文章编号: 1009-3443(2006)02-0137-04

PDF417纠错码原理及实现

戴水贵[1], 吴晓联[2]

(1. 解放军理工大学 工程兵工程学院, 江苏 南京 210007; 2. 解放军海军司令部, 北京 100841)

摘 要: 为了促进PDF417这种高效率低成本的条形码在中国的深入研究和推广, 介绍了Reed-Solomon纠错码原理和它在PDF417二维条形码中的应用方法。推导了在迦罗华域GF(929)内编译码方法, 并通过实例具体说明。提出了一种适用于PDF417纠错编码和译码的查表方法, 可以简单确定任意码字在GF(929)内所对应的本原元的幂次。运用该方法可以快速而准确地计算出纠错码生成多项式的系数列以及差错伴随式。

关键词: RS错误控制码; PDF417纠错码; 迦罗华域; 编码; 译码

中图分类号: TP391 **文献标识码**: A

PDF417 error correcting code and its implementation

DAI Shui-gui[1], *WU Xiao-lian*[2]

(1. Engineering Institute of Corps of Engineers, PLA Univ. of Sci & Tech., Nanjing 210007, China;
2. The Headquarters of the Navy, Beijing 100841, China)

Abstract: To promote the research and application of PDF417, a two-dimensional bar code with high efficiency and low cost, the principle of reed-solomon (RS) error correcting code and its application in two-dimensional bar code PDF417 were introduced, and details on encoding in GF(929) were presented. One example was used to make the application clearer. Meanwhile, a looking-through database method used for yielding PDF47 error correcting codes was also developed, making it easier and more accurate to compute the coefficients of the generating polynomial and results of syndrome polynomials.

Key words: RS(Reed-Solomon) error control code; PDF417 error correcting code; Galois region; encode; decode

PDF417二维条形码是一种可以不基于网络连接独立使用的数据传输方式, 具有存储信息密度高、容量大、纠错能力强和译码可靠性高等特点, 在数据传输和数据存储领域有很好的应用前景。它采用目前世界上最先进的纠错码技术之一——RS(reed-solomon)错误控制码。RS码的纠错能力非常强大, 尤其是对那些突发性的成片干扰特别有效[1]。可有效地提高PDF417码的抗干扰能力和可靠性。但是, RS码的译码原理和编译码过程都比较复杂, 加上PDF417使用的RS码以迦罗华域GF(929)为其循环域, 更增加了它在理解和使用上的难度。关于RS码在PDF417中的使用原理及方法, 国内外的文献中都鲜有提及, 在某种程度上限制了我国自主研发PDF417产品的发展。

收稿日期: 2005-09-09.

作者简介: 戴水贵(1952-), 男, 副教授; 研究方向: 条形码应用; E-mail: daishuigui@163.com.

1 RS码原理

1.1 预备知识

定义1 以素数q为模的整数剩余类构成q阶有限域GF(q)。在GF(q)中, 某一元素a满足$a^{q-1}=e$, 则称a的GF(q)的本原域元素, 简称本原元。在任何GF(q)中都能找到一个生本原元a, 能用它的幂次表

示所有 $q-1$ 个非零元素,从而组成一个循环群 $G(a)$: $1, a, a^2, \cdots, a^{q-1}$,其中: $a^{q-1}=1$[2]。

1.2 编 码

a 为GF(q)中的一个本原元,则能纠正 t 个错误的本原RS码的生成多项式为

$$g(x)=(x-a)(x-a^2)(x-a^3)\cdots(x-a^{2t})=g_0+g_1x^1+g_2x^2+\cdots+g_{2t-1}x^{2t-1}+x^{2t}, \quad (1)$$

