

·光电系统与amp;设计·

DXF 文件在激光打标系统的应用研究

张国庆

(武汉科技大学, 冶金自动化与检测技术教育部工程研究中心, 湖北 武汉 430081)

摘要:对于速度和重构性要求比较高的激光打标系统来说, 一种有效易用的文件格式将起到举足轻重的作用. 分析了 DXF(drawing exchange format)文件的结构及 DXF 图形交换文件在基于数字信号处理的激光打标系统中的应用, 详述了 DXF 文件的解析方法, 指出了 DXF 文件在此系统中的优势之处. 利用数据交换接口对 DXF 文件进行二次开发, 使得系统获得了比较好的图形处理效果.

关键词:图形交换文件; 面向对象; 激光打标

中图分类号: TP274⁺.2

文献标识码: A

文章编号: 1673-1255(2010)05-0008-04

Application Research of DXF File in Laser Marking System

ZHANG Guo-qing

(Automation and Detection Technology Metallurgical Engineering Research Center of the Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: For high requirement of speed and reconstruction for laser marking system, an effective and easy use file format will play a crucial role. The DXF (drawing exchange format) document structure and the DXF's application in laser marking system based on DSP (digital signal processing) are introduced. The method of analyzing DXF file is stated in details, and the advantages of DXF file in this system will be pointed out. Using the data exchange interface for DXF's secondary development can also make the system to achieve better graphics processing effects.

Key words: graphical exchange files; object oriented; laser marking

目前, 在打标流水线上的软件基本上都是基于 PC 平台的打标系统. 而新一代激光打标系统正在向体积小、效率高及较高的性价比方向发展. 利用具有强大的图形图像处理及分析设计功能的 AutoCAD 实现矢量图形的重构, 可使激光打标系统的性能得到较大的提升. DXF 是由 Autodesk 公司提供的与当前图形完全等价的通用 ASCII 码格式文件, 作为 AutoCAD 的数据接口的 DXF 接口文件已经成为一种标准接口格式. 由于其简单、通用、易读, 能够很有效地实现 AutoCAD 与嵌入式系统的连接, 便于高级编程语言进行编程, 可提高工程开发效率.

1 DXF 文件简介

DXF 格式是 AutoCAD 图形文件中包含的所有信息的一种带标记数据的表示方式. 带标记数据是指文件中的每个数据元素前面都带有一个称为组码的整数. 组码的值表明了随后的数据元素的类型. 还指出了数据元素对于给定对象(或记录)类型的含义. 完整的 DXF 文件由 7 个段组成, 每段都以一个后跟字符串 SECTION 的组码 0 开始, 其后是组码 2 和表示该段名称的字符串. 每段都由定义其元素的组码和值组成, 都以一个后跟字符串 ENDSEC 的组码 0 结束.

收稿日期: 2010-09-26

基金项目: 国家自然科学基金(60705035); 武汉市晨光计划(200950431200); 湖北省高校产学研合作资助重点项目(CXY2009B006); 湖北省教育厅科研项目优秀中青年项目(Q20091103); 武汉科技大学大学生创新基金

作者简介: 张国庆(1989-), 男, 湖北宜都人, 学士. 主要研究方向为光机电系统.

DXF 文件重要的结构如表 1.

表 1 DXF 文件结构

段名	定 义
HEADER 段	包含图形的基本信息. 它由 AutoCAD 数据库版本号和一些系统变量组成. 每个参数都包含一个变量名称及其关联的值.
CLASSES 段	包含应用程序定义的类的信息, 这些类的实例出现在数据库的 BLOCKS、ENTITIES 和 OBJECTS 段中. 类定义在类的层次结构中是固定不变的.
TABLES 段	包含以下符号表的定义: APPID(应用程序标识表)、BLOCK_RECORD(块参照表)、DIMSTYLE(标注样式表)、LAYER(图层表)、LTYPE(线型表)、STYLE(文字样式表)、UCS(用户坐标系表)、VIEW(视图表)、VPOR(视口配置表)
BLOCKS 段	包含构成图形中每个块参照的块定义和图形图元.
ENTITIES 段	包含图形中的图形对象(图元), 其中包括块参照(插入图元).

2 DXF 在激光打标系统中的应用

2.1 GDI+ 基本介绍

GDI+ 是 GDI(图形设备接口)的改进版, 是一个负责操作系统与绘图程序之间信息交换的高级应用程序编程接口. GDI+ 利用一个 Graphics 对象将屏幕的某一个窗口与之相关联, 并包含绘图操作所需要的相关属性.

