)) ) , BOX MOD 3 + 1
70   LINE- (160-BOX/2.5 * (COS (BOX*PI) +
        SIN (BOX*PI)) , 100+BOX/2.5 * (-SIN (BOX*PI) +
        COS (BOX*PI)) ) , BOX MOD 3 + 1
80   LINE- (160+BOX/2.5 * (COS (BOX*PI) -
        SIN (BOX*PI)) , 100+BOX/2.5 * (SIN (BOX*PI) +
        COS (BOX*PI)) ) , BOX MOD 3 + 1
90   LINE- (160+BOX/2.5 * (COS (BOX*PI) +
        SIN (BOX*PI)) , 100-BOX/2.5 * (-SIN (BOX*PI) +
```

```
      COS (BOX * PI))) , BOX MOD 3 + 1  
100  NEXT  
110  END
```

现在我们在高分辨率屏幕下运行这个程序：

```
SCREEN 2  
RUN
```

屏幕上出现了一幅很漂亮的图案。所有的线是白色或黑色的。（只有画在白线上的黑线才能看到。）注意图形的水平宽度。

如果程序有语法错误或只画出几个点，请键入LIST，和本书上的程序作比较检查，纠正你的程序。你可能需要参照BASIC手册第二章。

现在我们在中分辨率下运行这个程序：

```
SCREEN 1  
RUN
```

注意出现了三个颜色（第四个是背景色）。也请注意图形的大小发生了变化。

现在，我们改变背景色。键入：

```
COLOR 1
```

并改变色调板：

```
COLOR , 0
```

注意，如果没有键入背景色，仍保留原有的背景色。

让我们看看在黑色背景下色调板0的颜色：

```
COLOR 0
```

你可能想重新运行该程序。在进入下一章之前，请键入NEW。

小结

现在我们可以结束对IBMPC图形的介绍了。我们学习了有关两个图形屏幕；看到了怎样作图形使用准备以及运行了我们的第一个图形程序。

第二章 二维线图

第一类要学的图形是线图——目标都是由点和线组成的（图2.1）。

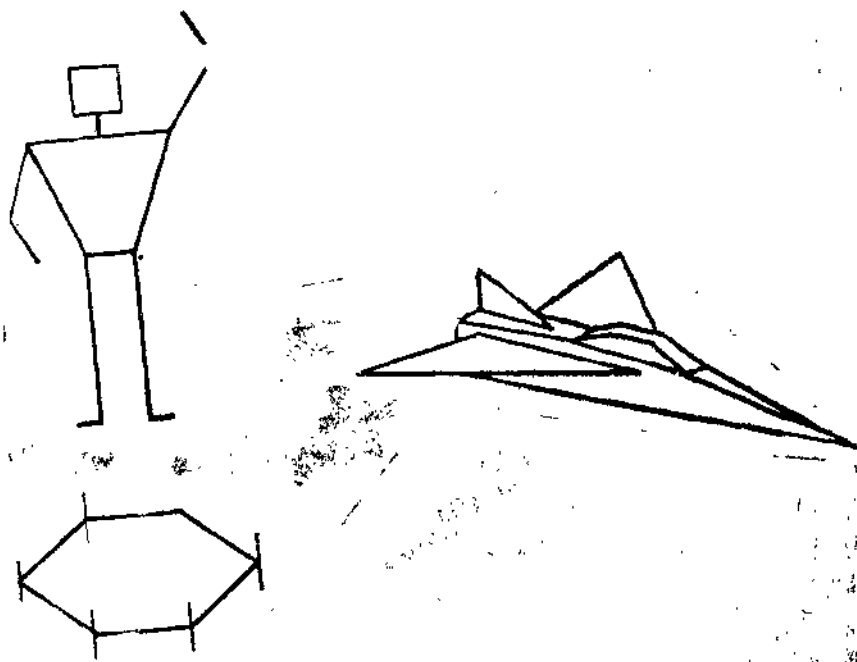


图2.1 简单的线图：站立的人，喷气机和葡萄分子。

我们从最简单的一类图：二维线图开始。在二维线图中，目标（例如正方形和站立的人）只有高和宽。本章将学习怎样以计算机能够接受的形式来描述二维线图，然后显示并使这些图形活动起来。

本章的程序

