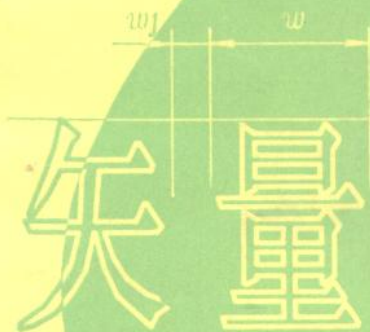


舒代宁等编著

Auto CAD
汉字包

计算机矢量汉字 的制作与应用

矢量



汉字



清华大学出版社

计算机矢量汉字的制作与应用

舒代宁 等 编著

科学出版社

1990

内 容 简 介

本书介绍用软件在工程图纸上书写汉字的方法,包括三部分内容:第一部分介绍矢量汉字的定义方法,包括子图形法、简单编码法及定义字符的编码系统;第二部分介绍矢量汉字库的建库原则与方法,在工程图纸上书写少量汉字、技术术语和常用词组以及书写技术说明的方法;第三部分介绍了矢量字符与矢量汉字的制作方法,即字模法、转换法与组合法。本书阐述的原理和方法适合于各种图形系统或 CAD 系统,是为从事计算机辅助设计与计算机绘图的工程技术人员编写的,适合于工程技术人员和计算机操作员使用,并可供工科院校师生参考。

计算机矢量汉字的制作与应用

舒代宁 等 编著

责任编辑 唐云江

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100707

国防科工委印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1990 年 4 月第一版 开本: 787×1092 1/32

1990 年 4 月第一次印刷(深圳) 印张: 3

印数: 0001—2 600 字数: 63 000

ISBN 7-03-001813-3/TP·133

定价: 2.50 元

前 言

我们在研制 Auto CAD 开发系统的第一阶段,着重研制了汉字软件包,包括建立 Auto CAD 的矢量汉字库,编写在图中标注少量汉字的标注程序,以及在图中书写文字较多的技术说明的书写程序,等等。这本小册子就是我们研制此汉字软件包的技术总结。

从事这项研究的,绝大多数都是学土木建筑工程或机械工程的。根据我们的实践经验,从事计算机绘图和计算机辅助设计的土木建筑工程师和机械工程师等工程技术人员,都应具备用软件定义矢量字符和矢量汉字的基本知识。因此,我们决定将这项技术加以总结,整理成册出版,供从事计算机绘图和计算机辅助设计的工程技术人员参考。

本书第九章还专门论述定义矢量汉字和矢量汉字库的标准化问题,提出了有关建议,希望能引起有关部门的重视,也希望从事计算机绘图和计算机辅助设计的广大工程技术人员积极支持矢量汉字的标准化工作,从而早日实现矢量汉字的标准化。

参加研制此汉字软件包的同志有:王道堂、应达之、吴旭、关森、钱春香、张晓川、罗毅和舒代宁等同志。本书由舒代宁同志执笔,缪兆杰、秋绍京、俞穗丰、布占民、张大庆和郝智琪等同志校阅书稿。

为了方便读者学习本书所讲述的内容,我们为读者配制

了一片 5 英寸、360K 的磁盘。该盘所录的程序是用 IBM PC BASIC 语言编写的,可用于 IBM PC XT/AT,具体内容请参看书末的磁盘文件清单。该磁盘由下列单位销售:

冶金部建筑研究总院计算机室

信通集团公司软件工程公司

北方交通大学土建系计算机应用学科组

冶金部建筑研究总院计算机室

北方交通大学土建系计算机应用学科组

1989 年 12 月

目 录

前 言		(i)
第一章	字符子图形法	(1)
第二章	简单编码法	(3)
第三章	定义字符的编码系统	(26)
第四章	矢量汉字	(40)
第五章	用拼音法在图中标注 少量汉字的方法	(53)
第六章	在图中标注常用汉字词组 和技术术语的方法	(57)
第七章	在图中书写汉字字数较多的 技术说明的方法	(63)
第八章	建立矢量字符库的转换法与组合法	(67)
第九章	矢量汉字的标准化的	(87)
附 录	磁盘文件清单	(89)

