一种自适应数据逐层分解的 Reed-Solomon 码迭代纠错方法及应用

王 娟 王 萍*
(天津大学电气与自动化工程学院 天津 300072)

摘 要：该文针对 Reed-Solomon 码纠错算法计算复杂度较高、运算时间较长等问题，提出一种自适应数据逐层分解的 Reed-Solomon 码的迭代译码纠错方法。首先，接收码通过逐层分解将随机错误或突发错误分散于不同的子序列中，缩小突发或随机错误的查找范围；其次，制定约束规则确定错误数目，同时根据不同的伴随矩阵维数自适应选择迭代求解关键方程的方法，定位子序列中误码的位置；最后，计算正确码字，结束纠错。实验测试表明，该算法在保证不漏检误码的前提下，能够有效简化计算多项式的维数，减少计算量和复杂度，纠错时效优于 DFT(Discrete Fourier Transform)算法和 BM(Berlekamp-Massey)算法。特别是对 2 维码数据的纠错测试中，与传统算法相比，该算法纠错时效可提升一个数量级。

关键词：Reed-Solomon(RS)码；逐层分解；降维；迭代求解
中图分类号： TN911.22 **文献标识码**： A **文章编号**：1009-5896(2015)05-1173-07
DOI: 10.11999/JEIT140907

An Adaptive Reed-Solomon Iterative Correction Method Based on Data Layer-wise Decomposition and Its Application

Wang Juan Wang Ping
(*College of Electrical and Automation Engineering*, *Tianjin University*, *Tianjin* 300072, *China*)

Abstract: In order to reduce the computational complexity, an improved decoding algorithm based on a layer-wise decomposition transform is proposed for Reed-Solomon (RS) codes in this paper. Firstly, the received codewords are split into a number of sub-sequence codewords by layer-wise decomposition. The random or burst error are dispersed in different sub-sequences, narrowing search areas of the burst or random errors. Secondly, the appropriate rules are developed to determine the number of errors. To help locate the error pattern of the sub-sequence, an adaptive iterative method to solve the key equation is used according to the adjoin matrix dimension. Finally, the correct codewords are obtained by subtracting error estimation from the received sequence. The tests show that in premise of detecting all errors the order of the polynomial is reduced and the computational complexity is lowered. The rate of error correction of the proposed algorithm is higher than DFT (Discrete Fourier Transform) algorithm and BM (Berlekamp-Massey) algorithm. Especially in the tests of the two-dimensional code, error correction efficiency is improved one order of magnitude.

Key words: Reed-Solomon (RS) code; Layer-wise decomposition; Dimensionality reduction; Iterative solution

1 引言

信道码字在快速传输过程中难免会受到噪声侵袭，使其携带的信息出现偏差。为此，人们在输入码中加入冗余校验码字辅以纠错，以提高传输的正确率。BCH(Bose ray-Chaudhuri Hocquenghem)码[1]是带有此类纠错码的典型代表，而 Reed-Solomon(RS)码[2]则是 BCH 码的一个重要子类，RS 码首先由 Reed 和 Solomon 应用 Mattson-Solomon 多项式于 1960 年构造出来，是一类具有很强纠错能力的线性分组码，具有纠正随机错误及突发错误的能力，被广泛应用于数字信号传输、深空通信、高密度磁盘存储、量子计算等方面[3]。

自 RS 码被创建以来，众多学者对其纠错算法中关键步骤进行深入研究。目前，纠错算法分为硬判决和软判决两种，本文重点讨论硬判决译码纠错算法。关于硬判决纠错算法，文献[4]使用 PGZ (Peterson Gorenstein Zierler)算法纠错，该算法实现简单，易于理解，对于较短码的纠错非常有效。但它使用穷举法求解误码位置多项式，耗时较长，不适合维数较高的误码位置多项式的求解。文献[5,6]将典型的欧几里德算法加入到求解过程中，该方法采用多项式分解的原理求解多项式的最大公因式，因此需要多次进行多项式长除运算，且在迭代过程

2014-07-11 收到，2014-11-18 改回
河北省科技支撑项目资助课题
*通信作者：王萍 wangps@tju.edu.cn

中需要计算多项式的次数，计算方法易陷入不收敛；文献[7–10]将具有自回归滤波性质的BM(Berlekamp-Massey)迭代算法用于纠错，目前此方法应用较广泛。但上述方法所需时间和纠错周期依然较长，计算步骤仍较复杂。为提高纠错效率，文献[11,12]使用离散傅里叶变换(Discrete Fourier Transform, DFT)算法进行纠错，即将输入码变换至频域，使用DFT算法的特性求解复杂的方程式，在GF(2^m)域上通过高效的DFT算法来计算错误信息多项式，提升纠错效率，但效果有限。

