中文文本压缩的LZW算法

陈庆辉[1,2],陈小松[1],韩德良[1]
CHEN Qinghui[1,2],CHEN Xiaosong[1],HAN Deliang[1]

1.中南大学 数学与统计学院,长沙 410083
2.中南大学 商学院,长沙 410083
1.School of Mathematics and Statistics, Central South University, Changsha 410083, China
2.School of Business, Central South University, Changsha 410083, China

CHEN Qinghui, CHEN Xiaosong, HAN Deliang. Compression algorithm LZW on Chinese text. Computer Engineering and Applications, 2014, 50(3):112-116.

Abstract: This paper presents a compression algorithm for Chinese text which is improved from LZW algorithm. By modifying LZW algorithm's dictionary size, basic set and the output way of dictionary code, the improved algorithm LZW_CH demonstrates about 19% higher compression ratio than LZW19's with almost the same execution speed. LZW_CH doesn't need any pre-processing work for the compressing data. As a single compression algorithm, LZW_CH's compression with long Chinese text has closed or exceeded the professional compression utility WinRAR.
Key words: Chinese text; data compression; compression algorithm; encoding; LZW

摘 要: 结合中文文本中的汉字编码方式、大字符集以及重复字串不长三个不同于英文文本的结构特点对LZW算法从读取数据方式、基本码集和字典码值输出方式三方面进行了修改。改进后的算法对中文文本的压缩比平均比LZW19提高了19%且压缩和解压速度与后者相当,其对较长的中文文本的平均压缩比已接近或者超过了压缩软件WinRAR。
关键词: 中文文本;数据压缩;压缩算法;编码;LZW
文献标志码: A **中图分类号:** TP311 doi:10.3778/j.issn.1002-8331.1205-0383

1 引言

LZW算法是1984年Terry A.Welch[1]在字典压缩算法LZ78[2]基础上改进的一种通用压缩算法。其较快的压缩速度和对各种数据文件的良好适应性使得其很快成为LZ系列压缩算法[1-6]中最优秀的压缩算法之一。但文本数据,特别是以中文为主的文本数据,有着不同于其他类型数据的结构特点[7-8]。利用这些特点对原有的LZW算法进行修正,就可以提高算法对中文文本文件的压缩比。对LZW算法的中文文本压缩的应用而言,最简单的方法就是直接使用通用的LZW算法。但这方法的缺点在于通用算法的基本处理单位为字节,这样在压缩过程中会人为地割裂中文数据编码中蕴含的语义信息,从而降低了算法的压缩比。对此,国内学者徐秉铮[9]、华强[10]等人针对汉字的编码以及大字符集的特点,从算法读取数据的方式和修改算法的基本码集的方法对算法进行了改进。改进后的算法对中文文本的压缩比有一定的提高,但仍远低于LZW算法对英文文本的压缩比。本文将提出一种新的中文文本压缩算法。这种算法是在现有的LZW中文文本压缩算法的基础上进行改进的。改进后的算法有效地利用了中文文本中数据编码和汉字大字符的特点,提高了由于中文文本中重复字串不长导致压缩比远低于英文文本的压缩比例。

2 LZW算法基础和讨论

LZW压缩算法的核心是其在压缩过程中维护的一个转换表,下称字典。这个字典将输入的不定长度的字

基金项目: 中南大学自由探索计划(No.201011200121)。
作者简介: 陈庆辉(1990—),男,本科,研究方向:数据压缩;陈小松(1956—),通讯作者,男,教授,研究方向:代数与符号计算、密码及编码;韩德良(1988—),男,本科,研究方向:数据压缩。E-mail:xschen@csu.edu.cn
收稿日期: 2012-05-31 **修回日期:** 2012-07-15 **文章编号:** 1002-8331(2014)03-0112-05
CNKI **网络优先出版:** 2012-08-17,http://www.cnki.net/kcms/detail/11.2127.TP.20120817.1027.001.html