其中: $g_i(i=0,1,2,\cdots,2t-1)$ 是GF(q)域中的元素; a 为GF(q)域的本原元。

若原始信息多项式为 $d(x)=d_0+d_1x+d_2x^2+\cdots+d_{n-1}x^{n-1}$,进行RS编码后的表达式为

$$v(x)=x^{2t}d(x)+x^{2t}d(x) \bmod g(x), \quad (2)$$

其中: $x^{2t}d(x)$ 是原始数据码; $x^{2t}d(x) \bmod g(x)$ 是纠错码部分,它跟在原始数据码后面。

1.3 译 码

1.3.1 伴随式 S_k 的定义及计算

式(2)是不含错误的信息表达式,若传输过程中发生错误,接收到的码字表达式为

$$r(x)=\sum_{i=0}^{n-1}r_ix^i, \quad (3)$$

其中: r_i 为接收码字,其中包含错误。由于 $v(x)$ 是不包含错误的原始表达式,所以令 $e(x)$ 为错误图样,于是 $r(x)=v(x)+e(x)$。用纠错码生成多项式 $g(x)$ 的各个零点 $a^j(j=1,2,\cdots,2t)$ 对 $r(x)$ 求值,其结果定义为伴随式 S_j。由于 $v(a^j)=0$,可得

$$S_j=r(a^j)=v(a^j)+e(a^j)=e(a^j)。 \quad (4)$$

假设 $r(x)$ 包含 r 个错误($0\leq r\leq t$),它们发生在未知位置 $i_1,\cdots,i_r$ 上。于是差错多项式 $e(x)$ 又可写成

$$e(x)=e_{i_1}x^{i_1}+e_{i_2}x^{i_2}+\cdots+e_{i_r}x^{i_r}, \quad (5)$$

其中: $e_{i_1},\cdots,e_{i_r}$ 为各个错误位置上错误值与正确值之差,也就是错误幅值。用 a 对接收多项式求值可以得到伴随式: $S_1=e(a)=e_{i_1}a^{i_1}+e_{i_2}a^{i_2}+\cdots+e_{i_r}a^{i_r}$。为简化记号,令错误幅值 $Y_l=e_{i_l}$,错误位置 $X_l=a^{i_l}$, i_l 是第 l 个差错的实际位置,而 X_l 是对应这些位置的域元素($l=1,2,\cdots,r$)。于是有 $S_1=Y_1X_1+Y_2X_2+\cdots+Y_rX_r$。同理,用 a 的各次幂,即式(1)的各个零点,对差错多项式(4)求值,可得到 $2t$ 个方程的方程组。

$$\left.\begin{aligned}S_1&=Y_1X_1+Y_2X_2+\cdots+Y_rX_r\\S_2&=Y_1(X_1)^2+Y_2(X_2)^2+\cdots+Y_r(X_r)^2\\&\vdots\\S_{2t}&=Y_1(X_1)^{2t}+Y_2(X_2)^{2t}+\cdots+Y_r(X_r)^{2t}\end{aligned}\right\} \quad (6)$$

事实上, $X_1,X_2,\cdots,X_r$ 未知, $Y_1,\cdots,Y_r$ 也未知,甚至 r 也未知,这些都必须通过计算来获得,从而纠正差错。

1.3.2 求错误位置 X_i 和错误幅值 Y_i 的算法

译码就是求解有限域GF(q)上的非线性方程组式(6)。当接受码字 $r(x)$ 包含 r 个错误时,可由伴随式 S 确定错误位置 X_i 和错误幅度 $Y_i(i=1,2,\cdots,r)$。为简化求解,引入错误位置多项式[2,3]

$$\sigma(x)=\prod_{i=1}^{r}(1-xX_i), \quad 0\leq r\leq t。$$

若第 i 个错误位置 $x=X_i^{-1}$,则 $\sigma(X_i^{-1})=0$,因此求错误位置就是求解错误位置多项式的根。如果知道 $\sigma(x)$ 的各项系数,就可以求出它的零点从而得到差错位置。试图从伴随式 S 计算 $\sigma(x)$ 各项系数。将 $\sigma(x)$ 展开,令系数分别为 $\sigma_1,\sigma_2,\cdots,\sigma_r$,得

$$\sigma(x)=1+\sigma_1x+\sigma_2x^2+\cdots+\sigma_rx^r, \quad (7)$$

将式(1)中 $g(x)$ 的各个零点代入式(7)。在错误位置 $i_1,\cdots,i_r$ 处 $\sigma(x)=0$ 都成立。将各式迭加,经过整理变形后可得到以下关于 S 和 $\sigma(x)$ 系数的联立方程[3]。

$$\begin{pmatrix}S_1&S_2&\cdots&S_r\\S_2&S_3&\cdots&S_{r+1}\\\vdots&\vdots&&\vdots\\S_r&S_{r+1}&\cdots&S_{2r-1}\end{pmatrix}\begin{pmatrix}\sigma_r\\\sigma_{r-1}\\\vdots\\\sigma_1\end{pmatrix}=-\begin{pmatrix}S_{r+1}\\S_{r+2}\\\vdots\\S_{2r}\end{pmatrix}。 \quad (8)$$