针对上述文献算法中的不足，本文提出了一种基于数据逐层分解的RS码的自适应迭代纠错算法，重新定义伴随式和误码数的判断，缩小误码的查找范围，快速判读误码数目，降低关键方程的求解维数，自适应选择穷举法或无求逆过程的iBM(inversionless Berlekamp-Massey)算法[9]求解关键方程。在不降低最大纠错能力的前提下，本文算法计算复杂度显著降低，纠错效率明显提升。

2 RS码及其纠错过程

RS码是被定义于伽罗华域GF(2^m)(m为大于1的正整数)的线性分组码，是BCH码的重要分支。由于RS码的码字和生成多项式的根均取自GF(2^m)，故能纠正t个错误的RS码生成多项式[13]可表示为：$g(x)=\prod_{i=0}^{2t-1}(x-\alpha^i)$。其中，$\alpha$称为本原元。RS码还可使用RS$(n,k,t)$标记。其中，$n$为码长，$n=2^m-1$, $m(m=3\sim 8)$ bit为单位组成一个码字，k为信息位长度，t为最大纠错数。t,k,n之间关系满足$n=k+2t$。由于RS码的编译码由一组非单独0或1的码字构成，因此RS码特别适合处理突发或随机出现的成片错误。

RS码纠错算法大致分为4个步骤[14]：

(1)计算接收码的伴随多项式：将长度为N的接收码定义为$r=\{r_0,r_1,\cdots,r_{N-1}\}$，表示为$R(x)$，它包括原始码$C(x)$和错误图样$E(x)$，其关系可表示为$R(x)=\sum_{i=0}^{N-1}r_ix^i=C(x)+E(x)$。定义$R(\alpha^j)$为$r$的伴随式，记

$$S_j=R(\alpha^j)=\sum_{i=0}^{t-1}r_i\cdot(\alpha^j)^i,\quad j=1,2,\cdots$$

(2)由伴随多项式得到伴随矩阵：用上述$2t-1$个伴随式计算结果组成伴随矩阵$\boldsymbol{M}$。可以证明[14]，接收码r中所含误码个数v等于矩阵$\boldsymbol{M}$的秩，即$v=\mathrm{rank}(\boldsymbol{M})$，当且仅当接收码中实际误码个数大于等于$t$时，矩阵$\boldsymbol{M}$满秩。

(3)基于伴随矩阵的误码定位：为求得v个误码位置，通常假设存在一个以所有的误码位置为根的误码位置多项式$\sigma(p)=\sigma_0+\sigma_1p+\sigma_2p^2+\cdots+\sigma_tp^t$，令$\sigma(p)=0$所得解为误码位置[6]。文献[6]进一步证明可通过求解方程组式(1)得到方程$\sigma(p)=0$的系数。方程组式(1)也被称为关键方程。

$$\boldsymbol{M}\begin{bmatrix}\sigma_1\\ \sigma_2\\ \vdots\\ \sigma_t\end{bmatrix}=\begin{bmatrix}S_{t+1}\\ S_{t+2}\\ \vdots\\ S_{2t}\end{bmatrix}\tag{1}$$

将本原元的各次幂的倒数$\alpha^{-1},\alpha^{-2},\cdots,\alpha^{-t}$依次代入$\sigma(p)$，若$\sigma(a^{-j})=-1$ ($j=1,2,\cdots,t$)，则表明接收码的第j个码字出现错误。

(4)修改误码位置的码字，完成纠错：根据Forney算法[14]计算出错误码字的差错值。将错误码字减去相应的差错值即可得到正确的码字。

3 基于逐层分解的纠错算法及分析

第2节中RS码的纠错过程表明，为确定接收码r中是否存在误码，需要在计算$2t-1$个伴随元素的基础上，对$t\times t$阶矩阵及其降阶矩阵进行求秩运算，以便于找出错误码字并进行校正。求秩运算的复杂度为$O(t!)$，运算耗时较长。因此，本文通过对接收码的逐层分解，减少伴随矩阵的计算量，进而降低求秩运算的复杂度。

3.1 纠错算法实现过程

3.1.1 逐层分解算法的实现 在不降低最大纠错能力且提高效率的前提下，本文遵循将接收码中的错码尽量分散至子序列中和尽量降低定位错码计算复杂度的原则逐层分解接收码。首先，依照图1，对长度为N的接收码字序列r进行逐层分解(分解过程见表1)。

接收码经过L ($L<m$且$L\in N$)层分解后，得到$2L$个子序列。由于分解后的子序列元素仍属于伽罗华域，根据其元素特点[11]，可以将分解所得的序列$\delta_i^{(j)}$和$\lambda_i^{(j)}$表示为以码字为权值的多项式形式：

$$\delta_i^{(j)}(x)=\sum_{f=0}^{N_i-1}\delta_{i_f}^{(j)}x^f,\quad i=1,2,\cdots,L;j=1,2,\cdots,2^{L-1}\tag{2}$$

