

嵌入式系统中的 TrueType 字体驱动

陈文文, 奚宏生

(中国科技大学网络传播系统与控制联合实验室, 合肥 230026)

摘要: Freetype 库应用于嵌入式高清电影播放机时, 存在 srt 字幕显示滞后的问题。针对该缺陷, 给出一种实现 TrueType 字体驱动的方法, 建立字体本地索引, 将 TrueType 字体文件中必要的数据库导入内存, 建立字体 glyf 内存数据库。测试表明, 该驱动稳定可靠, 适用于资源受限的嵌入式平台。

关键词: 嵌入式系统; TrueType 字体; 驱动; Freetype 库

TrueType Fonts Drive in Embedded System

CHEN Wen-wen, XI Hong-sheng

(Joint Lab of Network Communication System & Control, University of Science and Technology of China, Hefei 230026)

【Abstract】 If the Freetype libraries are used in the embedded high-definition movie player, the srt subtitle has a problem of lag. This paper presents a method to realize the TrueType fonts drive, establishes the local indexes for the fonts, and puts all the necessary data of TrueType fonts file into memory to found the glyf memory fonts database. Test shows that the developed drive is stable and reliable and is ideal for the resource-constrained embedded platform.

【Key words】 embedded system; TrueType fonts; drive; Freetype libraries

1 概述

在当前的嵌入式系统产品中, 主要是采用点阵字库来显示文字。然而点阵字显示缺乏灵活性, 改变大小容易引起失真并且无法改变其显示风格, 所以在高清产品中显得越来越不合适。而采用 TrueType 字库则可以显示出美观清晰的文字来。理论上无论放大到什么尺寸, 其字符边缘总是光滑的, 不会出现锯齿。它的不足之处只是在低分辨率的设备上或者使用小号字显示时显示效果不如点阵字。但在高清设备上, 此缺点可以忽略不计。

TrueType 字库文件格式非常复杂, 开发驱动困难很大。而将字体引擎 Freetype 嵌入式高清播放机中时, 在播放影片的过程中, srt 字幕显示过慢, 无法跟进视频播放的进度, 滞后现象非常严重。针对这些问题, 本文提出建立字体 glyf 内存数据库, 加快字体数据提取过程, 从而提高字体显示速度。

2 TrueType 字体描述技术

TrueType 采用几何学中二次贝塞尔(Bézier)曲线及直线来描述字体的外形轮廓。和 PostScript 相比, 其曲线方程次数降低, 在保证轮廓曲线光滑性的前提下, 极大地提高了字形还原的速度。

二次贝塞尔曲线的具体定义如图 1 所示。



图 1 由 3 点定义的二次贝塞尔曲线

给定 3 点 p_0, p_1, p_2 , 其中, 线上的点 p_0 和 p_2 表示线的端点, 线外的点 p_1 控制其弧度。由它们定义了这样一条曲线:

p_0p_1 与曲线在 p_0 点相切, p_1p_2 与曲线在 p_2 点相切。这 3 点确定的曲线可以用下面的参数方程来描述:

$$p(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2, 0 < t < 1$$

在 TrueType 字体中, 每一个字形都是由这样一系列连续的点来描述, 在这些点之间用直线或者二次贝塞尔曲线连接起来, 从而构成字形轮廓。如图 2 所示的字母 B, 就是由 3 条轨迹组成的, 其中, 在线上的点表示成黑点, 在线外起控制作用的点则表示为圆圈, 这些控制轨迹的点坐标都按顺序记录在 TrueType 字库的 glyf 表中。另外, 字符底部的 2 个黑点用于控制字体的水平布局, 其信息记录在 hmtx 表中。驱动的任务即是已经将数字化的字形还原出来。

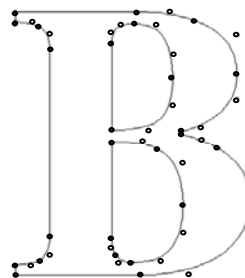


图 2 由 3 条轨迹组成的字符外形

3 TrueType 字体文件格式

TrueType 字体文件是用树型的表组织起来的, 它由 3 个部分组成: 文件头, 描述表目录和一系列的描述表^[1-2]。

基金项目: 国家“863”计划基金资助项目(2007AA01Z235)

