

基于 Windows 字库的点阵数据提取方法*

霍艳忠^{1,2}, 牛国玲³, 郑永春^{1**}, 张天顺¹

- (1. 云南农业大学 工程技术学院, 云南 昆明 650201;
2. 佳木斯大学 信息电子技术学院, 黑龙江 佳木斯 154007;
3. 佳木斯大学 机械工程学院, 黑龙江 佳木斯 154007)

摘要: 基于编程软件 Delphi, 介绍并实现了一种从 Windows 字库中提取字体信息, 并将其转换为点阵数据的方法。通过使用 Windows 的 API 函数此方法共分 2 步, 第 1 步从字库中提供字体数据并输出到位图, 第 2 步从位图中提取点阵数据。

关键词: Delphi; 点阵数据; 字库

中图分类号: TP 391.1 **文献标识码:** A **文章编号:** 1004-390X (2009) 04-0626-004

Based on Windows Font Library Picking up Pixel Data of Word

HUO Yan-zhong^{1,2}, NIU Guo-ling³, ZHENG Yong-chun¹, ZHANG Tian-shun¹

- (1. College of Engineering and Technology, Yunnan Agricultural University, Kunming 650201, China;
2. College of Information Science and Electronic Technology, Jiamusi University, Jiamusi 154007, China;
3. College of Mechanical Engineering, Jiamusi University, Jiamusi 154007, China)

Abstract: In this paper, based on Delphi, a method of picking up pixel data of word from Windows font library was introduced and implemented. By using API functions, this method could be divided into 2 steps: (1) getting data for Windows font library and putting it on bitmap; (2) picking pixel data from bitmap.

Key words: Delphi; pixel data; font library

随着计算机硬件技术的不断进步和软件功能的不断强大, 在计算机上显示汉字的问题已经得到了圆满地解决。但对于单片机系统和嵌入式系统, 如何高效、迅速地显示汉字仍然是一个有待进一步解决的问题。现今, 以液晶、大规模 LED 等为核心的显示产品应用得越来越普遍, 而这些产品的后台处理器通常是单片机或嵌入式芯片。由于受单片机或嵌入式芯片处理速度、硬件规模等的限制, 要求在这些系统上使用一些高效、快捷的显示策略。

针对单片机系统和嵌入式系统的特点, 人们制定了各种显示汉字的方法, 如“标准字”^[1]法。该方法是将整个字库先写到程序存储器中, 然后使用与计算机相同的方法从字库中检索文字。此方法的优点是软件容易编写, 但是由于字库较大, 一般会超过单片机的寻址能力, 需要采取一些特殊的手段才能使用。除此之外, 还有“固化显示字模”^[1]法、“建立带索引的小字库”^[1]法和比较先进的“汉字动态编码”^[1]法等, 这些方法都可以缩小字库的体积, 但前两种方法都要对存储器进行繁琐的规划及复杂的

收稿日期: 2008-03-29 修回日期: 2008-09-22

* 基金项目: 云南农业大学学生科技创新创业行动基金项目 (NDF010)。

作者简介: 霍艳忠 (1977-), 男, 吉林省柳河县人, 讲师, 在读硕士研究生, 主要从事微型计算机控制方面的研究。E-mail: hyz07158@126.com。

** 通讯作者 Corresponding author: 郑永春, 男, 云南保山人, 教授, 硕士生导师, 主要从事农业工程研究。E-mail: aiy5055@sohu.com。

查询运算,这对单片机系统和嵌入式系统的运行速度是个很大的考验。无论是“动态编码”^[1]法还是“固化显示字模”^[1]法,都需要先获得汉字(或字母)的点阵数据。

为了使用“动态编码”^[1]法等,可以从 UC-DOS 的字库文件中获取点阵数据,但这样做的一个最大的弊端就是:字体严重受限且点阵数固定,这对产品的显示效果会有很大的影响。Windows 支持多种字库格式,如 TTF, FON, TTC 等,这些字库含有丰富的字体式样,利用 Windows 的 API 函数可以对文字的点阵数进行随意放缩。针对具体的产品,可以设计具有独特字体样式的字库。从 Windows 字库中提取点阵信息,可以极大地丰富显示内容,带来生动的显示效果,这对显示产品的开发与应用非常有益。

下文基于编程软件 Delphi 介绍一种从 Windows 字库中提取点阵数据的方法。此方法不仅可以从 Windows 字库中提取汉字字模,还可以提取非汉字文字的字模。