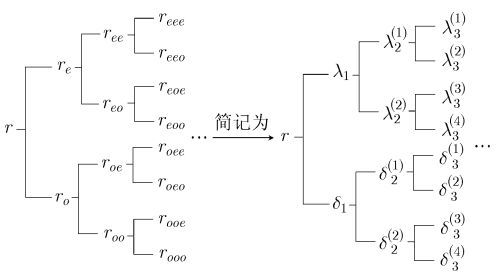


图1 逐层分解示意图

表 1 逐层分解算法步骤

算法 1 逐层分解算法
输入：接收码 $r=\{r_0,r_1,\cdots,r_{N-1}\}$
输出：$2L$ 个子序列 $\delta_i^{(j)}$ 和 $\lambda_i^{(j)}, i=1,2,\cdots,L; j=1,2,\cdots,2^{L-1}$
步骤 1　接收码进行逐层分解，得到子序列 $\{\lambda_i^{(j)}\},\{\delta_i^{(j)}\}$，$L$ 为分解层数，各子序列长度为 $N_i \le N/2^i$；
步骤 2　判断是否满足分解终止条件 $L \le \log_2(N/m)$，若满足则转向步骤 3；若不满足则转向步骤 1；
步骤 3　计算子序列伴随式。

$$\lambda_i^{(j)}(x)=\sum_{f=0}^{N_i-1}\lambda_{i_f}^{(j)}x^f,\quad i=1,2,\cdots,L; j=1,2,\cdots,2^{L-1} \tag{3}$$

3.1.2 伴随式计算与错误数判断　接收码经 L 层分解后，包含误码的所有码字被分配到第 L 级的 2^L 个子序列中，这些子序列的最大长度 $N_{\max}=N/2^L$。为判断出误码个数，需要分别计算这些子序列的伴随式，生成伴随矩阵并求出它们的秩。

下面以子序列 $\delta_L^{(j)}$ 为例说明其伴随式 $S_d^{\delta_L^{(j)}}(d=1,2,\cdots,\tau_{\delta_j})$ 的计算过程。其中，τ_{δ_j} 为序列 $\delta_L^{(j)}$ 对应的最大纠错数，根据 RS 码的编码规则可知 $\sum\tau_{\delta_j}+\sum\tau_{\lambda_j}\ge t$。可按照式(4)依次计算 $S_d^{\delta_L^{(j)}}$：

$$S_d^{\delta_L^{(j)}}=R_L^{(j)}(\alpha^d)=\sum_{k=0}^{\tau_{\delta_j}-1}\delta_{L_k}^{(j)}\cdot(\alpha^d)^k \tag{4}$$

其中，$j=1,2,\cdots,2^{L-1}; d=1,2,\cdots,\tau_{\delta_j}$。则其对应的伴随矩阵为

$$\boldsymbol{M}^{\delta_L^{(j)}}=\begin{bmatrix} S_1^{\delta_L^{(j)}} & S_2^{\delta_L^{(j)}} & \cdots & S_{\tau_{\delta_j}}^{\delta_L^{(j)}} \\ S_2^{\delta_L^{(j)}} & S_3^{\delta_L^{(j)}} & \cdots & S_{\tau_{\delta_j}+1}^{\delta_L^{(j)}} \\ \vdots & \vdots & \ddots & \vdots \\ S_{\tau_{\delta_j}}^{\delta_L^{(j)}} & S_{\tau_{\delta_j}+1}^{\delta_L^{(j)}} & \cdots & S_{2\tau_{\delta_j}-1}^{\delta_L^{(j)}} \end{bmatrix} \tag{5}$$

通过求解矩阵 $\boldsymbol{M}^{\delta_L^{(j)}}$ 的秩确定子序列 $\delta_L^{(j)}$ 中误码的数目。若满秩则存在 τ_{δ_j} 个错误；若不满秩，则记录其产生的实际误码数 v_{δ_j}。同理，依次计算子序列 $\lambda_L^{(j)}$ 的伴随式 $S_d^{\lambda_L^{(j)}}(d=1,2,\cdots,\tau_{\lambda_j})$ 和对应的伴随矩阵 $\boldsymbol{M}^{\lambda_L^{(j)}}$，算出实际发生的误码数 v_{λ_j}。具体算法步骤如表 2 所示。

3.1.3 误码定位及修正　此处仍以子序列 $\delta_L^{(j)}$ 为例说明误码位置多项式的求解过程，令 $\sigma(p)$ 改写为 $\sigma(p)=\sigma_0+\sigma_1 p+\sigma_2 p^2+\cdots+\sigma_{\tau_{\delta_j}}p^{\tau_{\delta_j}}$。则相应的关键方程可表示为