作者简介: 陈文文(1984-), 男, 硕士, 主研方向: 嵌入式系统; 奚宏生, 教授、博士生导师

收稿日期: 2009-11-30 **E-mail:** chenwen@mail.ustc.edu.cn

常用的 ttf 文件描述表共有 19 个,其中有 10 个描述表是必需的,另外 9 个则是可选的表^[2]。针对嵌入式平台的应用,实际上 name, post, OS/2 这 3 个表都不需要。而这些必备的表中又以 cmap, glyf, loca 这 3 个最重要。

(1)cmap 表

因 TrueType 字库中排列字模的顺序和系统字符编码不同,故使用该表定义从字符编码到字符轮廓的映射关系。对于不对应任何轮廓的字符编码,都映射到轮廓索引 0,这个位置的字符轮廓被设置成空字符,其形状为一个空心方框。

(2)glyf 表

glyf 表是 ttf 文件的主体部分,它包括每个字符的轮廓数据信息和修正轮廓的指令信息。每个字符的轮廓描述信息都有一个头,在这里记录了轮廓数目 numberOfContours 和 xMin, yMin, xMax, yMax 坐标。当 numberOfContours ≥ 0 时,这个字符是简单字;反之,就是复合字。Windows 下用的汉字字库和英文字库中标准的 ASCII 码都是简单字体,所以,暂时不考虑复合字情况。在轮廓的描述表中,记录了具体每一个轮廓点的坐标,并且还记录了修正字符用的指令 instructions[n]及其长度。这就是 hinting 技术所要用的指令,然而这种技术只是在低分辨率的输出设备上或者显示小字体时才能明显改善显示效果,因此,在高清设备上并不需要。另外,字体制造公司只是给少量字符(如 ASCII 码)制作了相关的 hinting 指令,中文字体因为其数量太大、字形太复杂,而并没有相应的 hinting 指令。

(3)loca 表

loca 表提供了一种快速访问字符轮廓数据的方法,它记录了每一个字符的轮廓数据信息在 glyf 表中的偏移位置。

4 字体驱动的开发

TrueType 字体的处理比较麻烦,需要进行矢量计算。为了正确显示每个汉字,首先要对每一个字符建立起准确无误的索引关系;然后将所有字符的 glyf 信息导入内存,建立起内存数据库,从而通过牺牲一小部分内存空间,提高字体轮廓数据读取的速度。

依据嵌入式平台的具体应用,本驱动对 ASCII 码 20H~7EH 共 95 个字符和 GB2312 中共 6 763 个汉字,建立一共包含 6 858 个字符的内存字库。

4.1 字符索引关系的建立

索引关系建立关键的问题在于:获取字符的国际码后,将它转换为 unicode,并进而通过 cmap 提供的映射机制找到字符在 loca 表中的 glyf_index。具体关系如图 3 所示。

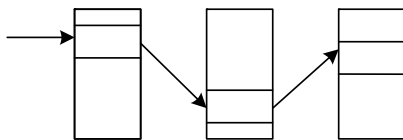


图 3 字符索引关系

对于英文字符而言,其内码和 unicode 码是一样的,因此,无需转换。Scriptics 公司为每一个汉字建立了从其国际码到 unicode 码的一一对应关系,另外,国际码中汉字的排序和 GB2312 编码中的排序是一样的。基于这种关系,建立如下索引:

```
unicodes[char_index] = char_unicode;
```

即将每一个字符的国际码转换成从 0 开始的连续的本地索引号([0,1,...,6 857]), char_index, char_unicode 正是对应该字符

的 unicode 编码。

而由 unicode 编码来获取该字符 glyph_index 的伪代码如下:

```
for(i=0; (i<segCount) && (char_unicode > endCount[i]); i++);
/*确定 char_unicode 所在的段*/
if(unicode_of_this_char >= startCount[i])
{if(idRangeOffset[i] != 0)
{ glyph_index = *(idRangeOffset[i]/2 + (char_unicode-
startCount[i]) + &idRangeOffset[i]);
}else{
glyph_index= char_unicode; }
glyph_index += idDelta[i];
}else
return 0;
```