符串转换为一个固定长度的代码。算法压缩的流程如图1所示。算法在压缩过程中根据一定的规则不断地将编码器中首次遇到的字符串加入到字典中，并且为加入的字符串分配一个唯一的称为码值的标志值。

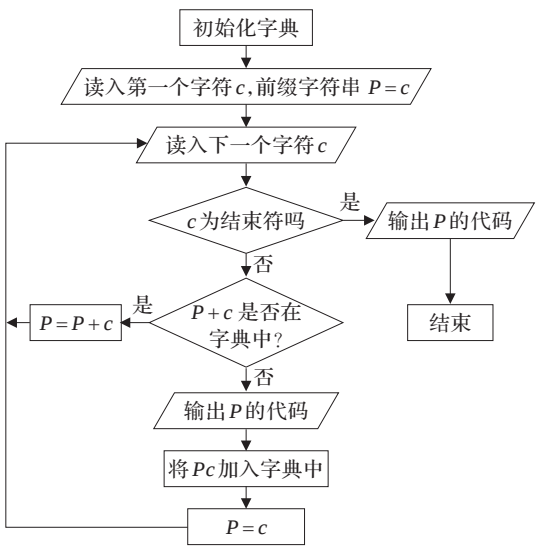

图1　LZW算法的压缩流程图

相对于压缩过程，LZW算法的还原过程如图2所示，还原过程关键在于其初始化的字典要与压缩程序一致，还原过程中维护的字典几乎与压缩过程同步。

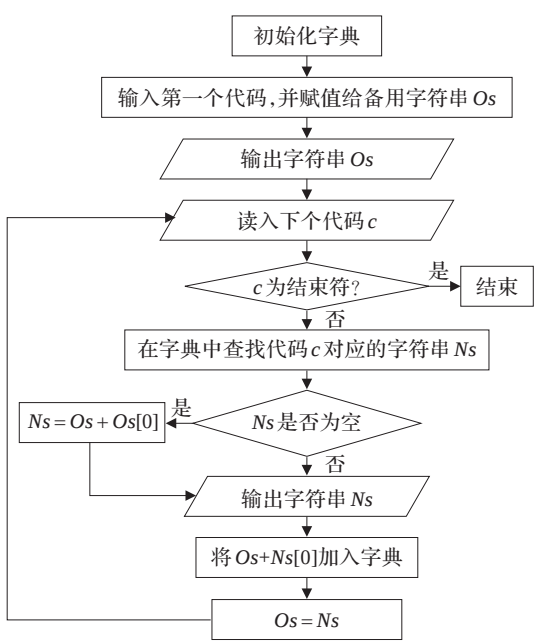

图2　LZW算法的还原流程图

LZW算法的压缩可分解成如图3所示的5个部分。这种流程的分解重点在于将模型和编码分开，模型处理得出的符号可以通过编码程序再次编码，这样可以根据模型处理得到的码值的频率分布特点对编码进行重编码，从而得到更好的压缩效果。

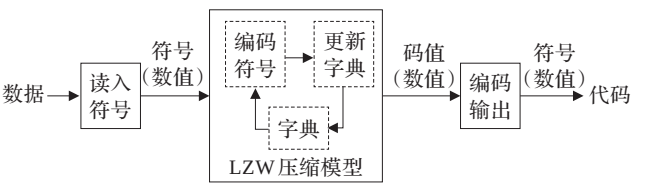

图3　LZW流程分解图

LZW文本压缩的参数修正。针对传统LZW对文本文件压缩存在的不足，提出了编码方式、字典大小和字典更新策略3个的修正参数，并对其进行了实验测试。

修正方式：第一，自适应扩位编码输出。编码输出的长度自适应于当前字典的长度，编码输出的长度随字典长度增大而增大。第二，字典大小的增大，字典越大字典中的长字符串也会越多，字典对文本的自适应性就越强，其压缩效果也会越好。第三，重建字典。程序在开始压缩时，记录输入字节数和输出字节数。用输出字节数除以输入字节数得到压缩过程的压缩率。字典在填满时，程序每压缩100个字节统计一次新的总压缩率，如果发现新压缩率比旧压缩率大时重建字典。若新压缩率比旧压缩率小，则将新压缩率赋值给旧压缩率。