表 2 误码数目判断步骤

算法 2 误码数的判断
输入：子序列伴随矩阵
输出：每个子序列的误码数目
步骤 1　初始化：令 $j=1$，累积误码数 $v_{\delta_L}=0, v_{\lambda_L}=0$；
步骤 2　依次计算 $\delta_L^{(j)}$ 和 $\lambda_L^{(j)}$ 的伴随式 $S_{d_1}^{\delta_L^{(j)}}$ 和 $S_{d_2}^{\lambda_L^{(j)}}$，$(d_1=1,2,\cdots,\tau_{\delta_j}; d_2=1,2,\cdots,\tau_{\lambda_j})$，并构建伴随矩阵 $\boldsymbol{M}^{\delta_L^{(j)}}$ 和 $\boldsymbol{M}^{\lambda_L^{(j)}}$；
步骤 3　计算两个伴随矩阵的秩 v_{δ_j} 和 v_{λ_j}，得到可纠正的累积错误数 $v_{\delta_L}=v_{\delta_L}+v_{\delta_j}$ 和 $v_{\lambda_L}=v_{\lambda_L}+v_{\lambda_j}$；
步骤 4　判断若 $v_{\lambda_L}+v_{\delta_L}\ge t$，则结束。否则令 $j=j+1$，返回步骤 2。

$$\boldsymbol{M}^{\tau_{\delta_j}}\begin{bmatrix}\sigma_1\\ \sigma_2\\ \vdots\\ \sigma_{\tau_{\delta_j}}\end{bmatrix}=\begin{bmatrix} S_{\tau_{\delta_j}+1}^{\delta_L^{(j)}} \\ S_{\tau_{\delta_j}+2}^{\delta_L^{(j)}} \\ \vdots \\ S_{2\tau_{\delta_j}}^{\delta_L^{(j)}}\end{bmatrix} \tag{6}$$

此处，需首先对 $\boldsymbol{M}^{\delta_L^{(j)}}$ 的维数进行评估：若维数小于 3，则采用穷举法来求解，以求计算过程简单高效；而维数较高时则使用迭代算法求解，规避穷举法求解计算繁琐的缺点。

迭代算法采用无求逆的 iBM 迭代算法，该算法利用自回归滤波原理迭代求解最小反馈误码位置多项式 $\sigma(p)$。设 $B_i(i=1,2,\cdots,\tau_{\delta_j})$ 为临时误码位置多项式系数。无求逆的 iBM 迭代算法的具体步骤[15]如表 3 所示。

通过算法 3 求出 $\sigma(p)$ 的系数后，将 $\alpha^{-1},\alpha^{-2},\cdots,\alpha^{-\tau_i}$ 依次代入 $\sigma(p)$，若 $\sigma(a^{-k})=-1$ $(k=1,2,\cdots,\tau_{\delta_j})$，则表明子序列 $\delta_L^{(j)}$ 的第 k 个码字不正确。同理，

表 3 无求逆 iBM 迭代算法步骤

算法 3 无求逆 iBM 迭代算法
输入：子序列关键方程
输出：每个子序列的关键方程的根
步骤 1　设 $\sigma_0(0)=1,\sigma_i(0)=0, B_i(0)=1, \eta=0, d_0=1, d_1=S_1^{\delta_L^{(j)}}$；
步骤 2　$\eta=\eta+1, d_i\sigma^{(\eta+1)}(p)=d_i\sigma^{(\eta)}(p)-d_\eta p^{\eta-i}\sigma^{(i)}(p)$，判断 d_η 是否为 0，是则转到步骤 3，否则转到步骤 4；
步骤 3　计算 $B^{(\eta)}(p)=pB^{(\eta-1)}(p), \sigma^{(\eta+1)}(p)=\sigma^{(\eta)}(p)$；
步骤 4　$B^{(\eta)}(p)=\sigma^{(\eta)}(p), \sigma^{(\eta+1)}(p)=d_{\eta-1}\sigma^{(\eta)}(p)+d_\eta[pB^{(\eta)}(p)]$；
步骤 5　计算 $d_{\eta+1}=\sum_{i=0}^{L(\eta+1)}S_{\eta+1-i}\sigma_i^{(\eta+1)}$，$(L(\eta+1)$是$\sigma^{(\eta+1)}(p)$最高项次数)；
步骤 6　判断 η 是否已达到 $2\tau_{\delta_j}$ 次，若 $\eta=2\tau_{\delta_j}$，迭代终止，否则跳转到步骤 2。