在程序的设计中, 涉及到的 GDI+ 的相关类主要有以下 2 个:

一是图形路径类 GraphicsPath 类. 表示一系列相互连接的直线和曲线, 它提供了很多方法供用户使用, 用户只需把绘图对象作为一个参数传递给图形对象的方法调用即可, 使作图变得方便高效. 如果要得到一个组合的图元, 只需要把单个的图元作为子路径分别添加到当前绘图路径中形成一个总体的路径即可. GraphicsPath 类中除了添加图元路径的方法外, 还有一个很重要的方法是 transform (&matrix), 其作用是将一个变换矩阵应用到该路径中进行变换.

二是几何变换类 Matrix 类. Matrix 提供的的是一个 3×2 仿射矩阵, 上面的 2×2 矩阵是线性变换后的数据, 下面一行则是平移数据. 利用这个矩阵类可

以对图形进行平移、比例、对称、旋转等变换. 重要的方法有:

(1) Translate (offset X, offset Y, MatrixOrder-Append); 用于矩阵的平移, offset X, offset Y 分别表示 X 和 Y 方向的平移量.

(2) Scale (scale X, scale Y, MatrixOrder); 用于矩阵的缩放调整, scale X, scale Y 分别表示 X 和 Y 方向的缩放比例

(3) Rotate (Angle); 用于将矩阵顺时针旋转, Angle 表示旋转的角度.

2.2 DXF 文件解析流程

一个 DXF 文件就是一个矢量图形, 主要由一些图元所组合而成. 因而, 解析一个 DXF 文件的过程就可以分解为解析各种图元, 然后将各种图元组合起来. 实际上在 DXF 文件的 7 个段中只有实体段是最有用的, 软件所需的图元数据信息都在实体段, 但由于这里是用在激光打标系统上, 故表段中的文字样式段也是应该关注的对象.

针对 DXF 文件的每一个段及子类, 只需要先搜索到对应的字符串, 然后顺序去读取其后的数据信息即可, 解析流程图如图 1 所示. 解析步骤大致有以下 3 步:

第一步: 打开一个 DXF 文件, 读取文件的文字样式, 文字样式处于表段中, 则先在表段中搜索到子类标记 (AcDbTextStyleTableRecord), 其后紧跟组码 2 和样式名, 取出样式名, 后面的 12 个字段可不关注, 直接跳过(需要跳过时直接浏览文件, 连续读取数据时直接读取), 直到需要的组码 3 和 4 对应的字段, 分别读出主要字体文件名和大字体文件名, 保存到系统的字体样式变量中, 作为该文件的标准文字样式.

第二步: 文件解析的重点在于内容丰富的实体段的解析了. 首先在文件中搜索实体段名的字符串 ENTITIES, 然后顺序读取图元形状和数据信息. DXF 文件对图元的描述是比较简洁的, 它记录操作和几个重要元素信息, 包括图元类型, 特别的点及重要的数据. 例如在记录一个样条曲线图元时, 只需要记录样条曲线的阶数、节点数、控制点数及样条曲线标志(通常包括闭全、周期、线性等属性), 这些信息已经能够完整描述一段圆弧或一个圆了, 与 GDI+ 描述图元的方式相同.

第三步: 将读取到的图元信息进行处理, 最后添

加到绘图路径中. 有很多地方, 需要耐心去测试, 有时由于坐标的标准不一致或文件生成时数据的处理会导致直接读取到的数据并不能代表原始图形, 而是要根据偏差的性质和大小利用 GDI+ 提供的 Matrix 对象装载参数用于 transform() 进行一些变换和比例调整.

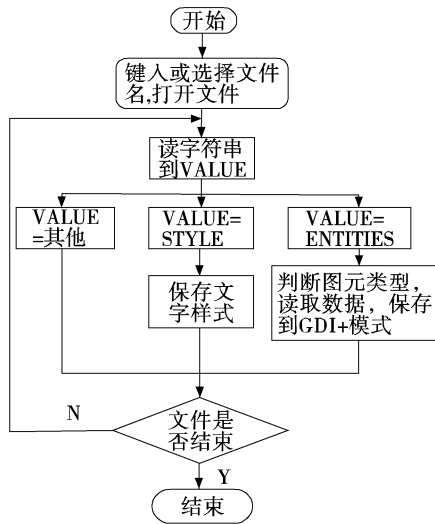


图 1 DXF 文件解析流程图

导入 DXF 文件时, 读入图元信息, 将各个图元按顺序组成一个链表, 连接起来就形成了一幅完整的图形. 系统能解析的 DXF 文件图形包含: 直线、圆、圆弧、椭圆、椭圆弧、多义线、贝赛尔曲线, 文本信息.