针对上面提到的3个修正点，对传统的LZW算法作了一些相应的修正。

表1给出了传统的LZW算法在改变字典大小时，对部分英文文本和中文文本的测试结果。其中压缩率=压缩后的文件大小/压缩前的文件大小，算法LZW1_12、LZW1_14、LZW1_16和LZW1_19分别表示字典大小值为12、14、16和19。这里的字典大小是指程序输出代码的最大长度。从表1中的测试结果可以看出无论是对中文文本还是英文文本，字典越大，其压缩效果也越好。当然其压缩效果的改善是以内存消耗的增大为代价的。

表2给出了在可重建字典的情况下改变字典大小

表1　在不重建字典下字典大小对压缩率的影响

	英文文本					中文文本				
	爱玛	学生时代	所罗门王的宝藏	暮光之城	飘	做最好的自己	管理长歌行	论语	亮剑	平凡的世界
文件大小/KB	212	330	447	655	2 298	124	173	324	660	1 670
LZW1_12/(%)	46.98	45.67	50.45	48.23	48.31	74.58	73.78	81.80	79.40	76.95
LZW1_14/(%)	39.41	40.65	42.98	41.55	42.44	66.15	65.67	62.12	72.21	65.97
LZW1_16/(%)	38.27	38.33	39.08	38.18	38.54	62.40	60.39	55.78	65.63	59.88
LZW1_19/(%)	38.27	38.42	38.72	30.09	36.30	62.04	60.39	55.68	63.18	55.86

表2　可重建字典情况下字典大小对压缩率的影响

	英文文本					中文文本				
	爱玛	学生时代	所罗门王的宝藏	暮光之城	飘	做最好的自己	管理长歌行	论语	亮剑	平凡的世界
文件大小/KB	212	330	447	655	2 298	124	173	324	660	1 670
LZW2_12/(%)	45.18	45.96	48.19	47.21	47.48	72.14	71.75	68.26	78.41	73.91
LZW2_14/(%)	39.41	41.02	43.21	42.70	42.69	65.77	65.73	62.22	71.46	66.91
LZW2_16/(%)	38.27	38.33	39.19	38.18	38.69	62.04	60.39	55.78	65.96	59.99
LZW2_19/(%)	38.27	38.42	38.72	30.09	36.30	62.04	60.39	55.68	63.18	55.86

表3　重建字典对压缩率的影响

	英文文本					中文文本				
	爱玛	学生时代	所罗门王的宝藏	暮光之城	飘	做最好的自己	管理长歌行	论语	亮剑	平凡的世界
文件大小/KB	212	330	447	655	2 298	124	173	324	660	1 670
LZW1_12/(%)	46.98	45.67	50.45	48.23	48.31	74.58	73.78	81.80	79.40	76.95
LZW2_12/(%)	45.18	45.96	48.19	47.21	47.48	72.14	71.75	68.26	78.41	73.91
LZW1_14/(%)	39.41	40.65	42.98	41.55	42.44	66.15	65.67	62.12	72.21	65.97
LZW2_14/(%)	39.41	41.02	43.21	42.70	42.69	65.77	65.73	62.22	71.46	66.91
LZW1_16/(%)	38.27	38.33	39.08	38.18	38.54	62.40	60.39	55.78	65.63	59.88
LZW2_16/(%)	38.27	38.33	39.19	38.18	38.69	62.04	60.39	55.78	65.96	59.99