可将其他子序列对应的误码位置求出。

最后使用 $Y_k = -k\omega(k^{-1})/\sigma(k^{-1})$ (其中， $\omega(k^{-1}) \bmod p^t = S(k^{-1})\sigma(k^{-1})$)求解每个子序列中的误码的差错值，用误码码字减去该差错值得到正确码字。

综上所述，基于数据逐层分解的 RS 码纠错算法流程图如图 2 所示。

3.2 算法复杂度分析

通过数据逐层分解后的 RS 码的纠错算法计算量明显下降，主要体现在：

(1)伴随式矩阵的降维。式(5)表明分解后的伴随矩阵维数由 $2t-1$ 降低至 $t/2^{L-1}-1$ 。矩阵维数的降低能够有效提高矩阵运算效率，为后续误码数目的快速判断和关键方程的简化求解提供强有力的保障。

(2)误码数目的快速判断。假设分解后的子序列中均含有误码，当 r 中的 N 个符号同时等分到 2^L 个子序列中，用于求取误码数的矩阵秩的计算复杂度由 $O(t!)$ 减少为 $O(2\times(t/2^{L-1})!)$ 。而由于

$$\frac{t!}{2\times(t/2^L)!}=\frac{1}{2}\left[t(t-1)(t-2)\cdots\left(t/2^L+1\right)\right] >>1,\ t>3 \tag{7}$$

因此，求秩运算的计算复杂度大幅减小。同时为得到所有的误码，需求解 2^{L-1} 个 $t/2^{L-1}$ 元方程组，其计算量远低于求解一个 t 元方程组的计算量。因此，单就求解误码数目一项而言，子序列的计算复杂度远远低于原接收码序列的计算复杂度。

实际应用中，RS 码主要用于突发和随机错误的纠错，每个子序列均含有误码的概率较小。逐层分解后，部分子序列误码数目可能为 0，因此，在实际中算法复杂度得到进一步的降低。同样，分解后误码均出现在同一个子序列中也较为少见，若发生此种情况，可根据误码的概率分布调整分解方案，以达到降低复杂度的目的[1]。

(3)关键方程的简化求解。采用无求逆 iBM 算法在时间上可将 RS 码计算的关键路径由 $2(T_{\text{mult}}+T_{\text{add}})$ 缩短到 $(T_{\text{mult}}+T_{\text{add}})$ 。其中， T_{mult} 和 T_{add} 分别是有限域乘法器和加法器的时延，从而缩减 RS 码纠错运算时间。

4 算法测试及应用

4.1 基于 AWGN 信道数据的算法对比测试

4.1.1 计算复杂度对比 实验数据取自随机生成的 AWGN[16](Additive White Gaussian Noise)加性高斯白噪声信道，误码率($\text{SER}=Q(\sqrt{2\gamma_b})$, γ_b 为信噪比)[17]为 10^{-3} (调制方式为 BPSK)的不同码长的各 1000 组数据。分别使用传统算法和本文算法在运算时间进行对比，结果见表 4。

如表 4 所示，在误码率为 10^{-3} 条件下，基于 DFT 变换和 BM 算法的 RS 纠错算法的矩阵维数较高，导致计算次数较高，运算时间较长。而本文算法在