2.3 直线和圆的解析

直线和圆的组成比较简单, 解析起来比较方便. 在得到直线的 2 个端点坐标之后, 直接用 AddLine (PointF1, PointF2) 方法添加直线到当前路径中. 圆的 2 个重要参数是圆心坐标和圆半径, 容易获取, 但在 GraphicsPath 类中没有直接添加圆路径的方法, 于是可用添加椭圆路径的方法 AddEllipse(RectangleF) 代替, 参数只需要圆的外接矩形即可.

2.4 椭圆解析实例

2.4.1 椭圆的组码说明(如表 2)

表 2 椭圆组码

100	子类标记 (AcDbEllipse)
10,20,30	中心点坐标 (X, Y, Z)
11,21,31	相对于中点的长轴端点坐标 (X, Y, Z)
40	短轴与长轴的比例
41	开始参数(起始角度)
42	结束参数(终点角度)

2.4.2 椭圆的具体解析

(1)基本数据的获取. 在实体段中用查询的方法找到椭圆的子类标记后, 就可以根据表格中组码分别读取到椭圆的中心点坐标 Center、长轴端点相对于中心点的偏移量坐标 Deta、短长轴之比 Ratio、起始角度 StartAngle 和终点角度 EndAngle. 在得到这些数据之后, 就可以计算出下列椭圆外接正方形的参数.

椭圆的长轴半径为 $LongRadius =$

$$\sqrt{Deta.X * Deta.X + Deta.Y * Deta.Y}$$

, 椭圆外接矩形宽度 $Width = 2 * LongRadius$, 矩形高度为 $Height = 2 * LongRadius$ 倾斜角度 $LeanAngle = \arctan(Deta.X) * 180/\pi$.

(2)处理数据, 得到图形. 为了表示任意位置任意倾斜角度的椭圆, 可以做如下处理: 将椭圆的中心点先移到原点, 使倾斜角为 0° , 外接矩形在 Y 方向上缩放 Ratio 倍, 然后将此椭圆按照原倾斜角度进行旋转, 最后再把旋转后的椭圆平移到原中心点. 这样就减少了直接做变换的复杂度和工作量.

(3)效果分析. 顺序导入水平和 45° 倾角的椭圆和椭圆弧各一个, 导入效果如图 2.

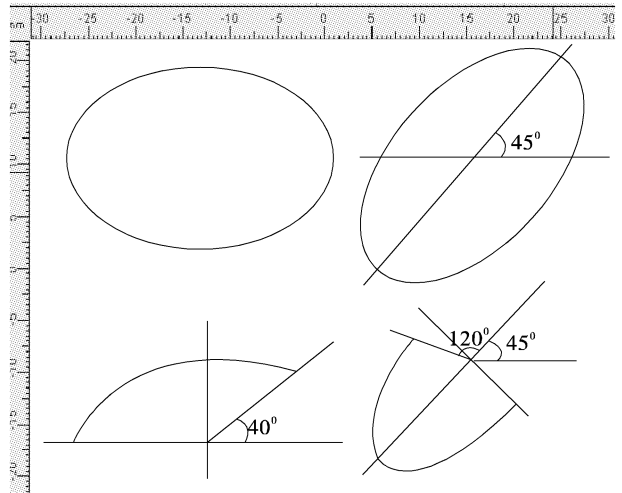


图 2 椭圆解析仿真图形

从图 2 中可以看到, 经过解析后的图形与原图形保持了很好的一致性. 左下椭圆弧所对应的原椭圆是水平方向, 弧的起始角度为 30° , 终点角度为 180° , 而右下椭圆弧的原椭圆是一个做倾斜角为 45° 的椭圆, 起始角度为 120° , 终点角度为 270° . 注意: 这个角度是相对于椭圆的第一个轴端点而言的, 所以在对角度进行处理时, 始终要是围绕这个端点来做变换, 否则, 椭圆会出现颠倒或偏转.