表4　LZW_CH1与LZW19测试结果的比较

文件名	三字经	小说	小说2	围城	巴黎圣母院	三国演义	平凡的世界
大小/KB	5	124	173	438	608	1 174	1 670
LZW_CH1/(%)	97.37	67.12	64.22	64.36	63.88	60.03	56.45
LZW19/(%)	77.45	62.04	60.39	61.64	63.94	59.90	55.86
较好	LZW19	LZW19	LZW19	LZW19	LZW_CH1	LZW19	LZW19

时,对部分英文文本和中文文本的测试结果。其结果同样是字典越大,压缩效果越好。结果说明对于普通的文本文件来说,字典越大其对文本的自适应性就越强,其压缩效果也会越好。

表3给出重建字典与不重建字典在不同字典大小的情况下,对部分英文文本和中文文本的测试结果。从测试结果可以看出,在字典为12位大小时,使用重建字典方式可以提高压缩效果,但当字典大小为14或16时,其压缩效果反而有略微的下降。这说明普通文本的上下文具有一定的相似性。因为在不重建字典的情况下,字典包含的信息只是在压缩的前面阶段建立的,在对文件后部分的压缩时,字典并没有更新,即并没对文件后部具有自适应性。

3　LZW中文文本压缩的算法改进

利用中文文本中不同于英文文本的结构特点,以算法LZW1_19为基础进行了改进。改进分别从算法的读取数据方式、基本码集以及字典码值的编码输出方式三个方面进行。

(1)读取方式的改进

为了使算法在压缩过程有效利用汉字编码的特点,改进后的必须具有两功能:第一,识别汉字编码;第二,对汉字进行重编码。识别汉字编码是因为汉字多为多字节编码。此时,如果算法读取数据时仍以单字节读取这样会破坏原来的汉字中蕴含的语义信息。对汉字进行重编码是因为汉字是个大字符集,如果直接以汉字为字符进行读取而不对读取的编码进行重编码那么字典的基本码集中的码值最大长度会很大,这样会导致算法压缩比的降低。汉字重新的编码方式与算法基本码集的确定(即字典的初始化方式)有关。

表4给出了LZW_CH1与LZW19对部分中文的测试结果。其中LZW19为上章中的算法LZW1_19,其最大字典编码长度为19位。LZW_CH1是在LZW19基础上对LZW19的读取数据方式进行修正,LZW_CH1读取数据时可以识别汉字,但并没有对其读取的汉字进行重新编码。所以,测试结果显示LZW_CH1对中文文本的压缩并没有产生比LZW19更好的压缩效果。

(2)基本码集的压缩

汉字是个大字符集,所以如果将所有的汉字都加入算法的基本码集(即初始字典)必然会使得字典码值的浪费。所以设计一种算法,尽可能地压缩基本码集,使程序充分利用字典的短位码区的编号成为了一种提高算法压缩率的有效方法。

基本码集的大小确定决定了汉字在初始字典中的

表5　五种算法对部分中文文本的压缩率比较

文件名	小说	小说2	围城	巴黎圣母院	三国演义	安徒生童话	亮剑	平均
大小/KB	123	172	437	608	1 173	1 733	659	700.71
LZW_CH2/(%)	53.16	51.68	55.03	56.20	52.97	45.83	55.61	52.93
LZW_CH3/(%)	53.24	52.18	56.02	56.19	54.54	46.64	55.83	53.52
LZW_CH4/(%)	52.43	51.24	54.93	56.22	53.16	45.81	55.48	52.75
LZW_CH5/(%)	52.23	51.10	54.86	56.16	53.12	45.80	55.42	52.67
LZW_CH6/(%)	53.12	52.66	56.30	57.53	53.82	46.27	56.67	53.77
最佳算法	LZW_CH5	LZW_CH5	LZW_CH5	LZW_CH5	LZW_CH5	LZW_CH5	LZW_CH5	LZW_CH5

