

二值图象压缩标准——JBIG2

边文奎¹ 何丕廉¹ 徐梦苏²

(天津大学计算机系,天津 300071)

(天津市电子计算机研究所,天津)

E-mail: bianwk@eyou.com

摘要 简要介绍 JBIG2 标准的特点、应用及其数据流格式,对其中用到的算术压缩算法与当前流行的 LZW 算法进行了比较。

关键词 JBIG2 数据压缩 算术编码 LZW 编码

文章编号 1002-8331-(2002)13-0103-02 文献标识码 A 中图分类号 TP391

Bi-level Image Compression Standard——JBIG2

Bian Wenkui¹ He Pilian¹ Xu Mengsu²

(Computer Science Dept. of Tianjin University, Tianjin 300071)

(Tianjin Electronic Computer Institute, Tianjin)

Abstract: This paper introduces a bi-level image compression standard — JBIG2 with its characteristics, applications and format of data stream. It also compares the arithmetic algorithm used by JBIG2 with the LZW algorithm.

Keywords: JBIG2, data compression, arithmetic coding, LZW coding

1 JBIG2 简介

联合二值图象专家小组 (Joint Bi-level Image experts Group-JBIG) 在 1993 年制定了 ITU-T T.82(ISO/IEC11544), 通常称之为 JBIG 或 JBIG1。JBIG1 的设计目标是支持无损或渐进编码。JBIG1 虽然也有无损压缩的能力,但其有损压缩的图象质量明显低下,图象还原后像素数量不及原图象的 1/4,所以几乎没什么用处。

1999 年 7 月该小组又制定 JBIG2。JBIG2 支持有损、无损和渐进编码,其设计目标是:无损压缩的性能超过现有的其他标准;有损压缩在取得比无损压缩更高的压缩比的情况下,具有几乎不可见的质量下降。

制定 JBIG1 和 JBIG2 标准的主要目的是替代 CCITT G3 和 G4 中的低效压缩算法。

对于扫描的文本或线条图形(line art),JBIG2 的压缩效率比 G4 大 10%~50%,而对于计算机生成的文本图象则要大 500%,对于由半色调或抖动生成的灰度图象则要好 2~30 倍。

JBIG 可以替代无损压缩方式的 JPEG 标准。在处理 2—5 像素的图象时,JBIG 可以比 JPEG 产生较好的效果;对于 6—8 像素的图象,二者有相同的效果。

JBIG 对噪声敏感,其压缩比会随着图象中的噪声增大而减小。

JBIG 标准具有如下特色:

(1)支持质量渐进编码(quality-progressive coding)和内容渐进编码(content-progressive coding)。质量渐进是指从低质量到高质量(或无损)渐进,内容渐进指 JBIG2 压缩文件中可以加

入不同类型的图象数据,比如文本和半色调数据。

(2)典型的 JBIG2 编码器可以将输入图象根据内容不同分解为几个不同的区域(region),并对每个区域分别用不同的方法进行编码。这种基于内容(content-based)的分解在交互式多媒体应用中尤为需要。

(3)JBIG2 可处理一系列图象(多文档),即可以将多个图象压缩到一个 JBIG2 文件中。

(4)JBIG2 标准只规定了输出数据流(压缩文件的格式)和解码器的工作流程,而没有明确定义一个标准的编码器,它为高明的编码器设计者提供了灵活施展的空间。

JBIG2 的应用:

a. 对位流数据的实时压缩与解压缩,如传真图象压缩,无线数据传输等。

b. Internet 应用,如 WWW 中的图象编码,远程会议等。

c. 文档图象化(Document imaging),如对纸张文档(paper-based documents)的扫描、存储和管理。

d. 假脱机打印(spooling)。

需要注意的是 JBIG1 和 JBIG2 是二值静态图象(bi-level still picture)的压缩标准,如果要处理多种颜色或动态图象,则需使用其它标准,例如 JPEG2000、MPEG2 等。