1 提取思路及相关函数

1.1 提取的思路

Windows 的 API 函数库中有丰富的字体操作函数,基于这些 API 函数,本文介绍的这种提取方法总体上分为两大步:第 1 步先将需要的文本信息从所选的字库文件中提取出来,并将其输出到位图上;第 2 步对第 1 步中得到的位图进行逐点地转换,进而得到点阵数据。在这 2 步中,第 1 步需要使用 API 函数与 Windows 操作系统进行交互;而第 2 步则对图像进行处理。Delphi 封装了许多图像操作函数,这些函数使我们对图像的操作变得简单、容易。

1.2 相关的 API 函数

下面对这种提取方法中用到的几个关键的函数作以简要的介绍。

1.2.1 CreateFont 函数

CreateFont^[2]函数是 Windows API 函数库中一个比较有用的函数。正如它的名字一样,这个函数用于创建逻辑字体。在调用这个函数时,我们可以通过相应的参数指定所创建字体的基础字

集、基础字库、字体宽高等特征。这个函数的原型如下:

```
HFONT CreateFont (int nHeight, int nWidth,
int nEscapement, int nOrientation, int fnWeight,
DWORD fdwItalic, DWORD fdwUnderline,
DWORD fdwStrikeOut, DWORD fdwCharSet,
DWORD fdwOutputPrecision, DWORD fdwClipPrecision,
DWORD fdwQuality, DWORD fdwPitchAndFamily, LPCTSTR lpszFace);
```

其中一些主要参数的含意如下:

nHeight: 字体的高度

nWidth: 字体的宽度

fnWeight: 字体的粗细

fdwItalic 和 fdwUnderline: 字体是否倾斜和是否有下划线

fdwCharSet: 字体使用的字符集

lpszFace: 基础字体的名称

下一小节将给出 Delphi 中的使用代码。

1.2.2 SelectObject 函数

在 Delphi 中基于 CreateFont 函数可以创建一个字体对象,而这个对象必须与具体的设备关联才能使该设备具有字体对象的字体特征。SelectObject^[2,3]函数可以使字体对象与一个设备句柄相关联。这样,就可以直接向这个设备输出文字,而设备会自动使用所关联的字体样式去格式化这些文字。SelectObject 函数的原型如下:

```
HGDIOBJ SelectObject (HDC hdc, HGDIOBJ
hgdiobj);
```

其中:

hdc: 设备句柄 hgdiobj: 字体对象

1.2.3 DeleteObject 函数

DeleteObject^[2,3]函数用于释放字体对象。函数原型为:

```
BOOL DeleteObject (HGDIOBJ hObject);
```

其中:

hObject: 为字体对象名。

1.2.4 TextOut 函数

TextOut^[4,5]函数用当前选定的字符、背景颜色和正文颜色将一个字符串写到指定设备的指定位置上。函数原型为:

```
BOOL TextOut ( HDC hdc, int nXStart, int
nYStart, LPCTSTR lpString, int cbString);
```

参数的含意如下:

hdc: 设备环境句柄。 nXStart: 指定用于字符串对齐的基准点的逻辑 X 坐标。

nYStart: 指定用于字符串对齐的基准点的逻辑 Y 坐标。 cbString: 字符串的字符数。

lpString: 指向将被绘制字符串的指针。此字符串不必以 \0 结束, 因为 cbString 中指定了字符串的长度。

2 提取方法的实现

2.1 从字库中提取数据

下面这段代码将文字按照指定的字体样式输出到位图的画布上。

```
Function GetFontPoints ( Text: String; TotalLine-
Num, TotalColNum: Integer; AFontName: String;
XOffset, YOffset: Integer; var FontPoints: Font-
PointsData; IsMultiLine: Boolean = False;
ZoomScaleLine: Real = 1.0; ZoomScaleCol: Real
= 1.0; SaveToBmpPath: String = "";
CharSets: DWord = GB2312 _ CHARSET );
String;
var
TempBmp: TBitmap; DC: HDC; TheFont: HFont;
i, j: Integer;
begin
TempBmp := TBitmap. Create; TempBmp. Pixel-
Format := pf8bit[5];
TempBmp. Height := TotalLineNum; TempBmp.
Width := TotalColNum;
TempBmp. Canvas. Brush. Color := clWhite;
DC := TempBmp. Canvas. Handle;
TheFont := CreateFont ( Trunc ( ZoomScaleLine *
TotalLineNum ),
Trunc ( ZoomScaleCol * TotalColNum ) div 2, 0,
0, 400, 0, 0, CharSets, Out_Default_Precis,
Clip_Default_Precis, Default_Quality, Default_
Pitch OR FF_SCRIPT, Pchar ( AFontName );
SelectObject ( DC, TheFont );
//下面这一行用指定的字体样式向位图的画
```