本章的许多节都有一些小程序。大多数是用来组合成一个长程序——一个越来越复杂的程序。你可以把每节的程序作为ASCII文件保留起来（即用命令SAVE“文件名”，A），并合并（MERGE）成一个主程序。

2.1 数据结构

在计算机上画目标之前，我们需要想出一个办法让计算机知道目标是什么。对二维线

图，我们仅需要有办法来表示直线。用卡得逊坐标系统就很容易办到。

卡得逊坐标系统提出了一种表示点的方法；直线仅仅是两点的连接。在卡得逊系统里，“世界”（亦即是我们绘图的区域）是划分成网格的，就象城市的街道地图一样。这个网格有2个轴：x轴是水平方向的，Y轴是垂直方向的。两轴的交点称作原点，用一对有序数 $(0, 0)$ 来表示。图2.2是一个卡得逊网格。

每个点用两个坐标表示——x坐标和y坐标——每个坐标值表示沿着该方向，该点与原点的距离。例如，原点右边3个单位和原点上边4个单位的这一点写作为 $(3, 4)$ 。同理，原点右边两个单位和原点下边2个单位的这一点写作 $(2, -2)$ 。总是先写x坐标，然后再写y坐标。图2.3就显示了卡氏网格上的点 $(3, 4)$ 和 $(2, -2)$ 。

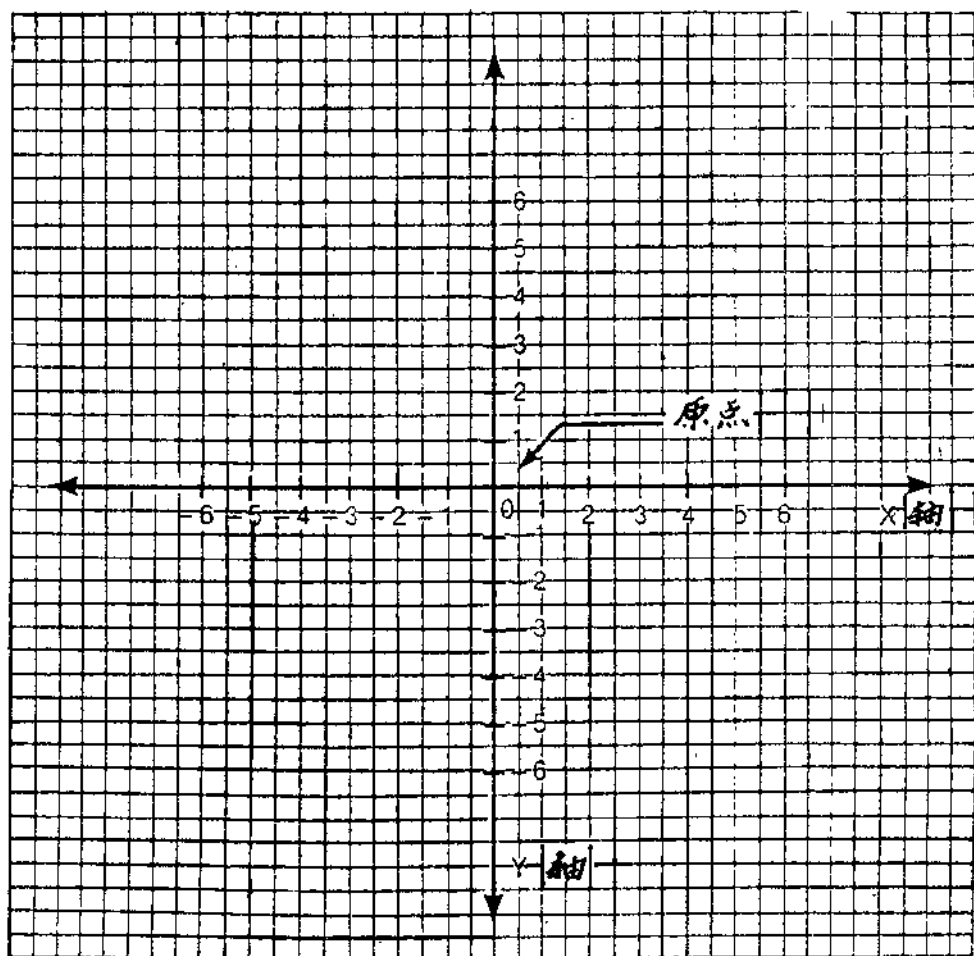


图2.2 卡得逊坐标系统

