Physical Communication 20 (2016) 48–60

Contents lists available at ScienceDirect

Physical Communication

journal homepage: www.elsevier.com/locate/phycom

Full length article

Algebraic soft decoding of Reed–Solomon codes with improved progressive interpolation

Yi Lyu, Li Chen *

School of Information Science and Technology, Sun Yat-sen University, Guangzhou, 510006, China

ARTICLE INFO

Article history:
Received 25 May 2015
Received in revised form 19 October 2015
Accepted 1 June 2016
Available online 10 June 2016

Keywords:
Algebraic soft decoding
Complexity reduction
Koetter–Vardy algorithm
Re-encoding transform
Reed–Solomon codes

ABSTRACT

The algebraic soft decoding (ASD) algorithm for Reed–Solomon (RS) codes can correct errors beyond the half distance bound with a polynomial time complexity. However, the decoding complexity remains high due to the computationally expensive interpolation that is an iterative polynomial construction process. By performing the interpolation progressively, the progressive ASD (PASD) algorithm can adapt the decoding computation to the need, leveraging the average complexity of multiple decoding events. But the complexity reduction is realised at the expense of system memory, since the intermediate interpolation information needs to be memorised. Addressing this challenge, this paper proposes an improved PASD (I-PASD) algorithm that can alleviate the memory requirement and further reduce the decoding complexity. A condition on expanding the set of interpolated polynomials will be introduced, which excepts the need of performing iterative updates for the newly introduced polynomial. Further incorporating the re-encoding transform, the I-PASD algorithm can reduce the decoding complexity over the PASD algorithm by a factor of 1/3 and its memory requirement is at most half of the PASD algorithm. The complexity and memory requirement will be theoretically analysed and validated by numerical results. Finally, we will confirm that the complexity and memory reductions are realised with preserving the error-correction capability of the ASD algorithm.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Reed–Solomon (RS) codes are widely used in digital communications and storage systems. The conventional unique decoding algorithms for RS codes include Berlekamp–Massey (BM) algorithm [1] and Welch–Berlekamp (WB) algorithm [2,3]. For an (n, k) RS code, where n and k are the length and dimension of the code, respectively, its error-correction capability is limited by $\lfloor \frac{d-1}{2} \rfloor$ where $d = n - k + 1$ is the code's minimum Hamming distance. The algebraic list decoding algorithm [4,5] improves the error-correction capability by a curve-fitting decoding approach, thereby correcting errors beyond the half distance bound. In this paper, it is referred as the algebraic hard decoding (AHD) algorithm. The algebraic soft decoding (ASD) algorithm [6] was later proposed, enhancing the AHD algorithm by introducing an extra process that maps the reliability information to the interpolation multiplicity information. Being able to utilise the soft information provided by the channel, it outperforms the AHD and the unique decoding algorithms.

In algebraic decodings, interpolation that is an iterative polynomial construction process [7–9] dominates the computational complexity and there exists various complexity reduction approaches. In [10], interpolation complexity is reduced by eliminating the interpolated polynomials with a leading order that is greater than the number of interpolation constraints. In [11], a low-complexity Chase (LCC) algebraic decoding algorithm was

* Corresponding author.
E-mail addresses: lvyi3@mail2.sysu.edu.cn (Y. Lyu), chenli55@mail.sysu.edu.cn (L. Chen).

http://dx.doi.org/10.1016/j.phycom.2016.06.001
1874-4907/© 2016 Elsevier B.V. All rights reserved.

proposed. It reduces complexity by exploiting the similarity among the interpolation test-vectors. In [12], by formulating the AHD as a rational curve-fitting problem utilising the outcome of the BM algorithm, i.e., the error locator and error-correction polynomials, it results in a significantly reduced interpolation multiplicity. The re-encoding transform [13–15] is another important approach. Interpolation complexity can be reduced by choosing k received symbols to perform re-encoding, alleviating the iterative polynomial construction computation. Meanwhile, it should be mentioned that there exists other efficient realisations for the interpolation problem, such as the Lee–O'Sullivan approach [16] and its modified variant [17], and the Beelen–Brander approach [18].

The above mentioned approaches were proposed to reduce the computation of a single decoding event. By further observing the fact that different decoding events may require different error-correction capability, the progressive ASD (PASD) [19] algorithm was proposed aiming to reduce the average decoding complexity of multiple decoding events. It functions with a progressively enlarged designed factorisation output list size (OLS), leading to a gradually strengthened error-correction capability. By enlarging the factorisation OLS, both the cardinality of the interpolated polynomial set and the size of each polynomial will be increased. Since such an expansion is realised at the cost of interpolation computation, the PASD algorithm adapts the computation of each individual decoding event to the need, leveraging the average decoding complexity. Other similar efforts include the work of [20] that analyses the interpolation cost's dependence on the received word's error weight. It also proposed an interpolation algorithm that gives priority to update the polynomials that are more likely to be chosen for factorisation. More recently, a multi-trial AHD approach was proposed in [21]. It performs a similar progressive decoding based on the Beelen–Brander interpolation [18].