第一章 字符子图形法

标注字符与书写技术要求或技术说明，是绘制工程图样中的一项重要作业。我们把数字、字母、汉字、标点符号以及工程图中的常用符号统称为字符。电器图中用的图形符号、机械图中的表面粗糙度符号、形位公差符号、土建图中的常用图例，所有这些都是字符。本书专门介绍用软件产生矢量字符的方法，可以用来解决计算机绘图时在工程图上标注字符的问题，特别是汉字书写的问题。本章介绍字符子图形法。

我们写字时是按字的笔画落笔书写或抬笔移动，这与计算机绘图时的情况没有两样。这样，我们就可以把每个字符看成一个简单的子图形，即编制一个子程序。我们可以将字符各笔画端点的坐标和表示落笔书写或抬笔移动的数字信息放在 DATA 语句中。下面是书写汉字“啊”的子程序（即磁盘中的 ZFZTX. BAS）：

```
10 'file name : zfztx. bas
20 CLS : KEY OFF
30 INPUT "input height of character h ( $1 \leq h \leq 10$ ) :"; H
40 SCREEN 2
50 WINDOW (0, 0) — (640, 1000)
60 READ I, X, Y : X = .5 * H * X : Y = H * Y
70 PRESET (X, Y)
```

```

80 READ I, X, Y : X=. 5 * H * X : Y=H * Y
90 IF I=0 THEN 130
100 IF I=1 THEN LINE -STEP (X, Y) : GOTO 80
110 PRESET STEP (X, Y)
120 GOTO 80
130 LOCATE 1, 1 : PRINT "press any key to end. ";
140 IF INKEY $ = "" THEN 140
150 SCREEN 0 : END
160 '-----
170 ' (1, 2 or 0), x, y .    x, y -- relative coordinate
180 ' 1 : pen down; 2 : pen up; 0 : end of character
190 DATA 2, 4, 90, 1, 4, -6, 1, 0, -57, 2, 0, 58, 1,
      11, 0, 2, 1, 3, 1, 0, -60, 2, -11, 11
200 DATA 1, 10, 0, 2, 10, 52, 1, 3, -3, 1, 0, -82,
      2, 1, 82, 1, 13, 0, 2, 0, 1
210 DATA 1, -9, -30, 1, 3, -5, 1, 2, -5, 1, 2, -8,
      1, -1, -5, 1, -2, -6, 1, -6, 1
220 DATA 2, 17, 52, 1, 32, 0, 1, -2, 4, 1, -4, -4,
      2, -6, -1, 1, 0, -78, 1, -10, 6
230 DATA 2, -11, 59, 1, 0, -35, 2, 1, 32, 1, 10, 0,
      2, 0, 3, 1, 0, -35, 2, -12, 5, 1, 11, 0
240 DATA 0, 0, 0

```

这种方法十分简单，但要占用较多的磁盘空间和内存空间，与其它的方法比较，效果最差。用这种方法来书写汉字则更不现实。编写上千个子程序，不仅费时费事，还要占用大量的磁盘空间和内存空间。

第二章 简单编码法

上一章介绍的字符子图形法的字符子程序包括两个部分，一部分是书写字符程序，另一部分是字符笔画端点的坐标和抬落笔的数字信息。我们可以把这两部分分开，分成书写字符的通用程序与定义字符的数字信息库，后者将字符笔画的数字信息按事先定义好的编码和结构存储起来，供书写字符的程序调用。书写字符时，运行书写字符的程序，程序先读取这些信息，然后按照这些数字信息控制绘图仪，或落笔书写，或抬笔移动。这样便可完成在图样上书写字符的作业。下面具体讨论用字符简单编码法完成书写字符作业中的几个具体问题。

