[3] Rodney Van Meter, "A Brief Survey of Current Work on Network Attached Peripherals", Operating Systems Review, Vol. 30, N。1, Jan. 1996, 63 – 70, Available at http://www.alumni.caltech.edu/~ rdv/

[4] Garth A. Gibson, David F. Nagle," File Server Scaling with Network-Attached Secure Disks", Proceedings of the ACM International Conference on Measurement and Modeling of Computer System

[5] Eric Burgener, "Storage Area Networks, The fabric of SANs looms in your future", available at http:// www.performancecomputing.com/features/9901f1.shtml

[6] Michael Peterson, Strategic Research Corp., "Storage Area Networking, The Next Network", available at http://www.sresearch.com

[7] Aloke Guha, "The Evolution to Intelligent Storage Area Networking", Storage Technology Corporation, available at http://www.storagetek.com/StorageTek/ network/StorageNet/WhitePapers/san/

[8] Tom Clark, "Designing Storage Networks with Fibre Channel Switches, Switching Hubs, and Hubs", available http://www.networkbuyergude.com/search/ 309008.htm

[9] Aloke Guha, Storage Technology Corporation, "The Evolution to Intelligent Storage Area Networking", available at http://www.storagetek.com/StorageTek/network/

[10] Benner, A. F., "Fibre Channel: Gigabit Communications and I/O for Computer Networks", McGraw Hill, New York, 1996

Sun Raster 格式图像文件数据存储结构详解

The Data Structure Description of Sun Raster Image File

邵才瑞 张福明(石油大学资源系, 山东省东营市, 257062)

SHAO CaiRui & ZHANG FuMing(Dep. of Oil Resource, Petroleum Univ., Dongying Shandong, 257062)

【摘 要】 本文详细阐述了Sun Solaris 中Raster 格式光栅图像文件存储结构, 编写了解编程序, 为异类平台上不同典型格式图像文件的转换和应用提供了技术基础。

【关键词】 Sun Raster(.RS); 图像文件; 存储结构

【ABSTRACT】 This paper describes the data structure of Sun Microsystems Raster image file. Programming confirms the data structure analyzing is correct. Also it establishes the foundation for sharing and processing different format images in different operation systems.

【KEYWORDS】 Sun Raster(.RS); Image File; Data Structure

1 引 言

Raster(.RS) 格式图像是Sun Solaris (或Sun OS) 支持的一种典型格式, 但却难以直接在Ms Windows 系统下操作。虽然一些图像工具可以实现两种不同格式的转换, 但由于在转换过程中会丢失某些图像信息而难以达到无失真转换。鉴于Sun Solaris 系统占有一定的系统平台份额, 因此破解.RS 格式图像文件的数据存储结构, 实现图像格式间的保真转换, 使非标准格式图像可跨平台使用成为必要。

2 存储结构

Sun Raster 是Sun Microsystems 公司独有的光栅格式图像。在Sun Solaris 上得到良好的支持, 但其它支持系统较少。该格式图像像素可以是1 位深(单色图像, 颜色表可有可无), 或8 位深(灰度或彩色图像), 也可以是24 或32 位深(高8 位被忽略, 为真彩色或每个像素带RGB 色彩分量的彩色图像), 很少有4 位深的。

2.1 数据文件组成及其物理存储结构

.RS 文件结构类似Ms Windows 中的.BMP 格式, 从文件头到文件尾一般依次由文件头、颜色表, 像素点阵数据三大部分组成, 物理结构如图1 所示。

文件头(占32个字节)	颜色表(真色彩无颜色表)	像素点阵数据(像素每行从左向右写, 各行从上到下逐行排列)
←—文件头		文件尾—→

图1 Sun Raster 格式文件存储结构

2.2 各部分详述

2.2.1 文件头 此部分包括8 组4 字节的整数(Sun 工作站中, 字节顺序为高字节存储在先, 并且以4 个字节存放一个整数), 长共32 个字节, 依次排列如表1 所列内容。