得到 glyf index 后,可直接在 loca 表中找到该字符在 glyf 表中轮廓数据的偏移位置。因此,结合以上 2 步就为本地字库中的每一个字符建立起了直接获取对应 glyf 数据的索引。

4.2 字符glyf数据内存库的建立

依据已经建立好的本地索引关系,按照以上确定的字符顺序依序把每一个字符的 glyf 数据和水平布局数据导入到相应的数据结构。

设计数据结构 ttf_glyph 和 ttf_glyph_metrics 如下所示:

```
struct ttf_glyph {
struct ttf_glyph_metrics metrics;
/*字体布局信息*/
unsigned long address;
/*该字体 glyph 在内存数据库中的位置*/
unsigned long size;
/*字体 glyph 的大小*/;
struct ttf_glyph_metrics {
unsigned short xMin;
unsigned short yMin;
unsigned short xMax;
unsigned short yMax;
unsigned short advance; /*取自 hmtx 表格里*/
unsigned short leftSideBearing;};
```

其中, ttf_glyph 用来描述字符的整体信息,而 ttf_glyph_metrics 则用来描述字符的布局信息。

建立内存数据库的流程如图 4 所示。

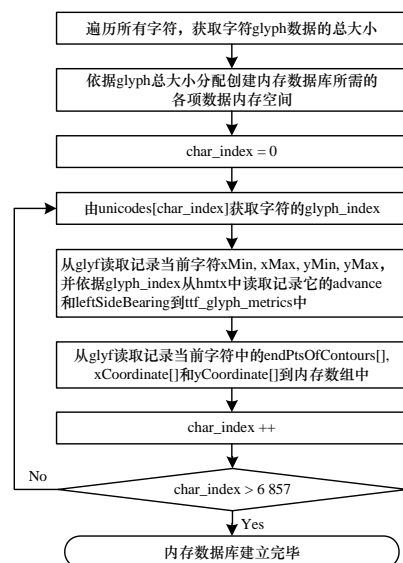


图 4 内存数据库创建流程

因为字符 glyph 的大小不一,而其他数据大小相同,所以分配 2 段内存分别记录这些信息。其中,在第 1 段内存中,记录每一个字符的 ttf_glyph 和 ttf_glyph_metrics 数据,并由 ttf_glyph 中的 address 记录存放该字符 glyph 的地址;在第 2 段内存中则具体记录每一个字符的 glyph 数据。可见,创建内存数据库后,已把 6 858 个字符导入到内存中,并由 ttf_glyph 结构体记录了绘制单个字符所需的所有信息。

4.3 字体的绘制

汉字的内码排序亦同 GB2312 编码,获取字符的内码后,将其转换为 char_index,这样便可以直接由 char_index 从内存数据库获取轮廓数据。随后执行矢量计算绘制字形并将它放大到指定的大小,再通过判断内部栅格点的回转数来生成位图(bitmap)^[1]。最后将位图信息写入(On Screen Display, OSD)的 buffer 中,即可以在输出设备上显示绘制出来的字符。使用高清电视机作为输出终端时,汉字显示效果如图 5 所示。



图 5 64 磅汉字在高清设备上的显示

5 对比及测试

作为开源的字体操作函数库 Freetype,它可以处理多种矢量字体,包括 TrueType 以及点阵字库^[3]。文献[4]给出了其具体的字体驱动显示流程。

Freetype 采用面向对象的思想开发,提供统一接口来访问多种字体格式文件,其最大的特色是函数库可移植、可扩展、可定制,并避开 Apple 公司的专利实现了 autohinting 技术,使得字符显示更加精美。然而,在绘制字符的过程中,必须每次计算每个字符的 glyph index,再调用 FT_Load_Glyph()函数从字库文件中加载字符的 glyph 数据,而没有建立相应的内存数据库,导致效率不高,引起 srt 字幕显示滞后