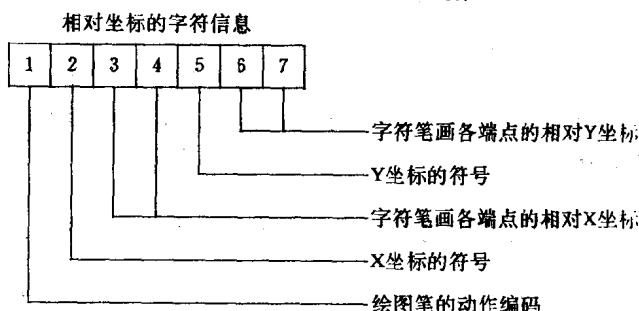
1. 字符数字信息的编码、数据结构和存储结构

工程技术人员在工程图样中书写字符时，为了使字符书写得工整，先要用硬铅笔轻轻地画一些方格，把字符书写在方格之中。书写字符时，先把笔移到一个恰当的位置，下笔书写字符的第一笔画，然后移笔再书写，这样反复动作直到书写完一个字符。根据用手书写字符的情况和便于书写程序处理字符数字信息的要求，对字符数字信息作如下规定：

把书写字符方格的左下角确定为字符的基点。基点有两个含义：一是作为字符笔画各点的坐标原点，二是作为该字符在工程图样中书写的起始点。

字符笔画各端点的坐标可定义为相对坐标，也可定义为

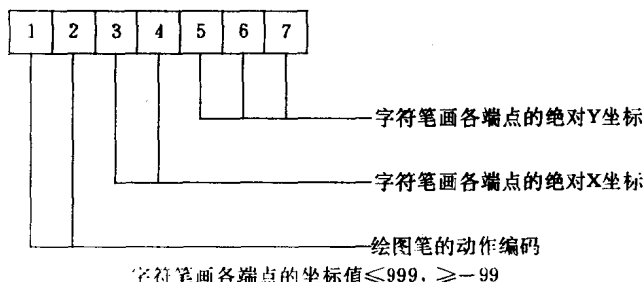
绝对坐标，坐标的数值为整型数。移笔和书写字符笔画均是从绘图笔的当前点到字符数字信息坐标指定的下一点。刚开始书写时，绘图笔是在字符的基点处。书写完一个字符后，绘图笔要移到该字符方格的右下角。至于定义坐标，使用者应该比较两者的差别，研究在何种情况应定义为相对坐标，在何种情况应定义为绝对坐标。定义字符时，选用一种坐标形式较好。也可以将定义字符抬笔移的坐标定义为绝对坐标，而将定义字符落笔画的坐标定义为相对坐标。我们规定字符数字信息的长度为七个字节，每个字节的意义见附图。



字符笔画各端点的相对坐标的绝对值 ≤ 99 。因为相对坐标等于前一点的坐标减当前点的坐标，故经常出现负值。为了对称，不论是正还是负都应带有符号。

我们规定绘图笔作抬笔动作的编码为“2”，落笔编码为“1”，字符的数字信息的结束码为“0”，即“0”为数据的结束符。为了方便检索，我们把这样的七个字节的数字信息作为一个记录，按顺序数据文件的形式存储在磁盘中，一个文件只存一个字符的数字信息，并用其所定义的字符或该字符

绝对坐标的字符信息



字符笔画各端点的坐标值 ≤ 999 , ≥ -99

的 ASCII 码作为文件名, 扩展名为 .DAT。下面是按上述定义字符编码的规则定义的字符“1”:

1 . DAT

2+01+25

1+04+05

1+00-30

2-04+00

1+08+00

0

编码中的坐标为相对坐标, 它存储在配合此书的磁盘的 1 . DAT 文件里。读者可按上面所述的规定, 一边读一边将文件所定义的字符画在坐标纸上, 这样可以帮助读者理解和熟悉上面所述的规定。

2. 字符书写程序

字符书写程序的主要部分, 是把从文件读出的信息分解为绘图笔动作编码和绘图笔移动到下一个点的坐标。分解按上述数据结构与存储结构存储的字符信息是十分容易的, 其分解方法有两种:

第一种分解方法是一个字节一个字节地从数据文件中读取字符数字信息，在读的过程中分解。将读入的第一个字节数据作为绘图笔的动作编码，将读入的第二、三、四三个字节的数据和第五、六、七三个字节的数据分别组合在一起，作为绘图笔移到下一个点的相对坐标或绝对坐标。磁盘中的 SXZF .BAS 程序是用这种处理方法编写的字符书写程序。