表1 Sun Raster 图像文件头结构

偏 移(字节)	名 称	描 述
0	ras _magic	文件标识符(魔数), 0x59a66a95
4	ras _width	用像素表示的图像宽度
8	ras _height	用像素点表示的图像高度
12	ras _depth	每个像素所用的位数
16	ras _length	像素点阵数据字节总数(即文件长度减去文件头和色彩表的字节数)
20	ras _type	编码类型, 见表2。
24	ras _maptype	彩色图颜色表类型(见表3)
28	ras _maplenth	彩色图颜色表的字节长度(见下面的说明)

魔数既可标志文件为一个.RS 文件, 同时又可标

〔收稿日期〕 1999-11-08

[作者简介] 邵才瑞(1966-), 男, 讲师, 1990 年毕业于石油大学应用地球物理专业。现从事软件开发和信息处理等教学科研工作。

志文件字节顺序。如果魔数为 0x956aa659, 则文件就是以逆字节顺序存储, 文件中所有多字节字段必须进行字序交换。

表 2 Sun Raster 图像的编码(type) 类型

码 值	描 述
0	旧文件(RT _OLD)
1	标准文件(RT _STANDARD) , 像素点阵数据无压缩
2	RLE 文件(RT _BYTE _ENCODED), 像素点阵数据有压缩
3	(RT _FORMAT _RGB)XRGB 或 RGB 而不是 XBGR 或 BGR
4	(RT _FORMAT _TIFF) 嵌入的 TIFF 文件
5	(RT _FORMAT _IFF) 嵌入的 IFF 文件
65535	(RT _EXPERIMENTAL) 各种实验格式

RT _OLD 和 RT _STANDARD 文件是大体相同的, 区别只在 RT _OLD 文件常常把头中的长度字段 ras _maplength 留为 0。

2.2.2 *颜色表* ras _maptype 有 3 种彩色格式, 如表 3 所示。

表 3 Sun Raster 图像的颜色类型

码值	描 述
0	(RMT _NONE) 无颜色表
1	(RMT _EQUAL _RGB) 表示 RGB 三原色, 每个原色分量 8 位。
2	(RMT _RAW) 无颜色表, 像素点阵数据为真色彩。

黑白图像不需要颜色表(RMT _NONE) , 真色彩也无颜色表(RMT _RAW) 。如果没有颜色表, 对于 24 位面或 32 位面数据为直接色彩, 对于 1 位面或 8 位面数据应构造颜色表。如果类型为 RMT _EQUAL _RGB, 则为彩色图像, 颜色表是长度和为 ras _maplength 字节的 3 个原色数组。第 1 个为红色值数组(物理存放为: 文件头结束~ ras _maplength/ 3) , 第 2 个为绿色值数组(物理存放为: ras _maplength/ 3~ 2* ras _maplength/ 3) , 第 3 个为蓝色值数组(2* ras _maplength/ 3~ ras _maplength) , 分别对应于每个颜色的 3 原色值 (RMT _EQUAL _RGB) 。颜色表应该包含 2^{ras_depth} 个表项。颜色表大小可以是 2, 这时它对应着像素所使用的两种颜色。设颜色表阵列定义为由红(r[ ]) 、绿(g[ ]) 和蓝(b[ ]) 3 个向量 unsigned char * r, * g, * b 组成, 颜色表 3 原色数据的物理存放顺序依次为:

首先是红(r[ras _maplength/ 3]) 即(ras _maplength/ 3) 长度的红色值;

然后是绿(g[ras _maplength/ 3]) 即(ras _maplength/ 3) 长度的绿色值;

最后是蓝(b[ras _maplength/ 3]) 即(ras _maplength/ 3) 长度的蓝色值。

物理存储顺序如图 2 所示:

<table>
<tr><td>r[0]</td><td>r[1]</td><td>…</td><td>g[0]</td><td>g[1]</td><td>…</td><td>b[0]</td><td>b[1]</td><td>…</td></tr>
<tr><td colspan="3">红(ras _maplength/ 3)</td><td colspan="3">绿(ras _maplength/ 3)</td><td colspan="3">蓝(ras _maplength/ 3)</td></tr>
</table>