布上输出文字

```
TextOut ( DC, XOffset, YOffset, Pchar
(Text), Length (Text));
.....
end;
```

2.2 点阵化处理

下面这段代码从含有文字图样的画布上提取点阵数据。

```
SetLength[6,7] ( FontPoints, TotalLineNum,
TotalColNum); Result := "";
for i := 0 to TotalLineNum-1 do
begin
if ( Result < > "" ) and ( IsMultiLine ) then Re-
sult := Result + #10;
for j := 0 to TotalColNum - 1 do
if TempBmp. Canvas. Pixels [ j, i ][5,8] < >
clWhite then
begin
FontPoints [ i, j ] := 1; Result := Result +
'1';
end
else
begin
FontPoints [ i, j ] := 0; Result := Result +
'0';
end;
end;
TempBmp. Free; DeleteObject ( TheFont );
.....
```

2.3 程序流程图

整个提取方法的逻辑比较清晰。提取算法的流程图如图 1 所示。

2.4 应用范例

应用本文所介绍的方法并结合单片机系统开发的经验, 作者已制作出实际的提模工具, 其运行界面如图 2 所示。

3 结束语

正如前文所述, 汉字提模及显示方法有多种, 实现起来难度各异, 本文只介绍并实现了其中的一种。经作者实际验证, 该方法可靠、可行。

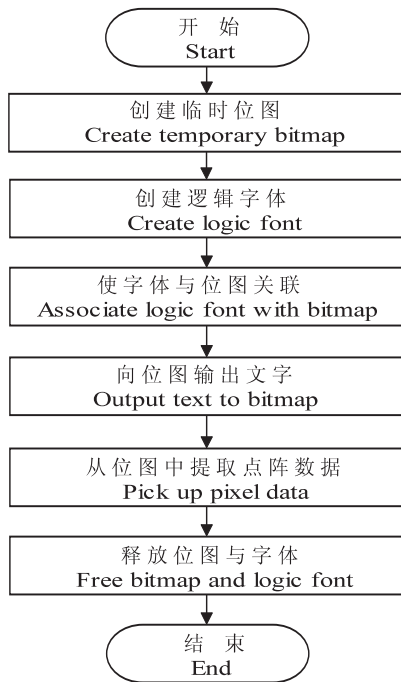


图 1 算法流程图

Fig. 1 Arithmetic flow chart

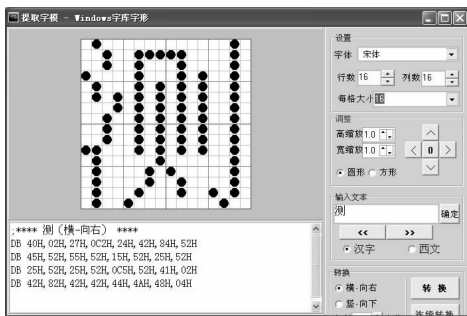


图 2 程序运行界面

Fig. 2 Program running interface

此方法中, 由于先将字模信息输出到了位图, 所以在转换为点阵数据之前, 可先利用 Delphi 中的位图操作函数对字模进行操作, 然后再转换, 这样可极大地丰富显示效果。

Windows 作为一种主流的操作系统, 提供了丰富的资源。在搞产品研发时, 首选的就是尽量利用 Windows 中已有的资源。Windows 强大的 API 函数库也最大限度地满足了需求。

[参考文献]

[1] 陈平晓. 点阵 LCD 驱动显控原理——基础版 [M]. 北京: 电子工业出版社, 2007.

[2] 王华. Delphi 5 编程实例与技巧 [M]. 北京: 机械工业出版社, 2000: 167 - 175.

[3] 张岭, 宋坤. Delphi 程序开发范例宝典 [M]. 北京: 人民邮电出版社, 2006: 128 - 183.

[4] 杨富国, 陈立俊, 唐巧琪, 等. Delphi 程序开发案例解析 [M]. 北京: 清华大学出版社, 2006: 208 - 235.

[5] 刘骏. Delphi 数字图像处理及高级应用 [M]. 北京: 科学出版社, 2003: 34 - 37.

[6] 何光渝, 雷群. Delphi 常用数值算法集 [M]. 北京: 科学出版社, 2002: 53 - 72.

[7] TEIXEIRA S, PACHECO X. DELPHI6 开发人员指南 [M]. 北京: 机械工业出版社, 2003: 53 - 72.

[8] 赵晓玲. 可视化程序设计——Delphi [M]. 北京: 机械工业出版社, 2005.