2 JBIG2 文件格式

JBIG2 标准规定了输出的比特流(即压缩文件)格式。从逻辑上讲,JBIG2 文件由页(Page)构成,每个页又分为若干段(Segment);从物理存储结构看,JBIG2 文件由文件头(File

基金项目:国家自然科学基金(编号:69783004);天津市自然科学基金(编号:993800111)

作者简介:边文奎,硕士研究生,主要从事计算机图象处理研究。何丕廉,教授,博士生导师,主要从事人工智能、CAI、计算机图象处理及神经网络方面的研究。主持多项国家自然科学基金和天津市自然科学基金项目。徐梦苏,高级工程师,硕士生合作指导教师,主要从事办公自动化和数字图书馆的研究。

Header) 段头(Segment Header)和段数据(Segment Data)构成,页结构并不显式出现。JBIG2 文件有两种独立的文件组织形式和一种非独立形式。JBIG2 标准推荐压缩文件的后缀名为“.jbig2”或“.jb2”。

2.1 顺序方式(Sequential organization)

这是一种独立的文件组织形式,用于流应用程序,解码器可以保证从流开始到流结束,解码中间的每个比特。文件结构为:

文件头
1# 段头
1# 段数据
2# 段头
2# 段数据
...
N# 段头
N# 段数据

图1 JBIG2 文件格式——顺序方式

各段一定要按照段号增大的顺序存放。

2.2 随机访问方式(Random-access organization)

这也是一种独立的文件组织形式,用于随机访问应用程序,解码器可以按任意顺序解码文件的各部分,比如解码所有偶数页之前先解码所有奇数页,或根据用户的输入解码某些单独页。文件结构为:

文件头
1# 段头
2# 段头
...
N# 段头
1# 段数据
2# 段数据
...
N# 段数据

图2 JBIG2 文件格式——随机方式

注意最后一个段必须是文件结束段(End of file segment)。否则,解码器不能判定何时读入最后一个段头,以确定段数据部分的开始。各段也必须按段号增大的顺序存放。

2.3 嵌入方式(Embedded organization)

这不是一种独立的文件格式,而是依赖于携带 JBIG2 段的其它文件格式。每个段存储时,段头与段数据部分放在一起。嵌入文件格式允许以任意顺序存放各段,还允许用任意数据将各段隔开。

应用程序可在 JBIG2 数据的前后各用两字节加以标记,以便从其它数据流中检测到 JBIG2 数据。建议使用 0xFF 0xAA 作为开始标记,0xFF 0xAB 作为结束标记,这些标记不被当作 JBIG2 文件的一部分。

JBIG2 中每个段都有特定类型,该类型指定了与段相关的数据的类型,也严格限制了该段所能引用的段以及引用该段的段类型。规定的段类型有 21 种,如立即一般区域段(Immediate generic region),页信息段(page information),页结束段(End of page),文件结束段等等。

有两种编码方法可供选择:

(1)基于像素的顺序编码法(sequential coding):该方法使用算术编码(arithmetic coding)和一个模板。模板用来决定编码状态。

(2)基于像素游程的哈夫曼编码(Huffman coding of runs of pixels)这种方法速度可以加快,但压缩率会降低。该技术也用于 MMR 算法,该算法在 ITU-T 推荐标准 T.6(G4)中有描述。

下面仅就算术编码方法进行重点介绍。

3.1 算术编码的基本原理

虽然算术编码与大家熟悉的 Huffman 编码都是基于概率的,但具体实现不同。算术编码的基本思想是用一个浮点数(大于等于 0,小于 1)代表一个输入符号流,而不是 Huffman 编码中用一个特定代码代替一个输入符号。信息越长,所需浮点数的位数越多。编码时,根据概率为每个符号在[0,1)上分配一个范围空间,概率越大,范围越大。这些范围空间互不交叉。初始范围为[0,1),每编码一个符号,就根据该符号的概率修改当前的范围,使其越来越“窄”,当一个符号流被编码后,就得到了一个[0,1)之间的浮点数。

例如,符号串“ARITHMETIC”共 10 个符号,每个符号所占概率见表 1 第二列,为每个符号分配的范围见表 1 第三列:

