

压缩字体文件格式(CFF)研究

那日松¹, 敖其尔²

(内蒙古大学 计算机学院, 内蒙古 呼和浩特 010021)

摘要: CFF 是当前一种用在 Open Type 字库中的字体文件格式。此字体文件格式配合字符描述程序 Type2 字符串使用, 主要的特点是减少字体程序存储空间。本文详细地分析了 CFF 文件的结构, 并描述了其中的一些重要环节, 为进一步研究 Open Type 字体格式和建立蒙文的 Open Type 字库提供了基本知识。

关键字: CFF 字体; Type2 字体; Type1 字体; CID-Keyed 字体

中图分类号: H085.6

文献标识码: A

一、相关字体格式简介

字体是一种计算机文件, 它告诉操作系统如何在屏幕上绘制字符。每种操作系统都有一个字符集, 用来创建对话框或加载屏幕等部件上的文字。最常用的字体形式叫做 True Type 字体 (TTF), 可以很容易地通过控制面板加载, 蒙文也有自己的 TTF 字体。其他字体, 如: Open Type、Type 1 或 Postscript 字体, TTF 目前最常用, 但是 Open Type 将很快成为主流字库。那么蒙文的 Open Type 字体的设计也正是我们现在研究的焦点。下面简单介绍文中涉及到的几种字体格式。

Type2 是一种字符串格式, 在轮廓字体文件中压缩表示字符描述过程。这个格式被设计成和 CFF 配合使用。CFF/Type2 格式是 Open Type 字库的 Type1 格式的主要部分, 而且用于 Acrobat 3.0 PDF (PDF 格式 1.2 版本) 文件的嵌入字体。

Type1 是通过 ATM (Adobe 字体管理器) 软件应用于各种平台、操作系统及应用软件的单字节罗马字体格式, 也用于 PostScript 打印机上。Type1 字库使用了一个 PostScript 语言的专用子集, 可以获得最优化的性能和更多的压缩表示。Type1 指令集包括 hint 信息。Hint 可以在显示字体时帮助小字体和低分辨率字体生成更精确的位图。

CID 是 Character ID 的简称。CID 是一个字体文件组织格式。CID-Keyed 字体格式被设计成为大字符集字库 (特别是台湾、大陆、日本、韩国等双位元语系的国家文字) 提供高性能和灵活性。CID-Keyed 字体格式被看成是对 Type1 格式的扩展。字符仍用标准 Type1 指令表示, 但是文件组织方式不同。

二、CFF 字体格式

CFF 字体是对一个或多个 Type1 或 CID-keyed 字体进行压缩表示。不同于 Type1 和 CID-keyed 字体的是 CFF 允许多种字体一同存储在称作 FontSet (字体集) 的单元中。

CFF/Type2 字库比起相同大小的 Type1 字库, 平均要节省 30%-40% 的空间。没有使用子程序的情况下, 可以进一步增加节省空间的能力。这种节省空间的特点正是使用 CID-Keyed 字体格式的亚洲语言大字库所需要的。CFF 格式完全类似于 Type1 格式, 都是基于相同的绘图路径和提示 (hinting) 的思想。但是 CFF 可以对字库数据编码和程序进行更多的压缩。Type1 字体可以转换为 CFF/Type2 格式, 而且这个格式也可以再转换回 Type1, 这个过程没有任何的损失。

节省空间的实现:

- (1) 通过对编码方法进行优化来实现。例如：指令被扩展到允许一个指令可以对应到多个变量集合。
- (2) 把经常使用的值和字符串设置为缺省值。
- (3) 编码值得到有效的赋值。
- (4) 子程序可以在字体集(FontSet)之内进行共享操作。

现在 CFF 在 Acrobat 3.0 PDF(使用 PDF 格式 1.2)文档中作为嵌入字体，而且是由 Adobe 和微软共同声明的新 Open Type 字体格式的一部分 Type1 的主要成分。这种格式可以作为包含 PostScript 软件的 Level 3 打印机的打印机 ROM 的嵌入字体。需要嵌入到 PDF 文件中的字体首先要利用 Acrobat Distiller 或者 Acrobat PDFWriter 转换为 CFF/Type2 格式。然后一个附加的 Zip 压缩被应用，接着结果存储在 PDF 文档文件中。当文档要浏览或打印时，字体再转换回用于 ATM 软件的 Type1 格式或者包含在 PostScript 语言打印文件的 Type1 格式。PostScript 2017 版本将仅解释下载的 CFF 罗马字体(单字节)。所有的 Level 3 解释器将支持单字节和双字节 CFF 字体。

(一) CFF 的文件格式布局

在下面的表 1 里列出了文件格式的整体布局情况。虽然数据结构中的一些数据项可以改变次序，但是前 5 个数据项的位置是固定的。

表 1 CFF 文件格式设计

Table 1 CFF Format Layout

入口	说明
Header(表头)	-
Name INDEX	-
Top DICT INDEX	-
String INDEX	-
Global Subr INDEX	-
Encodings	-
Charsets	-
FDSelect	只有 CIDFonts 有
CharStrings INDEX	每个字体都有
Font DICT INDEX	每个字体和 CIDFonts
Private DICT	每个字体
Local Subr INDEX	CIDFonts 的每个字体和每个私有字典
Copyright andTrademark Notices	-