(上接第 256 页)

$$\forall y \in V_x : w'(x, y) = \frac{w(x, y)}{R(x, y)}$$

$$\forall y \in V_x : w'(x, x) = 1 - \sum_{y \in V_x} w'(x, y)$$

其中, $V_x = \{y \in V \mid w(x, y) > 0\}$ 表示与 x 连接的节点。

5 TRCKMS 的应用

在 TRCKMS 中,如果预测存在有风险,则现场技术人员首先使用系统提供的风险判断规则确定风险原因,找到相应的解决办法。如果找不到,则系统启动案例推理,寻求解决问题的方法。如果还是找不到,则使用专家地图功能,查询系统中已有的专家,找到与预测到的风险最匹配的专家咨询。在 TRCKMS 中,专家地图是盾构施工风险控制的知识管理系统的一个模块,主要根据用户输入的信息(专家姓名),在后台执行搜索算法,从数据库中查找满足条件的专家信息。为了提高搜索的精度和使用的方便,搜索算法采用上面介绍的 2 种搜索方法。

6 结束语

在开发 TRCKMS 时,根据解决问题的需要,提出了构建专家地图的方法,从企业局域网的异构数据文档中发现专家地图的任务。本文定义了专家地图,提出 2 种构建专家地图的方法。在 W3C 语料库上进行的实验表明,这些方法能提高发现专家地图的效率。另外,基于图论的方法提出构建专家

问题。

本文建立了字体 glyf 内存数据库,在初始化的过程中,便将涉及 glyf 数据提取的相关计算全部完成,并把 glyf 数据导入到内存中,故绘制字符的过程中无须再进行二级映射的索引计算,通过定位字符的顺序后直接从内存数据库中读取数据即可。

本驱动全部用 C 语言实现,并使用不同的 TrueType 字体文件进行测试,对于英文字库(如 ARIAL.TTF, TIMES.TTF 等)和中文字库(如楷体 SIMKAI.TTF 等),驱动效果良好。最终驱动应用于 SIGMA 公司的单芯片多媒体处理器 EM8622 平台上,作为高清产品,能很好地显示人机界面字幕,并在播放高清影片时,能流畅地显示 srt 字幕。

6 结束语

本文通过建立字符 glyf 内存数据库,实现了一种适用于嵌入式平台的 TrueType 字体驱动,并将该驱动应用于高清电影播放器中。

后续的研究工作主要是实现字体显示的特效功能: Bordering(在字体的外围产生字形轮廓的特效)和 Shadowing(在原字形右边及下边产生阴影立体特效),从而在影片的背景和字幕颜色相同时,突出和美化字幕的显示效果。

参考文献

- [1] TrueType 1.0 Font Files Technical Specification(Revision 1.66)[Z]. (1995-08-02). <http://www.microsoft.com>.
- [2] 朱开玉, 宋国辉, 燕存良, 等. CAPP 系统中动态创建 TrueType 字体的研究[J]. 计算机工程, 2006, 32(8): 20-22.
- [3] A Free, High-quality, and Portable Font Engine[Z]. (2002-09-08). <http://www.freetype.org>.
- [4] 孙晓辉, 陈晓. 在嵌入式浏览器中使用 TrueType 矢量字库[J]. 电视技术, 2007, 31(8): 120-123.

编辑 顾逸斐

的社会协作网的算法,并将这些方法应用于 TRCKMS 系统的开发中。

参考文献

- [1] Maron M E, Curry S, Thompson P. An Inductive Search System: Theory, Design and Implementation[J]. IEEE Trans. on Systems, Man and Cybernetics, 1986, 16(1): 21-28.
- [2] Becerra-Fernandez I. The Role of Artificial Intelligence Technologies in the Implementation of People-finder Knowledge Management Systems[J]. Knowledge-based Systems, 2000, 13(5): 315-320.
- [3] Cross R, Parker A, Prusak L, et al. Knowing What Know: Supporting Knowledge Creation and Sharing in Social Networks[J]. Organizational Dynamics, 2001, 30(2): 100-120.
- [4] Enterprise Track[Z]. (2005-10-20). <http://www.ins.cwi.nl/projects/trec-ent/wiki/>.
- [5] Balog K, Azzopardi L, Rijke M. Formal Models for Expert Finding in Enterprise Corpora[C]//Proceedings of SIGIR'06. New York, USA: [s. n.], 2006.
- [6] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval[M]. [S. l.]: ACM Press, 1999.

编辑 索书志