可以看出,译码的关键是求解 $\sigma(x)$,求出错误位置值。当系数矩阵的行列式不为零时,方程有解,所对应的值就是信息中包含的错误数目。译码流程如图1所示。

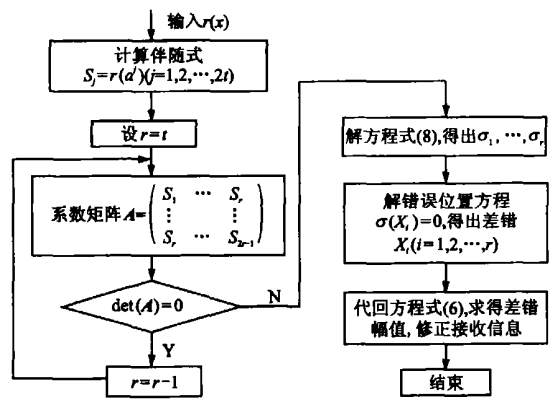

图1 RS码译码流程

Fig. 1 Flow chart of RS code decoding

2 RS码在PDF417条码中的应用

2.1 用本原元幂次形式表示PDF417码字的算法

PDF417的纠错码是以GF(929)为域的RS码。

根据式(1), 要得到纠错码字, 需要确定GF(929)的一个本原元a, 以得到纠错码生成多项式。文献[4]规定, PDF417 以3 为本原元, 也就是说GF(929)中任何一个元素都可以表示为3^n ($0 \le n < 929$)。所有的伴随式S_j 也都可以以该形式表示。下面证明3 为GF(929)的一个本原元, 并据此寻找一种可将GF(929)中任何元素表示为3 的某次幂形式的算法。

Euler 定理[5]　若$(k, m) = 1$, 则$k^{\varphi(x)} \equiv 1 \pmod m$。其中: $\varphi(m)$为1, 2, …, $m-1$ 中与m 互素的数的个数。

根据Euler 定理可得, 当p 为素数时, 同余方程$x^{p-1} \equiv 1 \pmod p$以1, 2, …, $p-1$ 为解[5]。因此, 根据本原元定义(定义1), 1, 2, 3, …, 928 都是GF(929)的本原元。

进一步的问题是, 对于GF(929)中的元素k, 如何确定$3^n \equiv k \pmod{929}$中的n 值, 一种可行的思路是, 计算出0～929 内所有的n 值所对应的k, 并将结果制表, 通过查表完成已知k 求取n 的工作。然而, 如果直接计算3^n, 当n 较大时, 会出现数据过大溢出的情况, 导致错误。针对这个问题, 本文设计了一种简化的算法。

根据欧几里德除法定理[3], 任何3^n 可表示为$3^n = q \times 929 + r$ ($0 \le r < 929$), 也就是$3^n \equiv r \pmod{929}$。进而可知, $3^{n+1} = 3q \times 929 + 3r$。若$3r \le 929$, $3^{n+1} \equiv 3r \pmod{929}$; 若$3r > 929$, 则$3^{n+1} = (3q+1) \times 929 + (3r - 929)$, 那么$3^{n+1} \equiv (3r - 929) \pmod{929}$。运用以上规则, 可以利用$3^{n+1}$和$3^n$ 之间的关系, 使用递推的方法, 根据已知的3^n 确定3^{n+1}。每次循环就是一个将前一次的余数乘以3 再对929 求余的过程, 大大减少了编程时的计算量。

将使用以上算法生成的结果导入数据库, 建立幂次—数据表。在编码及译码过程中, 由于所有元素都以3 的某次幂表示, 需要大量求取GF(929)的域元素所对应的幂次(纠错级别越高, 所需计算量越大)。采用查表法后不仅可以准确确定幂次和数据的对应关系, 而且显著地减少编译码的计算量, 提高编译码效率。

