基于嵌入式系统的改进快速压缩算法

兵工自动化

2003 年第 22 卷第 1 期

软件技 m

SoftwareTechnique

o.1.Automation

2003,Vo1.22,No.1

文章编号:i006-1576(2003)0 卜 0046—03

基于嵌入式系统的改进快速压缩算法

刘存良,张秉权,黄河燕

(沈阳工业学院信息工程分院,辽宁沈阳 l10015)

摘要:根据嵌入式系统的特点,针对无损压缩算法存在压缩率非最佳,以及提高查找率和存储利用率等

问题,提出了一种改进的无损压缩算法.该算法利用最小匹配 Hash 表的存储结构,采用综合匹配法查找最

佳匹配,以降低压缩过程中的比较次数.即采用回溯匹配算法,在快速定位最小匹配(MINMATCH)的基

础上,当匹配长度大于最小匹配时才进行后续符号比较.在实际中,终止寻找匹配应符合:当前匹配长度小

于前一匹配长度,超过最远索引距离,达到最大匹配等 3 个条件.

关键词:无损压缩;压缩算法;嵌入式系统;Hash;综合匹配

中图分类号:TP391 文献标识码:A

AnImprovedFastCompressionAlgorithm

BasedonEmbeddedSystem

LIUCun-liang,ZHANGBing-quan,HuangHe—yan

(CollegeofInformationEngineering,ShenyangInstituteofTechnology,Sheny
ang110015,China)

Abstract:Aimingatproblemsbeinginlosslesscompressionalgorithmthatimpr
ovethesearchingand

storingratioandcornpressionefficient,anadvancedlosslesscompressionalgo
rithmisintroducedaccordingas

peculiarityofembeddedsystem.Inthisalgorithm,memorystructureofmin-ma
tchingHashtablewasused,

andthebestmatchingwassearchedwithmeansofthesynthesismatching.Thism
ethodusedcandecrease

compressiontimes.Basedonsearchingthemin-matchstringquickly,thebestm

atchingstringcanbefound

whenthecurrentstringislongerthanthepreviousone.Inpractice,iftheCUrrents

tringisshorterthanthe

previousoneorjfthematchinglengthislongerthanthefarmostindexdistanceori

fit'Sthebestmatching.

searchingoperationisended.

Keywords:LosslesscornPression;Compressionalgorithm;Embeddedsyste

m;Hash;Synthesis

matching

1 引言

移动通信时代到达的 21 世纪,嵌入式系统

给人们带来了新的思维空间,从单片机到智能手

持装置(如:手机,PDA),再到信息家电,可

以说进入了一个后 PC 时代.然而以现在市场上

的 PDA 为例,由于其硬件成本和实现技术上的

原因,其存储容量小和处理器的运行速度慢,很

大程度上成为其能否提高市场普及的"瓶颈"问

题.基于嵌入式系统的改进快速压缩算法在此背

景下提出,其目的是为了更有效的利用存储空

问,而不明显的影响处理器运行速度.

国内外的无损压缩算法及其变体很多,但总

体上可分为:统计式和字典式两类.前者主要是

根据单个符号重复概率大者用短码表示,重复概

率小的用长码表示,并以此特点来缩短平均码

长,其典型是 Huffman 编码算法.后者是基于

“滑动窗口”的 LZ77 和基于”字典”的 LZ78

两种算法:基于 LZ78 及其变体算法(典型的是

LZW 算法)最显着的特点是经过字典编码后无

需存放字典本身,而在译码过程会根据编码自动

生成字典,这提高了压缩速度,但压缩率不是最

收稿日期:2002—07—02;修回日期:2002—08-26

作者简介:刘存良(1977 一),男.山东人.2000 年毕业于沈阳工业学院,现沈阳工业学院信息分院在读硕士研究生,从事嵌入式系统中数据压缩技术研究.

?

SoftwareTechnique

o.I.Automation

2003,Vo1.22,No.1

佳:基于 LZ77 及其变体算法(如 LZSS,LZH)

可以在滑动窗口中找到最长匹配,从而提高压缩率,但在查找最长匹配过程中引入反复回溯比较,势必影响压缩速度.在对上述嵌入式系统特点和压缩算法分析的基础上,提出基于 LZ77 的改进算法一 LzHL.以下,首先给出一种新的 Hash 表建立,然后介绍相应的压缩算法一综合匹配法,最后进行 Huffman 编码存储.由于解压是压缩的逆过程,不再详述.

2Hash 表(字典)结构

文件在计算机中是以二进制存放,按照 ASCII 表有 256 种不同符号,如果 P.每后移都进行匹配比较,花费的时间可想而知.为了提高查找效率和存储利用率,采用最小匹配 Hash,这样只有在匹配长度大于 MINMATCH 才进行后继符号比较.数据结构如下:

其中:MINMATCH:最小匹配长度;

Pi:窗口中当前指针所指位置;

Ii:位置为 i 处的索引项,其值为最小匹

配符号进行位移操作所得值.

以图 1 窗口为例,MINMATCH=3.由 a,

b,c 分别左移直接组成索引项为 I..当后继”窗

口”再次出现 a,b,c 时,首先利用 Hash 方法

找到 I.,再进行比较后面的字符是否相等.其算

法描述如下:

..