第二种分解方法是一次从数据文件中读取数据的一个记录，然后按上述的字符数字信息的结构将其分解为绘图笔动作编码、绘图笔移到下一点的相对坐标（或绝对坐标）X 和 Y 三部分。磁盘中的 SXZFC .BAS 程序是用这种处理方法编写的字符书写程序。

磁盘中的 SXZFC .BAS 程序是书写字符串的程序。该程序主要分读取字符的数据文件的检索和确定字符之间的间距两部分。因为我们规定一个文件只存储一个字符的数字信息，且以该字符或该字符的 ASCII 码为文件名，这样便使检索问题变得十分简单。读者只要阅读该程序的检索部分，即可了解程序中所用的检索方法。

这里，我们要讨论书写字符串时怎样使字符之间留有间距。前面介绍字符数字信息结构时，规定书写完一个字符之后，将绘图笔移到书写字符方格的右下角。因此在书写下一个字符之前，要将绘图笔向右移动一个字符间距，这样，程序可以根据标注字符的具体情况或用户的要求，确定字符之间的间距。SXZFC .BAS 程序是用这种方法处理的。另一种确定字符间距的方法是在字符数字信息结构中，选定一个固定的字符间距，规定书写完一个字符之后，把绘图笔向右移动到下一个字符的起点处。用这种方法书写字符串时，字符之

间的间距是固定的。

读者可以运行和阅读上面介绍的两个程序，进一步学习上面所述的字符书写程序的编程方法。运行这些程序时，读者可以输入磁盘中已有的定义字符的数据文件，这些文件的文件名为 0, 1, 2, ..., 9，文件名即为其所定义的字符；文件的扩展名为 .DAT。建议读者按上面介绍的字符数字信息的数据结构和存储结构，自己定义一两个字符，用上面介绍的程序将它们书写出来。读者可将要定义的字符放大，写在坐标纸上，根据坐标纸上的坐标网格线即可获得字符笔画端点的坐标，用文本编辑软件按上面的规定输入相应的数据，即可建立所定义字符的数据文件。

然而，上面介绍的方法有下列缺点：

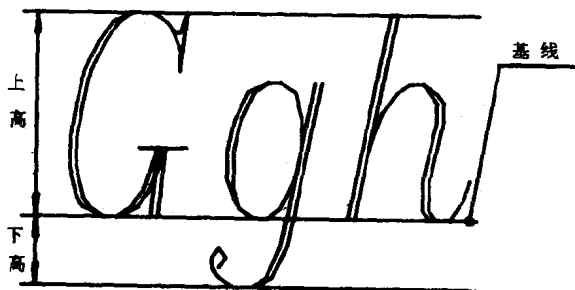
- (1) 没有定义字符的高度；
- (2) 一个文件只存储一个字符数字信息的数据；
- (3) 没有介绍生成字符数字信息数据的方法。

下面将讨论改进这些缺点的方法。

3. 定义字符高度的意义

在工程图纸中书写字符时，总要根据图样的情况选取恰当的字高，如字高为 5mm。如果没有定义字符高度，则在书写字符时很难确定字高的比例因子，即用多大的数乘字符的坐标，使写出的字符高度正好等于 5mm。如果定义了字符高度，则这个问题就很容易解决了。在定义字符高度之前，先介绍一下字符高度的术语。字符高度的术语见附图。从该图可以知道，定义字符的高度时，既要定义其上高，也要定义其下高。字符的下高与字符的行间距有关，因为字符的基点

是在字符的基线之上。



为了定义字符高度，在字符定义文件的开始处增加下列两个记录：

第一个记录 r (相对坐标) 或 a (绝对坐标)；

第二个记录 字符的上高，字符的下高。

例如在 1.DAT 文件中增加上述两个记录后，该文件变为：

```
r
30, 0
2+01+25
1+04+05
1+00-30
2-04+00
1+08+00
0
```