(二) CFF 的数据类型

下面主要介绍 CFF 使用的数据表示和数据类型。

所有多字节数值数据和偏移量字段存储在 bigendian 字节顺序中(高字节低偏移), 但是不支持任何对齐约束规则。这使得 CFF 格式可以从填充字节中识别出来。

CFE 数据对象经常由 CFF 数据中一些参考点有关的字节偏移量给出。这些偏移量可以是 1 到 4 字节长度。这里用到了把参考点围绕在括弧中的规范, 用 (0) 表示与 CFF 数据开始有的偏移量。CFE 的数据类型列在表 2 中。

表 2 CFF 的数据类型

Table 2 CFF Data Types

名称	范围	说明
Card8	0 - 255	1 字节无符号数
Card16	0 - 65535	2 字节无符号数
Offset	变量	1, 2, 3, or 4 字节 offset (由 OffSize 字段给出)
OffSize	1 - 4	用 1 字节无符号数给出 Offset 字段或字段大小
SID	0 - 64999	2 字节串标识码

大多数 CFF 数据是被包含在字典和索引两种数据结构中。

(三) 表头

CFE 字体文件的二进制数据开始于列在表 3 中格式的表头。

表 3 表头格式

Table 3 Header Format

类型	名称	描述(说明)
Card8	major	主版本格式(从 1 开始)
Card8	minor	次版本格式(从 0 开始)
Card8	hdrSize	表头大小(字节)
OffSize	offsize	绝对偏移量(0)大小

当执行读字体设置文件时, 此文件用自己所包含的代码去验证版本号, 倘若格式和版本号改变了, 旧的执行文件将拒绝新的版本。如果主版本号通过了执行文件的验证, 可以继续进行读字体的工作。次版本号对不被执行文件查出的格式进行了扩充。

当定位名称索引时, 会用到 hdrSize 字段。HdrSize 字段的给出使得 CFF 未来版本将以

一种和旧执行文件兼容的方式输入追加的数据到 OffSize 字段和名称索引之间。OffSize 字段指定了所有与 CFF 数据的开始有关的位移量(0)的大小。

(四) 字典和索引数据结构的信息

1 .字典数据结构信息

包含关键字-值对的字体字典数据用一个压缩权值格式来表示。这种格式非常类似于用于表示 Type1 字符串的格式。字典关键字由单字节或双字节指令来编码。字典值用可变字节的数值操作数表示。其中操作数可以是整数，也可以是实数。指令在前，操作数在后, 操作数给出指令值。因此，字典是由指令和操作数组成的简单序列。

实数操作数从一个 30 的字节值开始, 并跟随可变长度的字节序列。一个实数由一个(或 2 个)0xf 半字节终止, 使得可以填充字节尾部。因而数值 -2.25 由字节序列 (1e e2 a2 5f) 编码, 数值 0.140541E-3 由序列 (1e 0a 14 05 41 c3 ff) 编码。

指令和操作数可以查看它们的头字节来区别。0-21 指定为指令。28, 29, 30 和 32-254 指定为操作数。22-27, 31 和 255 被保留。

指令可以有一个或多个类型的操作数，但是这多种类型操作数都是由基本类型整数和实数构成的。这多种类型列在表 4 中。

表 4 操作数类型
Table 4 Operand Types

类型	说明
数值	整数或实数
布尔量	值为 0 和 1 的整数类型
SID	串标识码
数组	一个或多个数值
delta	数值或者数值的 delta 编码数组

Delta 中第二和后续数值是根据后续值之间的不同来编码的。例如: 一个数组 a0, a1, ..., an 将编码为: a0(a1-a0) (a2-a1) ..., (an-a(n-1))。

2 个字节指令有一个初始的 12 位长的换码字节。
字典数据的压缩是通过建立各种各样的字典关键字的缺省值来实现。
下面简单介绍一下顶层字典和私有字典的内容：

字典数据分离到顶层标准和私有字典的过程反映了 Type1 的用法。在这个过程中, 顶层字典数据在查找字体时间内被分析过后, 建立一个有效的 Type1 字体字典, 私有字典指令值则给出在字体生成之后的大小和偏移量。

顶层字典索引包括了 FontSet 中所有字体的顶层字典, 并把它们存储在索引数据结构中。总结起来: 字体通过名称索引结构的入口来进行确定; 字体的数据是通过对应的顶层字典结构来访问。(包含在顶层字典索引内的对象在顺序和数值上相应地对应到名称索引结构中的对象。其中每个对象都

是一个对应到 PostScript 字体的顶层字典的字典数据结构。)