2.5 多义线解析

从实体段中直接读取到多边形的顶点数和各个点的坐标,然后将这些顶点坐标数组指针和需要的线条的数目作为参数传递给方法 `AddLines(PointF, num)` 将这些点顺序连接起来. 如果多边形标志为 1, 则表明多义线是封闭的, 那么还要添加一条线段将第一点和最后一点连接起来.

2.6 文本解析

2.6.1 单行文本解析

(1) 读取基本信息. 在找到单行文本的子类标记 `AcDbText` 后, 然后按组码一一对应读出文本的初始坐标、文字高度、文字旋转角度、文本倾斜角.

(2) 确定文本字体样式. 但对于每一个具体的文本图元, 需要分析其字体样式是否和文件的标准字体样式一致, 方法是: 如果图元中出现组码 7, 则表示此图元文字样式非标准, 文字样式要从 7 组码后取出进行保存. 如果没有组码 7 出现, 则文字样式与文件标准一致. 在找到了样式名之后, 要得到真正的字体样式, 还需要根据字体样式名从字体库中查找出与字体名对应的字体样式, 取字体样式分隔符“.”前面的内容即可.

(3) 文本的显示. 调试过程中会发现如果仅利用从实体中读取的数据是远远不能正确的显示文本的, 文本的角度、坐标并非如 DXF 文件中所描述的那样, 在这里, 还有一个很重要的步骤就是: 对于不同倾角的文本, 文本的初始位置与第一个字符的位置是有相对变化的. 因此对应不同的倾角第一个字符的位置要根据文本的位置进行调整, 保证文本位置的准确性和后续变换的基准. 将文本平移到原点处, 先进行镜相变换, 然后进行旋转变换, 便可使文本达到预定角度.

2.6.2 多行文本解析

多行文本的解析和单行文本的解析大致一样, 只是在细微处有一些差别. 首先其子类标记为 `AcDbMText`. 其次多行文本的解析也会碰到许多问题, 例如:

① 如果不对 DXF 文本图元中的文本内容进行过滤的话, 会看到有“{”, “;”, “}”等一些多余字符出现.

② GDI+ 中只有一个输出文本的函数 `DrawString`, 那么多行文字的实现关键在于回车的处理,

如处理不好, 多行文本就会成为单行文本.

通过对文本内容的仔细分析, 以下方法可以解析上述问题:

① 对于“{”这些单个的特殊的字符可以逐个找出, 然后将其删除. 然而其他普通多余字符找起来就不太方便了, 但有一个特点应注意到: 这些字符一般都只会出现在“;”的前面, 回车的后面. 于是利用这两个标志就可以确定多余字符的区间, 然后将其删除即可.

② 回车的问题, 需要用好两点: 一是回车在文本中的字符串表现为“\P”, 使用 `Cstring` 类的 `Find` 函数可以在整个 DXF 文本内容中找出所有的子字符串“\P”, 然后用字符串“\n\r”对其一一进行替换, 替换后还并不能正确的显示. 二是默认的文本矩形宽度是一个字符的高度, 若要显示多行, 则需要增加文本矩形的宽度, 使其能容纳多行文本. 在这里用到了 GDI+ 画图的一个特性: 当矩形的宽度足够时, GDI+ 中可以对字符自动换行, 只需要有控制字符串“\n”.

2.6.3 文本实例

用 AutoCAD 创建文本对象: 字体为隶书文本“武汉科技大学”, 然后导入到激光打标软件中, 可以得到如图 3 的效果. 说明程序不仅能正确地解析多行文本的文字内容, 而且可以正确地辨别出文本的字体, 达到预期的显示效果.

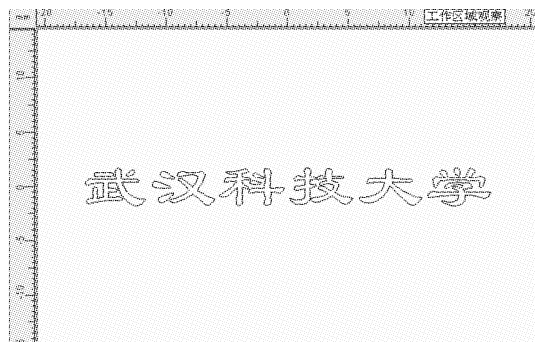


图 3 隶书文本

3 实例效果

对基本所有类型的图元进行解析后, 将各个图元分别添加到子路径中, 然后链接成为整个的图形路径, 可以集中进行显示和操作. 图 4 是经由多种图元所组合起来的一个复杂的文件解析所获得的效