在书写字符时，如要求字符高度为 h ，则在书写程序中应该用下列因子乘字符笔画坐标：

X 坐标的因子： $h \times \text{字宽系数} \div \text{字符的上高}$

Y 坐标的因子: $h \div \text{字符的上高}$

字宽系数 = 字符宽 $\div$ 字符的上高, 一般情况字宽系数 = .7。

经上述处理之后写出的字符, 其高度就等于要求的高度 h 。如果将字符的上高改大而不改动字符的坐标, 则写出的字符高度就小于要求的高度 h 。反之, 则写出的字符高度就大于要求的高度 h 。

增加的第一个记录是定义字符所采用坐标方式的标识符, 是为书写程序而设的。

4. 字符数字信息数据结构和存储结构的改进

如果在定义每一字符数字信息之前增加一个记录, 便可将所定义的诸字符存入一个顺序文件中。其格式如下:

* 定义字符的 ASCII 码, 定义字符码的记录个数 (包括结束码 0)

例如在 1. DAT 文件中增加了该记录之后, 1. DAT 文件就变为:

```
r
30, 0
* 49, 6
2+01+25
1+04+05
1+00-30
2-04+00
1+08+00
0
```

设置该记录的目的是为了便于检索。检索程序如下:

```

1000 IF EOF (1) THEN END
1010 LINE INPUT #1, A$
1020 IF LEFT$ (A$, 1) <>"*" THEN 1000
1030 L=INSTR (A$,"") : A1$=LEFT$ (A$, L-1)
1040 A1$=RIGHT$ (A1$, L-2) :
      A1$=CHR$ (VAL (A1$))
1050 IF A1$=C$ (J) THEN
      RECORD=VAL(RIGHT$ (A$, LEN(A$)-L)) :
      RETURN
1060 GOTO 1000

```

1050 返回后便可转入书写子程序。

这种检索方法检索时要从文件的头部依次读取每一个记录，不很方便。磁盘中的 JPZF.DAT (相对坐标) 与 JPZFA.DAT (绝对坐标) 两个定义字符的文件具有这种结构。该文件定义了包括空格键在内的键盘上的 95 个字符，即 ASCII 码 32—126 的 95 个字符，下面是 JPZF.DAT 中的一部分 (数据应按列阅读)：

r	1-05-10	2+20-24	1-70+00
100, 35	1+05-56	1+06+05	2+16-36
* 32, 3	1+00+66	1+04+09	1+24+94
2+50+00	2+50-50	1+00+10	2+20+00
2+50+00	2+00-50	1-05+05	1-24-94
0	0	1-05-05	2+51+48
* 33, 14	* 34, 19	1+05-05	2+00-50

2+50+00	2+30+81	1+05+05	0
1+05+05	1+06+05	2+40-55	* 36, 42
1-05+05	1+04+09	2+00-50	2+22+14
1-05-05	1+00+10	0	1+04+05
1+05-05	1-05+05	* 35, 11	1-05+04
2+00+33	1-05-05	2+19+59	1-04-04
1+05+57	1+05-05	1+70+00	1+00-05
1-05+10	1+05+05	2-06-21	...

磁盘中的 SXZFC1. BAS 程序是用这种数据结构与存储结构定义的矢量字符的书写程序。读者可以运行与阅读该程序,了解程序中所采用的检索方法。字符的 ASCII 码越大,则检索的时间就越长,这是因为字符的 ASCII 码越大,其存储在文件中的位置也就越靠后,这就是这种结构的缺点。运行该程序时,屏幕上将依次出现下面的提示:

which string do you want to write ?	输入要书写的字符
where X, Y ?	输入字符的起点坐标
rotation angle (in degree) :	输入字符的书写角度
input height and width factor of character ;	输入字符的高度与宽度系数
input distance between characters ;	输入字符间的间距
input file name of character data (without . ext) ;	输入JPZF 或 JPZFA,不输扩展名

输完上面的诸数据后,程序将依次在屏幕上显示要书写的字符。

如果将所定义的诸字符存入一个用 BASIC 语言定义的