2.2　运用幂次—数据表生成纠错码

PDF417 有8 个纠错等级, 所对应错误纠正码字数目为

$$k = 2^{s+1}, \tag{9}$$

其中: s 为纠错等级。对于一个给定的错误纠正等级, 其错误纠正容量由$e + 2t \le 2^{s+1} - 2$ 确定。其中: e 为拒读错误数目; t 为替代错误数目。其符号数据多项式为

$$d(x) = d_{n-1}x^{n-1} + d_{n-2}x^{n-2} + \cdots + d_1 x + d_0。 \tag{10}$$

含k 错误纠正码字的生成多项式为

$$g(x) = (x - 3)(x - 3^2)\cdots(x - 3^k) = x^k + g_{k-1}x^{k-1} + \cdots + g_1 x + g_0。 \tag{11}$$

对于一组给定的数据码字和一选定的错误纠正等级, 错误纠正码字为符号数据多项式$d(x)$乘以x^k, 然后除以生成多项式$g(x)$, 所得余式的各项系数的补数。

文献[6]给出一种确定纠错码生成多项式(11)的各项系数g_i ($i = 0, 1, 2, \cdots, k-1$)的计算机实现方法, 在它的方法中多次直接计算3 的各次幂值。事实上, 由式(9)可以看出, 在3 级纠错的情况下, k 就已经达到16。在幂次更高的情况下, 由于数据过大溢出而产生错误。g_i 与g_{i-1}之间的递推关系满足

$$\left.\begin{aligned} &g_{0,1} = -3, g_{1,1} = 1, j = 1;\\ &g_{0,j} = (-g_{0,j-1} \times 3^j) \bmod 929;\\ &g_{i,j} = (g_{i-1,j-1} - g_{i,j-1} \times 3^j) \bmod 929;\\ &g_{j,j} = 1;\\ &j = 2, 3, \cdots, k, i = j-1, j-2, \cdots, 1。 \end{aligned}\right\} \tag{12}$$

其中: j 表示第j 次迭代; i 表示每次迭代中x^i 项的系数。j 逐一递增到k 为止, j 每递增1, i 都从$j-1$ 递减至1 循环一次。所以当递推完成后, 所得结果即为各项系数。

采用与§1 中相同的原理, 可以证明, 对于式(11), 在GF(929)内, 以$3^j \pmod{929}$代替3^j 将得到相同的结果。使用数据库检索子程序检索幂次—数据表, 确定3^j 在GF(929)内对应的域元素并将其代入式(12)。该方法将上百次的乘方运算转化为查表, 从而大大减少了计算量, 实验证明可以准确完成所有8 级纠错码的生成多项式的系数确定工作。

知道数据码字和纠错码生成多项式系数序列后就可以应用文献[3]中的伪程序生成纠错码字。

3　PDF417 纠错编译码的实现

3.1　编码过程

将编制原始信息为"PDF417"的条码, 采用的纠错等级为2。通过对原始信息的分析, 可以知道, 对于字符串"PDF417", 采用文本压缩模式[1]其表示码字应为453, 178, 121, 239, 符号的长度码字为5(包括符号长度码字), 那么所有的数据码字为$d_4 = 5$, $d_3 = 453$, $d_2 = 178$, $d_1 = 121$, $d_0 = 239$。根据式(11)纠错码字生成多项式为$g(x) = (x - 3)(x - 3^2)\cdots(x - 3^8)$。运用计算生成多项式系数的程序, 得出系数序列值

a_6～a_7 分别为237、308、436、284、646、653、428、379。

根据式(10)(11)可知,纠错码表达式为式 $x^k\,\mathrm{d}(x)/g(x)$ 所得余式的各系数的补数。所得纠错码 c_0 至 c_7 依次为674、896、798、445、841、604、896、807,完成编码任务。