However, the PASD algorithm's merit in reducing the average decoding complexity is realised at the expense of system memory, since the intermediate interpolation information needs to be memorised. In particular, when a new polynomial is introduced into the set, it needs to be iteratively updated w.r.t. the constraints which have been satisfied by the existing polynomials of the set, during which the intermediate interpolation information is needed. Addressing this challenge, this paper proposes an improved PASD (I-PASD) algorithm that can alleviate the memory requirement and further reduce the decoding complexity. In particular, a condition on expanding the polynomial set will be established such that the newly introduced polynomial is excepted from performing the iterative updates. It further incorporates the re-encoding transform, offering a memory requirement that is at most half of the PASD algorithm and a complexity reduction over the PASD algorithm by a factor of 1/3. Our complexity analysis shows that when the decoding terminates with a factorisation OLS that is greater than one, such a complexity reduction is mainly attributed to the new polynomial set expansion. Both of the complexity and memory analyses of the I-PASD algorithm will be validated by numerical results. Finally, our simulation results confirm that the proposed low complexity algorithm preserves the error-correction capability of the ASD algorithm.

The rest of this paper is organised as the follows. The background knowledge of the paper is presented in Section 2. Section 3 presents I-PASD algorithm. The memory and complexity analyses of the new proposal will be presented in Sections 4 and 5, respectively. The proposed algorithm's error-correction performance will be presented in Section 6. Finally, Section 7 concludes the paper.

2. Background knowledge

2.1. Encoding of RS codes

Let $\mathbb{F}_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$ denote the finite field of size q, and $\mathbb{F}_q[x]$ and $\mathbb{F}_q[x, y]$ denote the univariate and bivariate polynomial rings defined over $\mathbb{F}_q$, respectively. Given a message vector $\underline{\mu} = (\mu_0, \mu_1, \ldots, \mu_{k-1}) \in \mathbb{F}_q^k$, the message polynomial $\mu(x) \in \mathbb{F}_q[x]$ can be written as:

$$\mu(x) = \mu_0 + \mu_1 x + \cdots + \mu_{k-1} x^{k-1}. \tag{1}$$

A codeword $\underline{c}$ of an (n, k) RS code is generated by:

$$\begin{aligned}\underline{c} &= (c_0, c_1, \ldots, c_{n-1}) \\ &= (\mu(\chi_0), \mu(\chi_1), \ldots, \mu(\chi_{n-1})),\end{aligned} \tag{2}$$

where $\underline{c} \in \mathbb{F}_q^n$. $\chi_0, \chi_1, \ldots, \chi_{n-1}$ are n distinct elements of $\mathbb{F}_q$ and they are called the code locators.

2.2. The ASD algorithm and its progressive variant

It is assumed that an RS codeword is modulated and transmitted through a memoryless channel, e.g., the additive white Gaussian noise (AWGN) channel. Given a received vector $\underline{y} \in \mathbb{R}$, the $q \times n$ reliability matrix $\mathbf{\Pi}$ can be obtained, whose entry $\pi_{ij} = \Pr[c_j = \alpha_i \mid \underline{y}]$. Matrix $\mathbf{\Pi}$ is then transformed into a multiplicity matrix $\mathbf{M}$ of the same size [6] and its entry m_{ij} represents the interpolation multiplicity for the point (χ_j, α_i), where $\chi_j \in \mathbb{F}_q$ and $c_j = \mu(\chi_j)$. Given a polynomial $Q(x, y) = \sum_{a,b} Q_{ab} x^a y^b \in \mathbb{F}_q[x, y]$ and a nonnegative integer pair (r, s), the (r, s)th Hasse derivative evaluation of Q at point (χ_j, α_i) is defined as [22]:

$$D_{r,s}(Q(x, y))|_{(\chi_j, \alpha_i)} = \sum_{a \geq r, b \geq s} \binom{a}{r} \binom{b}{s} Q_{ab} \chi_j^{a-r} \alpha_i^{b-s}. \tag{3}$$

Q interpolates point (χ_j, α_i) with a multiplicity of m_{ij} if $D_{r,s}(Q(x, y))|_{(\chi_j, \alpha_i)} = 0$ for all the (r, s) pairs with $r + s < m_{ij}$. In the following, we will simply use $(r, s)_{ij}$ to denote the interpolation constraint that implies $D_{r,s}(Q(x, y))|_{(\chi_j, \alpha_i)}$. The number of interpolation constraints defined by matrix $\mathbf{M}$ is

$$\mathcal{C}(\mathbf{M}) = \frac{1}{2} \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} m_{ij}(m_{ij} + 1). \tag{4}$$

In decoding an (n, k) RS code, monomials $x^a y^b$ are organised by the $(1, k-1)$-revlex order.[1] Given a poly-

[1] The $(1, k-1)$-weighted degree of monomial $x^a y^b$ is defined as: $\deg_{1,k-1} x^a y^b = a + (k-1)b$. Given two distinct monomials $x^{a_1} y^{b_1}$ and $x^{a_2} y^{b_2}$, we have $\mathrm{ord}(x^{a_1} y^{b_1}) < \mathrm{ord}(x^{a_2} y^{b_2})$, if $a_1 + (k-1)b_1 < a_2 + (k-1)b_2$, or $a_1 + (k-1)b_1 = a_2 + (k-1)b_2$ and $b_1 < b_2$.

Download English Version:

https://daneshyari.com/en/article/466262

Download Persian Version:

https://daneshyari.com/article/466262

Daneshyari.com