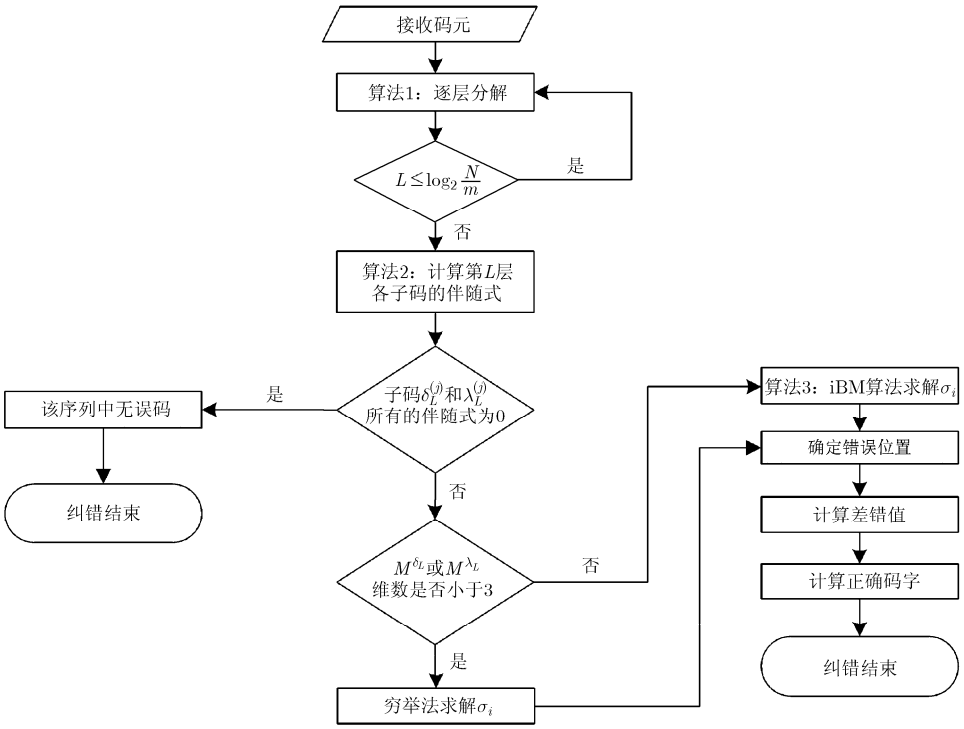


图 2 基于数据逐层分解的纠错算法流程图

表 4 运算时间对比统计结果(μs)

算法	阶数 m				
	4	5	6	7	8
BM 算法	1.894	2.021	2.475	2.964	3.627
DFT 算法	2.036	3.244	3.846	4.073	4.576
本文算法	0.837	1.257	1.389	1.621	1.885

降低求解矩阵维数的基础上进行计算，计算复杂度与其他两种算法相比，运算时间平均可缩短 1～1.5 倍左右。

4.1.2 误码纠正率比较　误码纠正率是指已纠正的误码数占所有误码数的比例。本文对不同码长的码字及不同的误码率中使用 RS 码对数据进行纠错，并与 DFT 算法进行误码纠正率对比实验，结果如图 3 所示。

图 3 的实验结果表明，在不同误码率的信道中，本文算法误码纠正率能够达到 95%以上，高于 DFT 算法。究其原因，DFT 算法在转向频域时，输入码有所亏损，致使能够纠正的误码数减少。综上，面对 AWGN 信道数据，本文算法纠错性能优于 DFT 算法。

4.2 2 维码纠错应用举例

2 维条码长期暴露于印刷品表面或复杂的工况当中，很容易受到污损破坏。为了达到预期的识别效果，除了要设计合理条码结构外，还需要采用纠错能力强的纠错算法进行纠错。因此，2 维条码采用 RS 码进行编码、译码和纠错[18]。

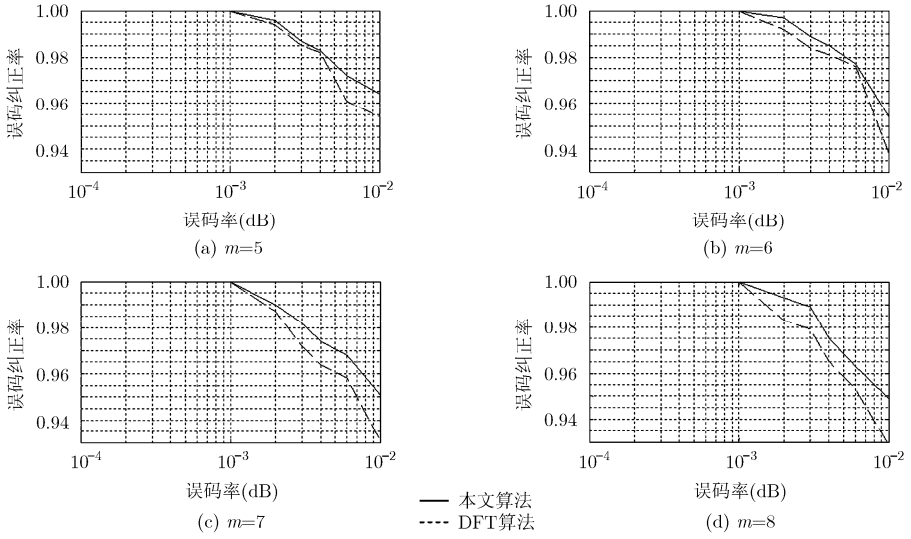
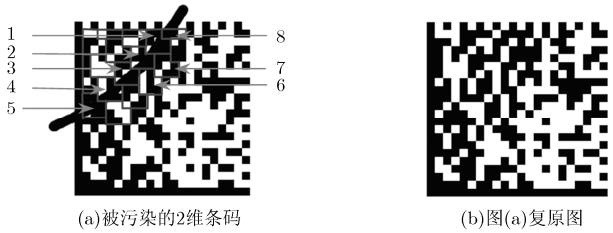


图 3 纠错率对比

4.2.1 被污染 2 维条码的纠错　图 4(a)为被墨迹污染的 2 维条码图像，编码规则为 ECC200，其条码区域大小为 20×20，接收码中含有 12 bit 错误，隶属于 8 个码字(图 4(a)中灰色框体划分出具体码字分割情况，并标示错误码字所在)，使用 RS(127,107,10) 编码[19]，其可纠错最大数为 10 个码字。