一些顶层字典指令名称与对应的 Type1 字典关键字是一样的。CIDFont 需要额外的顶层字典指令。

2. 索引数据结构信息

索引数据结构是不定长对象数组。它由一个头部、一个偏移数组和对象数据组成。偏移量数组存放的是对象信息的偏移地址。根据该偏移地址,可以得到对象。对象的索引过程:首先索引偏移量数组,然后在指定的偏移地址取得对象。对象的长度是由偏移数组中下一个偏移量减去当前偏移量来确定的。追加的偏移量加在偏移量数组的尾部,这样最后对象的长度就能确定了。偏移量数组中的第一个元素永为 1。保证了每个对象都有一个对应的非 0 偏移量,而且还保证了动态装入的有效执行。空索引数据结构由一个带 0 值的计数字段(Card16)来表示,而且没有追加的字段。因而一个空的索引数据结构总的大小是 2 字节。

(1) 名称索引

名称索引中包括了 FontSet 中所有字体的 PostScript 语言名称(FontName 或 CIDFontName),并存储在索引结构中。字体的名称已经排过序,因此在 FontSet 内定位一个指定的字体时,可以进行二进制查询。排列的次序是基于用 8 比特无符号整数表示的字符编码。名称索引中至少有一个入口,也就是 FontSet 至少包含一个字体。

考虑到和客户软件的兼容性,例如 PostScript 解释程序、Acrobat,字体名称将不再长于 127 字符,不再包含下面一些 ASCII 字符:[,], (,), {, }, <, >, /, %, 空字符, 空格, 制表符, 回车, 换行, 换页。建议字体名称被限制在代码 33 到 126 为可打印的 ASCII 子集上。ATM(Adobe Type Manager)软件加强进一步的限制使得字体名称长度在 63 个字符以下。

从 FontSet 中删除字体时,不是移去字体的数据,而是通过设置名称索引中字体名称的第一个字节为 0(NUL)来实现。这种类型的删除方式在没有重建整个 FontSet 的情况下,提供了一个简单的方法实现了字体的升级。二进制查询软件必须在索引数据结构中查出删除项并且重新查询前后项,以致确保所有适合的名称被匹配。

(2) 字符串索引

除去用在名称索引中的 FontName 字符串和 CIDFontName 字符串的所有字符串都由 FontSet 中的不同字体使用,而且都存储在索引结构中,用 2 字节无符号数,也就是 SID(串表识码)来引用。字符串索引表中的字符串都是唯一的,以免在字库中出现重叠。节省空间的办法是:给经常出现的串定义预定义的 SID。这些经常出现的字符串也称为标准串,描述了除了和 Type1 字库共有的一小部分串以外所有的用在 ISO(国际标准化组织)Adobe 和 Expert^①字符集中的名称。

客户程序将包含 nStdstrings 个元素的标准串数组。因此,标准串是 0 到(nstdstrings-1)范围的 SID。字符串索引结构的第一个串对应到值为 nStdstrings(第一个 non-standard string 非标准串)的 SID,以此类推。当客户需要确定一个 SID 的串时,它需做以下的工作:测试是否 SID 在标准范围中,如果是那么到内部表中访问串。否则,用(SID- nstdstrings)的值做为索引在字符串索引结构中访问串。SID 定义为一个 2 字节无符号数,但是限制在 0 到 64999 范围内取值。带有 0 个非标准串的 FontSet 用空索引结构来表示。

像蒙文这样一个名义字符对应多个显现字符的问题,使用 Open Type 是最好的选择。因为 Open Type 就能处理少数民族文字中出现的这种问题。如果我们掌握了涉及到 Open Type 的所有字体格式,那么蒙文的 Open Type 字体的建立将指日可待。通过本文介绍的 CFF 字体文件格式,用户可以为进一步理解 Open Type 字体文件格式打下基础。

注释

① “Expert” 字符一般指很小的大写字母、旧样式轮廓、分数等等。

参考文献

[1]<http://partner.adobe.com>

[2]Adobe Technical Note #5176:The Compact Font Format Specification

[3]Adobe Technical Note #5177:The Type2 CharString Format

Research of the Compact Font Format

Narisu¹, Ochir²

(School of Computer Science ,Inner Mongolia University, Inner Mongolia, Hohhot, 010021)

Abstract: The CFF (Compact Font Format) is a font file format that used in Open Type font at present. This font file format was designed to be used primarily with Type 2 charstrings (character description procedures), It's main feature is minimizing the amount of storage needed for font programs. This paper analyzed CFF font file structure in detail, and explained importance process in file. It will provide basic knowledge for Open Type researching and developing Mongolian Open Type fonts.

Key word: CFF font; Type2 font; Type1 font; CID-Keyed font

收稿日期: 2004-5-14;

基金项目: 教育部人文社会科学重大项目“蒙古文信息处理平台(MIPP)的研究”(02JAZJD850003); 国家自然科学基金项目(60163003)

作者简介: 1. 那日松(1980-), 女, 蒙古族, 内蒙古兴安盟扎旗人, 内蒙古大学计算机学院 2002 级研究生。

2. 敖其尔(1941-), 男, 蒙古族, 内蒙古兴安盟人。内蒙古大学计算机学院教授, 内蒙古大学蒙古学研究中心兼职教授, 硕士生导师, 主要研究蒙文信息处理和计算语言学。