正如前面提到的，直线是两点间的连接。因此，要表示一条直线只需告诉计算机这条直线的两个端点。图2.4就显示了一条直线。注意这两个端点是用粗黑园点表示的。一点的座标是 $(3, 4)$ ，另一点座标是 $(2, -2)$ 。让我们称前一点为点1，后一点为点2。要画这条直线，我们要告诉计算机连接点1到点2（或者说连接点2到点1，这没有什么关系。）这是很容易做的，我们将在2.3节中看到。

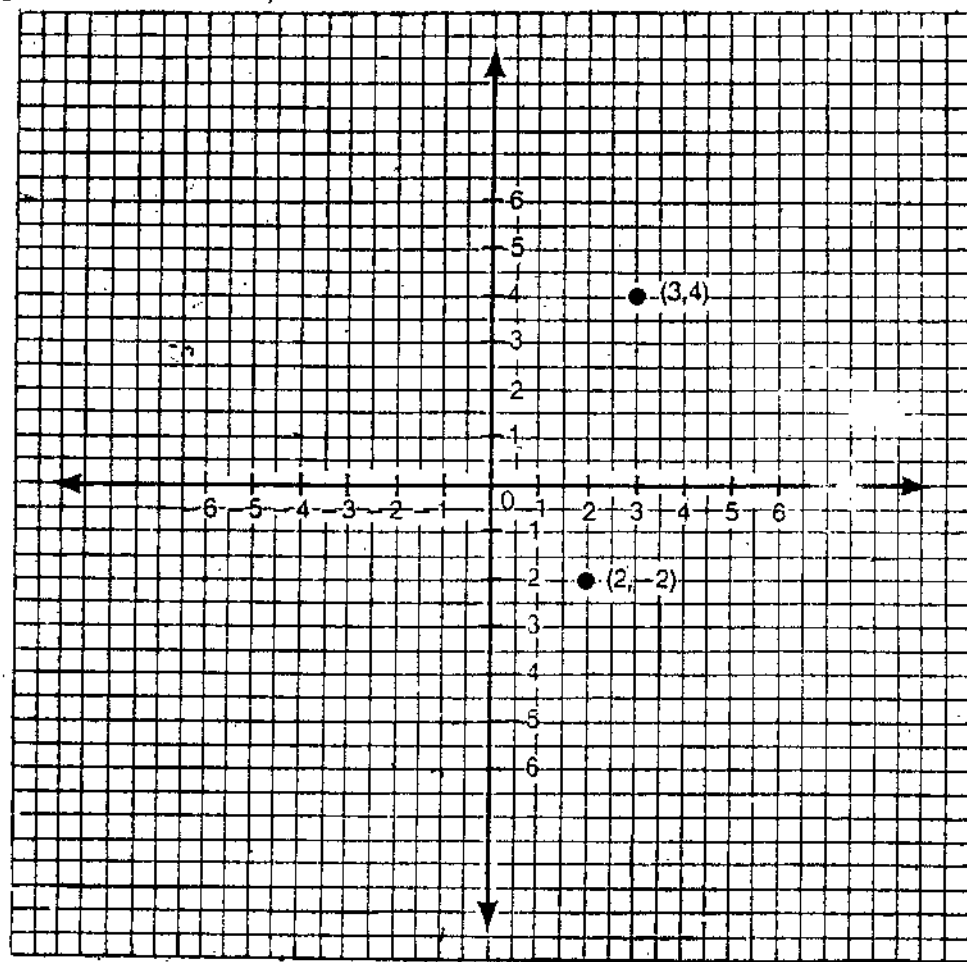


图2.3 点 $(3,4)$ 和 $(2, -2)$

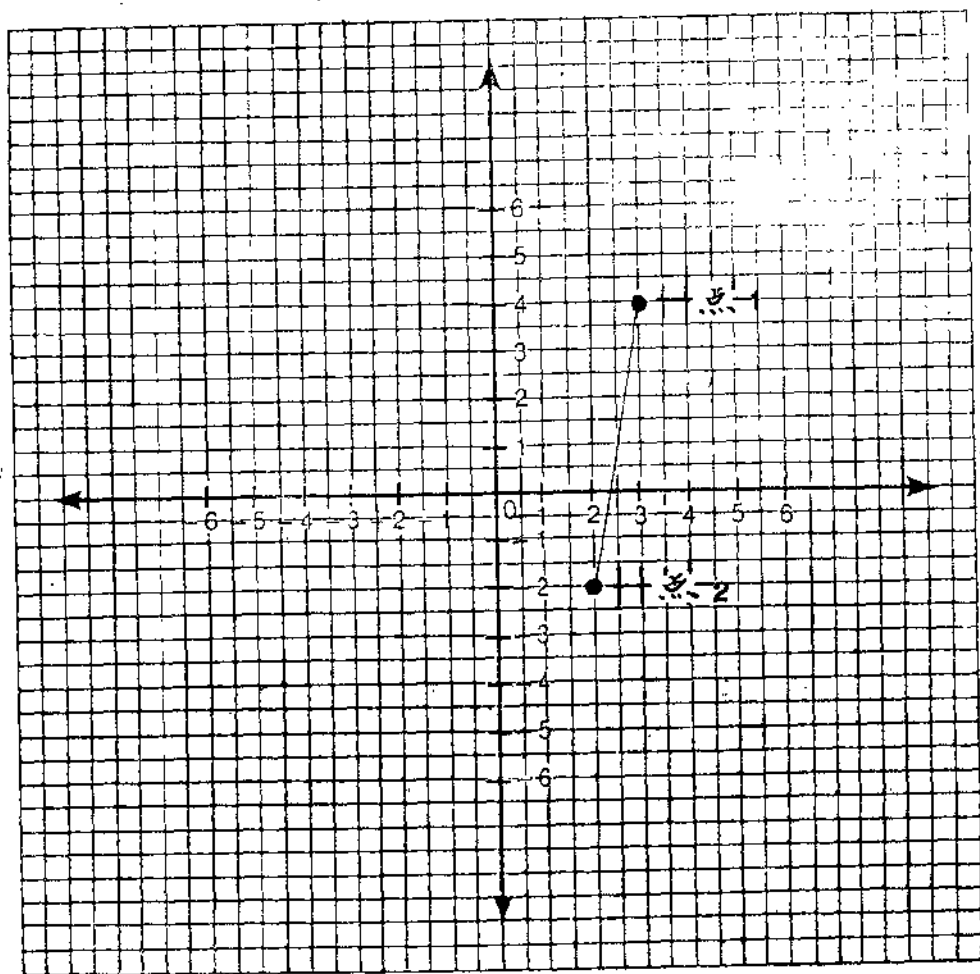


图2 4 直线

变换目标数据

现在我们需要设法把目标变换成计算机能识别的数据。我们已经知道计算机画直线是很容易的了，因此我们可以把一个图形分解成一组直线，然后把描绘这些线的数据告诉计算机。因为我们是线来表示目标，而线又是由点来表示的，因此我们称这个过程为建立点线数据。

让我们从一个简单的例子开始。图2.5是一个宇宙飞船的草图。首先在卡得逊坐标系里画好目标。一个比较好的办法是在座标纸上画（图2.6）。现在，计算出每个端点的座标（图2.7）。

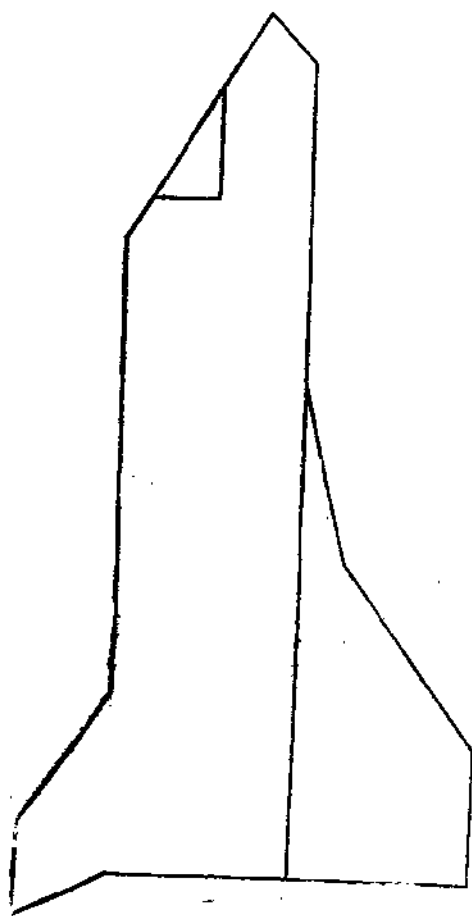


图2.5 宇宙飞船的草图