表 1

符号	概率	范围
A	1/10	[0,0.1)
C	1/10	[0.1,0.2)
E	1/10	[0.2,0.3)
H	1/10	[0.3,0.4)
I	2/10	[0.4,0.6)
M	1/10	[0.6,0.7)
R	1/10	[0.7,0.8)
T	2/10	[0.8,1.0)

编码算法为:

```
Low=0.0
High=1.0
WHILE 输入流中还有符号 DO
  读入一个符号 symbol
  Code_range=High-Low
  High=Low+Code_range*HIGH_RANGE(symbol)
  Low=Low+Code_range*LOW_RANGE(symbol)
END OF WHILE
输出 Low
```

表 2 为编码“ARITHMETIC”的情况:

表 2 对“ARITHMETIC”的编码过程

输入符号	Low	High
	0.0	1.0
A	0.0	0.1
R	0.07	0.08
I	0.074	0.076
T	0.0756	0.076
H	0.07572	0.07576
M	0.07596	0.07600
E	0.075968	0.075972
T	0.0759712	0.075972
I	0.07597152	0.07597168
C	0.075971536	0.075971552

最后的输出结果为 0.075971536,该浮点数就代表了输入
(下转 223 页)

信协议数据库、界面信息等,而辅助线程负责监视串口信号、串口数据接收和串口数据发送。用主线程来专门处理信息,使程序能迅速响应命令和其他事件,用辅助线程来完成费时的工作,这样可以避免由主线程处理这些工作时阻塞程序信息的处理。DNC 计算机与数控机床通信的具体过程如图 4。

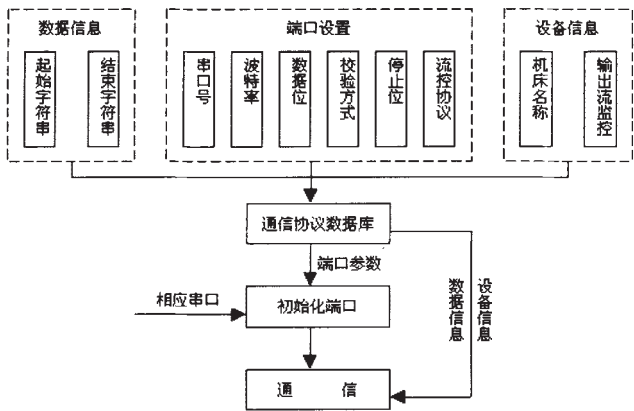


图 4 DNC 计算机与数控机床通信的具体过程

4 小结

(上接 104 页)

流“ ARITHMETIC”。

解码过程与编码过程正好相反,它是一个根据解码符号把范围空间不断变大的过程。

3.2 JBIG2 标准中算术编码的实现

对于以上的举例,在实际应用中有两个问题需要解决(1)如何表示一个非常长的浮点数(2)对于流式应用,在没有完全读入所有符号的情况下,如何提前知道每个读入符号的概率?

JBIG2 标准中规定初始范围为[0.75,1.5),用整数 0x8000 表示小数 0.75。这样,在编码过程就可以使用固定精度整数算术,节省了大量的浮点运算时间。每当间隔 A 的值低于 0x8000 时将其乘 2,使其保持在 $0.75 \leq A < 1.5$ 。当 A 每次加倍后,代码寄存器也加倍。为了保持代码寄存器不溢出,周期性地将其最高一个字节移出,放到一个外部压缩数据缓冲区中。这样就解决了第一个问题。

JBIG2 中有一个“模板(template)的概念。所谓模板是指一个被编码像素的一些相邻像素,这些相邻像素都是已经读入的。例如,如果一幅图象按从上到下,从左到右的顺序读取像素的话,则相邻像素是指被编码像素点上方、左方的像素。相邻像素的值就形成了一个上下文(Context),每个上下文都可以由 JBIG2 概率估算机(Probability estimator)得到一个适应性概率估计(adaptive probability estimate),供算术编码器使用。这样第二个问题就解决了。