对图 4 中被污染条码分别使用传统 BM 算法、DFT 算法以及本文算法进行纠错，耗时为 10.257 ms, 7.149 ms, 1.189 ms，由于未超过最大纠错数，3 种算法在纠错率上并无差别。为验证算法的有效性，本文使用尺寸大小不一、编码参数不同、污染程度不同的 50 幅 2 维条码图像作为实验样本，分别使用本文算法、DFT 算法以及 BM 迭代算法对进行纠错，统计实验结果如表 5 所示。



(a)被污染的2维条码　　(b)图(a)复原图

图 4 被污染的 2 维条码纠错效果

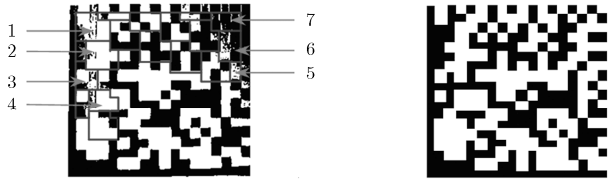
被污染的 2 维条码中错误码字成片出现，经过逐层分解后，误码被分散到不同的子序列中，因而可快速定位到误码位置。此处纠错率是指已纠正的

表 5 各算法在污染 2 维码图像中的性能比较

算法	乘法次数	加法次数	平均运算时间(ms)	纠错率(%)
BM 算法	57281	64740	12.810	89.2
DFT 算法	26516	37506	9.813	90.4
本文算法	7362	10937	2.762	91.1

误码数占所有误码数的比例的平均值，此数值小于100%是由于当误码数超出最大纠错数时，算法无能力检测出所有误码。表 5 中统计数据表明本文算法在不降低最大纠错能力的前提下，运算时间分别比BM 算法和 DFT 算法平均提高约 3.64 倍和 2.55 倍，运算复杂度大大降低。

4.2.2 非均匀光照下 2 维条码的纠错 图 5(a)为非均匀光照下 2 维条码典型代表，条码信息区域大小为 16×16。由于光照不均，条码区域出现模糊和缺损，共出现 24 bit 错误，隶属于 7 个码字。图 5(a)采用 RS(63,49,7)码编码，图中使用灰色框体划分出具体码字分割情况，并依次标示错误码字所在。



(a)非均匀光照引起的条码缺失 (b)图(a)复原图

图 5 非均匀光照引起的缺失条码纠错效果

对图 5 中缺损条码分别使用本文算法、DFT 算法以及传统 BM 算法进行纠错，耗时分别为 0.897 ms, 2.902 ms, 4.757 ms，纠错率均为 100%。同理对 50 幅尺寸、参数各异的非均匀光照条件下的 2 维条码分别使用 3 种算法进行纠错。统计实验结果如表 6 所示。

表 6 各算法在非均匀光照 2 维码图像中的性能比较

算法	乘法次数	加法次数	平均运算时间(ms)	纠错率(%)
BM 算法	75109	85643	14.654	92.1
DFT 算法	36516	48873	10.362	93.5
本文算法	6632	12184	3.409	94.7

表 6 中统计数据表明，针对非均匀光照条件下 2 维码中的错误码字，本文算法在运算复杂度和运算时间均优于其他两种算法。其中，运算时间较 BM 算法提高 3.29 倍，而与 DFT 算法相比也提高了 2.04 倍。

综上所述，本文算法应用在 2 维条码纠错时，在纠错时间及计算复杂度等方面优于传统算法。采用逐层分解变换对条码数据进行分解后，大幅度降低了伴随式和关键方程的计算维数，从而减少了运算时间和复杂度，与 BM 迭代算法和 DFT 算法相比优势明显。

5 结束语

RS 码纠错算法是经典的数据纠错算法之一。针对其计算复杂度高、运算时间长等弱点，本文提出从接收数据源头对数字序列进行逐层分解，设置分解结束条件，减少伴随式的计算长度和求解伴随矩阵秩的计算量。根据伴随矩阵维数，自适应选择穷举法或 iBM 迭代算法求解关键方程，进一步地降低计算复杂度。同时，误码数判断机制的提出，也为减少运算时间贡献力量。与其他算法相比，本文算法在应用实例中运算时间短，实时性好，可有效复原含加性白噪声的信道数据和 2 维条码中因局部受损造成的错误。

参 考 文 献

[1] 阔永红, 曾伟涛, 陈健. 基于概率逼近的本原 BCH 码编码参数的盲识别方法[J]. 电子与信息学报, 2014, 36(2): 332–339.
Kuo Yong-hong, Zeng Wei-tao, and Chen Jian. Blind identification of primitive BCH codes parameters based on probability approximation[J]. *Journal of Electronics & Information Technology*, 2014, 36(2): 332–339.