I1

————————abbCbbbbabbbb

图 1Hash 表的存储结构

算法 1

Update—

Hash(Ii.WindowIl1)//建立索引项函数

{

li=((Ii)<<H—Shift)(WindowIl1))&

Hash——Mask)

}

其中:Window[i】:滑动窗口位置 i 处的字符;

HShift:每次左移的位数;

Hash—

Mask:防止溢出,Ii 应为正数.

3 改进的快速压缩算法

由于目前 PC 机内存大,处理器运行速度

快,压缩多采用回溯匹配法,即在已编码的窗口

中查找最大匹配.这样要扫描整个已编码窗口,

压缩速度大大降低.鉴于嵌入式系统自身的特点

及其局限性,提出综合匹配算法.

考虑当前待压缩的字符串 S,根据算法 1 可

以计算出索引项 Ii 的值.把所有索引值为 Ii 的位

置值放入表 PrePosition【】,在后继窗口中如果

出现的索引值与 Ii 相等,首先找到最近出现索引

值相等的位置,按照综合匹配法进行匹配.其数

据结构如下:

structConfiguration

{

unsignedshortgood—

length;//较好匹配长度

unsignedshortmax—

lazy;//懒惰匹配长度

unsignedshortnice—

length;//最大匹配长度

unsignedshortmaxchain;//最远索引距离

}Configl;

其中:

good—

length<===max—

lazy<:==nice—

length

在实际压缩过程中,首先判断是否已有一个

较好的匹配(good—

length).如果有,最远索引

距离可小一些.如果已有一个懒惰匹配(max

lazy),则无需再找其它匹配.当某一匹配大于

nicelength,则不再比较匹配,直接输出最大匹

配.所以终止寻找匹配的条件有三个:当前匹配

长度小于前一个匹配长度;最远索引距离等于

零:达到最大匹配.

算法 2

SynthesisMatching{

intIookahead=strlen(S);//S 为待压缩串

while(Iookahead!=0)

{

Update—Hash(1i,Windowli]);

//建立 Hash 表(字典)

//根据 structconfiguration 参数

//查找当前位置的最佳匹配

if(pre—length<max—lazy)

{

match—

lengthLonger—Match(hash—head);

}

//当前位置匹配与前一个匹配比较

//决定输出最终匹配

if(pre—length>MIN—

MATCH&&

match—

length<prey—

length)

{

Store(pre—position,pre—length)

?

47?

兵工自动化

2003 年第 22 卷第 1 期

软件技市

SoftwareTechnique

o.LAutomation

```
//存储前一个匹配位置及长度
Iookahead 一 prev—
length;
)
else
{
if(match—available)
//是否满足最小匹配的标志
{
Store(cur—position,cur—length);
//存储当前匹配位置及长度
Iookahead 一 cur_
length;
)
else
{
Store(0,window[iJ);//输出单个字符
Iookahead 一;
)
)
)
```

)

Longest_

Match(hash—head)//找最佳匹配函数

{

if(prev_length>good—match)

{

chain

—

length>>2;

)

do

{

://查找当前位置最好匹配

)while(Pre—PisitionIcur—pitionl>0&&

—一

max—

chain>0)

l

4 试验结果

由于压缩算法支持最小匹配 Hash,避开了

反复回溯查找比较,其时间复杂度降低.改进前

的时间复杂度为 O(n),而改进后的为 o(n).

因为对大多数信息而言,要编码的字符串往往在最近的上下文中更容易找到匹配串.而在查找匹配过程中,采用综合匹配算法使压缩率并没明显降低.考虑到与 l6 位的嵌入式系统兼容,本算法的压缩窗口为 32K,最长编码 l5 位,综合匹配参数 Configl={4,4,16,16}.对一些数据进行压缩的性能比较如表 1,表 2.

5 结束语

针对嵌入式系统的自身特点,在对现有压缩算法进行研究的基础上,提出了一种改进的快速压缩方法,明显降低了时间复杂度.可以认为,压缩率速度的提高为提高压缩率创造一定的条

?

表 1 数据样本统计结果

表 2 不同数据类型的压缩结果

文件大小 1K512K1M3M5M

英文文本(%)3743484948

中文文本(%)3642404645

中荚混合文本(%)3741444545

tlTML 页面(%)3135405756

BMP 位图(%)4759657472

注:

①开发环境:一款联想 PDA,存储器 16M,

主频 33MHz,操作系统 WindowsCE3.0.

②数据样本:在 1k～5M 抽取文件 50 个统

计结果.

③压缩率:(Ssize-Osize)/Ssize.其中

Ssize 是压缩前的大小:Osize 是压缩后的大小.

④以上数据样本的压缩率为平均值.文本

资料主要是中英词典,中,英文小说;图像资料

主要是主页图库的一些标准图像;页面资料包括

图像,文字均有.

⑤以上统计表明,本算法对文本压缩有较

好的效果.

件,如把查找节省下来的时间用于寻求更好的编

码和存储方法,以提高压缩率,这也正是以后需

要不断深入研究的方面.

参考文献:

IlJ 吴乐南.数据压缩 IZ1.2000.43—66.

J2JWelchAtechniqueforhigh-performancedata

compressionIS1.IEEEComputer,1984,6,17(6):

8 一 l9.

I3JFiala,E.R,Greene,D.H.Datacompressionwith

finitewindowsIS1.Commun,OftheACM,32(4):

490—505.

4Jiang,Jones.ParalleldesignofarithmeticcodingIS1.

1EEPr0ceedings—E:ComputerandDigitalTechniques,

1994,(11):327—323.