图 2 颜色表存储结构示意图

2.2.3 *像素点阵数据* 紧随在彩色对应表(如果有) 之后是像素点阵数据, 像素在每行中的顺序是从左向右写, 而各行则从上到下排列。每行填充成 16 位的整数倍。对于 1 位深的图像, 每字节 8 位表示 8 个像素, 每个像素占 1 位, 最高位表示最左像素点(注意 0、1 的颜色表示) 。8 位无压缩每字节表示一个像素, 每字节的十进制数字表示颜色表中红(r) 、绿(g) 、蓝(b) 3 原色的下标值。对 24 位深或 32 位深的图像, 用 3 个 8 位字节分别表示每个像素的 r、g、b 值。在 32 位图像中, 忽略每个像素的高字节(第 1 个字节) , 通常为 0。24 位或 32 位忽略第 1 字节后保存下来的字节顺序通常为蓝、绿和红, 除非 ras _type 为 RT _FORMAT _RGB, 后一种情况 3 原色排列顺序为红、绿、蓝(一些图像的红和蓝位置相反, 但不用格式码指明) 。

如果编码类型是 RT _BYTE _ENCODED, 则图像点阵数据用简单的 RLE 方法压缩, 文件头中的长度字段表示压缩图像的字节数(未压缩图像的大小可以从头中的高、宽和深等 3 个字段计算得到) 。

RLE 方法压缩用序列 128、N、M 表示 N 个相同数值的字节序列, 每个字节的值都为 M, 值为 128(即 0x80) 的字节, 用字节序列 128, 0(即 0x80, 0x00) 表示。即除 128 以外的任何字节都表示数据自身。压缩时以字节为单位(不是以像素点为单位) , 行程码的长度可以为 1、2 或 3 个字节, 其解压算法如下。

¹ 读入 1 个字节 byte1;

④若 byte1 不等于 0x80(128) , 则 byte1 为原始图像数据, 转到¹ ;

㈣若 byte1 等于 0x80(128) , 则继续读入 byte2。若 byte2= 0, 则说明 byte1 为原始图像数据 0x80, 未做压缩, 转做¹ ;

¼ 若 byte2 不等于 0, 则 byte1 为压缩标志, byte2 为重复次数, 再读入 byte3 表示重复的图像数据。将 byte3 重复输出 byte2+ 1 次, 就得到了原始图像数据。