[2] Couvreur A, Gaborit P, Gauthier-Umaña V, *et al.*. Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes[J]. *Designs, Codes and Cryptography*, 2014, 73(2): 641–666.

[3] Lin Shu and Costello D J. Error Control Coding [M]. 2nd Edition, London, UK: Prentice Hall, 2004: 153–175.

[4] Tilavat V and Shukla Y. Simplification of procedure for decoding Reed-Solomon codes using various algorithms: an introductory survey[J]. *International Journal of Engineering Development and Research*, 2014, 2(1): 279–283.

[5] Wachter-Zeh A, Zeh A, and Bossert M. Decoding interleaved Reed-Solomon codes beyond their joint error-correcting capability[J]. *Designs, Codes and Cryptography*, 2014, 71(2): 261–281.

[6] Boito P. The Euclidean algorithm[J]. *Structured Matrix Based Methods for Approximate Polynomial*, 2009, 15(1): 45–58.

[7] Park J I and Lee H. Area-efficient truncated Berlekamp-Massey architecture for Reed-Solomon decoder[J]. *Electronics Letters*, 2011, 47(4): 241–243.

[8] Chen Y H, Truong T K, Chang Y, *et al.*. Algebraic decoding of quadratic residue codes using Berlekamp-Massey algorithm[J]. *Journal of Information Science and Engineering*, 2007, 23(1): 127–145.

[9] Sarwate D V and Shanbhag N R. High-speed architectures for Reed-Solomon decoders[J]. *IEEE Transactions on VLSI Systems*, 2001, 9(5): 641–655.

[10] Lin T C, Truong T K, Chang H C, *et al.*. A future simplification of procedure for decoding nonsystematic Reed-Solomon codes using the Berlekamp-Massey algorithm[J]. *IEEE Transactions on Communications*, 2011, 59(6):

1555–1562.

[11] Truong T K, Chen P D, Wang L J, *et al*.. Fast, prime factor, discrete Fourier transform algorithms over GF(2^m) for $8 \leq m \leq 10$[J]. *Information Sciences*, 2006, 176(1): 1–26.

[12] Schmidt G, Sidorenko V R, and Bossert M. Collaborative decoding of interleaved Reed-Solomon codes and concatenated code designs[J]. *IEEE Transactions on Information Theory*, 2009, 55(7): 2991–3012.

[13] 孙小钧, 刘晓健, 赵春明. 迭代译码的级联 Reed-Solomon 乘积码与卷积码[J]. 电子与信息学报, 2009, 31(12): 2917–2921.
Sun Xiao-jun, Liu Xiao-jian, and Zhao Chun-ming. Concatenated Reed-Solomon product code/convolutional code with iterative decoding[J]. *Journal of Electronics & Information Technology*, 2009, 31(12): 2917–2921.

[14] 邱昕, 张浩, 亓中瑞, 等. 一种高速自适应 Reed-Solomon 译码结构及其 VLSI 优化实现[J]. 电子与信息学报, 2009, 31(2): 484–488.
Qiu Xin, Zhang Hao, Qi Zhong-rui, *et al*.. An architecture and VLSI implementation for adaptive Reed-Solomon decoder[J]. *Journal of Electronics & Information Technology*, 2009, 31(2): 484–488.

[15] Wang K, Tang Z, Song Z, *et al*.. Verilog HDL optimisation design and simulation for modified inversionless Berlerkamp-Massey algorithm and the multiplier over canonical field[J]. *International Journal of Mobile Network Design and Innovation*, 2013, 5(1): 51–62.

[16] Van V T, Mita S, Li J, *et al*.. Bit-level soft-decision decoding of triple-parity Reed-Solomon codes through automorphism groups[J]. *IEEE Communications Letters*, 2013, 17(3): 553–556.

[17] Rahman M M and Enam F. Secure message transmission over wireless communication[J]. *Research Journal of Physical and Applied Sciences*, 2013, 2(3): 30–35.

[18] Dychka I A, Novosad M V, and Grybok T Y. Data conversion in creation and processing of multicolored graphic codes[J]. *Radio Electronics and Communications Systems*, 2013, 56(7): 335–344.

[19] Automatic identification and data capture techniques. ISO/IEC 16022-Data matrix bar code symbology specification[S]. Switzerland, IHS, 2006.

王　娟：女，1983 年生，博士生，研究方向为 DPM 工业 2 维码识别及其纠错.

王　萍：女，1955 年生，教授，博士生导师，主要研究方向为计算机视觉、运动目标的识别、跟踪与理解；模式识别、图像识别方法、分类特征的分析与提取.