表6　算法LZW_CH5与LZW_CH压缩比的比较

文件名	小说	小说2	围城	巴黎圣母院	三国演义	安徒生童话	亮剑	平均
大小/KB	123	172	437	608	1 173	1 733	659	700
LZW_CH5	1.91	1.96	1.82	1.78	1.88	2.18	1.80	1.91
LZW_CH	1.94	2.00	1.90	1.85	1.95	2.25	1.88	1.97
最佳算法	LZW_CH8	LZW_CH8	LZW_CH8	LZW_CH8	LZW_CH8	LZW_CH8	LZW_CH8	LZW_CH8

编号(即码值的分配)。字符在初始字典中的编号利用字符机内码与数据码的关系来的。本文给出了五种压缩码集的方案：方案一，将GB2312-80中的所有字符加入基本码集，得算法LZW_CH2；方案二，将方案一中的二级汉字和字符区去掉，得算法LZW_CH3；方案三，去掉方案一中的二级汉字，得算法LZW_CH4；方案四，只将一级汉字和1、3区的字符加入基本码集，得算法LZW_CH5。方案五，在方案四的基础上，预留975个码值给未定义的汉字，当遇到未定义的汉字再将码值加入基本码集中，得算法LZW_CH6。对部分中文文本的测试结果如表5所示。其中算法LZW_CH5压缩效果最好。

(3)字典码值输出方式的改进

针对中文文本重复字串不长的特点，结合字典码值分配的方式，本文提出了对字典码值进行再编码输出的改进方法。

按码值的最小二进制表示位数对字典进行分段，则字典可分为19段。其中码值 $2^{n-1}-1\sim2^{t}$ 为字典的第n段。

则改进算法的编码输出方式如下：

记当前字典中码值的最大长度为n，其中$11<n<20$。则

当$n<14$时，码值采用该码值大小的二进制编码输出，输出长度为n位。

当$n=14$时，对字典的1~12段的码值采用13位长度输出，其中第一位设置为0，后12位为该码值大小的二进制编码；对字典的13~14段的码值采用15位长度输出，其中第一位设置为1，后14位为该码值大小的二进制编码。

当$n<18$时，对字典的$1\sim n-3$段的码值采用$n-2$位长度输出，其中第一位设置为0，后$n-3$位为该码值大小的二进制编码；对字典的$n-2\sim n$段的码值采用$n+1$位长度输出，其中第一位设置为1，后n位为该码值大小的二进制编码。

当$n<20$时，对字典的$1\sim n-4$段的码值采用$n-3$位长度输出，其中第一位设置为0，后$n-4$位为该码值大小的二进制编码；对字典的$n-3\sim n$段的码值采用$n+1$位长度输出，其中第一位设置为1，后n位为该码值大小的二进制编码。

这里称当$n\geqslant14$时采用的变长编码输出方式为双模式变长编码输出。双模式编码输出的提出是利用在压缩过程中字典各段码值的使用频率不均，而且段号小的码值占多数。

至此，本文的算法改进已介绍结束，将改进的算法命名为LZW_CH。表6对改进后的算法LZW_CH与算法LZW_CH5对部分中文文本的压缩结果进行比较。从结果可以看出，改进后的算法比LZW_CH5对中文文本有更高的压缩比，其中压缩比为压缩前的文件大小除以压缩后的文件大小。

4　结论

由表7测试结果可得，改进后的算法LZW_CH比通用的LZW19算法在对中文文本的压缩比平均大约提高了19%，但执行速度与前者相当。改进的算法保留了LZW算法无需任何预处理和执行速度快的优点，对较

表7　三种算法的平均压缩、解压时间比和平均压缩比的相对比较[1)]

算法	平均压缩相对时间比	平均解压相对时间比	平均相对压缩比
LZW19	1.00	1.00	1.00
LZW_CH5	0.92	1.06	1.16
LZW_CH	0.94	1.07	1.19