3 文件解编程序简述

以下是对 Sun Raster 格式图像数据读取的主要程序步骤:

```
# include < pixrect/ rasterfile. h>
# define RAS _RLE 0x80
static int rle _read ( byte * , int, int, FILE * , int) ;
```

```
structure rasterfile sunheader;
/ * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * /
int ReadSunRas( char * fname, byte r[ 256] , byte g[ 256], byte b[256], byte
* image)
  { FILE * fp;
   int linesize, size, csize, i;
  byte * line;
  / * 打开文件* /
  fp = fopen(fname,“ rb”) ; if (! fp) printf(“ unable to open file”) );
  / * 第一, 读入文件头* /
fread (&sunheader, sizeof( struct rasterfile) , 1, fp);
/ * 根据头信息进行判断, 以确保输入的文件可以处理。主要判断魔
数标志是否正确, 深度 sunheader. ras _ depth 是否为 1、8、24 或 32, 图像
类型 sunheader. ras _type 是否吻合, 颜色表类型 sunheader. ras _ maptype
是否可操作等, 若可操作则继续, 否则退出* /
     . . . . . . / * 具体操作略* /
/ * 第二, 读入颜色表* /
/ * 定义颜色表大小* /
   csize = (sunheader. ras _maptype = = RMT _NONE) ? 0 : sunheader. ras
_maplength;
/ * 颜色表为三原色, 且颜色表大小已知 * /
if (sunheader. ras _maptype = = RMT _ EQUAL _ RGB && csize) {
  fread (r, (size _t) 1, (size _t) sunheader. ras _maplength/3, fp);
  fread (g, (size _t) 1, (size _t) sunheader. ras _maplength/3, fp);
  fread (b, ( size _t) 1, (size _t) sunheader. ras _maplength/ 3, fp); }
/ * 为真彩色, 且存在颜色表大小, 跳过颜色表不用 * /
   else if (sunheader. ras _ maptype = = RMT _RAW && csize) {
     fseek (fp, ( long) csize, 1); }
/ * 对于 1 位和 8 位, 若无颜色表则构造隐含颜色表 * /
   else { /* 1 位深, 构造黑白色* /
     if ( sunheader. ras _depth = = 1) {r[0] = g[ 0] = b[ 0] = 0;
                                      r[1] = g[1] = b[1] = 255;}
     / * 8 位深, 构造 256 种隐含颜色* /
     else if (sunheader. ras _ depth = = 8){ for ( i = 0; i < 256; i+ + )
                                                r[i] = g[ i] = b[ i] = i;}}
/ * 第三, 读入像素点阵数据 * /
/ * 定义图像数据内存字节大小。1 位和 8 位深图像每字节表示 1 个
像素点的颜色索引, 24 和 32 位用 3 字节表示 1 个像素的RGB值* /
  / * 定义图像像素点阵字节大小* /
  size = sunheader. ras _width * sunheader. ras _height ;
  if ( sunheader. ras _depth = = 24 || sunheader. ras _ depth = = 32) size
= size * 3;
  / * 定义每行图像数据字节大小* /
  linesize = sunheader. ras _width * sunheader. ras _depth;
  if ( linesize % 16) linesize + = ( 16 - (linesize % 16) );
  linesize /= 8;
  / * 开辟足够大的图像数据内存 * /
  image = (byte * ) malloc ( (size _t) size) ;
  / * 开辟一行数据内存 * /
  line = ( byte * ) malloc ((size _t) linesize);
/ * 按行读入图像数据 * /
for ( i = 0; i < sunheader. ras _height; i+ + ) {
  / * 若像素点阵数据采用行编码压缩, 则解码* /
  if ( sunheader. ras _type = = RT _BYTE _ENCODED) {
    if (rle _read (line, 1, linesize, fp, ( i= = 0)) ! = linesize) break; }
  / * 未压缩, 则直接读入 * /
  else {
    if ( fread (line, (size _t) 1, ( size _t) linesize, fp) ! = linesize) {
      free( image) ; free( line) ; fclose(fp) ;
      printf(“ file read error”) ; } }
    /* 按不同位面深度分别将每行数据填入图像数据内存 image
    中。* /      . . . . . /* ( 具体操作略) * / ; }
  free( line); fclose(fp) ; return 1;
}
/ * 行编码解编程序 * /
  static int rle_read ( byte * ptr, int size, int nitems, FILE * fp, int init)
  { static int count, ch;
    int readbytes, c, read;
    if ( init) { count = ch = 0; }
    readbytes = size * nitems;
    for (read = 0; read < readbytes; read+ + ) {
      if ( count) { * ptr+ + = (byte) ch; count- - ; }
      else { c = getc(fp) ;
        if ( c = = EOF) break;
        if ( c = = RAS _RLE) { / * 0x80 * /
        count = getc(fp) ;
        if ( count = = EOF) break;
        if ( count < 0) count &= 0xff;
    if ( count = = 0) * ptr+ + = c;
      else { if (( ch = getc( fp)) = = EOF) break;
          * ptr+ + = ch; } }
    else * ptr+ + = c;
      } } return (read/ size) ;
}
```

4 结束语

作者据上所述编程实现了 Sun Solaris 中. RS 与 Ms Windows 中. BMP 格式图像之间的保真转换, 证明上述对. RS 格式数据存储结构的认识是正确的。掌握不同格式图像文件数据的物理存储结构是对其进行正确的读写、处理、格式转换等操作, 实现非标准格式图像跨平台应用的必要基础工作。

参 考 文 献

[1] Sun Microsystems,《SunOS reference manual》, 1990

[2] David C. Kay and John R. Levine 著, 柏东译,《图形图像文件格式大全》, 学苑出版社, 1994