3.2 译码过程

被污染后的条码图像,如图2所示。

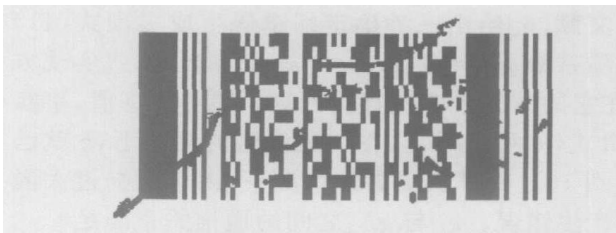

图2 原始信息为"PDF417",含2级纠错的被污染条码图像

Fig. 2 One contaminated barcode with error correcting level two, its original data is "PDF417"

由于污染,接收到的码字组发生了2个错误,如表1所示。

表1 原始码字与接收码字对照表

Tab. 1 Original barcode characters and the received barcode characters

码字位置	代码表示	原始码字值	接收码字值
12	d_4	5	5
11	d_3	453	453
10	d_2	178	20
9	d_1	121	121
8	d_0	239	15
7	c_7	674	674
6	c_6	896	896
5	c_5	798	798
4	c_4	445	445
3	c_3	841	841
2	c_2	604	604
1	c_1	896	896
0	c_0	807	807

在没有使用纠错码的情况下,译码结果为"PDAU EBAP"。开始纠错运算,根据式(3)计算接收码字多项式:

$$r(x)=\sum_{i=0}^{n-1} r_i x^i = 5x^{12}+453x^{11}+20x^{10}+121x^9+15x^8+807x^7+896x^6+604x^5+841x^4+445x^3+798x^2+896x+674。$$

根据式(4)计算多个伴随式的值。在计算过程中通过查找编制的幂次—数据表得到 x 的幂表达式结果,同时用3的幂次形式表达各项系数,将乘法转化为幂次加法,避免数据过大溢出。根据图1的译码流程计算结果为

$$S_1=219, S_2=58, S_3=25, S_4=367,$$
$$S_5=354, S_6=488, S_7=731。$$

计算系数矩阵行列式 $A_4\equiv 0 \pmod{929}$, $A_3\equiv 0 \pmod{929}$, $A_2\equiv 253 \pmod{929}\neq 0$。

由此可知,差错位置数 $r=2$。将差错位置数代回式(8)可知

$$\begin{pmatrix} 219 & 58 \\ 58 & 25 \end{pmatrix}\begin{pmatrix} \sigma_2 \\ \sigma_1 \end{pmatrix}=\begin{pmatrix} -25 \\ -367 \end{pmatrix}。$$

可得 $\sigma_2=3^{18}$, $\sigma_1=3^{302}$,因此错误位置方程为 $3^{18}x^2+3^{302}x+1=0$。解得 $x=3^{-10}$ 和 3^{-8} 为方程的解。根据错误位置多项式的定义式(7)可知,错误位置即第10位和第8位。得到差错位置后,将错误位置值代入式(6),可以解出错误幅值 $Y_1=771$, $Y_2=705$。

以错误幅值修正接收数据得到原始数据。第10位上的原始码字为 $(20-771)\equiv 178 \pmod{929}$,8位上的原始码字为 $(15-705)\equiv 239 \pmod{929}$。纠错后的译码结果为"PDF417"。完成了译码纠错的任务。

4 结 语

对于更高纠错等级的情况,可以通过BM(berlekamp-massey)迭代算法完成求解错误位置多项式(6)的任务,译码的原理和思路与上例大同小异。

可以看出,PDF417的纠错码是非常有效的。增强对它的应用方法的理解,对于推广这种高效低廉的信息存储与传输方式,具有深远的意义。

参考文献:

[1] 陶德元,何小海,吴志华. RS码编译码算法的实现[J]. 四川大学学报:自然科学版,2000,12,7(6):868-872.

[2] 王新梅,肖国镇. 纠错码原理与方法[M]. 西安:西安电子科技大学出版社,2001.

[3] BLAHUT R E. 差错控制码的理论与实践[M]. 广州:华南理工大学出版社,1998.

[4] 国家技术监督局. GB/T 17172—1997[S]. 北京:中国标准出版社,1997.

[5] 裴定一,祝跃飞. 算法数论[M]. 北京:科学出版社.

[6] 何 军,康景利. 条形码的计算机编码与识别[J]. 计算机测量与控制,2002,10(4):263-266.

(责任编辑:程 群)