注：1)相对于LZW19的比值

表8 LZW_CH算法与Zip和WinRAR对中文文本的压缩比的比较

文件名	安徒生童话	巴黎圣母院	亮剑	平凡的世界	宋词	三国演义	史记	围城	平均
大小/KB	1 734	608	660	1 670	3 726	1 171	1 171	438	1 397
LZW_CH	2.249	1.848	1.886	2.125	1.828	1.945	1.830	1.896	1.951
Zip	1.991	1.694	1.741	1.851	1.607	1.807	1.816	1.773	1.785
WinRAR	2.217	1.842	1.880	2.103	1.779	1.968	1.932	1.840	1.945

长的中文文本其压缩比已超过了现有压缩软件Zip和WinRAR。比较结果如表8所示。

参考文献:

[1] Welch T A.A technique for high-performance data compression[J].Computer,1984,17(6):8-19.

[2] Ziv J,Lempel A.Compression of individual sequences via variable-rate coding[J].IEEE Transactions on Information Theory,1978,24(5):530-536.

[3] Fiala E R,Green D H.Data compression with finite Windows[J].Communications of the ACM,1989,32(1):490-505.

[4] Yokoo H.Improved variations relating the ziv-lempel and welch-type algorithms for sequential data compression[J].IEEE Transactions on Information Theory,1992,38(1):73-81.

[5] Miller V,Wegman M.Variations on a theme by Ziv and Lempel[J].Combinationtorial Algorithms on Words.Berlin:Springer,1985.

[6] Ziv J,Lempel A.A universal algorithm for sequential data compression[J].IEEE Transactions on Information Theory,1977,23(3):337-343.

[7] 常为领,方兴滨,云晓春,等.一种支持ANSI编码的中文文本压缩算法[J].中文信息学报,2010,24(5):96-105.

[8] 王忠效.汉语文本压缩研究及其应用[J].中文信息学报,1997,11(3):57-64.

[9] 徐秉铮,吴立忠,Wei V K.中文文本压缩的LZW算法[J].华南理工大学学报:自然科学版,1989,17(3):1-9.

[10] 华强.中西文本压缩的LZWCH算法[J].计算机工程与应用,1999,35(3):22-23.

(上接29页)

[8] Qin C,Li B,Zhu A,et al.Multiple flow direction algorithm with flow partition scheme based on downslope gradient[J].Advances in Water Science,2006,17(4):450-456.

[9] Xu Rui,Huang Xiaoxue,Luo Lin,et al.A new grid-associated algorithm in the distributed hydrological model simulations[J].Science China Technological Sciences,2010,53(1):235-241.

[10] Wallis C,Watson D,Tarboton D,et al.Parallel flow-direction and contributing area calculation for hydrology analysis in digital elevation models[C]//Proceedings of PDPTA,2009:467-472.

[11] Do H T,Limet S,Melin E.Parallel computing flow accumulation in large digital elevation models[J].Procedia Computer Science,2011,4:2277-2286.

[12] Ortega L,Rueda A.Parallel drainage network computation on CUDA[J].Computers & Geosciences,2010,36(2):171-178.

[13] Qin C Z,Zhan L.Parallelizing flow-accumulation calculations on graphics processing units—from iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm[J].Computers & Geosciences,2012,43:7-16.

[14] Planchon O,Darboux F.A fast,simple and versatile algorithm to fill the depressions of digital elevation models[J].Catena,2002,46(2):159-176.

[15] Gaster B R,Howes L,Kaeli D R,et al.Heterogeneous computing with OpenCL[M].张云泉,张先轶,龙国平,等译.北京:清华大学出版社,2012.

[16] 刘学军,卢华兴,卞璐,等.基于DEM的河网提取算法的比较[J].水利学报,2006,37(9):1134-1141.

[17] Holmgren P.Multiple flow direction algorithms for runoff modelling in grid based elevation models:an empirical evaluation[J].Hydrological Processes,1994,8(4):327-334.