(下转第 19 页)

参考文献

- [1] 王忆锋,唐利斌.第三代碲镉汞器件的研究发展[J]. 光电技术应用,2009,24(5):17-22.
- [2] 张舒,褚艳利,李开勇,等.GPU高性能运算之CUDA[M].北京:中国水利水电出版社,2009:8-13.
- [3] 张天林,张思敏.CPCI-E与VPX总线标准的比较分析[J].工业控制计算机,2009,22(7):1-5.
- [4] 申洋,唐明文.机载红外搜索跟踪系统(IRST)综述[J].红外技术,2003,25(1):13-18.
- [5] 黄星明,余国文,王宏远.一种改进的红外焦平面非均匀性多点校正方法[J].红外与激光工程,2005,34(1):62-65.
- [6] 李欢,杨捷.求解二值图像连通域的改进算法[J].计算机与现代化,2005,(4):11-13.

(上接第11页)

果,图形中包含的图元有线段、圆、圆弧、椭圆弧、矩形、多边形、多义线、单行文本和多行文本.从图4中可以看出,此软件不仅可以很完美地解析各个图元,而且各个图元之间的相对关系也保持得很好,使得图形不会失真.同时也说明该软件可以很好地解析比较复杂的矢量图形,从而能够满足激光打标需要,更好地为激光打标系统服务.

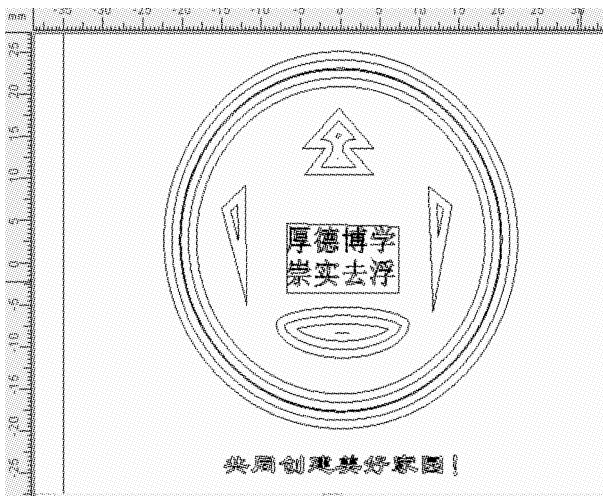


图4 DXF文件解析综合效果图

4 结束语

在激光打标时,激光笔只是在某个连续图元的起点处启动,终点处停笔,中间可处理为匀速运动,减少了激光笔的起停次数,激光笔的运动更为平滑,激光能量均匀分配在图线上,不用担心因为能量分配不均而导致打标深浅不一,同时节省了不少的时间,提高了激光加工的效率.因此,在激光加工系统

中利用DXF格式的文件拥有很大的优越性,只要获取各种图元,然后利用插补、填充等算法进行适当的处理就可以.激光标刻的图案一般都是比较简洁,轮廓比较清晰,如果利用jpg、bmp格式的图片的话,会需要灰度、二值化、拟合等处理然后才能提取出图形的轮廓,过程比较复杂,不如DXF文件应用性强.通常会有这样一个感受,DXF文件记录图形的方式与VC++中类的思想有几分相似,因此利用VC++来对DXF文件进行修改也会是相对较为方便的,可重构性得到了很大提高.

也可以考虑将其他格式的图形转换为DXF格式,然后导入系统进行处理,效率也许会更高.如果这个可以做到,系统的通用性必然得到大幅改善.

参考文献

- [1] 杨友东. Visual C++ 程序设计全程指南[M]. 北京:电子工业出版社,2009.
- [2] 王子茹,任清波.基于VC++的DXF数据文件接口的研究[J].厦门理工学院学报,2007,1(15):26-30.
- [3] 胡胜红.使用VC++编程实现DXF文件数据提取[J].福建电脑,2006,(11):189.
- [4] 刘启生,邵东伟,杜云明,等.基于DXF文件的数控加工系统的实现研究[J].佳木斯大学学报(自然科学版),2009,4(27):542-544.
- [5] 刘其斌.激光加工技术及其应用[M].北京:冶金工业出版社,2007.
- [6] 程铁皋.Windows动态数据交换程序设计用Visual C++和Microsoft C/C++[M].北京:北京航空航天大学出版社,1995.