3.3 与 LZW 压缩算法的比较

当前流行的编码方法从原理上基本可分为两大类,一是基于统计(statistical)或概率的方法,二是基于字典(Dictionary)的方法。算术编码属于前者,而经 Welch 改进的 LZ 编码,即 LZW 编码属于后者。所以在基本原理上二者是不相同的。

LZW 方法用一个固定长度的代码(一般为 9~12 Bits)来代替一个不定长的串来达到压缩的目的。显然,输入流中符号的重复率越高,则压缩比越大;相反,如果符号的重复率很低,有

DNC 系统的研究重点已由过去的以通信技术向集成化生产管理转变。其关键技术将主要围绕提高 NC 机床利用率,缩短加工辅助时间,提高整个 DNC 系统的柔性,提高可靠性,降低工人劳动强度等方面。使用 VC 语言编制的 DNC 信息系统,实现制造过程的状态采集及 NC 数据传输,该系统具有结构简单、运行快速、工作稳定等特点,易于实施,对促进 DNC 技术的发展具有积极的意义。文章所述的 DNC 信息系统已在实际中得到了应用,并取得了良好的应用效果。

(收稿日期 2001 年 12 月)

参考文献

1.王时龙,孙命,陶桂宝.DNC 系统设备组动态重构技术的研究[J].机械工程学报,2000,36(12):81~84
2.张旭梅,刘飞,但斌.基于集成 DNC 技术的车间运行模式研究[J].中国机械工程,1999,10(3):321~323
3.熊斌,谭建荣,何余仁等.敏捷 DNC 系统的理论探讨[J].计算机集成制造系统——CIMS,1999,15(6):1~6
4.黄强,蔡建国.一种开放式 DNC 系统平台设计及开发方法[J].机械科学与技术,1999,18(4):678~681
5.毕承恩.现代数控机床[M].北京:机械工业出版社,1991

时则会产生膨胀现象。这是因为通常用 8 位来表示一个符号(如文本中的 ASCII 符号),而 LZW 编码时要用 9~12 位来表示一个符号。所以一般当输入流较大从而有较多的重复符号出现,LZW 压缩才会起作用。

如前所述,算术编码是用一个浮点数表示一个输入流,浮点数的位数随输入流长度的增加而增加,不会发生膨胀现象。

下面对 LZW 和算术编码的压缩情况进行比较。由于常见的图象格式中 GIF 文件是 LZW 压缩的,所以就以其作为代表,JBIG2 文件则由作者编制的编码器压缩而成。比较情况见表 3:

表 3 原图、LZW 压缩与算术压缩的比较

示例	BMP 图象(无压缩)	GIF 图象(LZW 压缩)		JBIG2 图象(算术压缩)	
	大小	大小	压缩比	大小	压缩比
1	69526	13734	80.2%	3603	94.8%
2	69526	34687	50.1%	17318	75.1%
3	69526	23888	65.6%	11725	83.1%
4	69526	16472	76.3%	9630	86.1%

示例 1 为线条图形,示例 2 为扫描的古籍书页,示例 3 为计算机生成的文本图象(用的是该文的第一页),示例 4 为扫描的标准 A4 纸公文(印刷体)。所有图象均设为 600*914 象素。从以上的比较可以看出,在二值图象压缩方面 JBIG2 的算术压缩方法比 LZW 方法要更好一些。对于扫描图象 JBIG2 文件可压缩到原始图象的 1/4~1/6 左右。(收稿日期 2001 年 10 月)

参考文献

1.Coding of Still Pictures[S].ISO/IEC JTC1/SC 29/WG1(ITU-T SG8), 1999
2.ITU-T Recommendation T.82[S].ISO/IEC 11544,1993
3.Arithmetic Coding + Statistical Modeling=Data Compression[J].Mark Nelson Dr.Dobb's Journal,1991
4.Arithmetic Coding.http://www.rasip.fer.hr/research/compress/
5.JBIG Compression.http://www.leadtools.com/home2/VertMkts/LTJBIG.htm